CNN Architecture Performance Comparisons:
Building Neural Networks to Classify X-Ray Images

Dataset: https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia
Code file: CA3_code.ipynb

Michael Ross (SID: 201589412)
Shuhei Ishiwatari (SID: 201603195)
University of Liverpool
COMP 534: Applied AI
Assignment 3
11 May 2022

Background

The purpose of this project is to design a Convolutional Neural Network (CNN) to classify x-ray images as either Covid-19, pneumonia, or normal and compare these results using pretrained CNN architectures. The x-ray images have been provided by Kaggle (link provided above) with approximately 5,000 training and 1,200 testing images. The aim of this project is to observe the architecture of two pretrained CNN models and attempt to improve the accuracy one of these models.

Libraries Used

Pytorch:
- nn.functional
- optim
- Cudnn
- Torchvision
- datasets
- models
- transforms
- WeightedRandomSampler
- TensorDataset
- DataLoader

Data Processing:
- Matplotlib.pyplot
- Pandas
- Numpy
- Seaborn
- Sklearn.metrics: confusion_matrix

Miscelleneous:
- Time
- Counter

Pretrained Models Selected

For this project, we selected the Visual Geometry Group Convolutional Neural Network (VGGNet) and the Residual Network (ResNet) pretrained models to use for performance comparisons in our x-ray image classification set. More specifically, our project utilises VGG-16 and ResNet-18 where 16 and 18 both refer to the number of layers for their respective networks.

VGGNet is a CNN model proposed by researchers at the University of Oxford and has respectfully won second place at ImageNet Large Scale Visual Recognition Challenge (ILSVRC), achieving approximately 92.7% accuracy in the ImageNet dataset. To this day, VGGNet is one of the most popular object recognition architectures widely deployed in many applications around the world, with its ability to classify images into 1000 separate classes. By observing VGGNet's architecture, we can observe how this is accomplished.

VGGNet consists of 13 convolutional layers and three fully connected layers with an input layer that takes in images of size 224x224x3. What's great about the VGGNet structure is that is uses the smallest possible filter (3x3) during the convolving process with a stride of 1 to capture even greater number of correlations between pixels in an image. It then passes through a max pooling layer of size 2x2 with a stride of 2 using a ReLu activation function before passing through the next layer. VGG attempts to reduce pixel volume at each convolutional layer where the last three layers are the fully connected layers where the final pooling layer is flattened to output the softmax in order to predict class. The initial structure of VGGNet is designed to output probabilities of 1000 classes which we changed for our problem set which consists of only three classes (Covid-19, Pneumonia, Normal).

Resnet actually outperformed VGGNet on the same ImageNet dataset accomplishing this with only a 3.57% error rate. Resnet's primary concept is the implementation of a residual block, which in simplified terms essentially allows for the utilisation of much deeper neural networks by using a fastforwarding or shortcut process. Moreover, this process overcomes a common problem in many deep neural networks called vanishing gradients by allowing backpropagation to take shortcut paths, reducing the time it takes to calculate loss and subsequently reducing the chances of vanishing gradient.

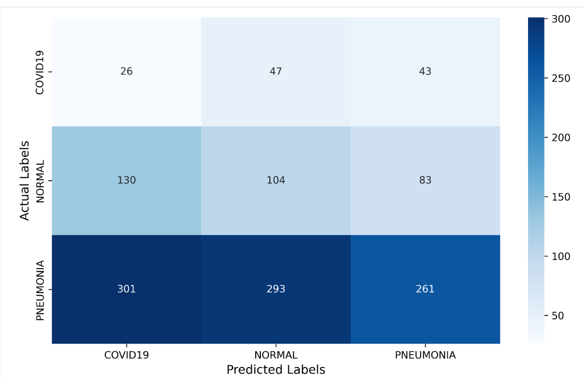
Data Processing and Analysis

The purpose of this assignment was to observe the pretrained models' performance on a completely new dataset. As stated previously, VGGNet and Resnet were both designed initially for the ImageNet datasets. Keeping this in mind, the parameters that make up the architecture of the respective models did not fit with the x-ray imaging dataset. The only change we made to these models was the final output layer to match the number of observed classes in our dataset, which was three.

