INTRODUCTION TO DIGITAL HEALTH AND ARTIFICIAL INTELLIGENCE IN MEDICAL APPLICATIONS

HW2 – Classification of Medical Image

Chew Zhi Qi (H34128412)

Dataset Description

Xray lung images of normal patients and patients with COVID

Train set

- 144 COVID and Normal images respectively

Validation set

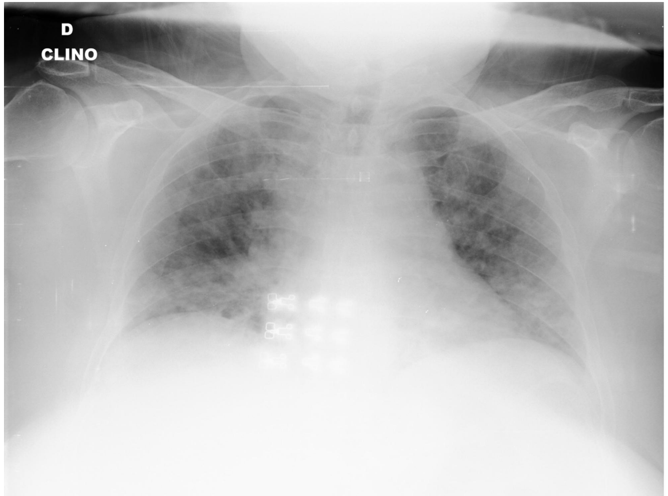
- 30 COVID and Normal images respectively

Prediction

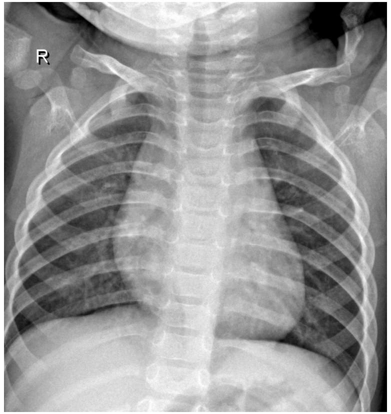
- 23 images

Task: Use deep learning models to predict whether a patient has COVID based on their Xray lung image

COVID



Normal



Data Preprocessing

Rescaling

- Normalize the pixel values of the images from the range [0, 255] to the range [0, 1]
- Faster convergence during training

Zoom Transformation

- Randomly zooms in on images by up to 20%
- Allows the model to learn features at different scales.

Horizontal Flip

- Randomly flips images horizontally
- Helps the model generalize better by learning from different orientations

Hyperparameter Tuning

Early Stopping

- Stop the training process when a metric stops increasing and use weights at the state where the metric was the highest
- Helps with preventing overfitting and saves computational resources

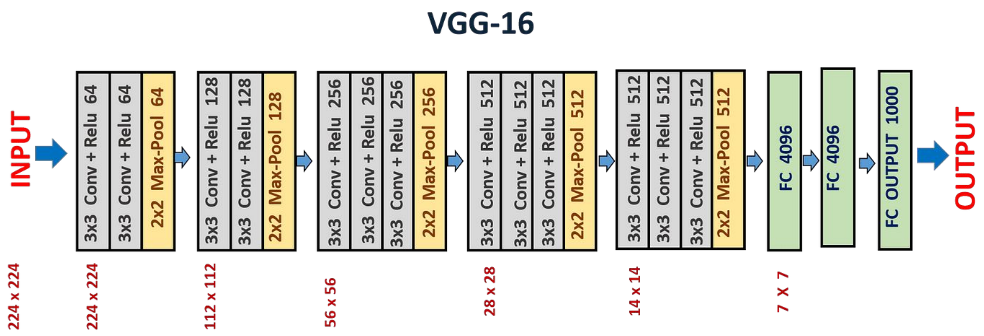
Learning Rate

- ReduceLROnPlateau
- Learning rate is decreased when a metric stops improving.
- Model is able to converge more effectively by taking smaller steps as it gets closer to the minimum of the loss function

Model 1: VGG16

Deep convolutional neural network model

- 16 layers of artificial neurons
- Uses convolution layers with a 3x3 filter and a stride 1 that are in the same padding and maxpool layer of 2x2 filter of stride 2.
- Pre-trained version is trained on over one million images from the ImageNet visual database



Model 1: VGG16

On top of the pretrained model, additional layers are added to adapt the model for the binary classification task.

GlobalAveragePooling2D Layer:

- Performs an average pooling operation, reducing the spatial dimensions
- Significantly reduces the output size by averaging each feature map

