Brian Joram Wandobire
Department of Computer Science
Makerere University
2100702314, 2021/HD05/2314U
email: brianjoramwandobire@gmail.com

Convolution Neural Networks for Prediction of Chest Opacities

1. Introduction

A dataset of images captured using ultrasound is provided to train a deep learning model on the known images it contains. This dataset also has a set of unknown images the model is used to classify and are stored in a csv file. A second dataset which is captured under different conditions is also provided to test how well the model performs on such images.
A 12 layer Convolutional Neural Network(CNN) is created to predict this binary classification problem for the classes sick or normal. The input layer is not counted as part of the layers.
We use python, keras, tensorflow libraries through google colab and upload our work to a repository.
The sections covered are as follows; section2 looks at related work for deep learning in medical imaging, section 3 looks at the data exploration and preprocessing, section 4 discusses the methods used to achieve the model structure, section 5discusses the results and various evaluation techniques used, section 6 concludes by highlighting possible areas of improvement for future work.

2. Related Work

Deep learning is a subfield of Artificial Intelligence(AI)/Machine learning that uses algorithms based on the structure of the human brain to perform complex tasks/operations. It is best applied in applications where a problem is too large or complex for humans and normal algorithms[1]. These algorithms can be referred to as Artificial Neural Networks(ANNs).
Part et.al design a model and demonstrate how it achieves higher performance on lesion detection and image classification for chest radiographs compared to observers with varying levels of experience in the same[2]. In [3], Sim et.al show that radiologists had better performance in detection of abnormalities using CNNs.

3. Dataset Exploration and Preprocessing

A dataset of 715 images provided for dataset1 known images

3.1. Data Exploration

The dataset1 known images provided are found to be 715, these are explored and found to be in the binary clases of normal and sick. The images are viewed as in figure one below, however they do not make much sense to the untrained human eye. The images are also in the Portable Network Graphics(.png) format.

