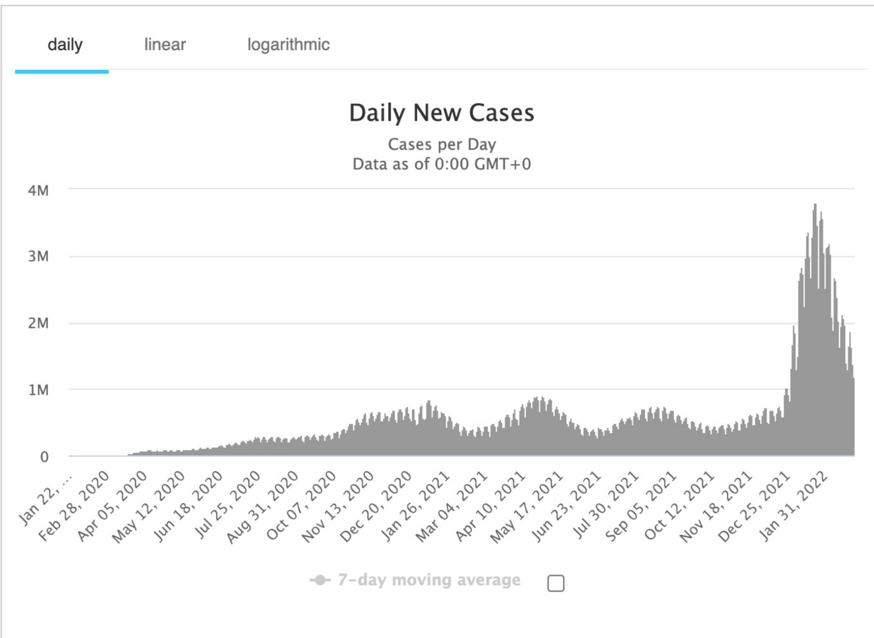
Covid 19 detection with CV

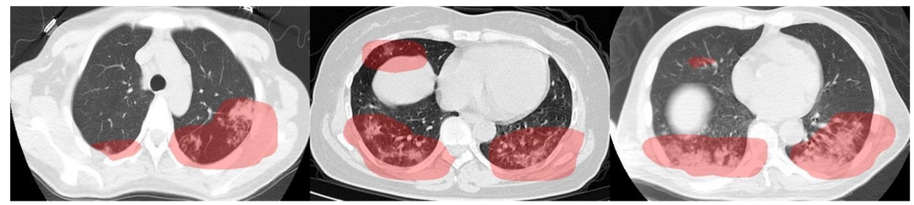
Megan Zhou

Outline

- Background
- Dataset and exploratory data analysis
- Data preprocessing
- Model development
- Conclusion
- Limitation and potential improvement
- Business use case

Project background





- More than 400 million cases
- Taken more than 5 million lives worldwide
- Need for rapid and effective screening tools
- Computed Tomography has been proposed

Dataset and exploratory data analysis

- 194,922 CT slices of cases with confirmed COVID-19 diagnoses
- 3745 patients
- Image data and metadata

Distribution of image data:

Chest CT image distribution

Type	Normal	Pneumonia	COVID-19	Total
train	35996	25496	82286	143778
val	11842	7400	6244	25486
test	12245	7395	6018	25658

Patient distribution

Type	Normal	Pneumonia	COVID-19	Total
train	321	558	1958	2837
val	126	190	166	482
test	126	125	175	426

Dataset and exploratory data analysis

The metadata includes:

- **Patient ID**
- Data source
- Country (if available)
- Age (if available)
- Sex
- **Finding (Normal, Pneumonia, or COVID-19)**
- **Verified finding, which indicates whether the finding is confirmed (Yes or No)**
- Slice selection, which indicates how slice selection was performed (either Expert, Non-expert, or Automatic)
- View (all are axial CT)
- Modality

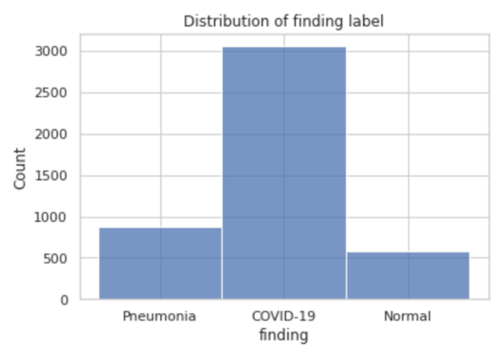
Dataset and exploratory data analysis

- Size of our data (28 GB)

```
[ ] #Configuration environment
    import os

    os.environ['KAGGLE_USERNAME'] = "skyeczy" # username from the json file
    os.environ['KAGGLE_KEY'] = "69bd62b67f259bdc16a56d9c423bf0fe" # key from the json file
```

