The Seybold Report

ISSN 1533-9211

PNEUMONIA DIAGNOSIS USING DEEP LEARNING APPROACH

[1]MARK JEDIDAIAH RAJ NELSON RAJ, [2]ABD SAMAD HASAN BASARI, [3]NUZULHA KHILWANI IBRAHIM, [4]NOORAYISAHBE MOHD YAACOB, [5]MOHAMED DOHEIR

[1, 3,4,5] Center for Advanced Computing Technology(C-ACT),

Fakulti Teknologi Maklumat dan Komunikasi

Universiti Teknikal Malaysia Melaka,

76100 Durian Tunggal, Melaka, Malaysia.

[2]ABD SAMAD HASAN BASARI,

[2]Faculty of Computer Science and Information Technology,

Universiti Tun Hussein Onn Malaysia (UTHM)

P.O. Box 101, 86400 Parit Raja, Batu Pahat, Johor Darul Takzim, Malaysia

DOI 10.5281/zenodo.6553624

ABSTRACT

Pneumonia has been one of the most popular disease affecting the lungs in Malaysia. Many people have lost their lives to this disease in Malaysia. Currently, experts and doctors use experience and knowledge to interpret X-Ray of patients to detect presence of pneumonia. This method is obviously not fool-proof. The risks that can occur with this current method is misdiagnosis due to human error. Lack of experts to handle the cases around the world has also become a growing issue. Liabilities for doctors and hospitals can be high in case of misdiagnosis and it can be terribly damaging and life threatening for patients. The purpose of this project is to design a deep learning model which can detect pneumonia with high accuracy. There are two types of pneumonia which are common, namely bacterial and viral. Our model should be able to detect the specific types of pneumonia if present, and if not, must be able to detect a healthy X-Ray. This can leverage the advantage of technology to help the medical and healthcare sector, as well as improving human lives. The methodology used is agile methodology to take as-fast-as-possible approach for this project. The data used is taken from Kaggle for proper structured datasets. Our aim is to ensure the model learns the patterns and generalizes its learning compared to memorizing the patterns from training data. The models will be evaluated using a validation set and validation accuracy will be the measure for performance of models. The results of

The Seybold Report
ISSN 1533-9211

this project should be able to detect presence of pneumonia in a given X-Ray image using deep learning with high accuracy.

Keywords: Five Keywords are Required Separated By Commas (Capitalize Each Work Italic)

I. INTRODUCTION

AI is becoming a buzz word in many topics nowadays. And it is changing many sectors and businesses. One major sector which is affected by this phenomenon is the medical field. Medical field has been depending on many methods which has been used for decades. It can be improved in many ways through technology such as software engineering, artificial intelligence and IoT.

Doctors depend on experience and knowledge to give diagnosis generally. Because of this, there is chances of human error and misdiagnosis. There are many times when doctors give wrong diagnosis for different diseases because of common symptoms. This can lead into many complications for patient's lives and liabilities for doctors and hospitals.

Humans also have tendency to forget some information over time. This is crucial as this can affect judgement of doctors in given situations. But this can be prevented by deep learning approach as it becomes even more accurate with more data and it will never forget what it learns. In this project, we will focus on a particular common lung disease which has been affecting millions of people around the world, which is pneumonia. The objective is to develop a pneumonia diagnosis solution model using deep learning approach which has a decent reliable accuracy.

Neural networks is needed to implement our solution as we intend to simulate the learning process of the human brain to identify generalized patterns in pneumonia based datasets. This requires accurate image processing and pattern recognition to perform well.

II. LITERATURE REVIEW & METHODS

Currently, there is no existing systems such as this, and there are no competing systems. Pneumonia has been diagnosed using human expert interpretation, experience and consultation. This led to our motivation to use technology to aid in pneumonia diagnosis.

The data is preprocessed first before training. The folder filepath and basic preprocessing is performed first. Then the images are normalized and set to grayscale. The images are also resized to 333*250 and single channel is used.

The batch size is defined as 32 in our case due to computational power consumption and memory allocation.

The Seybold Report
ISSN 1533-9211

Next, the shuffle was set to “True”. This is done to encourage shuffling of data within dataset. By doing this, we can prevent memorizing and promote generalization in learning. It also causes lesser chances of overfit and prevent a certain bias in learning.

