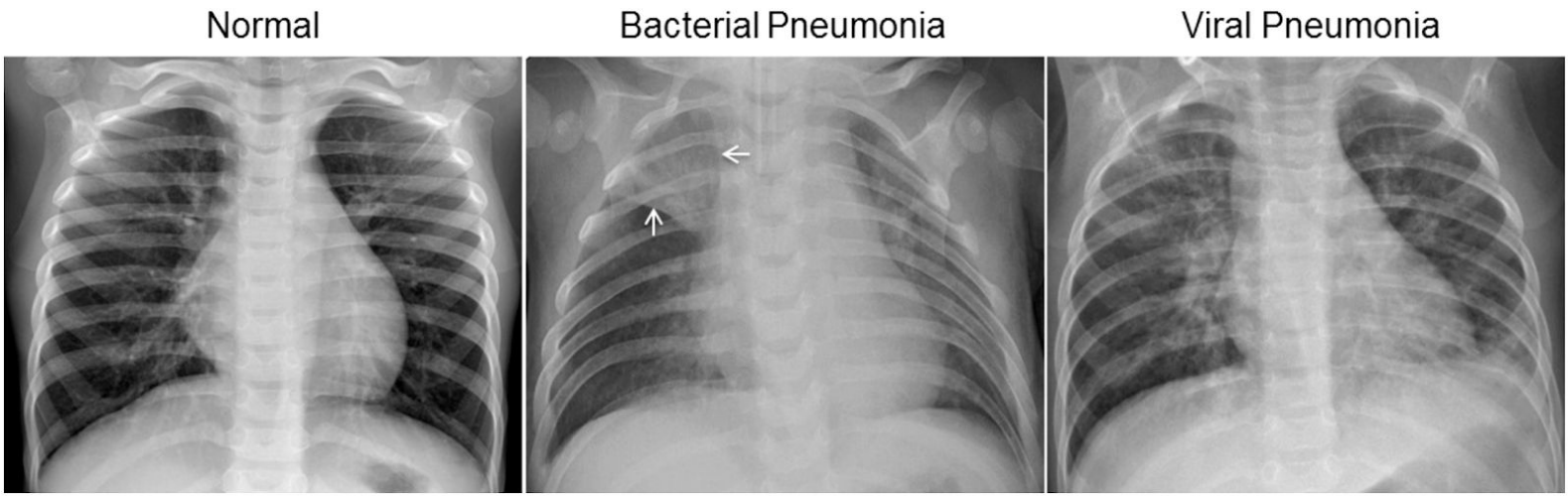
AI Model to Assist in Pneumonia Detection Final Report

Data Science Project by Chris Woods



Problem Statement

According to the World Health Organization (WHO), pneumonia kills about 2 million children under 5 years old every year and is consistently estimated as the single leading cause of childhood mortality (Rudan et al., 2008), killing more children than HIV/AIDS, malaria, and measles combined (Adegbola, 2012). The WHO reports that nearly all cases (95%) of new-onset childhood clinical pneumonia occur in developing countries, particularly in Southeast Asia and Africa. Bacterial and viral pathogens are the two leading causes of pneumonia (Mcluckie, 2009) but require very different forms of management. Bacterial pneumonia requires urgent referral for immediate antibiotic treatment, while viral pneumonia is treated with supportive care. Therefore, accurate and timely diagnosis is a matter of life or death.

The standard of care in the diagnosis of pneumonia is the chest radiograph (X-ray). However, since the greatest incidence childhood pneumonia occurs in poor socio-economic areas, timely interpretation of the x-ray images is not always available.

The purpose of this project is to determine if an AI model can be developed to assist in the diagnosis of pneumonia using common radiographs (chest x-rays). Because we are dealing with human beings, the *Accuracy* of the model will not be the critical factor in determining "best" model. For this project I will use *Recall*, or Sensitivity as the metric of choice. Recall answers the question, what proportion of actual positives are identified correctly. Mathematically, Recall is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$

This project will be carried out in two phases:

- **Phase I** - First, a baseline must be established. Create a model that will distinguish normal radiographs from those with pneumonia present. A good result in Phase I will allow us to remove normal radiographs from the analysis and focus solely on a model that can differentiate bacterial from viral pneumonia.
- **Phase II** - If pneumonia is present, develop a model to differentiate bacterial from viral pneumonia.
    - "Pneumonia caused by bacterial infections poses a much greater threat to the heart than pneumonia caused by viral infections, a new study suggests. Patients in the study who were diagnosed with bacterial pneumonia had a higher risk of heart attack, stroke or death, compared with patients diagnosed with viral pneumonia, the researchers found." Link

False negatives (FN) are of critical importance here. In this context, a false negative would mean that a child who actually had bacterial pneumonia would go on undiagnosed. The ramifications of this error could be life threatening.

In Phase I, I will be looking to maximize the Recall of the pneumonia present prediction. In Phase II, I will be concerned with maximizing the Recall of the bacterial pneumonia present prediction. Decreasing the numbers of FN's, or increasing the Recall, would facilitate more timely referrals of children needing immediate care, therefore increasing the odds of a positive outcome for the child.

Dataset Description

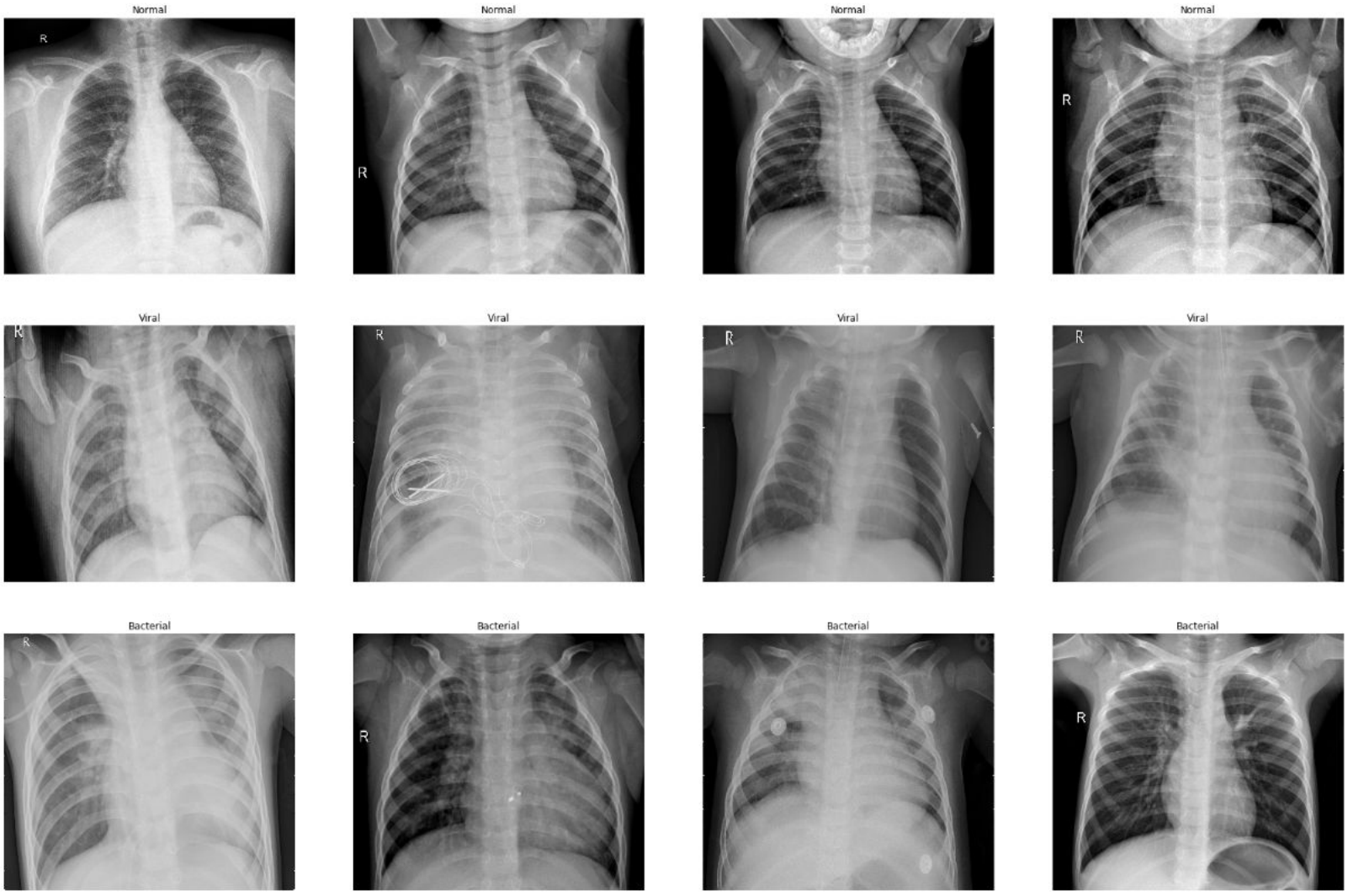
For this project I will be using the Chest X-Ray Images (Pneumonia) dataset. A total of 5,840 chest X-ray images from children, including 4,256 characterized as depicting pneumonia (2,772 bacterial and 1,493 viral) and 1,575 normal, from a total of 5,856 patients will be used.

