

COVID-19 X-RAY DETECTION

Vijay Saddi
August 2020



GitHub Repository: <https://github.com/vijay-ss/Covid19-Xray-Detection>

AGENDA



USE CASE



DATA SET



DATA PREP



MODEL BUILDING



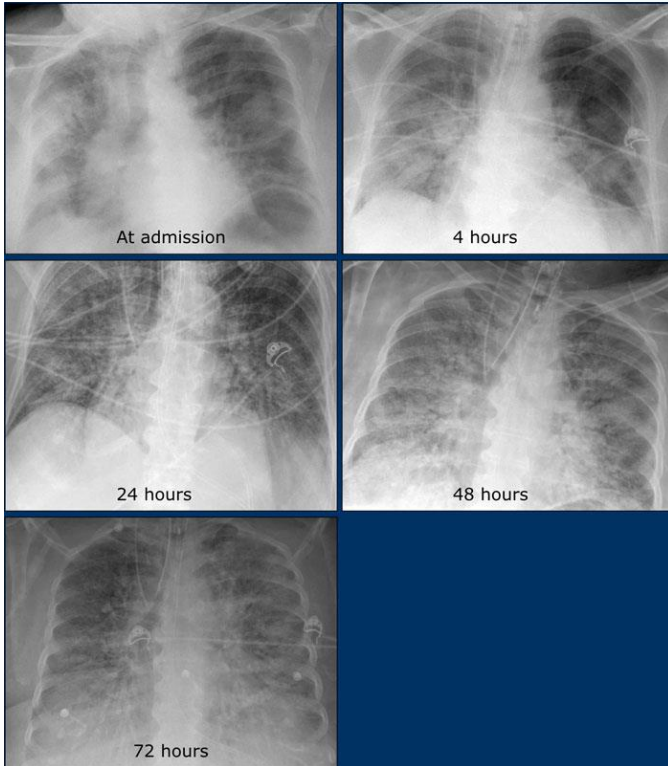
EVALUATION



COVID-19 BACKGROUND

- The 2019 outbreak of Covid-19 has kept the world on lockdown with the effort to stop the spread
- It is a highly contagious illness which affects the lungs of the respiratory system
- Transmitted via close proximity via small droplets produced by coughing , sneezing & talking
- X-rays and CT scans of Covid-19 victims can depict the damage inflicted on the lungs

RADIOLOGY



- Chest x-rays are useful in the follow-up of the disease (it is insensitive early on)
- The image on the left is an 83-year old male with Covid-19
- Image findings:
 - Admission: Ill defined bilateral alveolar consolidation
 - 4-hrs: Radiology worsening, affected lower lobes
 - 24-hrs: Bilateral alveolar consolidation¹
 - 48-hrs: Further radiology worsening
 - 72-hrs: Panlobar affectation², with typical radiology findings of ARDS³; 24 hrs later patient passed away

1. Infers an alveolar spread of disease, commonly due to pneumonia
2. Alveolar destruction within the lobule
3. Acute Respiratory Distress Syndrome: fluid buildup in the alveoli

CURRENT STATISTICS ON COVID-19

Last updated: August 08, 2020, 20:24 GMT

Coronavirus Cases:

19,725,620

Deaths:

727,540

Recovered:

12,659,896

- High rate of infection
- High rate of recovery (95%)
- 5% death rate (August 2020)

ACTIVE CASES

6,338,184

Currently Infected Patients

6,273,265 (99%)
in Mild Condition

64,919 (1%)
Serious or Critical

[Show Graph](#)

CLOSED CASES

13,387,436

Cases which had an outcome:

12,659,896 (95%)
Recovered / Discharged

727,540 (5%)
Deaths

[Show Graph](#)

