BIRMINGHAM CITY
University

# Leveraging Convolutional Neural Networks to identify Pneumonia

**Project Proposal**

Lewis Higgins - Student ID 22133848

CMP6228 - Deep Neural Networks

Module Coordinator: Khalid Ismail

Word count excluding figures, tables & references: 1647 / 1650

# Contents

# List of Figures

# List of Tables

# Introduction

Pneumonia remains a critical global health issue especially following the COVID-19 pandemic, particularly for more vulnerable populations like children and the elderly. As the leading infectious cause of death in children under five, claiming hundreds of thousands of young lives annually, and a significant risk factor for respiratory complications in later life, rapid and accurate diagnosis is essential to mitigate both immediate and lifelong consequences. This project aims to address these challenges by leveraging modern advancements in deep learning to develop a system for pneumonia detection using chest X-ray images.

To do so, a Convolutional Neural Network (CNN) architecture enhanced by transfer learning for image recognition is proposed. By leveraging deep learning approaches, the proposed model is planned to be an accurate tool for early pneumonia detection, ultimately reducing preventable deaths and long-term health burdens worldwide.

# Motivation and objectives

## 1.1 Subject area

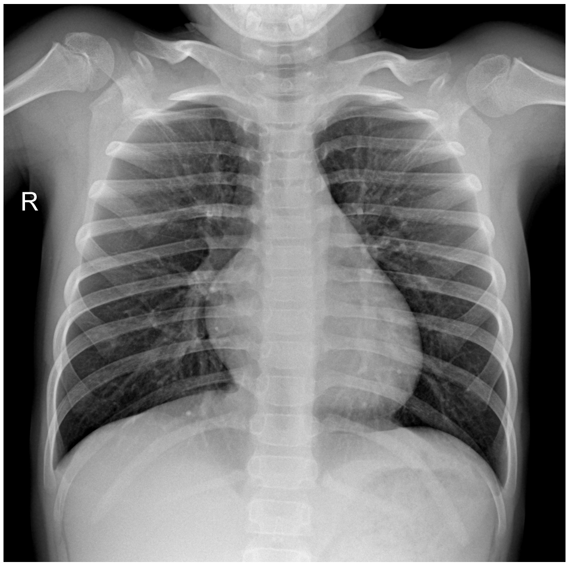
Pneumonia is a lower respiratory tract infection (LRTI) commonly caused by viruses or bacteria wherein the alveoli of the lungs become clogged with pus and fluid, which can be life-threatening in people of any age, but especially in children and the elderly (NHS, 2017). The World Health Organisation (WHO) state that pneumonia is the single largest infectious cause of death in children, killing 808,000 children under the age of 5 in 2017 (WHO, 2025). Furthermore, even if pneumonia is survived during the initial infection, Allinson et al. (2023) studied that those who contract the condition as a child are 93% more likely to die from respiratory diseases later in life than those who did not.

It is therefore imperative that recent technological advancements are leveraged for the quick diagnosis of the infection to allow swift treatment to avoid life-threatening consequences, both in the short-term and long-term.

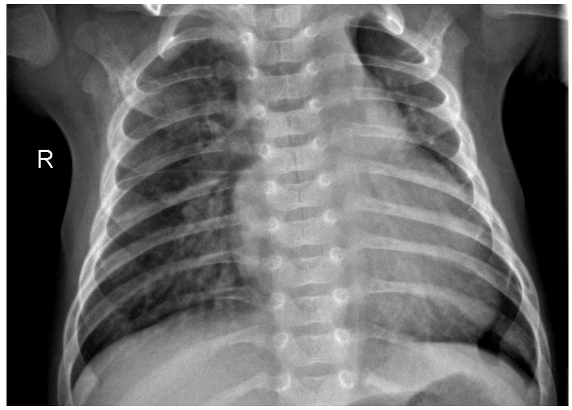
## 1.2 Dataset choice

The chosen dataset is sourced from the Mendeley data repository[1] (Mendeley, 2025), uploaded and created by Kermany et al. (2018, p.1127) in their research of the applications of neural networks for medical diagnoses. The dataset contains 5,856 images of chest X-ray scans of children taken from the Guangzhou Women and Children's Medical Center in China, and is 1.18GB in size. Kermany et al. (2018, p.e1) state that the data was heavily screened and scrutinized for confirmation before labelling, which was done by removing low quality or unreadable images, and then consulting two expert physicians before labelling the images. The physicians' labelled set was then presented to a third expert, who confirmed their validity. Because of this, there is absolute certainty in the authenticity and validity of the data itself as well as its ground truth labels. There are only two classes of images: those with pneumonia and those without, as depicted by Figures 1.1a and 1.1b.

[1] A slightly older version of the dataset is available on Kaggle (Mooney, 2018), though the Mendeley version is the latest. Despite there being no noticeable difference between the two, the Mendeley version was used for authenticity as it was from the original author.




(a) No pneumonia.




(b) Pneumonia.

Figure 1.1: Sample images from the dataset.

