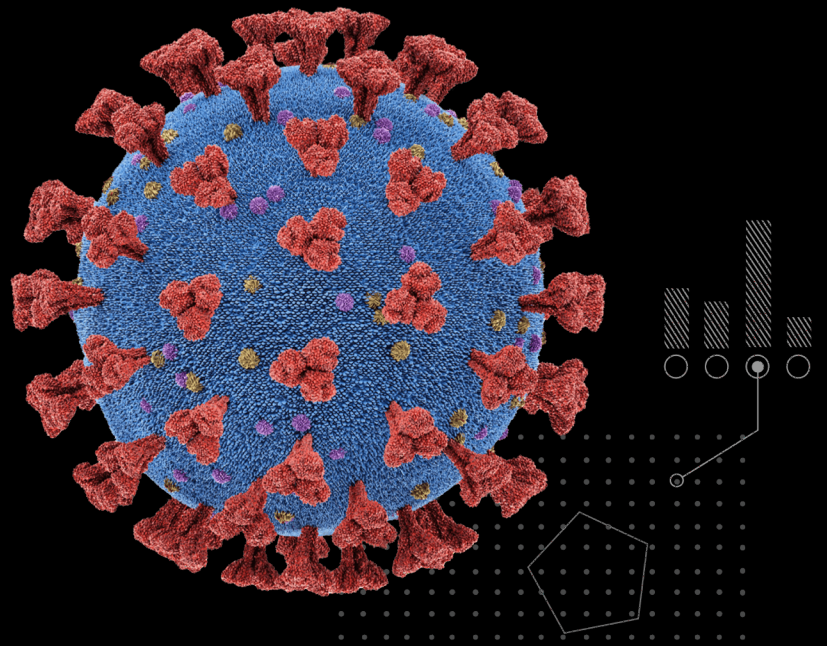
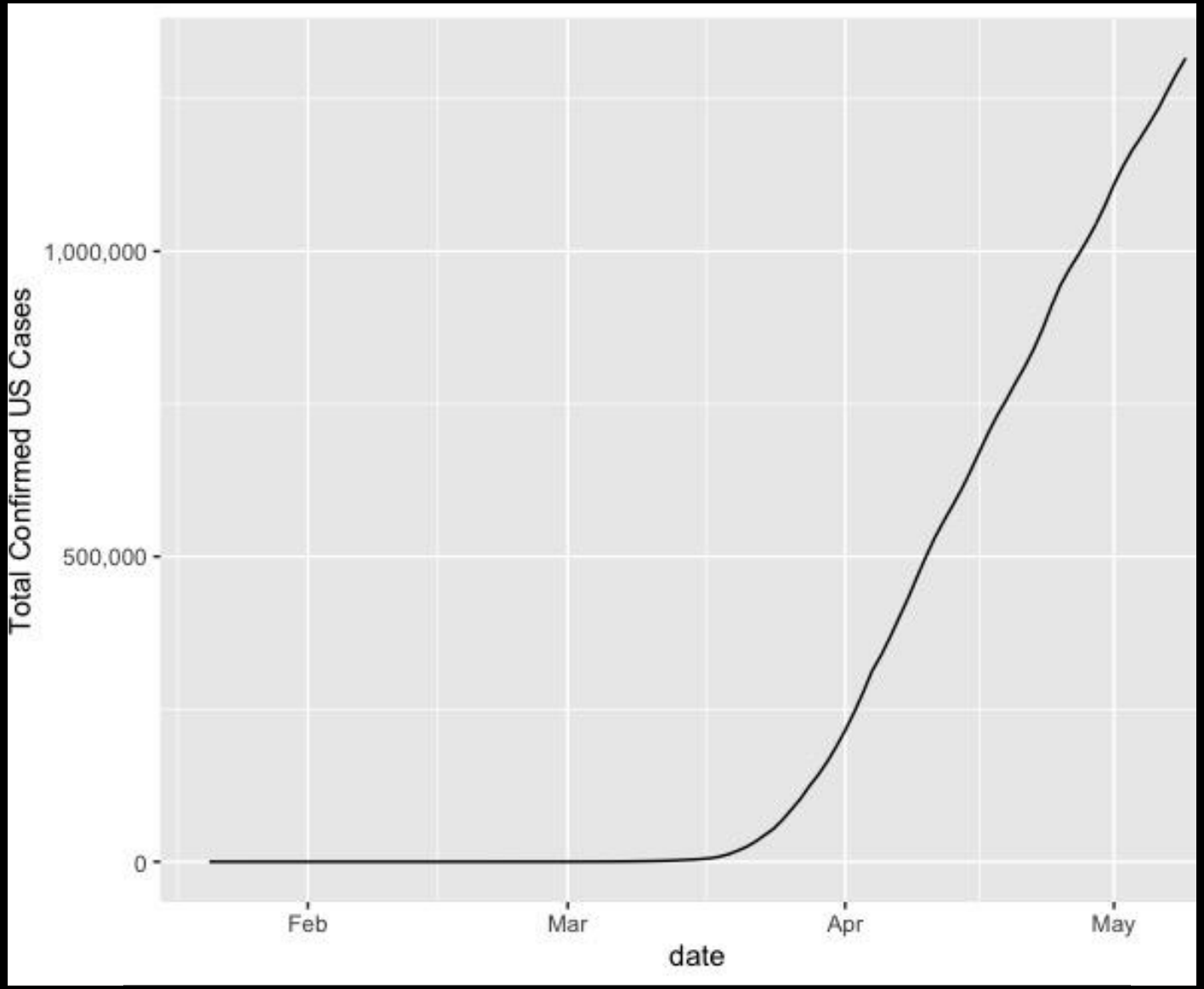
COVID-19 Detection Using Chest X-Ray Images

