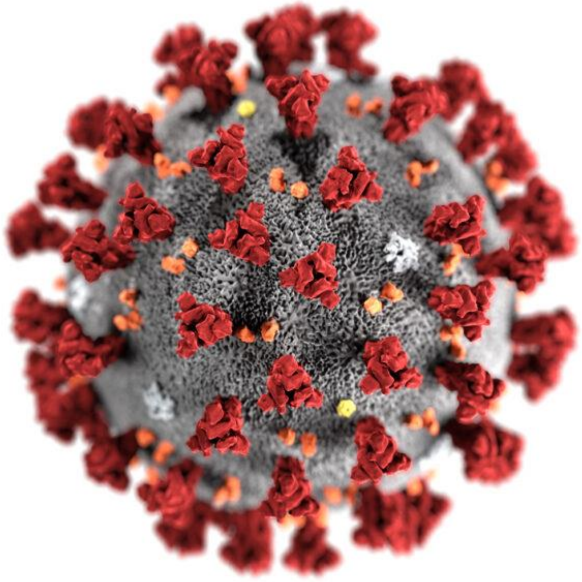
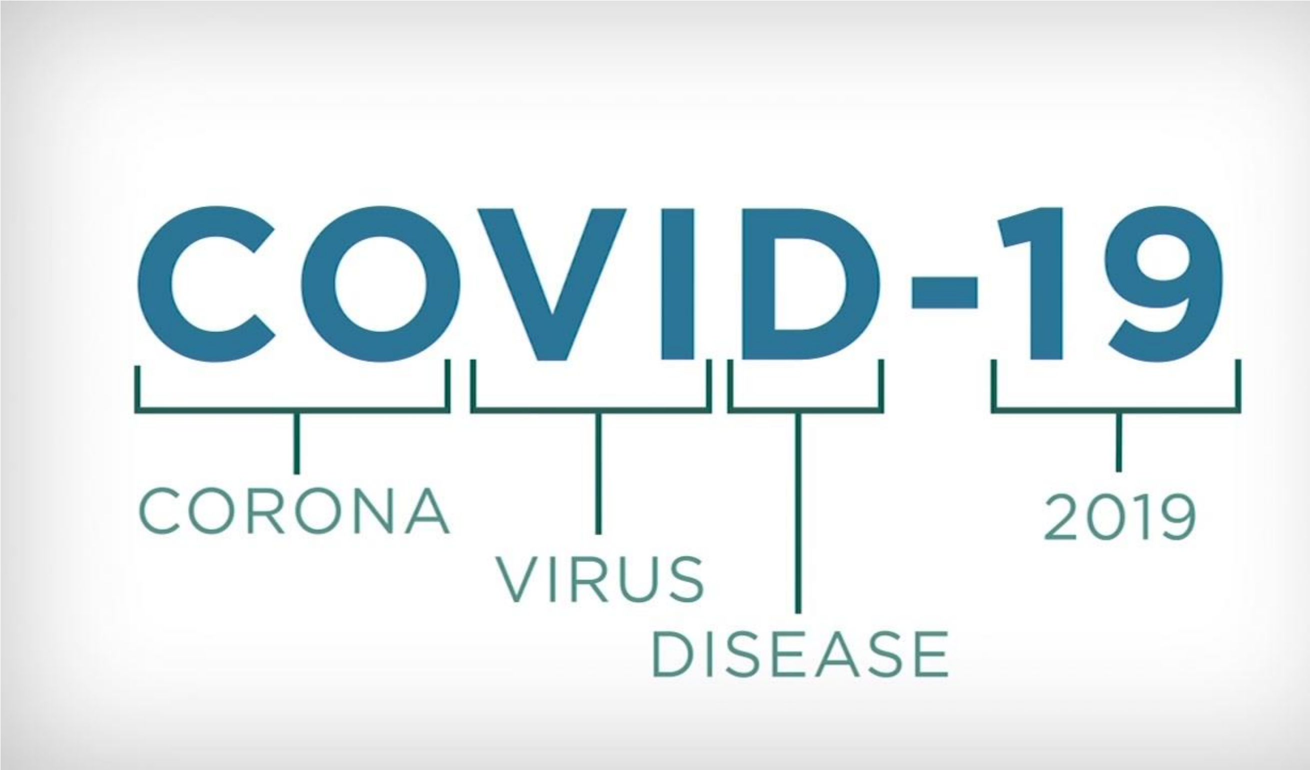
COVID 19 X-Rays

Jesse Mckinzie and Tyler Johnson



What is Covid-19?



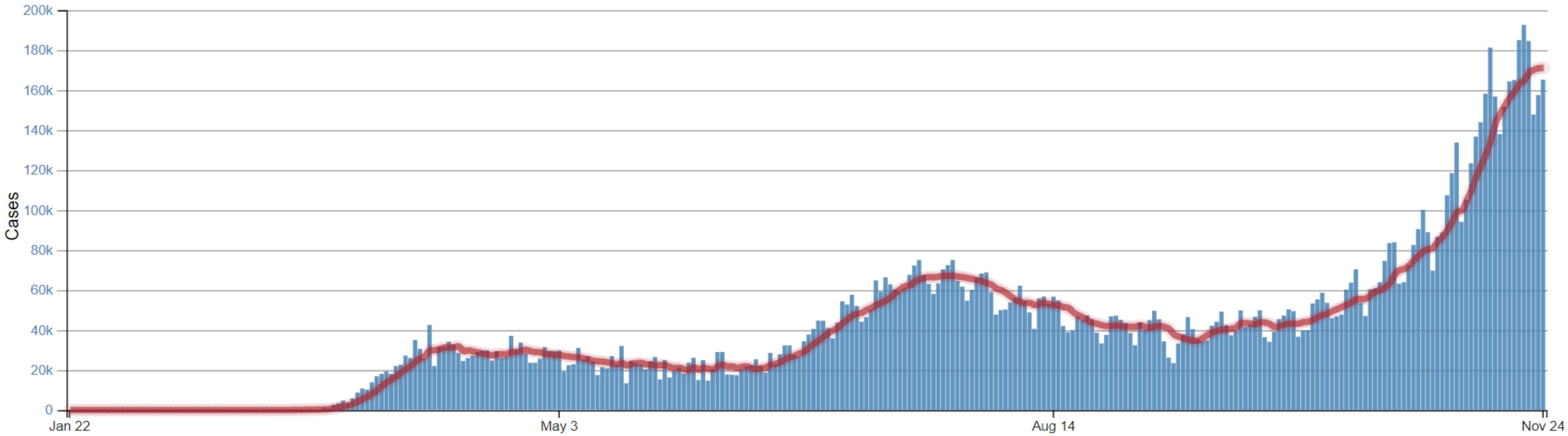
https://abc7news.com/how-did-coronavirus-get-its-name-of-for-new/6071514/