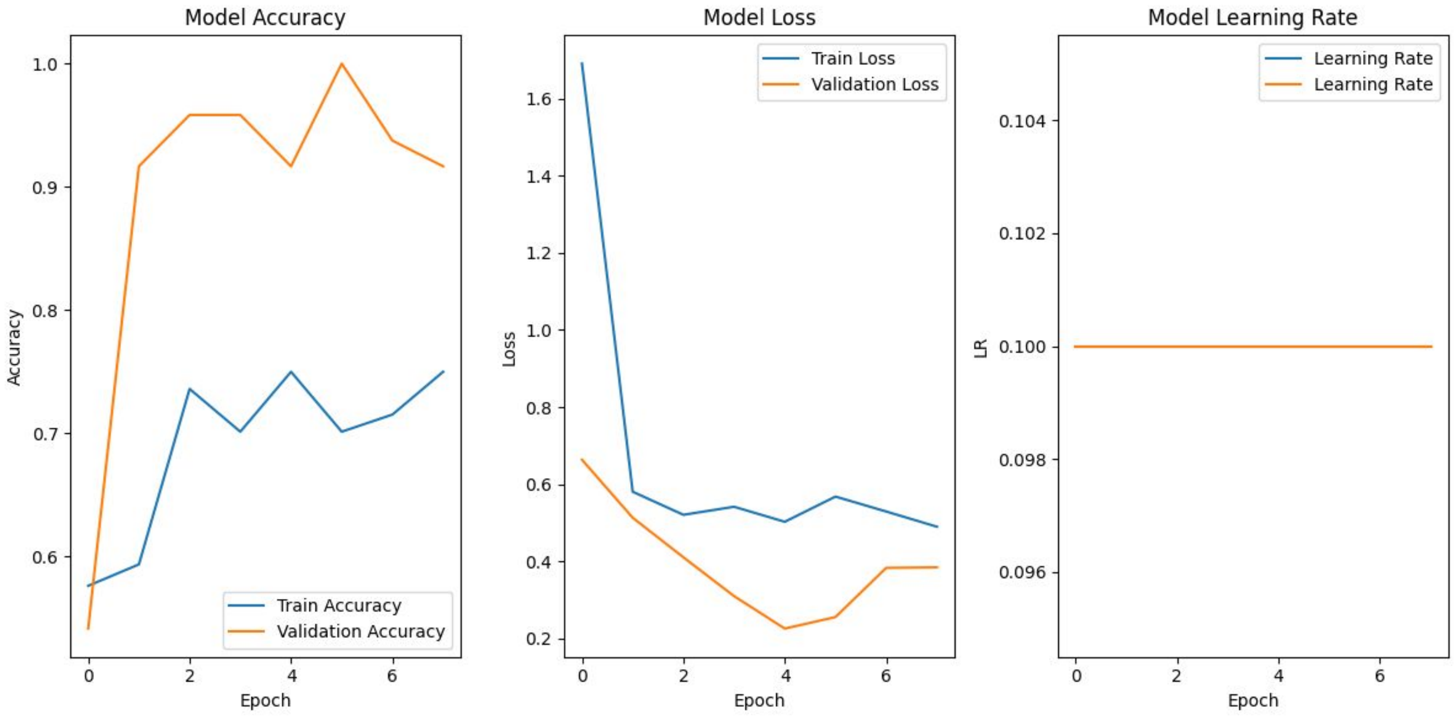
Dense Layers with Dropout:

- Adds a dense layer with 64 units and ReLU activation to learn complex patterns from the pooled features.
- Dropout regularizes the model by randomly dropping 50% of the neurons during training, preventing overfitting and improving generalization.

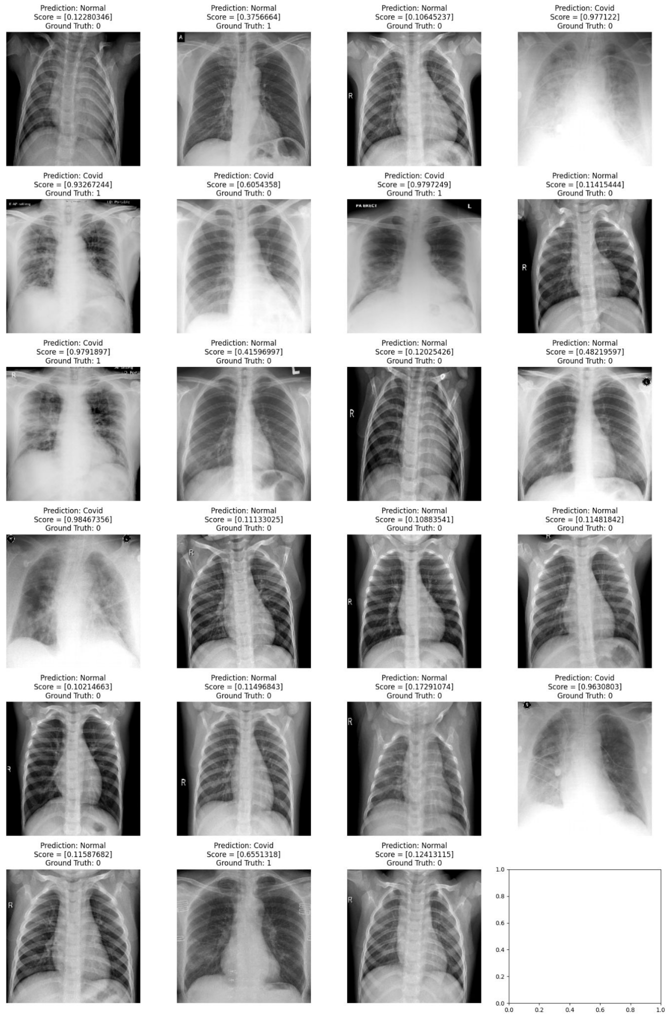
Final Dense Layer (Binary Classification):

- Outputs a single probability score indicating the likelihood of the input image belonging to the positive class.

