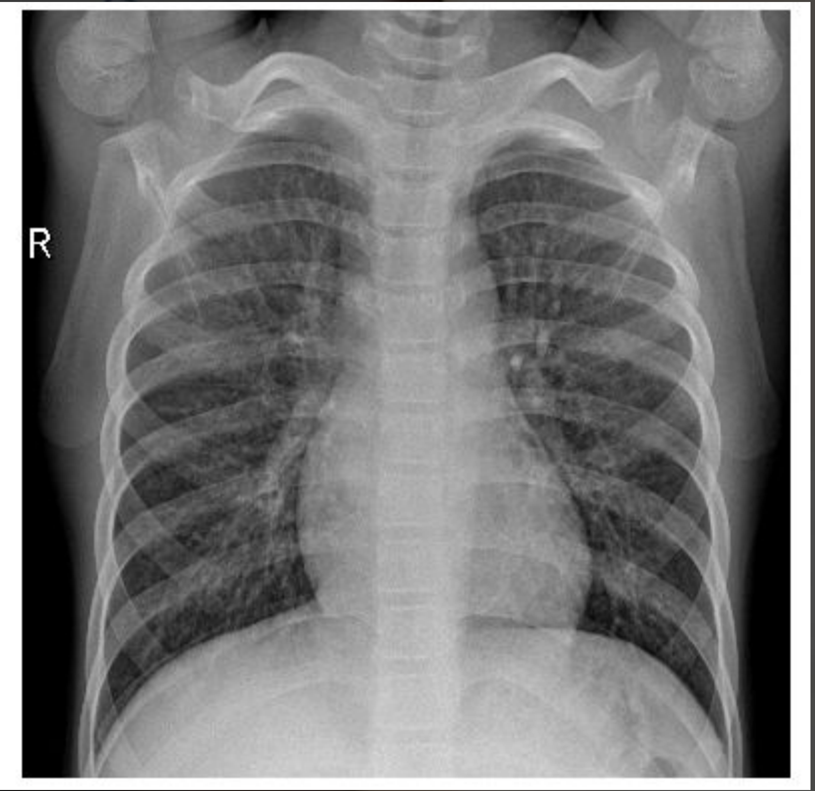
Pneumonia Classifier

Mod 4 Project
Alex Macy

Mission: Identify instances of Pneumonia from X-Ray images



Our Data

Three datasets:

1. Train, Validation, Test
   a. Normal
   b. Pneumonia

1. Train: 5216 images belonging to two classes.
2. Validation: 16 images belonging to two classes.
3. Test: 624 images belonging to two classes.

