# Technology in Medicine

By: Lauren Esser

# BACKGROUND

Pneumonia is a lung infection that can cause mild to severe illness in people of all ages.

Currently to diagnose pneumonia the doctor will ask about your medical history and may perform a blood test, chest x-ray, pulse oximetry, or sputum test.

# FACTS ABOUT PNEUMONIA

## VACCINE

- There is a vaccine to help fight against pneumonia, 68.9% of adults over the age of 65 have received the vaccine.

## HOSPITAL CARE

- More than 250,000 people have to seek hospital care each year due to pneumonia.

## TREATMENTS

- Current treatments are antibiotics, cough medicine, fever reducers, pain relievers, and potentially hospital care.
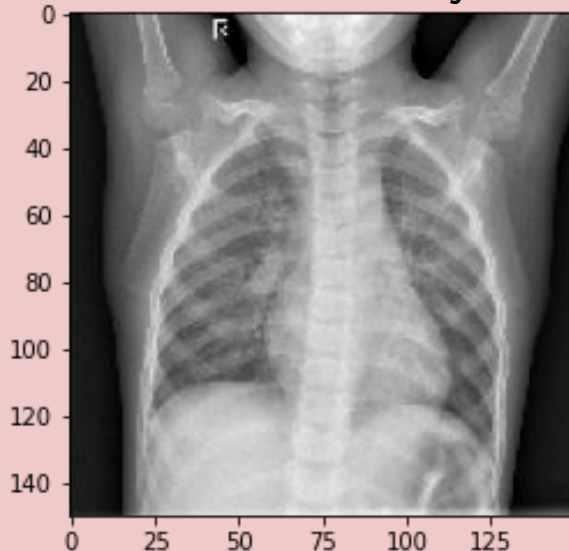
## DEATHS

- Pneumonia kills more children younger than 5 years old each year than any other infectious disease.
- About 50,000 people die from the disease each year in the U.S.
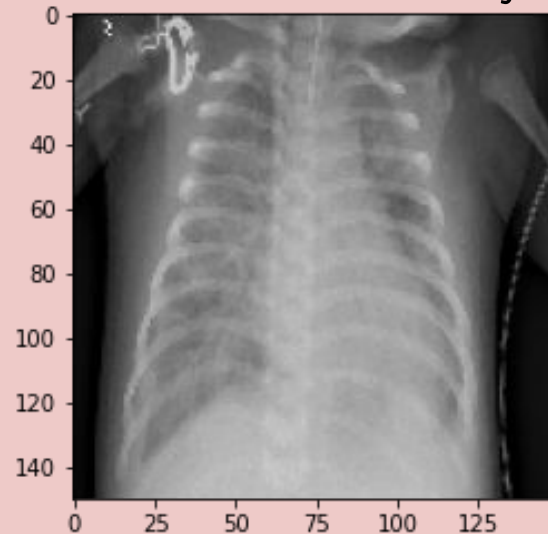
Source: Mayo Clinic and CDC

# PROBLEM STATEMENT

Build a neural network that classifies x-ray images of pediatric patients to identify whether or not they have pneumonia.

Normal X-Ray



Pneumonia X-Ray

# DATASET

From Kaggle.com titled
Chest X-Ray Images
(Pneumonia)

| | TRAIN | TEST | VALIDATION |
|---|---|---|---|
| **NORMAL** | 1341 | 234 | 8 |
| **PNEUMONIA** | 3875 | 390 | 8 |

# OSEMN

✓ OBTAIN DATA ──── Download from Kaggle. Import to Google Drive and unzip file for access.

✓ SCRUB DATA ──── Organize data into proper directories and check length.

✓ EXPLORE DATA ──── Visualizations, prepare model by reshaping and resizing images.

✓ MODEL DATA ──── First create baseline Neural Network models and work up to CNN Models.

✓ INTERPRET DATA ──── Explain best model use.

