Predicting COVID-19 Positive Cases From Chest X-Ray Images

Springboard Data Science Track - Capstone Project
Fellow: George Pinto
Mentor: Srdjan Santic

Problem Identification

- Medical images need to be interpreted because they are not self-explanatory.
- Medical images vary considerably, even within a particular exam type. Anatomical structures can camouflage features of clinical interest
- These complexities can lead to interpretation errors. **Clinicians do make mistakes** (Berlin, 2005, 2007, 2009).
- In radiology alone, estimates suggest that in some areas, there may be up to a **30% miss rate and an equally high false positive rate**.

https://www.ncbi.nlm.nih.gov/pmc/articles

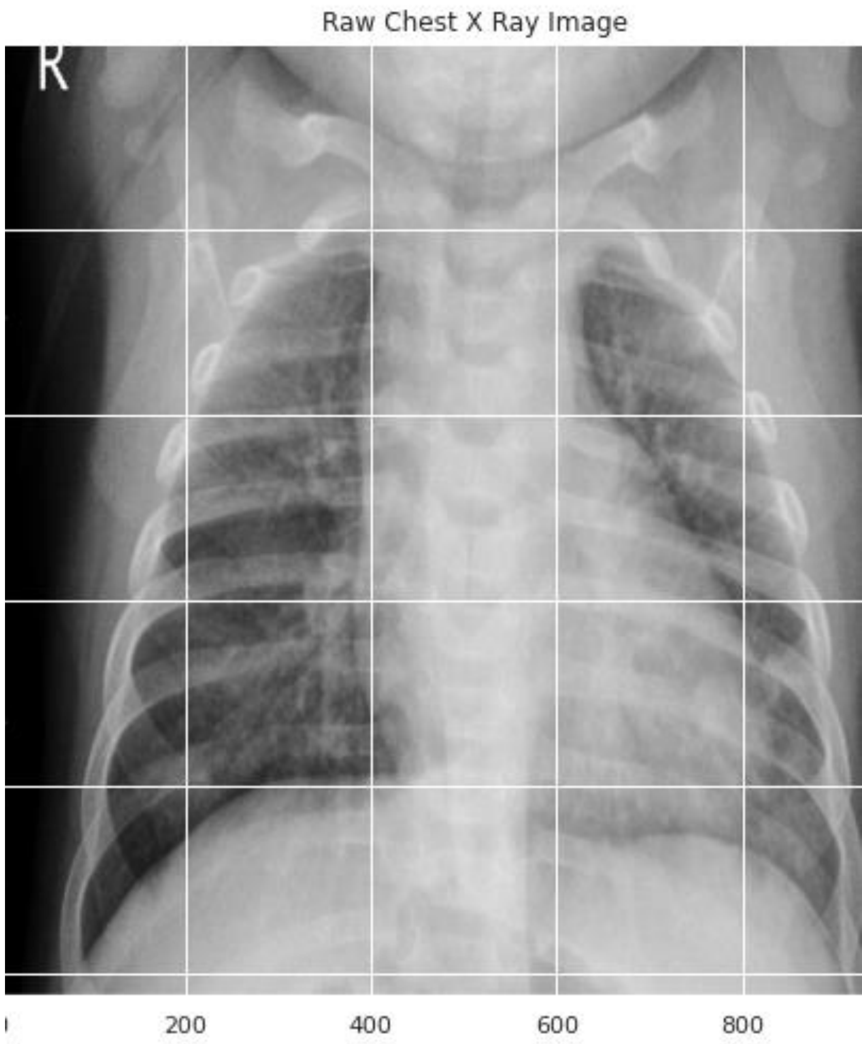
Problem Identification : Dataset

- How do we deal with a real situation in which we need medical images interpreted quickly and efficiently classify a novel disease?
- 219 COVID-19 positive images
- 1341 normal images
- 1345 viral pneumonia images.
- Dataset from Kaggle – not many sources of Covid-19 X-Rays available since this is a novel disease
- A team of researchers from Qatar University (Doha, Qatar) and the University of Dhaka in Bangladesh
- Collaborators from Pakistan and Malaysia
- Collaborating medical doctors

https://www.sirm.org/category/senza-categoria/covid-19/

https://github.com/ieee8023/covid-chestxray-dataset

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia



Recommendations and Key Findings

Category	Model	Precision	Recall	F1	Accuracy
Covid-19	VGG16	0.93	1.00	0.96	0.97
Normal	VGG16	1.00	0.92	0.96	0.97
Viral Pneumonia	VGG16	1.00	1.00	1.00	0.97
Covid-19	ResNet50	0.92	0.92	0.92	0.85
Normal	ResNet50	1.00	0.69	0.82	0.85
Viral Pneumonia	ResNet50	0.72	0.93	0.81	0.85
Covid-19	DenseNet121	1.00	1.00	1.00	0.93
Normal	DenseNet121	1.00	0.77	0.87	0.93
Viral Pneumonia	DenseNet121	0.82	1.00	0.90	0.93

