# Pneumonia Patient Diagnosis Image Classification

Using image classification models to pneumonia diagnosis.

Alex Beat

# Business Overview

Problem: Develop a model to predict which patients are healthy and which have pneumonia based on x-ray images.

Business Value: Having an accurate classification model will improve the medical business in several aspects, from technological advantages, economics and patient health.

Normal


Pneumonia

# Method

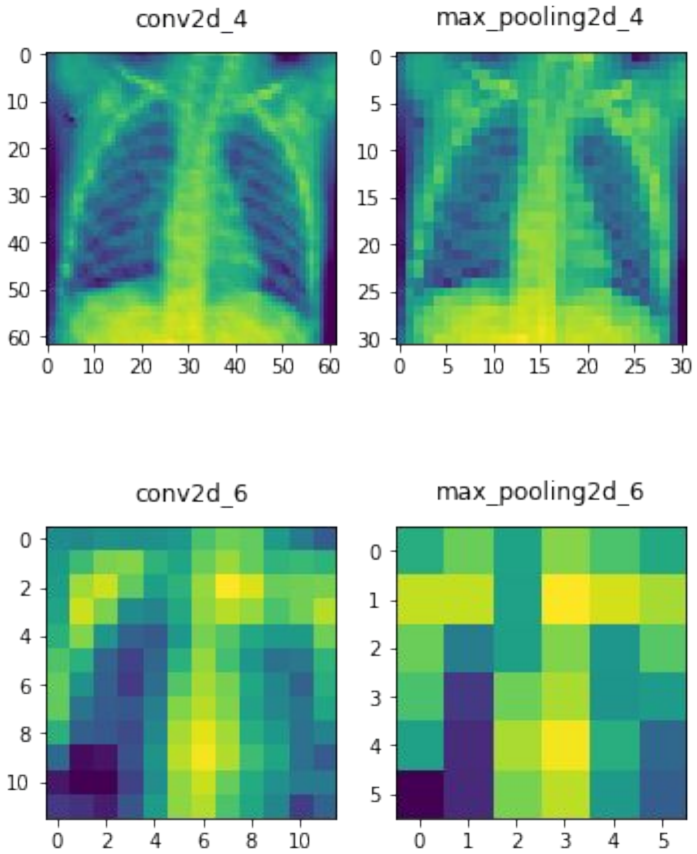
## Used a convolutional neural network (CNN) model:

Processed and trained a dataset of 4,700+ images.

Viewed feature maps that the model creates from pixels in each model layer.

The model looks for generic pixel patterns. Learns more complex patterns from those.

Trained multiple models, adjusting hyperparameters to find best performing model.




*Color added for example.

# Model adds technical advantages to aid in doctor review.

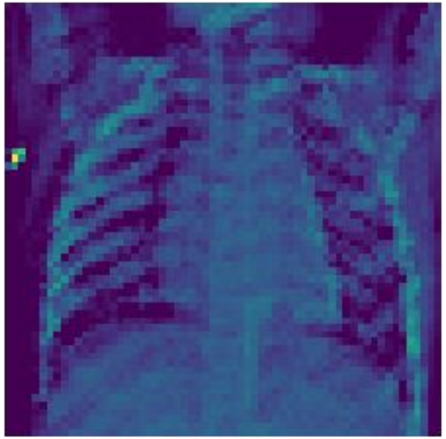
Model is cloud-based. Powerful GPU.

CNN - type of deep learning neural network. Faster than dense layer neural network used in deep learning.

Model generates the images at smaller resolution of only 64x64 px for faster processing.

It is able to make reliable and fast predictions based only on these small images.

Allows for more automation in the review of patient images.



