ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 6, Issue 6, June 2021

Detection of Pneumonia from Digital X-ray Imaged Thru Transfer Learning

[1] Jerick Lee, [2] Donabel D. Abuan
[1]Top Review Center, Manila, Philippines
[2] Department of Elecronics and Communications Engineering, Gokongwei College of Engineerng
Manila, Philippines

Abstract: - **The COVID-19 pandemic placed healthcare systems of every country under immense pressure. Diagnostics and treatment are pushed to its limits as medical frontliners struggle to manage the huge influx of incoming patients with respiratory symptoms. This study aims to assist diagnostics through pre-assessment of X-ray images to detect signals or features that strongly correlates to pneumonia. Specifically, we will train classification neural network on top of various pre-trained Deep Classification Models through existing X-Ray images with and without pneumonia. These models include VGG16, InceptionResNetV2, and MobileNetV2. To test the detection accuracy of each trained model, 25% of the training data will be separated, and will be evaluated after the model has been trained with the remaining images. All images in the dataset are pre-classified, and we will be able to generate accuracy metrics from the evaluation.**

Keywords— Deep Learning; Image Processing; Imaging; Diagnostics.

Introduction

The COVID-19 pandemic is pressuring national healthcare systems throughout the world in unprecedented scales. The flow of incoming symptomatic patients has been steadily increasing and will remain so for the next few years. Fast diagnostics are of utmost importance for decision making especially with limited medical resources such as isolation rooms and ventilators. This study aims to assist in pre-assessment of the presence of pneumonia from X-Ray images to give doctors a tool for managing which patients should receive priority care.

In this paper the researchers used Python and Keras with Tensorflow backend. The base models were downloaded from the Keras Applications repository, without the default top layers. New fully connected layers with a softmax endpoint were added as the final layers, and trained over Google Colab. The script consists of hyperparameters that can be fine- tuned to observe any improvement of the loss minimization, resulting to higher validation accuracy.

Design and application

Methodology

Due to the COVID-19 virus, the researchers found it best not to gather images themselves from medical institutes. Kaggle is a community of data scientists sharing codes, models, and data, for the purpose of general advancement in the field of Artificial Intelligence. Labeled X-Ray images with pneumonia classification are available from the site, generously provided byPaulMooney[3]. The researchers used this as the training data.

Train on Google Colab

Google Colaboratory allows researchers to utilize Google's Graphics processing Units ,GPUs and Tensor processing Unit, TPUs to train deep learning models, especially for people who do not have access to computers with gaming grade graphics. GPUs are used for training because of its high performance and distributed computing capabilities, allowing for simultaneous batch computation of gradients, weight and biases, allowing for faster training times especially with models with complex architecture.

Setup Google Colab to Use GPU

By default, new Google Colab scripts are assigned only a CPU. In order to utilize GPU for deep learning training:

- Click Runtime > Change Runtime Type
- Under Hardware Accelerator, select GPU or TPU

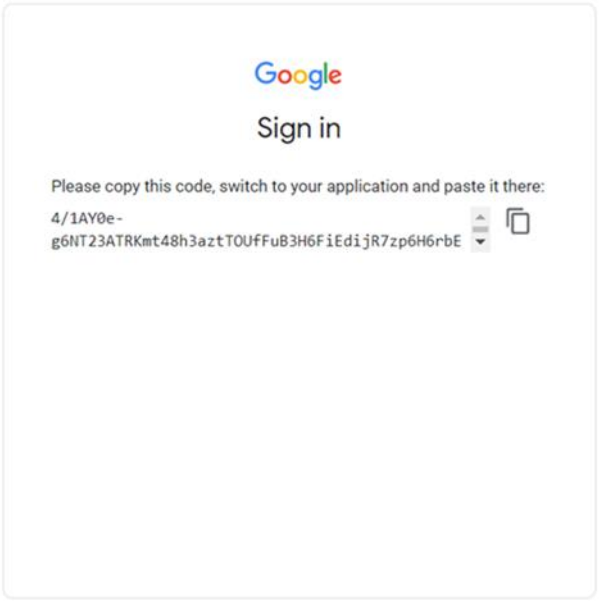
The GPU and TPU options should have the same performance. According to the Google, TPUs are just GPUs that are dedicated to serve Machine Learning purposes. In our study, we have selected GPU as our hardware accelerator.

