

CZ4042 Neural Networks and Deep Learning

Group Assignment 2

| **Matric No.** | **Name** |
| --- | --- |
| U2020308K | Cheng Mun Chew |
| U2021558H | Tang Zhi Tern |
| U2022842C | Tang Han Kang |

# Abstract

Pneumonia is a disease which occurs in the lungs caused by a bacterial infection. Early diagnosis is crucial in a successful treatment process. Usually, the disease can be diagnosed from chest X-ray images by an experienced radiologist. The diagnoses can be subjective for some reasons such as the appearance of disease which can be unclear in chest X-ray images or can be confused with other diseases. Therefore, computer-aided diagnosis systems are needed to guide the clinicians. In this study, we used two well-known convolutional neural network models Resnet50 and Vgg16 for diagnosing pneumonia. We used transfer learning and fine-tuning in our training stage. The test results showed that the Vgg16 network and Resnet50 network both achieved high accuracy (>80%). However, the Resnet50 network achieved a more successful result in detecting pneumonia cases.

# 2. Introduction

Pneumonia occurs when there is inflammation of the tissues in either one of the lungs that is typically caused by a bacterial infection. According to the Ministry of Health, Pneumonia has been a top 3 leading cause of death in Singapore in the past few years [1]. Chest X-ray images are a common clinical method for diagnosing pneumonia. However, diagnosing pneumonia from chest X-ray images is a difficult task for even experienced radiologists. This is because the symptoms of pneumonia in X-ray images are often unclear, can confuse with other diseases and contain abnormalities. These inconsistencies caused considerable subjective decisions and varieties among radiologists in the diagnosis of pneumonia [2]. Therefore, there is a need for

Artificial Intelligence support systems to assist radiologists in diagnosing pneumonia from chest X-ray images. Recent developments in the deep learning field, especially in convolutional neural networks (CNNs) , displayed significant success in image classification [3]. Researchers developed different CNN-based deep neural networks and these networks achieved state of results in classification and computer vision [4]. There exists some studies regarding detection of pneumonia using deep learning. In 2017, Antin et al. used a DenseNet-121 layer transfer learning method and they achieved 0.60 % area under the curve (AUC) value [5]. In 2017, Rajpurkar, et al. proposed a 121-layer convolutional neural network based on DenseNet [6] and named it as CheXNet [7]. They trained their network with 10.000 frontal view chest X-ray images with 14 different diseases. They assessed the performance of their network with four expert radiologists on the f1 score metric which is the harmonic average of the precision and recall metrics. CheXNet achieved a f1 score of 0.435 (95% CI 0.387,0.481), higher than the radiologist average of 0.387 (95% CI 0.330, 0.442).

We will attempt to improve results through data preprocessing, and also modify and train two well-known networks for classifying pneumonia from chest X-ray images. . Our first network is based on the VGG16 model. The second one is the Resnet50 based model. In addition, we utilized transfer learning with fine-tuning and data augmentation methods.

# 3. Methodology

## **3.1 Dataset**

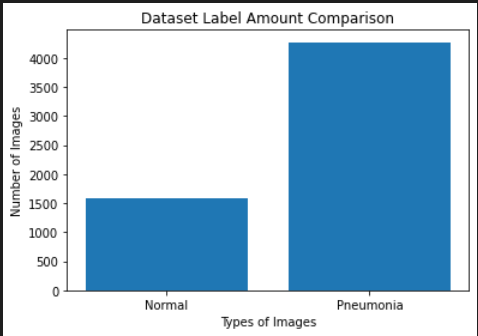
We used a publicly available dataset on Kaggle, “Chest X-Ray Images (Pneumonia)“. This dataset consists of chest X-ray images (anterior-posterior) selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children’s Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients’ routine clinical care [8]. It consists of 5863 images of chest X-rays, with 2 labeled categories - ‘0’ for Normal and ‘1’ for Pneumonia.