### 1.2.1 Potential issues

The dataset has already been pre-emptively split into training and testing sets, which saves having to perform a manual split. However, the data itself is not immediately usable and will require further preprocessing before being used to train a model, namely due to the following issues:

| Issue | Explanation |
|---|---|
| Images are of different resolutions. | The sample images vary widely in resolution, with some being substantially higher than others. The neural network's input layer will be fixed in size, meaning all input data must be the same size or the network will be unable to process it. This can be addressed programmatically to automatically resize all images to a given resolution. |
| Class imbalance | The training set contains 1,349 samples of patients without pneumonia, but 3,883 samples of patients with pneumonia. This can lead to the model favouring those with pneumonia rather than the underrepresented class of those without. This can be addressed using the techniques discussed in Section 2.4. |

Table 1.1: The issues with the dataset before any preprocessing.

## 1.3 Data science problem

This dataset poses a clear data science problem pertaining to the binary classification of these images which will be addressed through the development of a convolutional neural network image classification model leveraging supervised learning. The ground truth is present within the dataset through its file structure[2], shown below:

```
.
`-- chest_xray/
    |-- test/
    |   |-- NORMAL/
    |   |   |-- NORMAL-4512-0001.jpeg
    |   |   |-- NORMAL-11419-0001.jpeg
    |   |   `-- ...234 more images
    |   `-- PNEUMONIA/
    |       |-- BACTERIA-40699-0001.jpeg
    |       |-- VIRUS-4190128-0001
    |       `-- ...388 more images
    `-- train/
        |-- NORMAL/
        |   |-- NORMAL-28501-0001.jpeg
        |   |-- NORMAL-32326-0001.jpeg
        |   `-- ...1347 more images
        `-- PNEUMONIA/
            |-- BACTERIA-7422-0001.jpeg
            |-- VIRUS-12220-0001.jpeg
            `-- ...3881 more images
```

Files of the appropriate class are stored in the relevant subfolder[3]. When this dataset is loaded, it will be possible to assign the relevant label to each image based on its subfolder of origin. Note that while there are separate files for bacterial and viral pneumonia, an analysis of other works using this dataset, seen in Chapter 2, indicated that this will not be an issue.

[2] Because Kermany et al. (2018)'s work was not only on pneumonia, the Mendeley ZIP file contains two separate datasets. This proposal is only for the "chest_xray" dataset.

[3] Shown file names are given as examples, and are not in the direct order they appear in the folders.

# Related work

## 2.1 Introduction

This section of the report aims to demonstrate the main concepts of related techniques that have been previously used to solve the problem through a thorough review of surrounding literature.

## 2.2 Traditional machine learning methods

Traditional machine learning methods such as Random Forests, K-Nearest Neighbours and Support Vector Machines have previously been leveraged for pneumonia classification as seen in the works of Ortiz-Toro et al. (2022). They state that leveraging these traditional approaches alongside the creation of handcrafted textural features 'offers good performance with very low computational complexity', demonstrated in their results wherein they attained an F1-Score of 93%.

However, leveraging traditional methods on detailed image datasets such as Kermany et al. (2018)'s relies upon manual feature engineering of textural features, which can unintentionally introduce bias. Dataset bias can significantly harm the generalisation capabilities of produced models (Selvaraju et al., 2017). This is because it is possible that the new data given to a deployed model would differ from the preset features used on the original dataset, which a biased model would be unable to accurately predict, unlike deep learning methods which can automatically interpret features without manual intervention.

Furthermore, some researchers such as Wang et al. (2021) have leveraged both traditional and deep learning techniques on other unrelated datasets, wherein they discovered that the Support Vector Machine they used had an accuracy 10% lower than a convolutional neural network on a large dataset, though actually had an accuracy 3% higher on a smaller dataset. They also note that deep learning methods have considerably longer training times than traditional counterparts.

SVMs are the most commonly seen model in traditional machine learning works using this dataset. SVMs aim to find the optimal hyperplane which separates one class from another. They are likely the most common due to their high performance and evaluation metrics even when many dimensions are introduced, such as in image data.

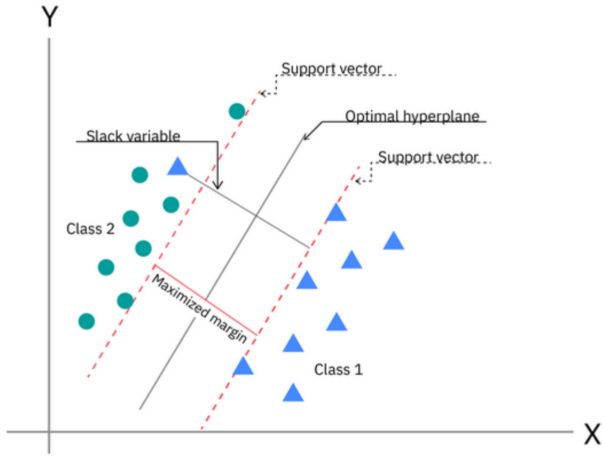



Figure 2.1: A cursory overview of an SVM's functionality (IBM, 2023).