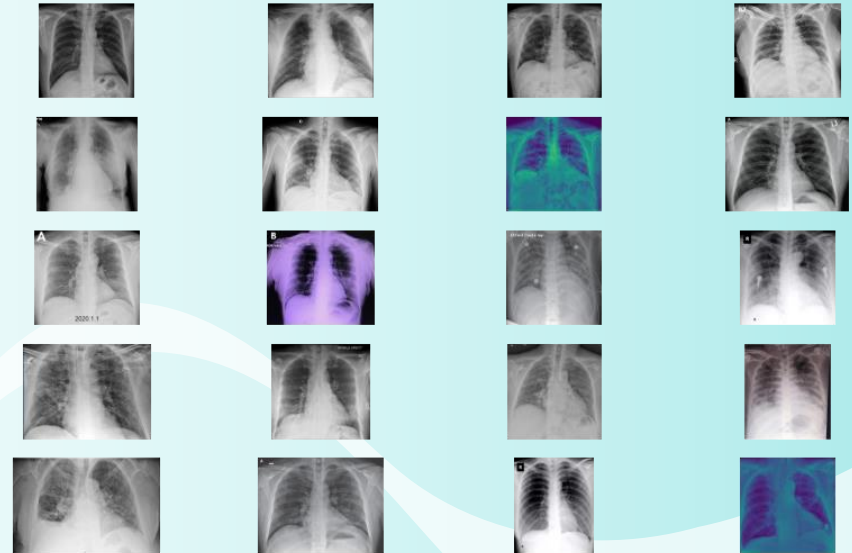


QUESTION

With the emergence of Deep Learning for AI in healthcare, can we build a model to help identify Covid-19 by evaluating lung x-rays?

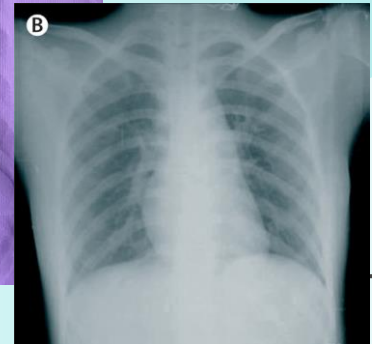
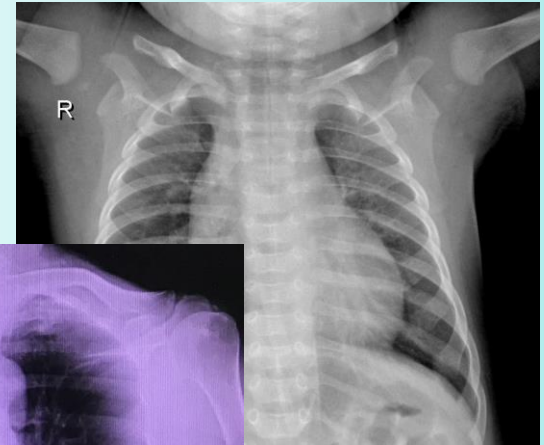
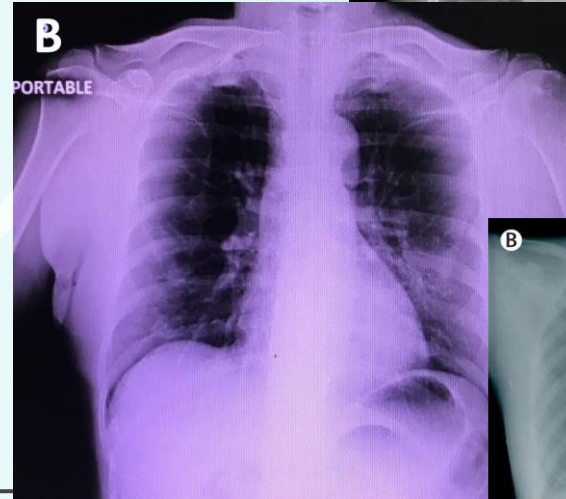
DATA SET

- 2 main data sources for Covid-19 and Normal x-ray images
- Covid-19 x-rays were taken from Github:
<https://github.com/ieee8023/covid-chestxray-dataset>
- Normal x-ray images were taken from Kaggle:
<https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- There were 180 images per group, with a total of **360**



EXPLORATION & QUALITY ASSESSMENT

- Image file sizes range from 400-600kb each
- Images are of high quality (720p-1080p resolution)
- Each image has different dimensions (width vs height)
- RGB & greyscale formats



DATA PRE-PROCESSING

- Github images were filtered to select only Covid-19 diagnosed, and Posteroanterior view
- From the entire pool of images, the *80/20 split* was implemented to divide the training/test data
- **Training:** 144 images per category (Covid & Normal)
- **Testing:** 36 images per category (Covid & Normal)

```
df_metadata[(df_metadata["finding"]=="COVID-19") & (df_metadata["view"]=="PA")]
```

| | patientid | offset | sex | age | finding | RT_PCR_positive | survival | intubated | intubation_prese |
|---|-----------|--------|-----|------|----------|-----------------|----------|-----------|------------------|
| 0 | 2 | 0.0 | M | 65.0 | COVID-19 | Y | Y | N | N |



