UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECB3203 - PROGRAMMING FOR BIOINFORMATICS

GROUP PROJECT

PNEUMONIA CLASSIFICATION THROUGH X-RAY IMAGES

Semester 1 2025/2026

Section 01

NAME	MATRIC NUMBER
AMELIA ADLINA BINTI AZRUL	A23CS0043
NUR AINA SYAFINA BINTI KAMASUAHADI	A23CS0152
NURUL ATHIRAH SYAFIQAH BINTI RAZALI	A23CS0163
PHAVANEE KATRIYA PHON-AMNUAISUK	A23CS0170

LECTURER: DR. SEAH CHOON SEN

Date: 26th December 2025

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Pneumonia is a respiratory disease characterised by inflammation of the lung alveoli, often accompanied by the accumulation of fluid or pus that interferes with normal oxygen exchange. The disease may be caused by bacterial, viral, or fungal infections, and its clinical presentation varies significantly among patients (American Lung Association, 2025). Symptoms such as cough, fever, and shortness of breath often overlap with those of common respiratory illnesses including influenza and the common cold, making early and accurate diagnosis challenging (Cleveland Clinic, 2025).

Chest X-ray imaging is commonly used to assist in pneumonia diagnosis due to its affordability and widespread availability (Cleveland Clinic, 2025). However, manual interpretation of X-ray images by radiologists can be time-consuming and susceptible to human error, especially during periods of high patient volume. With increasing demand on healthcare services, automated diagnostic tools powered by machine learning present a promising solution. This project therefore explores the use of machine learning models to classify chest X-ray images into ‘Normal’ and ‘Pneumonia’ categories, aiming to support clinical decision-making and improve diagnostic efficiency.

1.2 PROBLEM BACKGROUND

Despite advances in medical imaging, pneumonia remains difficult to diagnose accurately in its early stages. The variability of symptoms between patients and their similarity to other respiratory infections contribute to diagnostic uncertainty (NHLBI, 2022). Even when chest X-rays are used, subtle radiographic features can be missed, particularly under high workload conditions or limited specialist availability. In Malaysia, pneumonia has emerged as a major public health concern. In 2023, it was reported as the leading cause of medically certified deaths, accounting for 18,181 cases or 15.2% of total deaths (DEPARTMENT OF STATISTICS MALAYSIA, 2025), surpassing heart disease for the first time in over two decades. This alarming statistic highlights the urgent need for timely and accurate diagnostic support, particularly in resource-constrained healthcare settings.

The increasing number of pneumonia cases in Malaysia places significant pressure on healthcare facilities, leading to high patient volumes and limited radiological resources. Left unchecked, the healthcare system might miss cases of pneumonia in the population, further causing deaths by pneumonia. These challenges highlight the need for machine learning-based approaches that can assist clinicians by providing fast, accurate, and reliable classification of pneumonia from chest X-ray images. Encouragingly, the successful deployment of AI-based lung cancer screening tools in selected Malaysian government clinics (HAMSUDDIN, 2025) demonstrates the feasibility of integrating machine learning into clinical workflows.

Even so, training machine learning models to tackle clinical problems are difficult, as the law and ethics surrounding patient data collection mean that there is a limited amount of medical imaging data out there to train models with. The problem of scarcity, detailed by Gr(Gröger et al., 2025)ger et. al.. is not a new one, and several methods have been developed in order to overcome the problem of imbalance and small datasets (Gröger et al., 2025).

Given these challenges, there is a clear need for automated systems that are capable of assisting healthcare professionals by rapidly and accurately identifying pneumonia from chest X-ray images whilst tackling the problem of imbalanced data.

1.3 RESEARCH AIM

The aim of this study is to investigate the most effective methods for improving machine learning model performance in pneumonia detection while minimizing computational costs and training time.

1.4 RESEARCH QUESTION

To what extent does class imbalance affect model evaluation scores and computational efficiency?

1.5 RESEARCH OBJECTIVES

The main objectives of this study are as follows:

(a) To train machine learning models to recognize and learn discriminative patterns from chest X-ray images for the detection of pneumonia.

(b) To compare the performance of models when varying methods of handling class imbalance are used.

(c) To evaluate the effectiveness of each method of handling class imbalance including accuracy, precision, recall and F1-score.

1.6 RESEARCH SCOPE

This study focuses on the binary classification of chest X-ray images into Normal and Pneumonia categories using supervised machine learning techniques. The dataset employed is a publicly available Kaggle dataset originally sourced from Mendeley, which has been pre-organised into training, testing, and validation subsets consisting of 5216, 624, and 16 images respectively. We selected this dataset as all chest radiographs from the Mendeley dataset that were of low quality or unreadable were removed. Furthermore, the Kaggle authors of this dataset have been graded by two expert physicians before being cleared for upload. The scope of this project is limited to analysing posterior–anterior chest X-ray images and does not include other imaging modalities or multi-class disease classification.

The model architecture considered in this study is a custom Convolutional Neural Network (CNN). CNNs are well suited for medical image analysis due to their ability to automatically extract spatial features such as edges, textures, and patterns from images. In this project, the CNN serves as a baseline deep learning model, allowing the study to evaluate how a relatively simple, task-specific architecture performs in detecting pneumonia-related features from chest X-ray images. The CNN is trained from scratch, and the model performance is repeatedly evaluated when different techniques are employed during pre-processing, model parameter tweaking, and post-processing.

The model in this project will be trained on a Google Colab connected to a local server, hosted by a laptop with 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz (2.80 GHz), 16.0 GB installed RAM and running 64-bit Windows 11 operating system. The project will not explore the use of GPUs as a way to speed up model training.

The scope is strictly limited to posterior-anterior chest X-ray images, and multi-class disease classification or other imaging modalities are not considered.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

When patients present with pneumonia symptoms, a chest X-ray is typically ordered as part of the doctor's assessment to formulate a diagnosis (American Lung Association, 2025; Cleveland Clinic, 2025). This is because chest X-rays are one of the cheapest and most accessible ways for hospitals around the world to confirm the presence of pneumonia. However, errors persist in diagnosis, and can be affected by a physician's cognition or system related issues, among other things (Shimizu & Tokuda, 2012). Hence, machine learning tools that are able to be used clinically will speed up diagnosis and lessen the burdens on the healthcare system.

2.2 PROBLEM FORMULATION

The primary objective of this study is to develop a system that can automatically classify chest X-ray images into one of two categories, which are 'Pneumonia' or 'Normal'. Accurate classification is critical, as early detection of pneumonia can significantly reduce patient morbidity and mortality.

2.2.1 RESEARCH DOMAIN

This study falls under the research domain of computer vision and medical image analysis. This study aims to extract meaningful information from medical imaging data for the benefit of the medical community. In recent years, deep learning techniques such as convolutional neural networks (CNNs) have become one of the dominant approaches in the field. This is because CNNs have the ability to automatically learn hierarchical features representations. Other approaches include residual neural networks (ResNet), pre-trained networks and the use of vision transformers (Zhou et al., 2021). These methods have been widely applied to tasks such as disease detection, image classification, and lesion segmentation in medical imaging.

Within this domain, pneumonia detection from chest X-ray images is a popular problem because it has the potential to support clinical decision-making and alleviate the workload of radiologists. A major challenge in this domain is data imbalance, where the number of normal images often exceeds pneumonia images, and the limited size of datasets available for research. To address these issues, researchers have explored techniques such as transfer learning, data augmentation, and class weighting, which form the basis for the proposed methodology in this study.

2.2.2 DESCRIPTION OF RELATED STUDIES

Researchers have, in the past, trained deep learning models to classify chest X-rays in order to be used in clinical settings, such as in decision support systems.

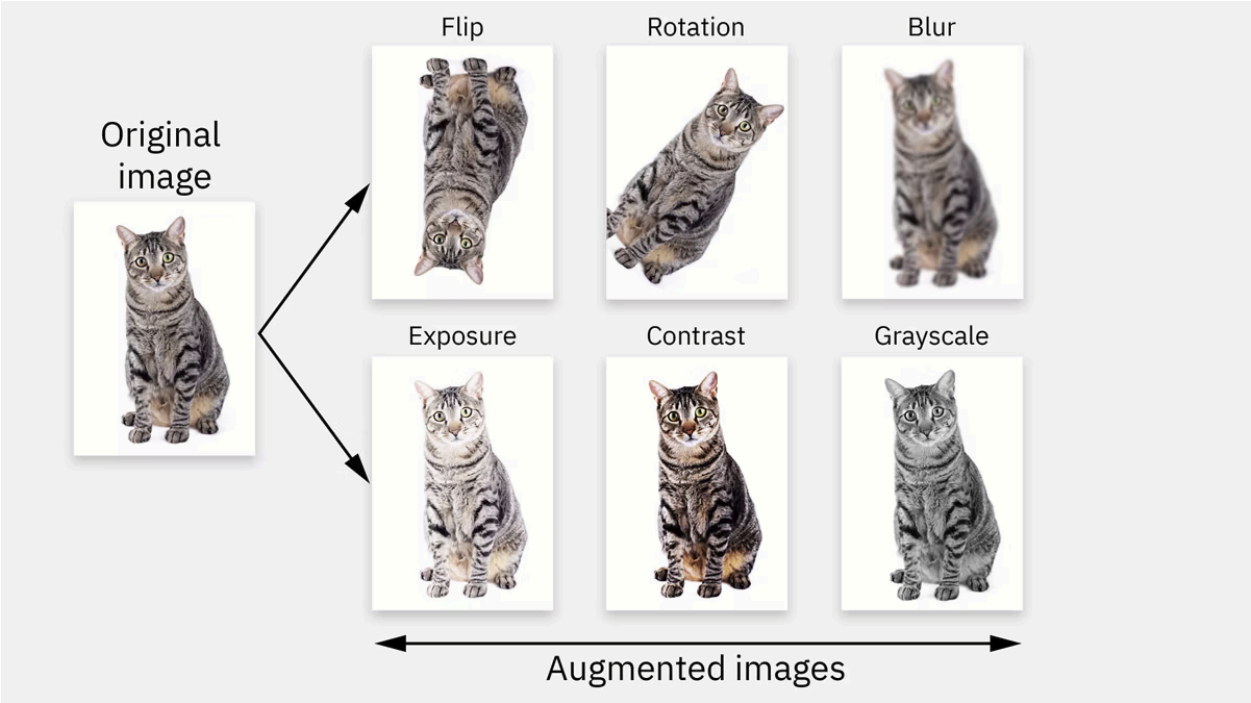
In order to tackle imbalanced datasets, researchers have turned to resampling methods in order to balance the classes in their dataset (Gröger et al., 2025). There are two ways to balance datasets with resampling: the minority class can be oversampled so that it has the same number of data as the majority class, or the majority class can be undersampled so that it has the same number of data as the minority class. These techniques are called oversampling and undersampling, respectively.

Some researchers have turned to oversampling minority classes with SMOTE to reduce overfitting (Chawla et al., 2011). However, this technique has been found to only help weak learners, and does not improve performance of strong learners (Elor & Averbuch-Elor, 2022). In the case of image classification, where the common approach is to use deep learning with CNNs, ViTs and ResNets, SMOTE might not be suitable. Other data-level techniques such as data augmentation are also commonly used and show promising results (Omoniyi et al., 2025). In another study on classifying chest X-rays dataset to detect pneumonia however, it is shown that data augmentation showed no significant improvements in accuracy, but AUC decreased with the use of data augmentation (Usman et al., 2025). The authors argued that the effect of data size could cause data augmentation to perform less effectively.