Muhammad Osama Khan
Muhammad Muneeb Afzal
Sean Robert Hornbuckle



Introduction And Motivation

- COVID-19 cases have increased exponentially over the past 4-5 months
- Widespread testing is essential to reduce the spread of the virus
- However, testing is currently limited and takes 1-2 days to produce results
- Alternatives?



Total Confirmed CoVID-19 Cases in the United States. Visualization done in R based off NY Times data.

COVID-19 and Pneumonia

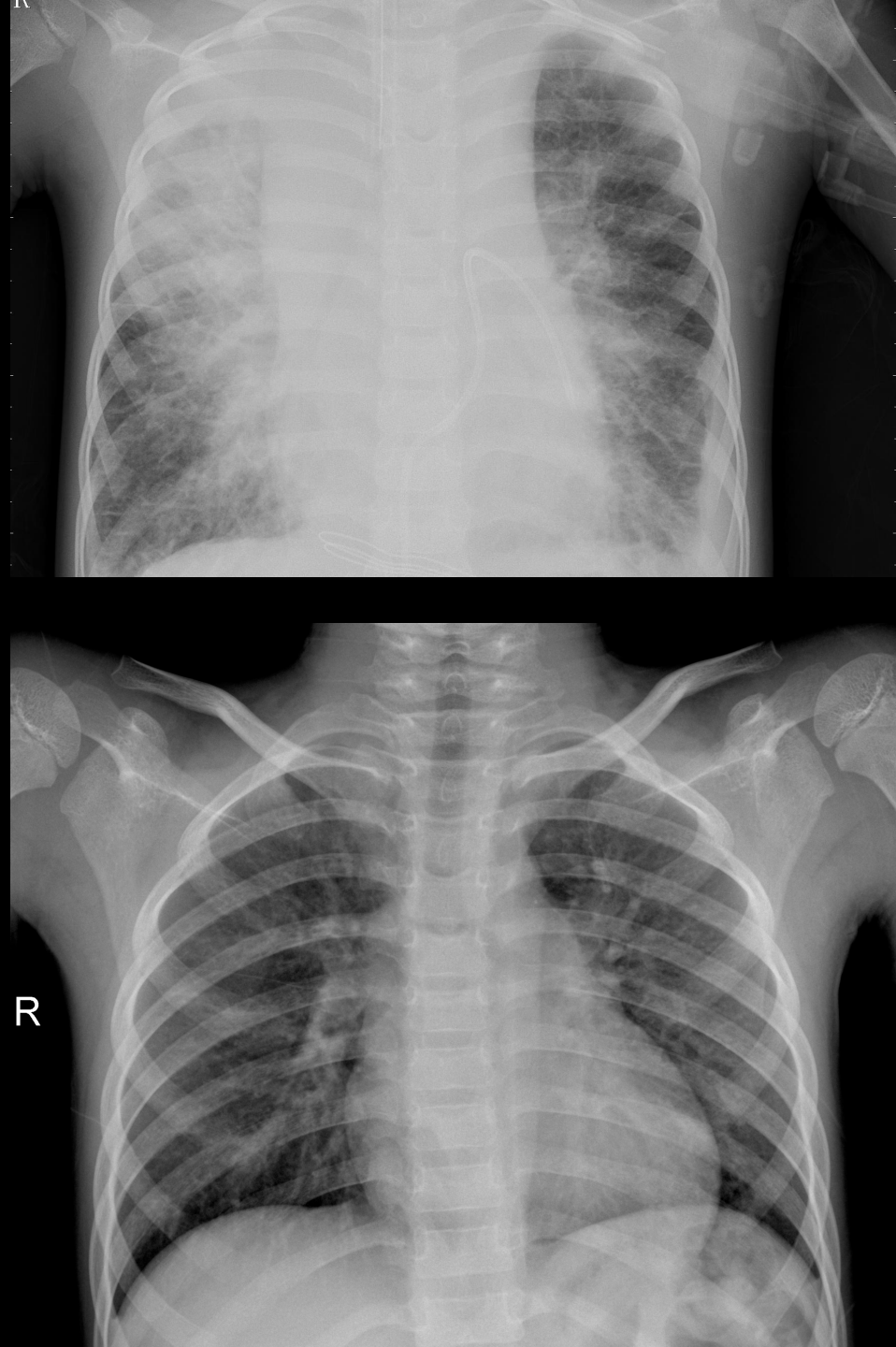
- COVID-19 poses special threat to people who develop lung conditions like **pneumonia**
- Pneumonia is generally diagnosed with a Chest X-Ray
- Experienced radiologists are needed for Chest X-Ray analysis
- Need to quickly identify patients who are in greatest need