# METHODS USED

# CNN

✓ Load Images & Design Model

✓ Alternate convolutional and pooling layers.

✓ Include more parameters in later layers.

✓ Add final dense later to add a classifier to the convolutional base

# METHODS USED

# My Recommendation: CNN Model 3

```
Model: "sequential_16"
_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_44 (Conv2D)           (None, 62, 62, 64)        1792
_____
max_pooling2d_44 (MaxPooling (None, 31, 31, 64)        0
_____
conv2d_45 (Conv2D)           (None, 29, 29, 128)       73856
_____
max_pooling2d_45 (MaxPooling (None, 14, 14, 128)       0
_____
conv2d_46 (Conv2D)           (None, 12, 12, 512)       590336
_____
max_pooling2d_46 (MaxPooling (None, 6, 6, 512)         0
_____
conv2d_47 (Conv2D)           (None, 4, 4, 512)         2359808
_____
max_pooling2d_47 (MaxPooling (None, 2, 2, 512)         0
_____
flatten_12 (Flatten)         (None, 2048)              0
_____
dense_61 (Dense)             (None, 128)               262272
_____
dense_62 (Dense)             (None, 64)                8256
_____
dense_63 (Dense)             (None, 32)                2080
_____
dense_64 (Dense)             (None, 1)                 33
=================================================================
Total params: 3,298,433
Trainable params: 3,298,433
Non-trainable params: 0
_____
```

# Reasoning:

1. High Accuracy
2. High Recall
3. High Precision

```
-------------------------------------------------
CLASSIFICATION REPORT
                precision    recall   f1-score   support

      Normal       0.83       0.87       0.85       234
   Pneumonia       0.92       0.89       0.91       390

    accuracy                             0.88       624
   macro avg       0.88       0.88       0.88       624
weighted avg       0.89       0.88       0.89       624

-------------------------------------------------
CONFUSION MATRIX
```
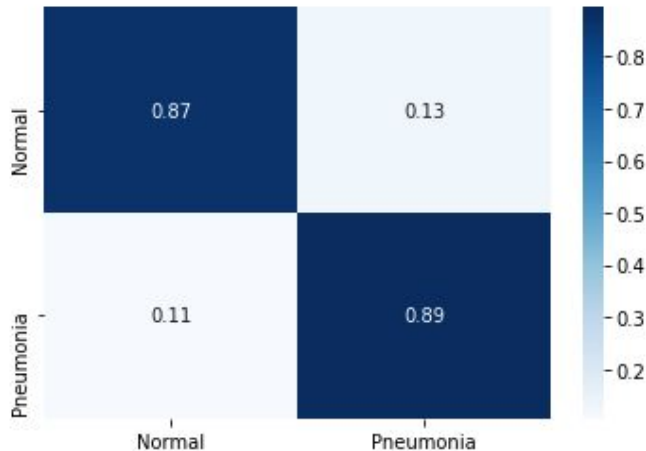
# Use a Convolutional Neural Network over a Basic Neural Network.

```
Model: "sequential"

Layer (type)                    Output Shape                Param #
=================================================================
dense (Dense)                   (None, 64)                  786496

dense_1 (Dense)                 (None, 32)                  2080

dense_2 (Dense)                 (None, 1)                   33
=================================================================
Total params: 788,609
Trainable params: 788,609
Non-trainable params: 0
```