Linking Google Drive to Colab Script

To use Google Colab, the script must be written on Jupyter notebook format. Files are accessed through Google Drive. To link a Colab script to a folder in Google Drive, the following code should be inserted:

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/<Folder Name>
```

After executing the cell that contains the code snippet above, Google will ask for an authorization code. A link will be provided where you need to login the Google Account of the drive you want to access. After successfully logging in, a code will be shown like the following snapshot:



Please copy this code, switch to your application and paste it there:

```
4/1AY0e-
g6NT23ATRKmt48h3aztTOUfFuB3H6FiEdijR7zp6H6rbE
```

Fig.1. Image: Authorization Code

Splitting the Dataset

The researchers set aside a test dataset to evaluate the trained model later. This will show an unbiased metric on how well the model can classify an XRay image with pneumonia. Standard test split varies from 20% to 30%. The researchers used 25% as the split percentage.

Downloading Models from Keras Repository

The Keras website shows its available pre-trained classifier models in their repository (https://keras.io/api/applications/). The researchers picked 3 models based on model sizes. They have settled with the following models:

- VGG16 - (528 MB)
- InceptionResNetV2 - (215 MB)
- MobileNetV2 - (14 MB)

Keras allows downloads and caching straight from the script.

Modifying the Models

The deep models come with their own classifier layers that were trained to recognize images from its training source, which is commonly ImageNet. These fully connected layers are replaced and trained over the X-Ray dataset.

The base model layers are frozen so that the weights and biases of their neurons remain unchanged while training. This is essential to retain the feature extraction capabilities of the deep model. Only the appended classification layers are unfrozen for classification purposes.

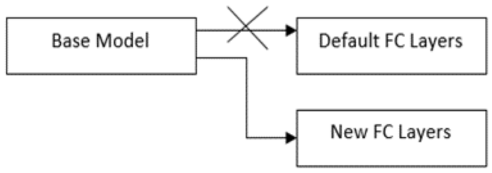


Fig.2. Appending New Fully Connected Layers for Classifying Pneumonia

Adjusting Hyper Parameters

With the script in place, initial hyper parameters are set, and adjusted every end of training in the attempt to improve the validation loss without overfitting. These hyper parameters include:

- Layers To Unfreeze – number of layers from the base model to unfreeze if required
- Neurons – number of neurons in each fully connected layer before the softmax layer
- Learning Rate – the learning rate while training
- Max Epochs – an upper limit of how many epochs the training should run
- Batch Size – the number of samples to use for training at an iteration
- Fixed Step Size – number of batches to use for each epoch. If useAllSamples is true, the number of steps will equate to using all samples, depending on batchSize
- Use All Samples – whether to use all training samples.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 6, Issue 6, June 2021

• Patience – number of epochs to wait where the validation loss is not decreasing before stopping training

Train and Monitor Loss and Accuracy Curves

There will be two objectives to training the base model with a new classifier:

• Minimize the validation loss

• As a result of a small loss, ensure the validation accuracy is high (preferably > 90%)

• Keep the distance between the training loss and validation loss small to prevent overfitting

• Stop the training when the validation loss does not improve within a number of epochs

Test Model with the Test Dataset

The model's accuracy can be evaluated by testing all the samples in the test dataset. Because the samples are labeled, we can compute the overall accuracy by checking how many times the model classified the sample. For further inspection of metrics for each class, we can generate a Confusion Matrix and an Accuracy Metrics report. These can be provided by the Scikit Learn Metrics module.

Results and Discussion

```
Log-loss (cost function):
training   (min:    0.054, max:    0.773, cur:    0.133)
validation (min:    0.157, max:    0.724, cur:    0.161)