Chest X-ray images (anterior-posterior) were selected from pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou, China. All

chest X-ray imaging was performed as part of patients' routine clinical care. The images were first screened for image quality and all low quality scans were removed from the dataset. The images were then graded by two expert physicians.

The goal of this project would be to increase the overall accuracy of pneumonia diagnosis, and if present, if the pneumonia is viral or bacterial, using machine learning principles.

Analysis of the Images



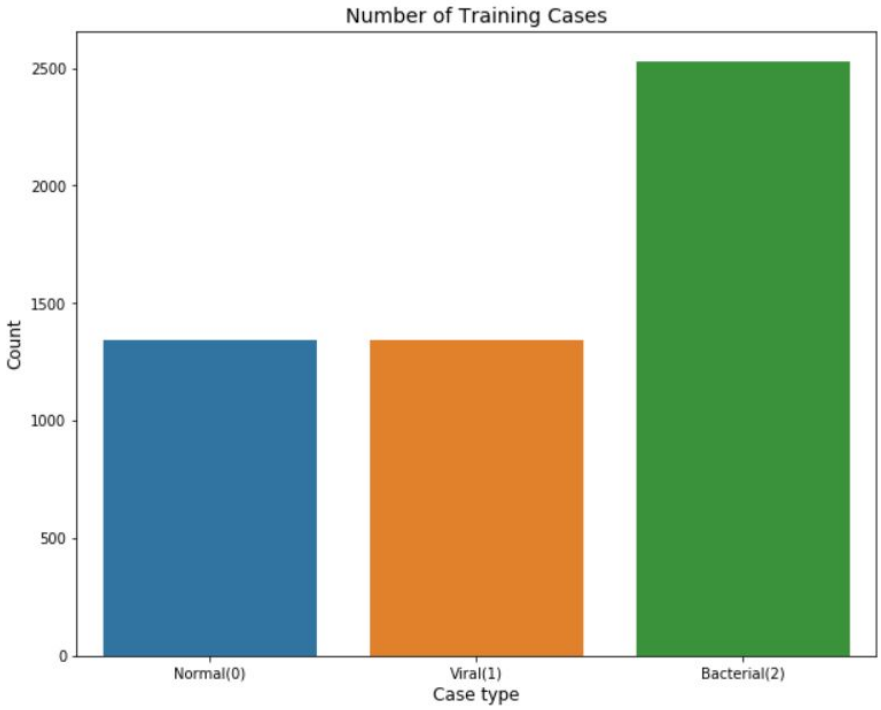
As we can see above, differentiating between a normal radiograph and one with pneumonia present can be quite challenging. Being able to detect alveolar infiltrates vs lobar infiltrates vs interstitial fluid on a chest radiograph can be difficult. Modern medical

facilities in first world countries have multiple modalities and tests at their disposal in order to make a proper diagnosis.

However, many third world clinics where death from pneumonia is prevalent, do not have access to the same modalities as we in the west. Which is why a machine learning application that could accurately diagnose pneumonia would be awesome.

Data Cleaning and Wrangling

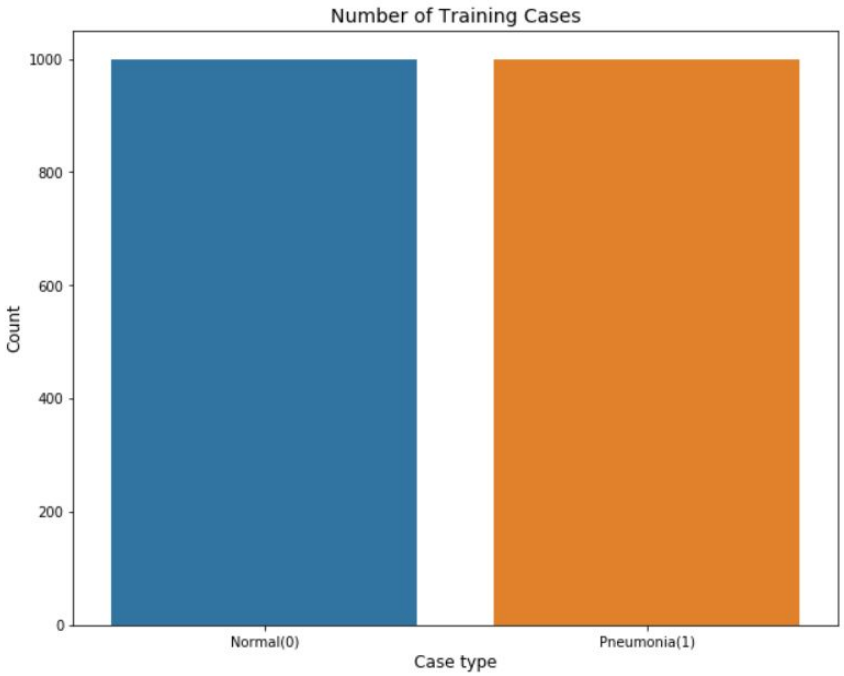
This was a clean dataset consisting of 5,840 x-ray images stored as jpeg files split into training and test directories. The original distribution of images in the training directory had substantially more bacterial pneumonia images than normal and viral pneumonia images.



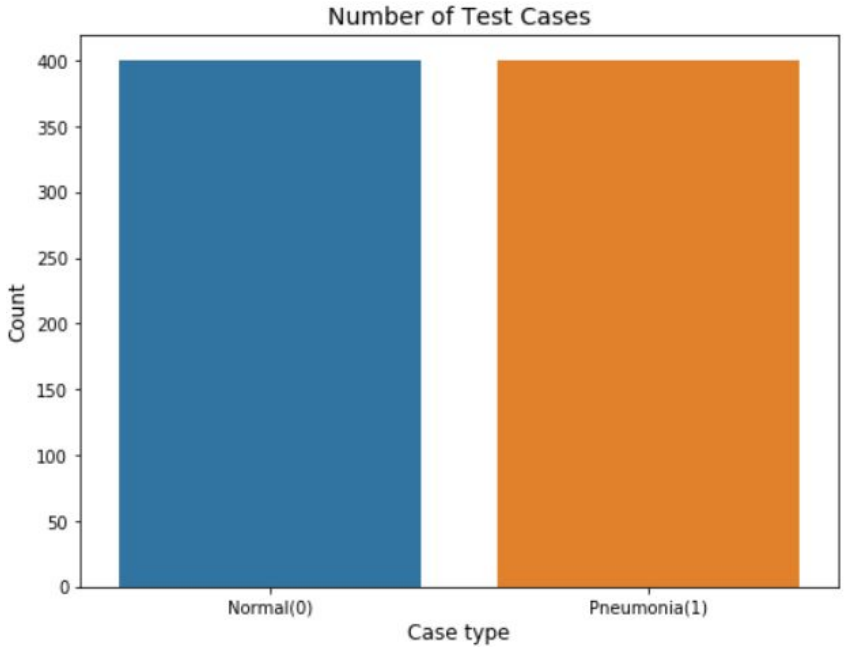
Here we can see somewhat of an imbalance in the training data cases, with bacterial pneumonia represented at almost twice the rate of the viral pneumonia and normal cases.