A CNN was built to perform learning on dataset to extract features from each data. This means the CNN performs the image processing technique to learn from given existing data. This is essential as to implement any AI or deep learning technique, data is one of the important keys to better performance. The CNN was built from scratch to customize the learning process and parameters completely.

Tensorflow was used with Keras to build and implement this CNN model. Kernel regularizer was also implemented to increase accuracy of overall model. There are various methods to implement penalties on layers. Regularizer allows us to apply penalties on layer parameters or layer activity during optimization. These penalties are summed into the loss function that the network optimizes.

For our case, we used the L2 class, which applies the L2 regularization penalty. The loss function of L2 is “Loss= L2 * reduce_sum (square(x))”. The value of L2 is randomly given by us. In our case, we set L2= 0.02. Here is the summary of our model.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 329, 246, 32)	832
activation_1 (Activation)	(None, 329, 246, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 109, 82, 32)	0
conv2d_2 (Conv2D)	(None, 106, 79, 128)	65664
activation_2 (Activation)	(None, 106, 79, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 53, 39, 128)	0
conv2d_3 (Conv2D)	(None, 51, 37, 512)	590336
activation_3 (Activation)	(None, 51, 37, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 25, 18, 512)	0
conv2d_4 (Conv2D)	(None, 23, 16, 512)	2359808
activation_4 (Activation)	(None, 23, 16, 512)	0
max_pooling2d_4 (MaxPooling2	(None, 11, 8, 512)	0
conv2d_5 (Conv2D)	(None, 9, 6, 512)	2359808
activation_5 (Activation)	(None, 9, 6, 512)	0
max_pooling2d_5 (MaxPooling2	(None, 4, 3, 512)	0
conv2d_6 (Conv2D)	(None, 3, 2, 512)	1049088
activation_6 (Activation)	(None, 3, 2, 512)	0
max_pooling2d_6 (MaxPooling2	(None, 1, 1, 512)	0
flatten_1 (Flatten)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
activation_7 (Activation)	(None, 512)	0
dense_2 (Dense)	(None, 3)	1539
activation_8 (Activation)	(None, 3)	0

Total params: 6,689,731
Trainable params: 6,689,731
Non-trainable params: 0

Figure 1 : Summary of CNN Model

As shown above in Figure 1, there are 2 dense layers in our model. The regularizer is set in the first dense layer only. From Figure 1, we can also see how we built our CNN

The Seybold Report
ISSN 1533-9211

model layer by layer. No elaboration is needed as it is clearly shown in the summary. In short, Conv2D, activation, max pooling, flatten, dense and softmax layers were used. Softmax was used because we are expecting 3 outputs and softmax is most suitable for it. The type of activation layer used was ReLU.

An optimizer is then defined to help optimize our results. We used the Adam optimizer for our case. The learning rate was set at 0.00001 and every other values were set at default. The type of loss used is “categorical_crossentropy” and metrics is “accuracy”.

Next, we used a method called checkpoint defining. We did this to save best weights of each epochs when there is increase in monitored metric value. Usual practice will be to monitor one particular metric value. But in our case, we monitored 3 out of 4 metric values. The 3 are validation accuracy, accuracy and validation loss. The reasoning behind this move is to record overall better performance. There are situations where Epoch 50 and Epoch 51 may have similar validation accuracy values, but validation loss may improve. In such case, if usual method of monitoring one value is implemented, the weights of Epoch 51 will not be recorded if monitored metric was set as “validation accuracy”. This situation is avoided by monitoring few values simultaneously to capture weights if there is improvement in any monitored metric values. A filepath is set to save the weights of the respective epochs and can be accessed after the training is done. We set save best weights only so new weights will be save only if there is improvement on monitored metrics.

Training process is carried out after that. The results are completely saved in a CSV file after training. Once saved, they can be manually checked also to see values of monitored metrics from each epoch. Then, the resulting weights and model are saved respectively in “h5” and “json” format.

III. RESULTS & DISCUSSION

In this section, we will discuss the results obtained.