# Best CNN Model.

**CNN Model Recall for Pneumonia: 97%**

Precision: 88%

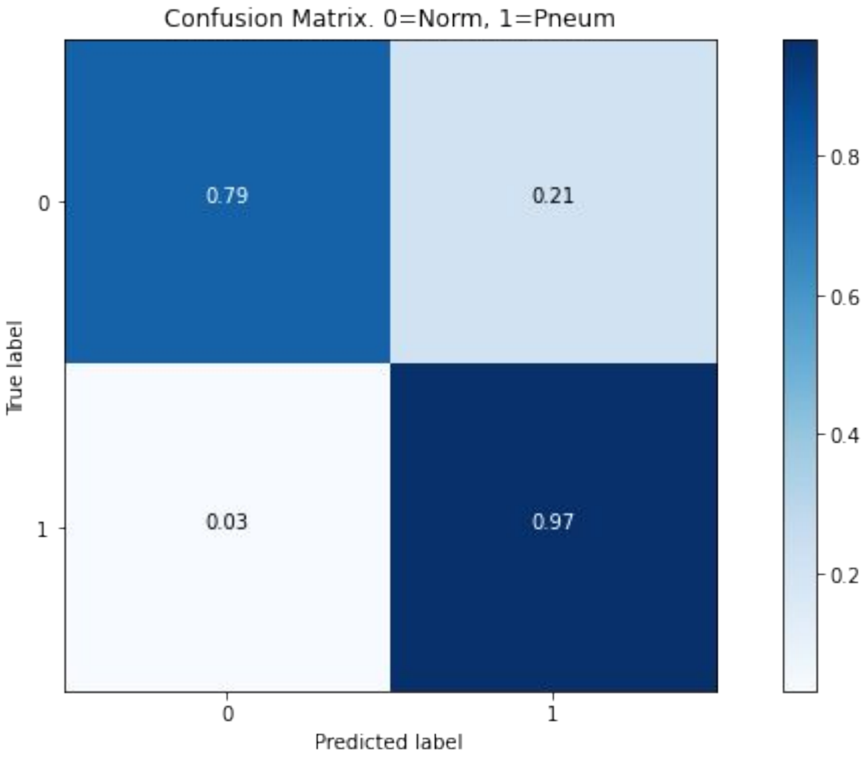
Overall accuracy 90%

- Recall is priority to capture all pneumonia cases.

6 CNN layers and 2 dense layers.

Rec. image size - 64x64 px

Training time is only 17 minutes for over 4700 patient x-ray images.




*Additional model reports in the appendix.

# Use machine learning to scan lab images before doctor's manual review.

Doctor/staff can run this model process before there review of patient images.

They will have reliable results to go off of, allowing for faster image review prior to their patient meeting.

Less time needed to manually scan each image. Less labor hours needed for this repetitive task.

Machine learning will help ease workload.

Redirect that saved time/hours to patient face-to-face. Allocate labor hours, for more efficient practice.

# Help prevent human error.

Using this model before the doctor's review of the image will help to prevent human error.

Will help to pinpoint patient cases that need critical attention.

Confidence in diagnosis and in doctor. Business reputation goes up.

Faster diagnosis. Less chance of disease spreading or getting worse. Better chance of recovery.

Less need for patient to seek out second opinion (less cost and time for them).

More patients are able to be treated more efficiently. Better overall patient health.

# Summary & Recommendations

1. Use this model for its technical advantages.
2. Process before doctor’s manual review.
3. Prevent human error.

# Future work

- Acquire more data for a better working model.
- Include more model tuning to achieve better accuracy and get model to perform with less overfitting - possibly run for more epochs and with a smaller patience level.
- Try larger image sizes such as 150x150px.
- Train model to help diagnose other diseases/injuries.

# Thank you.

# Appendix.

CNN model layers.

Each layer develops more patterns and more channel dimensions of the images, going from 32 up to 128 feature mapped dimensions.

Dense layers at the end connect all of the parameters as final learning step for output.