## Basic Neural Network Model

```
------------------------------------------------
CLASSIFICATION REPORT
                precision     recall  f1-score    support

      Normal         0.78       0.74      0.76        234
   Pneumonia         0.85       0.87      0.86        390

    accuracy                              0.82        624
   macro avg         0.81       0.81      0.81        624
weighted avg         0.82       0.82      0.82        624

------------------------------------------------
CONFUSION MATRIX
```
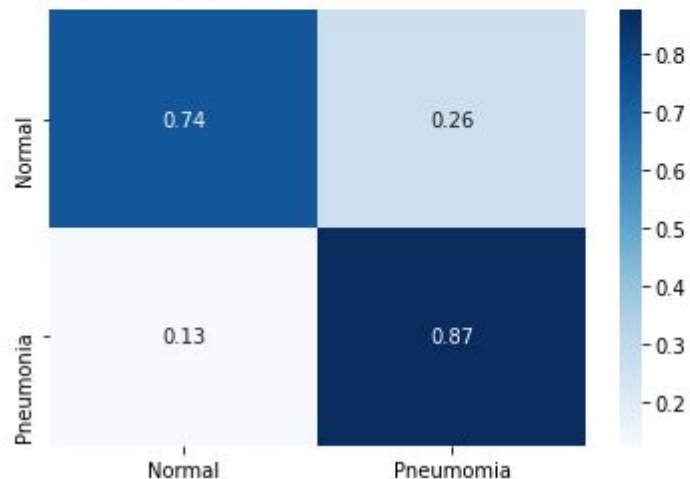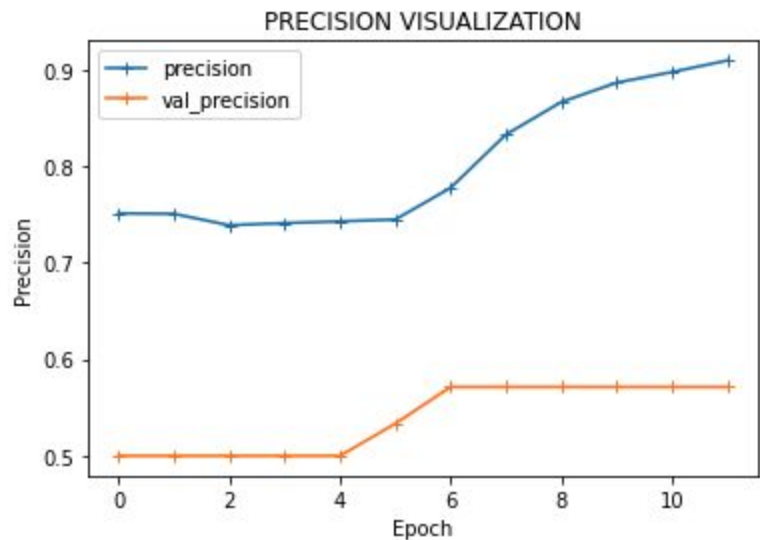


The Basic Neural Network has lower accuracy, precision, and recall

PRECISION VISUALIZATION

Convolutional Neural Network
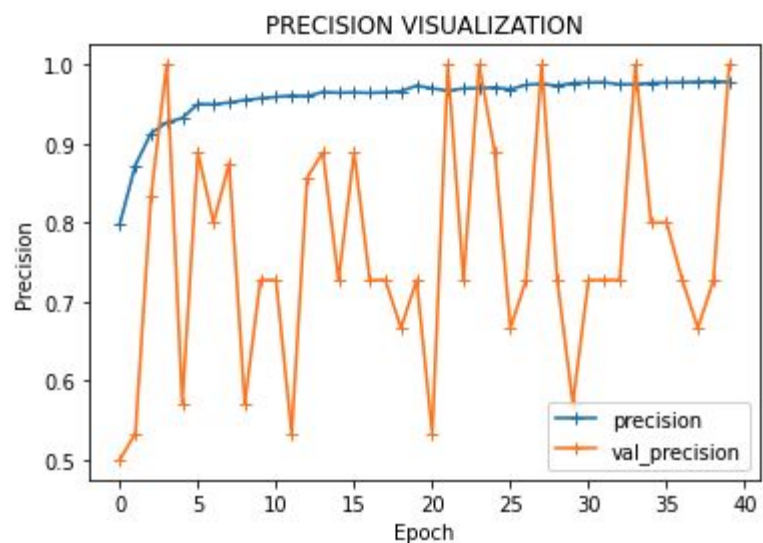
Basic Neural Network

