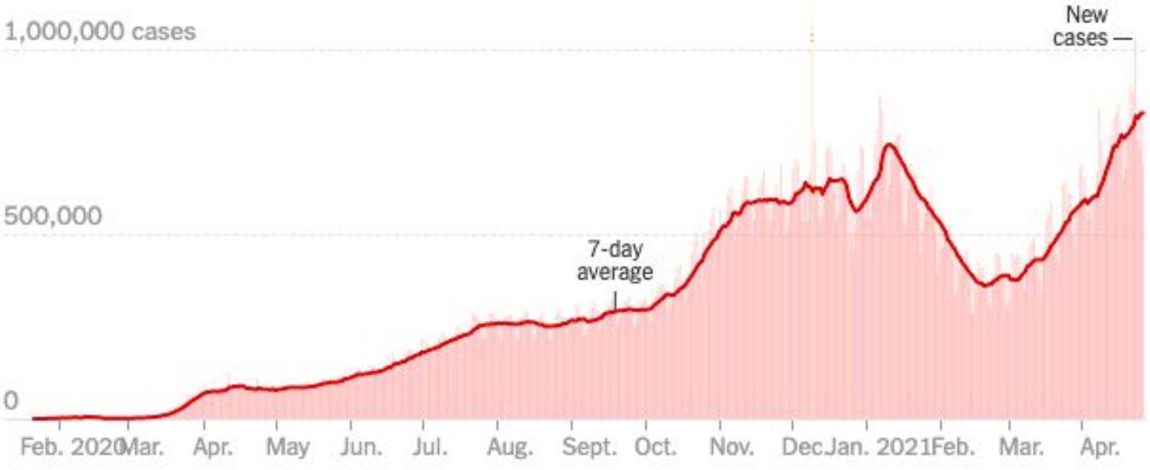
COVID-19 Detection Using CT Scans: STAT453 Project Presentation

James Thomason, Steven Xia, Saniya Khullar

Context / Background



	TOTAL REPORTED	ON APRIL 26	14-DAY CHANGE
Cases	147.5 million+	680,472	+19% →
Deaths	3.1 million+	10,294	+7% →

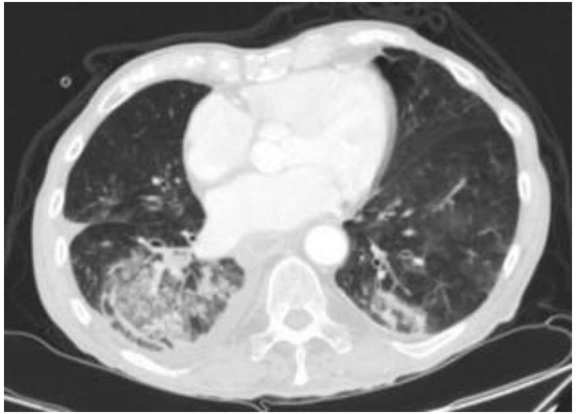
- COVID-19 has greatly impacted the entire globe
- Impoverished communities are especially at risk
  - Lack of available testing

Goal: Develop an object detection model to test for COVID-19 using a given subject's CT scan.

Introduction

- Computed Tomography (CT) scans are a type of imaging test
  - More detailed than X-ray
  - Produces pictures of organs and chest structures

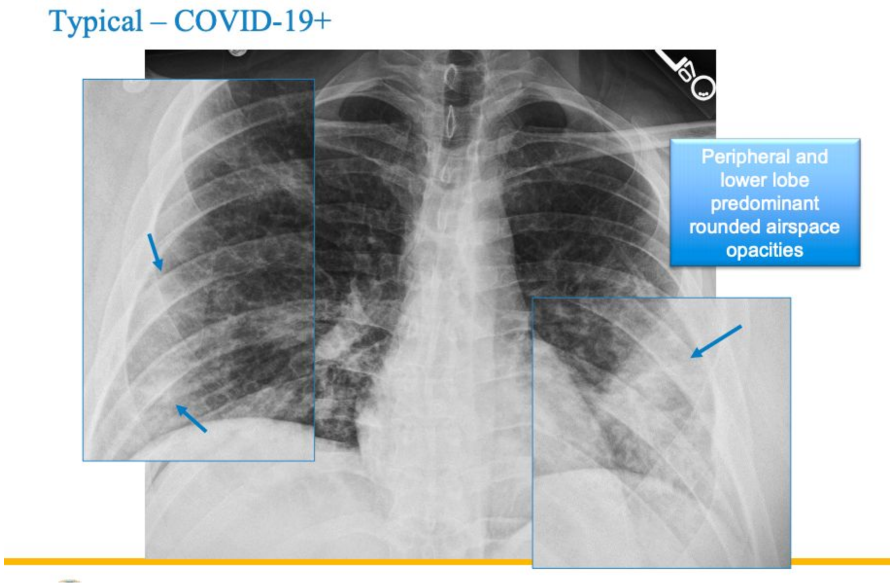
- CT scans of 742 individuals' chest cavities
  - 374 with COVID-19
  - 397 without COVID-19



