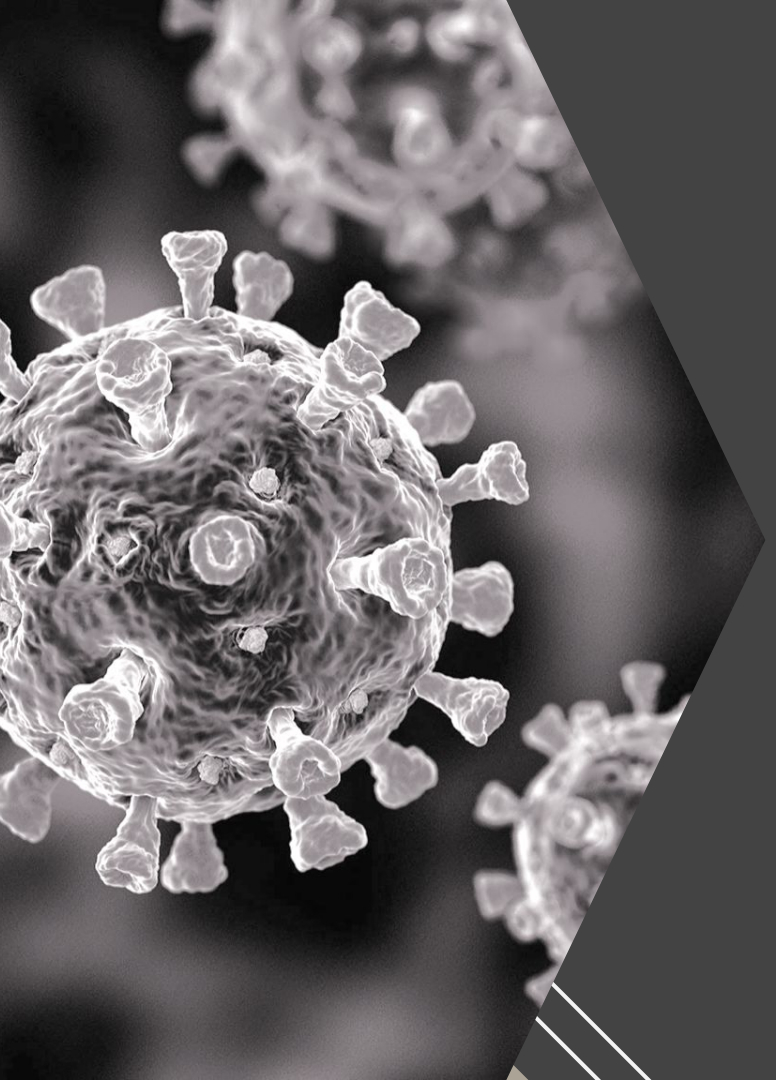


On the Improvement of COVID-19 Pneumonia Classification on Chest Radiographs through Transfer Learning

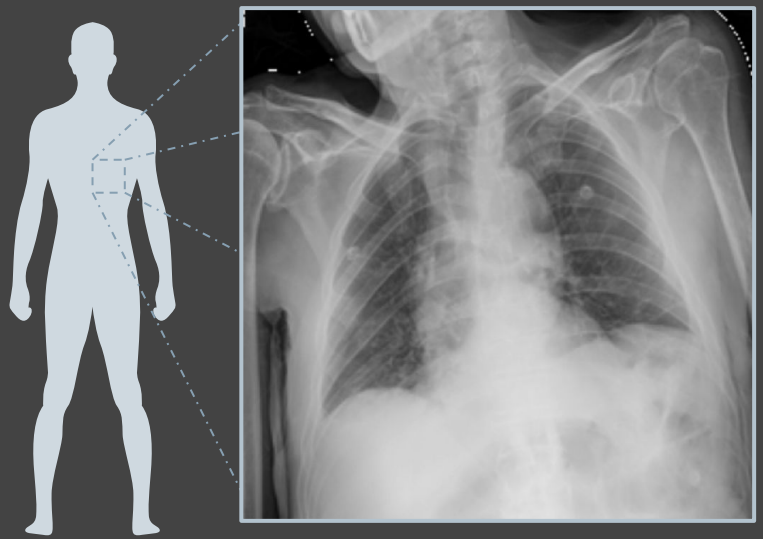
University of Connecticut - Summer Research
June - August 2021