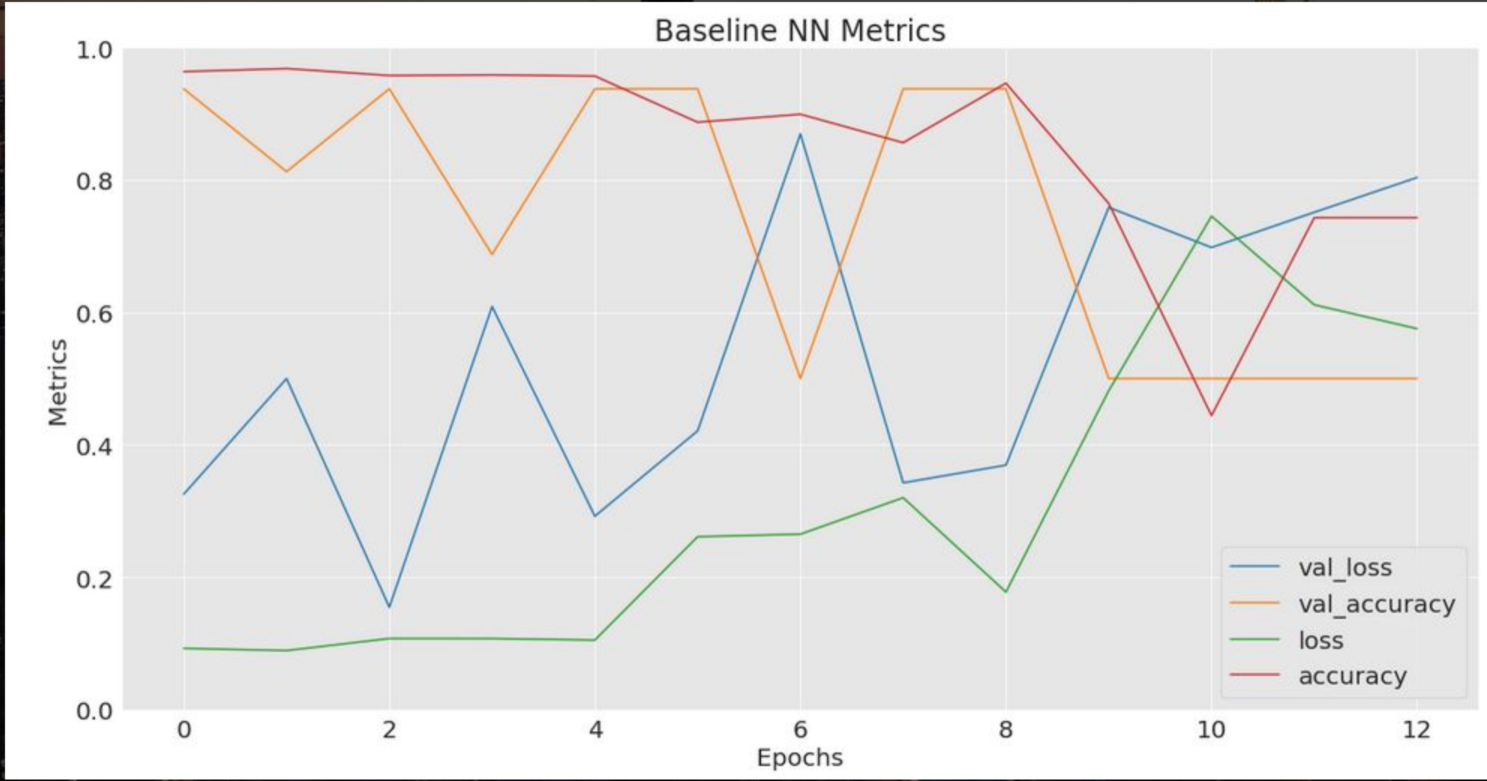
Baseline Modeling

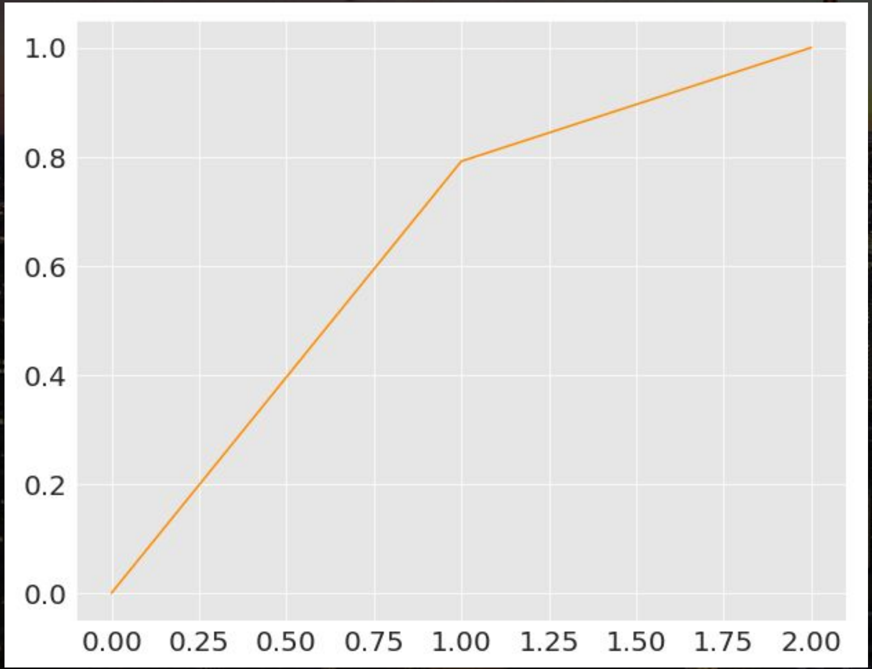
```
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_10 (Flatten)         (None, 442368)            0
_________________________________________________________________
dense_21 (Dense)             (None, 300)               132710700
_________________________________________________________________
dense_22 (Dense)             (None, 100)               30100
_________________________________________________________________
dense_23 (Dense)             (None, 1)                 101
=================================================================
Total params: 132,740,901
Trainable params: 132,740,901
Non-trainable params: 0
```

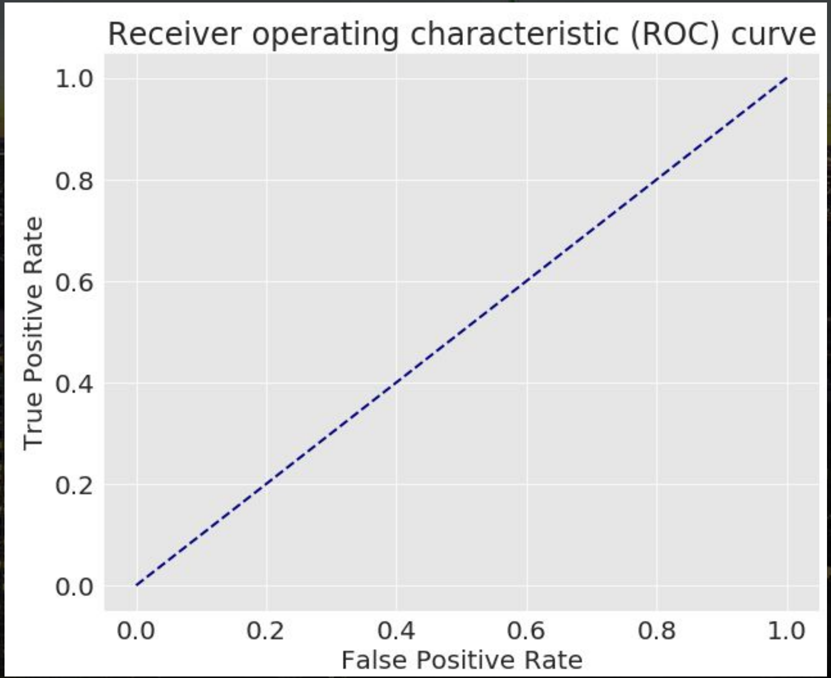
Create a base Sequential Neural Network to guage data