Augmenting or resampling data does not introduce any new information into the training pool. Instead of augmenting or resampling data, some researchers have turned to data synthesis techniques in order to tackle data imbalance. In 2014, Goodfellow et. al. introduced generative adversarial networks (GANs), which has been of great interest in order to create synthetic data to handle data imbalance. GAN-based medical image synthesis has been shown to increase CNN performance in classification, as found in (Frid-Adar et al., 2018). However, as with data synthesis techniques, GAN requires a lot of computational power.

Figure 1

Data augmentation applied to an image showing transformation such as flipping, rotation and scaling.



Note. From *'What is Data Augmentation?'* by Jacob Murel, 2024, IBM (https://www.ibm.com/think/topics/data-augmentation)

Yet other researchers have studied transfer learning in order to achieve greater accuracy with image datasets. Transfer learning is promising because it allows you to use less data and training time to achieve a desired metric or evaluation score. To create a good transfer learning model, there are some constraints: both learning tasks must be similar, and the source and target dataset must have data distributions that do not vary too greatly. There are techniques in research on finding out whether transfer learning would be suitable for your dataset and tasks, discussed in (W. Zhang et al., 2020). If these constraints are not met, transfer learning can negatively affect performance, also known as negative transfer.

Aside from data-level techniques and transfer learning, there are also algorithm-level techniques used by researchers in order to tackle data imbalance. One such technique is class weighting, which assigns different weights or penalties for misclassifications made of different classes. Oftentimes, class weights are calculated automatically using libraries like sci-kit learn, and the default setting calculates the 'balanced' class weight, or the class weights that are obtained by inversely scaling weights to class frequency, as shown in (1).

$$w_j = \frac{\text{total samples}}{K \times \text{samples in class } j} \tag{1}$$

An implementation of class weighting with CNN model by Razali et. al. shows promising results to identify plant deficiencies using image data of palm oil plant leaves (Razali et al., 2025). As with other techniques, class weighting has limitations: class weighting risks overfitting if the weights are too high, and it requires tuning for optimal performance. Another implementation of class weighting with an object detection model was shown by Y. Zhang et. al., who improved upon balanced class weighting with dynamic weighting (Y. Zhang et al., 2021). In dynamic weighting, the F1-score obtained at the end of each training batch is dynamically sampled and the weights of the class are modified.

2.3 PROPOSED SOLUTION

The three approaches discussed above (data-level techniques, transfer learning, and algorithm-level techniques) all have different advantages and limitations. For this project, we wish to compare the performance of undersampling, oversampling and class weighting to handle dataset imbalance on the chosen dataset. These approaches were chosen due to their simplicity in implementation, and the different ways in which they handle data imbalance. We predict that performance of class weighting based on F1, recall, and precision will be comparable to oversampling using the chosen dataset, and the performance of the model trained by undersampling will perform the worst.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 INTRODUCTION

This chapter presents the methodology adopted in this study to develop and evaluate machine learning models for pneumonia detection from chest X-ray images. The methodology is designed to ensure reproducibility, scientific rigor, and clear evaluation of model performance. It covers the research workflow, dataset preparation, preprocessing, handling class imbalance, model development, training, and evaluation metrics. The goal is to provide a comprehensive approach that supports accurate and efficient pneumonia classification while minimizing computational cost.

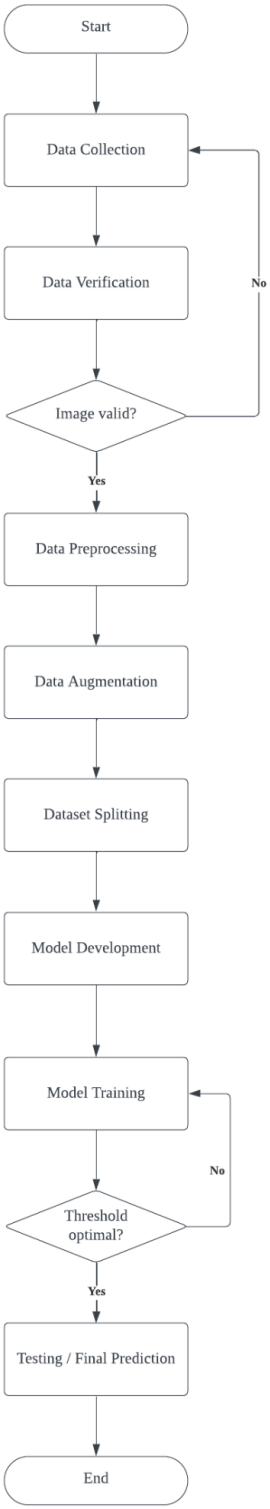
3.2 OPERATIONAL FRAMEWORK/RESEARCH WORKFLOW

The overall workflow of this study is illustrated in Figure 3.2.1. The process begins with dataset acquisition, where chest X-ray images are obtained from a publicly available Kaggle dataset. The dataset is then subjected to exploratory data analysis (EDA) to understand class distributions, image quality, and characteristics. Following EDA, preprocessing techniques such as resizing, normalization, and image verification are applied to prepare the dataset for model training.

To handle class imbalance, three strategies will be compared: undersampling (by removing samples from the majority class), oversampling (by producing augmented versions of the minority class), and class weighting (by assigning higher weights to misclassifications of the minority class when calculating model loss). A custom convolutional neural network (CNN) is used as the baseline model. The data goes through the necessary preprocessing steps before being used for training the model. The performance for the baseline methods and all other methods are evaluated using multiple metrics.

Figure 2

Operational workflow for pneumonia detection using X-ray images.



3.3 JUSTIFICATION

The methodology adopted in this study was chosen to ensure accurate, efficient, and reproducible pneumonia detection:

1. Dataset Selection	:	The Kaggle chest X-ray dataset was selected for its accessibility, expert-verified annotations, and adequate representation of both 'Normal' and 'Pneumonia' classes. This ensures reliable training and evaluation of machine learning models.
2. CNN Model	:	Convolutional Neural Networks (CNNs) were chosen due to their ability to automatically extract hierarchical image features, such as edges, textures, and patterns. CNNs are widely used in medical image analysis tasks and provide a balance between model performance and computational efficiency.
3. Handling Class Imbalance:		Class imbalance is common in medical datasets, which can bias models toward the majority class. Three strategies: undersampling, oversampling, and class weighting are selected. All the approaches are simple to implement and have demonstrated effectiveness in prior research, while allowing a comparison of their impact on model performance.
4. Evaluation Metrics	:	Accuracy, precision, recall, and F1-score were selected to evaluate model performance. Recall is prioritized to minimize missed pneumonia cases, which is critical in medical diagnosis. F1-score provides a balanced view of precision and recall, ensuring the model's predictions are both sensitive and reliable.

This methodology ensures a systematic approach from data preparation to performance evaluation, aligning with the research aim of optimizing model performance while maintaining computational efficiency.

3.4 PERFORMANCE MEASUREMENT

In order to evaluate the performance of data augmentation and class weighting, several evaluation metrics are used: accuracy, recall, precision, and F1-score.

Accuracy, as seen in (2), is used to measure the overall correctness of the classification results. However, since the dataset may exhibit class imbalance, precision and recall are also employed to provide a more reliable evaluation of the model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Precision, as shown in (3), reflects the model's ability to correctly identify positive instances. In the case of the dataset used, precision measures the accuracy of positive predictions, i.e. for all instances where the model predicts 'Pneumonia', how often is the prediction actually 'Pneumonia'?

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall, as seen in (4), measures the effectiveness of the model to detect all relevant instances. In the case of the dataset used, recall measures the ratio of correctly predicted positive instances to all positive instances, i.e. for all real instances of pneumonia, how many was the model able to detect?

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1, as seen in (5), is the harmonic mean between precision and recall. It is used to evaluate the performance of a machine learning model, especially in cases where the dataset is imbalanced. It provides a balanced measure between precision and recall.

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

For the purposes of medical diagnosis, recall is the more important metric, as it is better for the model to falsely predict a case of pneumonia than to miss it. However, precision will also be given attention, as a model with high recall and low precision may just be predicting positive and score high on recall. Furthermore, as the dataset chosen is imbalanced in the positive class (pneumonia), hence the model might have a bias towards positive cases, and measuring precision will help detect overfit. Finally, a good balance of precision and recall (i.e. high F1-score) will mean that a model would be able to help make diagnostic decisions without overburdening the healthcare system.

CHAPTER 4

RESEARCH DESIGN AND IMPLEMENTATION

4.1 INTRODUCTION

This chapter describes the research design and implementation of the proposed solution. It presents the overall system design, followed by the experimental setup used to evaluate the proposed approach. The parameters and testing methods employed in this study are also discussed.

4.2 PROPOSED SOLUTION

The proposed solution consists of three main components: data preprocessing, undersampling, oversampling, class weighting, and classification. Input data is first preprocessed to remove noise and normalize the format. Input data is also resized to a uniform size before being fed into the model. Depending on the phase of the experiment, training data is undersampled, oversampled, or class weighting is used in order to address the data imbalance. Finally, the model is fit on the data to produce the final prediction.

4.3 EXPERIMENT DESIGN

The project is conducted using a dataset obtained from Kaggle (*Chest X-Ray Images (Pneumonia)*, n.d.). The dataset has been divided into training, validation and testing sets by the author and the dataset distribution is as shown in Table 1. All components used in this experiment are listed in Table 2.

Table 1

Distribution of dataset according to set and class

Set/Class	Pneumonia	Normal	Total
Training	3875	1341	5216
Testing	390	234	624
Validation	8	8	16
Total	4273	1583	5856

Note. This table shows the data distribution of the dataset obtained from Kaggle.

Table 2

Components used in the experiment with details and purpose

Component	Details	Purpose
Core Libraries and Frameworks	TensorFlow / Keras	Build and train CNN models
	NumPy	Numerical computations and array manipulations
	Pandas	Dataset handling and preprocessing
	Matplotlib/Seaborn	Visualization of data, training metrics, and results
	OpenCV (cv2) / Pillow (PIL)	Image loading, preprocessing, resizing
	Scikit-learn	Metrics and evaluation (accuracy, precision, recall, F1-score, confusion matrix)
Version Control / Collaboration	Git and GitHub	Track changes, share code, and document project progress
Dataset	Kaggle Chest X-Ray Pneumonia Dataset	Contains labeled NORMAL and Pneumonia chest X-ray images

Note. This table shows the different components used in this experiment with details and purpose.

The evaluation scores using undersampling, oversampling, and class weighting is evaluated against a baseline approach to ensure a fair comparison. The baseline model is obtained through finding the best convolutional neural network suited for the task at different image sizes, and then preserving that model for evaluation.

All model training is run on a Google Colab connected to a local server, hosted by a laptop with 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz (2.80 GHz), 16.0 GB installed RAM and running 64-bit Windows 11 operating system.

4.4 PARAMETER AND TESTING METHOD

All experiments are conducted using a single run and the evaluation scores are recorded.

4.4.1 DATA COLLECTION AND IMPORTS

The pneumonia image dataset that will be used in this project is taken from Kaggle, whose author sourced it from Mendeley and only included images which were approved by expert physicians. The required packages and datasets for this project are imported before starting the project. Dataset import was done using the kagglehub package. The code for imports is provided in Appendix A.