By: Grace Liu

TABLE OF CONTENTS

INTRODUCTION

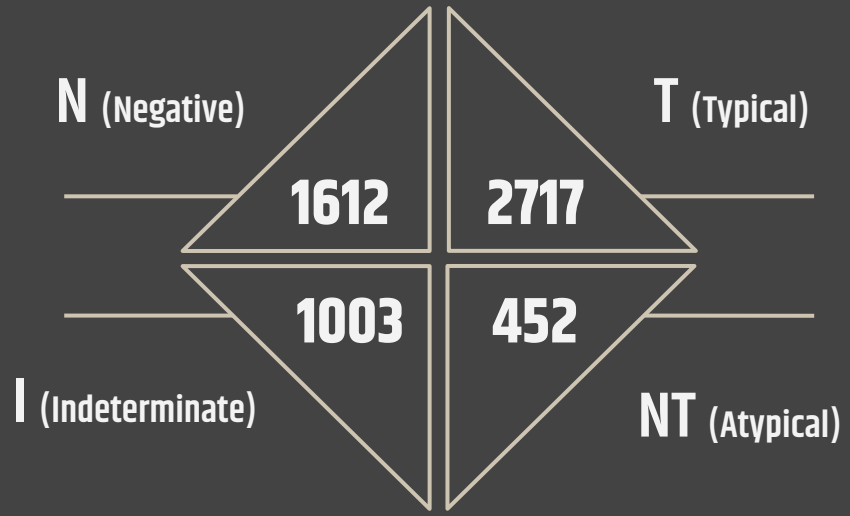
COVID-19 has caused over 4 million deaths worldwide… how can we diagnose it with machine learning?

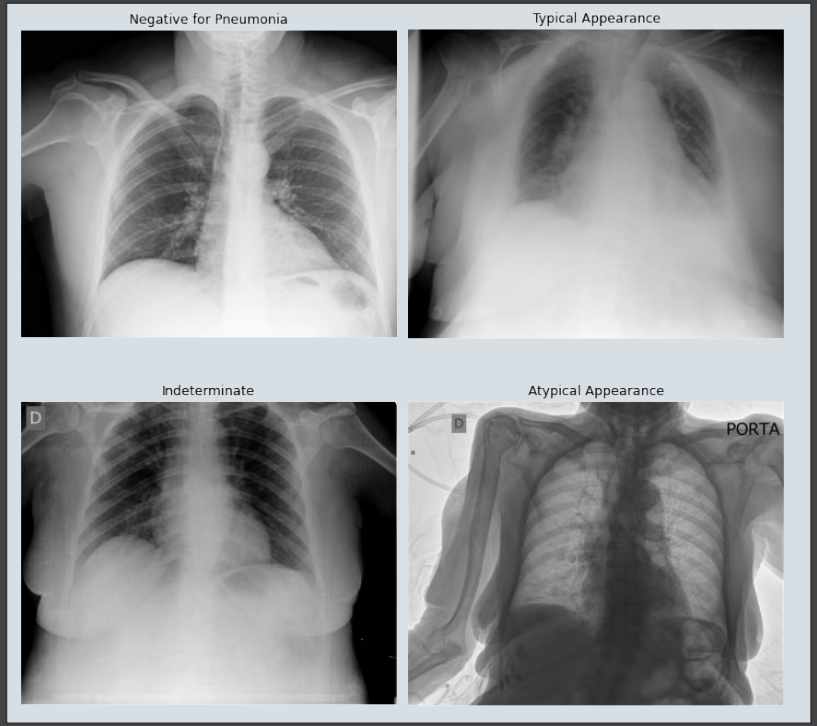


Specific Issue: Identifying viral COVID-19 pneumonia on chest radiographs

Kaggle Competition:
SIIM-FISABIO-RSNA COVID-19 Detection

DATA OVERVIEW





METHODS

- Python, Tensorflow, Keras
- Lopsided data distribution → a model built from scratch is expensive and prone to overfitting
  - Using transfer learning
  - Def: Features learned on one problem are used for another related problem

VGG16 BASIC MODEL

- Using Keras's pre-trained models and adding classifier layers
- Mismatch between medical file format + model default input sizes
  - 1 channel vs RGB
- Broadcasting
- Very low accuracy ~41% & overfitting
- Simplifying by focusing on binary classification (typical vs atypical)

DATA AUGMENTATION

'equalize_adaphist'