## 2.3 Deep Neural Networks

Deep neural networks are the most heavily used methods for classification tasks with this dataset, which can be seen on both the dataset's Kaggle page (Mooney, 2018), and through a thorough literature search.

There are many neural network types, though the most suitable and most commonly used neural network architecture across related literature[1] for image classifications such as the one proposed here is a Convolutional Neural Network (CNN). CNNs are heavily used for image classification tasks across many fields, including for identifying pneumonia. A more detailed description of the functionality of a CNN can be found in Section 4.1, as this section instead details the accomplishments of other works with Kermany et al. (2018)'s dataset.

El Asnaoui (2021)'s work is one of the most informative[2], and tests three different pretrained CNNs and combinations of them merged into different ensemble models. They tested implementations of pretrained InceptionResNet_V2, MobileNet_V2, and ResNet50 models, achieving F1-scores of 93.52, 91.62 and 93.47 respectively. They also produced an ensemble model containing all three, which achieved their highest F1 score of 94.84. Their work clearly depicts the effectiveness of CNNs in classifying pneumonia.

Analysing the works of others published on the dataset's Kaggle page also follows this trend - CNNs are the most optimal choice for this particular classification task, with many achieving accuracies of at least 92% with minimal overfitting.

[1]Specifically, the works of El Asnaoui (2021), Rajpurkar et al. (2017), Sourab and Kabir (2022), Stephen et al. (2019) and Umar Ibrahim et al. (2022)

[2]It should be noted that they used an altered dataset which they created from merging Kermany et al. (2018)'s data with a COVID-19 dataset.

## 2.4 Data augmentation and overfitting

Many works on this dataset and related ones cite the previously mentioned class imbalance. Mathur (2020) and others solve this using data augmentation to boost the number of samples without pneumonia so that the classes are balanced, thereby reducing bias in the eventual trained model. To do so, they leverage the ImageDataGenerator from Keras to create artificial additional samples based on the original data through 30-degree rotation, 20% zoom, 10% horizontal and/or vertical shifting, and horizontal flipping. These processes are all performed randomly, and not always in the same order. This ensures a wide variety of differing samples for the model to train on, thereby greatly assisting the training process and eventual model performance metrics, as Mathur (2020) found with their model attaining an accuracy of 92.6% and loss of 0.29 on the testing set after 11 epochs of training.

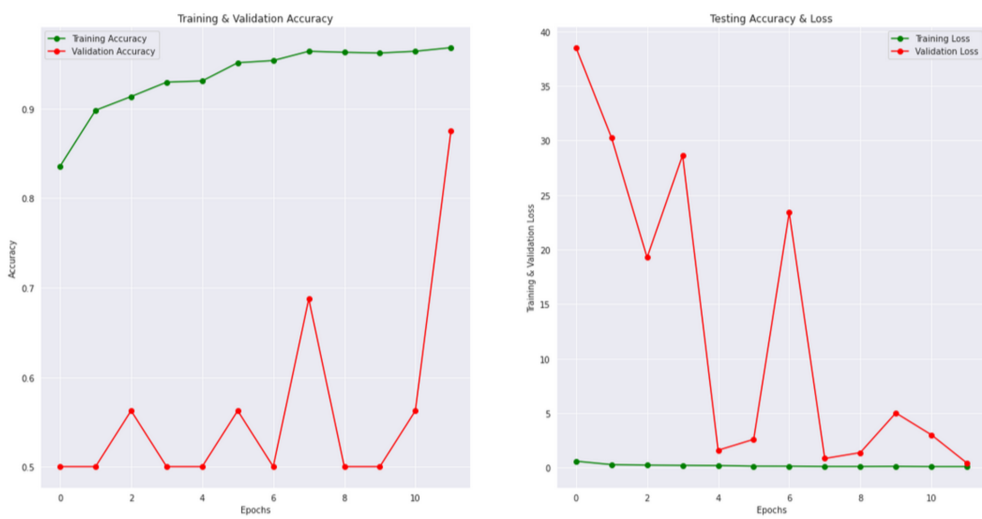



Figure 2.2: Mathur (2020)'s model performance over 11 epochs.

Through the graphs they generated, there were visible substantial improvements on the 11th epoch specifically, wherein validation accuracy and loss massively improved. This was likely found to be the perfect balance before the model started to overfit, which is defined as 'when an algorithm fits too closely or even exactly to its training data' (IBM, 2021b). An overfit model would show patterns of excellent training accuracy and loss, but poor validation accuracy and loss, which could be seen notably on epochs 3 and 6 of Mathur (2020)'s work. If an overfit model was not remedied, it would perform terribly on real-world unseen data, which would go against the objectives of this project.

# Deep learning fundamentals

## 3.1 What are deep neural networks?

Deep neural networks are a significant advancement in computing which build greatly upon traditional machine learning techniques, especially in fields such as computer vision and image recognition. By leveraging newfound processing power typically found in modern graphics cards, **neural networks** formed of layers of neurons which replicate the functionality of the human mind can be created to solve problems at higher degrees of accuracy than ever before. A visualisation of a deep neural network is depicted in Figure 3.1, with key concepts detailed in Table 3.1.

