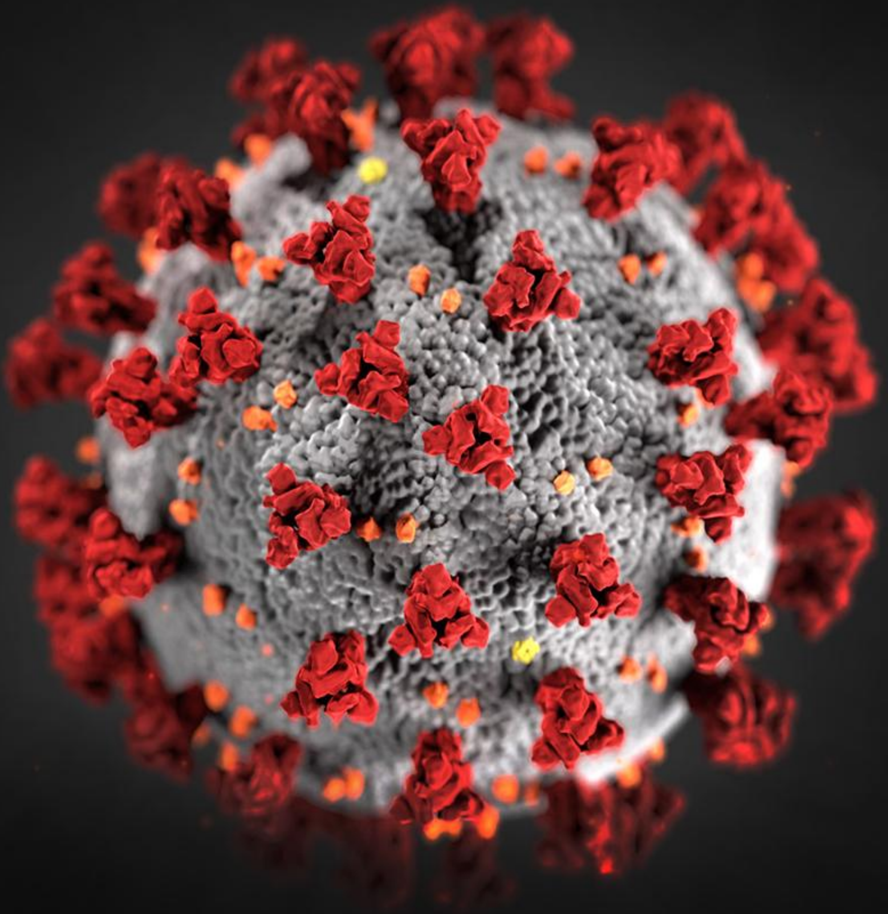
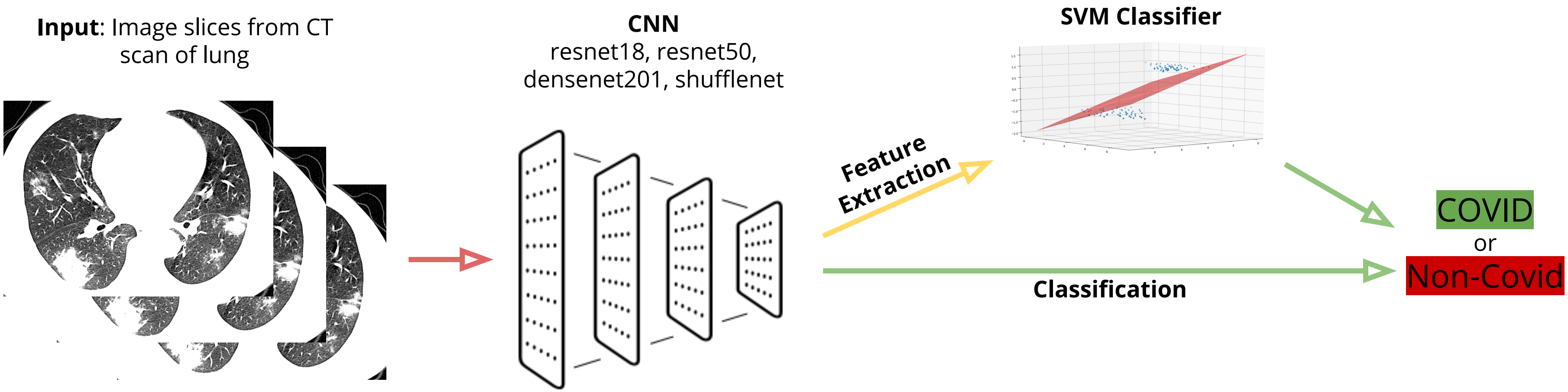
COVID19 Detection

By Carson Whitt, Dylan Curry,
Jakob Evans, Tommy Le



The Goal

- Research current literature on Covid-19 detection methods
- Create multiple models for detecting Covid-19 in CT scan slices