Baseline Test Accuracy: 80%

Result unreliable -- Accuracy and Loss metrics too tacky



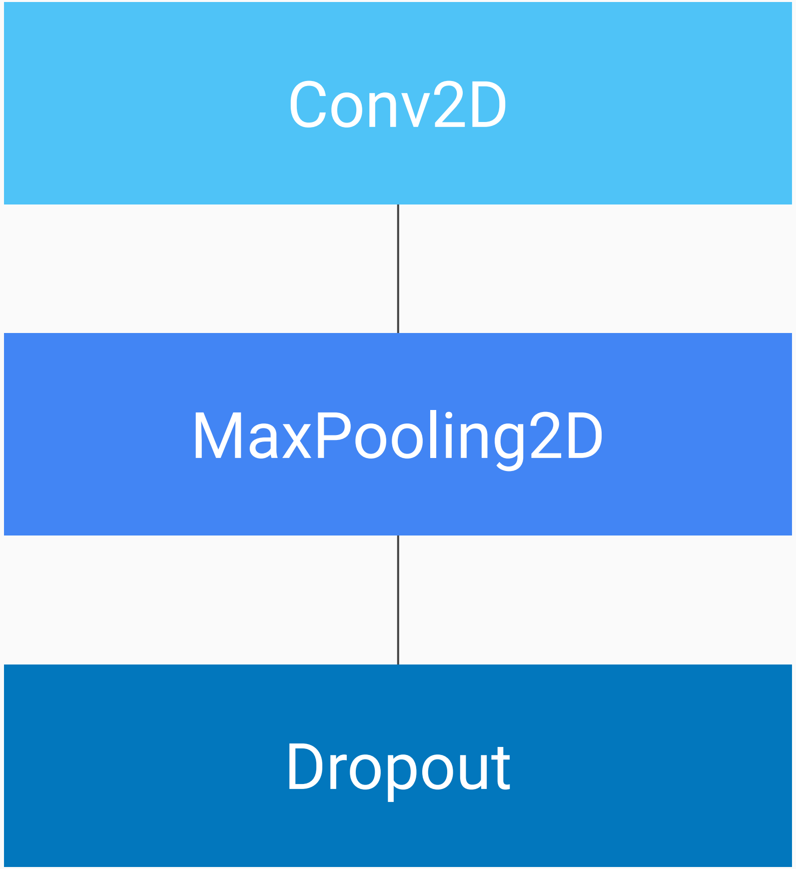

1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Receiver operating characteristic (ROC) curve
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Next Steps:
A more complex model

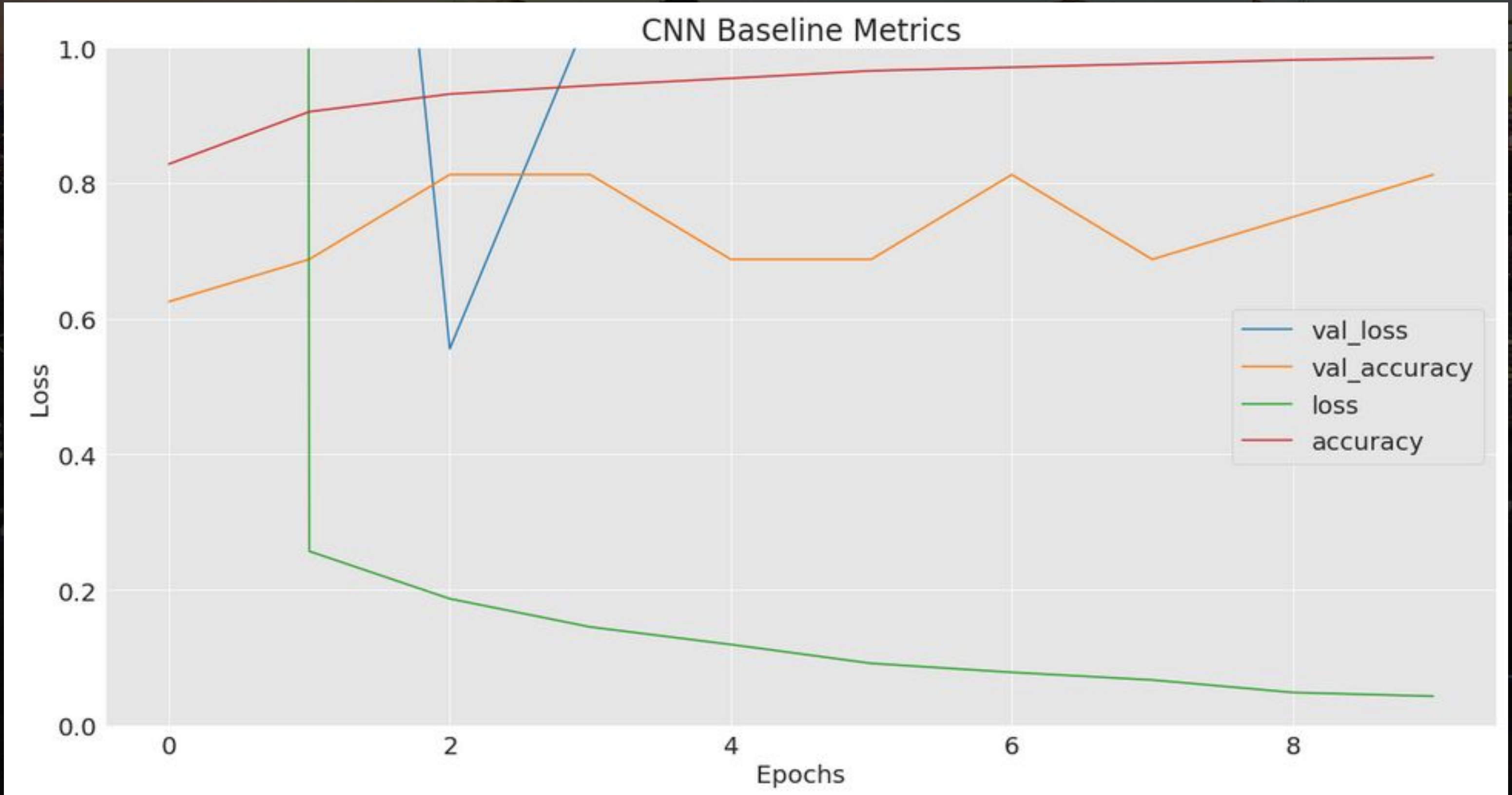
Layering

Our baseline, naive model consisted of two Dense layers. Perhaps a more complex model will perform better.