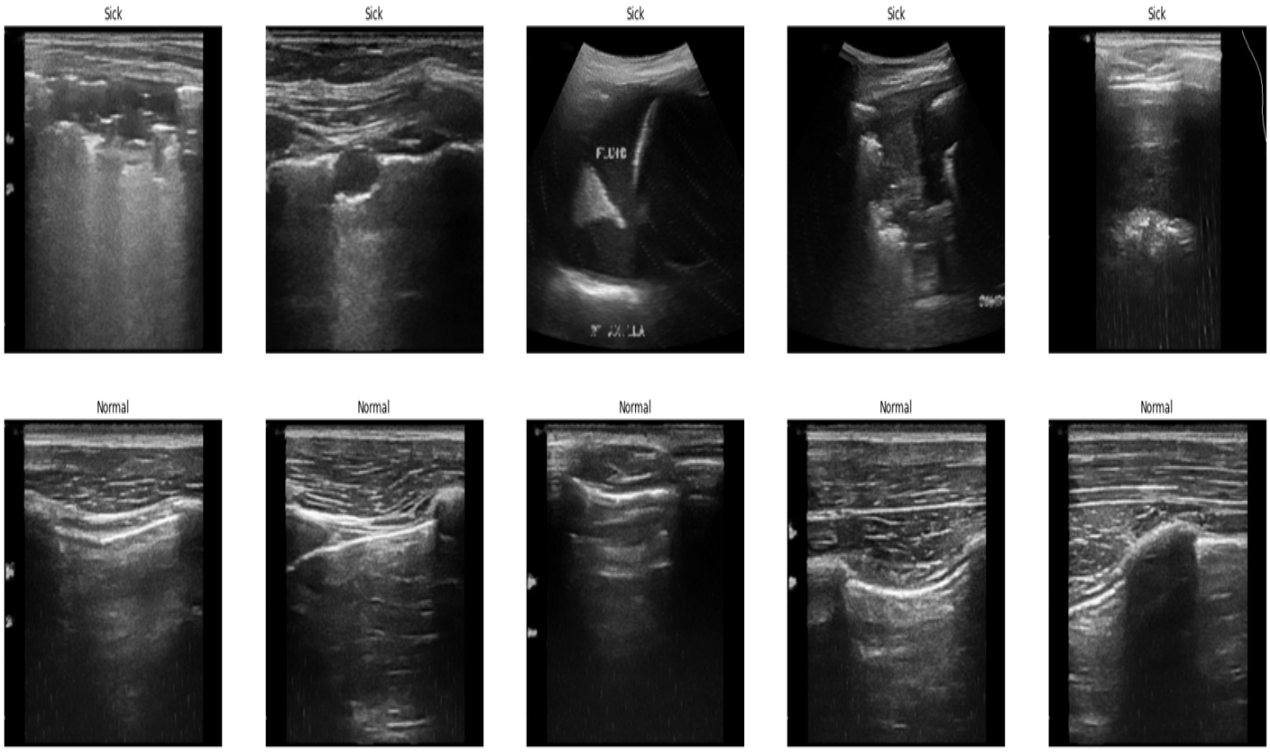

Figure 1: Images for both sick and normal cases from dataset 1

3.2. Preprocessing Steps

The dataset is split into test, train and validation sets in the ratio of 0.7, 0.15, 0.15. The test dataset is explored and found to be imbalanced as shown in figure 2 with 254 normal images and 245 sick images represented. The normal and sick samples retrieved can be observed from figure 1 above.

There is a chance that images provided may either be in grayscale or color, and starting with validation, these are converted to color for 3 channels. The validation dataset consists of 106 images.

The images are also all resided to 224 x 224. These are then converted to numpy arrays. Augmentation is also done to help with solving imbalance and also to facilitate in generating more distinct images from the sparse dataset. This is achieved through a combination of horizontal scaling, rotation, blurring and brightness manipulation of the images using a python library for image augmentation, imgaug.

A data generator function is defined. This uses techniques from imgauag to augment and generate more images. It one-hot encodes the images generated to the respective classes based on the label of base image used, and helps in alleviating the challenge of sparse data or few images.