- **Use State of the Art VGG16 Deep Learning Model trained on a curated library of approx.14 million images (Imagenet) to assist with classification and interpretation**
- Covid-19 presents a good case to expedite and facilitate image interpretation in medical settings with the use of AI
- We can use Transfer learning and take advantage of the massive amount of features this model learned and use the model to classify Covid-19 positive X-Ray Images
- The key metric to optimize is recall, because we want to maximize the number of images predicted as Covid-19 that are correctly classified as Covid-19
- Highlighted on the left, the VGG16 Model has a recall of 1 (or 100%) for Covid-19, 100% for Viral Pneumonia and the highest recall for normal patients obtained between the three tested deep learning models.
- **To compare to the human error of up to 30% in our opening page the miss rate of the neural network is 0**

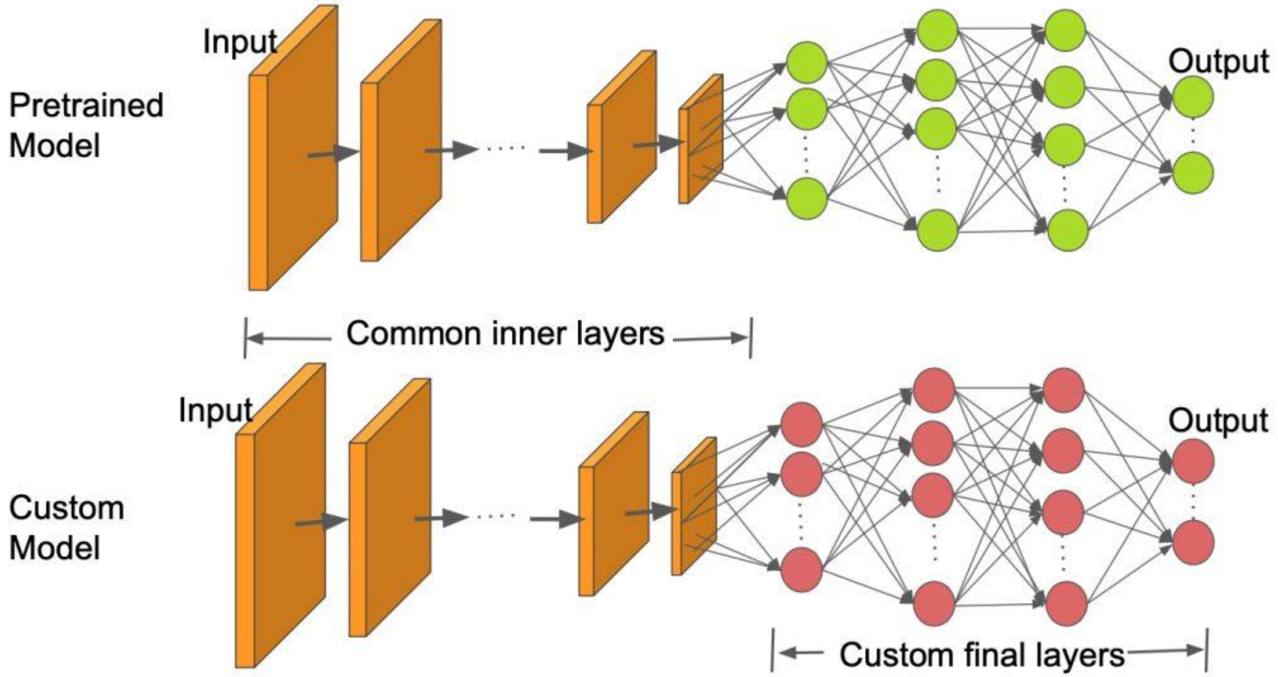
Recommendations: Advantages of Disruptive Technologies

- Cloud computing and AI are disruptive technologies
- We can leverage the power of these two technologies to streamline the process of image interpretation
- Tasks that have been historically time consuming for radiologists, like image processing are automated in the AI pipeline.
- The model is reliably served through a cloud API that is flexible and scalable
- Images can be uploaded from anywhere (desktop, mobile, tablet) to make a request and obtain a prediction in minutes from the cloud

Modeling Results and Analysis: Transfer Learning

- Class Imbalance
  - Small number of Covid-19 images resulted in Class imbalance
  - necessary to under-sample the other classes.
  - 219 images for each Category matching the Covid-19 class
- Transfer learning is the best choice in terms of model architectures
  - Deep learning models require a lot of data to train, and we don't have a lot of data
  - The features necessary for classification are instead extracted from a state-of-the-art image dataset (Imagenet)
  - Pre-trained extraction layers from 3 state-of-the-art models
- The head (customized final layers) is built specifically for the task at hand and has 3 outputs representing our 3 categories
  - Each model requires a unique head
- Training
  - The model trains using randomly initialized weights on the head
  - The head learns the more specialized features of the actual classes
  - The pre-train features that we are transferring are the harder to learn more generalizable features and they have already been learned so the weights on this section are frozen
- Metrics are used to decide on the best model.
- Pre-trained models provide an excellent alternative because Learning such vast and rich image features is time and cost prohibitive for most researchers/data scientists

Modeling Results and Analysis: an Example of Transfer Learning



(Head)