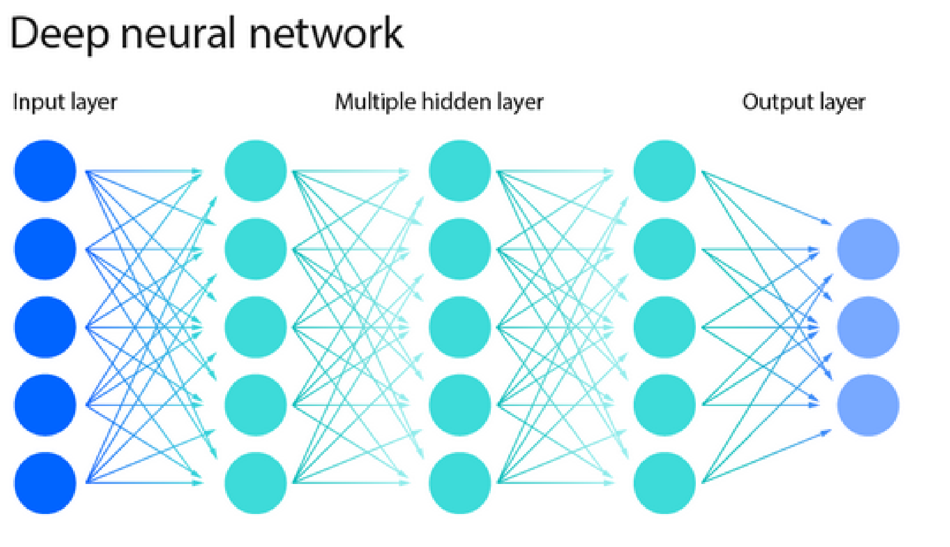



Figure 3.1: A neural network with input, hidden and output layers (IBM, 2021a).

| Concept | Description |
|---|---|
| Neuron | Similar to the human mind, neural networks are formed of synthetic neurons which receive, process and output one data feature. All layers of a neural network are formed of neurons. |
| Input layer | The first layer of any neural network, in which data is first taken into the network. Input layers will be of a size equal to the amount of features; for example, a 32x32 image will need an input layer of 1024 neurons. |

| | |
|---|---|
| Hidden layers | The main processing logic of the neural network is performed in hidden layers. There can theoretically be any amount of hidden layers in a network, though processing power requirements will grow exponentially. Hidden layers are executed sequentially, meaning that the first hidden layer will feed processed data directly into the second, and so on. This means that models with many hidden layers ('deep models') will often yield higher accuracies, especially on their training data as they could begin to overfit. Hidden layers can apply **weights** and **biases** to each neuron, which controls the importance of individual features to the overall model. |
| Output layer | The final layer of any neural network, where a prediction is received. The output layer will contain neurons equal to the number of classes in a classification problem (2 in binary, or $n$ in multi classification where $n$ is the number of classes), or exactly one neuron in a regression problem. In a classification problem, the neuron with the highest output is the predicted class. In a regression problem, the single output neuron directly outputs its prediction. |
| Loss functions | Used to measure how well the model is predicting in relation to the actual target values in supervised learning. The loss function used is dependent on what problem the model is solving; regression tasks often use functions such as Mean Squared Error (MSE), whereas classification tasks will use a cross-entropy function. The specific cross-entropy function once again depends on the problem, with binary classification models often using Binary Cross-Entropy, whereas multi-class models will typically use Categorical Cross-Entropy. The overall objective of a model is to minimise the value returned by these functions (the 'loss value'). |
| Optimisers | Another key element in any neural network is the **optimiser**. There are a wide variety of options, with some of the most common being Stochastic Gradient Descent (SGD), RMSProp, or Adam. The objective of an optimiser is to minimise model loss by dynamically adjusting the weights and learning rates of the network through a process called **backpropagation**, which efficiently computes the gradient of the loss function while accounting for the network's weights. |
| Learning rate | A key parameter of an optimiser is its learning rate, which dictates how fast patterns will be learned in the data for backpropagation. If this is too high, the model may learn patterns which are not truly present, and will see lower accuracy and other metrics as a result. If it is too low, the model may take considerably more epochs to train. Balancing the learning rate is a good way to maximise model performance while minimising training time. |