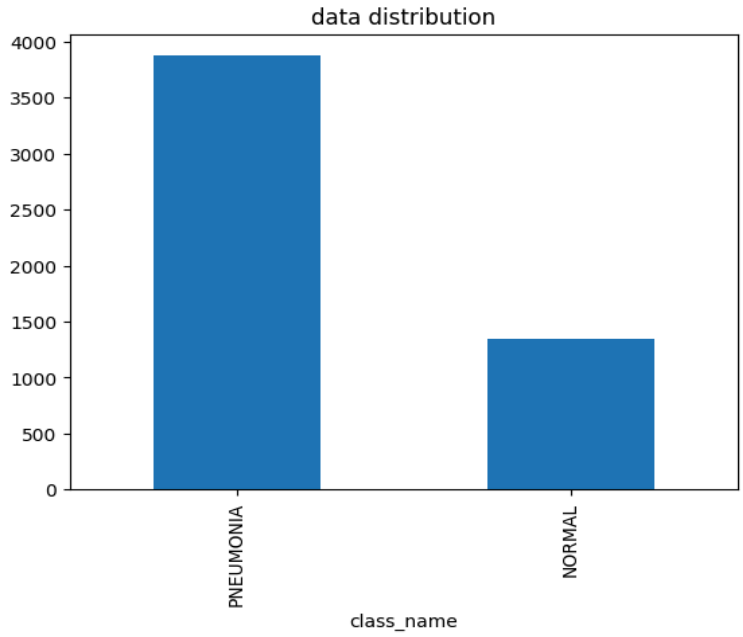
4.4.2 EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) was conducted on this dataset in order to understand the data better and find patterns, errors, and check data quality.

4.4.2.1 CLASS DISTRIBUTION ANALYSIS

Figure 3

Distribution of data across classes in test set

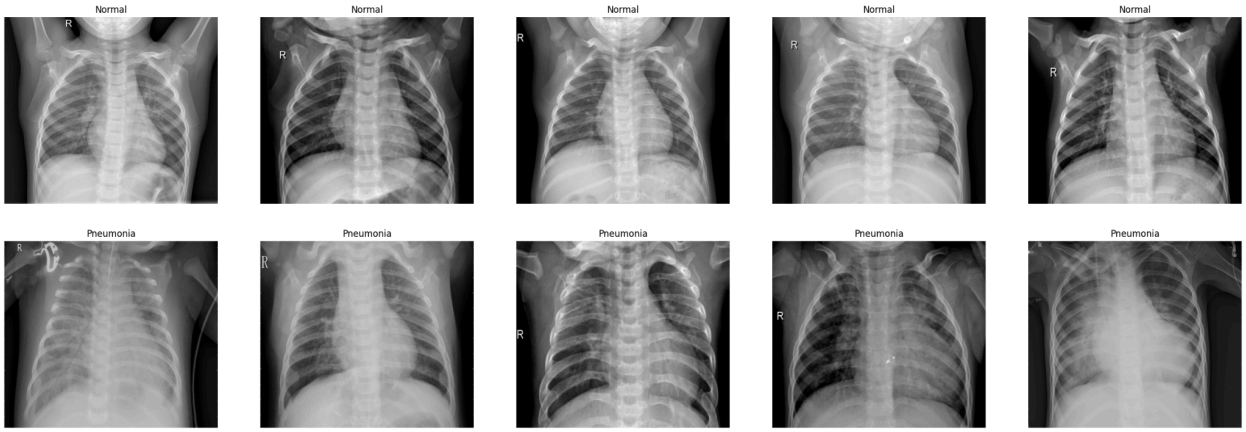


The training set obtained from the dataset is imbalanced, with the positive class (Pneumonia) having 3875 images, while the negative class (Normal) has 1341 images. The ratio between the two classes is about 3:1. The positive class is overrepresented, and may present a bias to the model.

4.4.2.2 VISUAL INSPECTION OF CHEST X-RAYS

Figure 4

Random sample of chest X-rays from test dataset

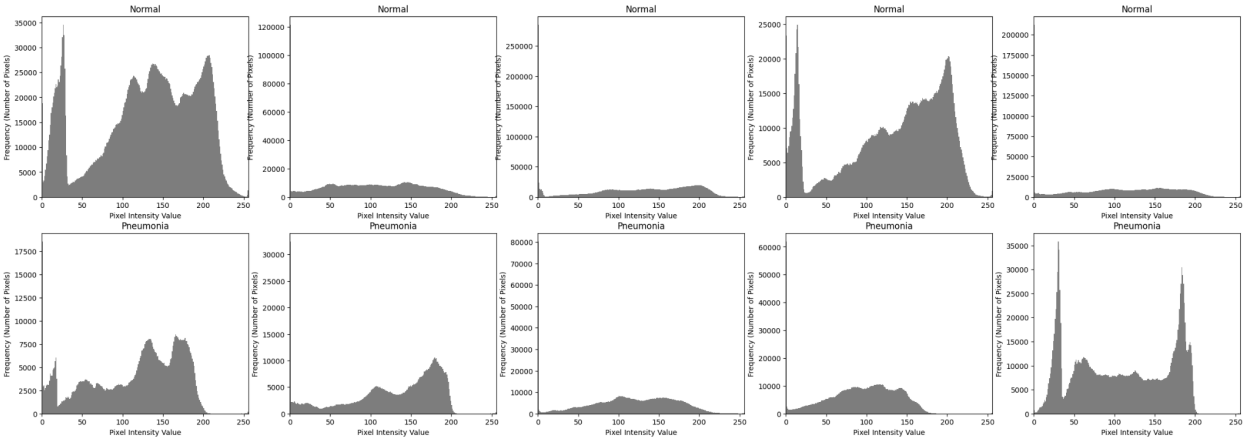


A random sample of 5 X-rays are obtained from the two classes, as shown in Figure 3. In some cases, it is quite easy to discern whether an X-ray might belong to pneumonia class, because of the cloudiness. In other cases, it is quite hard to discriminate between the classes.

4.4.2.3 PIXEL INTENSITY HISTOGRAM ANALYSIS

Figure 5

Histogram analysis of pixel intensities based on random sample of chest X-rays



A histogram analysis of the 10 X-rays sampled across the two classes are conducted. From the pixel intensities, patients with normal chest X-rays tend to have a wider spread of pixel values, although this is not always consistent. Pixel frequencies at different intensities are also inconsistent within classes, suggesting a need for normalization.

4.4.2.4 IMAGE DIMENSION SCATTER PLOT

Figure 6

Image sizes of training dataset on a scatter plot

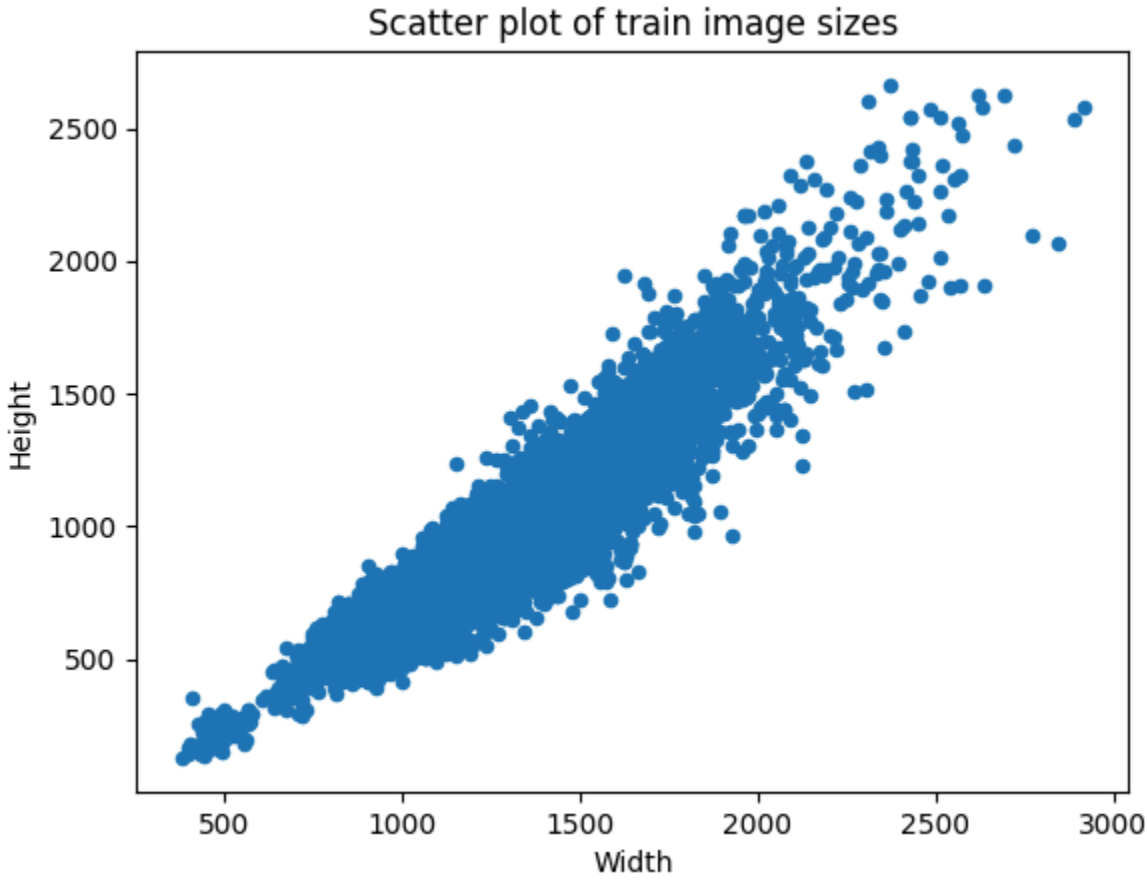


Table 3
Training set image sizes description

	Width	Height
count	5216.000000	5216.000000
mean	1320.610813	968.074770
std	355.298743	378.855691
min	384.000000	127.000000
25%	1056.000000	688.000000
50%	1284.000000	888.000000
75%	1552.000000	1187.750000
max	2916.000000	2663.000000

Charting the image sizes on a scatter plot reveals that the smaller images on the dataset are about 500x500 pixels, while the larger images tend to be about 2000x2000 pixels. From Table 3, we can see that the biggest width of an image is 2916 pixels, while the largest height is 2663 pixels, and the smallest width and height are 384 and 127 pixels respectively. Image sizes need to be consistent before placing into a model, and this shows that image resizing is required with this dataset.

4.4.3 DATA PRE-PROCESSING

Data pre-processing is an essential step in machine learning. Raw data is messy, and often does not meet model specifications, which makes models run into errors when training on the data. Pre-processing data cleans, transforms and organizes data to improve quality, accuracy, and consistency. Typically, data is pre-processed to handle missing values and remove noise in order to make it suitable for machine learning models.

4.4.3.1 IMAGE INTEGRITY AND HANDLING

Given that the chosen dataset is composed of images, image integrity has to be checked to ensure there are no corrupted files present. Only .jpg, .jpeg and .png files will be used for training, testing, and validation. The code for checking image integrity is provided in Appendix A

4.4.3.2 DATA NORMALIZATION

As seen during the exploratory analysis, the images have raw pixel values ranging from 0 to 255, although it is not consistent across classes. Data normalization is implemented to scale the value of each pixel between 0 to 1. This is done by dividing the value of each pixel by 255. Data normalization has the added benefit of allowing the model to converge faster, as the values are smaller.

4.4.3.3 RESIZING DATA

All images are resized to 128x128 pixels to ensure consistent input dimensions for the CNN model. This dimension was selected, as when testing between 128x128 pixels, 256x256 pixels and 512x512 pixels, it was found that the model trained on 128x128 pixels performed as well as when using larger image sizes, with evaluation scores (precision, recall) differing by about 2%. Furthermore, the models trained 3 times faster with the smaller image size, and hence this image size was chosen.