Source:

https://learnopencv.com/image-classification-using-transfer-learning-in-pytorch/

Modeling Results and Analysis: Tools and Models

- Data manipulation libraries: Pandas, NumPy, matplotlib
- Deep Learning Framework: TensorFlow 2
- GPU(s)
- Well known state of the art models chosen, all pretrained on ImageNet
  - VGG16
  - ResNet50
  - DenseNet121
- Head layers were customized trying different architectures consisting of an average pooling layer, dense layers, dropout and SoftMax activation
  - If the model underfits then more complexity is needed on the final custom layers (more neurons or units on each layer or simply more layers)
  - if the model overfits, then we need less

  https://towardsdatascience.com/four-common-types-of-neural-network-layers

Modeling Results and Analysis: Learning Rate Parameters and Tuning

- Optimizers and learning rate
  - Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses [minimize loss and maximize learning]
  - The Adam optimizer was chosen as it can be an excellent choice for deep learning models focusing on computer vision.
  - A learning rate scheduler was used to plot learning rates on a log scale to determine the best learning rate ranges for each model (to achieve the best loss)
  - It was not necessary to modify the default settings on the Adam optimizer (learning rate=0.001)

  https://towardsdatascience.com/optimizers-for-training-neural-networks

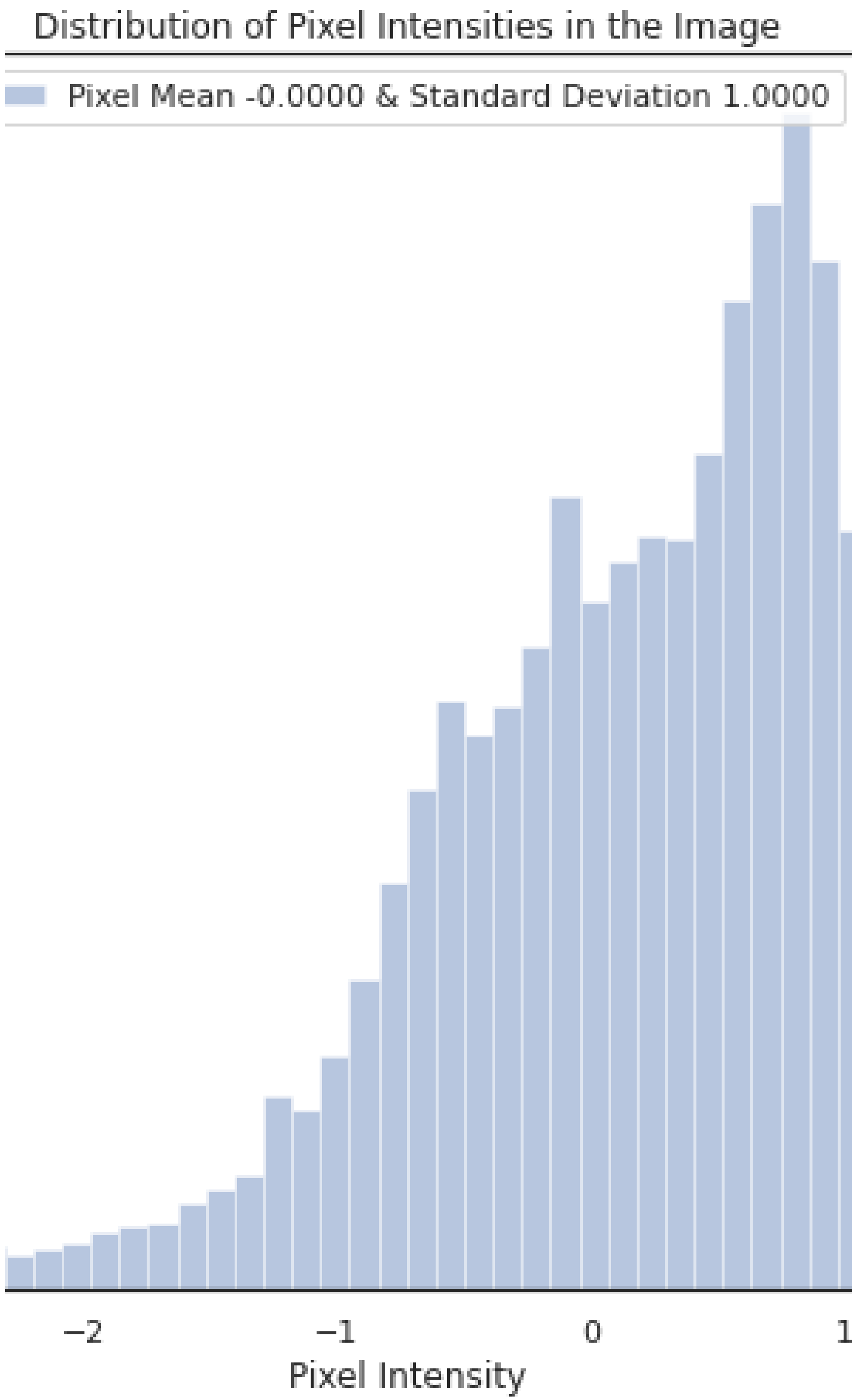
- Callbacks
  - A callback or function to save the best weights of the model was used
  - This callback monitored validation accuracy (or improvement on results of unseen samples) and saved the best learned weights (highest achieved validation accuracy)
  - A reduce on plateau callback was used, this function reduces the value of the learning rate if the model stops making improvements on the validation loss after a custom number of epochs (learning cycles) for a customized reduction amount
  - Recommended settings for the Adam optimizer were used on this function
  - Lastly an early stopping callback was used to stop the model if validation accuracy stopped increasing for a custom number of epochs