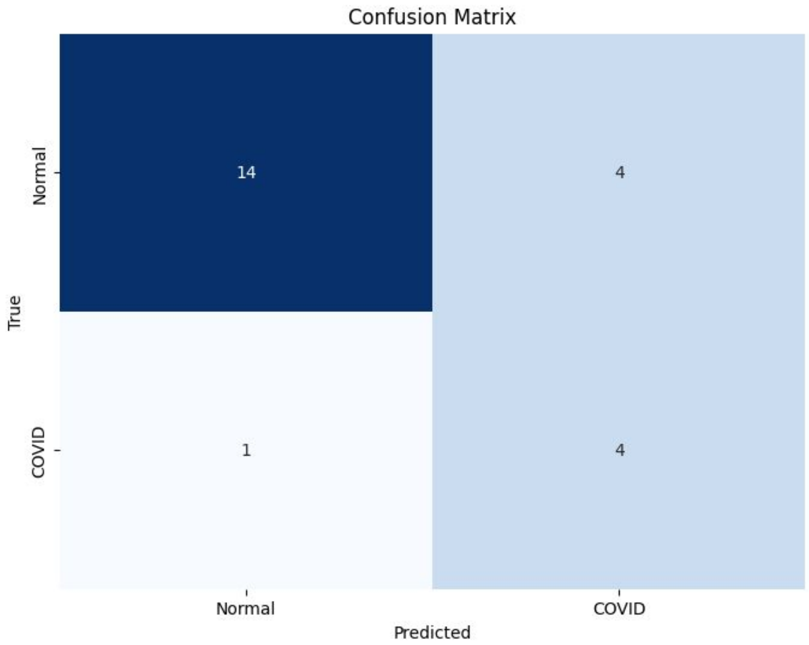
VGG16 Training Performance

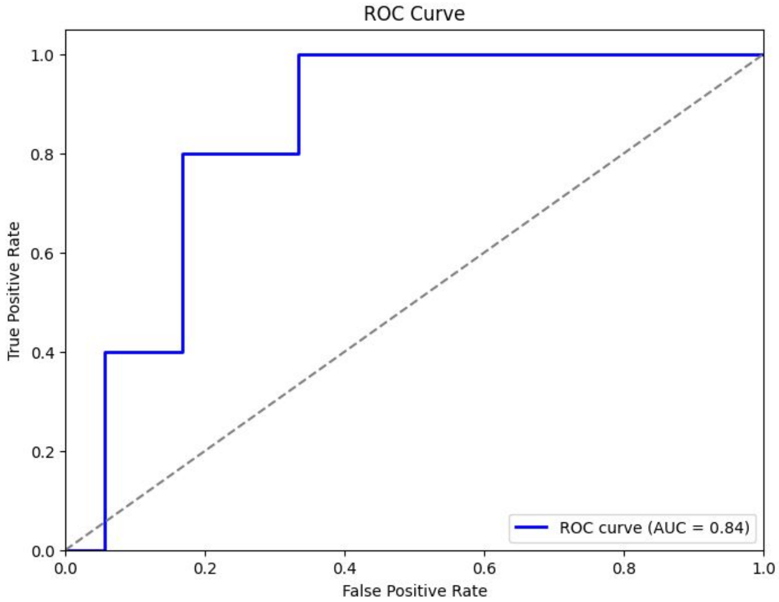


VGG16 Performance on Test Set



VGG16 Performance on Test Set





Model 2: Self Designed CNN

Convolutional Layers:

- Five convolutional layers with ReLU activation function and a 3x3 filter.
- Each convolutional layer is followed by a max pooling layer with a 2x2 pool size and stride of 2.

Flattening and Dense Layers:

- The feature maps are flattened into a vector.
- Two dense layers with ReLU activation, having 128 and 64 units respectively.

Output Layer:

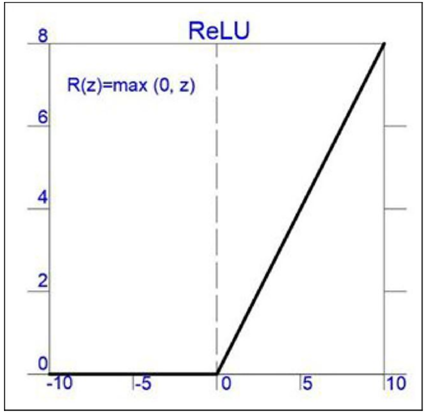
- Final dense layer with sigmoid activation ('sigmoid') for binary classification (outputting a probability score).