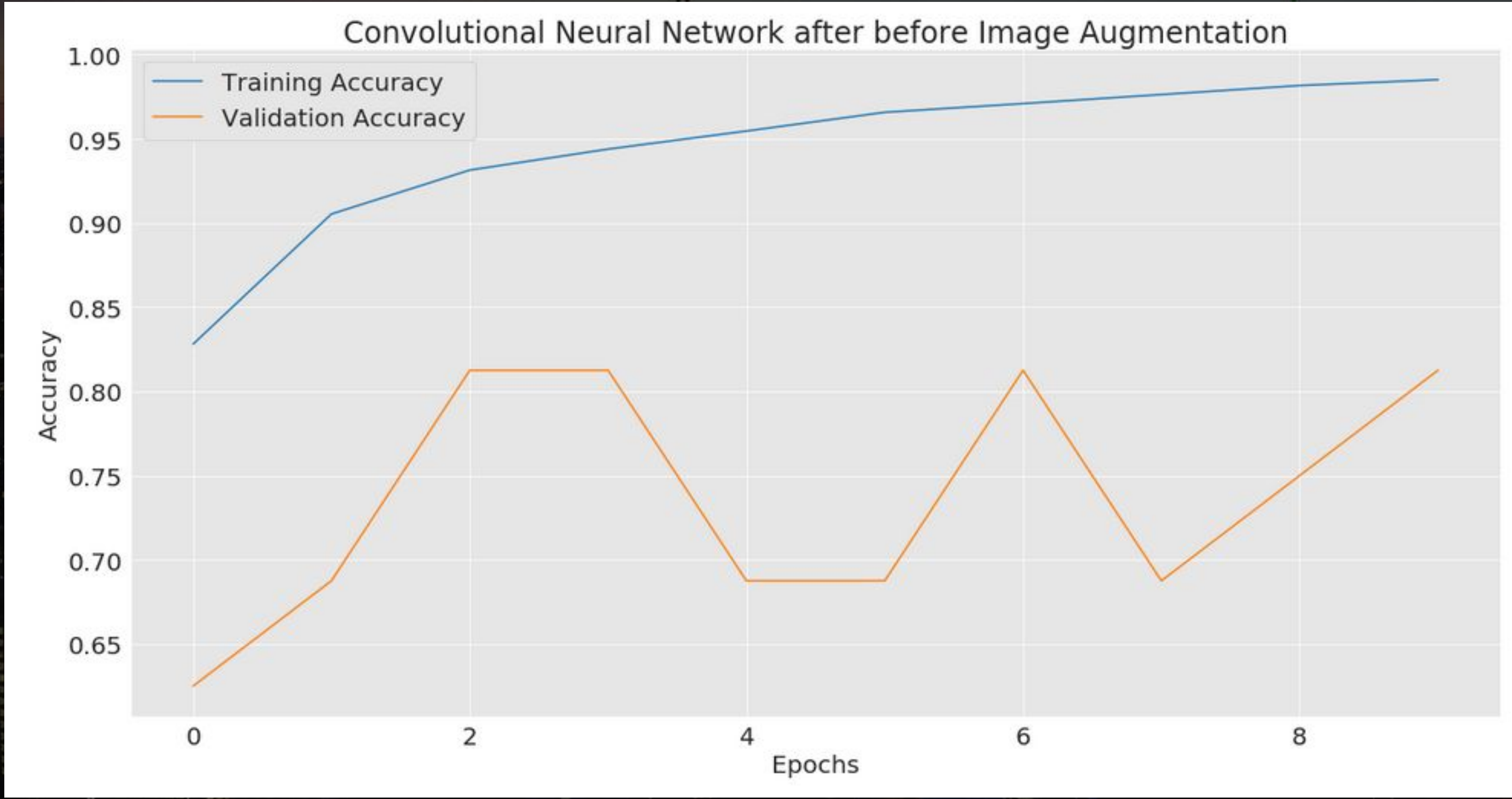


Updated Test Accuracy: 70%

Result unreliable -- Accuracy and Loss metrics too tacky

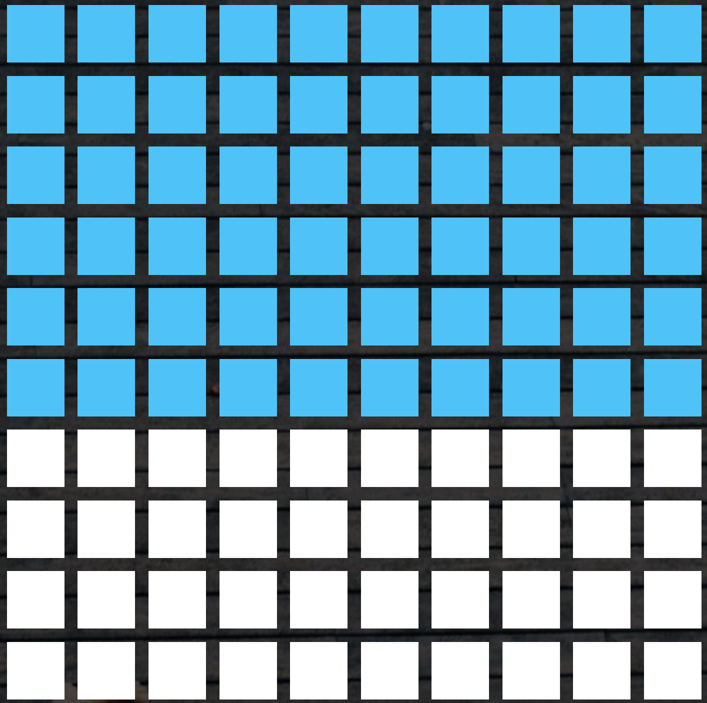


But improvement...



What now?

Enhancing the scope of our Neural Network only provides so much improvement. Manipulating the data may generate better results.