  https://www.kdnuggets.com/2019/08/keras-callbacks-explained-three-minutes

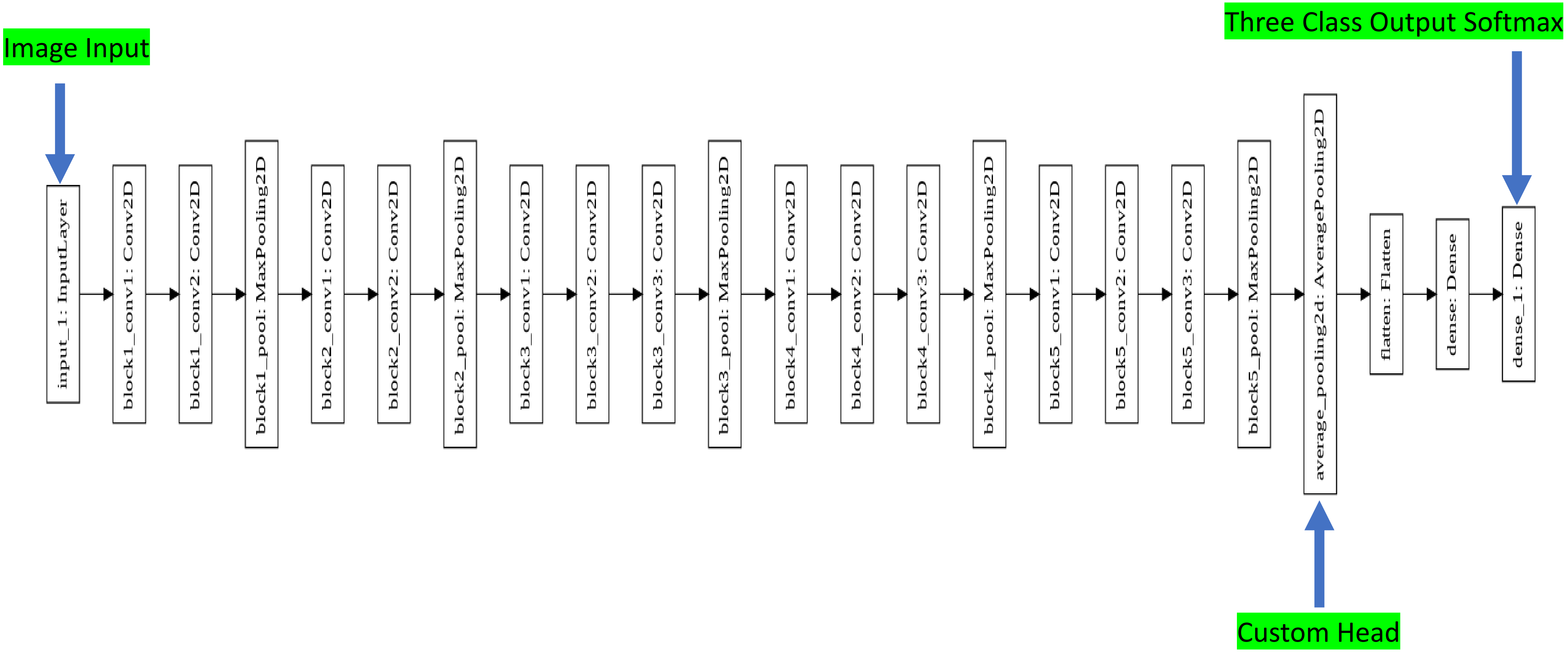
Modeling Results and Analysis: Image Processing

- Normalizing Image Inputs
  - Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network
  - Data normalization is done by subtracting the mean from each pixel and then dividing the result by the standard deviation
  - For image inputs we need the pixel numbers to be positive, so we might choose to scale the normalized data in the range [0,1]
  - When splitting the data into training, validation and testing sets, the normalization statistics (mean and standard deviation) should come from the training Data
- Dimensionality Reduction
  - Collapse the RGB channels into a single gray-scale channel
  - This approach makes the learning problem more tractable

  https://becominghuman.ai/image-data-pre-processing-for-neural-networks

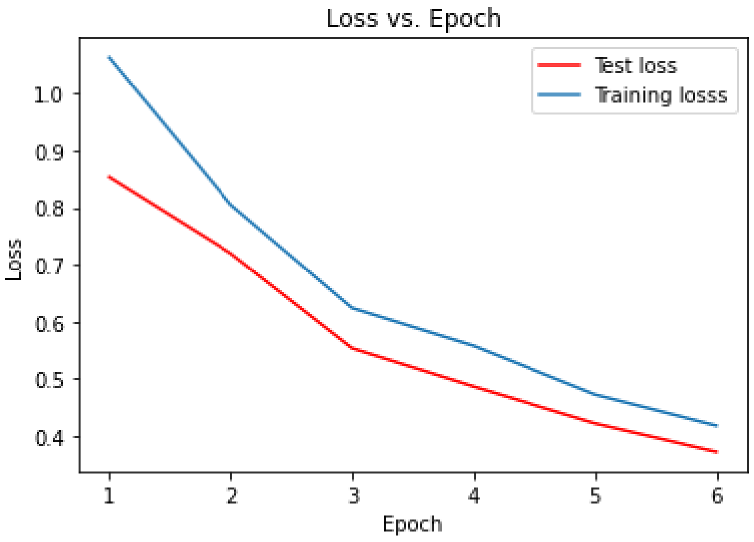


Modeling Results and Analysis: Example Model – Chosen Model VGG16 with Custom Head