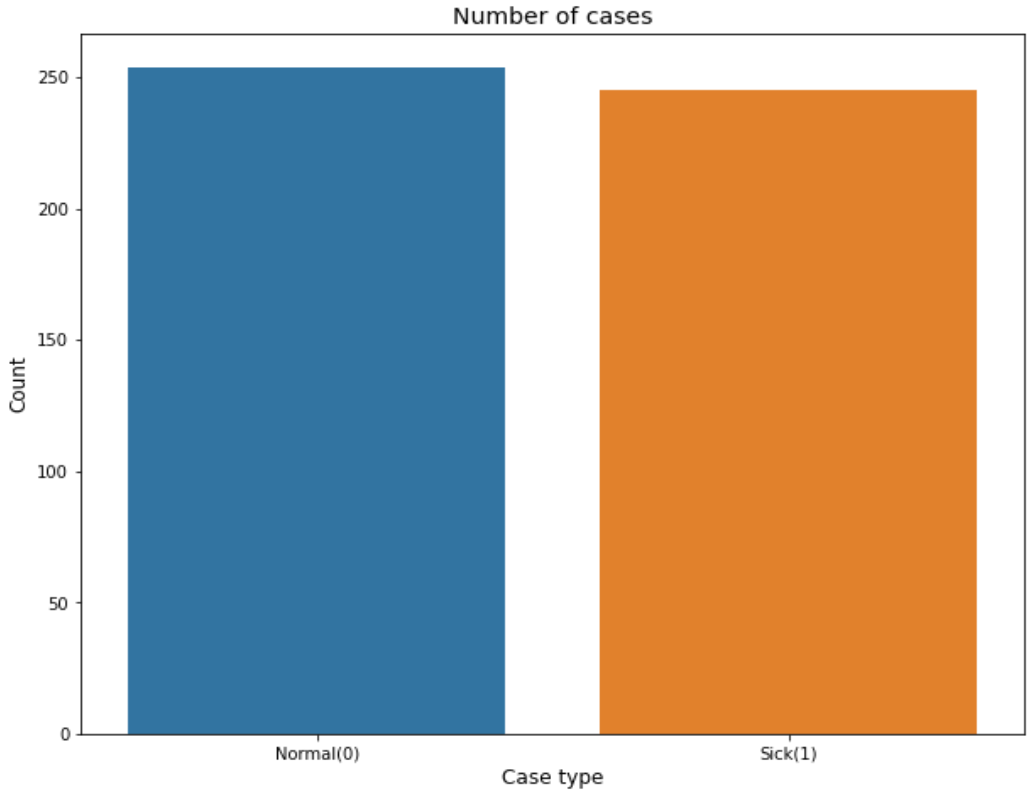

Figure 2: Class distribution observed for normal and sick during data exploration

4. Methods & Approach

We build a 12 layer CNN from scratch consisting of layers shown in figure 3 and do some partial transfer learning using weights for Visual Geometry Group 16 (VGG16) model, a 16 layer CNN that takes in images of 224x224 dimensions and 3 channel inputs for color. It is pretrained on more than a million images from the ImageNet database to classify a number of objects, diseases based on image diagnosis inclusive [2]. Previously, we attempted to use a 25 layer model with over 100,000,000 parameters but did not see any difference from models with fewer layers for this application scenario.

We use depthwise convolution by applying a single convolutional filter for each input channel; keeping each channel separate through these steps; split and filter the input into channels, convolve each input with the corresponding filter and stack together the convolved outputs. The advantage of depthwise convolution over normal convolution is that it introduces less parameters to deal with which is important as the complexity/size of our model increases.

Using a vgg16 weight file with pre-trained weights of the model, we initialize the weights with first weight at the first layer(Conv1_1 as in figure 3) of our model to extract general features of an image, thus the 224x224 dimension since this is the size vgg16 can read. Our kernel remains 3x3 throughout because this is the size theat VGG16 supports/ works with. The rest of the convolutional layers in the model operate in this way but with varying filters and also the VGG16 weights applied to the other convolutional layers to extract details like patches , edges, etc. We use padding in all the convolution layers to not lose data due to pixels that may be left out, usually at the boundary.