Accuracy:
training   (min:    0.594, max:    1.000, cur:    0.922)
validation (min:    0.711, max:    0.941, cur:    0.936)
8/8 [==============================] - 15s 2s/step - loss: 0.1329 - a
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
```

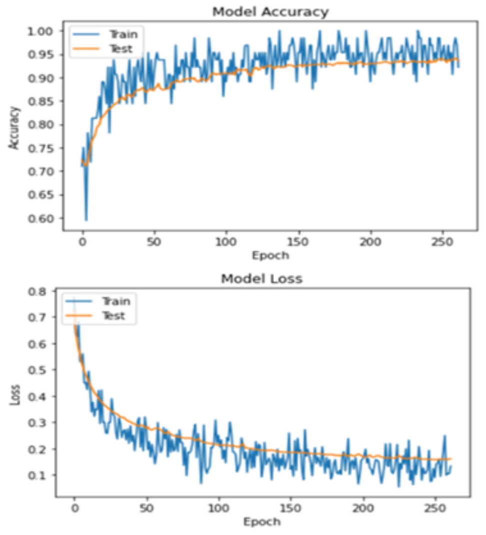
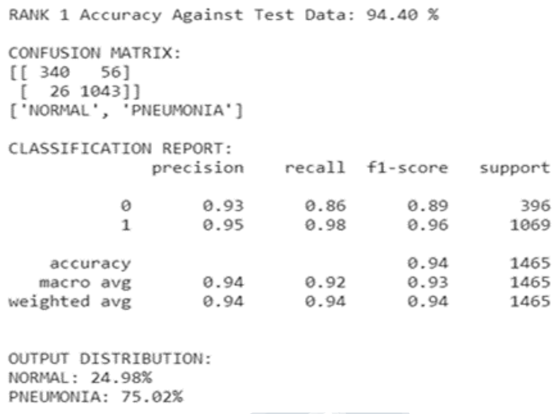
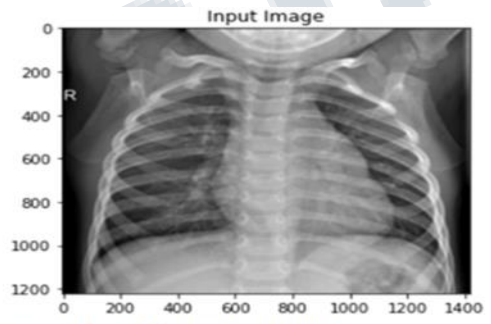


Fig.3. *Image: VGG16 Training Results*

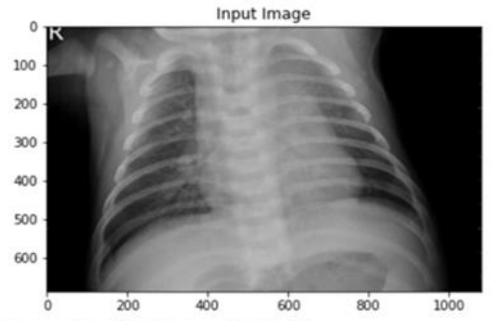


```
RANK 1 Accuracy Against Test Data: 94.40 %

CONFUSION MATRIX:
[[ 340   56]
 [  26 1043]]
['NORMAL', 'PNEUMONIA']

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.93      0.86      0.89       396
           1       0.95      0.98      0.96      1069

    accuracy                           0.94      1465
   macro avg       0.94      0.92      0.93      1465
weighted avg       0.94      0.94      0.94      1465