Modeling Results and Analysis: Loss

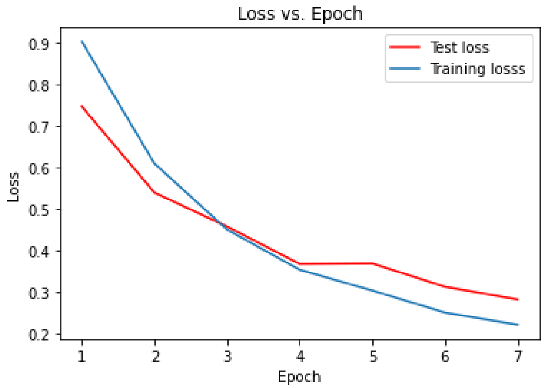
LOSS: PROCEED WITH CAUTION!



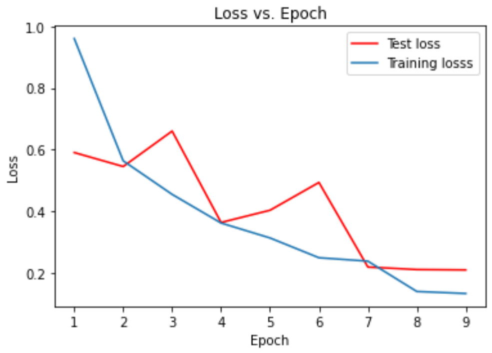
VGG16

- Loss (error between predictions and real target values)
  - We want the loss to go down
  - This means predictions are getting better
- Loss Function
  - Categorical Cross Entropy is the loss function (function used for multiple category problems) to calculate the loss and derivative (to minimize the loss)
  - The activation used on the output layer of the network is SoftMax (used to get probabilities for each class)
- Training and Validation
  - At first, validation (unseen samples) was performing better than training. See image on the left the red validation line is under the blue training line. This is normally reversed.
  - This can be dangerous for unseen samples because the model could be overlearning on the validation set and wildly fluctuating on results on new unseen data.
  - Initially some regularization (as dropout) was used on the ResNet50 and VGG16 models, but the data is truly too little, and regularization did not allow the model to learn properly
  - Best to pull back on the dropout. Other forms of regularization, like data augmentation would probably make this worse as well
  - Dropout was removed for VGG16 and ResNet50 and pulled back from 0.5 to 0.2 on the DenseNet121.
  - Model complexity was increased on ResNet50 and DenseNet121
  - New architectures were trained and validated

BEST

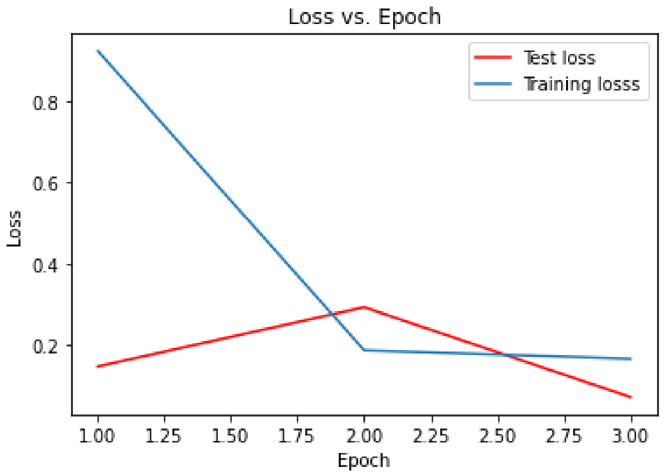


VGG16



RESNET50

DENSENET121

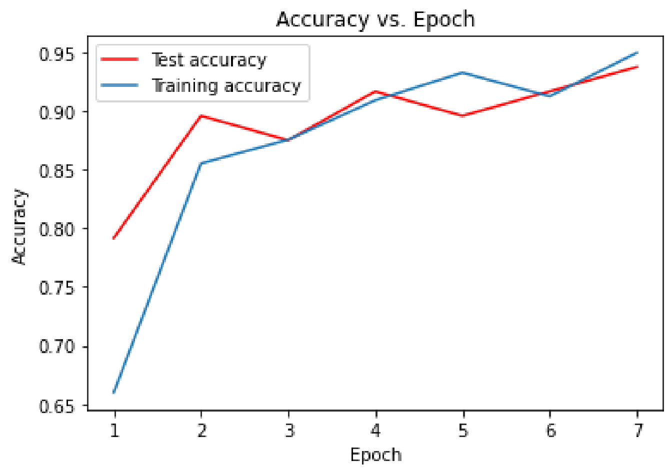


Modeling Results and Analysis : Loss

The best model training is denoted by training and validation losses converging as close together as possible and dropping down smoothly as the model learns.