- 1341 Normal images
- 1345 Viral Pneumonia images
- 2530 Bacterial Pneumonia images

In Phase I, to lessen the chances of overfitting and bias toward classification as bacterial pneumonia, I redistributed the images in both test and training directories. Now I have 1,000 normal images and 1,000 pneumonia images to train the model.



And 400 normal, and 400 pneumonia images for the test dataset.



In order to process the images in the ConvNet, Keras ImageDataGenerator() class was used. Rather than performing the operations on your entire image dataset in memory, the API is designed to be iterated by the deep learning model fitting process, creating augmented image data for you just-in-time. This also provided a platform for image augmentation.

Machine Learning

For this project I will use Deep Learning, a type of supervised learning algorithm that uses multi-layered neural networks to make predictions and classify outcomes. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer — that represent the predictions.

- In the most basic sense, deep learning models first identify lower level features (straight lines, diagonal lines, horizontal lines, ect.) that are most relevant to properly identifying the image given.
- It then builds on this hierarchy to find what combination of shapes and edges are present.
- After consecutive hierarchical identifications of more complex features the model then makes a prediction.

Convolutional Neural Network

The type of network I use in this project is called a Convolutional Neural Network (CNN).

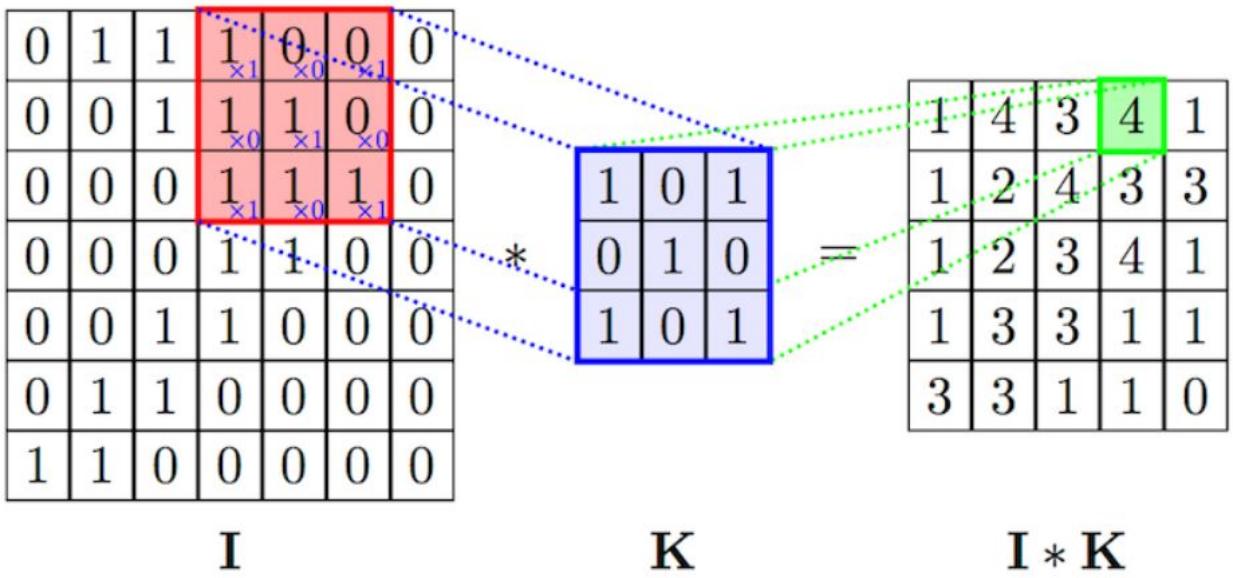
CNN's are slightly different than regular neural networks.

- The layers are organised in 3 dimensions: width, height and depth.
- The neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it.
- Final output will be reduced to a single vector of probability scores, organized along the depth dimension.

CNN's have two components:

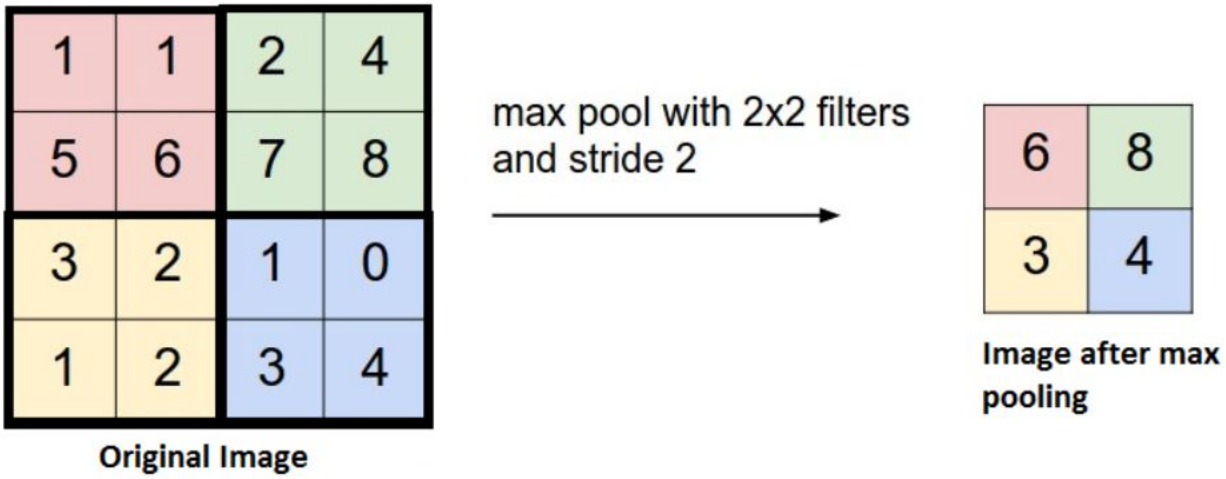
Hidden layers / Feature extraction - Here the network will perform a series of **convolutions** and **pooling** operations during which features are detected.

A **convolution** is executed by sliding the filter over the input image. At every location matrix multiplication is performed and sums the result onto the feature map.

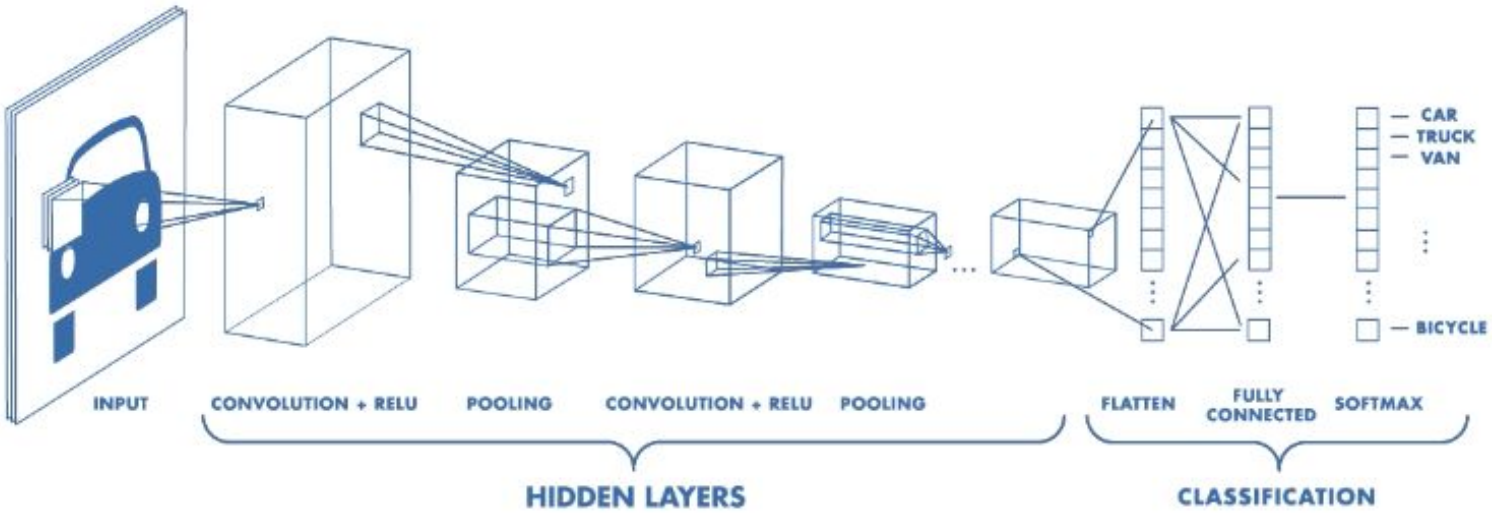


Convolution layers are the main powerhouse of a CNN model. Given an image, the convolutional layer detects the features that make the image unique. The convolution layers learn such complex features by building on top of each other. The first layers detect edges, the next layers combine them to detect shapes, to following layers merge this information to infer a particular feature. The CNN doesn't really know or care what the feature is, but by seeing a lot of them in images, it learns to detect it as a feature.