- Covid-19 is caused by what is known as a Coronavirus
  - Severe Acute Respiratory Syndrome (SARS)*
  - Middle East Respiratory Symptom (MERS)
- Covid-19 is caused by the virus SARS-CoV-2
- World Health Organization declared Covid-19 a pandemic in March 2020.

What is Covid-19?

- In the United States, as well as around the world, the theorized "Second Surge" seems to be occurring.

Daily Trends in Number of COVID-19 Cases in the United States Reported to CDC

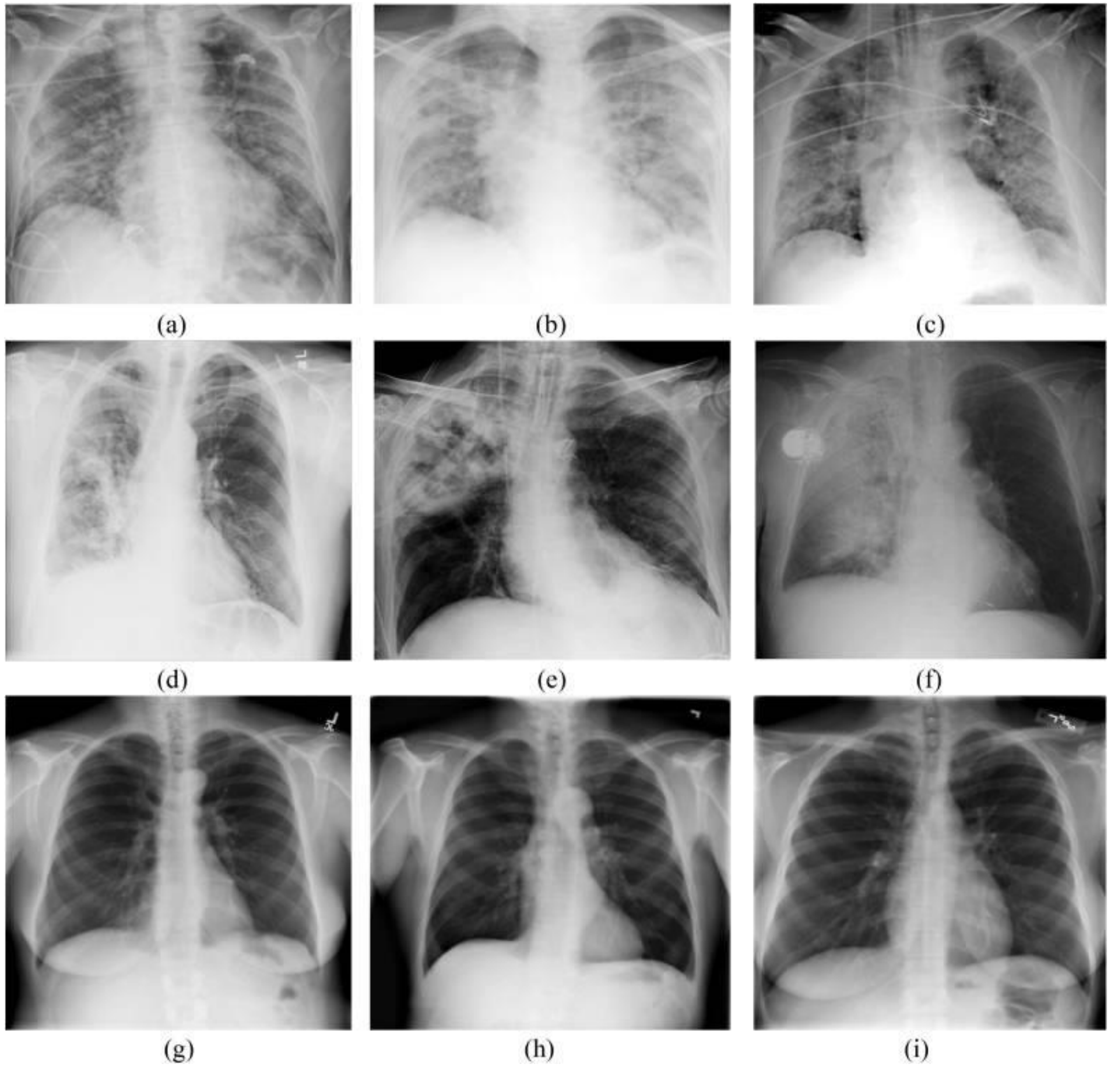


https://covid.cdc.gov/covid-data-tracker/#cases_casesper100klast7days

Using X-Rays to Identify Cases

- Physicians began using X-Rays to try to determine severity of Covid-19 cases around May.
- This evolved into attempting to use X-Rays to diagnose.
  - Radiologists are in short order.
- This inevitably led to Machine-Learning based classification problems.
  - Some incorrect classifications of other respiratory diseases.