Prior to testing these models on our data, we first imported the images from Kaggle and applied an image normalisation function by first converting the images into a PyTorch tensor and scaling the pixel values between 0 and 1. This is an important step in image classification as it reduces the skewness of the data by transforming the images to have a mean and standard deviation of 0 and 1 respectively and thus resulting in a faster and reliable learning process.

We then observed the number of class instances and immediately noticed that there was a significant class imbalance which could give heavier weight and consideration to the class with the most instances. As an experimental process of this project, we created two separate datasets to load into our models - an original dataset with class imbalance and another one where we applied oversampling to create class balance. Due to the significant amount of resources required to run deep neural networks, we ultimately decided to run the original dataset for the pretrained models while running both into our improved model which will be discussed later in this report.

Performance of Pretrained Models



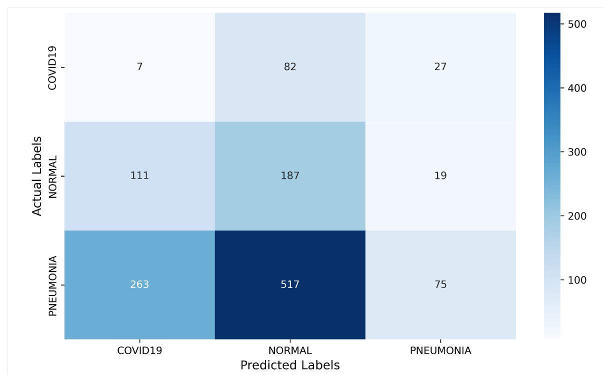


Figure 1. Left: *Confusion matrix of pretrained ResNet18. Classification accuracy was 65.3%.* Right: *Confusion matrix of pretrained VGG16. Classification accuracy was 35.9%.*

As predicted, both pre-trained models performed poorly in predicting the x-ray image classes without fitting the models to our training dataset (Figure 1). However, the performance of ResNet18 is almost twice as efficient as that of VGG16, which is in line with previous research suggesting ResNet based models are superior to other counterparts in the similar problem (Elgendi et al., 2020). Furthermore, the poorer performance of VGG16 indicated that adding more layers to the network would lead to the vanishing gradient problem. Hence, based on the initial result and previous research, the new architecture of ResNet18 was proposed for the next part of the project.

Proposed Network

Layer Type	**Output Shape**
ResNet18 (without output layer)	[1, 512, 7, 7]
1st convolutional layer	[1, 128, 4, 4]
2nd convolutional layer	[1, 128, 4, 4]
Flatten layer	[1, 2048]

1st fully-connected layer	[1, 256]
2nd fully-connected layer Dropout layer	[1, 256]
3rd fully-connected layer	[1, 256]
4th fully-connected layer Dropout layer	[1, 256]
5th fully-connected layer Softmax layer Output layer	[1, 3]

Table 1. The architecture of the proposed network ResNetXray

Table 1 shows the proposed new architecture (ResNetXray) for the classification problem. Two additional convolutional layers were added on top of the pretrained ResNet18, followed by five fully-connected layers including two dropout layers. The current architecture stems from the work of Youssef et al. (2020) where the author's model achieved 97.65% accuracy of classifying pneumonia from x-ray images. However, the following changes were made to tailor for the current classification problem, that is:

- Youssef et al. (2020)'s model is based on ResNet50v2. The ResNetXray model is based on ResNet18.
- The ResNetXray model applied a dropout rate of 0.2 as opposed to 0.7 in Youssef et al. (2020).

Due to the limited access to computational resources, ResNet18 was used over other alternatives to achieve faster training. Furthermore, it was observed that training accuracy plateaus at 80% during the initial testing, dropout rate was therefore set at 0.2 as the current network is smaller than that of Youssef et al. (2020).

Training and Testing Process

Due to the lengthy process of training and testing deep CNN architectures for image classification, our group concluded that it would be best to run a train and test together under one function. To ensure we properly recorded the train and test accuracies of our improved model on the kaggle dataset, we made sure to use a plot accuracy function to show the differences between the train and test accuracies. However, with our code consisting of two separate loaders (an original dataset and an oversampled dataset), we ran our train_test_model function twice.