| | |
|---|---|
| Epochs | The amount of times the model is trained on the data, affecting backpropagation as weights and biases are calculated with each epoch. Too many epochs will eventually lead to overfitting as the model becomes too calibrated to its training data and becomes unable to generalise. |
| Activation functions | Used to make a model less linear, thereby allowing it to learn complex patterns in data, which is especially important when processing image data. The overall purpose of these functions are to dictate whether a neuron should activate (pass data on to the next layer) or not. If the neuron is activated, the output it receives is transformed based on its weights. Activation functions are essential for backpropagation to ensure the model improves with each training epoch. Common activation functions include Rectified Linear Unit (ReLU), Sigmoid and Softmax. Sigmoid and Softmax are particularly important in classification problems, with Sigmoid outputting 0 or 1 for binary classification, and Softmax outputting probabilities for multi-class tasks. Because of this, an activation function is often used on the output layer. |
| Batch size | Not all data is given to the model at once during training for multiple reasons, one of the most important being memory usage. Large datasets use considerable amounts of memory, and RAM is a limited resource to the point where some datasets simply cannot fit in memory all at once. Additionally, by using smaller batches, training can be faster and less computationally intensive. |
| Validation split | Data is typically split across three sets when training machine learning models. These sets are the training set encompassing the vast majority of the data that the model will learn from, the testing set containing a small amount of data to evaluate the model's performance on data it has never previously seen and is used after training has completed, and the validation set containing a similar proportion of data to the testing set which is used during training to evaluate performance on unseen data, which helps to identify overfitting and to fine-tune hyperparameters. |
| Hyperparameters | Epochs, batch size and validation split are a few examples of hyperparameters, which are used to determine the model's behaviour during training. It is an essential part of model development to tune these hyperparameters to ensure the best possible model is produced. |

Table 3.1: Common concepts in any neural network.

## 3.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are deep learning architectures designed to process data with a grid-like topology, such as images. They consist of multiple specialised layers that extract hierarchical features from input data, enabling tasks like image classification and object detection. Table 3.2 details the key elements of CNNs.

| Feature | Description |
|---|---|
| Convolutional filter | Also sometimes referred to as a kernel. These filters are small matrices which effectively slide across an image, creating a 'feature map', or convoluted feature. The filter **must** be smaller than the image itself. A key feature of convolutional filters is that they use shared weights, ensuring that an object recognised in one part of an image (the number 5 in the middle of the image, for example) can be recognised in a different part of an image. Early convolutional filters will identify lower-level features such as large shapes, whereas further filters can learn deeper features such as specific objects or human faces. |
| Pooling layer | The feature map from convolutional filters cannot be directly passed into a new dense layer, and must first be pooled. Pooling layers reduce the dimensionality of convolutional filters, downsampling them while retaining their key information by summarising features within select regions dependent on the pooling kernel size, which works similarly to the convolutional filter size. There are two common pooling strategies: max pooling and average pooling, which are visually represented in Figure 3.2. |
| Max pooling | When summarising each region, the maximum value is taken. This emphasises more prominent features in the image, while rejecting less prominent features. |
| Average pooling | Computes the average value in each region, which allows more data to be retained at the possible expense of not emphasising key details. |
| Stride | The stride of the convolutional or pooling kernel determines how far the kernel moves every time it takes a sample. For example, a kernel with stride (2, 2) will move 2 pixels across to the right until it reaches the edge of the image, at which point it will move 2 pixels down. This also applies to the pooling layer, where it would move 2 columns across the feature map. |

Table 3.2: Key features of Convolutional Neural Networks.