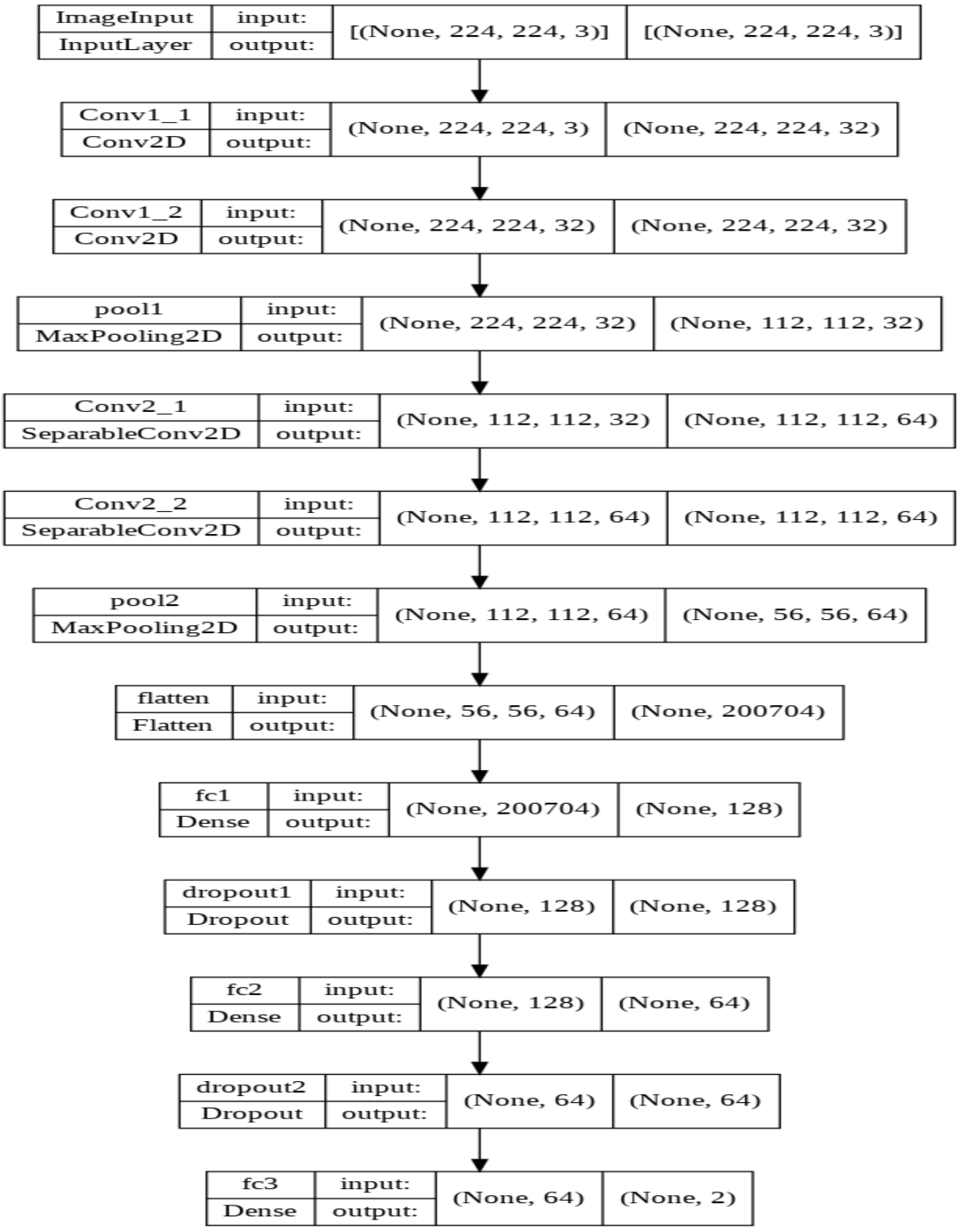

Figure 3: Layers of our CNN Model

The maxpool layers reduce the size of feature maps through a function to summarize subregions, such as taking the average or the maximum value when sliding a filter window[4]. This is an important step to reduce overfitting and make computations faster.
The flattening layer converts our data into a one dimensional array(long feature vector) to pass onto the next layer
The rectified linear activation unit(relu) is used in both the convolution and dense layers. It prevents the exponential growth in the computation that would be required to operate the neural network. The softmax layer is used in the last layer(which is dense), to convert the scores that are not properly scaled to a normalized probability distribution, and in this case our predicted classes.
The dropout layers also play a role to prevent overfitting. We use a dropout of 0.5 and 0.8 to introduce randomness to the data during training.
The three dense/fully connected layers provide output arrays of size 128, 64 and 2 respectively. The final dense layer of two equals the number of classes to be predicted.
Finally, Our model has 25,715,906 parameters, and all are trainable.

We use the Adam optimiser with a learning rate of 0.0001 and decay of 1e-5.
The early stopping is set to after 5 epochs of no improvements during training.
We use binary cross entropy since we have a binary classification problem, to compare each of the predicted probabilities to actual class output.
We use the batch size of four and set a threshold of 50 epochs through which the model can train.
We also set the class weight to deal with data imbalance initially seen during data exploration with 0.98 for the normal class and 1.02 for the sick class.