Table 1 : Test Images for Each Classes

Class	Number of images
Bacteria	148
Normal	148
Virus	148

The Seybold Report
ISSN 1533-9211

As shown in Table 1, 148 images were used in each class as test images. It was about 11% of training images for each classes.

```
Epoch 1/100
 - 51s - loss: 11.1362 - accuracy: 0.3655 - val_loss: 10.9097 - val_accuracy:
0.4234
Epoch 2/100
 - 45s - loss: 10.6532 - accuracy: 0.4811 - val_loss: 10.4046 - val_accuracy:
0.5225
Epoch 3/100
 - 45s - loss: 10.0610 - accuracy: 0.6075 - val_loss: 9.9072 - val_accuracy:
0.5090
Epoch 4/100
 - 46s - loss: 9.5684 - accuracy: 0.6451 - val_loss: 9.3032 - val_accuracy:
0.6554
Epoch 5/100
 - 45s - loss: 9.1614 - accuracy: 0.6975 - val_loss: 9.1552 - val_accuracy:
0.6532
Epoch 6/100
 - 45s - loss: 8.7965 - accuracy: 0.7045 - val_loss: 8.8546 - val_accuracy:
0.6599
```

Figure 2 : Example Of Results Of Each Epoch During Training

As shown in Figure 2, the results of each epoch is shown one epoch after another. Note that there are 4 metrics value being shown in each training epoch. As the epoch number is increasing, there is improvement in validation accuracy and validation loss.

	val_accuracy	accuracy	val_loss	loss
0	0.423423	0.365460	10.909685	11.136874
1	0.522523	0.481059	10.404598	10.651285
2	0.509009	0.607521	9.907244	10.060627
3	0.655405	0.645125	9.303210	9.566424
4	0.653153	0.697493	9.155241	9.162077
..	...	...	...	...
95	0.759009	0.942340	1.854003	0.990906
96	0.756757	0.933426	1.493124	0.990839
97	0.738739	0.938440	3.405276	0.976314
98	0.810811	0.937604	1.151235	0.969246
99	0.813063	0.936212	1.611011	0.961049

Figure 3 : Summary Of First Few Epochs & Last Few Epochs Evolution

The Seybold Report
ISSN 1533-9211

Figure 3 shows the results of the first few epochs and last few epochs which obviously has shown improvement in all categories. Note that in any machine learning or deep learning model, there will never be a fixed method or hard evidence in a particular way, but there will be higher probability towards a

particular trend and that is what we are always aiming to achieve in any given situation or target. There is a clear increase in validation accuracy from first epoch to final epoch whereas there is a steady decrease in validation loss from first epoch to final epoch. The validation accuracy shows the percentage of accuracy of the model towards unseen and unknown data whereas the accuracy shows the training accuracy of the model. On the other hand, the validation loss denotes the score of wrong predictions made by model and loss denotes score of mistakes made during training prediction. The smaller the score, the lesser the mistakes made. Note that value of validation loss and loss are not in percentage values. The models were evaluated using the validation accuracy and validation loss. By using the checkpoint method, we could extract any of the earlier epoch's weights if values were desirable for us.