Optimizer and Loss Function:

- Optimized using Adam (Adaptive Moment Estimation) optimizer
- Loss function is binary cross-entropy

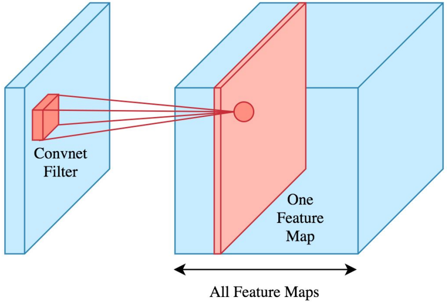
ReLU

- Most popular activation function (function that defines the output of a node given an input and introduces the property of nonlinearity into the model) for training convolutional layers and deep learning models
- Easy to implement and less time consuming compared to sigmoid

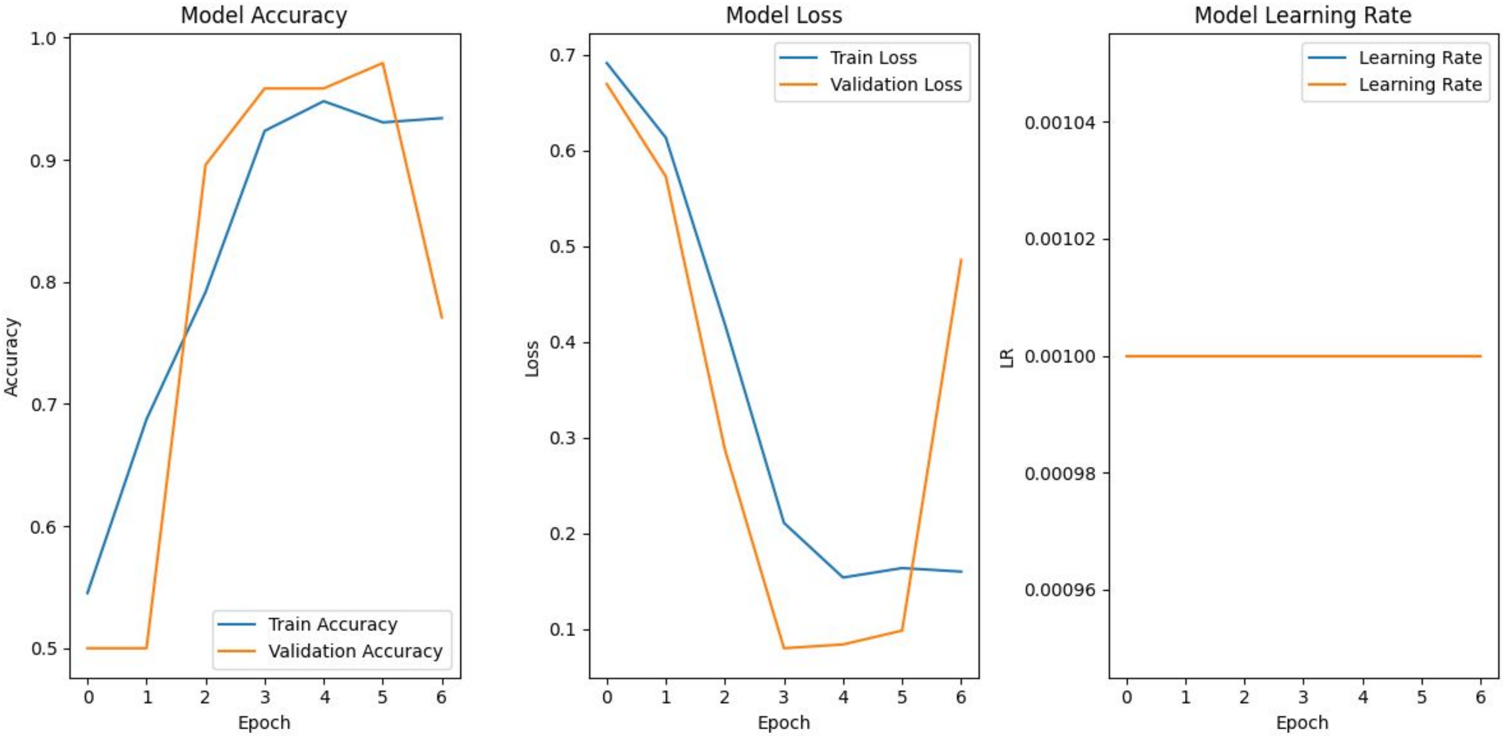


Feature Maps

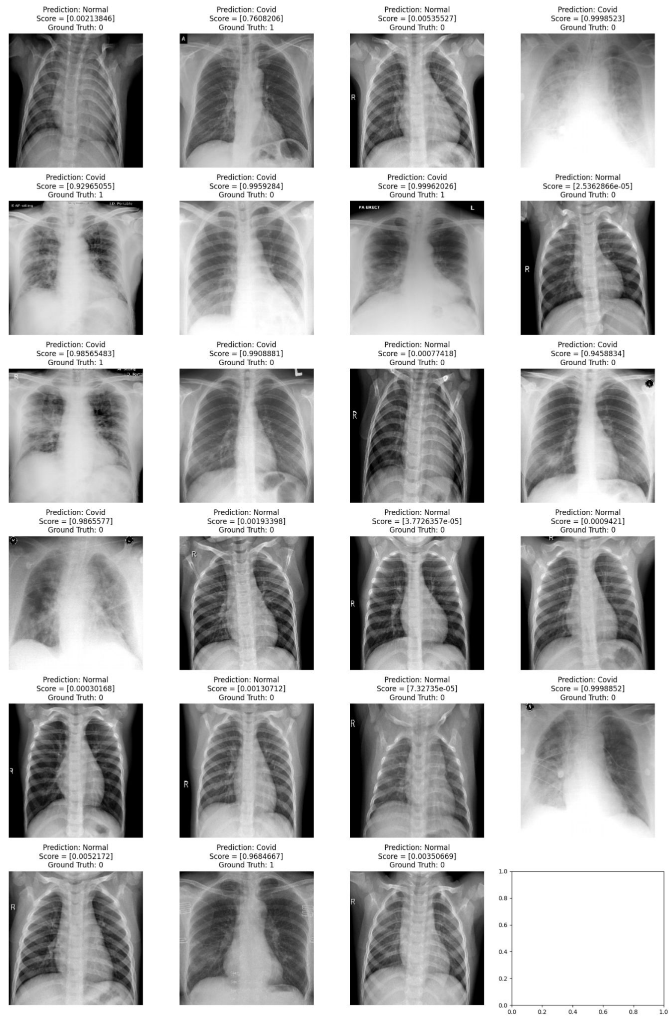
- Two-dimensional array generated from the application of convolutional filters/kernals to an input image or a previous layer's feature map.