4.4.3.4 DATA AUGMENTATION

Data augmentation is a technique to increase the variety present in a dataset by creating modified copies of existing data, through techniques like width shift, height shift, blur, rotation, etc. Data augmentation serves the purpose of reducing overfit and enhancing a model's ability to generalize to new and unseen data. The methods that we used for data augmentation include rotation with an allowance of 20 degrees, width and height shift by 10% of the original image, shear within the ranges of -0.1 to 0.1 radians (roughly 5.7 degrees), zoom between the ranges of 0.8 and 1.2 times, and horizontal flips. The rationale for choosing a small degree of rotation, width and height shifts, shear, and zoom, is that we don't expect raw X-ray images in real life to present with too much variation in these aspects, and so we do not want the model to learn noise if we introduce augmentations that are too severe. The rationale for choosing only horizontal flips, is we do not expect chest X-rays to be submitted in vertical flip. All training sets, regardless of which technique to handle data imbalance is used, undergoes data augmentation.

The code for performing data augmentation is as shown below.

```
train_datagen = ImageDataGenerator(
    rescale=1./255
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
val_datagen=ImageDataGenerator(rescale=1./255)
test_datagen=ImageDataGenerator(rescale=1./255)
```

Note. The code above includes data normalization for train, test, and validation sets.

4.4.4 TECHNIQUES TO HANDLE DATA IMBALANCE

Resampling methods and class weighting are two techniques to handle imbalanced data, discussed in Chapter 3. Oversampling, undersampling, and class weighting will be used to handle data imbalance in this experiment. The techniques will be used independently of each other in order to evaluate the results returned and compare them to each other.

4.4.4.1 OVERSAMPLING

Figure 7

Class distribution after oversampling of minority class (normal)

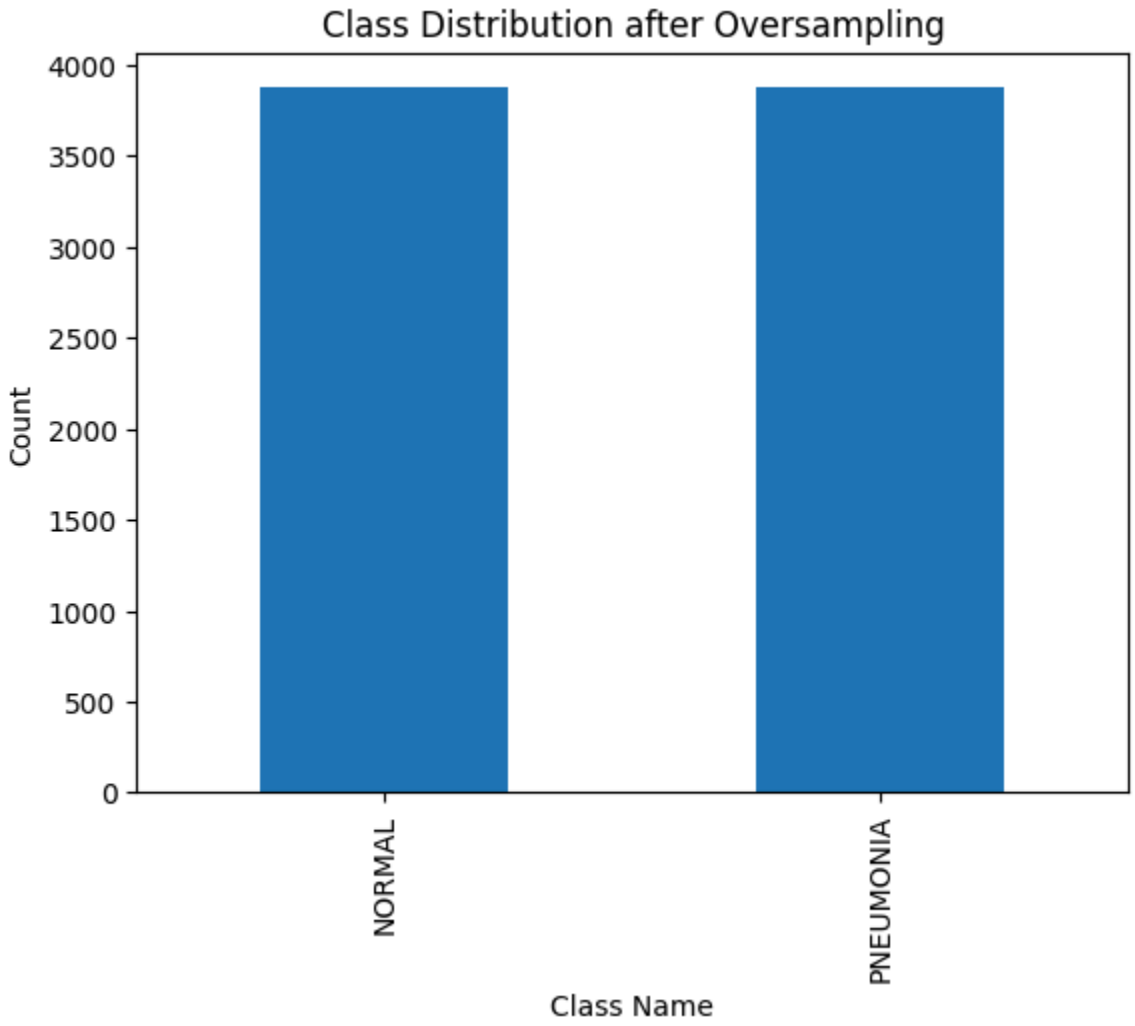


Figure 7 shows the class distribution after oversampling the minority class ('Normal'). The original distribution of 1341 images belonging to the 'Normal' class and 3875 images belonging to the 'Pneumonia' class is altered, so that both 'Normal' and 'Pneumonia' classes have 3875 images respectively. This is done by randomly copying images from the 'Normal' class and duplicating them until they reach the number of the 'Pneumonia' class.

4.4.4.2 UNDERSAMPLING

Figure 8

Class distribution after undersampling majority class ('Pneumonia')

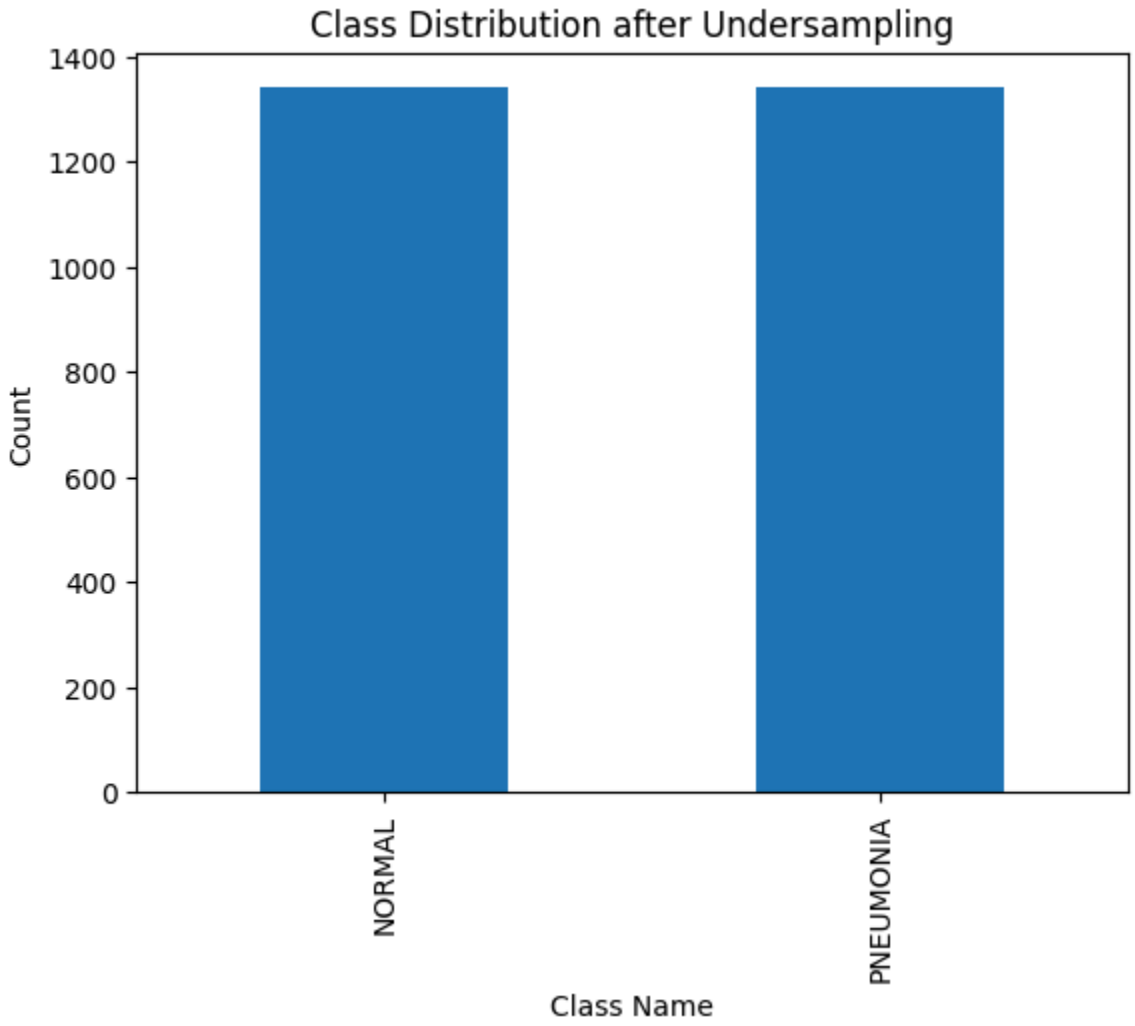


Figure 8 shows the class distribution after undersampling the majority class ('Pneumonia'). The original distribution of 1341 images belonging to the 'Normal' class and 3875 images belonging to the 'Pneumonia' class is altered, so that both 'Normal' and 'Pneumonia' classes have 1341 images respectively. This is done by randomly removing images from the 'Pneumonia' class until it reaches the number of images in the 'Normal' class.

4.4.4.3 CLASS WEIGHTING

Class weighting is another technique to handle imbalance datasets, by assigning higher penalties to misclassified minority classes, forcing the model to pay more attention to underrepresented categories. 'Balanced' is a class weight option that requires no tuning, and is suitable for moderately imbalanced classes. In the case of this project, the ratio of positive to negative classes is 3:1, which is not severe by any means.

The code to create a class weight dictionary is as shown below.

```
classes = np.unique(train_df['class_name'])
class_weights = compute_class_weight(class_weight='balanced',
classes=classes, y=train_df['class_name'])
class_weights_dict = dict(zip(np.unique(train_generator.classes),
class_weights))
```

Note. The dictionary is passed as a parameter during `model.fit` in order to train the model with the class weights generated.

4.4.5 MODEL DEVELOPMENT

The model developed in this study is a convolutional neural network designed for image classification. CNNs are selected due to their effectiveness in learning spatial features from image data. The proposed CNN architecture consists of 3 conv2d layers that are followed by a max_pooling2d layer and ReLU activation. A fully connected layer is used to map extracted features to the final classification output using softmax function. Categorical cross-entropy is employed as the loss function, since the model outputs class probabilities for two categories using a softmax activation. The model is optimized using the Adam optimizer due to its adaptive learning rate and its ability to converge fast. The model is trained using the training dataset, while validation is done at every epoch using the validation dataset to monitor convergence and prevent overfitting. Dropout layers are incorporated to reduce overfitting by randomly deactivating neurons during training. Early stopping with a patience of 3 is implemented in order to prevent overfit.