We made sure to keep the integrity of the ResNet architecture within our improved model, but did ensure that our improved model added two altered parameters. First, we defined our loss function to be CrossEntropyLoss and used the Adam optimiser during backpropagation. Due to the limitation of our computing resources, we decided against conducting cross validation for optimising our hyperparameters.

Analysis of Improved Model

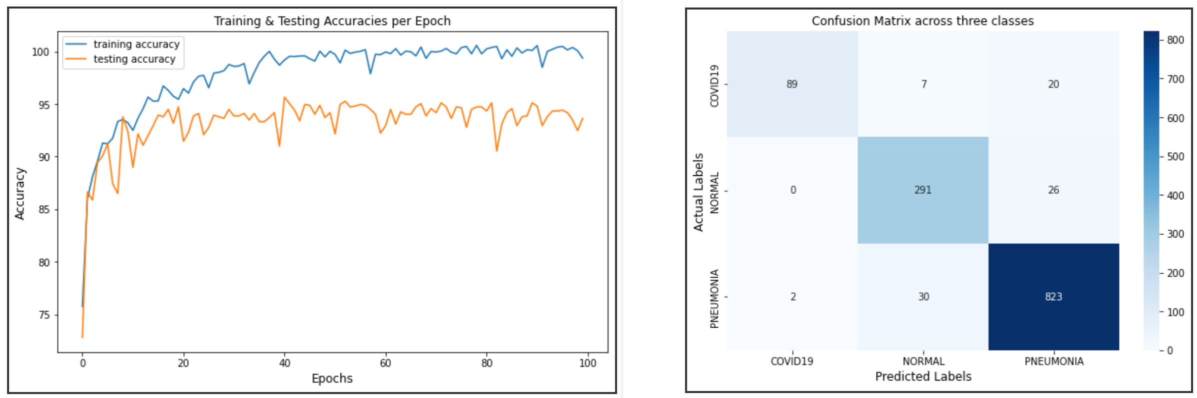


Figure 2. Left: *Train/test accuracies over 100 epochs.* Right: *Confusion matrix of the ResNetXray. Figures shown are based on our original dataset.*

The ResNetXray model performed outstanding in our initial test using the original dataset without oversampling. The accuracy, recall, and precision are shown in table 2. Classification accuracy improved significantly from 65.3% of the pretrained ResNet18 model to 93.3%. Exploration of the confusion matrix (Figure 2) suggested that the ResNetXray model could successfully classify the medical images despite being heavily imbalanced.

	ResNetXray	
	Non-oversampled data	**Oversampled data**
Accuracy (%)	93.25	94.41
Recall (%)	93.25	94.41
Precision (%)	93.25	94.41

Table 2. *Comparison of accuracy, recall and precision after rounding each value to two decimal places. The values are calculated using the micro-average scheme. Since this is a multi-class classification task, the values are identical on the scheme.*