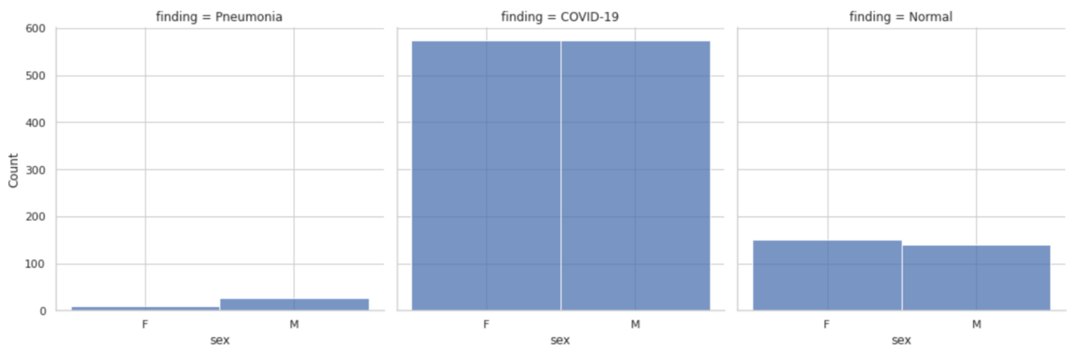
```
[ ] !kaggle datasets download -d hgunraj/covidxct

    Downloading covidxct.zip to /content
    100% 28.7G/28.7G [05:50<00:00, 121MB/s]
    100% 28.7G/28.7G [05:50<00:00, 87.9MB/s]
```