Patient without COVID-19

Patient with COVID-19

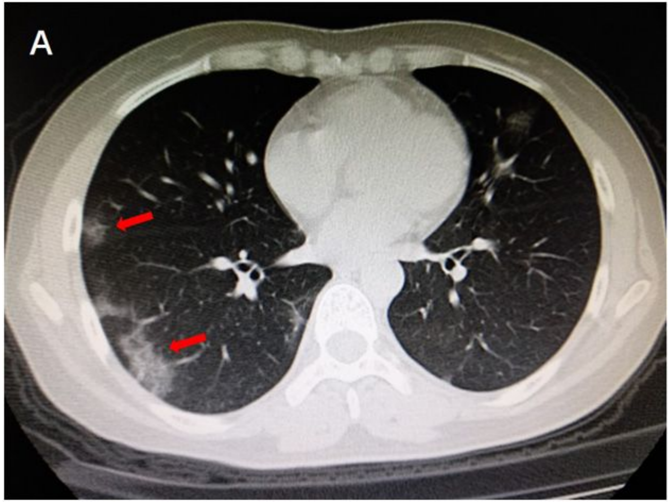
Related Work



- COVID-19 detection using X-Ray
  - Less expensive and more widespread than CT
  - Less detailed than CT
- Diagnosis of spine disorders using CT scans
  - Similar deep learning techniques were used
  - 3D imagery performs better than 2D imagery

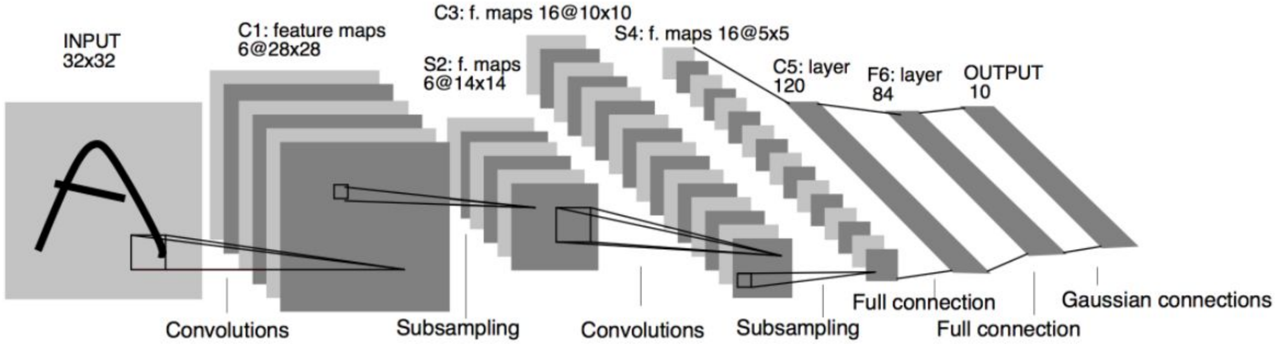
Proposed Method

- Some original CT scans contained markings and label
  - Gray scale vs. non-grayscale models
  - Labels and marks were not removed
- Considered architectures:
  - LeNet-5 (only architecture with gray scale)
  - AlexNet
  - ResNet
- LeNet-5
  - 64x64 images
  - Gray scale, so images dimensions are 64x64x1