Data Augmentation

Using ImageDataGenerator, we can create new feature mappings that will improve our neural network's performance

Shear Range

Zoom Range

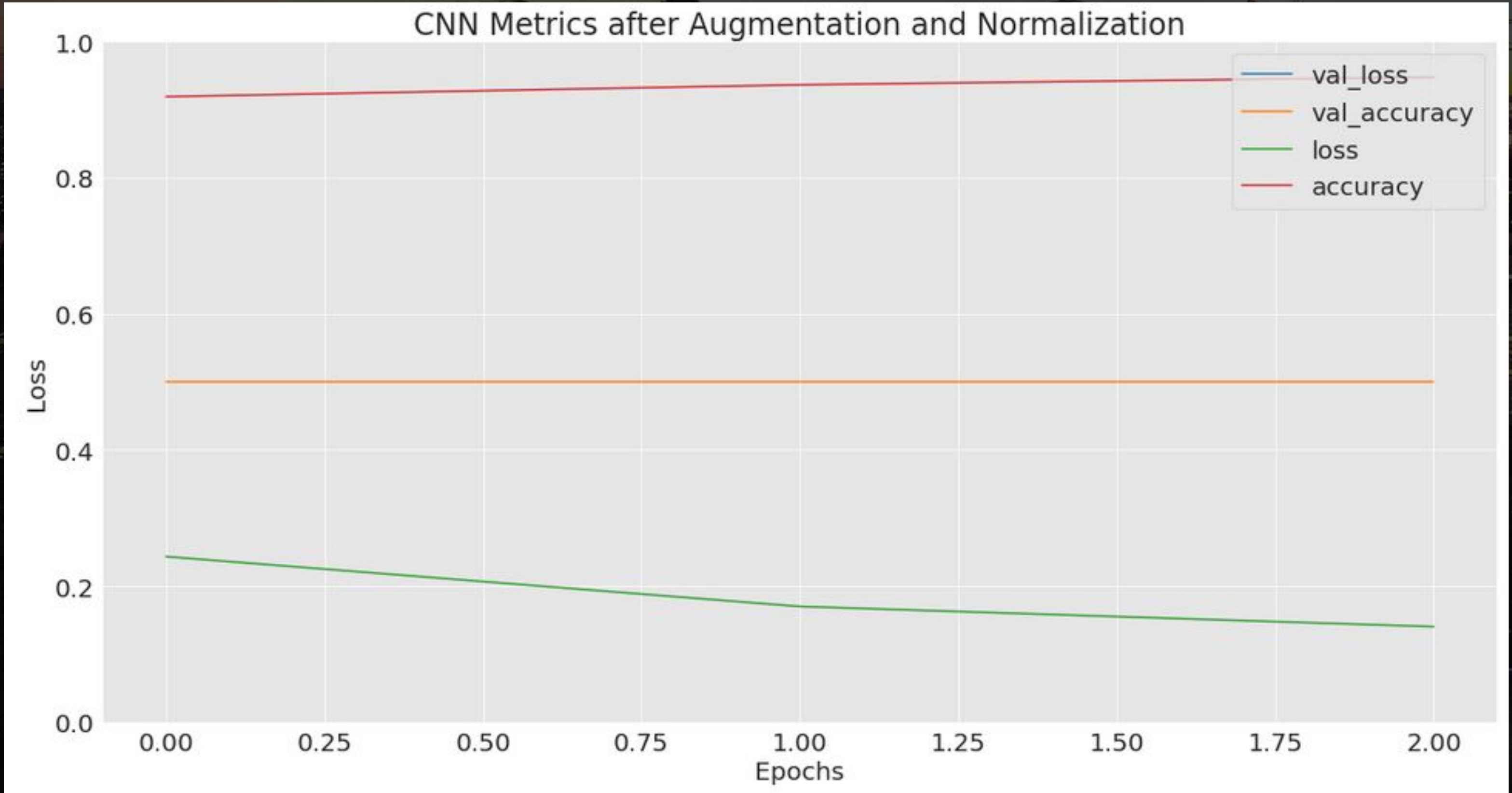
Horizontal Flip

Enhanced Modeling

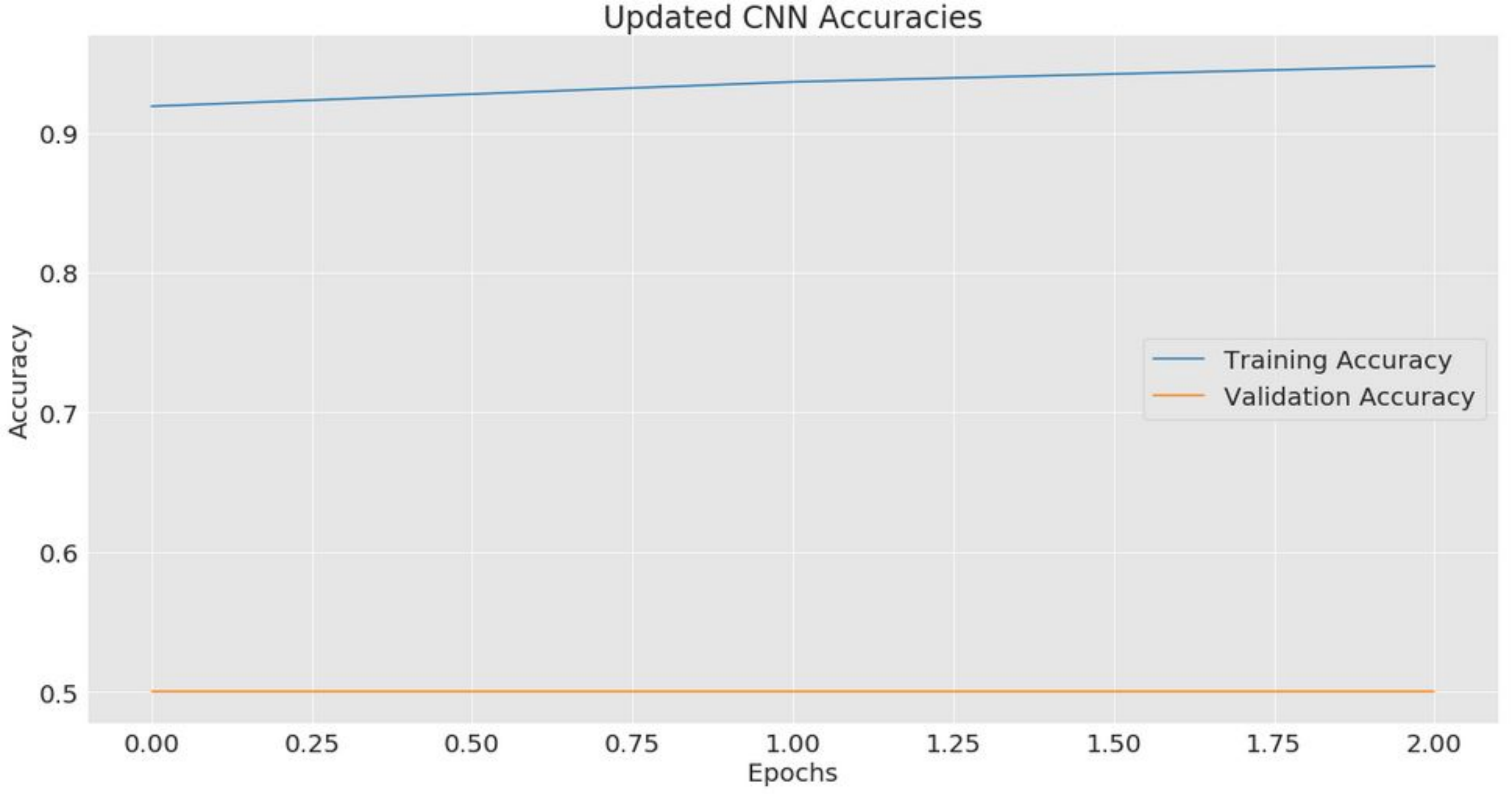
With more layers, image augmentation, and batch normalizing, our network should perform best

```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_18 (Batc (None, 384, 384, 3)       12
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 384, 384, 32)      896
_________________________________________________________________
batch_normalization_19 (Batc (None, 384, 384, 32)      128
_________________________________________________________________
activation_55 (Activation)   (None, 384, 384, 32)      0
_________________________________________________________________
batch_normalization_20 (Batc (None, 384, 384, 32)      128
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 382, 382, 32)      9248
_________________________________________________________________
batch_normalization_21 (Batc (None, 382, 382, 32)      128
_________________________________________________________________
activation_56 (Activation)   (None, 382, 382, 32)      0