## 3.2 Dataset Analysis

Upon analysis, we noticed there was an issue of data imbalance and some images may be deemed “noisy”

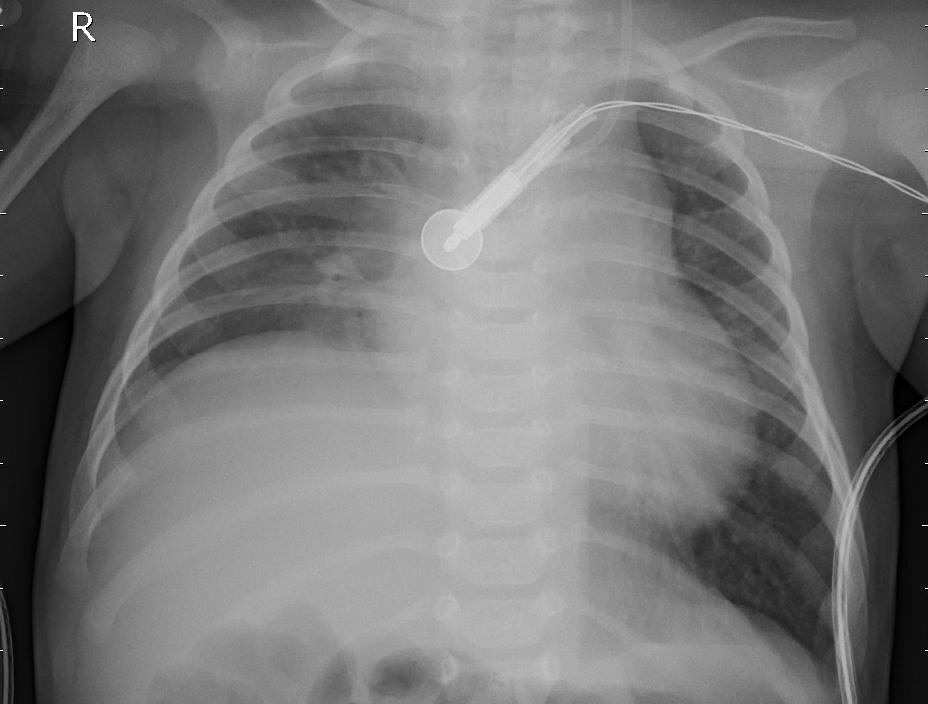
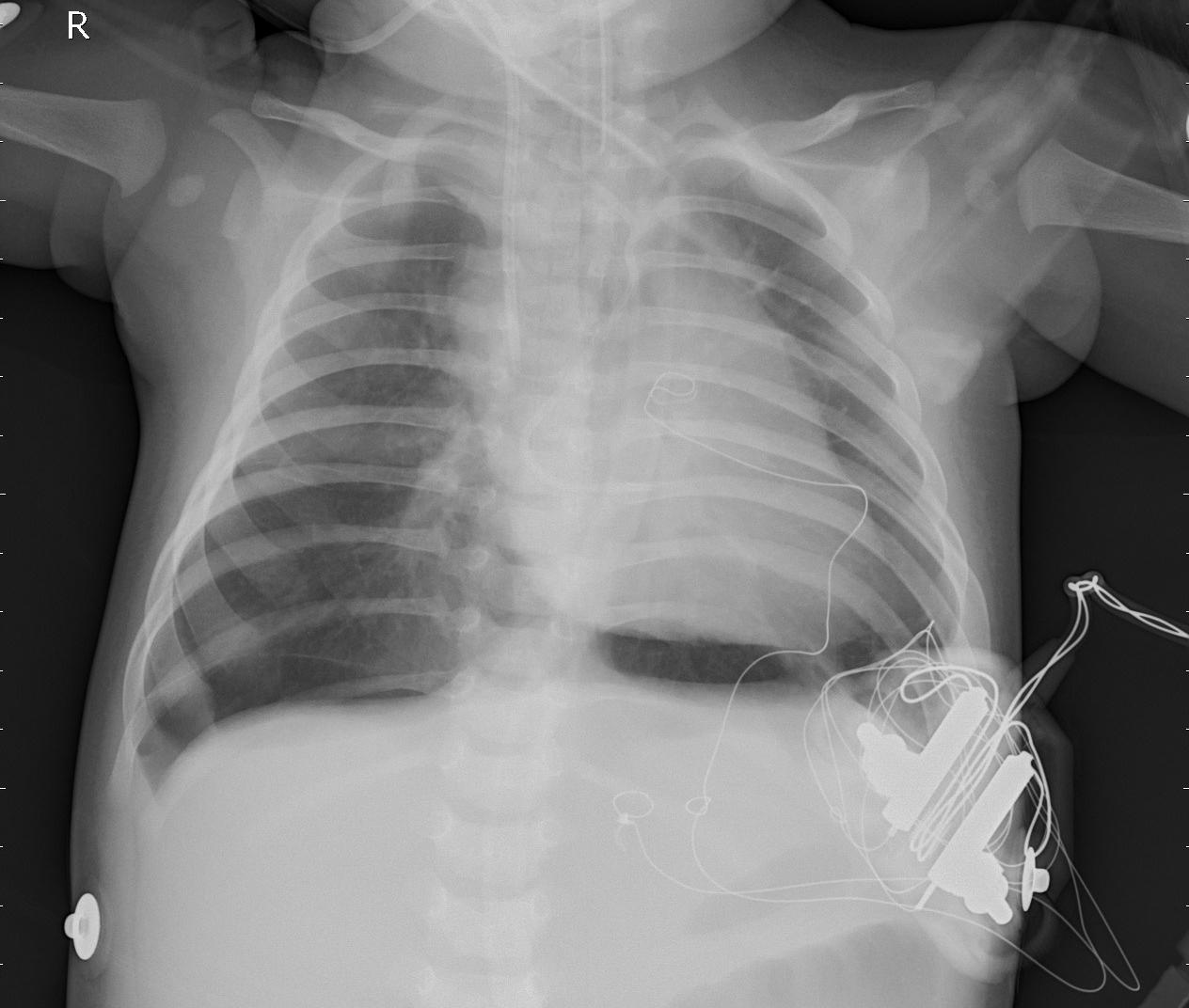
### 3.2.1 Data Imbalance