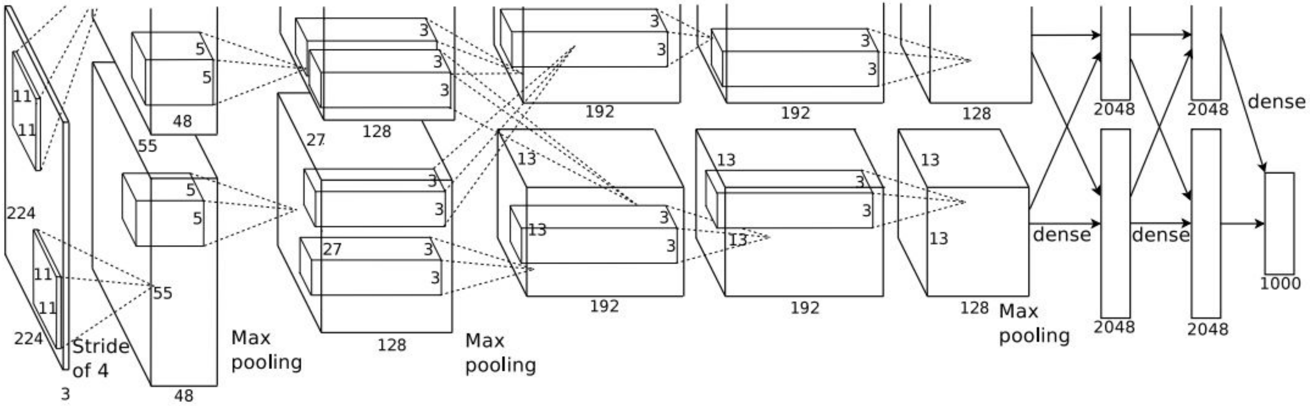


CNN Models Used

- LeNet-5



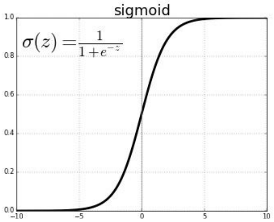
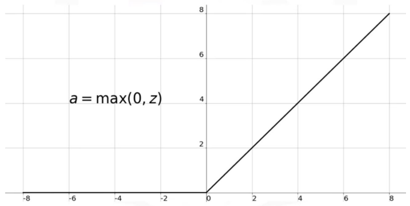
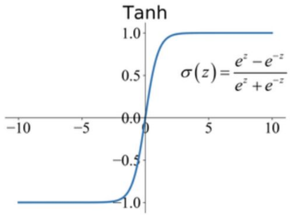
- AlexNet



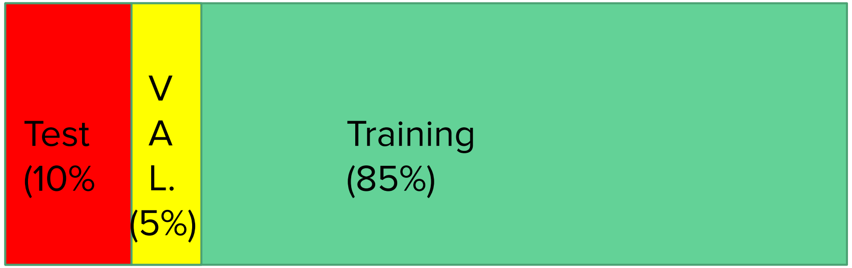
Optimizers Used

	ADAM ***(Adaptive Moment Estimation)***	Stochastic Gradient Descent ***(SGD)***
Tunable Parameters	3 Parameters (Learning Rate, Epsilon, Momentum, RMSprop)	1 Parameter (Learning Rate)
Advantages	• Default settings are typically optimal (usually 2nd best of other classifiers). • Tends to be the default used in Deep Learning	• Typically needs 0 memory on the GPU (computationally friendly) • Best generalization usually (lower testing error
Disadvantages	• Computational Burden (requires a lot of memory for the state) • Worse generalization (higher testing error)	• Computational Burden (requires a lot of memory for the state) • Worse generalization (testing error)

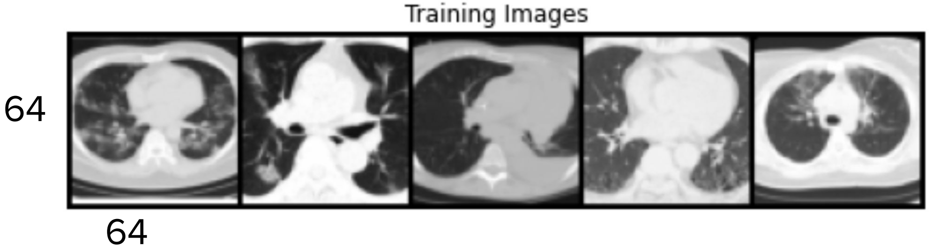
Python implementation	`tf.keras.optimizers.Adam(`
	`    learning_rate=0.001,`
	`    beta_1=0.9,`
	`    beta_2=0.999,`
	`    epsilon=1e-07,`
	`    amsgrad=False,`
	`    name="Adam",`
	`    **kwargs`
	`)`
		`tf.keras.optimizers.SGD(`
		`    learning_rate=0.01, momentum=0.0, nesterov=False, name="SGD", **kwargs`
		`)`