FEATURE ENGINEERING

- Prior to model training, augmentation methods were required to account for variations in the image makeup:
 - **Normalization:** images rescaled by 1/255 to ensure pixel values range between 0 and 1
 - **Shear range:** shear angle for variations in image rotation
 - **Zoom range:** account for variations in zoom between images
 - **Horizontal flip:** reverses the rows/columns of pixels in random images – as lung images show symmetry in nature
 - **Target size:** resize all images to standard 224x224

```
# prepping & augmenting the images for loading into the model as inputs for training  
train_data = image.ImageDataGenerator(  
    rescale = 1./255, #normalize images  
    shear_range = 0.2,  
    zoom_range = 0.2,  
    horizontal_flip = True  
)
```

```
# converting image size, set the batch size (standard 224 by 224)  
# feed the augmented images from previous step into the generator  
train_generator = train_data.flow_from_directory(  
    'Covid_Dataset/Train',  
    target_size = (224,224),  
    batch_size = BS,  
    class_mode = 'binary'  
)
```

DEEP LEARNING MODEL

- A Convolutional Neural Network (CNN) is best suited for image processing
- Summary:
 - 4 convolutional layers
 - 3 pooling layers (2,2)
 - 4 dropout layers
- Dropout Layer: reduction by 25% to prevent overfitting in first 3 layers
- Loss function: binary cross-entropy to classify between 2 classes
- Optimizer: Adam shows better performance on deep learning models

```
model = Sequential()

model.add(Conv2D(32, kernel_size=(3,3), activation='relu', input_shape=(224,224,3)))
model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3,3), activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

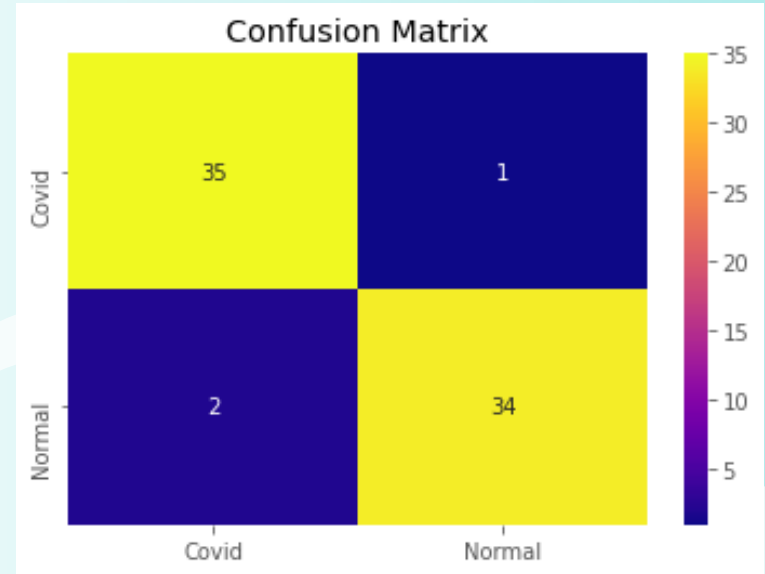
model.compile(loss=keras.losses.binary_crossentropy, optimizer='adam', metrics=['accuracy'])
```

Model: "sequential_3"

| Layer (type) | Output Shape | Param # |
|--------------------------------|----------------------|---------|
| conv2d_9 (Conv2D) | (None, 222, 222, 32) | 896 |
| conv2d_10 (Conv2D) | (None, 220, 220, 64) | 18496 |
| max_pooling2d_7 (MaxPooling2D) | (None, 110, 110, 64) | 0 |
| dropout_9 (Dropout) | (None, 110, 110, 64) | 0 |
| conv2d_11 (Conv2D) | (None, 108, 108, 64) | 36928 |
| max_pooling2d_8 (MaxPooling2D) | (None, 54, 54, 64) | 0 |
| dropout_10 (Dropout) | (None, 54, 54, 64) | 0 |
| conv2d_12 (Conv2D) | (None, 52, 52, 128) | 73856 |
| max_pooling2d_9 (MaxPooling2D) | (None, 26, 26, 128) | 0 |
| dropout_11 (Dropout) | (None, 26, 26, 128) | 0 |
| flatten_3 (Flatten) | (None, 86528) | 0 |
| dense_5 (Dense) | (None, 64) | 5537856 |
| dropout_12 (Dropout) | (None, 64) | 0 |
| dense_6 (Dense) | (None, 1) | 65 |
| Total params: 5,668,097 | | |
| Trainable params: 5,668,097 | | |
| Non-trainable params: 0 | | |

MODEL PERFORMANCE

- Based on the confusion matrix, the model has a high accuracy when predicting Covid-19 and Normal x-rays
- **97%** accuracy when predicting Covid-19 x-rays
- **94%** accuracy when predicting normal x-rays