As shown in the bar plot below, there are significantly more Pneumonia labeled images in the dataset than Normal. The ratio of this imbalance is about **1:2.7**. In this scenario, we will experiment on some techniques and attempt to resolve this imbalance. The techniques will be covered in 3.2 Mitigating Data Imbalance.



### 3.2.2 Data Noise

There are images in the dataset with clearly visible medical equipments attached to the body. We decided to leave it in and continue the training process as we believe this will be a natural occurrence in future images to be predicted. In this addition, the images have been cleared by expert physicians for AI systems[8].



*Above are some examples of “noise” in the dataset.*

## 3.3 Mitigating Dataset Imbalance

Referencing the attempts of various top notebooks on kaggle[8], we will be implementing the techniques below to resolve Data Imbalance, and comparing the results against each other and a control set. All scenario will share similar model architecture and hyperparameters.

### 3.3.1 Adding Class Weights to Training Process

Class weights will be added to the training process to adjust the impact of each data’s contribution. This should allow the minority to contribute more evenly to the learning process. For example, if we were to train a model with just 4 images

* 1 Normal image
* 3 Pneumonia image

The follow will be the class contribution

* Normal: contributes 2x to the learning outcome
* Pneumonia: contributes 0.666…x to the learning outcome

The idea of using class weights

### 3.3.2 Data Augmentation

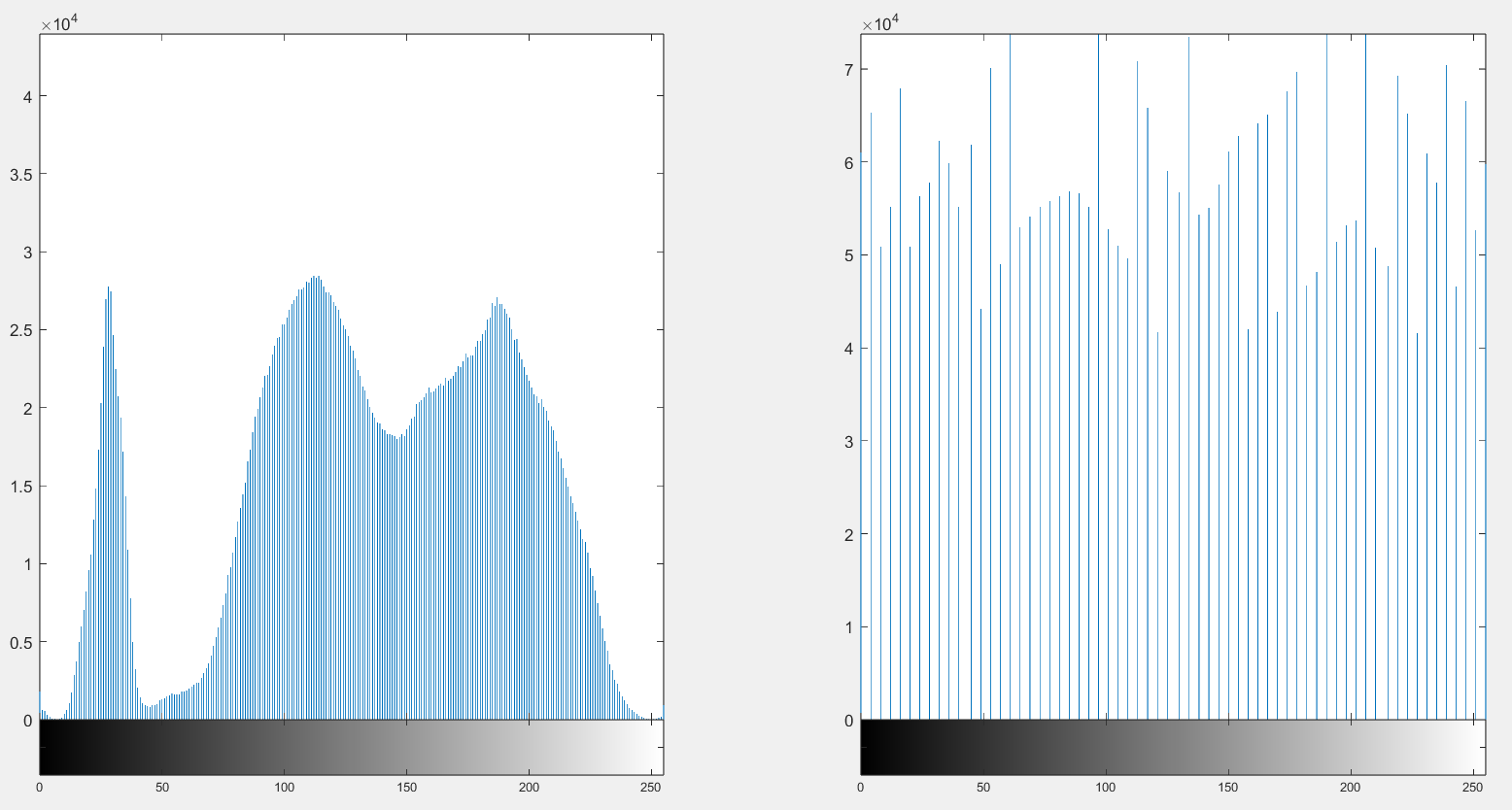
Data Augmentation will generate more data from current dataset, with added tweaks. We will generate more “Normal” data to make up for the imbalance. In our scenario, the amount of tweaks or modifiers we choose to use will be very limited as we believe a medical dataset such as X-Ray will be strictly controlled.

## 3.4 Dataset Preprocessing

We will be attempting to preprocess the data in hopes to tweak the inputs to be better processed by our CNN. We will generate an entirely new dataset based on the old one using the following methods.