Our Solution: Pneumonia Detection using Machine Learning

- Our software will allow health care professionals to upload a patient chest X-ray and get an instant prediction
- Benefits:
  - Faster diagnosis leads to faster allocation of resources
  - Software pre-screening can help get the patients at the highest risk evaluated by an experienced radiologist in a risk-based priority order
  - Reduces workload of experienced radiologists

How It Works

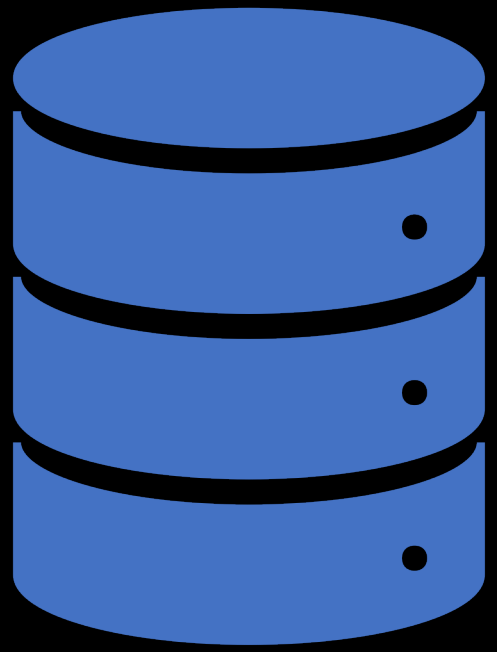
- The data set that was used to train our models contained thousands of healthy patients and CoVID-19/Pneumonia patients.
- Even with an experienced radiologist, it is challenging to produce a diagnosis from only observing an X-ray.