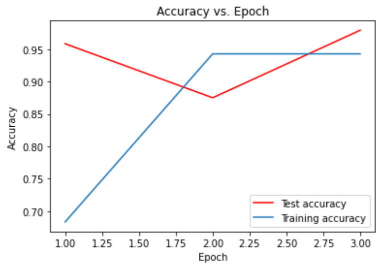
Modeling Results and Analysis: Accuracy

- The same principles go for training and validation accuracy
- We want to see the training and testing accuracy converge as much as possible and to go up
- Note the last few epochs on the VGG16 and how training and validation are going up together, increasing together until the early stopping on epoch 7 on the x axis



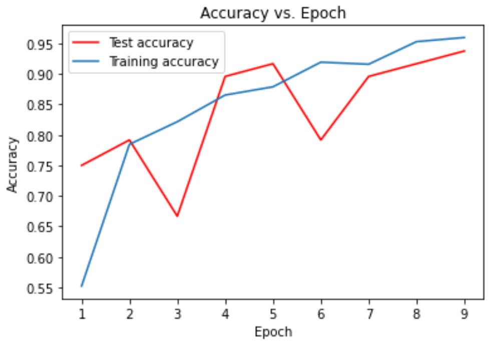
VGG16

BEST

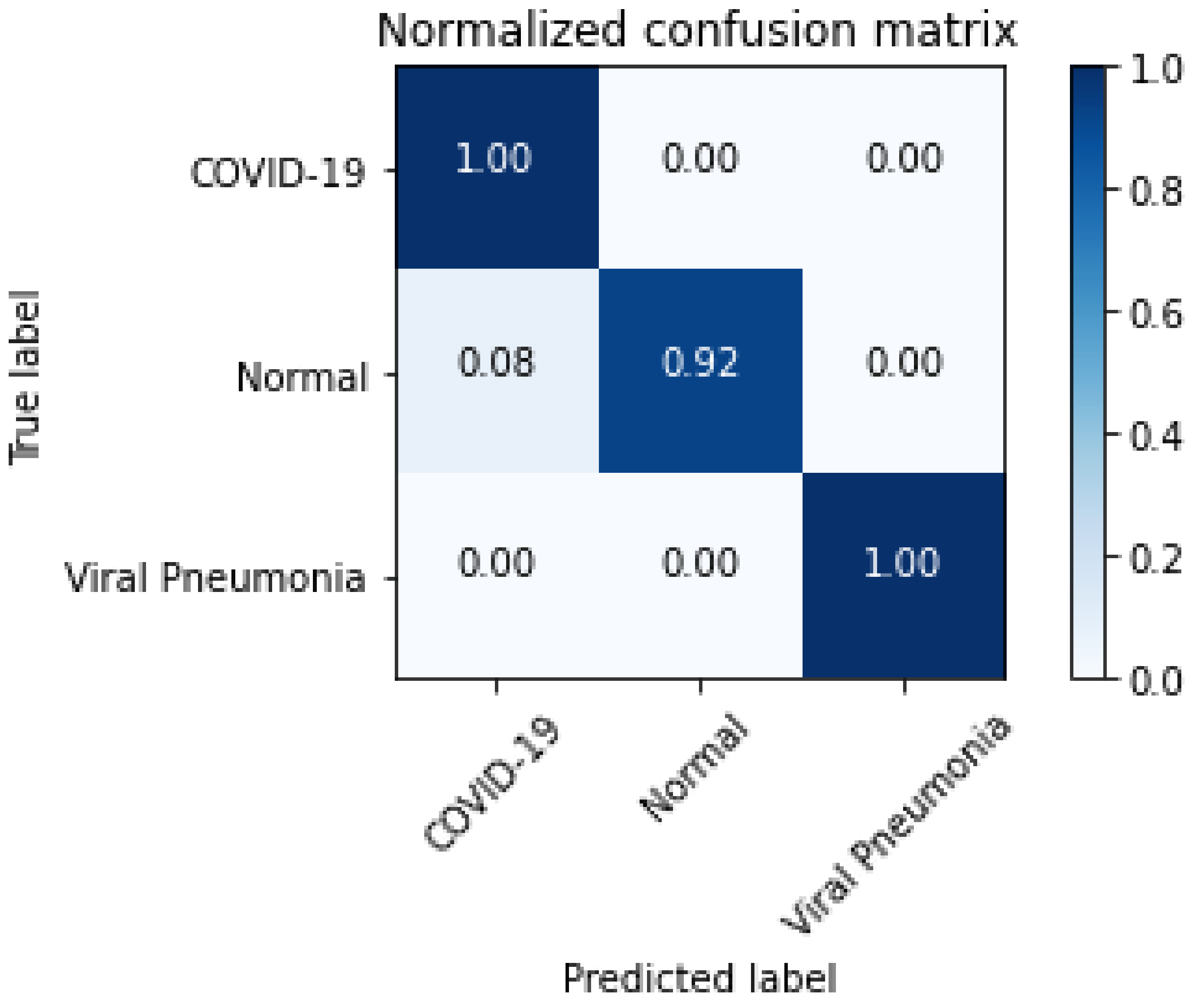


DenseNet121

ResNet50



Modeling Results and Analysis: VGG16 Performance on Unseen Test Data



Modeling Results and Analysis: VGG16 Make prediction Request and get Prediction - TensorFlow Serving

Input Chosen X-Ray Image

```
import json
data = json.dumps({"signature_name": "serving_default", "instances": input_image.tolist()})
import requests
headers = {"content-type": "application/json"}
r = requests.post('http://localhost:8501/v1/models/covid_vgg16_no_reg_grad:predict', data=data, headers=headers)
j = r.json()
pred = np.array(j['predictions'])
pred = pred.argmax(axis=1)
class_names = ['COVID-19', 'Normal', 'Viral Pneumonia']
pred = [class_names[i] for i in pred]
print('The prediction for this image is: ',pred[0])
The prediction for this image is:  COVID-19
```