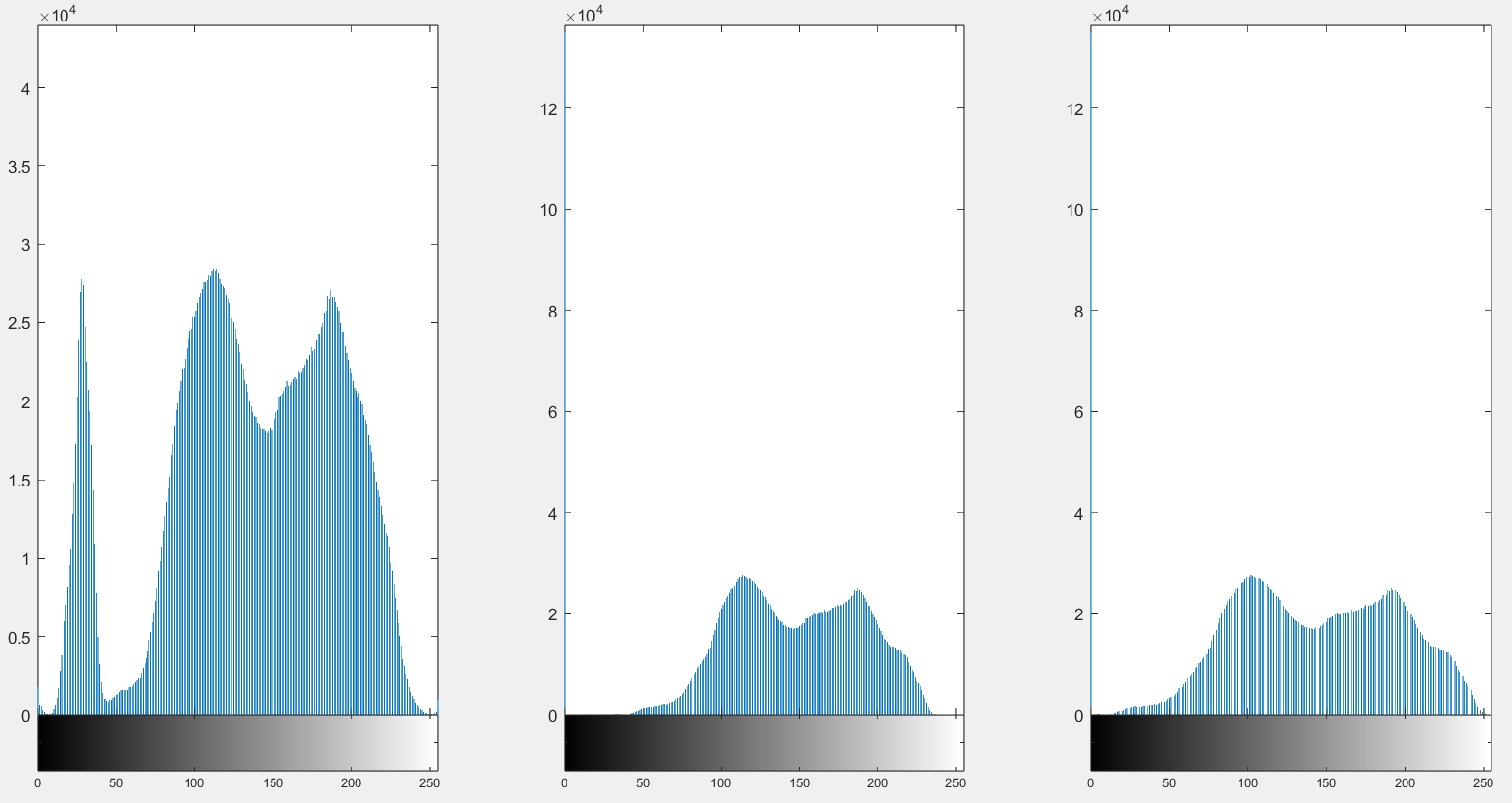
### 3.4.1 Contrast Stretching using Histogram Equalization

Histogram Equalization spreads out the most frequent intensity values of the image yet maintains the frequency of each intensity.



### 3.4.1 Customized Algorithm

We first set it to 0, the lowest 30% and highest 15% of pixel values. Next, we stretch the histogram bins, maintaining the shape. The idea behind this algorithm is to remove what we deem as unnecessary information, and allow the model to focus on learning from what matters in the images; which seemed to be white noise or spots around the intensity values of 75-200.



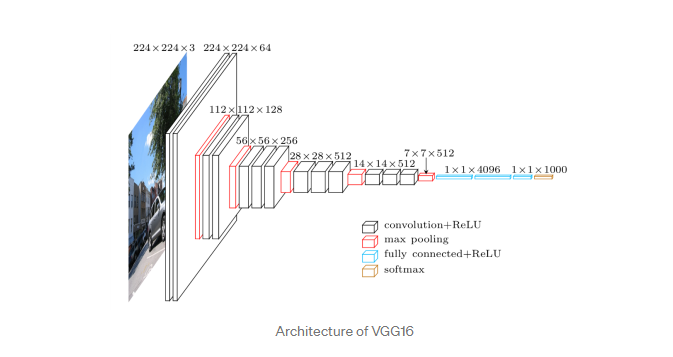
The above image shows histogram of

1. The original image
2. The image with lowest 30% and highest 15% set to zero. Notice an extremely high frequency count at pixel value 0.
3. The 2nd histogram but stretched to fit the range of 0 to 255

## 3.5 Proposed Architectures

### 3.5.1. VGG16

In 2014, Simonyan et al. proposed a deep neural network model called VGG-16 [9]. The model has 16 convolutional layers with small receptive fields (3x3), estimated 140 million parameters, five max-pooling layers (2x2 size) and three fully-connected layers, with the final output layer having a soft-max activation function.