Using X-Rays to Identify Cases



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7448820/#b0165

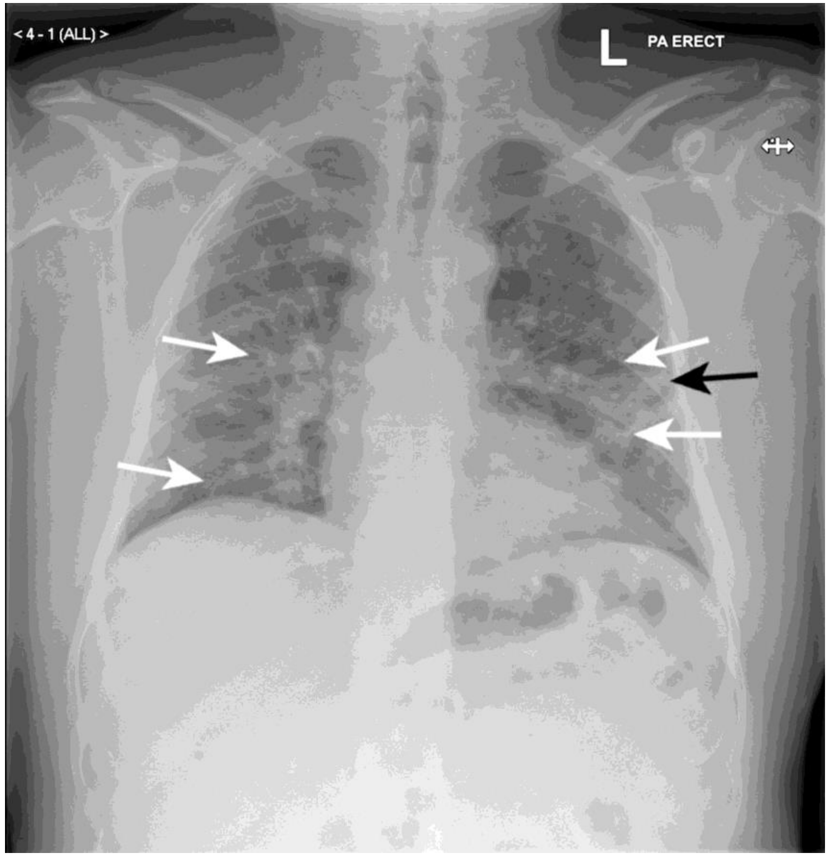
Our Goal

- We want to expand the previous models using more data.
  - Most previous applications used ~ 2000 images.
- We believe that more training data will allow for higher accuracy, even in the presence of fewer layers.
- Found a paper that did a similar approach
  - Convolutional neural network
  - 89% accurate
    - Included Pneumonia as a category
  - ~1200 x-rays

Dataset

- Initial dataset:
  - 219 COVID-19 positive x-rays
  - 1341 normal x-rays
  - 1345 pneumonia x-rays

- Final dataset:
  - 277 COVID-19 positive x-rays
  - 2,921 normal x-rays
  - 5,617 pneumonia x-rays
  - Total of 8,815 x-rays (2.5 GB of data)



https://www.bmj.com/content/370/bmj.m2426

Training and test data

- Used train_test_split() from sklearn to give a fair split between the three categories for the train and test data
  - Used a 75%-25% perecent split for the training and test data, respectively
  - Random split of 42

- Train size: 6,611 x-rays
- Test size: 2,204 x-rays

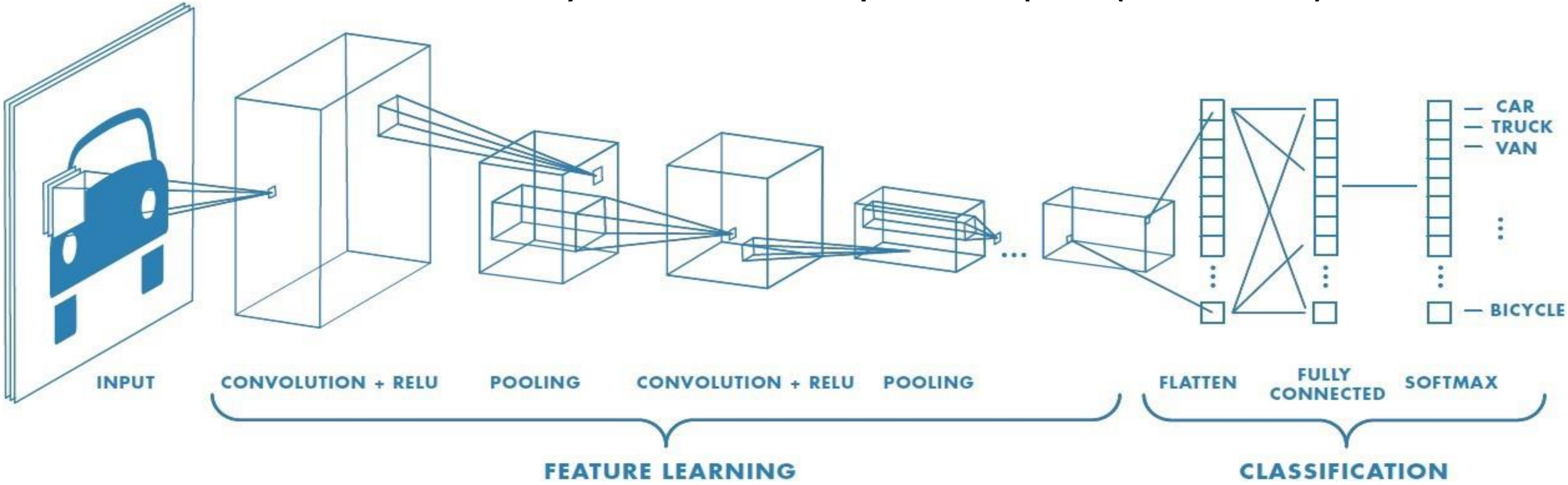
Machine Learning options

- Convolutional neural network
  - Work well for image classification and medical image processing
  - Has been used to identify abnormal x-rays in the past (Nov. 2018)