- Distribution of our label

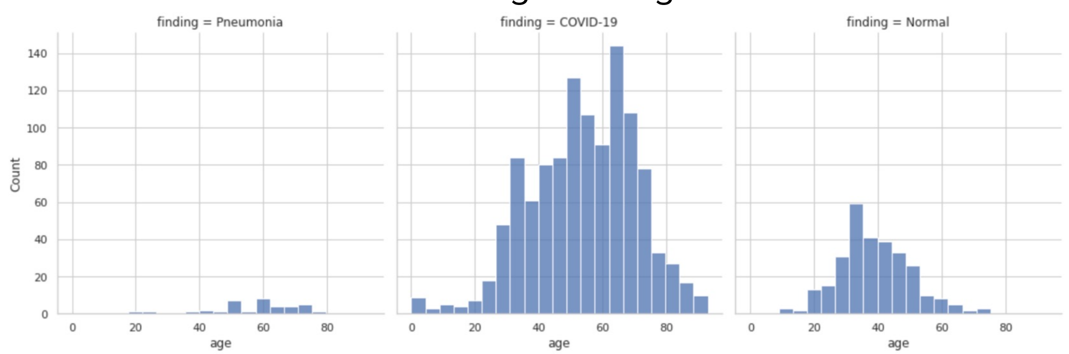


Dataset and exploratory data analysis

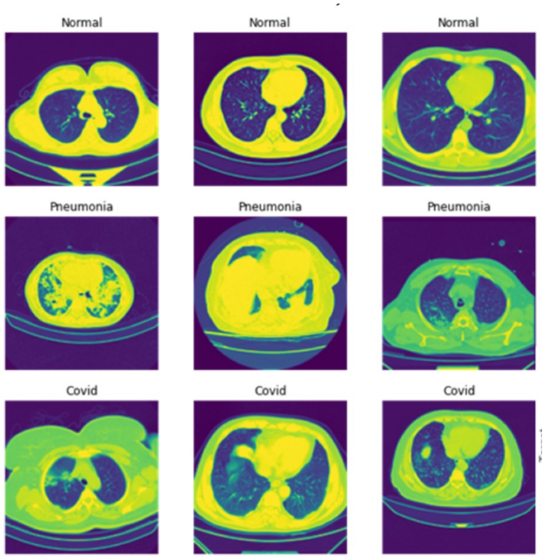
- Label distribution with regard to sex



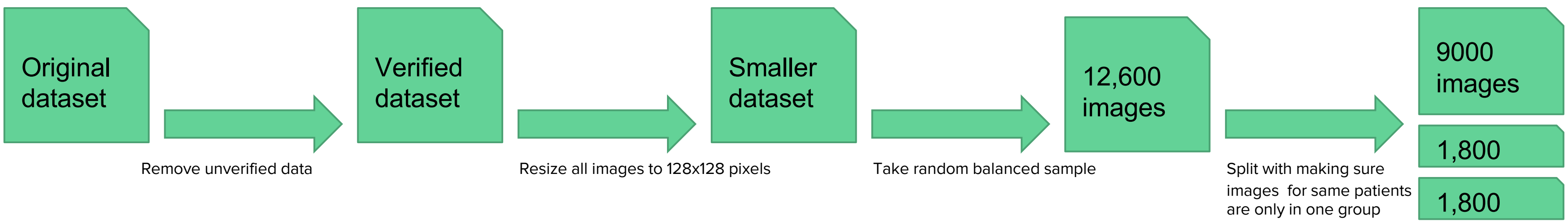
- Label distribution with regard to age

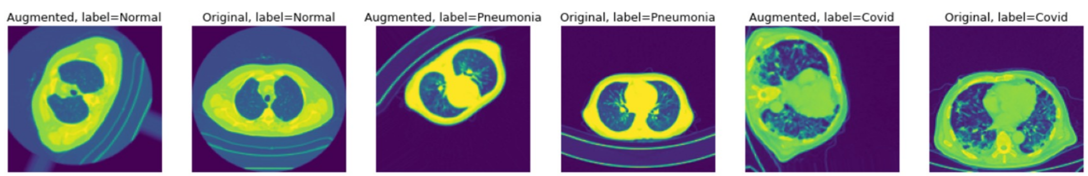


- Visual inspection of x-ray images



Data preprocessing






```
augmented_datagen = ImageDataGenerator(horizontal_flip = True,
                                       vertical_flip = True,
                                       rescale=1./255,
                                       width_shift_range=(-8, 8),
                                       height_shift_range=(-8, 8),
                                       rotation_range = 90,
                                       preprocessing_function=other_adjustment)

datagen = ImageDataGenerator(rescale=1./255)
```

Model development

- CNN
- Transfer learning
  - VCG16
  - ResNet50
- Traditional Machine Learning models

CNN

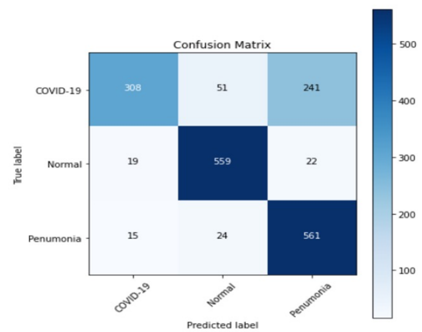
```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_9 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_10 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 conv2d_11 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_11 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 flatten_3 (Flatten)         (None, 25088)             0

 dense_9 (Dense)             (None, 96)                2408544

 dense_10 (Dense)            (None, 64)                6208

 dense_11 (Dense)            (None, 3)                 195

=================================================================
Total params: 2,508,195
Trainable params: 2,508,195
Non-trainable params: 0
_________________________________________________________________
```

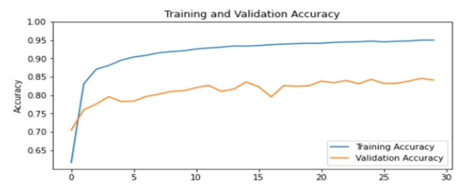
With augmentation

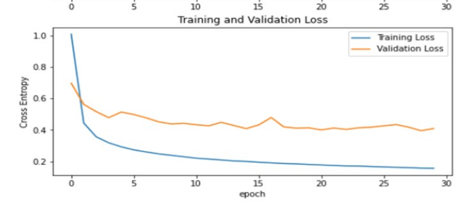
CNN with augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.9	0.51	0.65	
Pneumonia	0.68	0.93	0.79	
Normal	0.88	0.93	0.9	
Overall	0.82	0.79	0.78	0.79

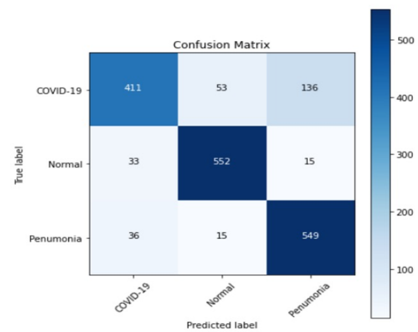
Without augmentation

CNN without augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.85	0.68	0.76	
Pneumonia	0.78	0.91	0.84	
Normal	0.89	0.92	0.9	
Overall	0.84	0.84	0.84	0.84