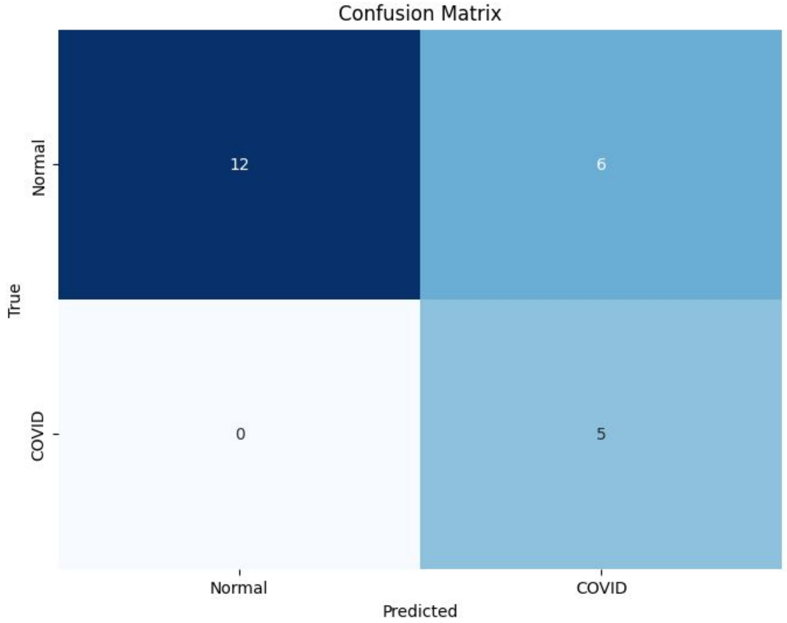
CNN Training Performance

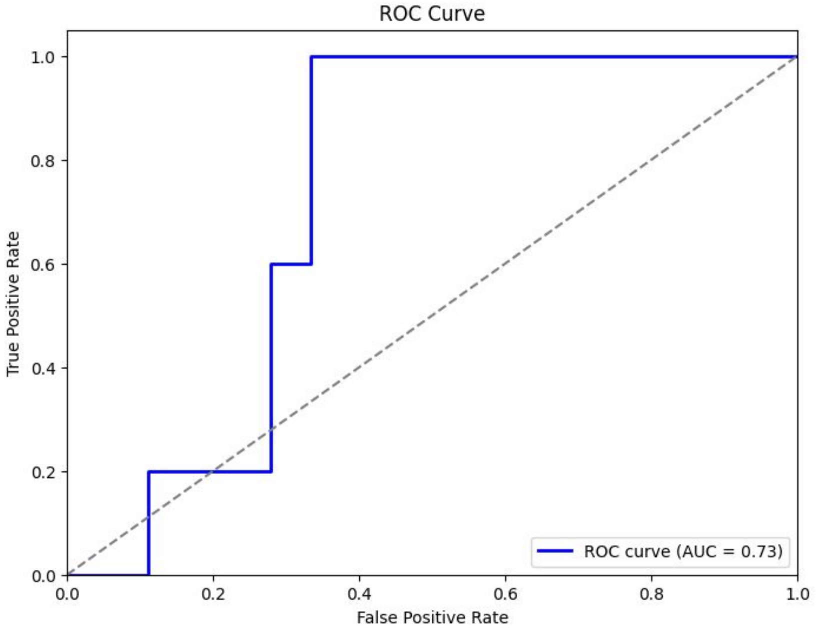


CNN Performance on Test Set



CNN Performance on Test Set





Performance Comparision

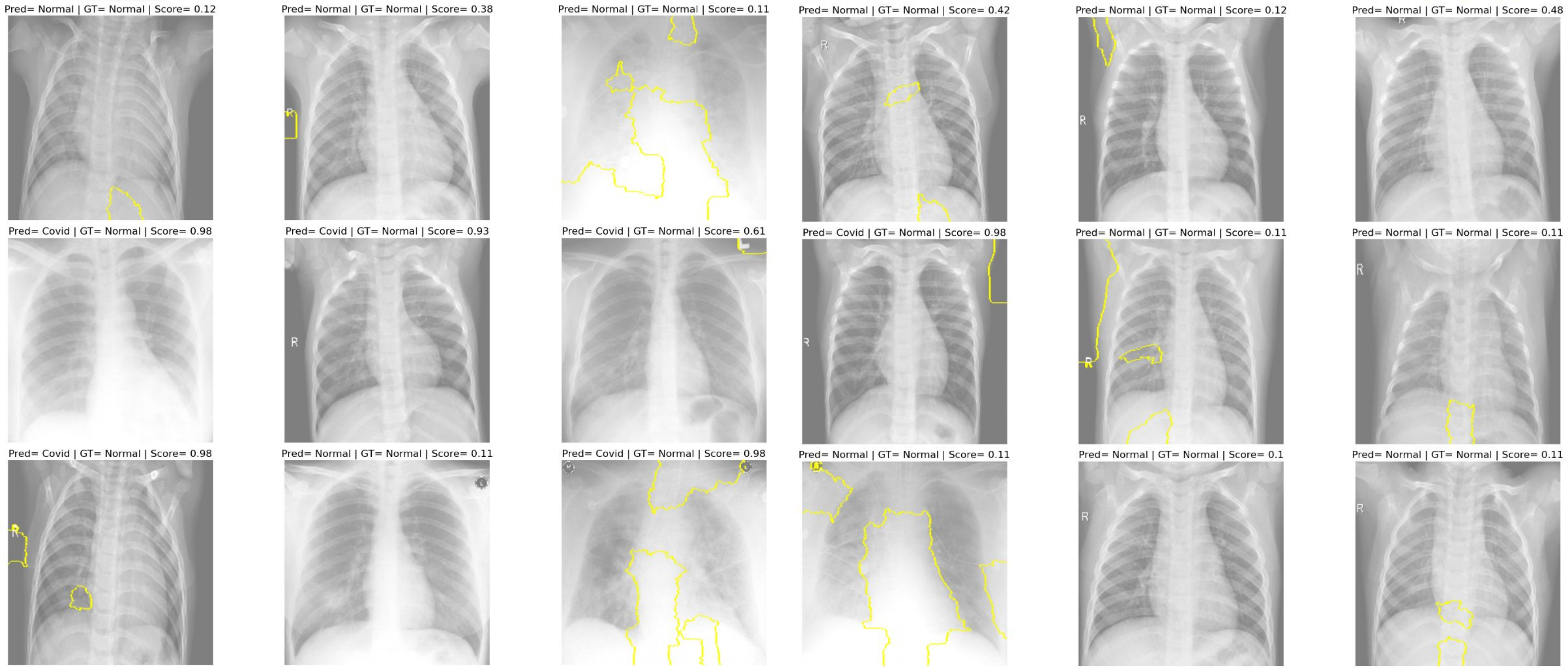
Metrics	Formula	Focus	VGG16	CNN
Accuracy	(TP+TN)/ (FP+FN+TP+TN)	Overall correctness	0.78	0.74
Precision	TP/ (TP+FP)	Correctness of positive predictions	0.50	0.45
Sensitivity / Recall	TP/ (TP+FN)	Ability to find all positive instances	0.80	1
Specificity	TN/ (TN+FP)	Ability to find all negative instances	0.78	0.67
F1-Score	2× ((Precision*Sensitivity)/(Precision+Sensitivity))	Balance between precision and sensitivity	0.62	0.62
ROC AUC	Integration	Overall performance across all classification thresholds	0.84	0.73

Model Interpretability

LIME (Local Interpretable Model-agnostic Explanations)

- Technique designed to explain the predictions of any machine learning model by approximating it locally with an interpretable model.