- Clustering was another option but CNN seemed the most popular

Our Approach

- Convolutional Neural Network (CNN)
  - Work well for image classification and medical image processing.
  - Has been used to identify abnormal x-rays in the past (Nov. 2018)



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Tools

- Decided to implement our CNN in Python due to the convenient libraries for our project such as:
  - Pandas
  - Keras
  - TensorFlow
  - sklearn
  - matplotlib

Accessing x-rays

- Stored names of images in a data frame and flowed in images using a ImageDataGenerator object and flow_from_dataframe method
  - Allowed us to access images in batches
  - No need to store all images in memory simultaneously

- Dataset could fit in memory in current state
  - Want our code to be scalable in the case of more data

Initial CNN on smaller data set

- Began testing on the smaller data set with 2,905 x-rays
  - Accuracy of about 98.5% (similar paper had 98.3%)

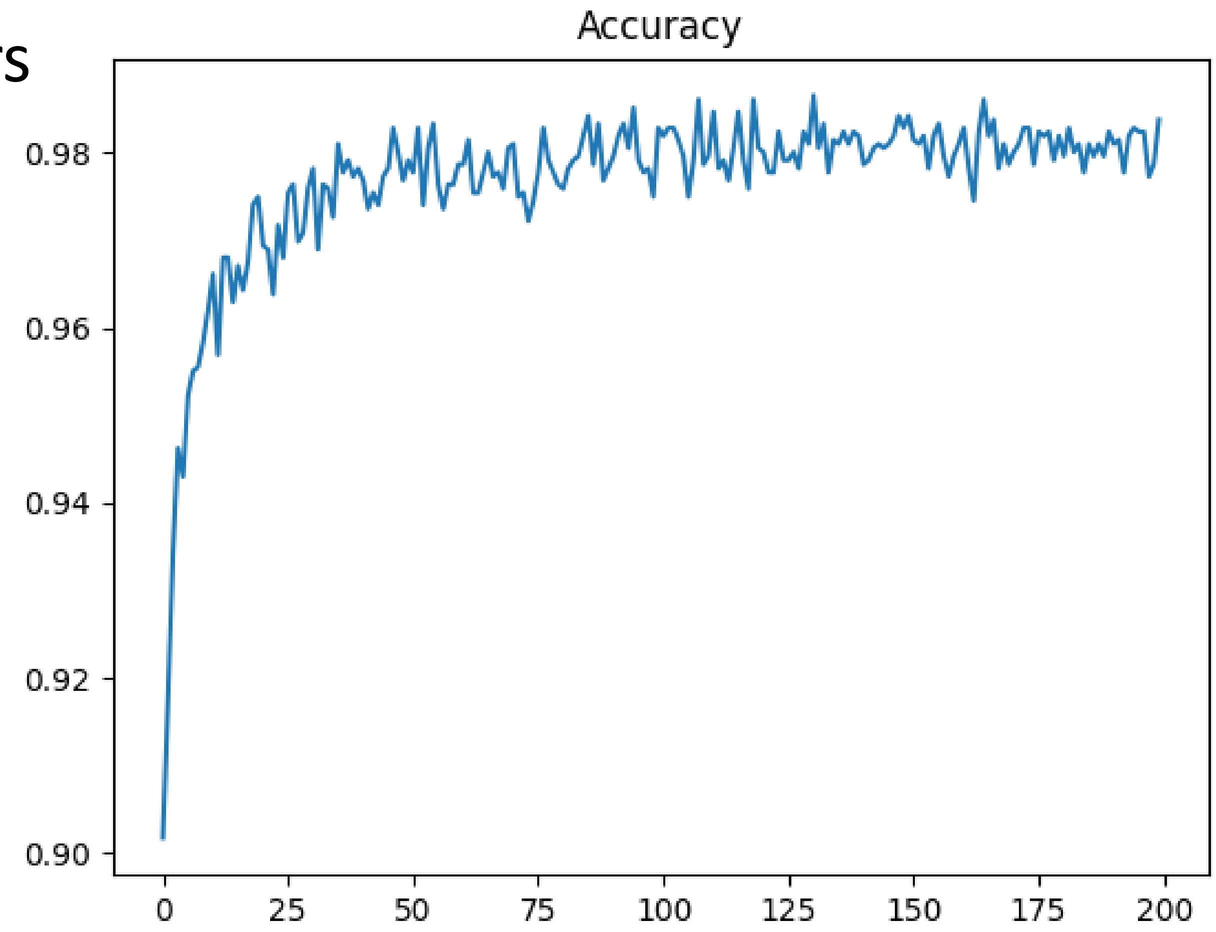
- Architecture:
  - Two convolutional layers
  - Two max pooling layers
  - Two dense layers

Analysis of parameters on test CNN

- Tried different number of filters
  - Recommended to increase number of filters on each layer to pick up fine details, i.e. 64 filters on first, 128 on second, etc.
  - Did not work as well as having 64 filters on every layers
    - Accuracy went down
- Tried different number of layers – worked the best
  - 4 convolutional
  - 4 max pooling
  - 2 dense