PERFORMANCE INDICATORS

- **F1-Score** is used as the best performance indicator for binary classification problems
- **Precision** – correctly identified 97% of positive cases from all the predicted positive cases
- **Recall** – correctly identified 94% of positive cases from all the actual positive cases

| | precision | recall | f1-score | support |
|-------------------------|-----------|--------|----------|---------|
| Covid | 0.97 | 0.95 | 0.96 | 37 |
| Normal | 0.94 | 0.97 | 0.96 | 35 |
| accuracy | | | 0.96 | 72 |
| macro avg | 0.96 | 0.96 | 0.96 | 72 |
| weighted avg | 0.96 | 0.96 | 0.96 | 72 |
| Precision score: 0.9714 | | | | |
| Recall score: 0.9444 | | | | |
| <u>F1 Score: 0.9577</u> | | | | |

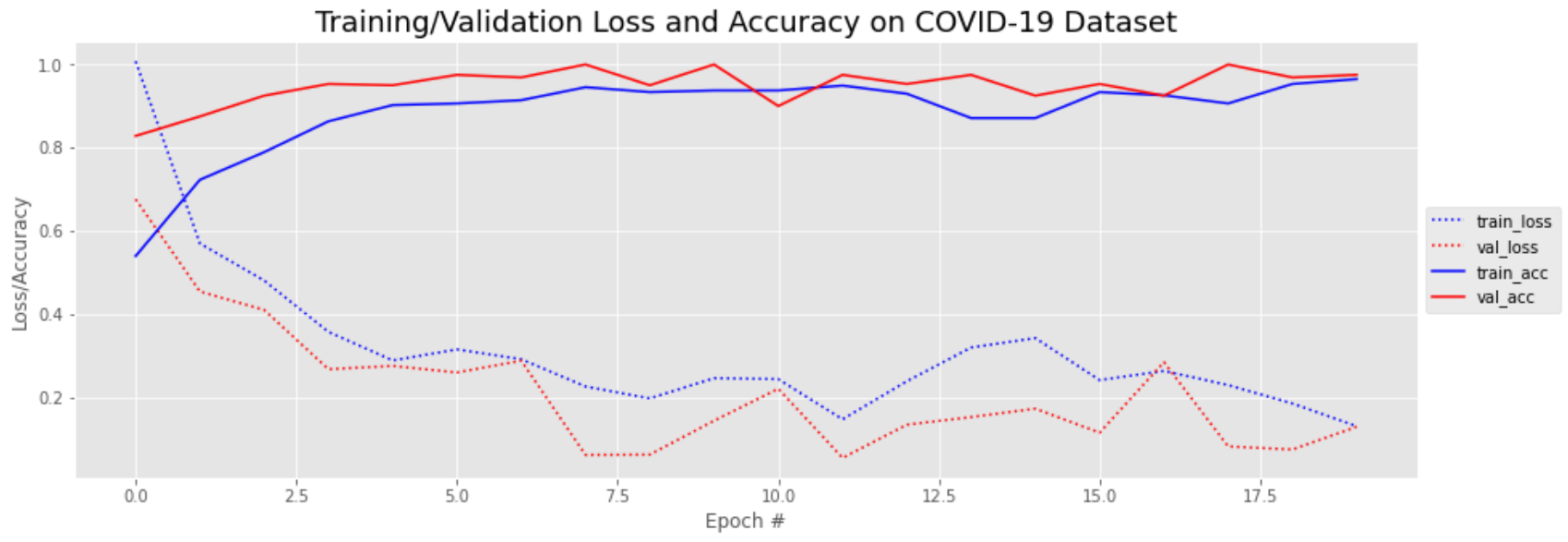
MODEL EVALUATION



96%

The CNN model yields an F1-score
of 96% on test data

TRAINING HISTORY



- Does the model overfit to the training data?
- Based on the plot history, the accuracy and loss follows the same trend, although does not overfit!
- Throughout each Epoch, there are slight variations in loss and accuracy

SUMMARY/NEXT STEPS



CNN MODEL

- Successful working model
 - 96% Accuracy
 - Does not overfit



LIMITATION

Data set is very small for large scale use; more data needed



NEXT STEP

- Using ImageNet to import a larger data set for training
- Compare model with a transfer learning model results

Link to GitHub Code



<https://github.com/vijay-ss/Covid19-Xray-Detection>

THANKS!

CONTACT

Does anyone have any questions?

Email: vijay.saddi@outlook.com

LinkedIn: <https://www.linkedin.com/in/vijay-ss/>

GitHub: <https://github.com/vijay-ss>

Portfolio: https://vijay-ss.github.io/Vijay_Portfolio/