It is common to add a **pooling** layer between CNN layers. The function of pooling is to continuously reduce the dimensionality to reduce the number of parameters and computation in the network. This shortens the training time and controls overfitting. The most frequent type of pooling is max pooling, which takes the maximum value in each window. This decreases the feature map size while at the same time keeping the significant information.



Classification - Fully connected layers will serve as a classifier on top of the extracted features and will assign a probability for the image being what the algorithm predicts.



Architecture of a CNN

CNN's are considered the gold standard model for image related problems. The main advantage of CNN compared to other models is that it automatically detects important features without any human intervention! It learns the distinctive features for each class by itself.

Machine Learning Phase 1 - Differentiate Normal X-Rays from Those with Pneumonia Present

As stated above, the goal of Phase I of the project is to create a model that will distinguish normal x-rays from those where pneumonia is present. Phase I was conducted in three parts:

1. To establish a baseline for model performance, the Python library Keras will be used to develop a small convolutional neural network (ConvNet) based on the framework suggested by Francois Chollet in his book "Deep Learning with Python" was executed.
2. Next, a more complicated model utilizing Data Augmentation and a dropout layer to the ConvNet was executed.
3. Finally, I incorporated Transfer Learning to the ConvNet.

1 - Building a Baseline ConvNet

The general structure of the ConvNet is four, alternated convolutional and pooling layers. We will begin with 150 x 150 images and end up with feature maps of size 7 x 7, while the depth of the feature maps goes from 32 to 128.

After the convolutional and pooling layers, the classification part of the ConvNet consists of a few fully connected layers. These fully connected layers can only accept one dimensional data. To convert the 3D data from the convolutional and pooling layers to 1D, a Flatten layer must be added.

Then once flattened I add a fully connected Dense layer of 512, followed by the final Dense layer of 2. This final Dense layer is where a probability is assigned to each classification, normal or pneumonia, occurs

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 36, 36, 64)	0
conv2d_3 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 17, 17, 128)	0
conv2d_4 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_4 (MaxPooling2	(None, 7, 7, 128)	0
flatten_1 (Flatten)	(None, 6272)	0
dense_1 (Dense)	(None, 512)	3211776
dense_2 (Dense)	(None, 2)	1026

Configuring the Model for Training

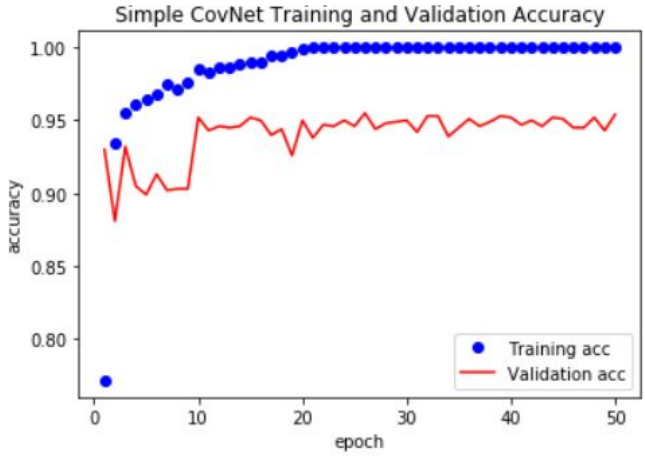
In compiling a model in Keras there are 3 primary parameters:

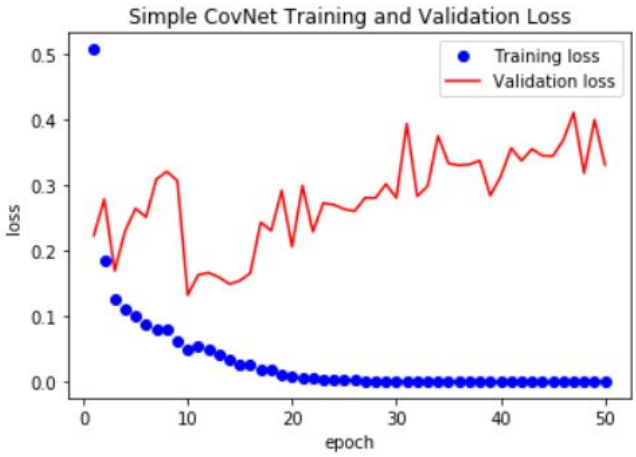
1. **Optimizer** - This controls the learning rate. In this model we will use **'adam'** as the optimizer. Adam is generally a good optimizer to use for many cases. The adam optimizer adjust the learning rate throughout training. Learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer.
2. **Loss Function** -I will use **'categorical_crossentropy'** for our loss function. This is the most common choice for classification. A lower score indicates that the model is performing better.
3. **Metrics** - To make things even easier to interpret, we will use the **'accuracy'** metric to see the accuracy score on the validation set when we train the model.

Fit the Model

When we call on Keras to fit the model, the model will iterate on the training data in mini batches of 32 samples, 50 times over. Each iteration over the training data is called an **'epoch**'. At each iteration the model will compute the gradients of the weights with regard to loss on the batch and update the weights accordingly. Accuracy will also be calculated for the batch.

At the same time we will be monitoring the loss and accuracy of the validation data set.



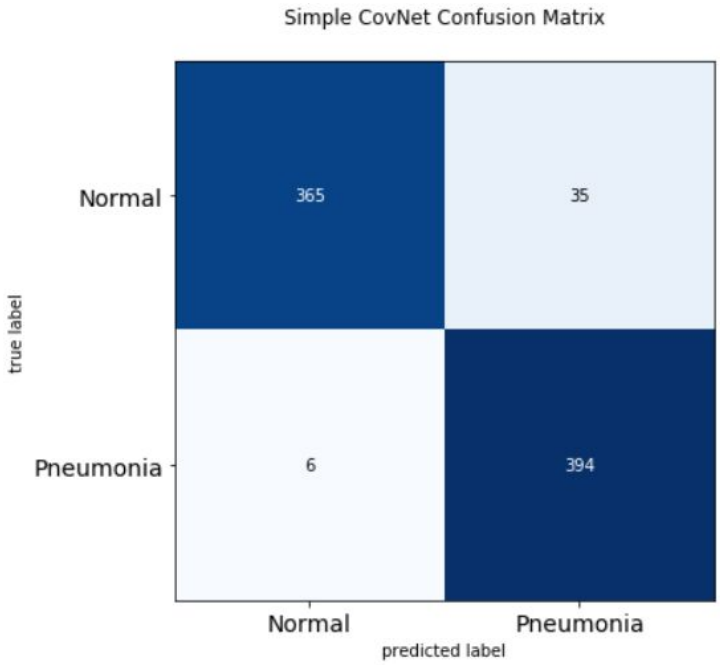


Here we can see the training loss decreases with every epoch, and the training accuracy increases with every epoch. This is what you would expect when using the adam optimizer - the quantity you try to minimize should be less with every iteration. But this is not the case for the validation loss and accuracy. They seem to peak around the 10th epoch.

The model is performing well on the data it is trained with and not so well on the data it has not seen before. This is an example of **overfitting**. After the 10th epoch, the model is over optimizing on the training data. This causes the model to learn representations that are specific to the training data and do not generalize to data outside of the training data set.

Given all the overfitting, it is still a reasonably good model. Validation accuracy is close to 95%, and validation loss below 0.4.

Confusion Matrix and Classification Report



	precision	recall	f1-score	support
0	0.98	0.91	0.95	400
1	0.92	0.98	0.95	400
micro avg	0.95	0.95	0.95	800
macro avg	0.95	0.95	0.95	800
weighted avg	0.95	0.95	0.95	800

Good Results for a first pass at the data. Good Recall with the true pneumonia radiographs, 98.5%. Or to put another way, 1.5 % who truly had pneumonia would be classified as normal with this model.