Get Prediction

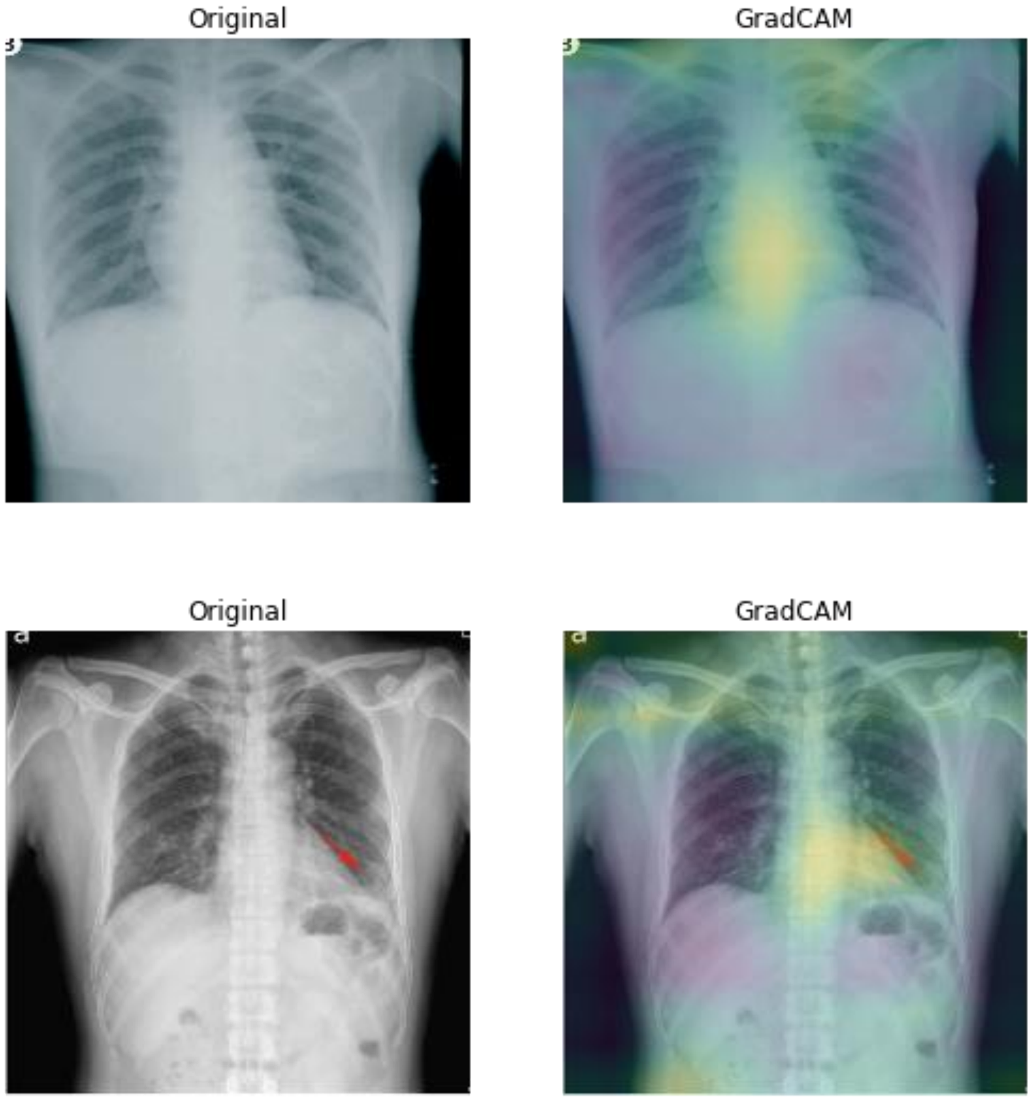
Model Interpretability: GradCAM

- Classification scores on this final VGG16 model were very promising but on what regions of the images was the model focusing its activations?
- We can use the GradCam algorithm to explore the activations
- An algorithm that can be used visualize the class activation maps of a Convolutional Neural Network (CNN), thereby allowing you to verify that your network is "looking" and "activating" at the correct location source: https://www.pyimagesearch.com/2020/03/09/grad-cam-visualize-class-activation-maps-with-keras-tensorflow-and-deep-learning/