5. Results and Discussion

We obtained two sets of results from our predictions. The first were the results of the test set for dataset one. The second were for dataset two where the labels were known. For dataset 1 unknown images, the predicted labels were saved to a csv file to be handed in with this report.
A 0 represents a normal case, while a 1 represents a sick case.
The classification reports and confusion matrices of the different datasets are used to evaluate the model performance.

5.1. Dataset 1 Results

Dataset 1 had a total of 110 images from the test set on which to predict.
The model had an accuracy of 94% on the test data as observed from the classification report for rest Dataset 1
. With only two images for the sick class being misclassified. With a precision of 91.23, the model is easily able to distinguish positive and sick classes well as observed from the confusion matrix where there are only 5 false positives detected as seen in the confusion matrix for figure

5. With a recall of 96.3, the model is easily able to predict positive classes correctly as observed from the confusion matrix where only two positive clases were misclassified(false negatives). After 35 epochs, the model stopped learning and the early stopping came into play to prevent overfitting as seen in figure 4.

Classification Report for Test Dataset 1

	precision	recall	f1-score	support
0	0.96	0.91	0.94	56
1	0.91	0.96	0.94	54
accuracy			0.94	110
macro avg	0.94	0.94	0.94	110
weighted avg	0.94	0.94	0.94	110

Recall on Test Data: 0.963
Specificity on Test Data: 0.9107
Accuracy on Test Data: 0.9364
Precision on Test Data: 0.9123
F1 Score Test Data: 0.9369