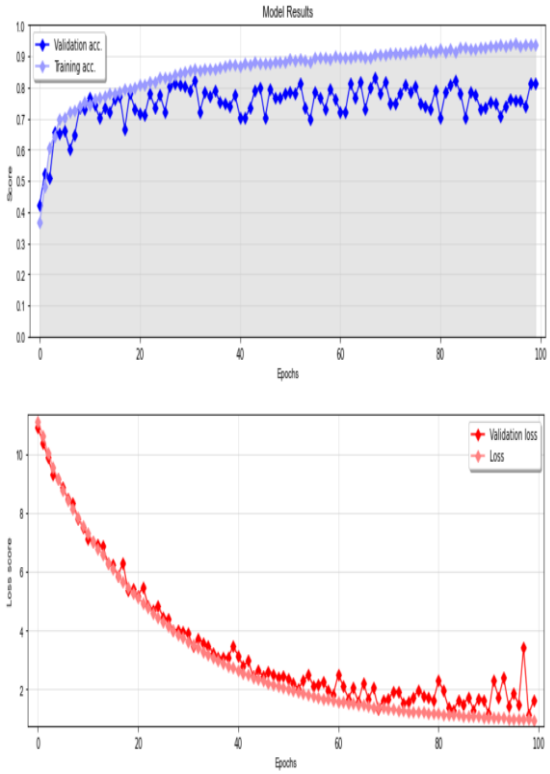

Figure 4 : Graph Of 100 Epochs Model

The Seybold Report
ISSN 1533-9211

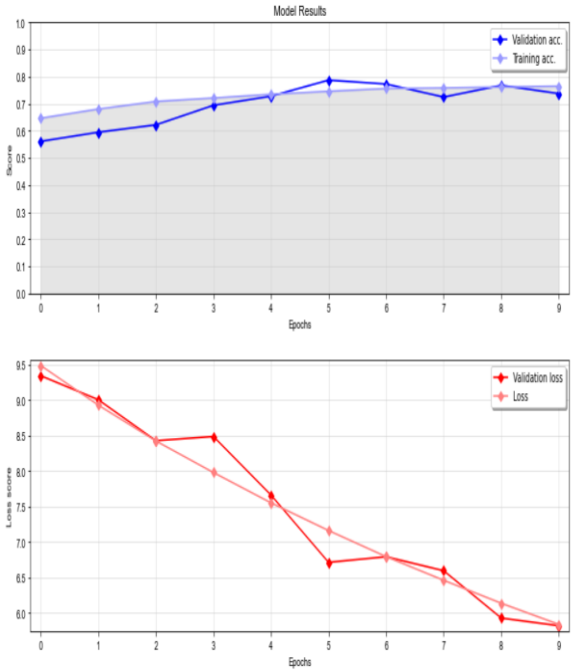

Figure 5 : Graph Of 10 Epochs Model

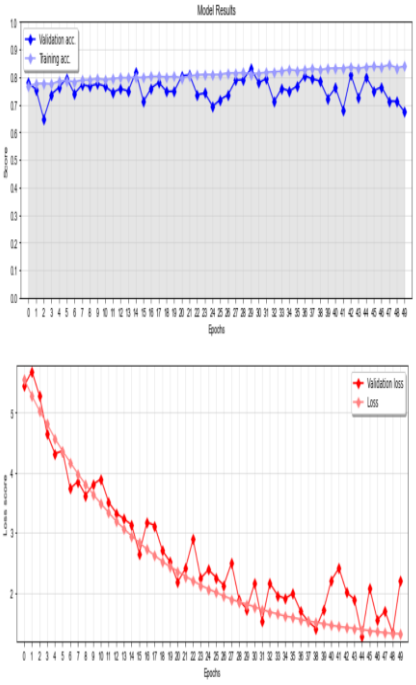

Figure 6 : Graph Of 50 Epochs Model

Figure 4, 5, and 6 shows graphs of different training executions using different epoch numbers respectively. The results can differ with different epoch numbers although

The Seybold Report
ISSN 1533-9211

other values are not adjusted. This proves the theory that there is no perfect or standard output in machine learning or deep learning as they are not hard coded.

Various changes were introduced into our model such as learning rate values, data amount, number of epoch, regularizer value, number of layers and many more. But eventually, the most satisfactory results were gained with the values explained in this paper. Our best results were finally obtained with 100 epoch set.

We noticed the validation accuracy and validation loss were steadily increasing and decreasing respectively. And using the checkpoint method, we can take the weights from earlier epoch before the pattern is memorized in case we feel necessary after viewing results.

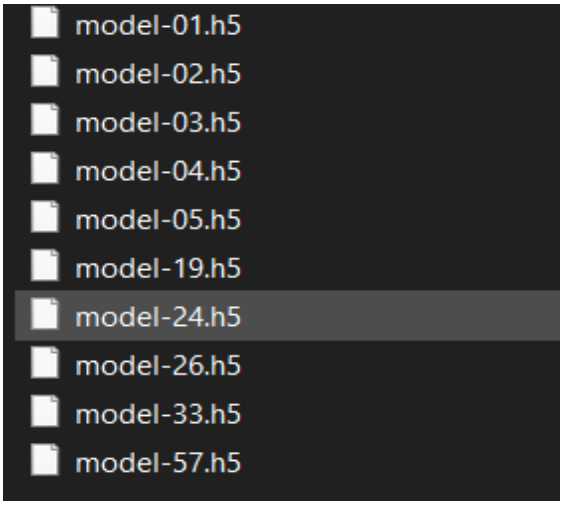

Figure 7 : Weights Stored Using Checkpoint Method

In some cases, early stopping is a popular method. But sometimes after several epochs there may be some good improvement in trend and learning. To capture this we use the checkpoint method.