Bottom: X-Ray of Healthy Patient ; Top: X-Ray of Infected Patient

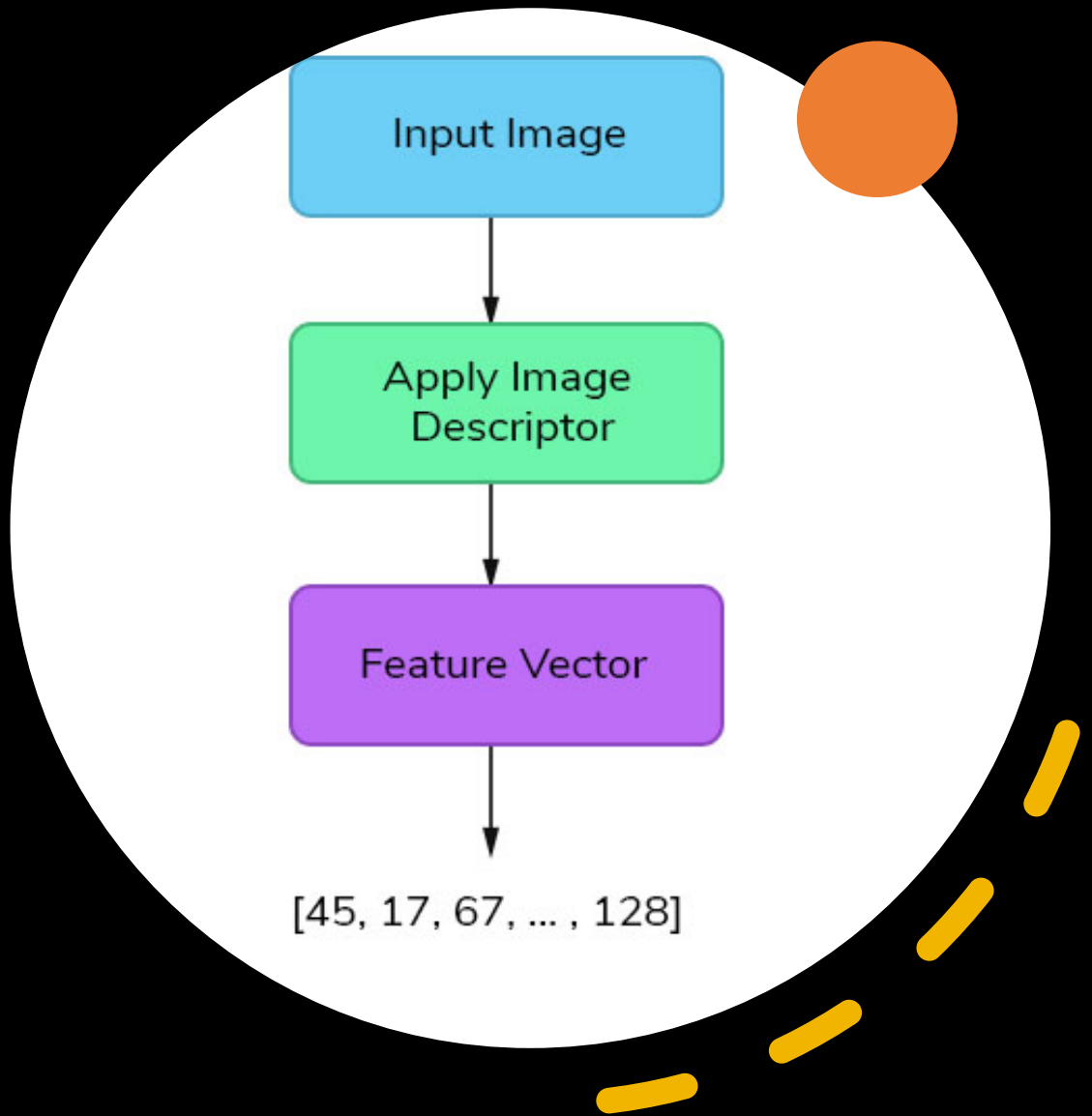
Implementation Overview

- Machine Learning:
  - Pre-processing procedure to clean the data and transform it into usable form
  - Designed hand-crafted features
  - Applied various ML algorithms
- Deep Learning:
  - Designed own architecture
  - Transfer Learning



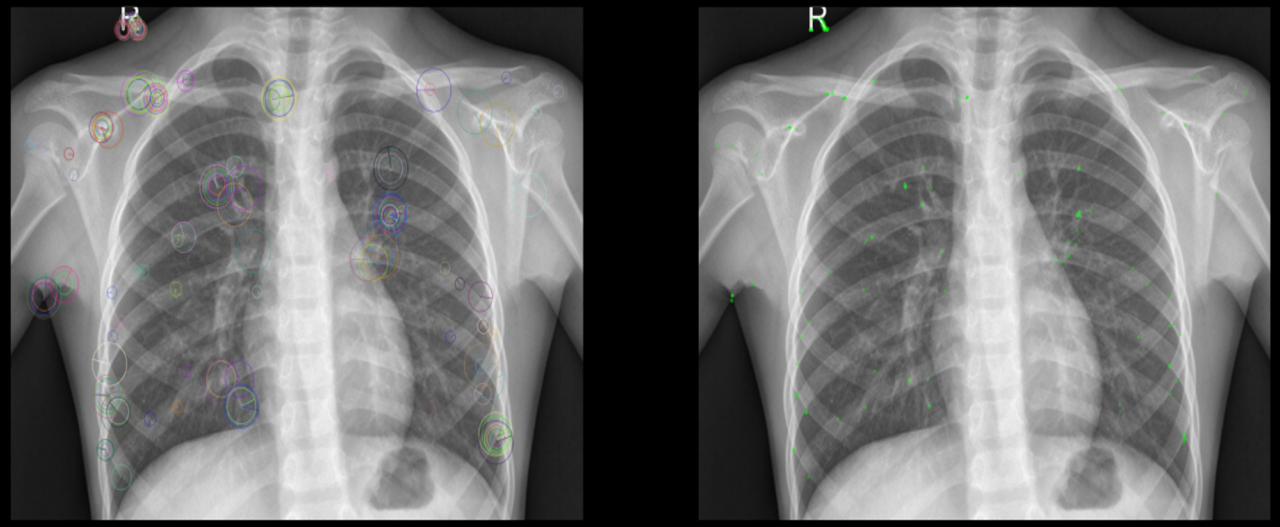
ML: Data Preprocessing

- First, we needed to convert the images into vectors.
- Select features that can help us distinguish between infected and normal.
- We tried different strategies for this conversion:
  - Flattening the image
  - ORB (Oriented FAST and Rotated BREIF)
  - Hu Moments and Haralick