This model architecture was chosen over others as it provided similar results at a fraction of the size and speed of other models we tested with. A summary of the model is shown below.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 64)	640
max_pooling2d (MaxPooling2D)	(None, 63, 63, 64)	0
conv2d_1 (Conv2D)	(None, 61, 61, 128)	73,856
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 128)	0
conv2d_2 (Conv2D)	(None, 28, 28, 256)	295,168
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 256)	0
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 512)	25,690,624
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1,026

Total params: 26,061,314 (99.42 MB)

Trainable params: 26,061,314 (99.42 MB)

Non-trainable params: 0 (0.00 B)

4.4.6 MODEL TRAINING

The model developed in 4.4.5 is run using model.fit() with parameters such as the training set (train_generator), the number of epochs, which, in our case, we have chosen 20, the validation set, callbacks which are executed at every epoch, and class weights which are specified when we want to train the model on class weights. The code snippet below shows how we train the model and its parameters. When testing resampling methods, class_weight parameter is omitted from model.fit(). The code for callbacks is included in Appendix A.

```
history = model.fit(
    train_generator,
    epochs=20,
    validation_data=val_generator,
    callbacks=callbacks,
    class_weight=class_weights_dict # omitted when checking data
augmentation performance)
```

4.4.7 TESTING AND VALIDATION

This section describes the evaluation process used to assess the performance of the pneumonia detection model.

4.4.7.1 VALIDATION PROCEDURE

At each epoch, validation was carried out using the validation dataset in order to monitor the model's performance during training, and to prevent overfitting. The training and validation loss are plotted to visualize the model's performance.

4.4.7.2. TESTING PROCEDURE

After the model was trained, testing and validation were carried out using the unseen test dataset to measure its generalization capability. Several evaluation metrics were used to analyze model performance, including accuracy, precision, recall, F1-score, and confusion matrix. Threshold tuning was also applied to improve diagnostic reliability.

4.4.7.3 EVALUATION METRICS

The model's performance was evaluated using several metrics commonly employed in medical image classification, which are accuracy, precision, recall, F1-score and confusion matrix.

4.5 CHAPTER SUMMARY

This chapter has explained in detail how the proposed pneumonia detection system was designed and implemented. It started by introducing the overall experimental setup and the proposed solution, which combines data preprocessing, class imbalance handling techniques and a convolutional neural network for classification.

CHAPTER 5

RESULTS, ANALYSIS AND DISCUSSION

5.1 INTRODUCTION

This chapter presents the experimental results obtained from the proposed pneumonia classification model. The models were trained and evaluated using the dataset 'Chest X-Ray Images (Pneumonia)' from Kaggle Paul Mooney, 2025). The evaluation metrics used include accuracy, precision , recall, F1-score and confusion matrix. In addition, this chapter includes a detailed analysis of the results, a comparison of different techniques to handle data imbalance, and a discussion on model performance, limitations, and potential improvements.

5.2 RESEARCH RESULTS AND ANALYSIS

In this section, the results are presented, analyzed and interpreted including the overall model performance, confusion matrix analysis and a comparative evaluation between the baseline model and models trained using different methods to handle data imbalance. The four model specifications were trained and evaluated under varying conditions, and overall, the models performed comparably. A summary of the training times and performance metrics for each model is presented in Table 4.

5.2.1 OVERALL MODEL PERFORMANCE

The oversampled model achieved the highest overall performance, attaining an F1-score of 93.76%. This indicates that augmenting the minority class allowed the model to learn discriminative patterns more effectively and generalize better to unseen data. The baseline model with augmentation on both classes produced a very high recall of 99.49% but a lower precision of 75.63%, suggesting that while it successfully detected most pneumonia cases, it also generated many false positives.

The undersampled model significantly reduced training time due to fewer images, but this came at the cost of lower recall (79.74%), as several pneumonia cases were removed during undersampling. The combination of augmentation with class weighting offered a more balanced trade-off between precision (93.31%) and recall (85.90%), though the training time was longer than the other models. These results emphasize the impact of different class imbalance handling techniques on both model performance and computational efficiency.

Table 4

Model performance and training time with different specifications

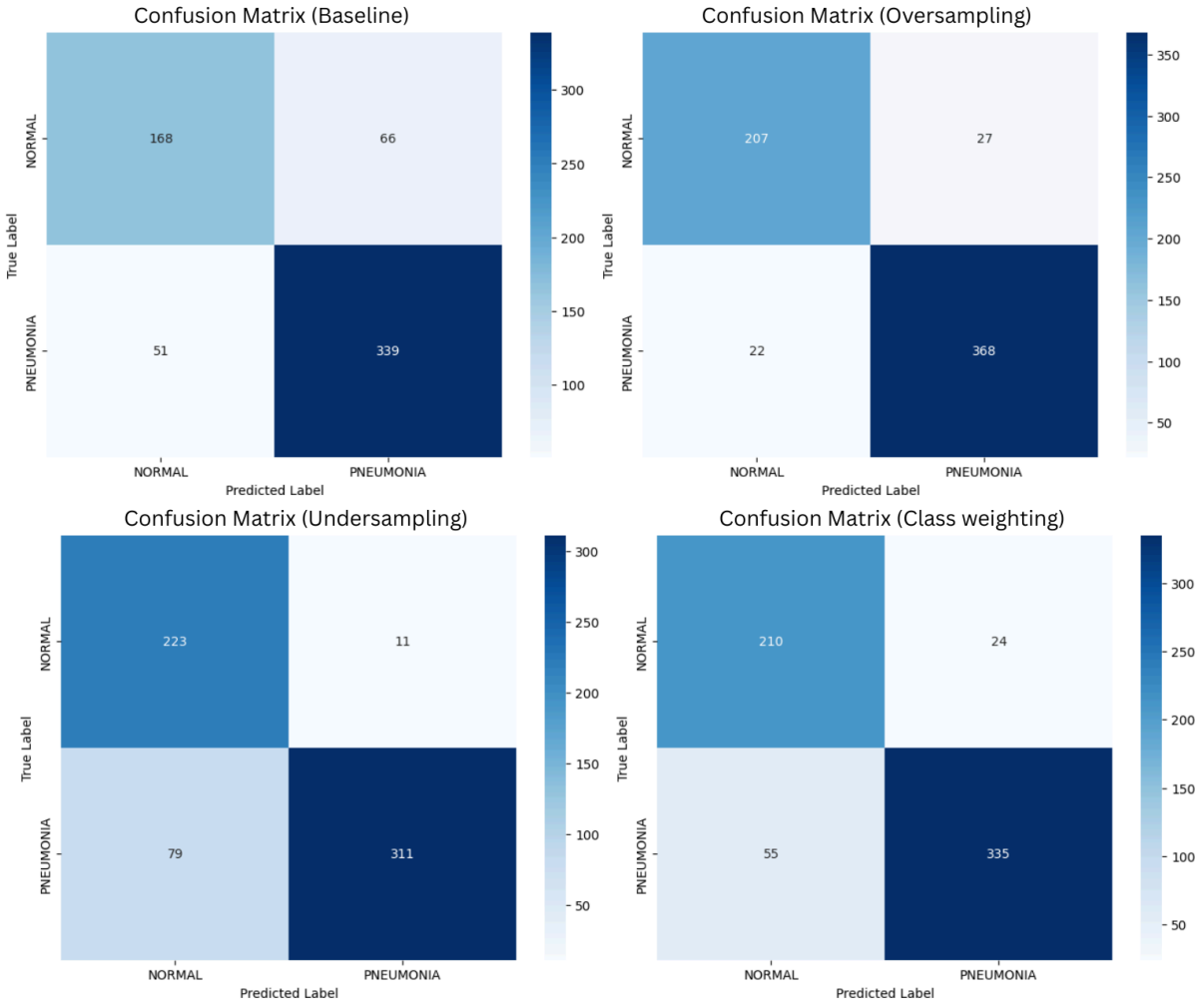
	Model specification	Model Training Time (seconds)	Metrics
1	Baseline model, trained with train dataset with augmentation on both classes.	2082.08	Accuracy: 79.65% Precision: 75.63% Recall: 99.49% F1: 85.94%
2	Baseline model, trained with train dataset with oversampling by augmentation on minority class.	3211.46	Accuracy: 92.15% Precision: 93.16% Recall: 94.36% F1-score: 93.76%
3	Baseline model, trained with train dataset with undersampling majority class.	605.03	Accuracy: 85.58% Precision: 96.58% Recall: 79.74% F1-score: 87.36%
4	Baseline model, trained with train dataset with augmentation on both classes and balanced class weighting.	3688.64	Accuracy: 87.34% Precision: 93.31% Recall: 85.90% F1-score: 89.45%

5.2.2 CONFUSION MATRIX ANALYSIS

Confusion matrices were generated for each model to visualize classification performance. The oversampled model produced the highest number of true positives while maintaining a low number of false negatives, indicating reliable detection of pneumonia cases. The baseline model with general augmentation showed a high number of false positives, consistent with its lower precision. The undersampled model, while producing fewer false positives, misclassified a larger proportion of pneumonia cases as normal, reducing recall. Combining augmentation and class weighting improved the balance of missclassifications across classes.

Figure 9

Confusion matrix for all four models



Note. The confusion matrices presented, from left to right, top to bottom, are: baseline model, model trained on oversampled data, model trained on undersampled data, and model trained with class weights. The confusion matrix follows sci-kit learn's convention where TP, TN, FP, FN are mirrored diagonally from standard convention.

From the confusion matrices, it can be seen that the oversampled model achieved the most balanced classifications, with relatively few false positives and false negatives. The baseline model with the augmentation demonstrated a tendency to over-predict pneumonia, leading to a large number of false positives. Undersampling reduced false positives but missed several pneumonia cases, as shown by the higher number of false negatives.

5.2.3 COMPARATIVE ANALYSIS

Figure 10 shows the overall performance of each model across the four key metrics: accuracy, precision, recall, and F1-score. Model training time is also plotted alongside these four metrics, as percentages, where the baseline model scores 100%, and models which take longer to train score more than the baseline, and vice versa.

The accuracy achieved by the different models evaluated in this study is shown in Figure 10. The oversampled (92.15%) and class weighted (87.34%) models achieved higher accuracy compared to the baseline (79.65%) and undersampled (85.58%) models. The baseline model recorded the lowest accuracy among the evaluated approaches. The undersampled model demonstrated a slight reduction in accuracy, likely due to the loss of training samples during resampling. The improvement in accuracy observed in the oversampled model may be attributed to improved class balance during training. The class-weighted approach achieves comparable accuracy while avoiding explicit data duplication.

Although accuracy provides an overview of classification performance, it does not fully reflect the model's ability to correctly identify pneumonia cases, particularly under class imbalance conditions. Therefore, additional metrics such as precision, recall, and F1-score are considered to provide a more comprehensive evaluation.

Figure 10 also presents the precision comparison across the evaluated models. Precision tells us how often a model correctly predicts pneumonia. The baseline model exhibits the lowest precision (75.63%), indicating a higher proportion of false positives. The undersampled model achieves the highest precision score, although this comes at the expense of recall. The class weighted (93.31%) and oversampled model (93.16%) perform relatively well. The oversampled model achieves improved precision, suggesting more reliable positive predictions. Precision is an important metric to reduce unnecessary follow-up examinations caused by false alarms. Higher precision can mean reduced workload at later stages of clinical assessments, allowing the hospital to focus on cases that are most likely to have pneumonia.

As shown in Figure 10, the baseline model performed the best at recall (99.49%) overall. This indicates that the baseline model is able to correctly detect many actual pneumonia cases. However, the high recall comes at the cost of the low precision by the baseline model. The reason recall might be high for the baseline model might be because the baseline model has learned the distribution of the original data, which has more pneumonia images, and learnt to predict pneumonia more often.

The oversampled (94.36%) and weighted (85.90%) class achieved higher recall compared to the undersampled class (79.74%), which performed the worst in recall across models. The undersampled model demonstrates reduced recall, indicating a higher number of missed

pneumonia cases. This behavior is expected, as undersampling reduces the representation of pneumonia samples during training. In medical diagnosis tasks, recall is often seen as the more important metric between recall and precision, as a higher recall can minimize the risk of undetected disease cases.