# IMAGE SIZE

64 x 64 image size took 19.6 seconds to run a baseline neural network model. 82% accuracy

150 x 150 image size took 41.3 seconds to run the same baseline neural network model. 80% accuracy

*See Appendix One A & B for Images of runtime/accuracy*

- CNN Model 2 had a Pneumonia recall of 100%
- Recall states how many of the actual cases the model found.

```
--------------------------------------------------
CLASSIFICATION REPORT
                precision    recall   f1-score   support

       Normal      0.99       0.33      0.49        234
    Pneumonia      0.71       1.00      0.83        390

     accuracy                           0.75        624
    macro avg      0.85       0.66      0.66        624
 weighted avg      0.82       0.75      0.70        624

--------------------------------------------------
```

**Recall / Sensitivity**

$$\frac{TP}{(TP + FN)}$$

Appendix 3 for Model Structure

# Precision or Recall?

- **CNN Model 3 the highest precision at 92%**
- Precision states the percentage of positive predictions the model stated correctly.

```
--------------------------------------------------
CLASSIFICATION REPORT
                precision     recall    f1-score     support

      Normal       0.83        0.87       0.85          234
   Pneumonia       0.92        0.89       0.91          390

    accuracy                              0.88          624
   macro avg       0.88        0.88       0.88          624
weighted avg       0.89        0.88       0.89          624

--------------------------------------------------
```

**Precision**

$$\frac{TP}{(TP + FP)}$$

Appendix 4 for Model Structure

# Precision or Recall: Depends on the Case

| Choose Precision if... | Choose Recall if... |
|---|---|
| The cure is a tough or dangerous processes. Example: You don't want to give someone chemo or radiation when they don't actually have the illness. | Miss identifying leads to more deaths. Recall will help identify more people as sick. |
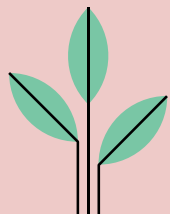
# Future Recommendations

1 Use a larger dataset to improve accuracy and precision

2 Look at X-Rays of adult lungs with and without pneumonia

3 Use premade CNN to see if they could potentially work better.

# THANK YOU FOR YOUR TIME!

Questions?
Feel free to contact me at:
Lauren.Esser02@gmail.com
or
Lauren Esser on LinkedIn

# APPENDIX 1A: 64x64

```
Epoch 37/40
105/105 [==============================] - 0s 4ms/step - loss: 0.0906 - acc: 0.9655
Epoch 38/40
105/105 [==============================] - 0s 4ms/step - loss: 0.0925 - acc: 0.9670
Epoch 39/40
105/105 [==============================] - 0s 4ms/step - loss: 0.0885 - acc: 0.9691
Epoch 40/40
105/105 [==============================] - 0s 4ms/step - loss: 0.0854 - acc: 0.9674
Time to Run:  0:00:21.512352
```

```
--------------------------------------------------
CLASSIFICATION REPORT
                precision    recall   f1-score    support

      Normal       0.76       0.75       0.76        234
   Pneumonia       0.85       0.86       0.85        390

    accuracy                             0.82        624
   macro avg       0.81       0.80       0.80        624
weighted avg       0.82       0.82       0.82        624

--------------------------------------------------
```

# APPENDIX 1B: 150x150

```
Epoch 37/40
105/105 [==============================] - 1s 10ms/step - loss: 0.1022 - acc: 0.9618
Epoch 38/40
105/105 [==============================] - 1s 9ms/step - loss: 0.1081 - acc: 0.9584 -
Epoch 39/40
105/105 [==============================] - 1s 10ms/step - loss: 0.1000 - acc: 0.9649
Epoch 40/40
105/105 [==============================] - 1s 10ms/step - loss: 0.0981 - acc: 0.9653
Time to Run:  0:00:41.168897
```

```
-------------------------------------------------
CLASSIFICATION REPORT
                 precision    recall   f1-score   support

       Normal        0.81      0.63      0.71       234
    Pneumonia        0.80      0.91      0.85       390

     accuracy                            0.80       624
    macro avg        0.81      0.77      0.78       624
 weighted avg        0.80      0.80      0.80       624


-------------------------------------------------
```