- **Model-Agnostic**: LIME is not dependent on the type of model. It can be used with any classifier or regressor, whether it's a neural network, decision tree, or any other type.
- **Local Explanations**: LIME focuses on explaining individual predictions. It provides insight into why the model made a specific prediction for a given input.
- Benefits
    - **Transparency**: Helps in understanding complex models by breaking down predictions into understandable components.
    - **Debugging**: Identifies which features are driving predictions, useful for debugging and improving models.
    - **Trust**: Increases trust in model predictions by providing explanations
- Drawbacks
    - **Computationally Intensive and Time-Consuming**: Involves training a local surrogate model for each instance, which can be computationally expensive and time-consuming

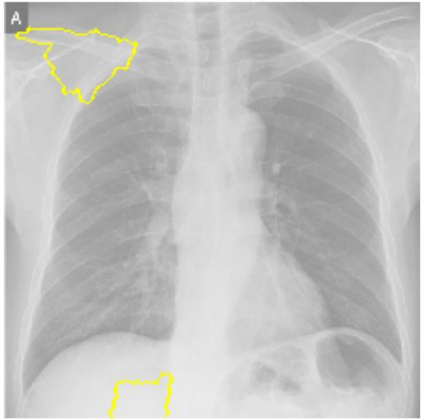
VGG16 LIME Results on Normal Patients



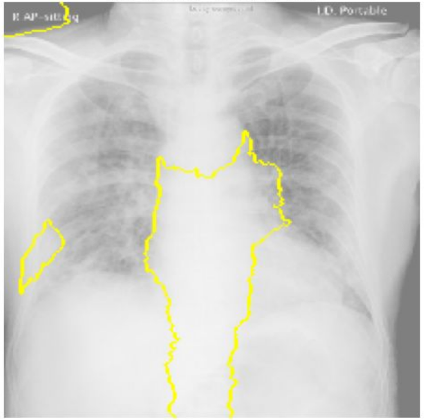
Highlighted areas = important features that influenced the model’s prediction