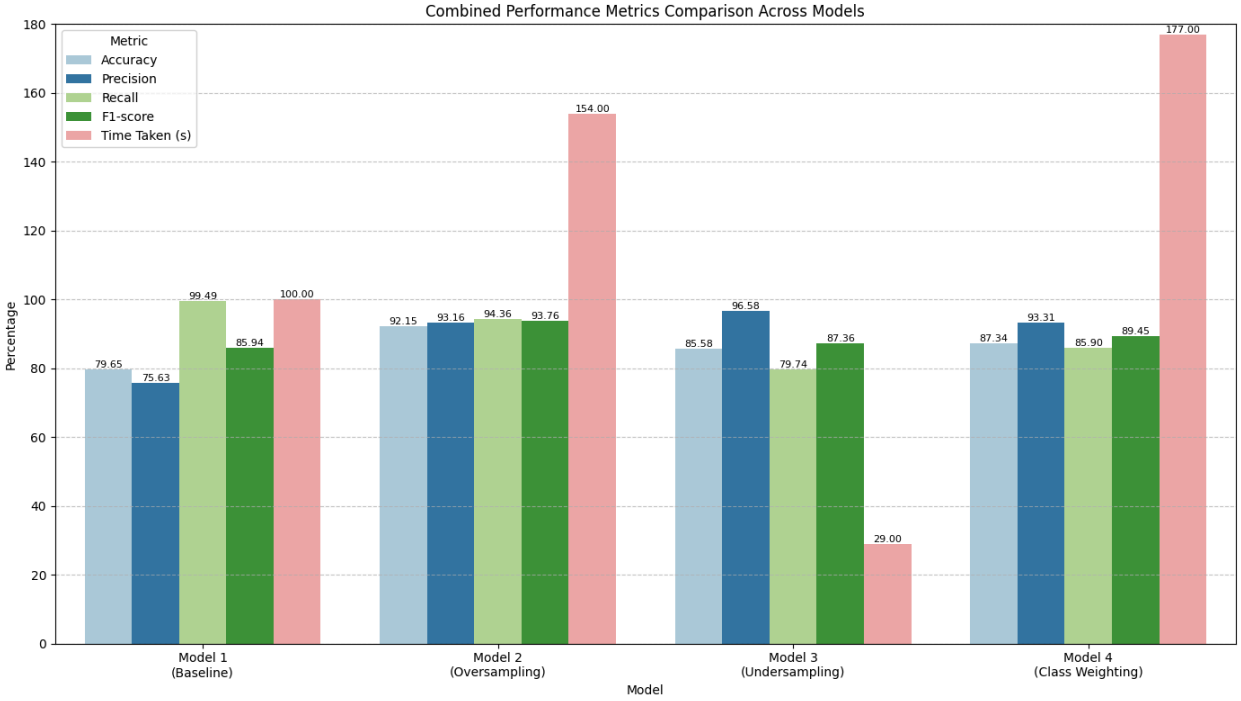
Figure 10 also illustrates the F1-score achieved by each model. The model trained with oversampling data achieves the highest F1-score (93.76%), indicating a more balanced trade-off between precision and recall. The baseline model records the lowest recall (85.94%) due to a low precision score, as seen in Figure 10. The class-weighted and undersampled approach score slightly higher than the baseline model (87.36% and 89.45% respectively), and while the class weighted model performed average in precision and recall, the model trained on undersampling data performed the best in precision, but the worst in recall.

Finally, Figure 10 compares the execution time required by each model configuration. Execution time tells us how computationally expensive the model is. Models incorporating oversampling and augmentation exhibit longer execution times due to increased training data volume. The baseline model demonstrates the shortest execution time but at the expense of reduced classification performance. The class weighted model is the most computationally expensive. Theoretically, the oversampled model should take the most time to run, as it has to process a much larger training set than the other models. However, the model trained on oversampling data converged faster than the class weighted model, and generalized on the data faster than training on oversampled data did. Execution time is an important consideration for real-world deployment, particularly in resource-constrained clinical environments.

The model trained on oversampling data emerges as the most effective strategy when considering a balance of all performance metrics for pneumonia detection. It provides a good balance between identifying actual pneumonia cases (high recall) and minimizing false alarms (high precision), making it suitable as a decision support system. The baseline model performs the worst overall, highlighting the importance of addressing class imbalance when training models.

Figure 10

Summary of metrics across four models



5.2.4 MISCLASSIFICATION ANALYSIS

For the baseline model, false positive cases (‘Normal’ predicted as ‘Pneumonia’) frequently exhibit subtle opacity, which may be caused by poor X-ray quality or higher lung density, which may resemble early-stage pneumonia to the model (Figure 11). Their pixel intensity histograms often show a shift toward higher intensity ranges, which is typically associated with the ‘Pneumonia’ class during EDA. False negative cases (‘Pneumonia’ predicted as ‘Normal’) tend to present with less severe or diffuse opacities that are visually difficult to distinguish from normal lung tissue (Figure 12). Correspondingly, their histograms closely resemble those of normal X-rays during EDA, indicating that some classes may present with pixel intensity distributions that look like other classes, which challenge the model’s classification.

The oversampled model demonstrates a reduction in false negatives, indicating improved sensitivity to pneumonia cases (Figure 14). Remaining false negatives often correspond to extremely subtle pneumonia presentations with pixel intensity distributions similar to normal images. However, the increased sensitivity introduced by oversampling leads to some ambiguous normal cases being misclassified as pneumonia (Figure 13). The histograms of these false positives show only slight or marginal shifts toward higher intensity values, suggesting an expanded decision boundary for the pneumonia class.

In contrast, the undersampled model produces fewer false positives, reflecting a more conservative classification behaviour (Figure 15). The remaining false positives often correspond to genuinely ambiguous normal images whose pixel intensity distributions closely resemble pneumonia cases. However, this conservativeness results in a higher number of false negatives, covering a broader range of pneumonia severities (Figure 16). The histograms of these false negatives frequently resemble those of normal images. The reduced training data diversity limits the model’s ability to capture the full variability of pneumonia-related features.

The class-weighted model shows a low number of false positives, suggesting that class weighting effectively reduces incorrect pneumonia predictions on normal images (Figure 17). However, a notable number of false negatives remain, resulting in moderate recall and indicating that some pneumonia cases are still misclassified as normal. Visual inspection and pixel intensity histograms of these false negative samples suggest that they often correspond to subtle or atypical pneumonia presentations with distributions similar to ‘Normal’ class (Figure 18).

Across all models, misclassifications commonly occur when visual characteristics and pixel intensity distributions overlap between normal and pneumonia classes. While oversampling improves sensitivity and undersampling reduces false alarms, all approaches appear to rely heavily on global pixel intensity patterns rather than fine-grained, localized pathological features. These observations highlight the need for a more balanced approach that enhances feature robustness while maintaining sensitivity.

Figure 11

Baseline model: misclassified false positive images along with pixel intensity histograms

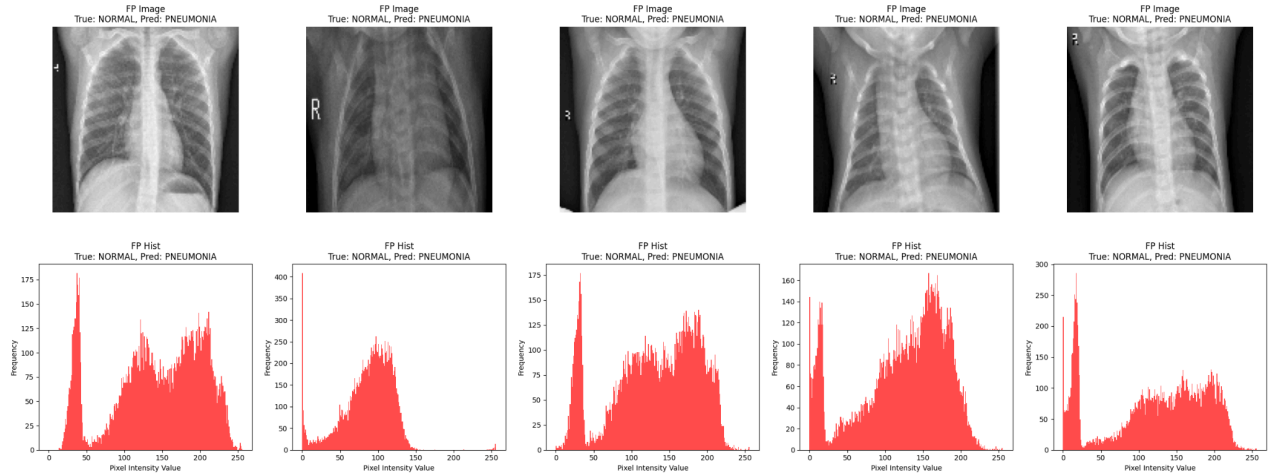


Figure 12

Baseline model: misclassified false negative images along with pixel intensity histograms

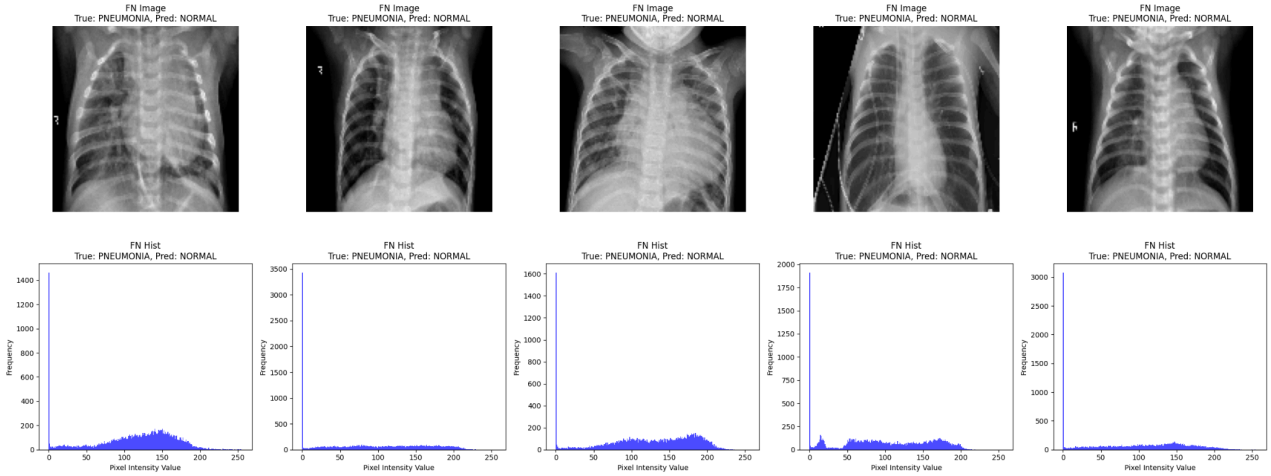


Figure 13

Oversampling: misclassified false positive images along with pixel intensity histograms