_________________________________________________________________
batch_normalization_22 (Batc (None, 382, 382, 32)      128
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 191, 191, 32)      0
_________________________________________________________________
batch_normalization_23 (Batc (None, 191, 191, 32)      128
_________________________________________________________________
dropout_28 (Dropout)         (None, 191, 191, 32)      0
_________________________________________________________________
batch_normalization_24 (Batc (None, 191, 191, 32)      128
_________________________________________________________________
conv2d_40 (Conv2D)           (None, 191, 191, 64)      18496
_________________________________________________________________
batch_normalization_25 (Batc (None, 191, 191, 64)      256
_________________________________________________________________
activation_57 (Activation)   (None, 191, 191, 64)      0
_________________________________________________________________
batch_normalization_26 (Batc (None, 191, 191, 64)      256
_________________________________________________________________
```

Updated Test Accuracy: 72%
Much more reliable! But still problems...

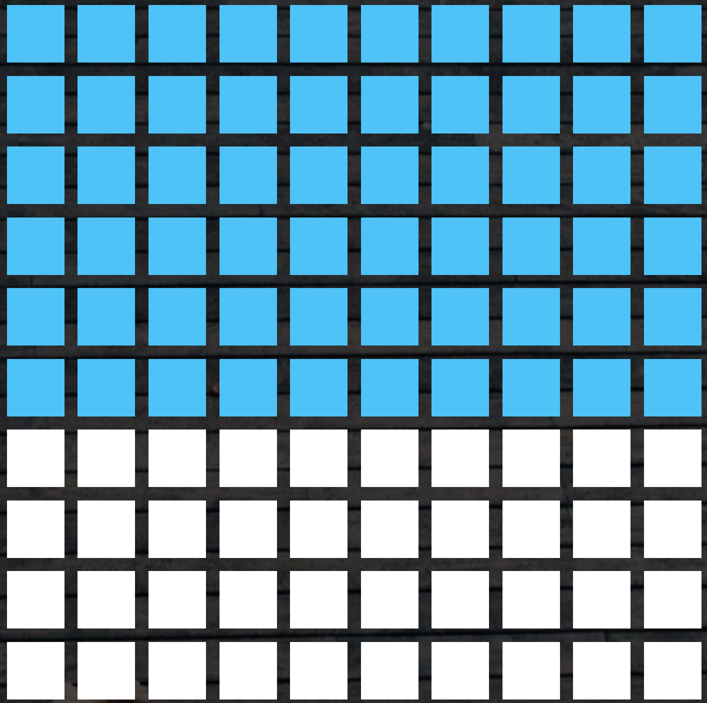


Validation set has been a reoccuring problem.