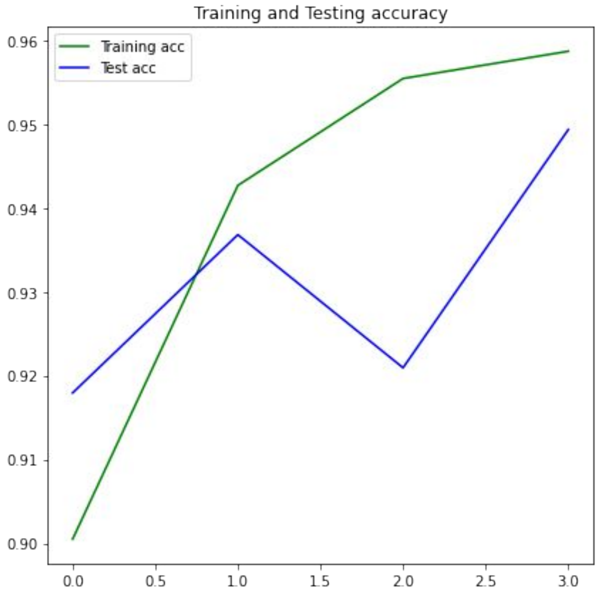
```
Model: "sequential_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_64 (Conv2D)           (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_64 (MaxPooling (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_65 (Conv2D)           (None, 29, 29, 64)        18496
_________________________________________________________________
max_pooling2d_65 (MaxPooling (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_66 (Conv2D)           (None, 12, 12, 128)       73856
_________________________________________________________________
max_pooling2d_66 (MaxPooling (None, 6, 6, 128)         0
_________________________________________________________________
flatten_22 (Flatten)         (None, 4608)              0
_________________________________________________________________
dense_43 (Dense)             (None, 512)               2359808
_________________________________________________________________
dense_44 (Dense)             (None, 1)                 513
=================================================================
Total params: 2,453,569
Trainable params: 2,453,569
Non-trainable params: 0
```

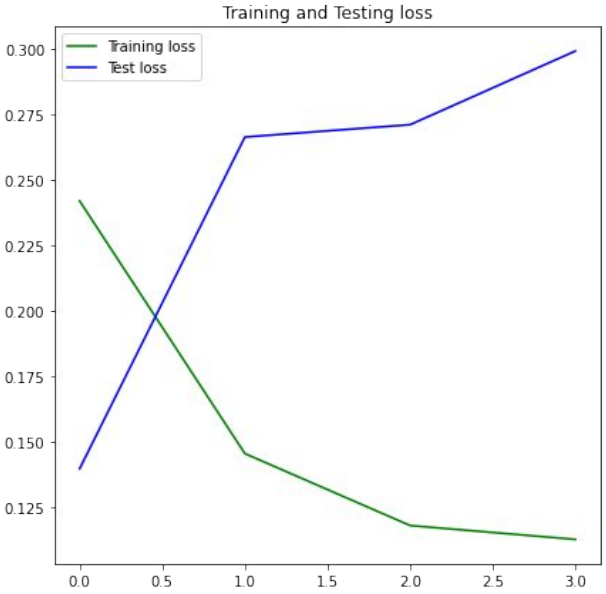
# Appendix.

Training and testing accuracy shown through model training process.

X-axis steps are known as epochs.

Loss shows how well the model is predicting during training.



# Appendix.

Epoch steps of training a model show the time for each step, loss, accuracy and compared validation data which the model separates before training.

```
None
Epoch 1/10
500/500 [==============================] - 259s 517ms/step - loss: 0.2420 - acc: 0.9005 - val_loss: 0.1398 - val_acc: 0.9179
Epoch 2/10
500/500 [==============================] - 258s 516ms/step - loss: 0.1456 - acc: 0.9427 - val_loss: 0.2664 - val_acc: 0.9368
Epoch 3/10
500/500 [==============================] - 257s 514ms/step - loss: 0.1181 - acc: 0.9554 - val_loss: 0.2711 - val_acc: 0.9209
Epoch 4/10
500/500 [==============================] - 256s 513ms/step - loss: 0.1128 - acc: 0.9587 - val_loss: 0.2993 - val_acc: 0.9493
Training took a total of 0:17:10.934737
```