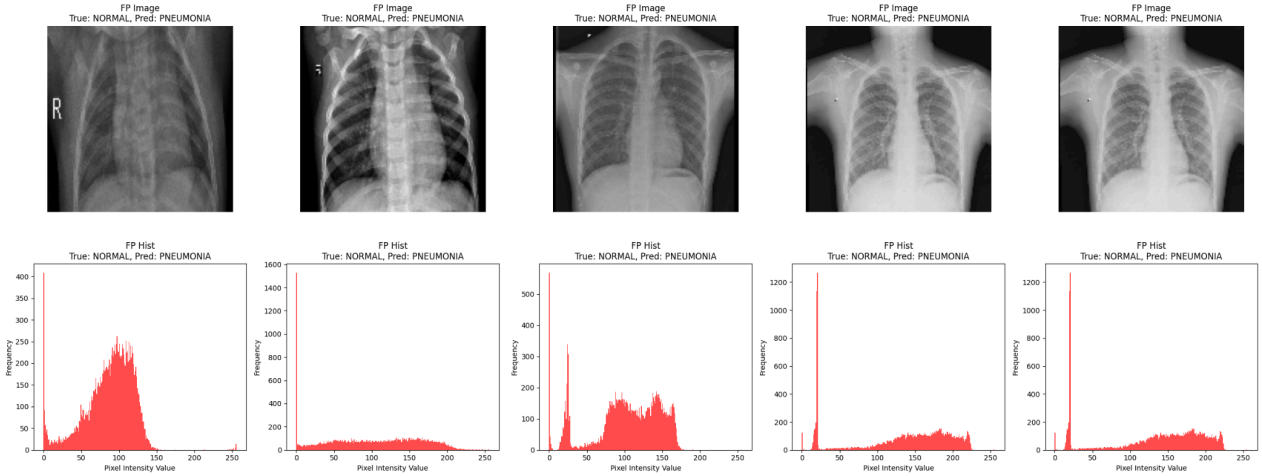


Figure 14

Oversampling: misclassified false negative images along with pixel intensity histograms

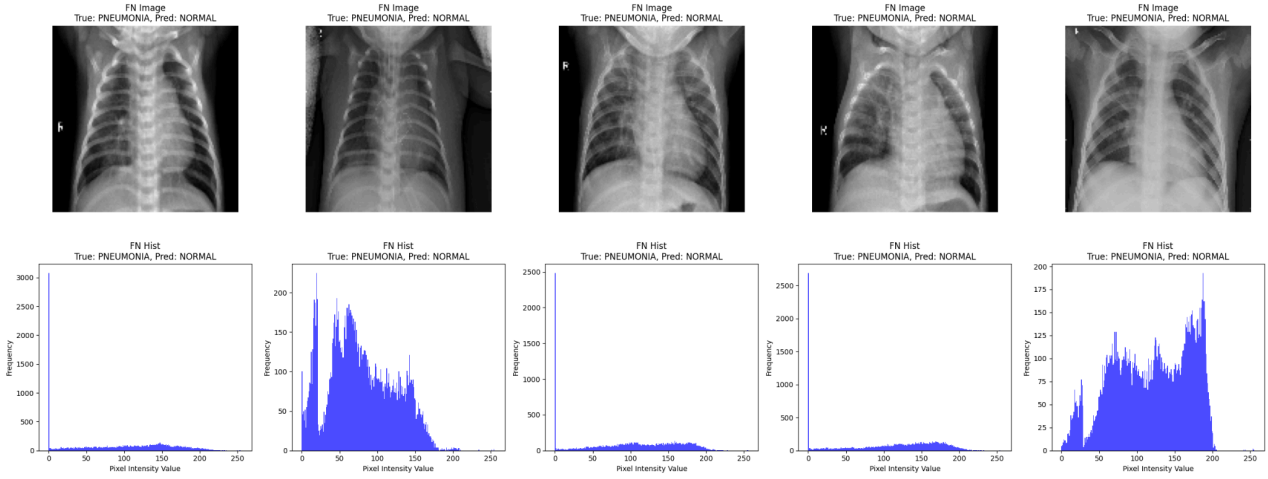


Figure 15

Undersampling: misclassified false positive images along with pixel intensity histograms

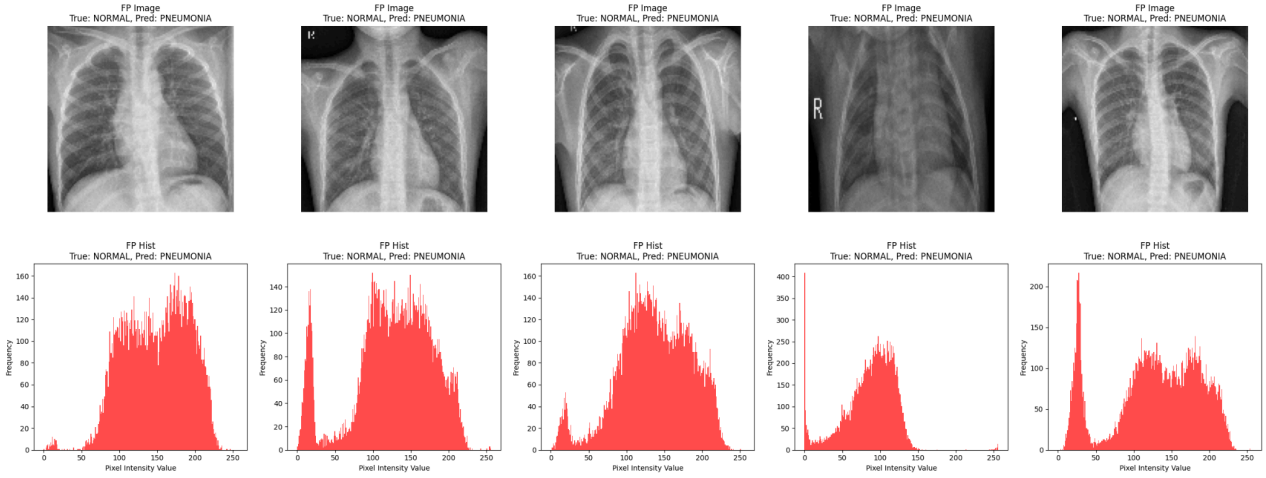


Figure 16

Undersampling: misclassified false negative images along with pixel intensity histograms

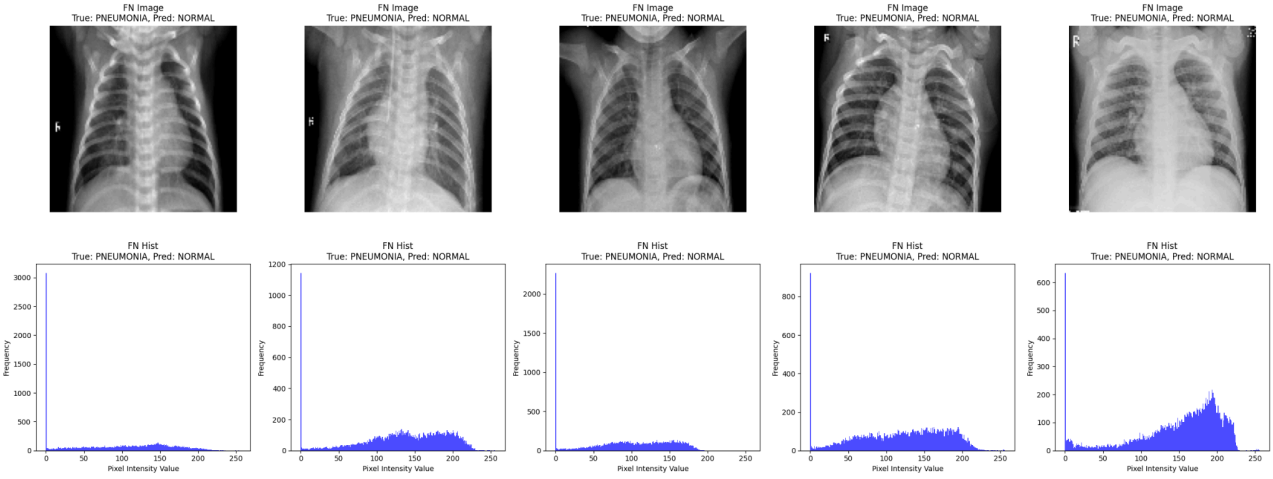


Figure 17

Class weighting: misclassified false positive images along with pixel intensity histograms

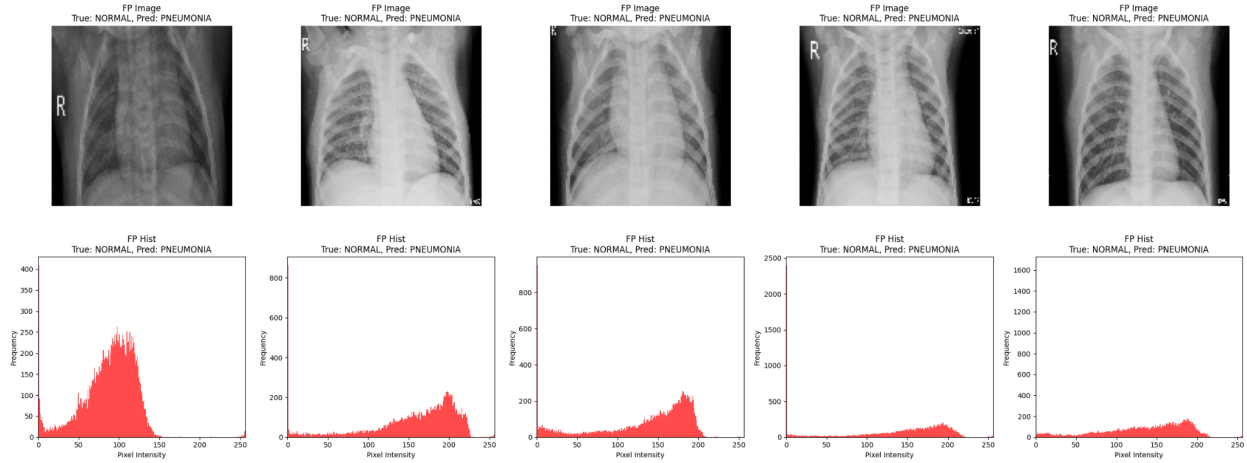
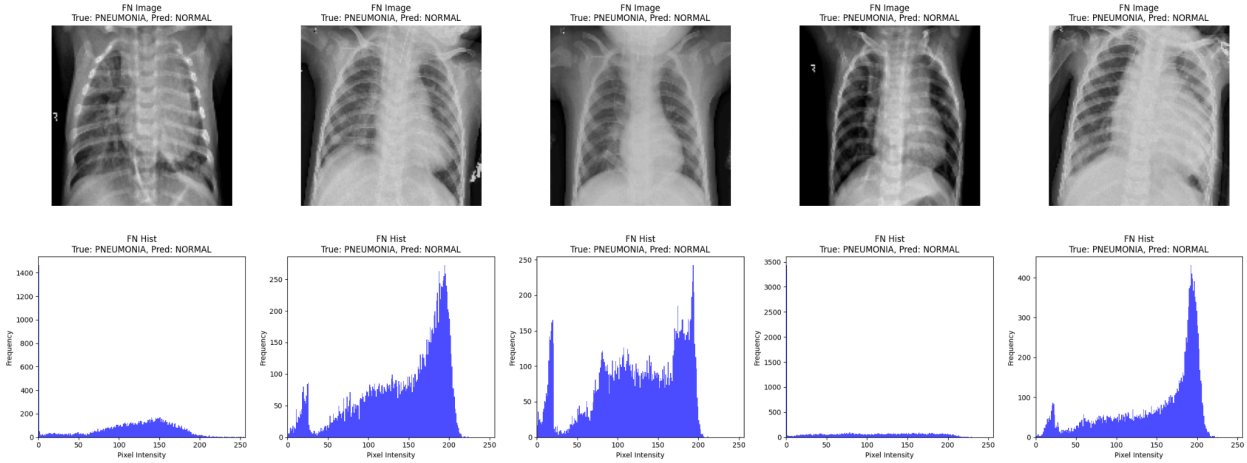


Figure 18

Class weighting: misclassified false negative images along with pixel intensity histograms



5.3 DISCUSSION

The results indicate that approaches addressing class imbalance generally improve pneumonia detection performance. Among the methods of tackling class imbalance, those that try to improve class imbalance by increasing the weights or samples of the minority class ended up performing better than those that try to improve class imbalance by reducing the samples of the majority class. This is likely due to the increased representation of minority-class samples during training, or the increased loss value when minority classes are given more weight, which allows the model to learn more discriminative features for pneumonia cases. As a result, the model exhibits fewer missed detections, which is critical in medical screening tasks.