OUTPUT DISTRIBUTION:
NORMAL: 24.98%
PNEUMONIA: 75.02%
```

Fig.4. *VGG16 Confusion Matrix*

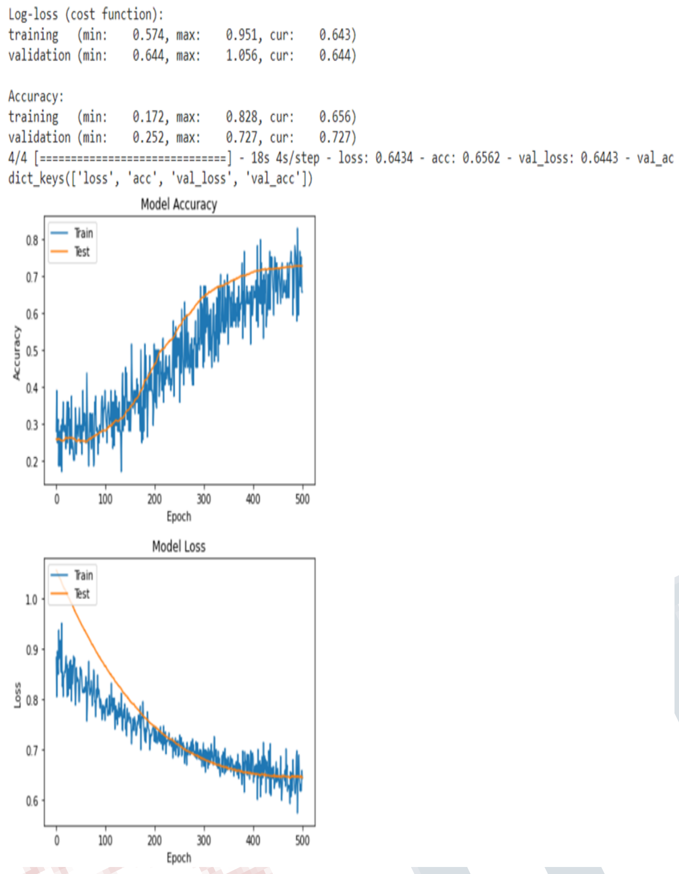


```
Image Classified as:  NORMAL
Actual Class: NORMAL
Classification Score:  0.96717215
```



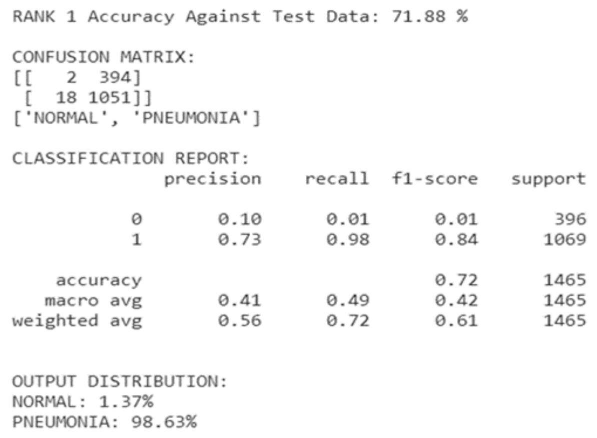
```
Image Classified as:  PNEUMONIA
Actual Class: PNEUMONIA
Classification Score:  0.9931704
```

Fig.5. *Classification on Different Classes using VGG16*



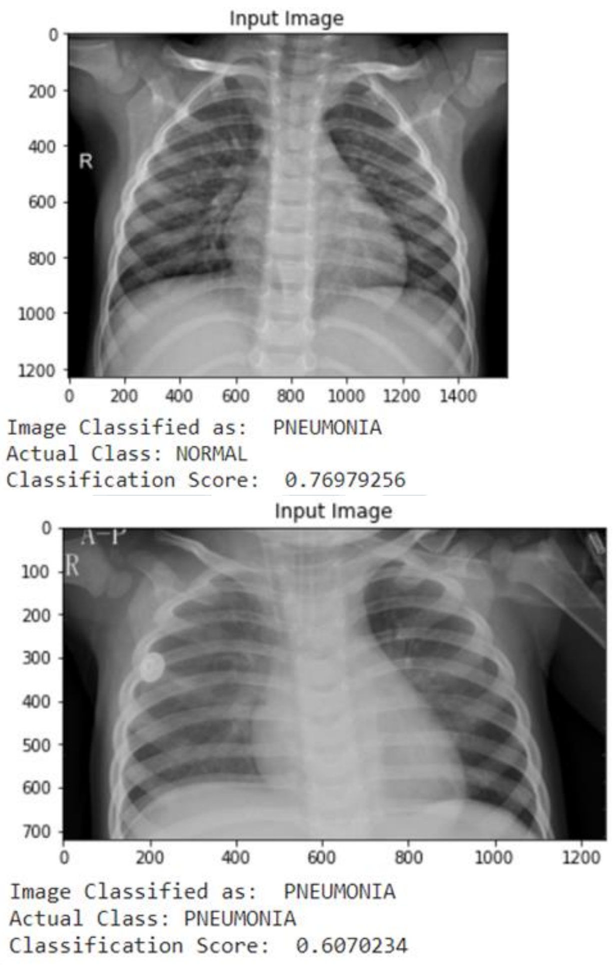
```
Log-loss (cost function):
training   (min:    0.574, max:    0.951, cur:    0.643)
validation (min:    0.644, max:    1.056, cur:    0.644)