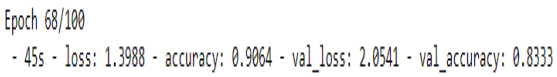

Figure 8 : Results of Epoch 68

As shown in Figure 8, our eventual choice of weights was Epoch 68 which achieved validation accuracy of 0.8333 and validation loss of 2.0541.

The Seybold Report
ISSN 1533-9211

IV. CONCLUSION

In summary, we were able to create a reliable model which can detect presence of pneumonia in given dataset with a high accuracy as planned. We tried multiple CNN designs to come up with better results, and the best architecture is what we have shown in this paper. The best validation accuracy we could achieve was 83% which is really good for a newly built architecture from scratch. The variation introduced in Adam optimizer usage in our implementation also is key in getting this results. The checkpoint method too helped in retrieving good results without stopping complete execution of training process. By using kernel regularizer, we were able to introduce penalties and better results overall. Lower validation loss achieved shows good results too.

Acknowledgement

This paper is part of research work under the Fundamental Research Grant Scheme (FRGS) number FRGS/2018/FTMK-CACT/F00395. The research is conducted in the Fakulti Teknologi Maklumat Dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM).

REFRENCES:

[1] What is an artificial neural network? Here's everything you need to know By Luke Dormehl, 6 January 2019, accessed 26 February 2020, <https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/>

[2] What is Machine Learning? A definition, Blog, Machine learning 7 March 2017, accessed 26 February 2020, <https://expertsystem.com/ machine-learning-definition/>

[3] Introduction to Artificial Neural Networks (ANN) By Nagesh Singh Chauhan, 3 Oct 2019, accessed 27 February 2020, <https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9>

[4] Albawi, Saad & Abed Mohammed, Tareq & ALZAWI, Saad. (2017). Understanding of a Convolutional Neural Network, accessed 2 March 2020. <10.1109/ICEngTechnol.2017. 8308186.>

[5] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way by Sumit Saha, 16 Dec 2018, accessed 3 March 2020, <https://towardsdatascience.com/a- comprehensive-guide-to-convolutional-neural-networks-the-eli5-way- 3bd2b1164a53>

[6] Convolutional neural networks Jianxin Wu, LAMDA Group, National Key Lab for Novel Software Technology, Nanjing University, China, accessed 11 April 2020. <https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf >

[7] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, accessed 18 March 2020, <arXiv 1409.1556>

The Seybold Report
ISSN 1533-9211

[8] VGG-16 | CNN model by pawangfg, accessed 18 March 2020, <https://www.geeksforgeeks.org/ vgg-16-cnn-model/>

[9] Using a pre-trained convent by jjallaire, accessed 19 March 2020, <https://jjallaire.github.io/deep-learning-with-r-notebooks/notebooks/5.3-using-a-pretrained-convnet.nb.html.>

[10] Transfer learning with a pretrained ConvNet by Tensorflow, accessed 19 March 2020, <https://www.tensorflow.org/tutorials/images/transfer_learning#freeze_the_convolutional_base>

[11] Activation Functions : Sigmoid, ReLU, Leaky ReLU and Softmax basics for Neural Networks and Deep Learning by Himanshu Sharma, 19 Jan 2019, accessed 21 March 2020, <https://medium.com/@himanshuxd/activation-functions-sigmoid-relu-leaky-relu-and-softmax-basics-for-neural-networks-and-deep-8d9c70eed91e>

[12] Dying ReLU: Causes and Solutions (Leaky ReLU) by Naresh Kumar 6 June 2019, accessed 21 March 2020, <http://theprofessionalspoint.blogspot.com/2019/06/dying-relu-causes-and-solutions-leaky.html>

[13] B. Hanin. (2018). Which neural net architectures give rise to exploding and vanishing gradients? In Advances in Neural Information Processing Systems, pages 580–589.