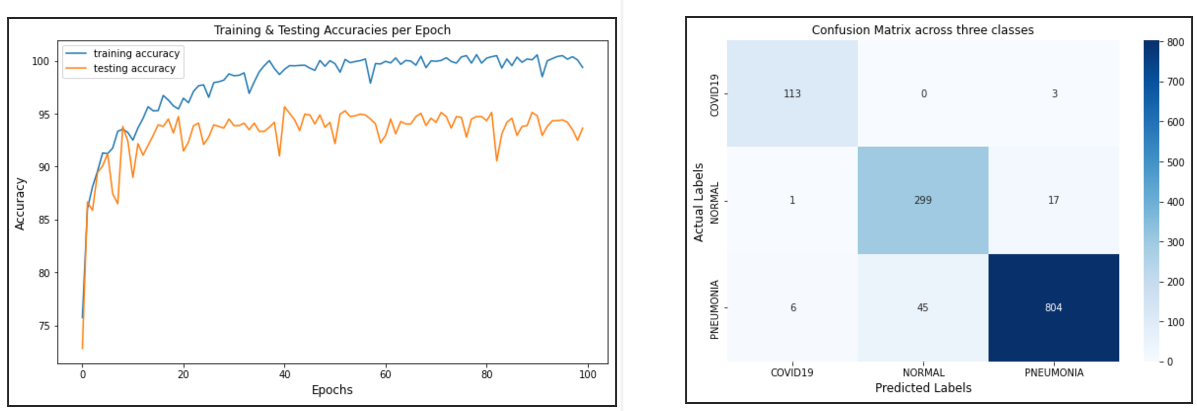


Figure 3. Left: *Train/test accuracies over 100 epochs.* Right: *Confusion matrix of the ResNetXray. Figures shown are based on our oversampled dataset.*

The ResNetXray model performed slightly better using the oversampled dataset (Table 2), but does not show any significant improvement. It is important to note that as the epochs tested increase, the accuracy of both of our models tend to stagnate and actually indicates that too many epochs result in overfitting. Ideally, we should have limited our epochs to anywhere from 10-20, which would still result in >90% accuracy. Another observation from the confusion matrix (Figure 2) is that, although the oversampled data resulted in approximately 1% better performance than our original dataset, specific instances such as incorrect predictions for pneumonia increased, which would not be a good practise in a real-world setting. By that definition, our model actually had a slightly better performance, in terms of recall, for our original dataset.

Future Work

In future research, one can implement other alternatives of ResNet models, such as ResNet34 or ResNet50, as additional layers are generally associated with better performance in classifying x-ray images (Elgendi et al., 2020). Furthermore, the only data augmentation applied in the current experiment is class-balancing oversampling. Other image augmentation techniques, such as horizontal flipping, translation and rotation (Yoo et al., 2020), histogram equalisation and gaussian blur (Giełczyk et al., 2022) has been suggested to increase a classifier's performance. Combined with the suggested tools and new architectures, the ResNetXray model can further be improved for the current classification problem.

Challenges of the Project

The process of deep learning is an incredibly complicated topic that has a great amount of depth to it. Our group overcame the challenges of understanding the intricacies of the pretrained models readily available in CNN implementations for image classification, but ran into several issues when running these models due to the significant amount of computational resources required to execute these models.

We learnt this lesson in the previous assignment in which we were required to only implement basic neural networks using no more than 5 layers, but with the pretrained architectures of CNN's, we were now dealing with neural networks with 16 layers for VGGNet and 18 for Resnet respectively. Furthermore, our improved model added even more layers on top of Resenet which proved to be too demanding even for cloud-based services such as Google Colab.

Ultimately, for this assignment, we had no choice but to purchase a Google Pro + subscription to have access to a far greater amount of GPU, CPU, and storage space. With our new subscription, we experienced a drastic reduction in train/test times for our models, but still proved to be a hindrance in freely testing out our code.

Project Allocation

	Michael Ross	Shuhei Ishiwatari
Pretrained CNN model research	x	x
Data Importing/Cleaning	x	
Data Analysis	x	x
Oversampling Dataset	x	
VGGNet Implementation	x	
ResNet Implementation		x
Improved ResNet Model		x
Performance Metrics		x
Report Writing	x	x

References

Youssef, T.A., Aissam, B., Khalid, D., Imane, B. & Miloud, J. (2020). Classification of chest pneumonia from x-ray images using new architecture based on ResNet. *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science.* 1-5 December.

Elgendi, M., Nasir, M. U., Tang, Q., Fletcher, R. R., Howard, N., Menon, C., Ward, R., Parker, W., & Nicolaou, S. (2020). The Performance of Deep Neural Networks in Differentiating Chest X-Rays of COVID-19 Patients From Other Bacterial and Viral Pneumonias. *Frontiers in medicine,* 7(1).

Yoo, SH., Geng, H., Chiu, TL., Yu, SK., Cho, DC. & Heo, J. Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. (2020). *Front Med*. 7. pp. 427.

Giełczyk, A., Marciniak. A., Tarczewska, M. & Lutowski, Z. (2022). Pre-processing methods in chest X-ray image classification. *PLoS ONE.* 17(4). pp. 1-11.