Contrast Limited AHE: local contrast enhancement

random flip

Flipping the image horizontally/vertically

random zoom

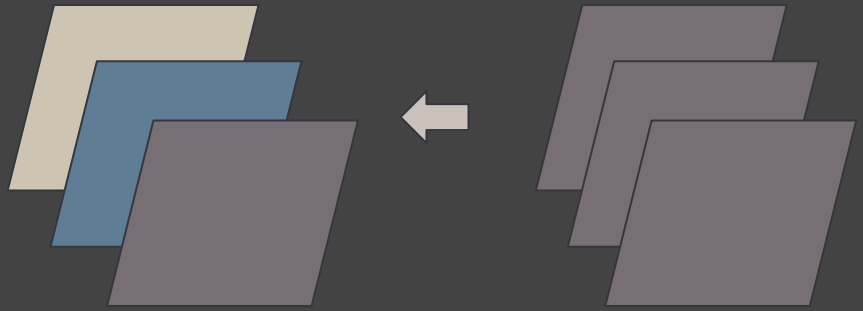
Magnifies the image by a random factor

random contrast

Adjusts contrast by a random factor

AUGMENTATIONS IN CHANNELS

- Based on previous success with random contrast, we decided to augment the color channels
- Instead of broadcasting, applying RandomContrast(0.15) to the 2 other channels



CODE SAMPLE

```
x1 = inputs
x2 = keras.layers.experimental.preprocessing.RandomContrast(0.15)(x1)
x3 = keras.layers.experimental.preprocessing.RandomContrast(0.15)(x1)
x = tf.keras.layers.Concatenate(axis=-1)([x1, x2, x3])
```

TESTING PRE-TRAINED MODELS

VGG16

Original experiment

Inception_V3

Keras Model

ResNet50

Keras Model

InceptionResNetV2

Combination of Inception + ResNet

EfficientNetB7

B0-B7 Models, 7 is best accuracy

ENSEMBLED MODEL

InceptionResNetV2

Features extracted with this model

EfficientNetB7

Combined with features extracted from this model

RESULT

Significant improvement in accuracy

CODE SAMPLE

```
def make_transfer_inceptionresnet_efficient_model(image_size, num_classes):

    base_model1 = keras.applications.EfficientNetB7(
        weights="imagenet",  # Load weights pre-trained on ImageNet.
        input_shape=(600, 600, 3),
        include_top=False,
    )  # Do not include the ImageNet classifier at the top.
    base_model1 = add_prefix(base_model1,'b1')

    base_model2 = keras.applications.InceptionResNetV2(
        weights="imagenet",  # Load weights pre-trained on ImageNet.
        input_shape=(299, 299, 3),
        include_top=False,
    )  # Do not include the ImageNet classifier at the top.
    base_model2 = add_prefix(base_model2,'b2')
```

MODEL SUMMARY

```
Model: "model_5"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_6 (InputLayer)            [(None, 600, 600, 1) 0
__________________________________________________________________________________________________
random_contrast_4 (RandomContra (None, 600, 600, 1)  0           input_6[0][0]
__________________________________________________________________________________________________
random_contrast_5 (RandomContra (None, 600, 600, 1)  0           input_6[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 600, 600, 3)  0           input_6[0][0]
                                                                 random_contrast_4[0][0]
                                                                 random_contrast_5[0][0]
__________________________________________________________________________________________________
resizing_1 (Resizing)           (None, 299, 299, 3)  0           concatenate_2[0][0]
__________________________________________________________________________________________________
tf.math.truediv_1 (TFOpLambda)  (None, 299, 299, 3)  0           resizing_1[0][0]
__________________________________________________________________________________________________
tf.math.subtract_1 (TFOpLambda) (None, 299, 299, 3)  0           tf.math.truediv_1[0][0]
__________________________________________________________________________________________________
b1efficientnetb7 (Functional)   (None, 19, 19, 2560) 64097687    concatenate_2[0][0]
__________________________________________________________________________________________________
b2inception_resnet_v2 (Function (None, 8, 8, 1536)   54336736    tf.math.subtract_1[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_2 (Glo (None, 2560)         0           b1efficientnetb7[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_3 (Glo (None, 1536)         0           b2inception_resnet_v2[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 4096)         0           global_average_pooling2d_2[0][0]
                                                                 global_average_pooling2d_3[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 4096)         0           concatenate_3[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 2)            8194        flatten_1[0][0]
==================================================================================================
Total params: 118,442,617
Trainable params: 8,194
Non-trainable params: 118,434,423
```