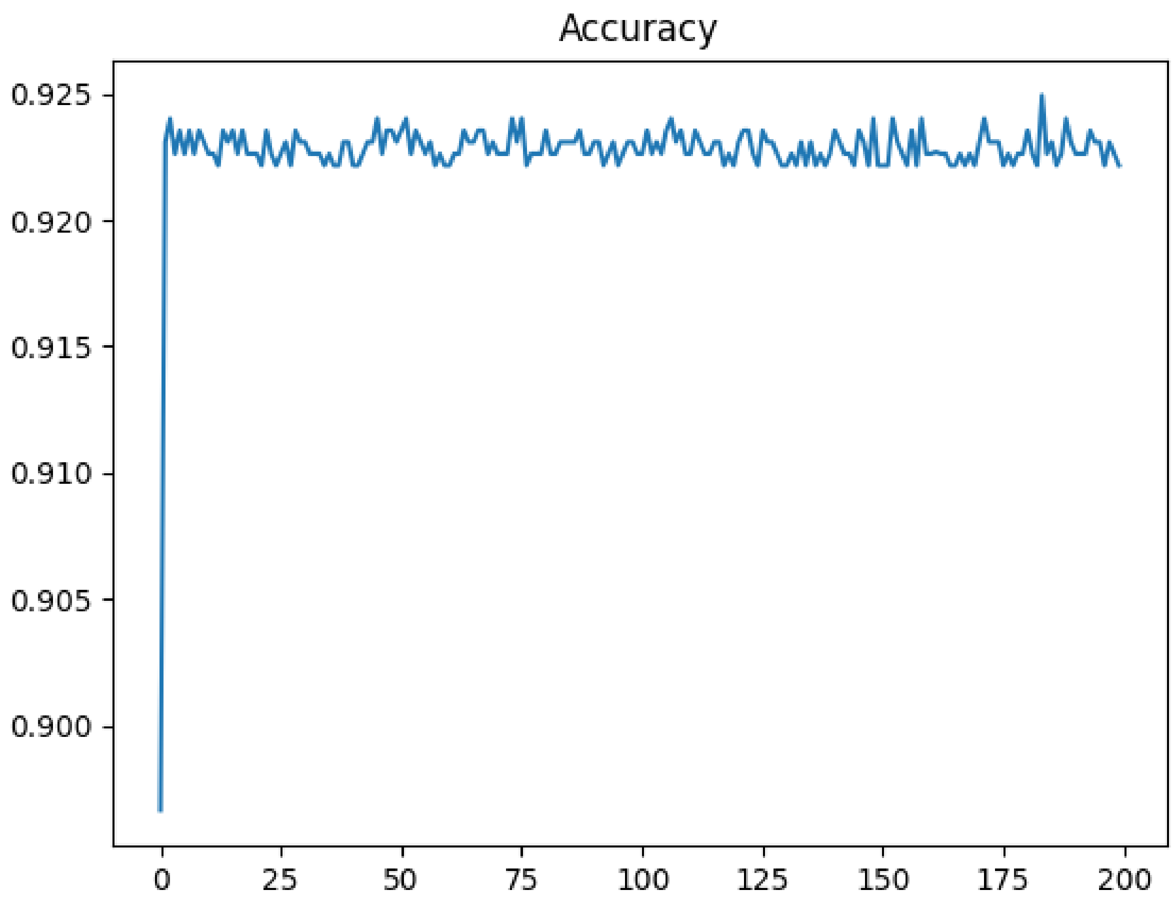
More filters

- Used 64 filters then 128 filters



More layers and more filters

- Used increased number of filters with 64,128,256,512 filters

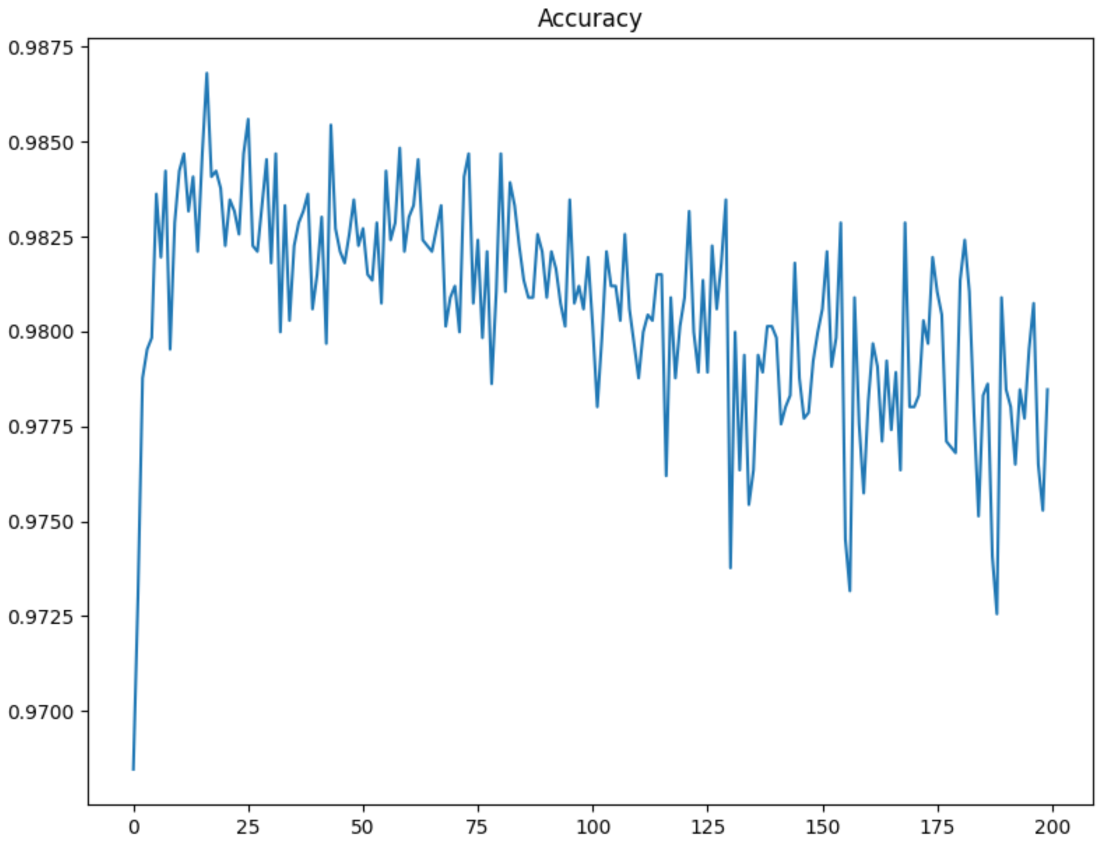


Final CNN Architecture

- Four convolutional layers:
  - 64 filters each
  - Stride of (3, 3)
  - relu activation function
- Four max pooling layers:
  - 64 filters each
  - Stride of (3, 3)
- Two dense layers:
  - First with 64 filters and relu activation
  - Second with 2 filters and softmax activation