Also, a reasonably good Recall for normal radiographs, 91%. With this model 9% of children without pneumonia would get diagnosed with pneumonia. Worst case is one of these children would receive antibiotics.

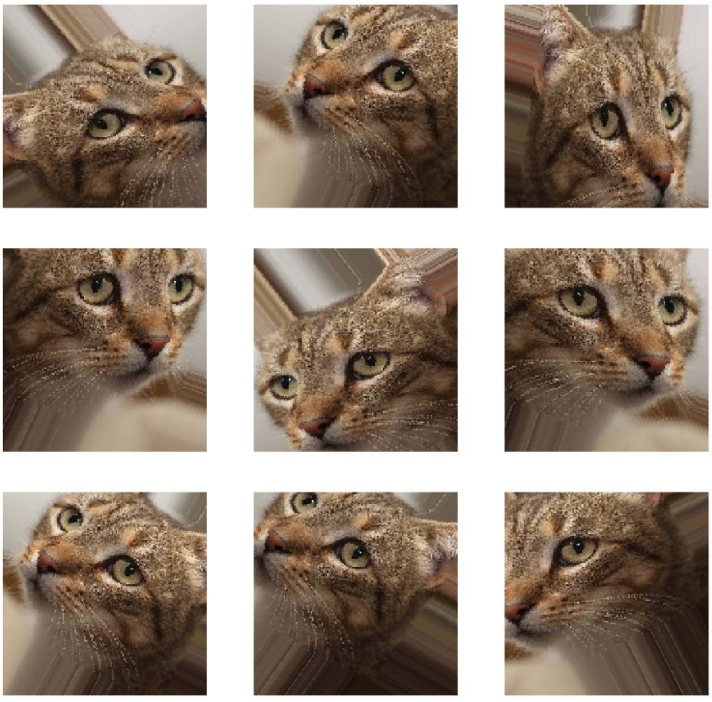
2 - ConvNet with Data Augmentation and a Dropout Layer

Overfitting was noted in the first model. Overfitting can be caused by having too few samples to learn from. This makes it very difficult to train a model that will generalize well to new data. One technique to combat overfitting that is used almost universally when processing images with deep learning models is **data augmentation**.

Data augmentation works by generating more training data from existing training samples, by augmenting the samples via a number of transformations that yield believable looking images. The premise is that during model training, the model will never see the exact same image twice. This helps expose the model to more aspects of the data and generalize better.

For example, if this is your original image.

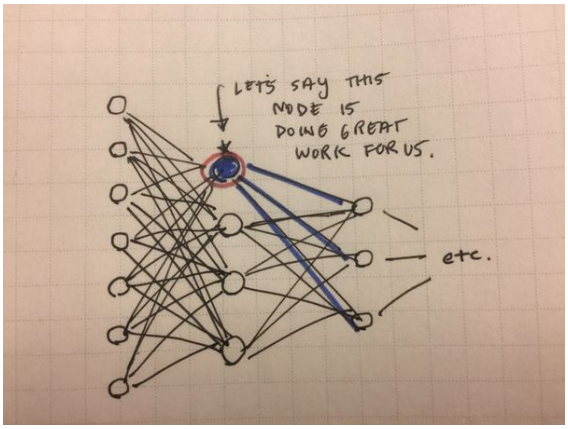
Using data augmentation we can generate the artificial training images below. These are new training instances. Applying transformations on the original image doesn't change the fact that this is the image of a cat.



The next method we will employ to reduce overfitting and develop a stronger overall model is a regularization technique called **dropout**, which randomly goes through and

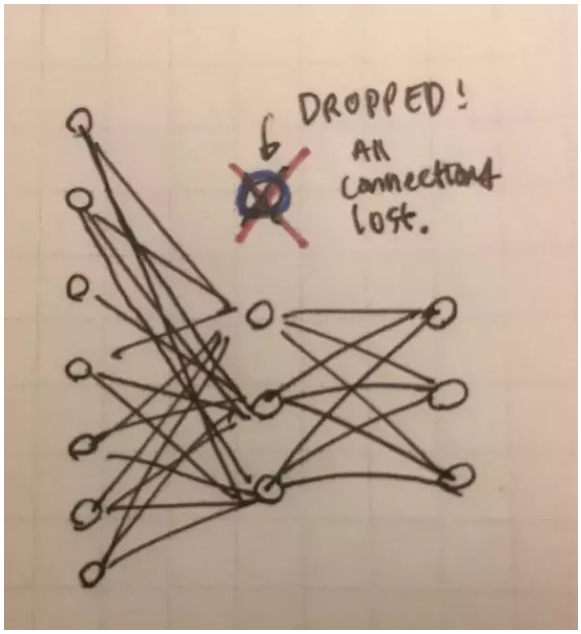
drops nodes. The reason why this works is because nodes are unequal in their explanatory power in the training set. The following explanation is borrowed from Jason Yum.

Imagine you have the following network, and the node highlighted (blue, red circle) is doing awesome work for us and sending a signal that helps us classify the end result well. This node is so great that when we run backpropagation, the neural network optimizes itself by giving this node a very high weight.



With dropout, we randomly turn off some of the nodes, and under some instances, we will "drop" this awesome node and force the model to train as if it wasn't there. When a node is dropped, all of its connections are turned off. This adds multiplicative noise to a neural net because we're multiplying by 1 or 0 (on or off).

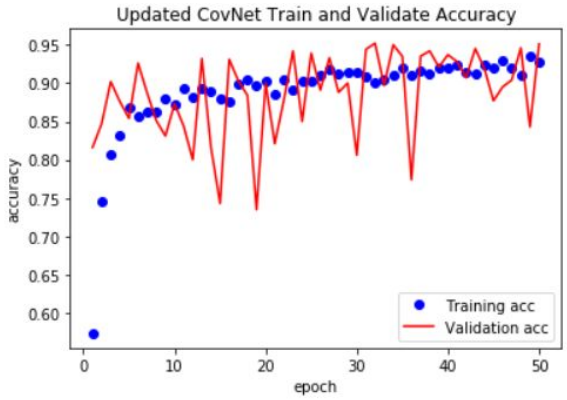
So under some instances you get a network that looks like:

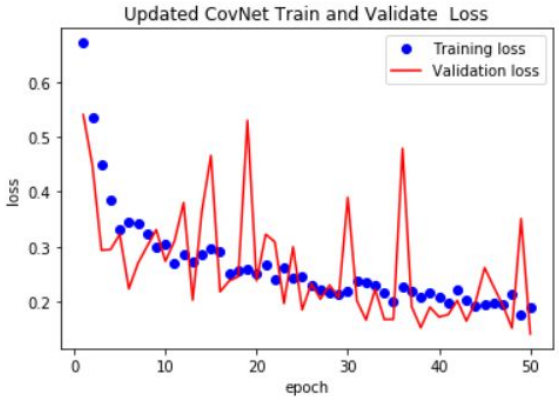


This forces the network to apply weights to the other three nodes, thus spreading the weights. This is what it means when people say that it forces the neural net to “not rely” on any specific node.

The idea is that we want the model to generalize beyond the training set, and we’ve already seen that we have an overfitting problem. Hence, we know that this awesome node can sometimes lead us astray when given new data.