Model Interpretability: GradCAM Covid-19 X-ray Images

Activations are the strongest in brighter yellow regions

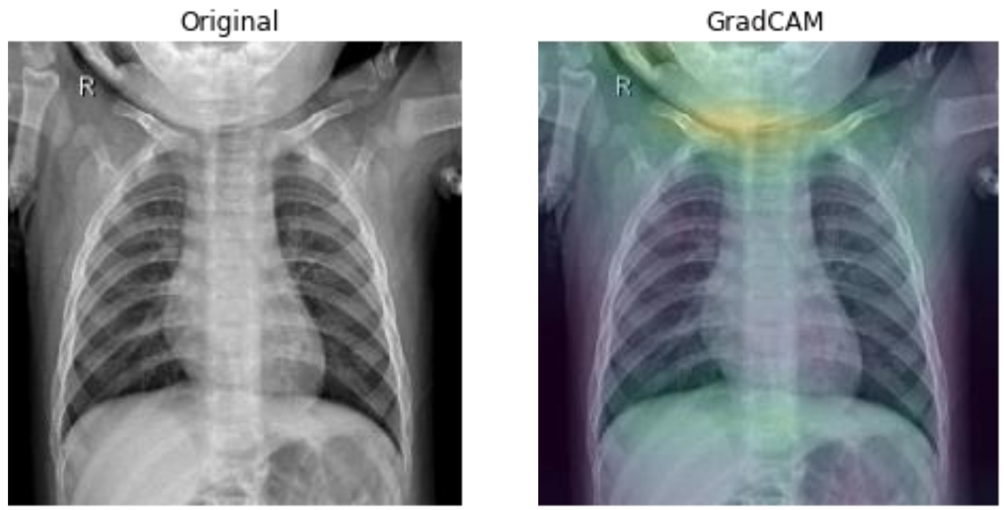
Two Covid-19 examples



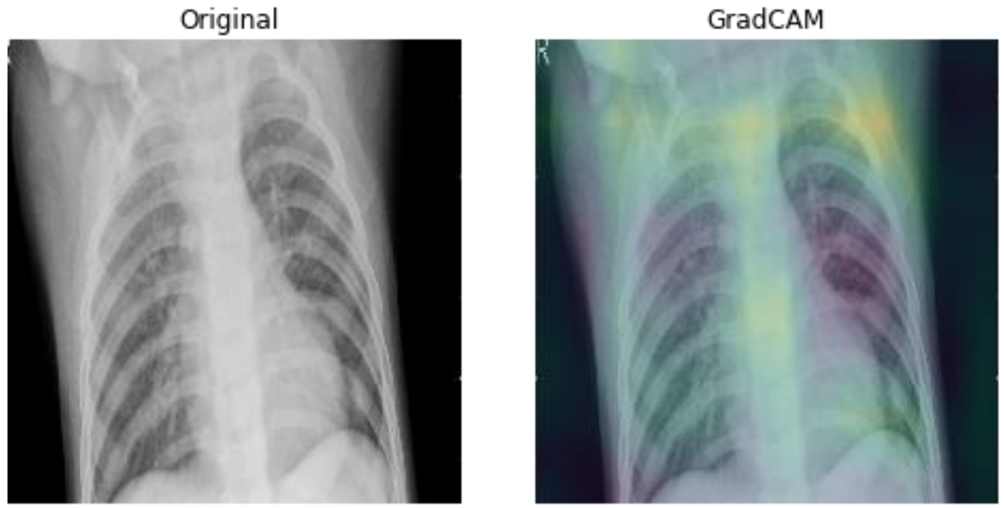
Model Interpretability: GradCAM Normal and Viral Pneumonia X-Ray Images for Reference

Activations are the strongest in brighter yellow regions

Normal



Viral Pneumonia



Model Interpretability: GradCAM Further Studies

- GradCAM images can be inspected by expert medical staff to reach a consensus on the significance of anatomical regions recognized by the model
- Further studies can be done by the data science teams and expert medical staff
    - Image segmentation can be used to identify the anatomical regions
    - New models can be created that identify the anatomical regions and the activation regions can be superimposed to immediately visualize the names of these regions

Summary

- The VGG16 Deep Learning Model can be a useful addition to the clinician/radiologist tool-kit
- This model can be trained and used to Classify Covid-19 on chest X-Ray Images
- Transfer learning allows the model to perform well even with very limited Covid-19 data
- The model can obtain the following results on unseen test data
  - Precision 0.93
  - Recall 1.00
  - F1 0.96
  - Accuracy 0.97
- Model Advantages
  - Automates Image Processing Tasks
  - Serves Predictions from the cloud using any device
  - Provides Activation regions associated with image classification for interpretation

References

Current Perspectives in Medical Image Perception, Elizabeth A.Kuprinsky (2010)

https://www.ncbi.nlm.nih.gov/pmc/articles

COVID-19 Database, S.I. S. o. M. a. I. Radiology. (2020)

https://www.sirm.org/category/senza-categoria/covid-19/

COVID-19 image data collection, Joseph Paul Cohen and Paul Morrison and Lan Dao (2020)

https://github.com/ieee8023/covid-chestxray-dataset

Chest X-Ray Images (Pneumonia), P. Mooney. (2018)

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Image Classification using Transfer Learning in Pytorch, Sunita Nayak (2019)

https://learnopencv.com/image-classification-using-transfer-learning-in-pytorch/

Various Optimization Algorithms for Training Neural Networks, Sanket Doshi (2019)

https://towardsdatascience.com/optimizers-for-training-neural-networks

Grad-CAM: Visualize class activation maps with Keras, TensorFlow, and Deep Learning, Adrian Rosebrock (2020)

https://www.pyimagesearch.com/2020/03/09/grad-cam-visualize-class-activation-maps-with-keras-tensorflow-and-deep-learning/

Four Common Types of Neural Network Layers, Martin Isaksson (2020)

https://towardsdatascience.com/four-common-types-of-neural-network-layers

Keras Callbacks Explained in Three Minutes, Andre Duong (2019)

https://www.kdnuggets.com/2019/08/keras-callbacks-explained-three-minutes

Image Data Pre-processing for Neural Networks, Nikhil B (2017)

https://becominghuman.ai/image-data-pre-processing-for-neural-networks

End

Thank you.