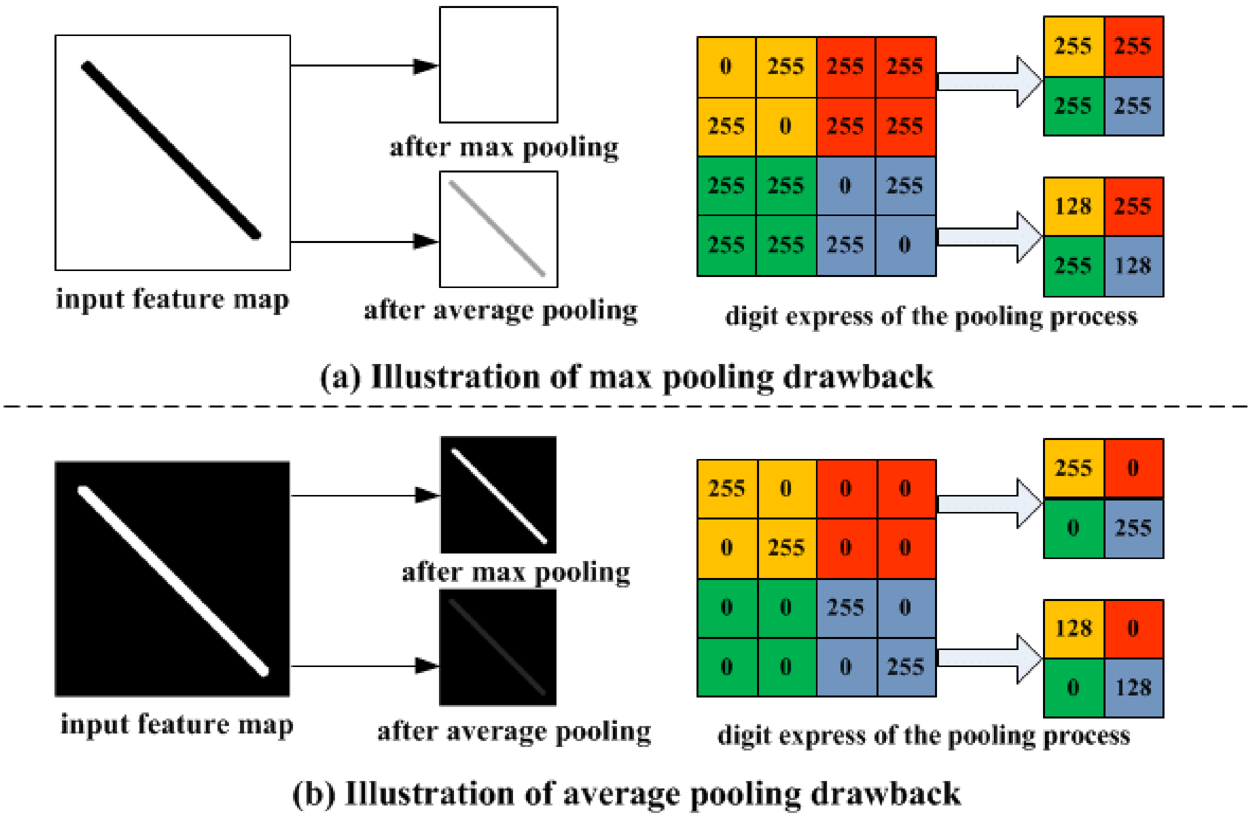



Figure 3.2: An example of average and max pooling (Yu et al., 2014).

# Proposed model

## 4.1 Architecture

As evidenced by the review of related work conducted across Chapter 2, the most suitable model for classifying the pneumonia chest X-rays will be a CNN due to their high proficiency in identifying image features compared to other models. Unfortunately, training a CNN to a good degree of accuracy requires immense amounts of training data which will not be found in this dataset alone. As such, it will be best to leverage transfer learning with pretrained models such as ResNet50 or EfficientNet due to their large training corpus and high performance even in scenarios with class imbalance such as the one presented by this dataset. The model will then have additional layers added for this specific binary classification task.

## 4.2 Preprocessing

As mentioned previously in Section 1.2.1, the dataset's images are not all the same resolution, and a class imbalance is present, so significant preprocessing steps will need to be taken; most notably data augmentation and image resizing. In doing so, we can ensure that all data fed to the input layer is consistent and usable in training and evaluating[1] the network.

## 4.3 Hyperparameters

The proposed model's hyperparameters for its initial development, which were previously defined in Table 4.1, will be as follows:

| Hyperparameter | Value | Explanation |
|---|---|---|
| Loss function | Binary cross-entropy | The objective of the model will be to classify images in **one of two** classes: having pneumonia or not having pneumonia. As such, a loss function designed for binary classification is necessary, with binary cross-entropy fulfilling this purpose. |
| Optimiser | Adam | Multiple works[2] utilise the Adam optimiser on this dataset and others, which allowed them to attain minimal loss values with their trained models. However, this parameter is subject to change, with SGD and RMSProp also in consideration. The best performing of the three will be used for the final model. |

[1]While data from the testing set will be resized so that it can be used as input for evaluation, it will not be augmented, as this would defeat the purpose of a train/test split.

[2]Specifically, those of Malik et al. (2023) and El Asnaoui (2021).

| | | |
|---|---|---|
| Epochs | 20 | While 20 may seem like a low amount of epochs especially in comparison to El Asnaoui (2021)'s 250-epoch model, it is important to note the processing power limitations and time constraints of this project, which are detailed in Appendix A. Furthermore, works with as few as 11 epochs (Mathur, 2020) still demonstrated excellent results, showing that more epochs do not directly correlate to higher model performance. |
| Input shape | (224, 224, 1) | The images are originally of variable high resolutions that would greatly impact the processing requirements to train the model. Resizing them to a standardised lower resolution minimises these requirements while also allowing them to be used for transfer learning, as this resolution is used for the input layer of ResNet50 models. The 1 in the input shape represents the colour channels, but because these are all greyscale images, there is only one channel. |
| Validation split | 80% to 20% | The training set will be split at an 80% training to 20% validation ratio to evaluate the model's performance as it trains. This maximises training data while still maintaining a good amount of data to evaluate on. |
| Evaluation metrics | Accuracy, F1-Score | See Section 4.4 for further information. |

Table 4.1: Proposed hyperparameter values and explanations for initial development.

## 4.4 Evaluation strategy

The model's performance against its testing data will be thoroughly evaluated, primarily using accuracy, loss and F1-Score as key metrics. Accuracy is defined as the total number of correct positive and negative classifications (true positive and true negative) divided by the total number of samples (including false positives and false negatives), which can be visualised as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

F1-Score is the harmonic mean of the model's precision and recall, providing an equally weighted combination of both statistics as one metric. It is particularly useful with imbalanced datasets, including medical datasets, where false classifications can have dire consequences.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.3}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

Additionally, these metrics can be paired with a confusion matrix, which is a simple grid showing the amount of data that was correctly classified as positive ('true positive'), negative ('true negative'), and incorrectly classified ('false positive', 'false negative'). An example of a confusion matrix is depicted in Figure 4.1.