# APPENDIX 2

```
Model: "sequential_16"

_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_44 (Conv2D)           (None, 62, 62, 64)        1792
_____
max_pooling2d_44 (MaxPooling (None, 31, 31, 64)        0
_____
conv2d_45 (Conv2D)           (None, 29, 29, 128)       73856
_____
max_pooling2d_45 (MaxPooling (None, 14, 14, 128)       0
_____
conv2d_46 (Conv2D)           (None, 12, 12, 512)       590336
_____
max_pooling2d_46 (MaxPooling (None, 6, 6, 512)         0
_____
conv2d_47 (Conv2D)           (None, 4, 4, 512)         2359808
_____
max_pooling2d_47 (MaxPooling (None, 2, 2, 512)         0
_____
flatten_12 (Flatten)         (None, 2048)              0
_____
dense_61 (Dense)             (None, 128)               262272
_____
dense_62 (Dense)             (None, 64)                8256
_____
dense_63 (Dense)             (None, 32)                2080
_____
dense_64 (Dense)             (None, 1)                 33
=================================================================
Total params: 3,298,433
Trainable params: 3,298,433
Non-trainable params: 0
_____
```

# APPENDIX 3

```
Model: "sequential_5"

_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 62, 62, 64)        1792

max_pooling2d_2 (MaxPooling2 (None, 31, 31, 64)        0

conv2d_3 (Conv2D)            (None, 29, 29, 96)        55392

max_pooling2d_3 (MaxPooling2 (None, 14, 14, 96)        0

flatten_1 (Flatten)          (None, 18816)             0

dense_17 (Dense)             (None, 64)                1204288

dense_18 (Dense)             (None, 32)                2080

dense_19 (Dense)             (None, 16)                528

dense_20 (Dense)             (None, 1)                 17
=================================================================
Total params: 1,264,097
Trainable params: 1,264,097
Non-trainable params: 0
_____
```

# APPENDIX 4

```
Model: "sequential"

_____
Layer (type)                    Output Shape              Param #
=================================================================
dense (Dense)                   (None, 64)                786496
_____
dense_1 (Dense)                 (None, 32)                2080
_____
dense_2 (Dense)                 (None, 1)                 33
=================================================================
Total params: 788,609
Trainable params: 788,609
Non-trainable params: 0
_____
```

# APPENDIX 5A

```
Model: "sequential_16"

Layer (type)                Output Shape              Param #
=================================================================
conv2d_44 (Conv2D)          (None, 62, 62, 64)        1792

max_pooling2d_44 (MaxPooling (None, 31, 31, 64)       0

conv2d_45 (Conv2D)          (None, 29, 29, 128)       73856

max_pooling2d_45 (MaxPooling (None, 14, 14, 128)      0

conv2d_46 (Conv2D)          (None, 12, 12, 512)       590336

max_pooling2d_46 (MaxPooling (None, 6, 6, 512)        0

conv2d_47 (Conv2D)          (None, 4, 4, 512)         2359808

max_pooling2d_47 (MaxPooling (None, 2, 2, 512)        0

flatten_12 (Flatten)        (None, 2048)              0

dense_61 (Dense)            (None, 128)               262272

dense_62 (Dense)            (None, 64)                8256

dense_63 (Dense)            (None, 32)                2080

dense_64 (Dense)            (None, 1)                 33
=================================================================
Total params: 3,298,433
Trainable params: 3,298,433
Non-trainable params: 0
```

# APPENDIX 5B