Number of epochs

- We choose the number of epochs right before accuracy decreases to avoid overfitting the data



Optimizer

- Final choices came down to:
  - Gradient decent
  - Stochastic gradient decent
  - RMSprop (Root Mean Square Propagation)
- Same accuracy from all optimizing functions but RMSprop had the smallest lost and was the quickest algorithm
- RMSprop also does not need to optimize batch size (run time) against accuracy

RMSprop

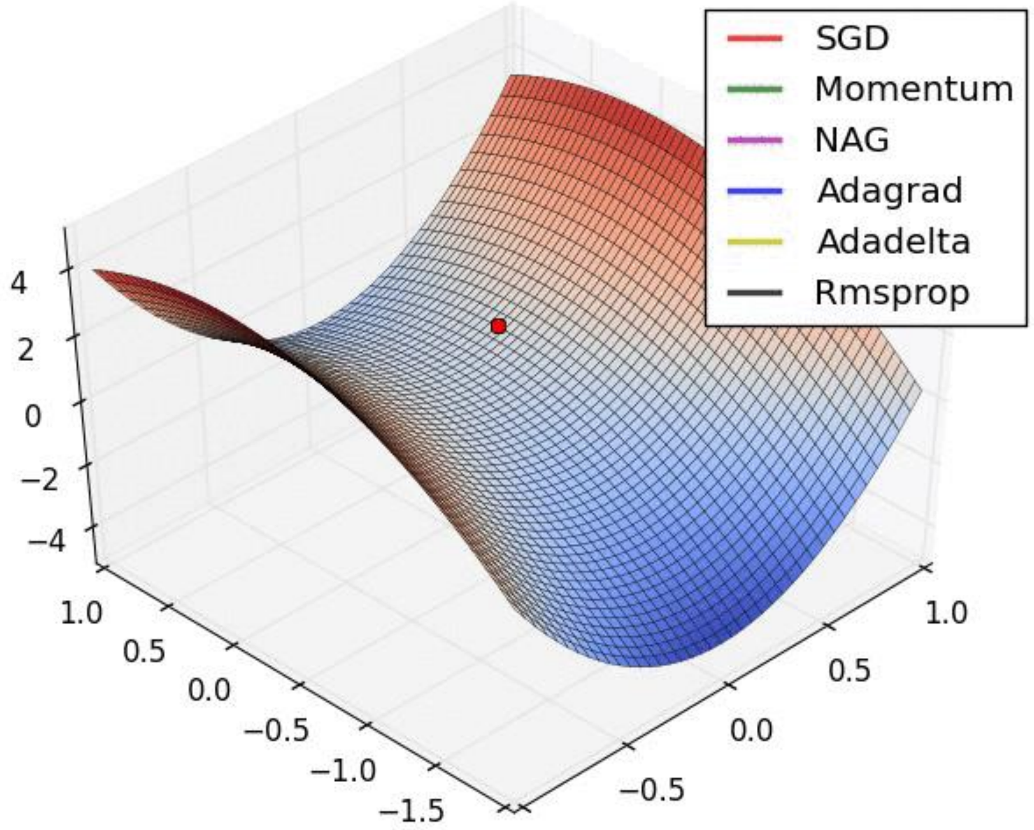
$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\delta C}{\delta w}\right)^2$$

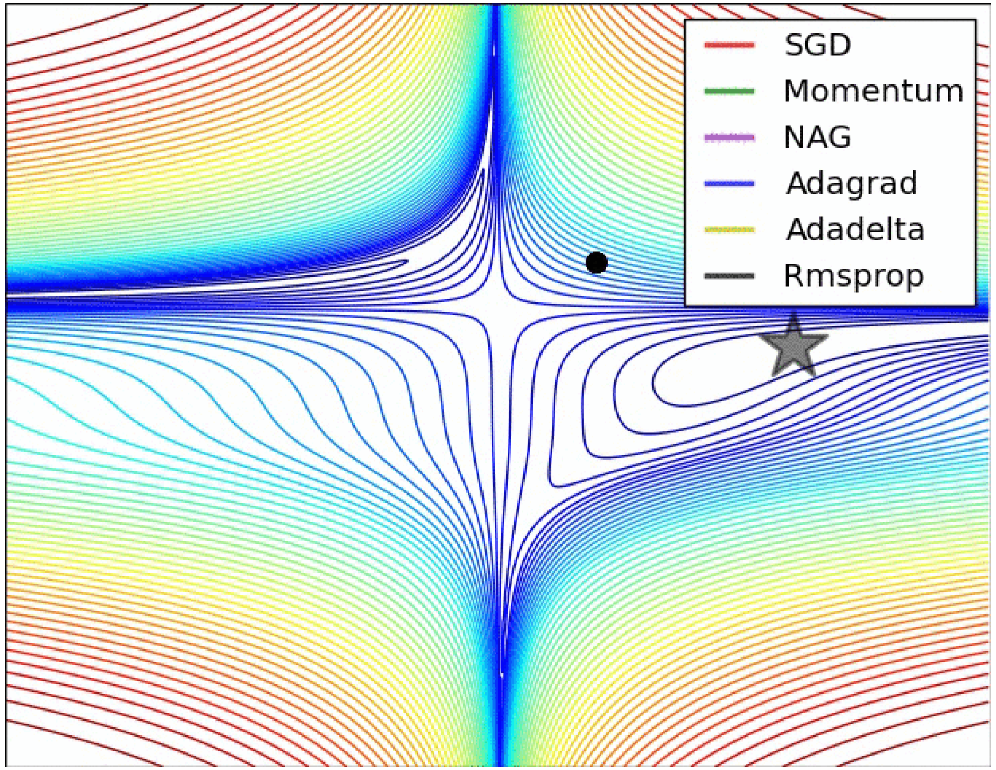
$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}}\frac{\delta C}{\delta w}$$