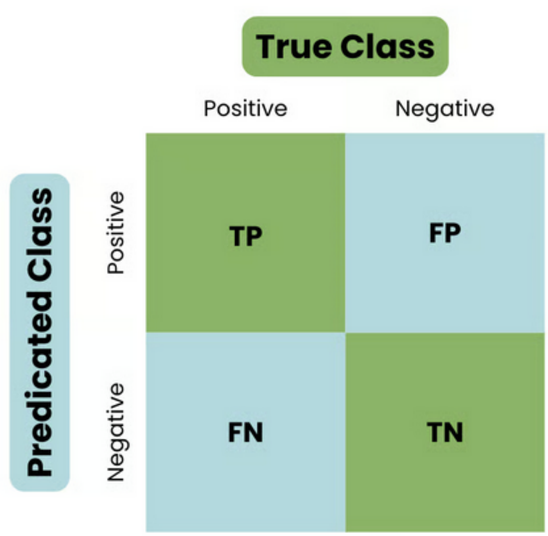



Figure 4.1: An example confusion matrix (DataCamp, 2025).

# Bibliography


Allinson, James Peter, Nishi Chaturvedi, Andrew Wong, Imran Shah, Gavin Christopher Donaldson, Jadwiga Anna Wedzicha and Rebecca Hardy (8th Apr. 2023). 'Early Childhood Lower Respiratory Tract Infection and Premature Adult Death from Respiratory Disease in Great Britain: A National Birth Cohort Study'. In: *The Lancet* 401 (10383), pp. 1183–1193. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(23)00131-9. pmid: 36898396.

DataCamp (2025). *What Is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained.* URL: https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning (visited on 19/03/2025).

El Asnaoui, Khalid (1st Mar. 2021). 'Design Ensemble Deep Learning Model for Pneumonia Disease Classification'. In: *International Journal of Multimedia Information Retrieval* 10 (1), pp. 55–68. ISSN: 2192-662X. DOI: 10.1007/s13735-021-00204-7.

HardwareDB (2025). *Tesla T4 vs Radeon RX 9070 XT GPU Comparison.* hardwareDB. URL: https://www.hardwaredb.net/tesla-t4-vs-radeon-rx-9070-xt (visited on 21/03/2025).

IBM (6th Oct. 2021a). *What Is a Neural Network? | IBM.* URL: https://www.ibm.com/think/topics/neural-networks (visited on 13/03/2025).

IBM (15th Oct. 2021b). *What Is Overfitting? | IBM.* URL: https://www.ibm.com/think/topics/overfitting (visited on 15/03/2025).

IBM (12th Dec. 2023). *What Is Support Vector Machine? | IBM.* URL: https://www.ibm.com/think/topics/support-vector-machine (visited on 13/03/2025).

Kermany, Daniel S., Michael Goldbaum, Wenjia Cai, Carolina C. S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y. L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A. N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia and Kang Zhang (22nd Feb. 2018). 'Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning'. In: *Cell* 172 (5), 1122–1131.e9. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2018.02.010. pmid: 29474911.

Malik, Hassaan, Tayyaba Anees, Muizzud Din and Ahmad Naeem (1st Apr. 2023). 'CDC_Net: Multi-Classification Convolutional Neural Network Model for Detection of COVID-19, Pneumothorax, Pneumonia, Lung Cancer, and Tuberculosis Using Chest X-rays'. In: *Multimedia Tools and Applications* 82 (9), pp. 13855–13880. ISSN: 1573-7721. DOI: 10.1007/s11042-022-13843-7.

Mathur, Madhav (2020). *Pneumonia Detection Using CNN(92.6% Accuracy).* URL: https://kaggle.com/code/madz2000/pneumonia-detection-using-cnn-92-6-accuracy (visited on 13/03/2025).

Mendeley (2025). *Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images - Mendeley Data.* URL: https://data.mendeley.com/datasets/rscbjbr9sj/3 (visited on 27/02/2025).

Mooney, Paul (2018). *Chest X-Ray Images (Pneumonia).* URL: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia (visited on 28/02/2025).

NHS (23rd Oct. 2017). *Pneumonia*. nhs.uk. URL: https://www.nhs.uk/conditions/pneumonia/ (visited on 27/02/2025).

Ortiz-Toro, César, Angel García-Pedrero, Mario Lillo-Saavedra and Consuelo Gonzalo-Martín (June 2022). 'Automatic Detection of Pneumonia in Chest X-ray Images Using Textural Features'. In: *Computers in Biology and Medicine* 145, p. 105466. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2022.105466. pmid: 35585732.

Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren and Andrew Y. Ng (25th Dec. 2017). *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. DOI: 10.48550/arXiv.1711.05225. arXiv: 1711.05225 [cs]. URL: http://arxiv.org/abs/1711.05225 (visited on 11/03/2025). Pre-published.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh and Dhruv Batra (Oct. 2017). 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization'. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626. DOI: 10.1109/ICCV.2017.74.

Sourab, Sazzad Yousuf and Md Ahasan Kabir (1st Jan. 2022). 'A Comparison of Hybrid Deep Learning Models for Pneumonia Diagnosis from Chest Radiograms'. In: *Sensors International* 3, p. 100167. ISSN: 2666-3511. DOI: 10.1016/j.sintl.2022.100167.