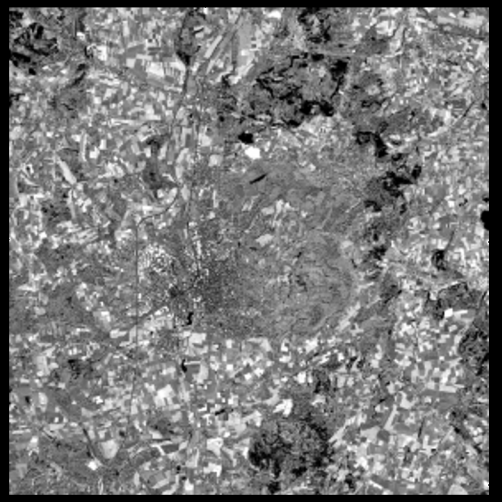
Feature Descriptor

ORB Descriptor



ORB algorithms detecting key points in the images

Haralick (Texture)



Hu Moments (Shape)

Machine Learning Models

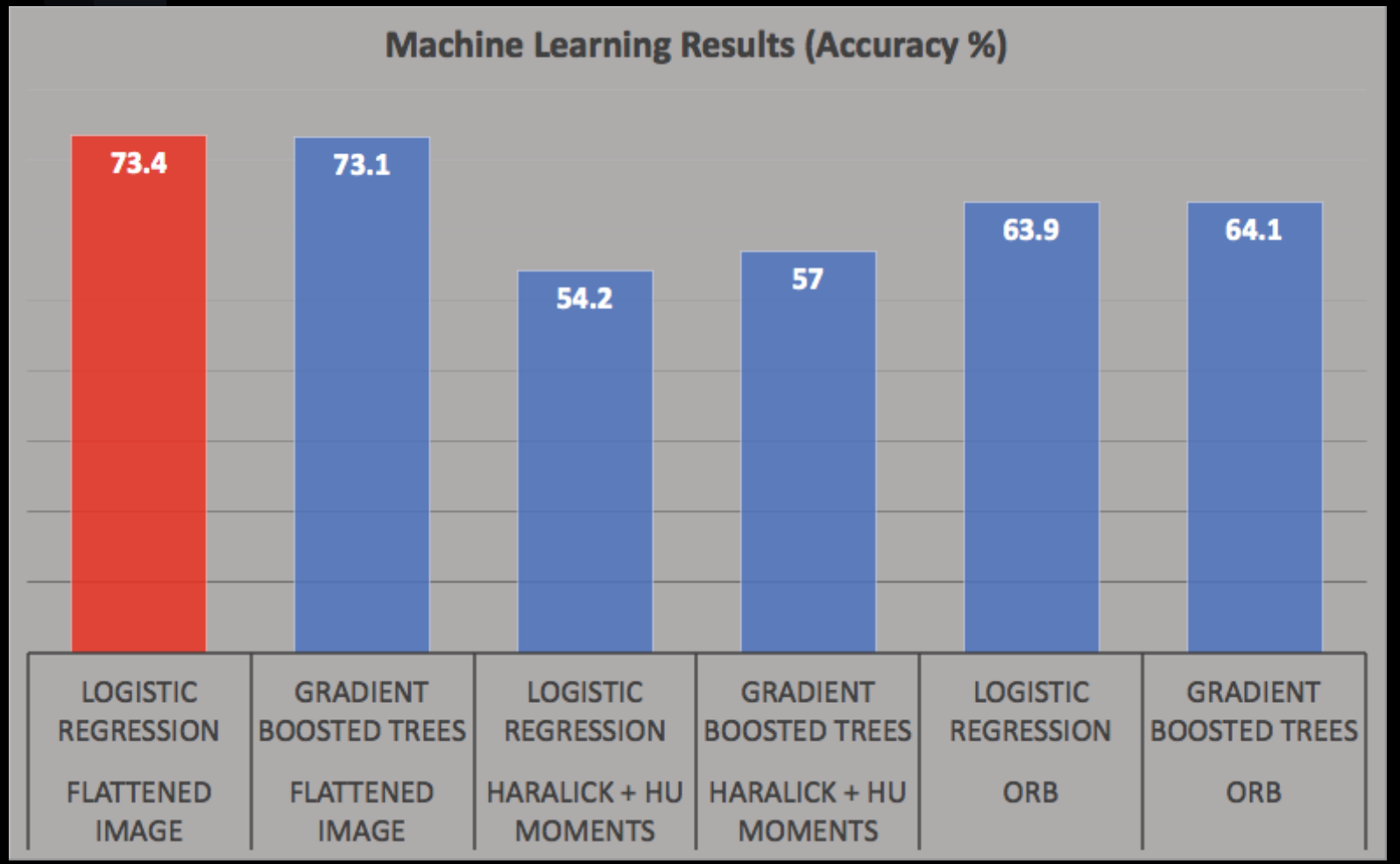
- Using Pyspark, the image files were read in and translated into vectors (parallel computing)
- The generated vectors were then fed into our ML models
- We tried the following ML models:
  - Logistic Regression
  - Gradient Boosted Trees