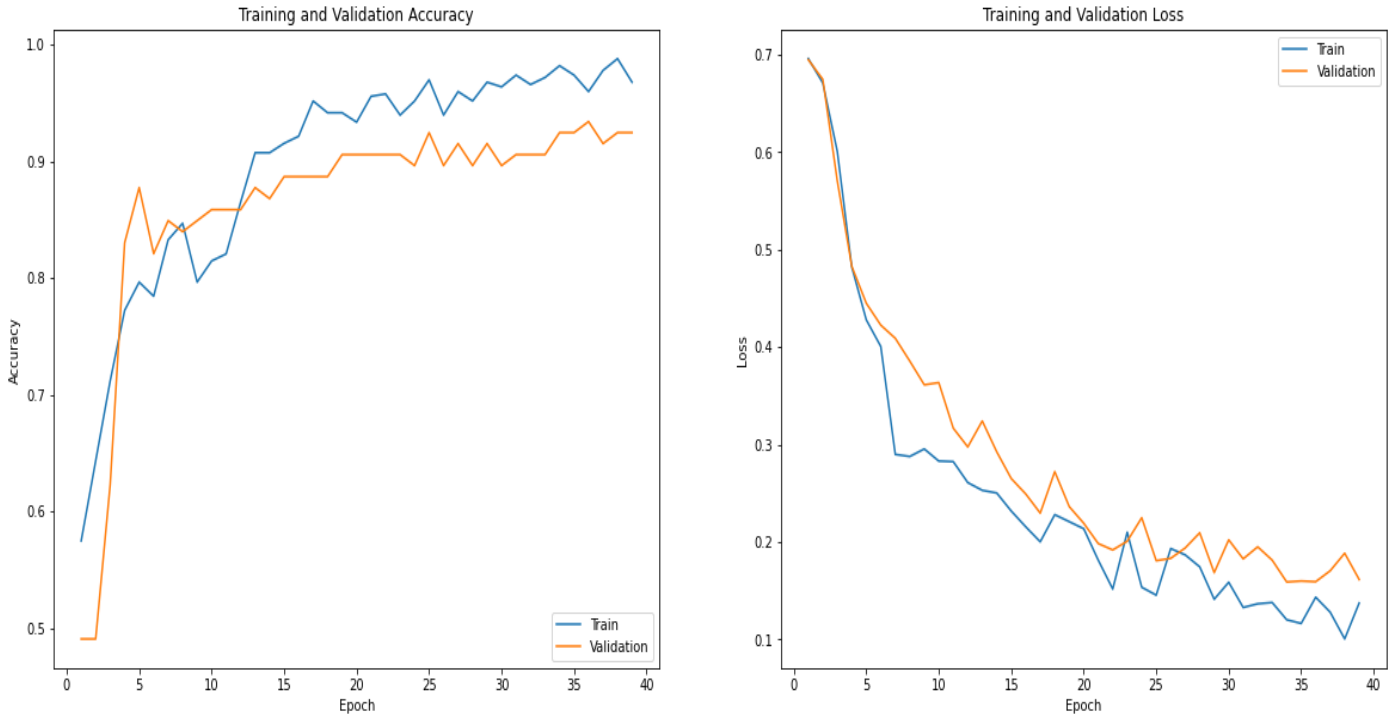

Figure 4: Accuracy and Loss for Validation and Training Sets for Dataset 1

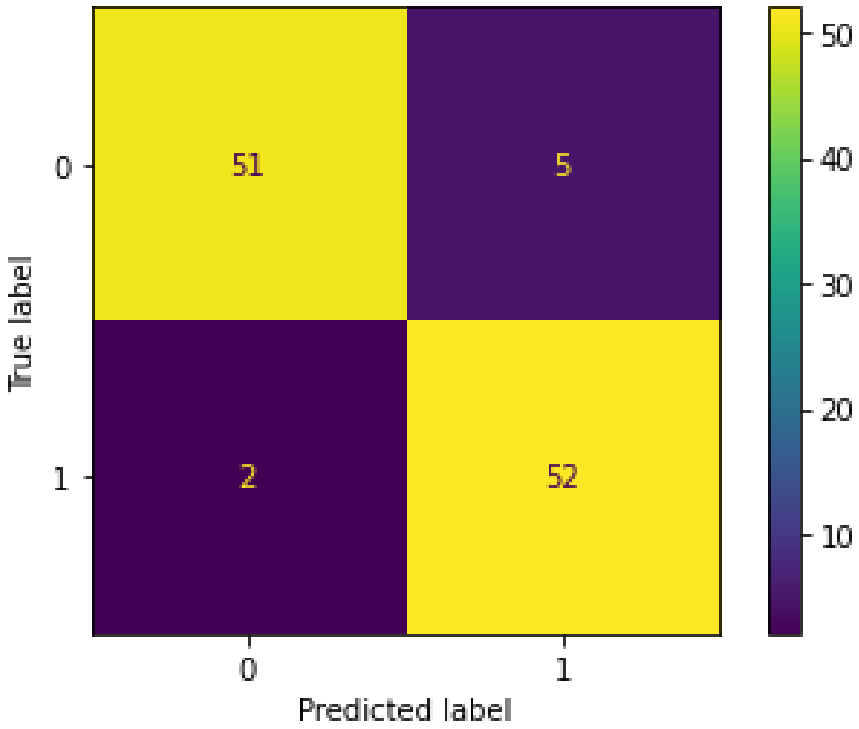

Figure 5: Confusion Matrix for Prediction during training for Dataset 1

5.2 Dataset 2 Results

Dataset 2 had a total of 224 images that were captured under different conditions from what the model trained on.
The model got an accuracy of 48.7894% on the dataset which is poor and shows the model may not perform well for data captured under different conditions as observed from the classification report. A total of 117 images from both classes is misclassified. With a precision of 48.78%, the model is not able to clearly distinguish positive and sick classes well as observed from the confusion matrix where there are 105 false positives detected, as observed from the confusion matrix of figure 6. With a recall of 89.29%, the model is easily able to predict positive classes correctly as observed from the confusion matrix where only 12 positive clases were misclassified(false negatives).