- $E[g^2]_t$ is the moving average of squared gradients
- $\left(\frac{\delta C}{\delta w}\right)$ is the gradient of the loss function with respect to the weights
- η is the learning rate
- β is the moving average – default value is 0.9

https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a

RMSprop





https://imgur.com/a/Hqolp#NKsFHJb

Loss Function

- Chose categorical cross entropy
  - Best function for classification techniques
  - Trained model gives a percent confidence of classification which is useful for the applications of our project
- Cross entropy

$$CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

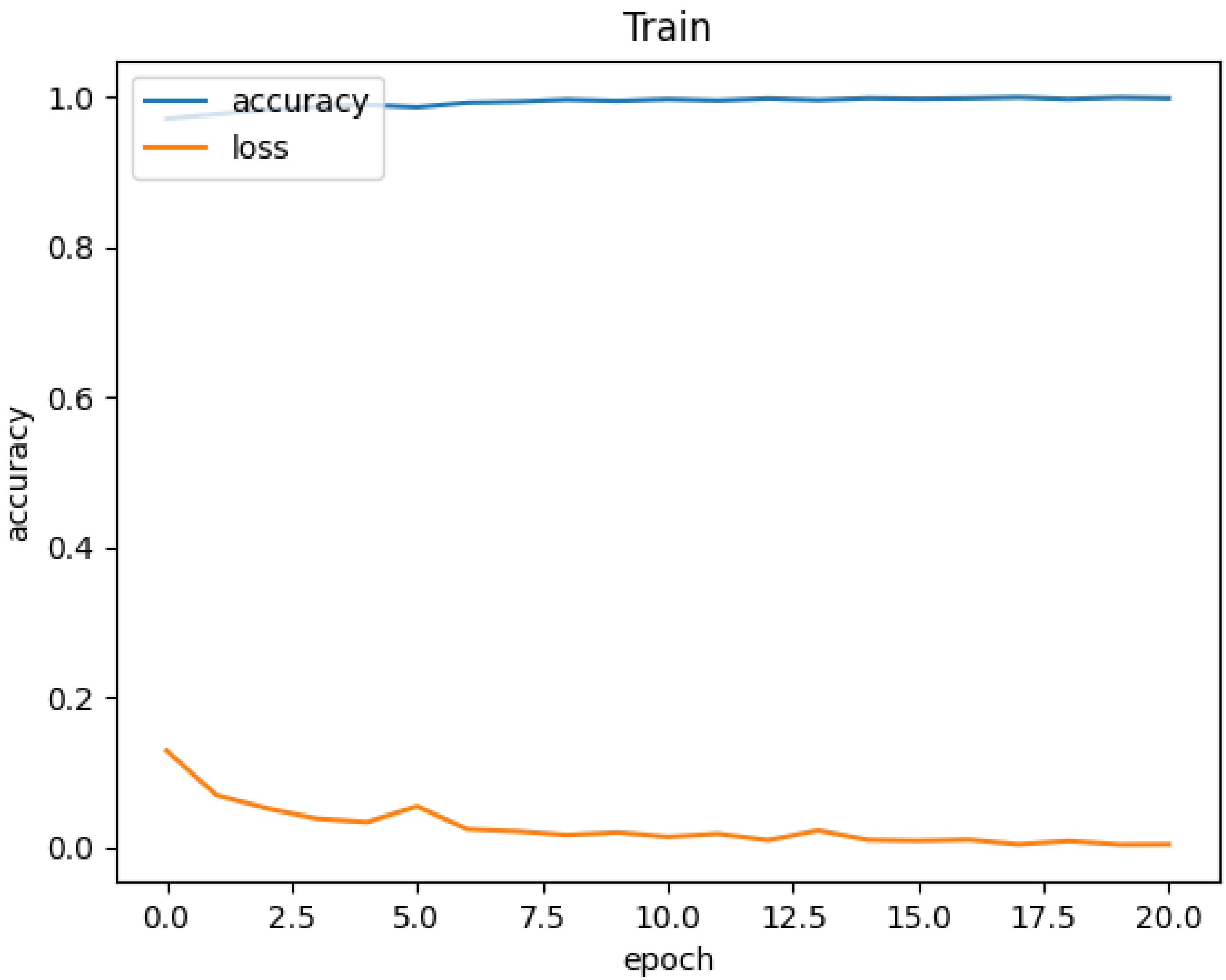
  - CE is cross entropy
  - s_j is the ground truth score for $j \in \{negative, positive\}$