image_path	new_label	descriptc
Users/mma525/Doc...	0	[9, 221, 17, 251,
Users/mma525/Doc...	0	[129, 58, 159, 24.
Users/mma525/Doc...	0	[249, 189, 17, 62.
Users/mma525/Doc...	0	[5, 121, 121, 101.
Users/mma525/Doc...	0	[100, 37, 61, 101.
Users/mma525/Doc...	0	[232, 109, 63, 10.
Users/mma525/Doc...	0	[93, 114, 152, 10.
Users/mma525/Doc...	0	[204, 180, 19, 24.
Users/mma525/Doc...	0	[32, 237, 25, 249.
Users/mma525/Doc...	0	[36, 48, 17, 97, .
Users/mma525/Doc...	0	[56, 72, 20, 104,
Users/mma525/Doc...	0	[36, 100, 185, 69.
Users/mma525/Doc...	0	[29, 205, 26, 239.
Users/mma525/Doc...	0	[24, 140, 27, 254.
Users/mma525/Doc...	0	[116, 104, 61, 12.
Users/mma525/Doc...	0	[252, 72, 214, 18.
Users/mma525/Doc...	0	[120, 37, 101, 10.
Users/mma525/Doc...	0	[45, 117, 24, 235.
Users/mma525/Doc...	0	[182, 99, 225, 10.
Users/mma525/Doc...	0	[124, 180, 152, 2.

Image Pre-Processing results excerpt

Machine Learning Results (I)

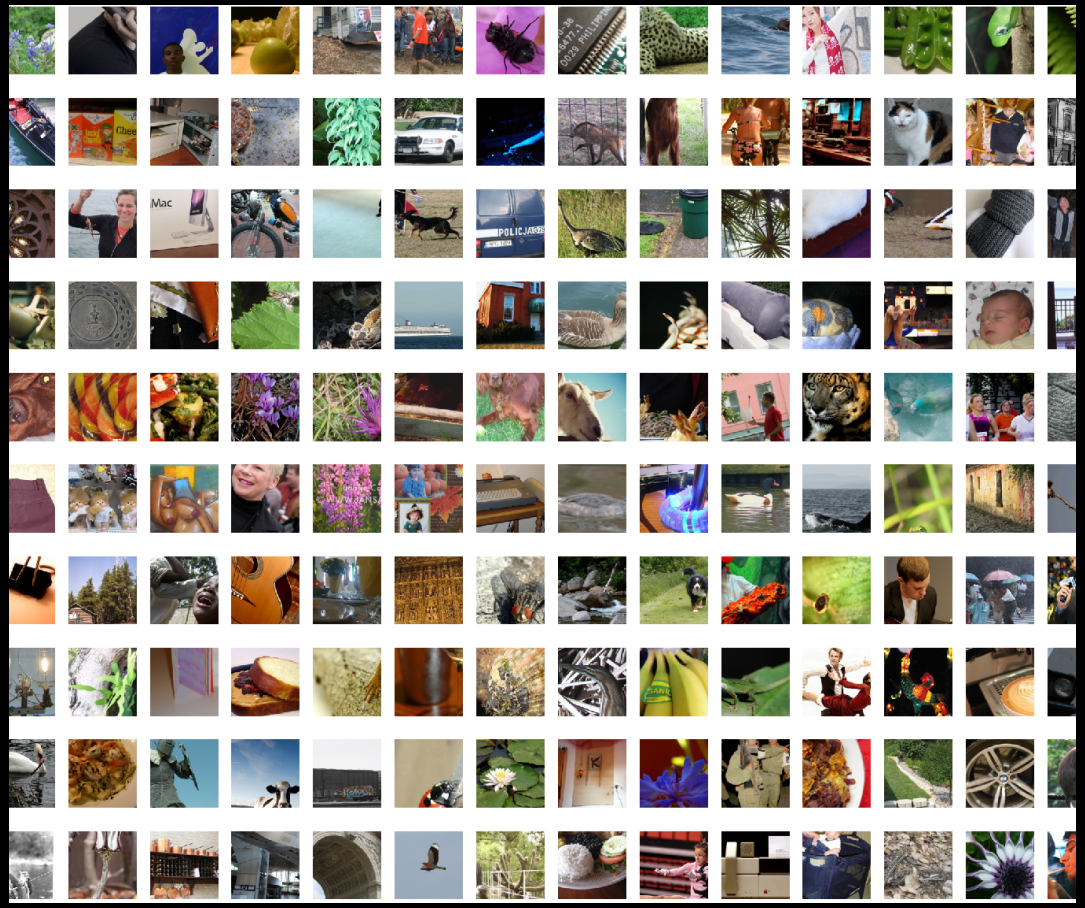
Feature Descriptor	Classifier Algorithm	Train Area under ROC	Train Accuracy	Test Area under ROC	Test Accuracy
Flattened Image	Logistic Regression	0.999	0.999	0.85	0.734
Flattened Image	Gradient Boosted Trees	0.999	0.999	0.87	0.731
Haralick + Hu Moments	Logistic Regression	0.887	0.83	0.61	0.542
Haralick + Hu Moments	Gradient Boosted Trees	0.889	0.81	0.63	0.57
ORB	Logistic Regression	0.999	0.997	0.674	0.639
ORB	Gradient Boosted Trees	1	1	0.7	0.641