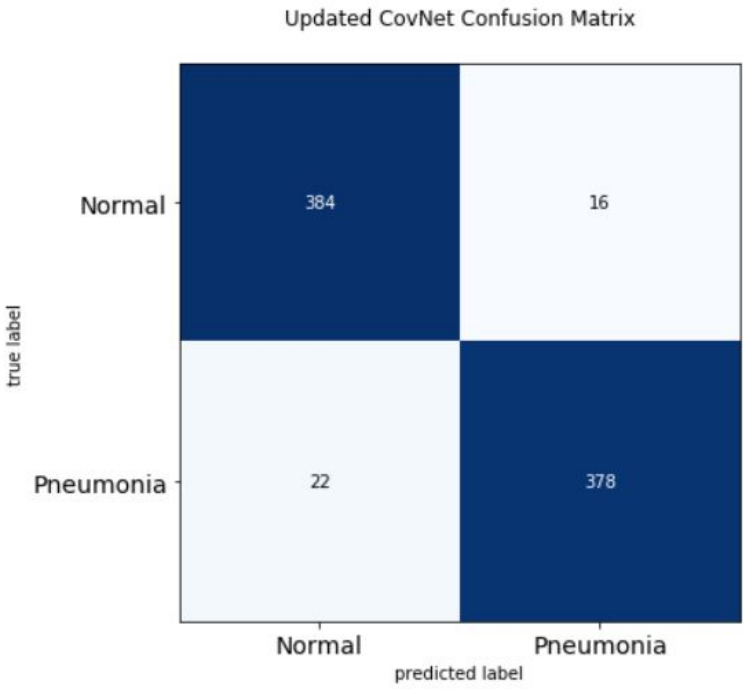
Re-running the previous model with data augmentation and a dropout layer:





Here we can see that we are no longer overfitting: the training curves are closely tracking the validation.

Confusion Matrix and Classification Report



	precision	recall	f1-score	support
0	0.95	0.96	0.95	400
1	0.96	0.94	0.95	400
micro avg	0.95	0.95	0.95	800
macro avg	0.95	0.95	0.95	800
weighted avg	0.95	0.95	0.95	800

Good Recall with the true pneumonia radiographs, 94.5%. However, this is 4% lower than the first ConvNet. With this model, 5.5% who truly had pneumonia would be classified as normal with this model.

There was also a sizable improvement in the Recall for normal radiographs, 96%. With this model, only 4% of children without pneumonia would get diagnosed with pneumonia. Worst case is one of these children would receive antibiotics.

3 - Transfer Learning ConvNet

We have demonstrated relatively good performance with our first models. In order to understand **Transfer Learning**, we must first look at what the different layers of a convolutional neural network are really learning.

When we train a deep convolutional neural network on a dataset of images, during the training process, the images are passed through the network by applying several filters on

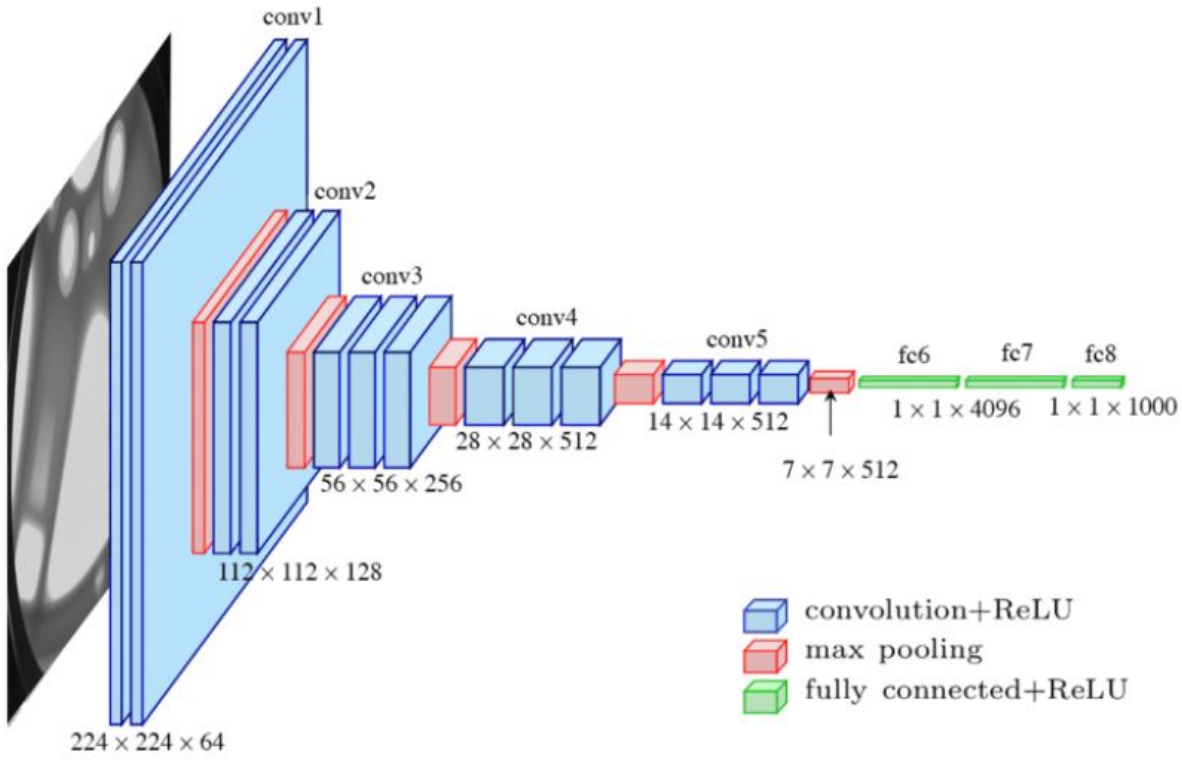
the images at each layer. The values of the filter matrices are multiplied with the activations of the image at each layer. The activations coming out of the final layer are used to find out which class the image belongs to.

When we train a deep network, the goal is to find the optimum values on each of these filter matrices so that when an image is propagated through the network, the output activations can be used to accurately find the class to which the image belongs. The process used to find these filter matrix values is gradient descent.

The reason why transfer learning works so well is that, we use a network which is pretrained on the ImageNet dataset and this network has already learned to recognize the trivial shapes and small parts of different objects in its initial layers.

By using a pretrained network to do transfer learning, we are simply adding a few dense layers at the end of the pretrained network and learning what combination of these features help in recognizing the objects in our new dataset.

The next step is to leverage the inputs from the VGG16 application that was trained on the ImageNet dataset (1.4 million labeled images in 1,000 different classes).



For our purposes we will use the learning from the convolutional base of the VGG16 model and rework the fully connected layer where classification will be occurring according to our model.

To instantiate the VGG16 ConvNet base we need to pass three arguments:

1. **weights** - This specifies the weight checkpoint from which to initialize the model. In our case the ImageNet dataset.
2. **include_top** - This refers to including the densely connected classifier on top of the network. We intend to use our own densely connected classifier (with only two classes: normal and pneumonia).
3. **input_shape** - The shape of the images that will be fed into the model.

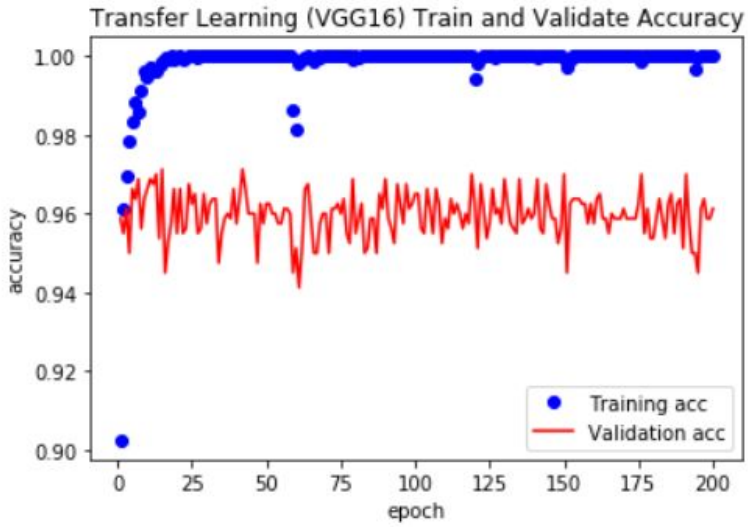
Convolutional Base Summary:

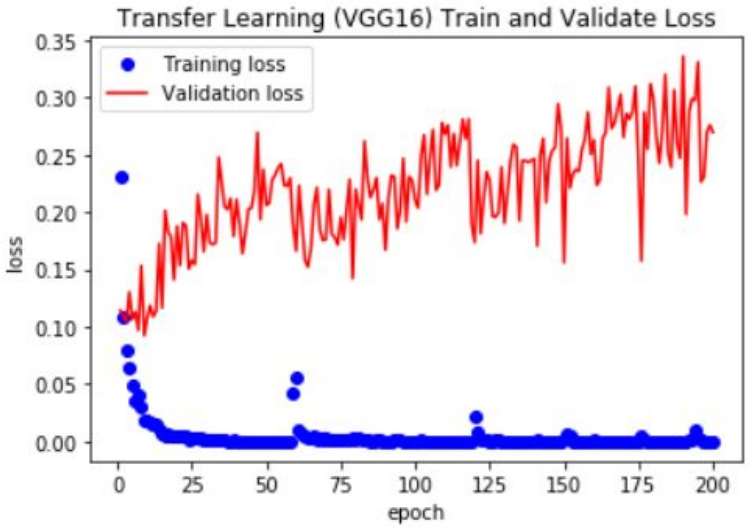
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

In order to use the VGG16 convolutional base we must first reshape out training and testing features to 7 x 7 x 512 before running through the fully connected layer of our model.