Accuracy:
training   (min:    0.172, max:    0.828, cur:    0.656)
validation (min:    0.252, max:    0.727, cur:    0.727)
4/4 [==============================] - 18s 4s/step - loss: 0.6434 - acc: 0.6562 - val_loss: 0.6443 - val_ac
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
```

Fig.6. *InceptionResnetV3 Training Results*



```
RANK 1 Accuracy Against Test Data: 71.88 %

CONFUSION MATRIX:
[[    2  394]
 [   18 1051]]
['NORMAL', 'PNEUMONIA']

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.10      0.01      0.01       396
           1       0.73      0.98      0.84      1069

    accuracy                           0.72      1465
   macro avg       0.41      0.49      0.42      1465
weighted avg       0.56      0.72      0.61      1465


OUTPUT DISTRIBUTION:
NORMAL: 1.37%
PNEUMONIA: 98.63%
```

Fig.7. *InceptionResnetV3 Confusion Matrix*



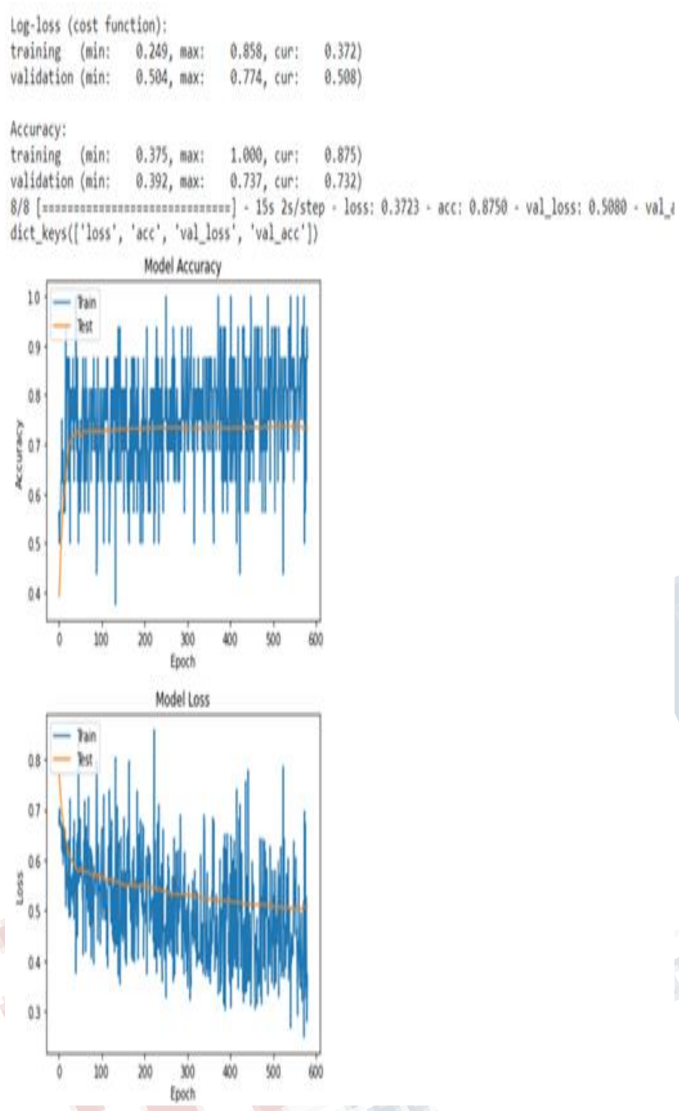
```
Image Classified as:  PNEUMONIA
Actual Class: NORMAL
Classification Score:  0.76979256
```