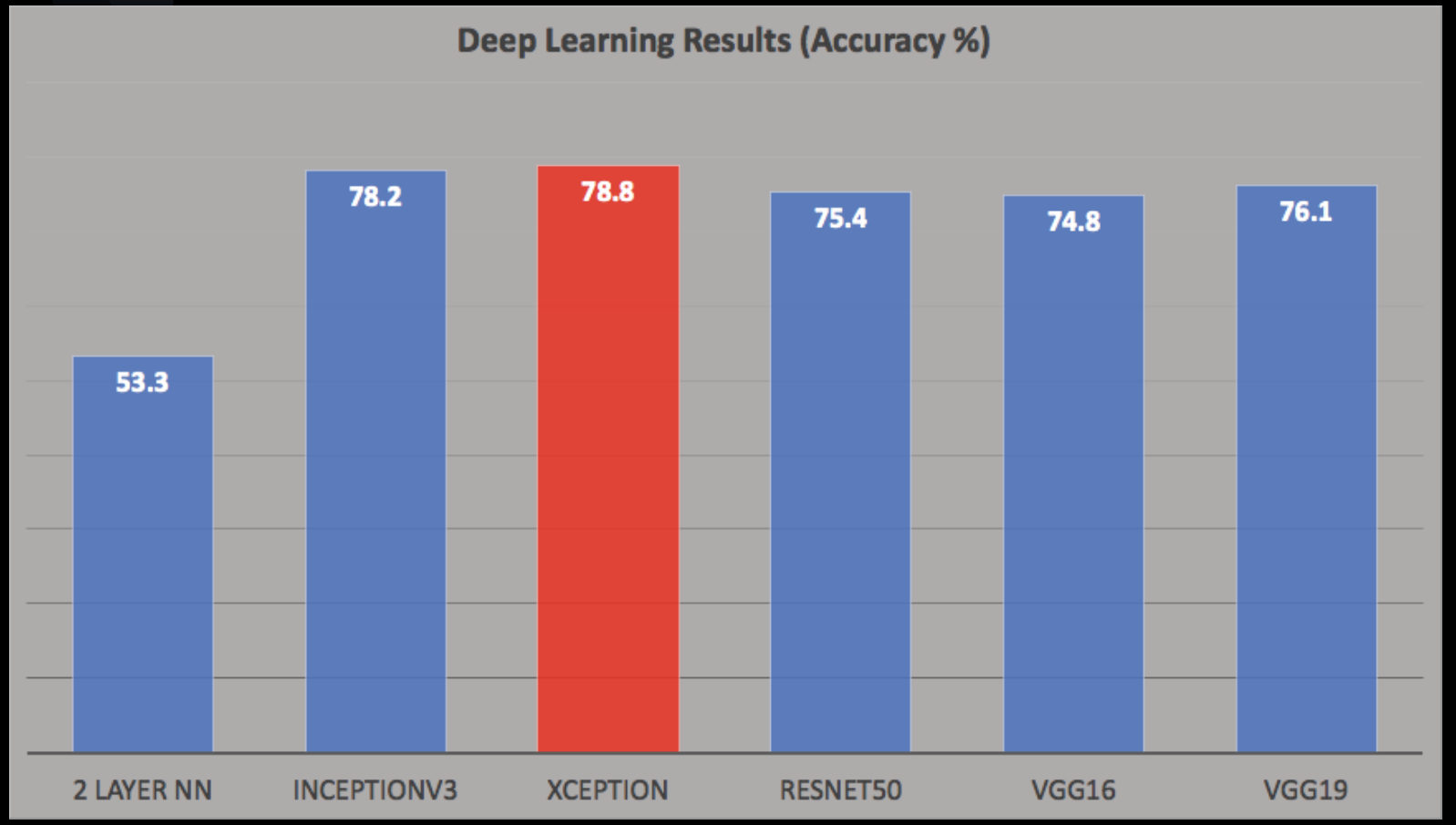
Machine Learning Results (II)