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 256)	6422784
dropout_2 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 2)	514

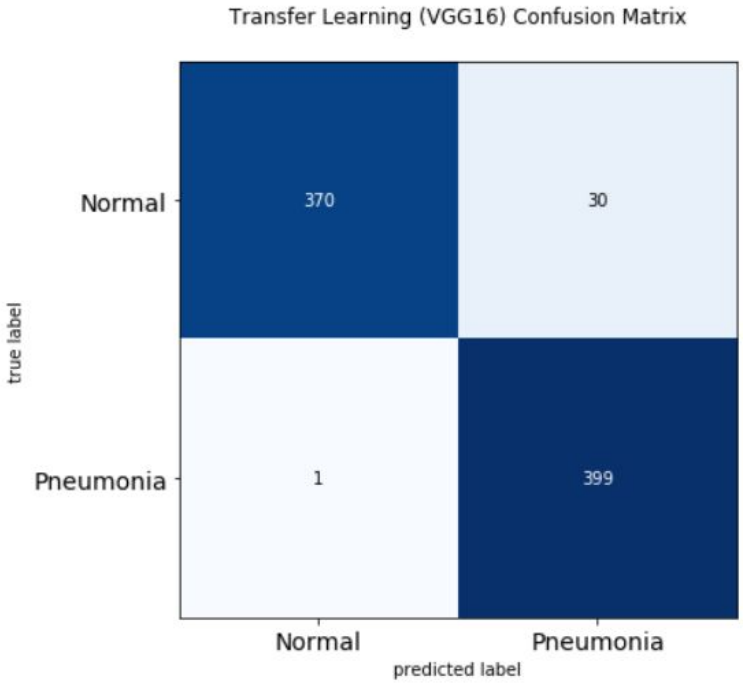
Now we will fit the transfer learning model using 200 epoch's instead of the 50 used in the previous models in an attempt to see we can make some additional gains in Recall.





Here we see a very good model, validation accuracy ~96% and validation loss below 0.3.

Transfer Learning Confusion Matrix and Classification Report



	precision	recall	f1-score	support
0	1.00	0.93	0.96	400
1	0.93	1.00	0.96	400
micro avg	0.96	0.96	0.96	800
macro avg	0.96	0.96	0.96	800
weighted avg	0.96	0.96	0.96	800

Excellent Recall with the true pneumonia radiographs, 99.8%. With this model, only 0.2% who truly had pneumonia would be classified as normal with this model.

There was a small loss in the Recall for normal radiographs, 92.5%. With this model, 7.5% of children without pneumonia would get diagnosed with pneumonia if chest x-rays were the only available modality for diagnosis.

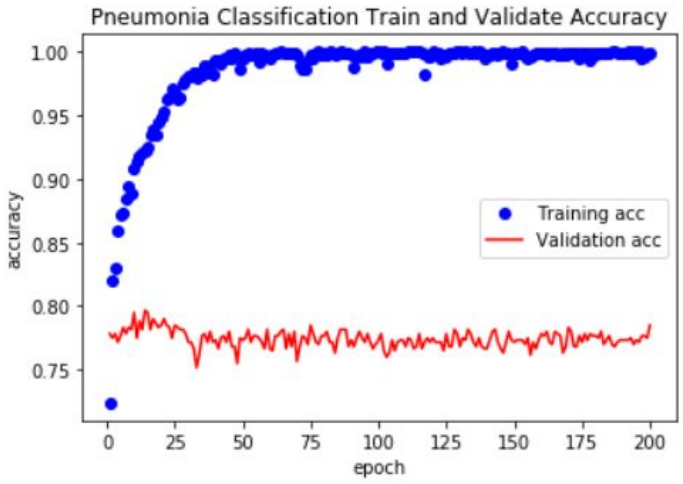
The resulting high-accuracy models suggests that this AI system has the potential to effectively learn from increasingly complicated images with a high degree of generalization using a relatively small repository of data.

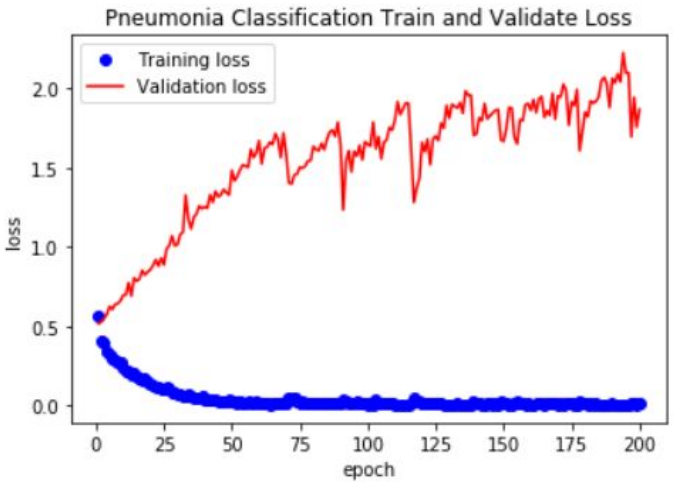
Machine Learning Phase 2 - Distinguish Bacterial Pneumonia from Viral Pneumonia

So far we have created a ConvNet that performs very well when distinguishing a normal x-ray and one where pneumonia is present. The challenge now is to differentiate between bacterial and viral pneumonia.

Again, Recall is the primary scoring method because it considers False Negatives. A False Negative in this model would mean we predict a child has viral pneumonia when in fact they have bacterial pneumonia. The consequences of misdiagnosing (False Negative) bacterial pneumonia can be life threatening, We strive for a model whose Recall is as close to 100% as possible.

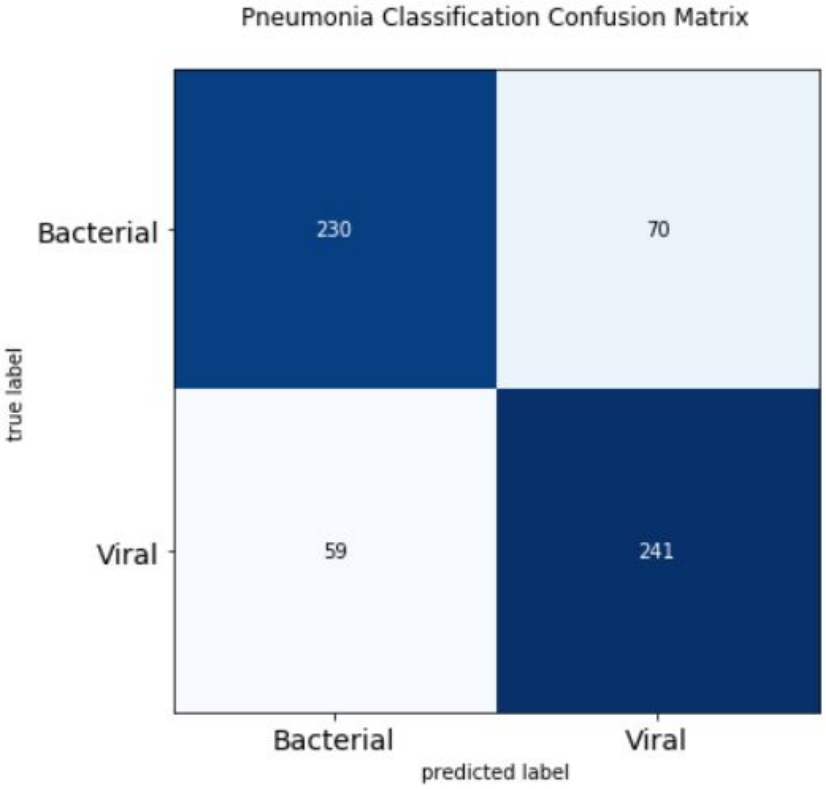
Because we achieved the best results with the VGG16 transfer learning model, we will use it again here. The only difference is that we will be using bacterial and viral pneumonia images, and no normal images.