```
Image Classified as:  PNEUMONIA
Actual Class: PNEUMONIA
Classification Score:  0.6070234
```

Fig.8. *Evaluation on Different Classes InceptionResnetV3*

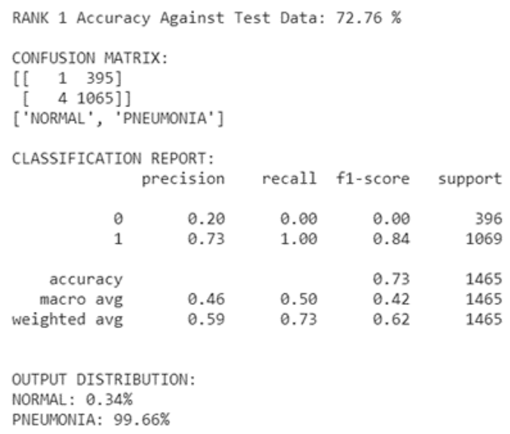
ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 6, Issue 6, June 2021



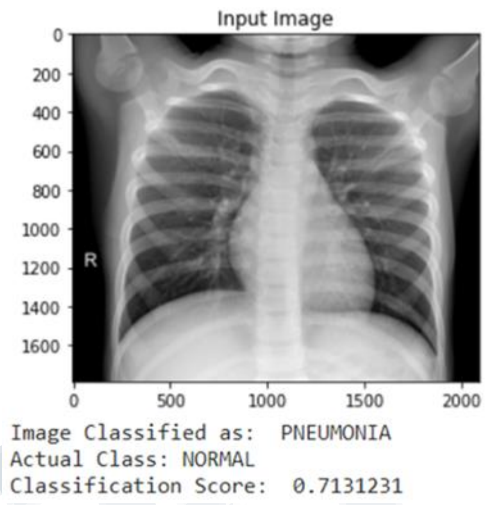
```
Log-loss (cost function):
training   (min:    0.249, max:    0.858, cur:    0.372)
validation (min:    0.504, max:    0.774, cur:    0.508)

Accuracy:
training   (min:    0.375, max:    1.000, cur:    0.875)
validation (min:    0.392, max:    0.737, cur:    0.732)
8/8 [==============================] - 15s 2s/step - loss: 0.3723 - acc: 0.8750 - val_loss: 0.5080 - val_a
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
```

Fig.9. *MobileNevV2 Training Results*

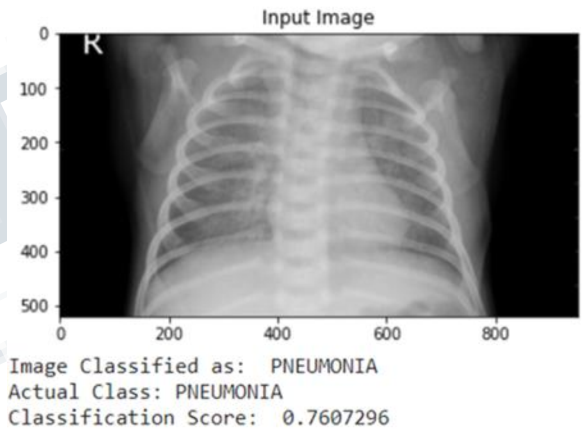


```
RANK 1 Accuracy Against Test Data: 72.76 %

CONFUSION MATRIX:
[[   1  395]
 [   4 1065]]
['NORMAL', 'PNEUMONIA']

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           0       0.20      0.00      0.00       396
           1       0.73      1.00      0.84      1069

    accuracy                           0.73      1465
   macro avg       0.46      0.50      0.42      1465
weighted avg       0.59      0.73      0.62      1465