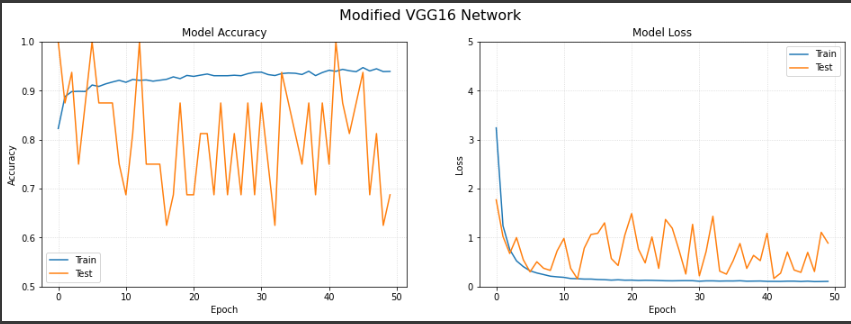
Source:https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

We used this model with pre-trained weights on ImageNet and modified fully connected layers of the network. VGG16 is capable of receiving inputs of dimensions 224x224x3 so our images are resized accordingly to feed into the input layer. The training parameters that we used were binary cross entropy as our loss function since there are only 2 classes to label, 50 epochs of training, RMSprop with a learning rate set to 1e-4 as our optimizer and batch size of 64. In order to reduce possible overfitting, we included l2 regularizers for weight decay set at 1e-2, added Dropout layers with 0.4 rate in the fully connected layers and BatchNormalization layer was also included to stabilize and speed up training. Class weights were used in training to reduce the impact of data imbalance.The figure below shows the architecture of our modified VGG16 model. This modified network was implemented using Keras deep learning framework in Python and trained on Google Collab GPU runtime session. The model took an estimated 120 minutes to train on the 5216 training image dataset and was validated against 16 validation images.

*Modified VGG16 Network*



*Plot of training/validation accuracy/loss against epochs*

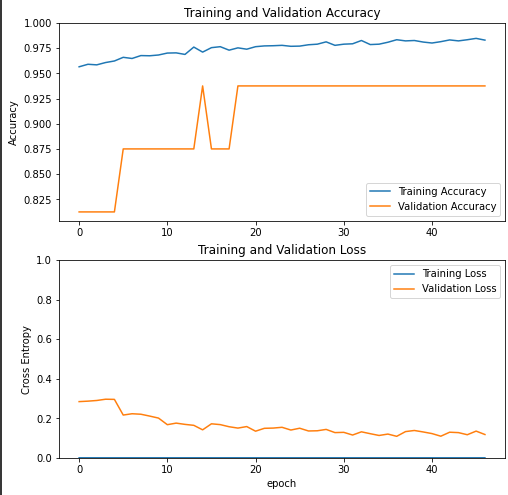


### 3.5.1. ResNet50

ResNet50 is a convolutional neural network (CNN) belonging to the deep residual network family, where residual blocks are stacked atop each other to form the network. Conventional CNN becomes more difficult to train with added layers, at the same time, the model's accuracy will also decrease. This can be attributed to the problem of vanishing and exploding gradients in CNNs.

In order to address this issue, ResNet utilizes “skip” connections in residual blocks. These connections provide an alternative route for the gradient to pass through during training, while enabling the model to learn identity functions. Identity functions ensure higher layers of the model perform just as well or better than lower layers, but never worse. The pros of ResNet makes it a viable option for our project.

For the classification problem, our group used transfer learning to reduce the probability of over fitting on our small data set. Using tensorflow built-in transfer learning functions, we can load the model weights that were trained on the image net dataset. Generally, in the earlier layers, the model is attempting to detect more general features (shapes, edges), while the last few layers are more focused on the classification task at hand. By removing the last few layers of the trained model and inserting layers needed for our binary classification task, we are able to capitalize on the model’s ability to detect general features, while saving time and effort in collecting data and training the model.