Activation Functions	**Sigmoid**	**Rectified Linear Unit (ReLU)**	**Hyperbolic Tangent (Tanh)**
Activation Function			
	Outputs real numbers to range between [0,1] Large - # → 0 Large + # → 1	Outputs real #s $\geq$ 0 Large - # → 0 Large + # the same	Outputs real numbers to range between [-1,1]. Centered around 0.
Advantages	Has an easier interpretation. Neuron does not fire when it outputs 0, but fires when it outputs 1.	6x improvement in convergence (compared to tanh). Become very popular	Scaled Sigmoid. Usually preferred to the nonlinear sigmoid function.
Disadvantages	May saturate and kill gradients. Neurons near	ReLU units could die irreversibly during training (can be delicate)	Has vanishing gradient problem, similar to Sigmoid

Experiments:



All CT Chest Scans were resized to 64 x 64
Then, a center crop was applied so width = height



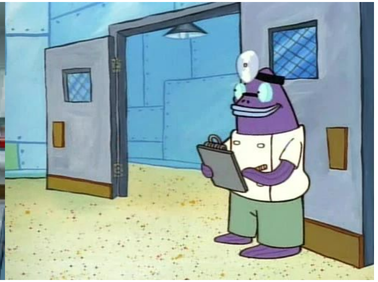
DataLoader was used with a batch size of 32

Random Seed = 123

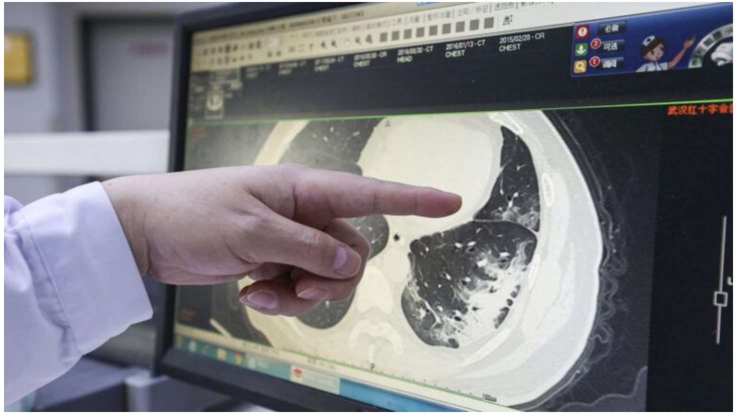
normalization vs. no normalization: normalization yielded the best results for every architecture that was tried

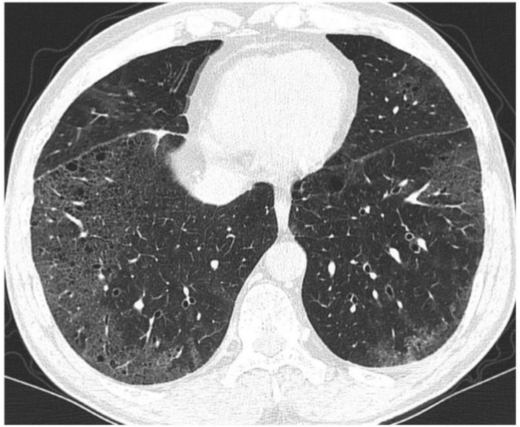
Architectures:	LeNet-5, AlexNet, ResNet
Activation Functions:	Sigmoid, ReLU, Tanh
Optimizers:	ADAM, Stochastic Gradient Descent
Normalization:	With Normalization, Without Normalization

Dataset

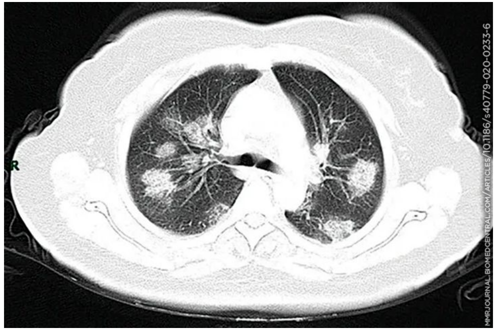


- Source: Kaggle.com
- It can be difficult to obtain CT Scan data:
    - Costly and challenging to recruit patients and ensure safety standards and protocols are met
- Easier to obtain X-ray data
- CT Scans can be very accurate

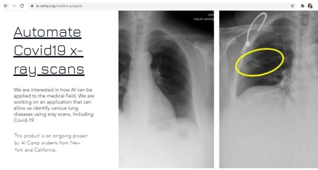




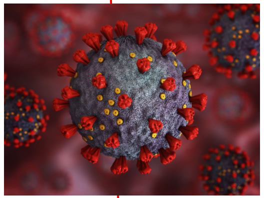
NO Covid-19
(Covid-19 Negative)