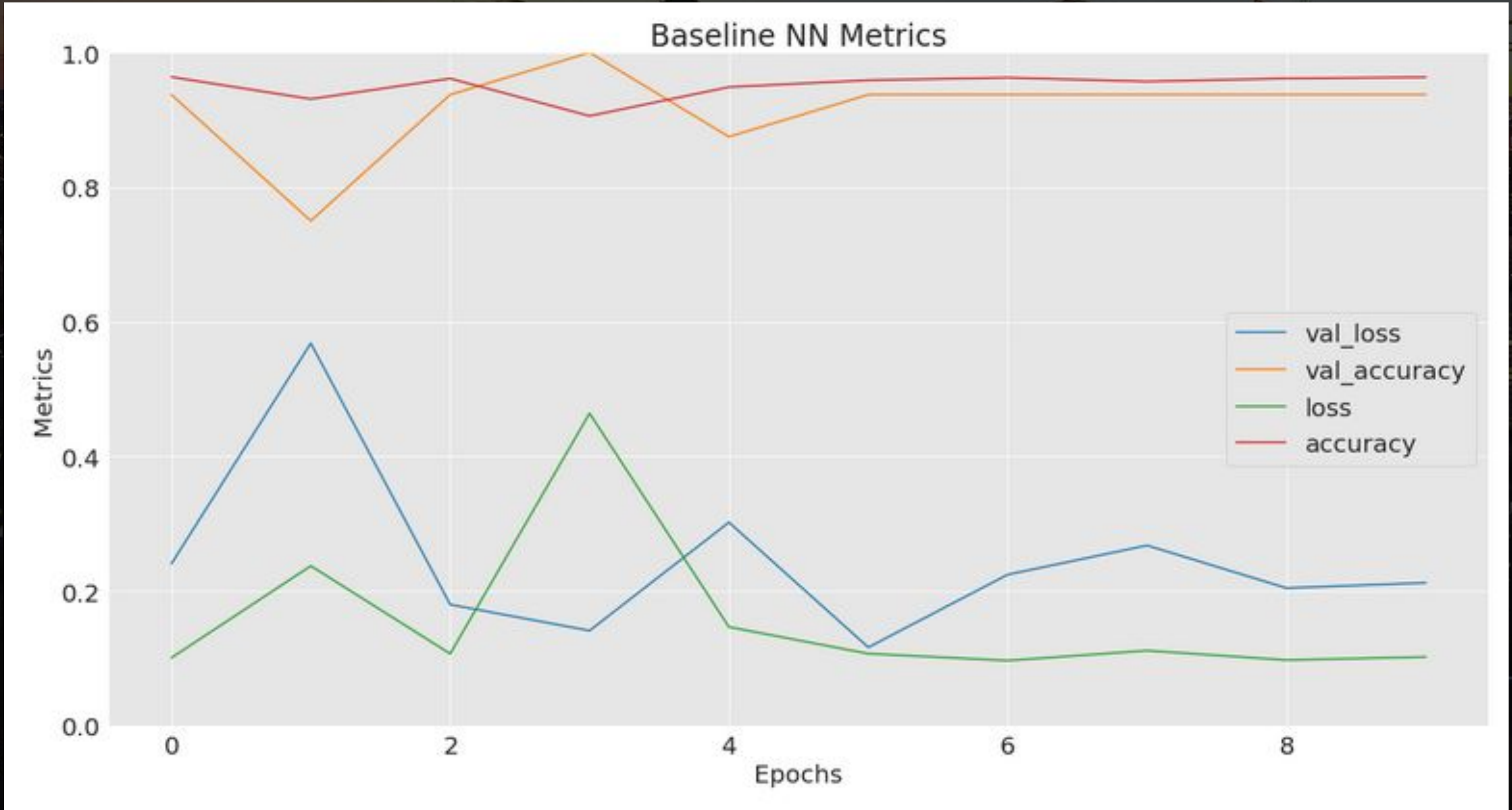
Class Imbalance

Since there are so few examples for our validation set, our NN is predisposed to incorrectly identify pneumonia. Class balancing will help reconcile this...