RESULTS

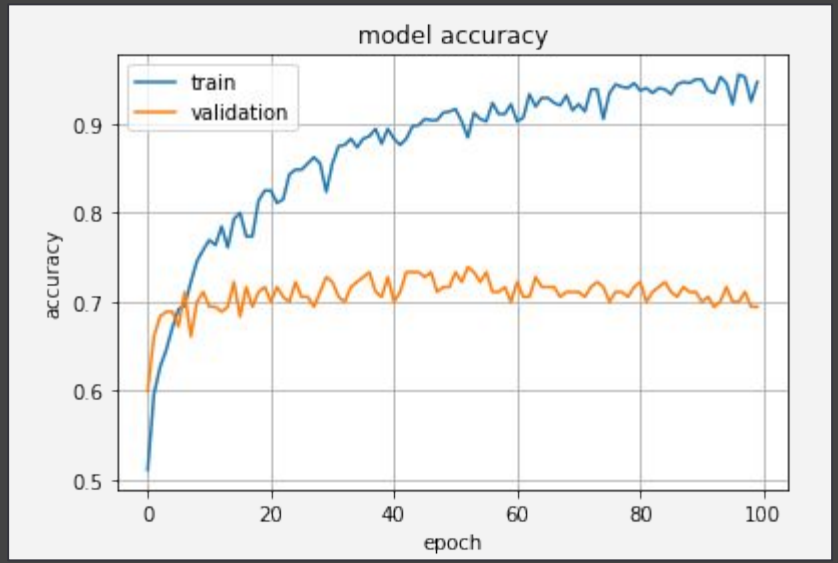
Discussing final accuracy obtained and conclusions

03

TABLE SUMMARY

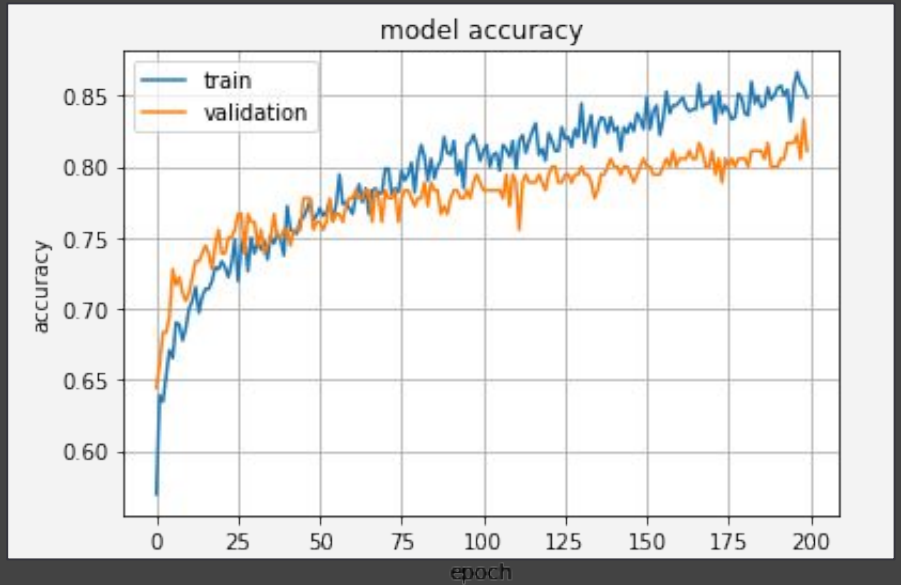
Table 1: A comparison of results from all different models		
Model Name and Description	Classes Compared	Accuracy
VGG16 model	N, T, I, NT	41%
VGG16 model with Contrast Limited AHE (CLAHE)	N, T, I, NT	39%
VGG16 model	T, NT	69%
VGG16 model with variety of data augmentation functions	T, NT	66%
VGG16 model with channel augmentation	T, NT	73%
Inception_V3 model with channel augmentation	T, NT	68%
ResNet50 model with channel augmentation	T, NT	67%
InceptionResNetV2 with channel augmentation	T, NT	76%
EfficientNetB7 model with channel augmentation	T, NT	76%
InceptionResNetV2 + EfficientNetB7 model with channel augmentation	T, NT	81%

ACCURACY CURVE COMPARISON



Earliest Attempt (VGG16)