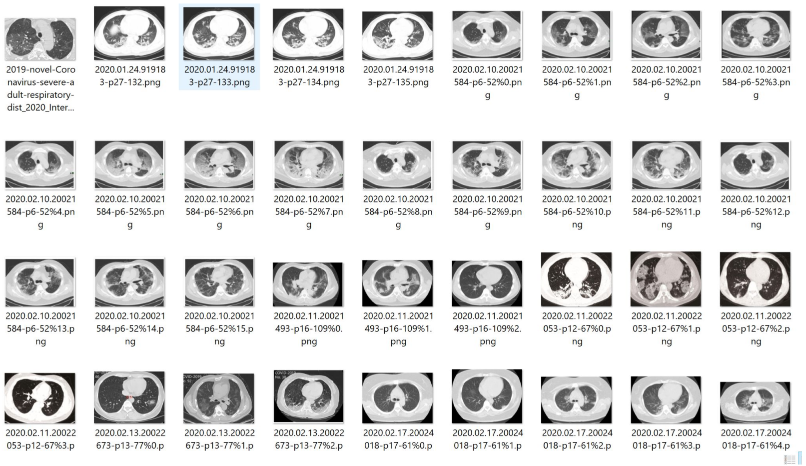
Covid-19
(Covid-19 Positive)



Feature Extraction and Feature Representation



Covid-19 Positive



Covid-19 Negative

Software / Hardware

Software:

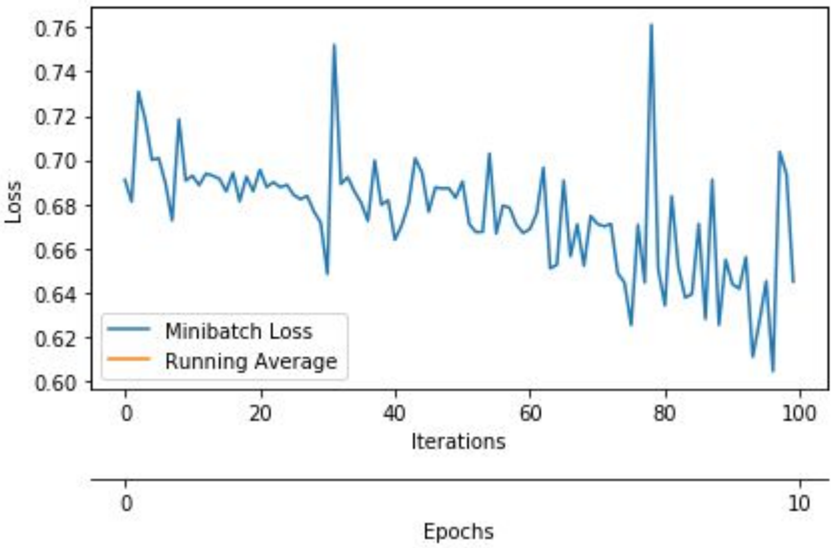
- All models were built within Jupyter Lab Notebooks (Python 3)
- Packages: PyTorch, NumPy, Torchvision, pandas, and matplotlib packages.
- Source Code adapted from: Professor Sebastian Raschka’s Code on Github (https://github.com/rasbt/stat453-deep-learning-ss21).
    - Standard LeNet-5, ResNet, and AlexNet architecture
    - Helper files: evaluation, train, plotting

Hardware:

- Code was executed on a laptop running windows. The code was implemented utilizing a GTX 980m nvidia graphics card.

Results and Discussion

```
Total Training Time: 1.06 min
Test accuracy 60.81%
```




```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 60, 60]             456
              ReLU-2            [-1, 6, 60, 60]               0
         MaxPool2d-3            [-1, 6, 30, 30]               0
            Conv2d-4           [-1, 16, 26, 26]           2,416
              ReLU-5           [-1, 16, 26, 26]               0
         MaxPool2d-6           [-1, 16, 13, 13]               0
            Linear-7                  [-1, 120]         324,600
              ReLU-8                  [-1, 120]               0
            Linear-9                   [-1, 84]          10,164
             ReLU-10                   [-1, 84]               0
           Linear-11                    [-1, 2]             170
================================================================
al params: 337,806
inable params: 337,806
-trainable params: 0
----------------------------------------------------------------
ut size (MB): 0.05
ward/backward pass size (MB): 0.56
ams size (MB): 1.29
imated Total Size (MB): 1.90
----------------------------------------------------------------
```

Conclusion

- Model wasn't helpful
- A similar model even with high accuracy won't help
- Time is your most valuable resource