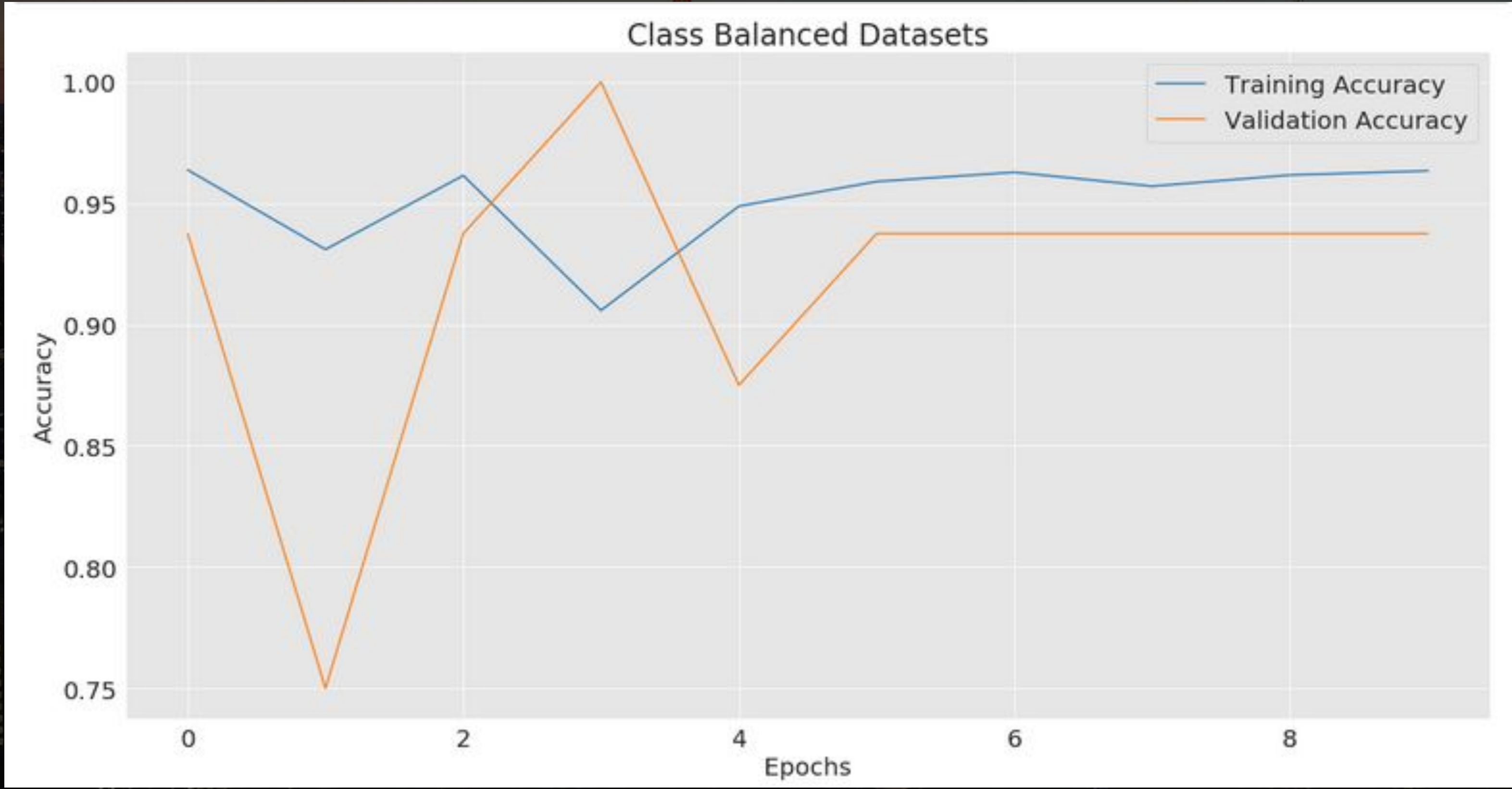


Balanced Test Accuracy: 77%

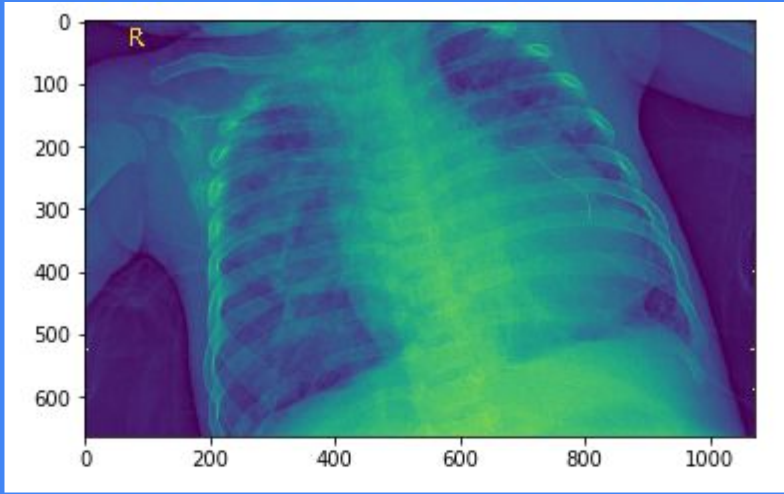
Validation data no longer overfitting!

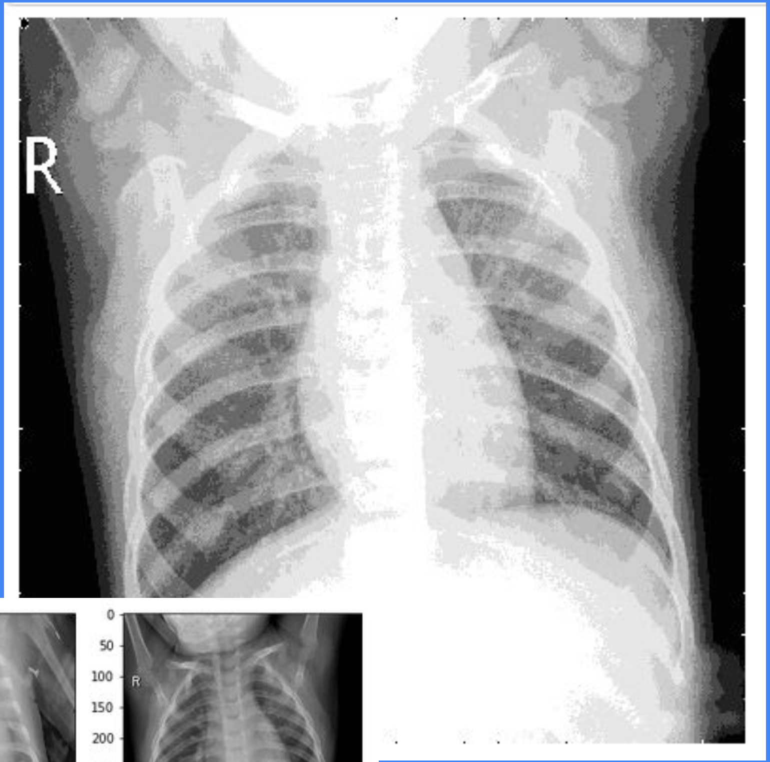


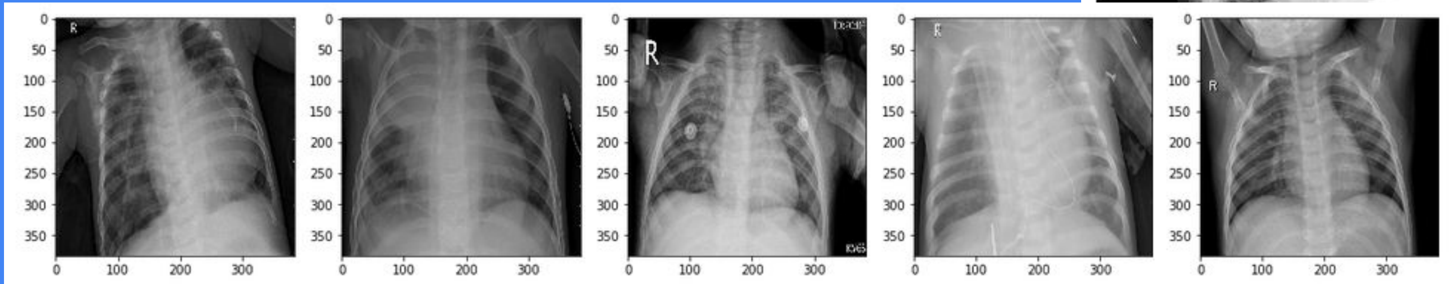
Balanced Test Accuracy: 77%
Validation data no longer overfitting!



Further Augmentation







Conclusion:

Cost-Benefit Analysis is not yet reliable, but consequences clear. Further modeling is needed, aided by better data. Data Augmentation can help make better data if it is not readily available