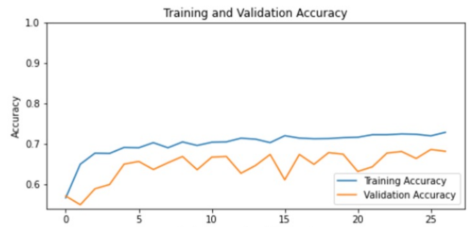
Transfer Learning with VCG16

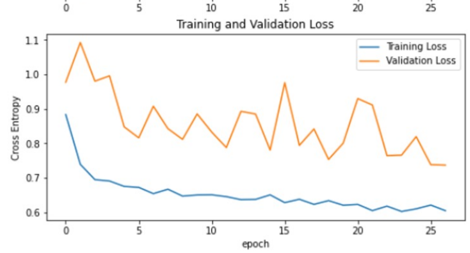
-one of the winning models in ILSVRC

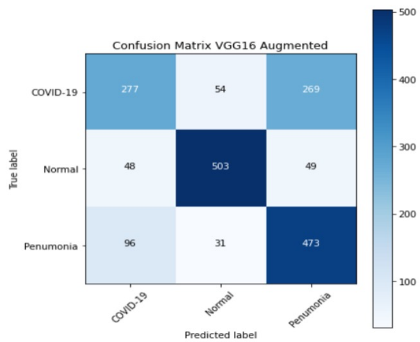
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 4, 4, 512)         14714688

 global_average_pooling2d (G  (None, 512)              0
 lobalAveragePooling2D)

 dense (Dense)               (None, 64)                32832

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 14,747,715
Trainable params: 33,027
Non-trainable params: 14,714,688
```

With augmentation

VGG with augmentation				
	Precision	Recall	F1-score	Accuracy
COVID-19	0.66	0.46	0.54	
Pneumonia	0.6	0.79	0.68	
Normal	0.85	0.84	0.85	
Overall	0.7	0.7	0.69	0.7

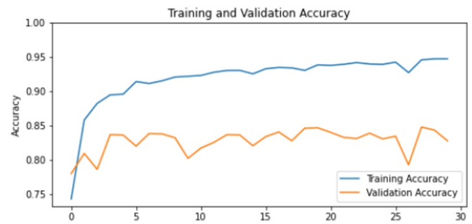


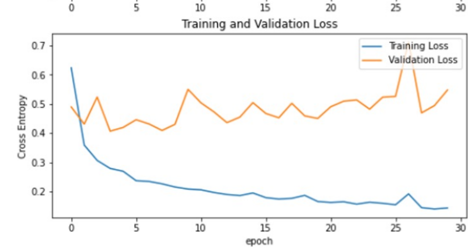


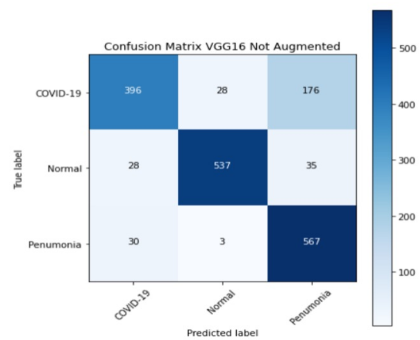


Without augmentation

VGG without augmentation				
	Precision	Recall	F1-score	Accuracy
COVID-19	0.87	0.66	0.75	
Pneumonia	0.73	0.94	0.82	
Normal	0.94	0.89	0.92	
Overall	0.85	0.83	0.83	0.83







Transfer Learning with ResNet50

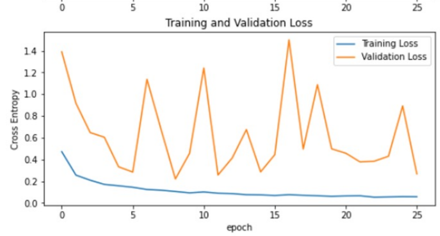
```
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 4, 4, 2048)        23587712

 global_average_pooling2d_1   (None, 2048)             0
 (GlobalAveragePooling2D)