OUTPUT DISTRIBUTION:
NORMAL: 0.34%
PNEUMONIA: 99.66%
```

Fig.10. *MobileNevV2 Confusion Matrix*



```
Image Classified as:  PNEUMONIA
Actual Class: NORMAL
Classification Score:  0.7131231
```



```
Image Classified as:  PNEUMONIA
Actual Class: PNEUMONIA
Classification Score:  0.7607296
```

Fig.11. *Evaluation on Different Classes using MobileNetV2*

Usually normal x-rays have clear defined outlines throughout the lung cavity, and the cardiac boundary region is well defined. Another identifier for a normal chest x-ray is the lower left and right pointed extremities of the lungs can be clearly seen. Pneumonia presents different artifacts and features within the x-ray images. The most common feature is cloudiness in certain regions or within the overall lung cavities. The cardiac region's boundary is blurred, indicating that the cavity contains fluid.

It can be seen that the VGG16 model works well in finding these features to contrast the normal x-rays against those with pneumonia. Through visual inspection, these features are very subtle and are difficult to differentiate with an untrained eye. It is due to this fact that the other two models did not performed well. Both InceptionResnetV3 and MobileNetV2 were unable to find any correlation between the presence of pneumonia and the respective images.

TABLE 1. SUMMARY OF ACCURACY WITH THEIR CORRESPONDING MODELS

Deep Model	**Macro Average Recall**	**Overall Accuracy**
VGG16	92 %	94.4 %
InceptionResnetV3	49 %	71.88 %
MobileNetV2	50 %	72.76 %

Conclusion

It was found that the different Deep Classifier models do not produce the same performance when used in applications beyond classification with significant visual differences. With the classification of the presence of pneumonia, the x-rays show only subtle differences where only a trained professional can correctly differentiate and analyze. These subtle differences are generally not observed by majority of Deep Learning classifiers as they are optimized to look for visually significant difference.

The VGG16 model however were able to localize on the subtle features and the classifier layer were able to converge generally well at over 92% macro accuracy. Newer models seem to give more importance to generalization and perform better on visually different classes with wider variation (to best classify the ImageNet repository).

With this finding it is encouraged to find similar applications where the differences between classes are minimal. Transfer learning does not require the feature extraction layers to be retrained on the application dataset,only the classification layers, making training cost effective and fast.

Acknowledgment

We would like to thank our family for being so supportive to us . Special acknowledgement to De La Salle University for accepting our paper to publish this and especially to our Lord Jesus who gave us all the wisdom, knowledge and strength to finish this paper completely.

References

[1] Aditi, R., Ioannis A., Amar, K.A., Dmitriy, F., Arquimedes, C., Kaushik, K., Tugba, K., (2020). Diag2graph: Representing Deep Learning Diagrams In Research Papers As Knowledge Graphs.IEEEXplore.DOI:10.1109/ICIP40778.2020.919 1234. 2020 IEEE International Conference on Image Processing (ICIP). Abu Dhabi, UAE.

[2] Ali Bou, I., Atinan, A., Mohammad, A., Khaled, S.,(2019, February 01). Speech Recognition Using Deep Neural Networks: A Systematic Review. IEEE Access. Volume: 7. 19143 – 19165. DOI: 10.1109/ACCESS.2019.2896880. IEEE.

[3] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[4] https://keras.io/api/applications.

[5] Jianyu, W., Zhenling , M., Heng, Z., Qiang, M., (2019, March 27). Advances in Prognostics and System Health Management. IEEE Access. Volume 7. 42373 – 42383. DOI: 10.1109/ACCESS.2019.2907131 IEEE.

[6] Justin, K., Lipo , W., Jai R.,Tchoyoson L., (2017, December 29). Soft Computing Techniques for Image Analysis in the Medical Industry Current trends, Challenges and Solutions. Volume 6. 9375 – 9389. DOI: 10.1109/ACCESS.2017.2788044. IEEE.

[7] Ziwei, Z., Peng, C., Wenwu, Z., (2020, March 17). IEEE Transactions on Knowledge and Data Engineering. DOI: 10.1109/TKDE.2020.2981333. IEEE.