VGG16 LIME Results on COVID Patients

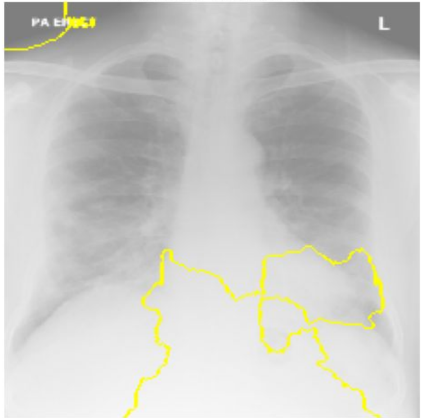
Pred= Normal | GT= Covid | Score= 0.12



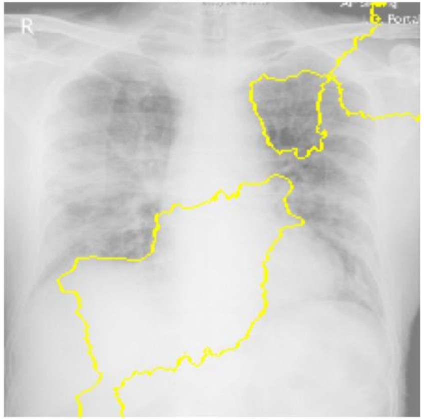
Pred= Normal | GT= Covid | Score= 0.38



Pred= Normal | GT= Covid | Score= 0.11

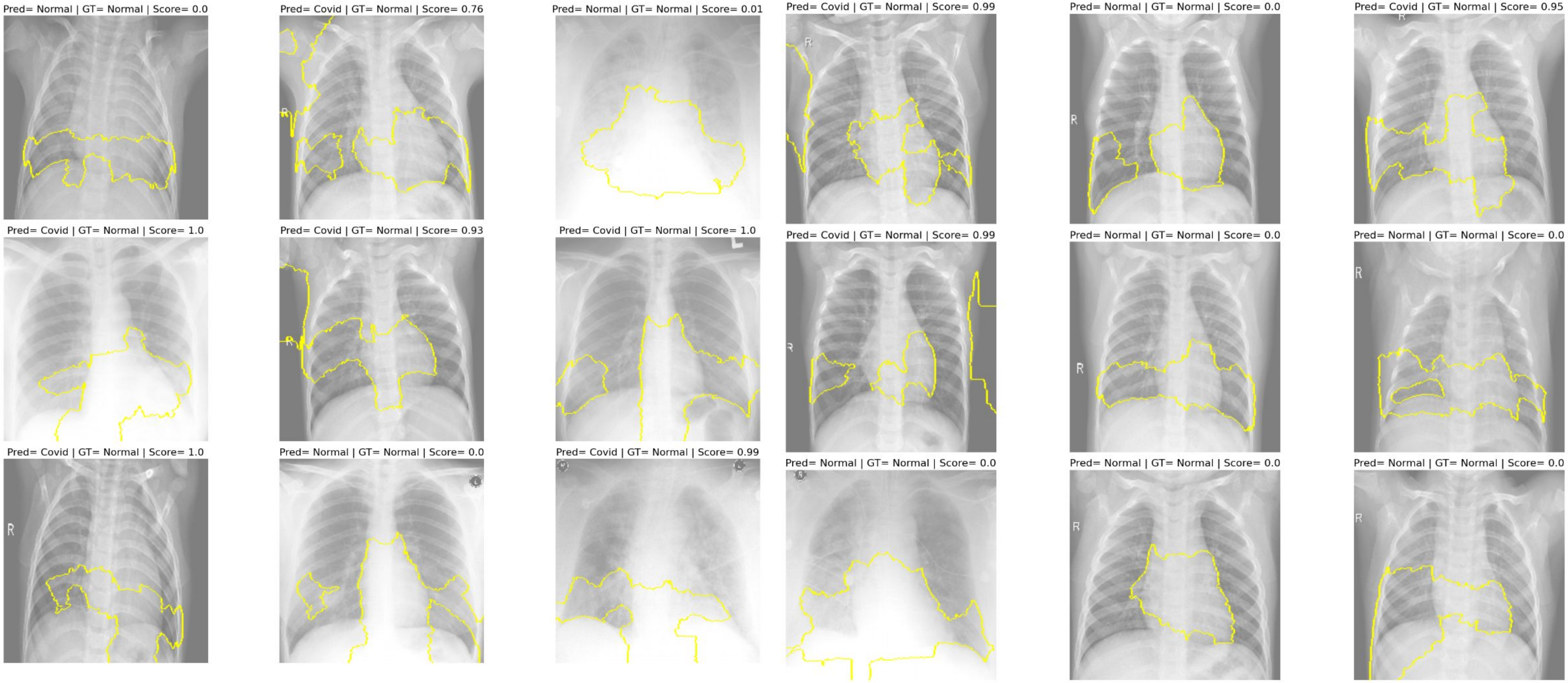


Pred= Covid | GT= Covid | Score= 0.98

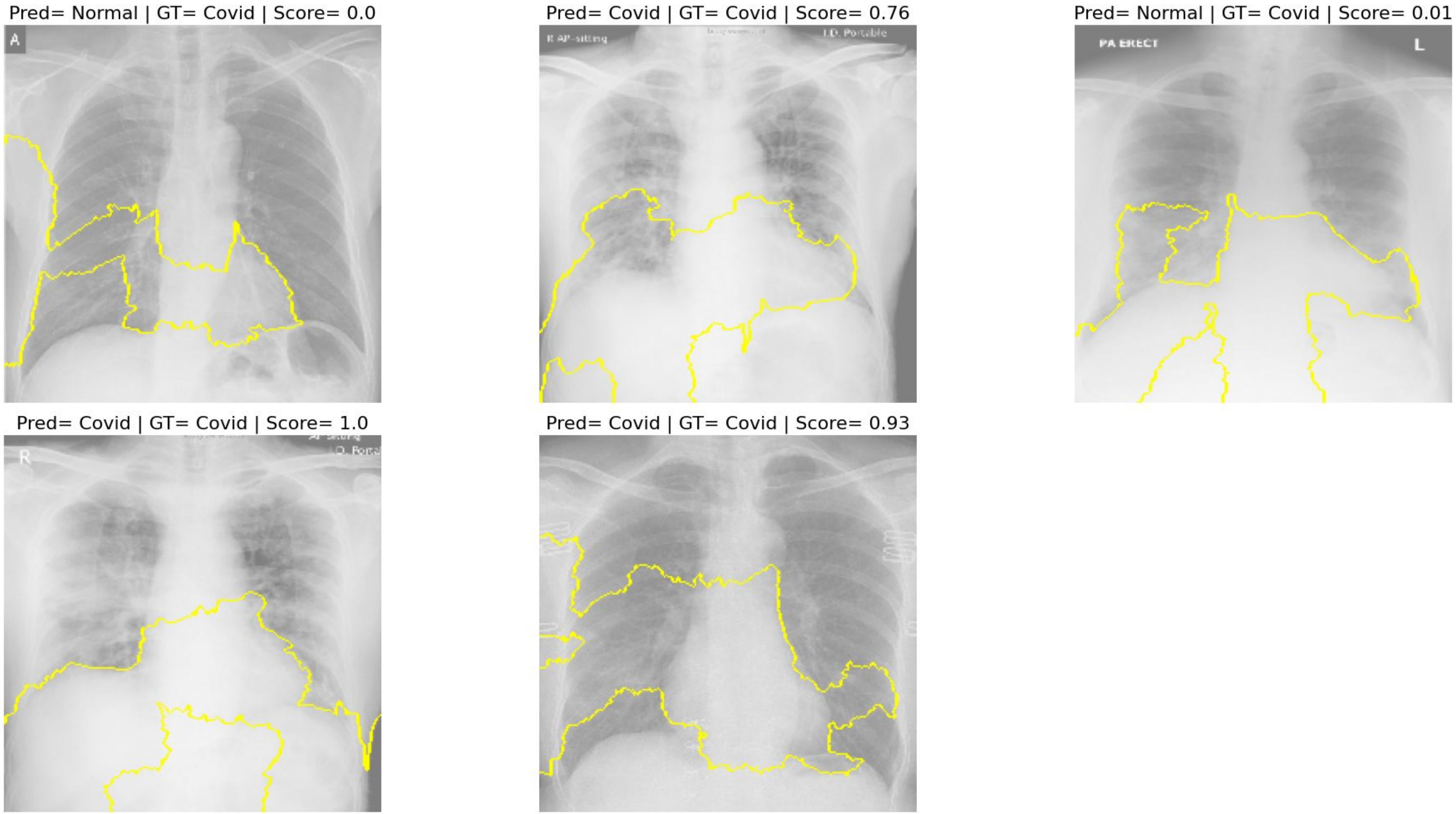


Pred= Covid | GT= Covid | Score= 0.93

CNN LIME Results on Normal Patients



CNN LIME Results on COVID Patients



Conclusion from Model Performance and LIME Results

Model Performance

- VGG16 performed better generally (higher ROC AUC, accuracy)
- However, recall score for CNN is higher than VGG16
    - Recall is an important metric to look at
    - For COVID19 detection, false negative should be avoided as much as possible (someone who has COVID is not diagnosed)

LIME Results

- Important features detected by VGG16 are more inconsistent (often consisting of areas outside the lungs or not having a specific area at all)
- For CNN, the important features identified are often the lower part of the lungs

Conclusion

- CNN model has more potential to do better if more hyperparameter tuning is done, specifically in the context of Xray image classification.
- VGG16 is trained on ImageNet dataset (>1 million images and >20000 categories), hence when training it on the small xray dataset, overfitting most likely occurred.