Classification Report for Dataset2

	precision	recall	f1-score	support
0	0.37	0.06	0.11	112
1	0.49	0.89	0.63	112
accuracy			0.48	224
macro avg	0.43	0.48	0.37	224
weighted avg	0.43	0.48	0.37	224

```
----------------------------------------------------------------
----------
Recall on Test Dataset2:  0.8929
Specificity on Dataset2:  0.0625
Accuracy on Dataset2:  0.4777
Precision on Dataset2:  0.4878
F1 Score Dataset2:  0.6309
----------------------------------------------------------------
----------
```

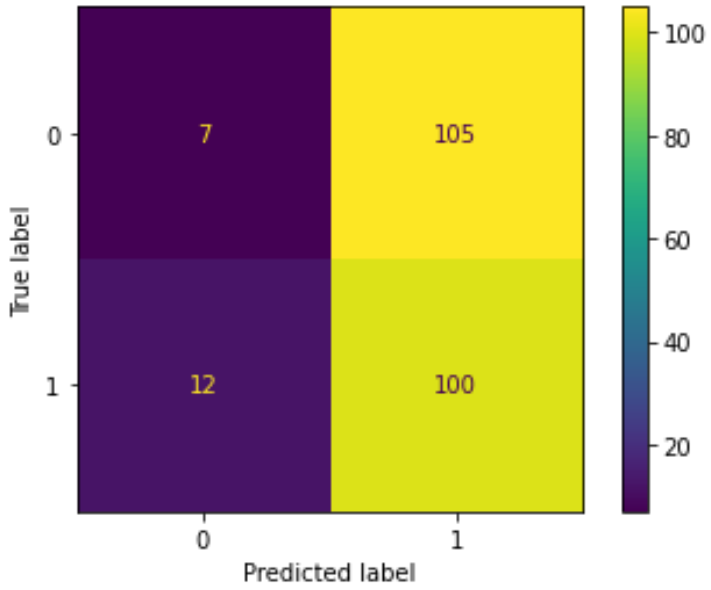

Figure 6: Confusion Matrix for Dataset 2

For dataset two, changing the batch size between 2 to 256 did not yield a huge difference for classification. Change of the learning rate and even augmentations(zooming, cropping, brightness, shearing) used did not yield a change in the results. The biggest challenge is found to be on the prediction of the normal class, which did not go above 17 images for the different methods attempted. Attempting to bias the model towards a better prediction of the normal class instead causes all the sick images to be misclassified as false negatives.

Some other limitations to training were the Graphical Processor Units(GPUs) from google colab that had time out sessions after a period of use. Therefore, computational resources remain a challenge.
There is a need to have a better understanding of the different techniques or conditions under which images for chest opacities are retrieved and what that means for differences in images attributes.

6. Conclusion

Zech et. al in [5] observed that CNNS trained on pneumonia achieved better performance on internal data than external data where their performance was relatively poor. Therefore the challenge faced with dataset two is expected, although we believe there are ways to improve on this dataset but the technique lies in the augmentation techniques, number of layers and loss function to be used.

REFERENCES

[1] M. Z. Alom *et al.*, “A State-of-the-Art Survey on Deep Learning Theory and Architectures,” *Electronics*, vol. 8, no. 3, Art. no. 3, Mar. 2019, doi: 10.3390/electronics8030292.

[2] A. Krishnaswamy Rangarajan and R. Purushothaman, “Disease Classification in Eggplant Using Pre-trained VGG16 and MSVM,” *Sci. Rep.*, vol. 10, no. 1, Art. no. 1, Feb. 2020, doi: 10.1038/s41598-020-59108-x.

[3] Y. Sim *et al.*, “Deep Convolutional Neural Network–based Software Improves Radiologist Detection of Malignant Lung Nodules on Chest Radiographs,” *Radiology*, vol. 294, no. 1, pp. 199–209, Jan. 2020, doi: 10.1148/radiol.2019182465.

[4] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” p. 31.

[5] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, “Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study,” *PLOS Med.*, vol. 15, no. 11, p. e1002683, Nov. 2018, doi: 10.1371/journal.pmed.1002683.