Models with higher recall tend to exhibit a slight reduction in precision, suggesting an increased sensitivity to pneumonia cases at the cost of additional false positives. Thus, there exists a precision-recall tradeoff, where when a model tries to catch more positive cases (increasing recall), it will inevitably bring in more false positives, lowering precision, while being stricter to minimize false positives (increasing precision) causes the model to miss actual positives. Hence, these metrics tend to move together, and the precision or recall of a model can be adjusted by adjusting the threshold value, thus modifying at which probability a model would predict a positive class.

The strong performance of the oversampling approach can be attributed to increased exposure to minority-class patterns during training, enabling the model to learn more discriminative pneumonia-related features. The method of oversampling training data produces a more balanced error distribution, which contributes to improved F1-score performance. The method of undersampling also has the effect of producing a more balanced error distribution, but it significantly reduces execution time by decreasing dataset size and removing majority-class samples, which limits the model's ability to generalize, leading to reduced recall. For class weighting, the model was trained on the original dataset size, which is why it performed better than the model trained on undersampled data. The execution time was longer for class weighting even though dataset size was unchanged, because the loss computation for each batch of data is weighted. The gradients for minority class-samples are amplified when misclassification occurs, leading to the model taking more epochs to converge, and overall training becomes slower.

In the context of pneumonia screening, higher recall is desirable, as failing to identify an infected patient may lead to delayed treatment. In this regard, the baseline model and the oversampled model might perform well in clinical settings, as false negatives are more costly. However, the baseline model might overburden the hospital with false positives because of its low precision. From a deployment perspective, the choice of imbalance handling strategy should consider both available computational resources and the acceptable balance between precision, recall, and efficiency.

The visualizations of misclassification analysis suggest that the model might heavily rely on overall pixel intensity distribution to make its predictions. This approach struggles when there is subtle pathology (leading to false negatives) or when normal variations mimic pathology (leading to false positives). The variability in image quality and presentation across the dataset is likely a significant contributing factor to misclassifications. A ‘Normal’ image from one machine might have a higher overall intensity than a ‘Pneumonia’ image from another, confusing the model. These specific misclassifications suggest that the model’s feature extraction might not be robust enough to capture fine-grained diagnostic details or to be invariant to imaging variations. Improving the model’s ability to localize and identify specific patterns associated with pneumonia, rather than just overall intensity shifts, would be beneficial.

One limitation of this study is that the execution time was evaluated under a single hardware configuration, which may affect generalizability. Another limitation is that oversampling may introduce redundant samples, increasing the risk of overfitting to repeated minority-class patterns. The evaluation relies on a fixed train-test-validate split provided by the authors of the dataset, which may introduce variability in performance estimation. Finally, the dataset originates from a single public source, which may not fully represent variations in imaging equipment or patient demographics.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

This chapter summarizes the key findings of the study, evaluates the achievement of research objectives, and provides suggestions for future improvements. The main focus of this study was to explore different methods to handle class imbalance in pneumonia detection using chest X-ray images, and to evaluate their impact on model performance and computational efficiency. The chapter also highlights the limitations of the current work and potential directions for further research.

6.2 ACHIEVEMENT OF PROJECT OBJECTIVES

The project successfully achieved its objectives by implementing and evaluating a convolutional neural network (CNN) for binary classification of chest X-ray images into Normal and Pneumonia categories. Several techniques for handling class imbalance were explored, including oversampling of the minority class, undersampling of the majority class and combining augmentation with class weighting.

The oversampled model achieved the highest overall performance, attaining an F1-score of 93.76%. This indicates the augmenting the minority class allowed the model to learn discriminative features more effectively and generalize better to unseen data. The baseline model with augmentation on both classes exhibited a very high recall of 99.49% but a lower precision of 75.63%, suggesting that while it successfully detected most pneumonia cases, it also produced a significant number of false positives.

The undersampled model reduced training time due to fewer images but resulted in lower recall (79.74%), indicating that some pneumonia cases were excluded during undersampling. Combining augmentation with class weighting produced a more balanced trade-off between precision (93.31%) and recall (85.90%), although it required longer training time. These findings emphasize the trade-offs between model performance, class imbalance handling techniques, and computational efficiency.

The results align with the objectives set in Chapter 1, confirming that preprocessing, model parameters, and class imbalance handling techniques significantly influence evaluation metrics and computational cost. Additionally, if the study reinforced the importance of recall in medical diagnosis, ensuring that the model is unlikely to miss actual pneumonia cases.

6.3 SUGGESTIONS FOR IMPROVEMENT AND FUTURE WORKS

Although this study achieved encouraging results in detecting pneumonia from chest X-ray images, there are several areas that can be improved and explored further in future research.

One key limitation of this study is the evaluation was all done on a single hardware configuration, which may affect generalizability. The size of the dataset, particularly the validation set, should also be expanded by incorporating additional chest X-ray images from other publicly available sources or real clinical data, which would likely improve the model's ability to generalize and reduce overfitting. It would also represent variations in imaging equipments or patient demographics. The evaluation relies on a fixed train-test-validate split provided by the authors of the dataset, which may introduce variability in performance estimation. Finally, the dataset originates from a single public source, which may not fully represent variations in imaging equipment or patient demographics.

In this project, a custom convolutional neural network was used to balance performance and computational efficiency. Future work could investigate more extensive hyperparameter tuning. Adjusting parameters such as the learning rate, dropout rate, batch size, and augmentation settings may help optimize the balance between precision and recall. Automated tuning methods could be explored to systematically identify optimal configurations.

Finally, this study was conducted using CPU-based training to maintain accessibility and low computational cost. Future research could leverage GPU or cloud-based computing resources to train deeper models, experiment with higher-resolution images and conduct more comprehensive evaluations. This would allow for broader experimentation while potentially improving diagnostic accuracy.

In conclusion, the results of this study demonstrate that effective handling of class imbalance, particularly through oversampling, can produce reliable and computationally efficient models for pneumonia detection. With further enhancements in data size, model architecture, interpretability and computational resources , this work can be extended into a more robust and clinically applicable diagnostic support system.

REFERENCES

American Lung Association. (2025, August 14). *Pneumonia Symptoms and Diagnosis*. https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). *SMOTE: Synthetic Minority Over-sampling Technique*. https://doi.org/10.48550/ARXIV.1106.1813

Chest X-Ray Images (Pneumonia). (n.d.). Retrieved November 25, 2025, from https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

Cleveland Clinic. (2025, November 18). *Pneumonia: Causes, Symptoms, Diagnosis & Treatment*. Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/4471-pneumonia

DEPARTMENT OF STATISTICS MALAYSIA. (2025). *STATISTICS ON CAUSES OF DEATH, MALAYSIA, 2024* (p. 2). MINISTRY OF ECONOMY. https://www.dosm.gov.my/site/downloadrelease?id=statistics-on-causes-of-death-malaysia-2024&lang=English&admin_view=

Elor, Y., & Averbuch-Elor, H. (2022). *To SMOTE, or not to SMOTE?* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2201.08528

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, *321*, 321–331. https://doi.org/10.1016/j.neucom.2018.09.013

Gröger, F., Amruthalingam, L., Lionetti, S., Navarini, A. A., Ille, F., & Pouly, M. (2025). A review and systematic guide to counteracting medical data scarcity for AI applications.

Computer Methods and Programs in Biomedicine Update, *8*, 100220. https://doi.org/10.1016/j.cmpbup.2025.100220

HAMSUDDIN, A. A. (2025, November 2). *AI tech used in govt clinics detects 22 cases in screening pilot*. NST Online. https://www.nst.com.my/news/nation/2025/11/1306983/ai-tech-used-govt-clinics-detects-22-cases-screening-pilot

NHLBI. (2022, March 24). *Pneumonia—Diagnosis | NHLBI, NIH*. https://www.nhlbi.nih.gov/health/pneumonia/diagnosis

Omoniyi, T. M., Abel, B., Omoebamije, O., Onimisi, Z. M., Matos, J. C., Tinoco, J., & Minh, T. Q. (2025). The Effect of Data Augmentation on Performance of Custom and Pre-Trained CNN Models for Crack Detection. *Applied Sciences*, *15*(22), 12321. https://doi.org/10.3390/app152212321

Razali, M. N., Arbaiy, N., Lin, P.-C., & Ismail, S. (2025). Optimizing Multiclass Classification Using Convolutional Neural Networks with Class Weights and Early Stopping for Imbalanced Datasets. *Electronics*, *14*(4), 705. https://doi.org/10.3390/electronics14040705

Shimizu, T., & Tokuda, Y. (2012). Pivot and cluster strategy: A preventive measure against diagnostic errors. *International Journal of General Medicine*, 917. https://doi.org/10.2147/IJGM.S38805

Usman, C., Rehman, S. U., Ali, A., Khan, A. M., Ahmad, B., Usman, C., Rehman, S. U., Ali, A., Khan, A. M., & Ahmad, B. (2025). Pneumonia Disease Detection Using Chest X-Rays and Machine Learning. *Algorithms*, *18*(2). https://doi.org/10.3390/a18020082

Zhang, W., Deng, L., Zhang, L., & Wu, D. (2020). *A Survey on Negative Transfer*. https://doi.org/10.48550/ARXIV.2009.00909

Zhang, Y., Lei, Z., Zhuang, L., & Yu, H. (2021). A CNN Based Method to Solve Class Imbalance Problem in SAR Image Ship Target Recognition. *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 229–233. https://doi.org/10.1109/IAEAC50856.2021.9390936

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., & Summers, R. M. (2021). A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises. *Proceedings of the IEEE*, *109*(5), 820–838. https://doi.org/10.1109/JPROC.2021.3054390

APPENDICES

APPENDIX A

Listing A.1: Imports

```
import tensorflow as tf
import matplotlib.pyplot as plt
import keras
import cv2
from PIL import Image
from keras.models import Sequential
from keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout , Input, MaxPooling2D
from keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
from sklearn.utils.class_weight import compute_class_weight
import os
import kagglehub

base_dataset_path = kagglehub.dataset_download("paultimothymooney/chest-xray-pneumonia")
train_dir = os.path.join(base_dataset_path, 'chest_xray', 'train')
test_dir = os.path.join(base_dataset_path, 'chest_xray', 'test')
val_dir = os.path.join(base_dataset_path, 'chest_xray', 'val')

print(f"train_dir is set to: {train_dir}")
print(f"test_dir is set to: {test_dir}")
print(f"val_dir is set to: {val_dir}")
```

Listing A.2: Image integrity and handling

```
def get_valid_images(folder):

    image_paths = []
    labels = []

    for class_name in os.listdir(folder):
        class_dir = os.path.join(folder, class_name)
        if not os.path.isdir(class_dir):
            continue

        for fname in os.listdir(class_dir):
            if not (fname.lower().endswith(('.jpg', '.jpeg', '.png'))):
                continue

            path = os.path.join(class_dir, fname)
            try:
                Image.open(path).verify()
                image_paths.append(path)
                labels.append(class_name)
            except Exception:
                print(f"Skipping invalid image: {path}")

    df = pd.DataFrame({'filename': image_paths, 'class_name': labels})
    return df

train_df = get_valid_images(train_dir)
val_df   = get_valid_images(val_dir)
test_df  = get_valid_images(test_dir)ist
```

Listing A.3: Model development

```
model = Sequential([
    Input(shape=(128, 128, 1)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2,2)),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D((2,2)),

    Conv2D(256, (3,3), activation='relu'),
    MaxPooling2D((2,2)),

    Flatten(),
    Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```