https://gombru.github.io/2018/05/23/cross_entropy_loss/

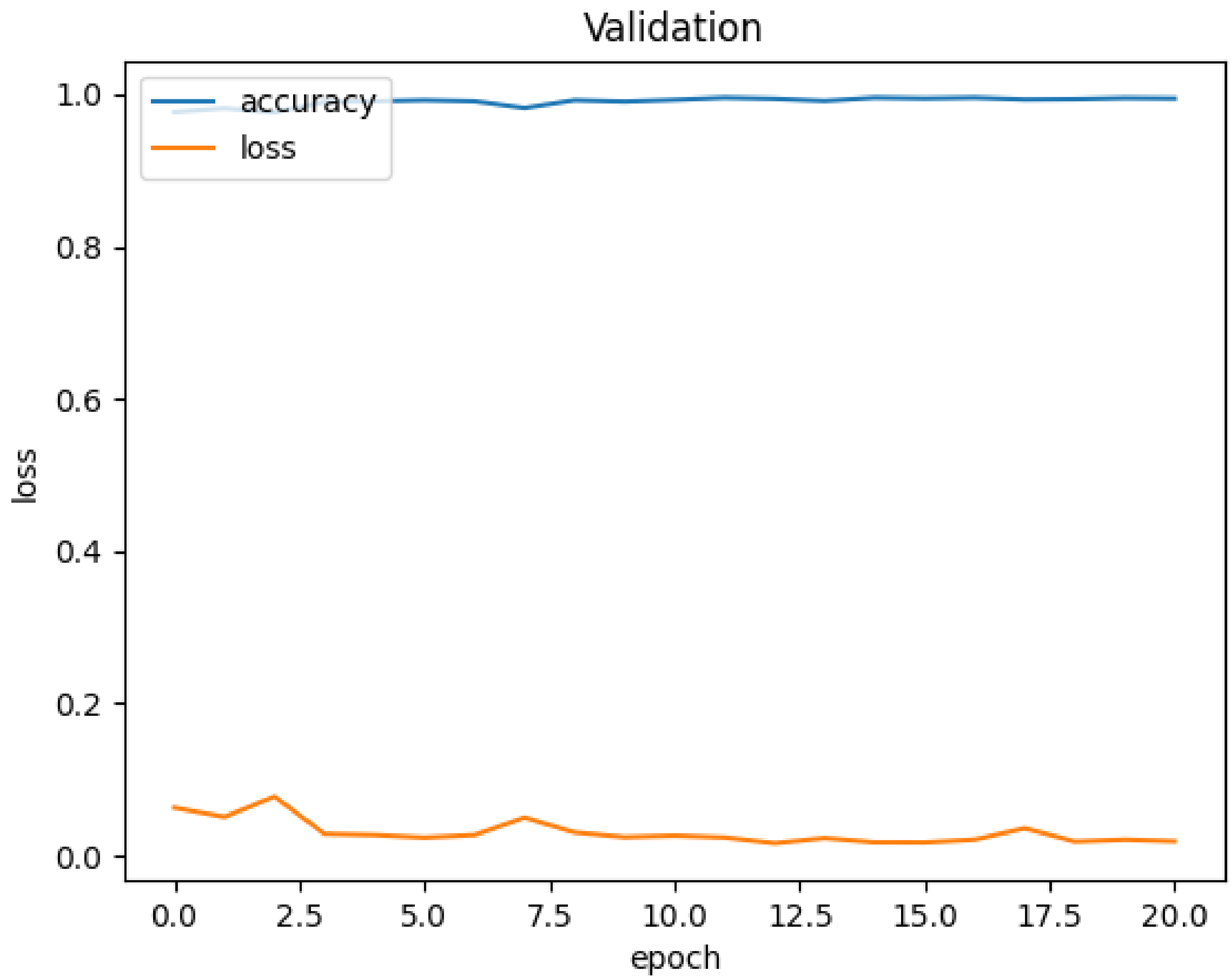
Results

- We ended with an accuracy of 99.55% with a loss of 0.032 on the validation set
  - Batch size of 50
  - 21 epochs (how many times the CNN saw the dataset)
- A paper using the same dataset as our small dataset using a convolutional neural network
  - 98.3% accuracy for binary classification
  - 2,905 x-rays

Training results



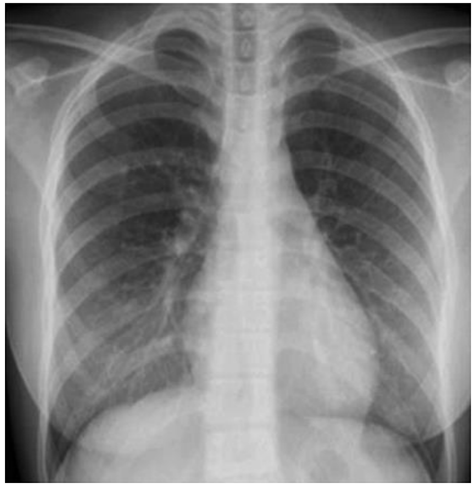
Validation results



Results of a couple unseen images

- Input two images without COVID-19 into trained mode
  - Both had >99% confidence of being COVID-19 negative
- Input two images with COVID-19 into trained model
  - Both had >99% confidence of being COVID-19 positive

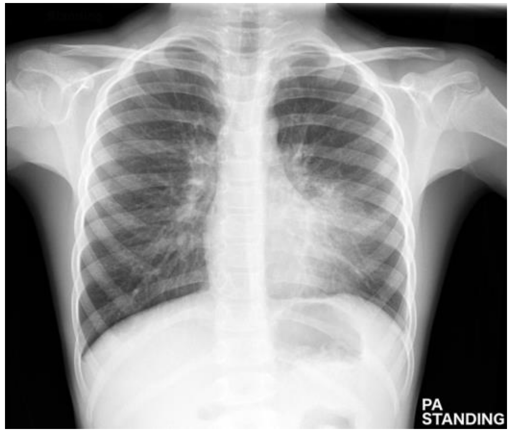
Normal



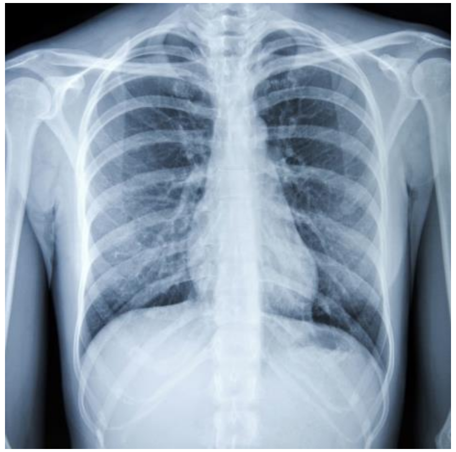
COVID-19 positive

Results of a couple unseen images

- Input an x-ray with pneumonia into model
  - Had >99% confidence of being COVID-19 negative
- Input have an image with lung cancer
  - Had >99% confidence of being COVID-19 positive



pneumonia



Lung cancer

Hypothetical future work

- Train on a larger dataset with better balance between the three categories
- Train a ternary classification model of COVID-19, normal, or pneumonia
- Add in more types of lung disease in such as lung cancer, smokers, etc.
- Get TensorFlow to function on GPU properly to speed up code