Deep Learning based COVID-19 Detection

- Designed our own model
  - Pros:
    - Freedom to choose neural network architecture
  - Cons:
    - Computationally expensive
    - Difficult to train effectively on small datasets
- Transfer Learning
  - Uses pretrained models on large image datasets such as ImageNet
  - Pros:
    - Computationally cheaper
    - Produces better results



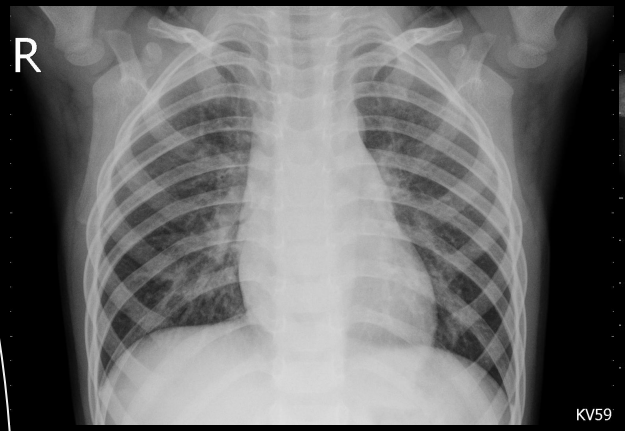
ImageNet Dataset excerpt (~14 million images from 1000 classes)

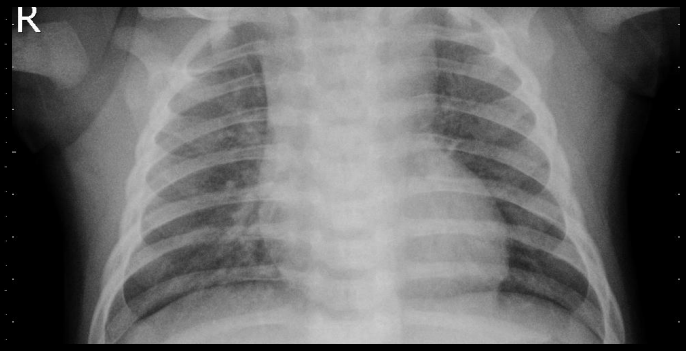


Deep Learning Results

Visualizations and Analysis

- Due to the application of our software, false positives are more tolerable than false negatives. It is less of an issue to have a healthy patient flagged than to miss a sick patient.





Examples of images that were false positives (healthy patients)

Model	Accuracy (%)	False Positives	False Negatives
Xception	78.8	81	43

Deployment

- Trained models are deployed as Spark SQL UDFs
- Hence, users don’t need to have any ML expertise and can directly interact with the model using SQL queries.
- This will allow technicians to use the software and keep billing rates low.

```
registerKerasImageUDF("corona_model_udf", "/tmp/corona_model.h5") # register model as a Spark SQL UDF
df = spark.read.format("image").load(img_dir + "/test") # load image data as a column formatted as an image struct ImageSchema
df.registerTempTable("sample_images") # register table
%sql
SELECT corona_model_udf(image) as predictions from sample_images # get predictions
```

Improvements

- Don’t resize images to 299 * 299 (use higher resolution images). Our target clients will likely have the resources to purchase or rent the level of processing power needed to avoid resizing.
- Use pre-trained models for grayscale images and pre-trained models on medical images.
- Remove bad images from our dataset.

NOT a chest X-ray

Packages and Technologies Used

- Pyspark
  - Pyspark ML Features
  - Pyspark MLlib Clustering
    - Kmeans
  - Sparkdl Library
  - Numpy
  - PIL
  - CV2
  - ORB
- R and R-Studio
- Databricks + AWS Clusters

Conclusion

- Implementing our software package into your hospital/clinic workflow will allow for instant pre-screening of CoVID-19/Pneumonia patients.
- Patients flagged by the pre-screening process can have their X-rays and other prognostic information evaluated at a higher priority, potentially saving both money and lives.
- By choosing to share your X-ray images along with the final patient diagnosis with us, you will help us improve upon our modeling, which will in turn produce more accurate field results.

References:

- CoVID-19 Deaths Dataset: https://github.com/nytimes/covid-19-data
- Image Dataset: https://github.com/ieee8023/covid-chestxray-dataset
- Google Image Classification: https://gogul.dev/software/image-classification-python