Final Attempt (Ensembled Model)



METHOD SUCCESSES AND FAILURES

1. Augmenting the color channels ~7% increase in accuracy from basic data augmentation
2. Combining features extracted from 2 models ~5% increase in accuracy

1. Basic data augmentation methods - Filters, basic manipulation, etc.
2. Broadcasting
3. Using single model

FUTURE IMPROVEMENTS

- Experimenting further with the ensemble model -- would combining features from more than 2 models be more successful?
- I used equal sample sizes from each class at the beginning
  - Making full use of all the images and adding more weights
- Finally, testing out additional augmentation strategies

REFERENCES

Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

Roberts, M., Driggs, D., Thorpe, M. et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. Nat Mach Intell 3, 199–217 (2021). https://doi.org/10.1038/s42256-021-00307-0

Ismael AM, Şengür A. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl. 2021;164:114054. https://doi.org/10.1016/j.eswa.2020.114054

Kaggle Competition:

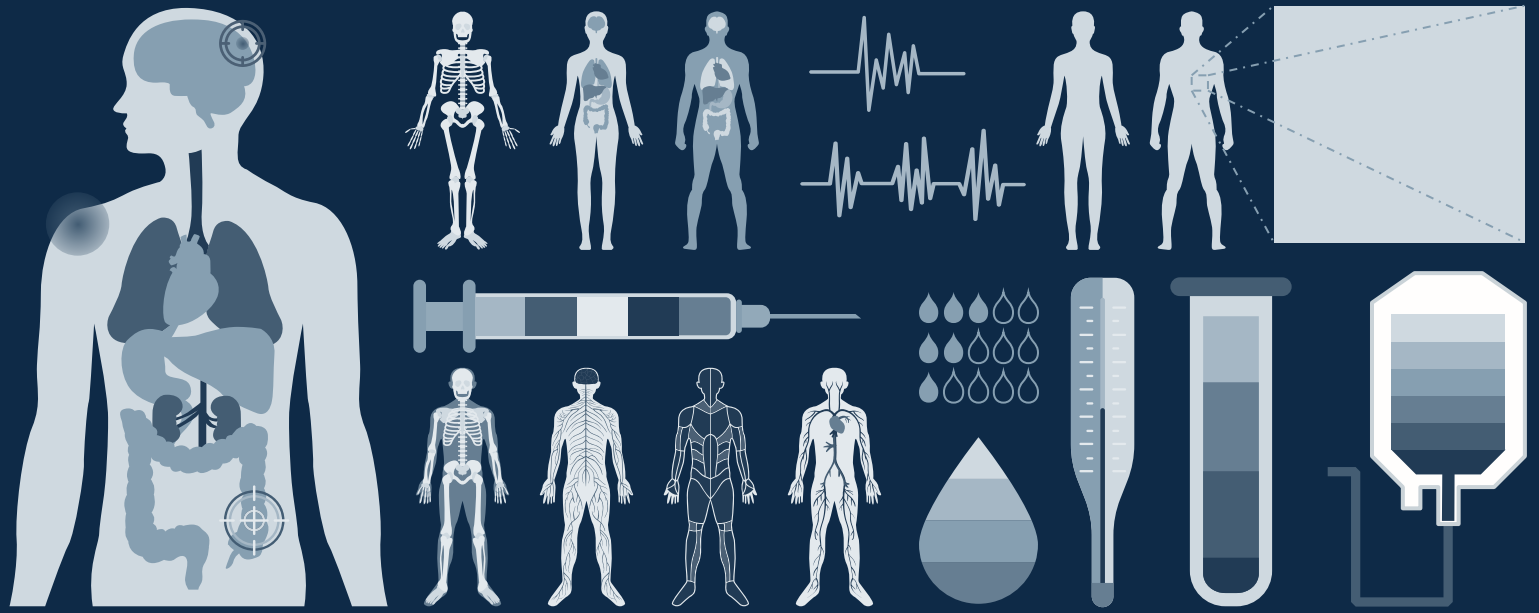
ACKNOWLEDGEMENTS

- Professor Jinbo Bi for advising this project
- Professor Shengli Zhou for organizing this summer research opportunity
- All the other professors and graduate students who presented and provided suggestions
- Jian Liu for also advising this project

THANKS!

Any questions?

Medical Infographics



Educational Icons

Medical Icons

SEO & Marketing Icons