Overall you can see the model does not perform as well predicting between the two types of pneumonia. The accuracy slips from 96.1% in the previous model where we were comparing normal x-ray to those with pneumonia (both viral and bacterial), to 78.5%.

Transfer Learning - Bacterial vs. Viral Pneumonia Confusion Matrix and Classification Report



	precision	recall	f1-score	support
0	0.80	0.77	0.78	300
1	0.77	0.80	0.79	300
micro avg	0.79	0.79	0.79	600
macro avg	0.79	0.79	0.78	600
weighted avg	0.79	0.79	0.78	600

Due to the potential consequences of misdiagnosing bacterial pneumonia, our primary measure of interest is Bacterial (0) Recall. Marginal Recall with the bacterial pneumonia radiographs,76.7%. With this model, 23.3% who truly had bacterial pneumonia would be classified as having viral pneumonia with this model.

The Recall for viral pneumonia radiographs, 80.3%. With this model, 19.7% of children without viral pneumonia would get diagnosed with bacterial pneumonia if chest x-rays

were the only available modality for diagnosis. This is not a potentially life threatening outcome. Worse case, a child is treated with antibiotics who may not really need them

Discussion

In the original study "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning", Both models: Normal vs Pneumonia, and Bacterial Pneumonia vs Viral Pneumonia were adapted using the Inception-v3 architecture pre-trained on the ImageNet dataset. The models were trained on an Ubuntu 16.04 computer with 2 Intel Xeon CPUs, using a NVIDIA GTX 1080 8Gb GPU for training and testing, with 256Gb available in RAM memory.

The original study considered 3 metrics. Recall (Sensitivity), Accuracy, and Specificity.

- **Recall (Sensitivity)** is the probability that a test will indicate 'disease' among those with the disease. Or, **tp / (tp + fn)**
- **Accuracy** is the fraction of correctly identified images to the total number of images. Or, **(tp + tn) / (tp + tn + fp + fn)**
- **Specificity** is the fraction of those without disease who will have a negative test result. Or, **tn / (tn + fn)**

The table below summarizes the comparison between the original research model and the three ConvNet's developed in this project when comparing normal radiographs to those with pneumonia:

Normal vs. Pneumonia Comparison

	Metric		
Model	Recall	Accuracy	Specificity
Original Study (TL Inception-v3 ConvNet)	93.2%	92.8%	90.1%
Simple ConvNet	98.5%	94.9%	91.3%
Data Augmentation and Dropout ConvNet	94.5%	95.3%	96.0%
Transfer Learning VGG16 ConvNet	99.8%	96.1%	92.5%

The items highlighted in green represent the model with the highest score for Recall, Accuracy, and Specificity. For Part I of this project, create a model that will distinguish normal radiographs from those with pneumonia present, all three ConvNet's developed outperformed the original study in Recall, Accuracy, and Specificity.

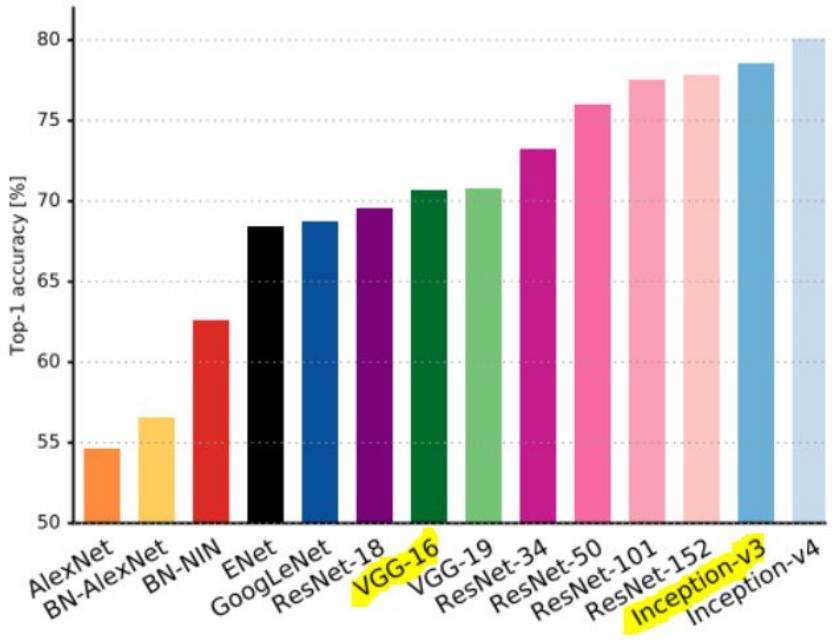
Additionally, these models were trained with a relatively small number of images and achieved very good performance.

In Phase II of this project our goal was to develop a model to differentiate bacterial from viral pneumonia. We utilized the same transfer learning model, VGG16, used in Phase 1, using only images with either viral or bacterial pneumonia

Bacterial vs Viral Comparison

Model	Metric		
	Recall	Accuracy	Specificity
Original Study (TL Inception-v3 ConvNet)	88.6%	90.7%	90.9%
Transfer Learning VGG16 ConvNet	76.7%	78.5%	80.3%

Here our results were not as impressive, nor as good as the original study. A couple of factors may explain the difference. The sheer computing power used in the original study was considerably higher than my laptop with a single i7 processor. Also, I am but one individual with a limited amount of time, whereas they probably had a team of researchers and grad students. They also used a different transfer learning model, Inception-v3.



Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
ResNeXt50	96 MB	0.777	0.938	25,097,128	-
ResNeXt101	170 MB	0.787	0.943	44,315,560	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset. As you can see above Inception-v3 outperforms VGG16. In a nutshell, the VGG model is larger than Inception with more parameters and a more shallow depth.

Chest X-rays present a difficult classification task due to the relatively large amount of variable objects, specifically the imaged areas outside the lungs that are irrelevant to the diagnosis of pneumonia.

Several studies have suggested that bacterial pneumonia cannot be differentiated from non-bacterial pneumonia on the basis of the chest radiograph. Virkki R, Juven T, Rikalainen H, et al Differentiation of bacterial and viral pneumonia in children Thorax 2002;57:438-441. One of the more interesting findings of this study was that in 30% of cases there was evidence of a mixed viral/bacterial infection. Finally, the last sentence in the discussion section of the paper the researchers state ***"It is evident that all children with radiologically confirmed pneumonia should be treated with antibiotics because, in clinical practice, it is virtually impossible to distinguish exclusively between viral pneumonia and bacterial pneumonia."***

Conclusion

The purpose of this project is to determine if an AI model can be developed to assist in the diagnosis of pneumonia using common radiographs (chest x-rays).

The findings of this report affirm the stated purpose of the project. In differentiating normal radiographs from those where pneumonia was present, our very first model showed excellent results with a Recall of 98.5%!

Our final model which utilized transfer learning, using the pre-trained VGG16 convolutional base, achieved a Recall of 99.8%. In other words, only 1 of the 400 x-rays where pneumonia was present was classified as being normal. This is exceptional!

For Phase II of the project, differentiating between bacterial and viral pneumonia, the results were not as impressive with a Recall of 76.7%. As stated in the Discussion section of this report there are several reasons why the reduced Recall compared to Phase I.

In order to achieve better model performance when classifying viral vs. bacterial pneumonia, the following suggestions are recommended:

- Increase the number of images to train the model. More images make models that predict more accurately.
- Come up with a way to systematically crop the portions of the x-ray that have nothing to do with pneumonia diagnosis. This way the model could focus on only the most important features.

- And finally, the images used in this project may have some inaccuracies. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert. X-rays were the only modality used to classify the images. In the study referenced above, “in 30% of cases there was evidence of a mixed viral/bacterial infection.”. Image classification needs to be based and confirmed using multiple tests, not just x-rays.