Stephen, Okeke, Mangal Sain, Uchenna Joseph Maduh and Do-Un Jeong (2019). 'An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare'. In: *Journal of Healthcare Engineering* 2019 (1), p. 4180949. ISSN: 2040-2309. DOI: 10.1155/2019/4180949.

Technical City (2025). *RX 9070 XT vs Tesla T4 [1-Benchmark Showdown]*. URL: https://technical.city/en/video/Tesla-T4-vs-Radeon-RX-9070-XT (visited on 21/03/2025).

Umar Ibrahim, Abdullahi, Mehmet Ozsoz, Sertan Serte, Fadi Al-Turjman and Salahudeen Habeeb Kolapo (2022). 'Convolutional Neural Network for Diagnosis of Viral Pneumonia and COVID-19 Alike Diseases'. In: *Expert Systems* 39 (10), e12705. ISSN: 1468-0394. DOI: 10.1111/exsy.12705.

Wang, Pin, En Fan and Peng Wang (1st Jan. 2021). 'Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning'. In: *Pattern Recognition Letters* 141, pp. 61–67. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2020.07.042.

WHO (2025). *Pneumonia*. URL: https://www.who.int/health-topics/pneumonia (visited on 27/02/2025).

Yu, Dingjun, Hanli Wang, Peiqiu Chen and Zhihua Wei (24th Oct. 2014). 'Mixed Pooling for Convolutional Neural Networks'. In: pp. 364–375. ISBN: 978-3-319-11739-3. DOI: 10.1007/978-3-319-11740-9_34.

# Appendix A - Model Training

Training convolutional neural networks is a highly intensive task, requiring substantial amounts of processing power, meaning many have to turn to cloud service providers to train their models. Many such options exist to find the necessary processing power to train an advanced CNN, such as Google Colab, which offers free (rate-limited) access to Tesla T4 GPUs. Personal experimentation has revealed that it can be difficult to leverage Colab when performing data augmentation, often encountering rate limitations resulting in forced session termination. Additionally, model training can take a long time on these GPUs, and there is always the possibility of a sudden internet connection loss that could cause all work to be lost.

Therefore, which Colab likely could be suitable for training the proposed model, it is instead also possible to leverage a local GPU which will be able to train the model at a substantially faster rate. The proposed training method will be a local Jupyter Notebook running on a desktop PC with an AMD Radeon RX 9070 XT graphics card. This GPU is not only substantially faster than a Tesla T4 as per HardwareDB (2025) and Technical City (2025) as well as personal anecdotal experimentation seen in Figures A.1 and A.2, but would also not impose rate limits due to it being personally owned. This should allow for the model to be trained much faster, while not having to depend on cloud service availability.

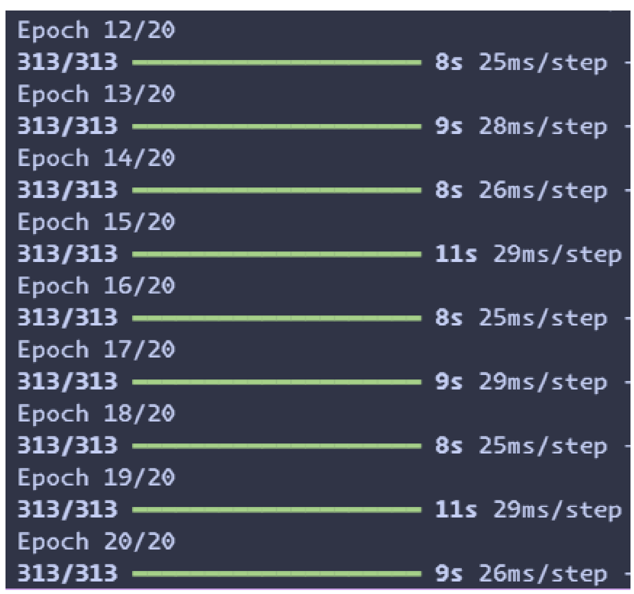



Figure A.1: Training time for a CNN on the CIFAR10 dataset using Colab (Approx. 27ms/step, 8-11s/epoch).

```
Epoch 11/20
313/313 [==============================] - 2s 7ms/step
Epoch 12/20
313/313 [==============================] - 2s 7ms/step
Epoch 13/20
313/313 [==============================] - 2s 7ms/step
Epoch 14/20
313/313 [==============================] - 2s 7ms/step
Epoch 15/20
313/313 [==============================] - 2s 7ms/step
Epoch 16/20
313/313 [==============================] - 2s 7ms/step
Epoch 17/20
313/313 [==============================] - 2s 7ms/step
Epoch 18/20
313/313 [==============================] - 2s 7ms/step
Epoch 19/20
313/313 [==============================] - 2s 7ms/step
Epoch 20/20
313/313 [==============================] - 2s 7ms/step
```

Figure A.2: Training time for a CNN on the CIFAR10 dataset using an RX 9070 XT (7ms/step, 2s/epoch).