 dense_2 (Dense)             (None, 64)                131136

 dense_3 (Dense)             (None, 3)                 195

=================================================================
Total params: 23,719,043
Trainable params: 23,665,923
Non-trainable params: 53,120
```

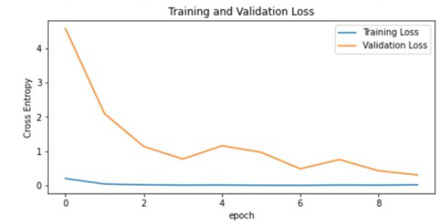
With augmentation

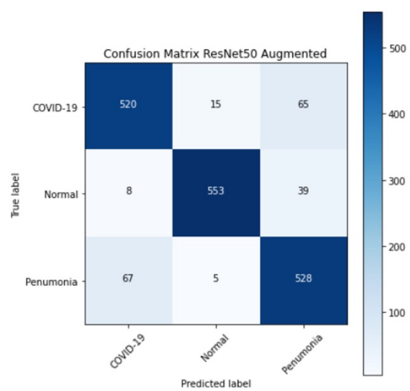
Resnet with augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.87	0.87	0.87	
Pneumonia	0.83	0.88	0.86	
Normal	0.96	0.92	0.94	
Overall	0.89	0.9	0.89	0.89

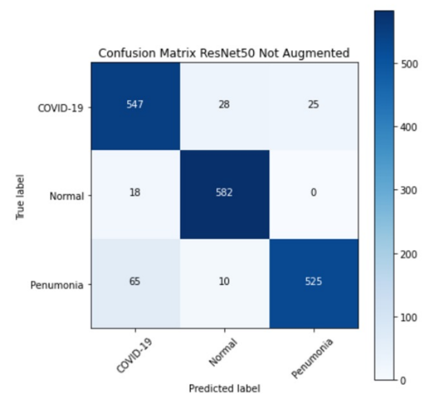
Without augmentation

Resnet without augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.87	0.91	0.89	
Pneumonia	0.95	0.87	0.91	
Normal	0.94	0.97	0.95	
Overall	0.92	0.92	0.92	0.92









Traditional ML

- Processing: raw and histogram

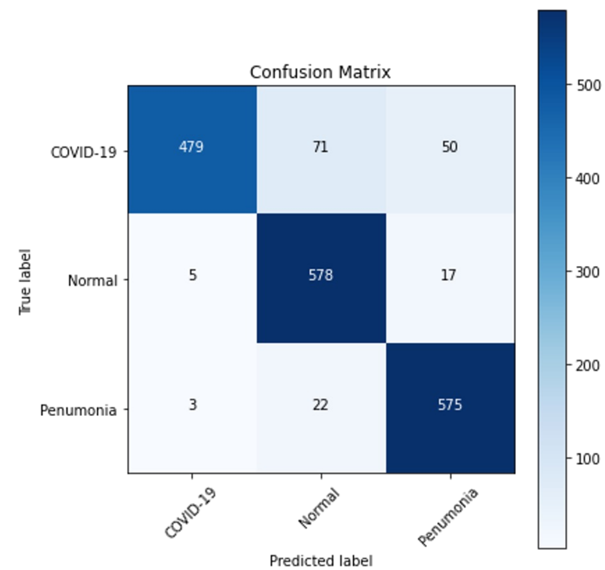
```
[ ] def extract_color_histogram(image, bins=(8, 8, 8)):
        # extract a 3D color histogram from the HSV color space using
        # the supplied number of `bins` per channel
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
        hist = cv2.calcHist([hsv], [0, 1, 2], None, bins,[0, 180, 0, 256, 0, 256])

        if imutils.is_cv2():
            hist = cv2.normalize(hist)

        else:
            cv2.normalize(hist, hist)
        # return the flattened histogram as the feature vector
        return hist.flatten()
```

- Different Performance

Performance of traditional ML models		
Model	**Raw pixel accuracy**	**Histogram accuracy**
Naïve Bayes	0.47	0.62
Linear SVC	0.58	0.59
Random Forest	0.9	0.81

- Result of Random Forest

Random Forest Classifier				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.98	0.8	0.88	
Pneumonia	0.9	0.96	0.93	
Normal	0.86	0.96	0.91	
Overall	0.91	0.91	0.91	0.9



Conclusion

Comparison of different models		
Framework	**Model**	**Accuracy**
Deep Neural Network	CNN with augmentation	79%
Deep Neural Network	CNN w/o augmentation	84%
Transfer Learning	VGG16 with augmentation	70%
Transfer Learning	VGG16 w/o augmentation	83%
Transfer Learning	ResNet50 with augmentation	89%
Transfer Learning	ResNet50 w/o augmentation	92%
Traditional ML model	Naïve Bayes with raw pixel data	47%
Traditional ML model	Naïve Bayes with color histogram data	62%
Traditional ML model	Linear SVC with raw pixel data	58%
Traditional ML model	Linear SVC with color histogram data	59%
Traditional ML model	Random Forest with raw pixel data	90%
Traditional ML model	Random Forest with color histogram data	81%

Limitation and possible improvement

Limitation:

- Size of our data

Possible improvement:

- Use of GPU
- Combine with text data for symptoms, or demographic variables
- Tune threshold based on recall

Potential business use case

- Rapid diagnosis
- Incorporated into insurance underwriting process
    - Ask user to submit their chest ct image when filling application
    - Compute risk of chest-related disease
    - Include the risk when considering approval/pricing of insurance

Question