We also fine tuned the weights in the earlier layers, to study its impact on the model's accuracy. In order to perform fine tuning, we unfreezed certain layers of the ResNet50 model, before training it on the same set of training data. Since the number of layers to keep frozen is a tunable hyperparameter, the source code provided by our group allows for testing of multiple hyperparameter settings, study minimum loss at each setting, and decide on the optimal number of layers to keep frozen. Performing fine tuning, our model’s accuracy is improved slightly.

# 4. Evaluation and Results

We used our trained models to make predictions on the test split of the initial dataset. The test set consists of 234 ‘NORMAL’ and 390 ‘PNEUMONIA’ labeled data.

The models we used were evaluated using metrics such as accuracy, recall, precision and f1-score. The calculations for these metrics are as shown below:

* Accuracy =(TP+TN)/(TP+FN+TN+FP)
* Precision =TP/(TP+FP)
* Recall =TP/(TP+FN)
* F1 score=2 x (precision x recall) / (precision + recall)

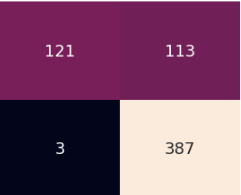
TP, TN, FN, FP refers to the number of true positive, true negative, false negative, false positive respectively.

## 4.1 Data Preprocessing

|  | **Accuracy** | **Precision** | **Recall** | **Loss** |
| --- | --- | --- | --- | --- |
| **Default Data** | 0.697952 | 0.697952 | 1.000000 | 0.561974 |
| **Equalized Data** | 0.887372 | 0.911271 | 0.929095 | 0.342857 |
| **Custom Data** | 0.875427 | 0.913793 | 0.907090 | 0.300132 |

## 4.2 VGG16

*Confusion matrix of VGG16 predictions on test data*

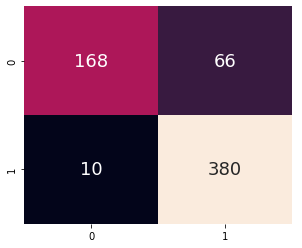


*Figure XX: Table of VGG16 performance metrics predictions on test data*

| **VGG16** | Accuracy | Precision | Recall | f1-score |
| --- | --- | --- | --- | --- |
| NORMAL | **0.81** | **0.98** | **0.52** | **0.68** |
| PNEUMONIA | **0.77** | **0.99** | **0.87** |

## 4.3 ResNet50

*Confusion matrix of ResNet50 predictions on test data*



*Table of Resnet50 performance metrics predictions on test data*

| **Resnet50** | Accuracy | Precision | Recall | f1-score |
| --- | --- | --- | --- | --- |
| NORMAL | **0.88** | **0.94** | **0.72** | **0.82** |
| PNEUMONIA | **0.85** | **0.97** | **0.91** |

# 5. Conclusion

In summary, with the use of transfer learning techniques, we can circumvent the need for lots of new data and speed up the training process. It may also achieve better performance than other networks. However, there are some challenges. This includes fine tuning to ensure the model learns features specific to the current task. Negative transfer may also occur, leading to a decrease in the model’s performance.

Going forward, we think that more medical images such as bone fracture x-rays, Magnetic Resonance Images (MRI) scans can also be diagnosed with the use of deep learning neural networks.

# REFERENCES

[1] <https://www.moh.gov.sg/resources-statistics/singapore-health-facts/principal-causes-of-death>

[2] M. I. Neuman et al., "Variability in the interpretation of chest

radiographs for the diagnosis of pneumonia in children," Journal of

hospital medicine, vol. 7, no. 4, pp. 294-298, 2012.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification

with deep convolutional neural networks," in Advances in neural

information processing systems, 2012, pp. 1097-1105.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image

recognition," in Proceedings of the IEEE conference on computer vision

and pattern recognition, 2016, pp. 770-778.

[5] B. Antin, J. Kravitz, and E. Martayan, "Detecting Pneumonia in Chest

X-Rays with Supervised Learning."

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely

connected convolutional networks," in Proceedings of the IEEE

conference on computer vision and pattern recognition, 2017, pp. 4700-

4708.

[7] P. Rajpurkar et al., "Chexnet: Radiologist-level pneumonia detection on

chest x-rays with deep learning," arXiv preprint arXiv:1711.05225,

2017.

[8]<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

[9]K. Simonyan and A. Zisserman, "Very deep convolutional networks for

large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[10]<https://www.tutorialspoint.com/dip/histogram_equalization.htm>