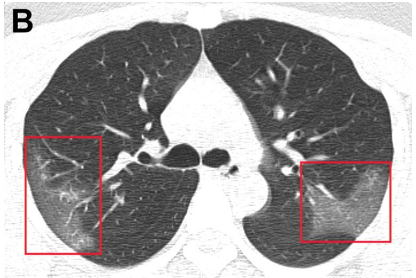
The Datasets

UCSD Covid & Non-Covid dataset

- 384 Patients (213 Covid, 171 Non-covid) - 746 images total
- Patient images taken from various research papers
- Used to train multiple networks and do preliminary testing

Coronacase & Radiopaedia Dataset

- 20 native CT scans from 20 different patients (all Covid patients)
- Three slices with thresholding taken from each patient, focusing on the lung area



Nature Paper Overview [1]

- “Study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks”
- Used the UCSD 384 patient dataset
- Tested 16 pre-trained CNNs with and without data augmentation
- Performance was very good for a majority of CNNs
- But…there was a lot of unnecessary bias

SCIENTIFIC
REPORTS
natureresearch

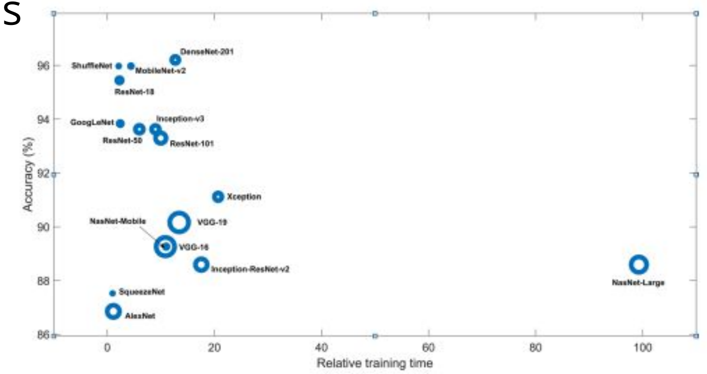
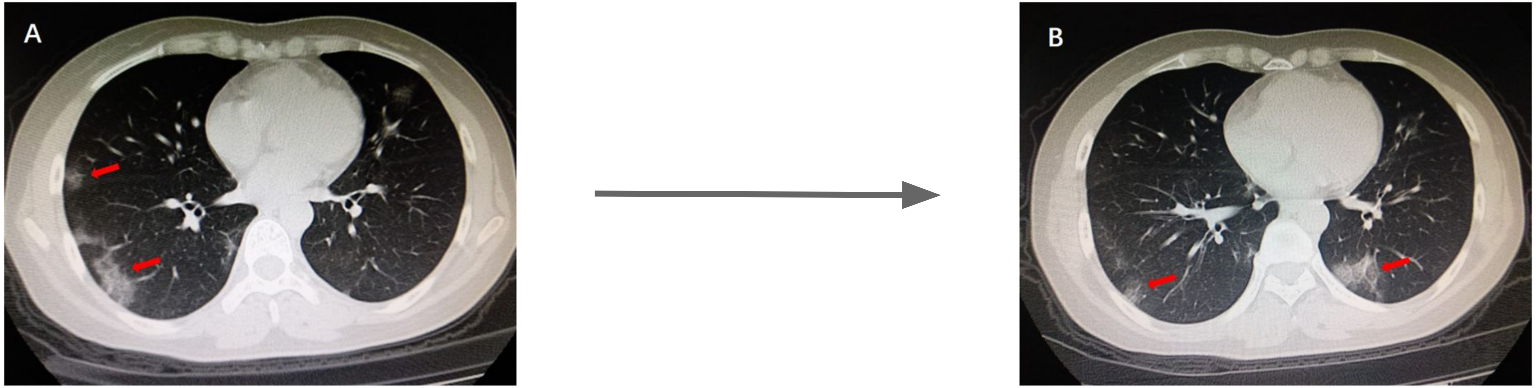


Figure 2. Plots of accuracy vs. relative training time (ratio of training time of a network to the training time of the SqueezeNet) of 16 pretrained CNNs using COVID-19 CT database, where the circle size indicates the magnitude of memory in MB.

[1] Tuan D. Pham. *A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks*

Addressing Bias

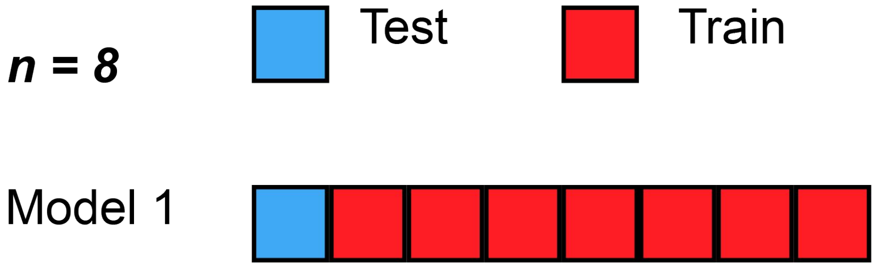
- Bias occurs from many sources
- In the Nature research paper, same patient bias inflated results
    - They performed a random data split (80% training, 20% testing)
    - Same patient images in both training and testing
    - CNN finds subjective features of same patient images (rather than actual Covid19 features)



- To address these biases we used a data splitting method called Leave One Patient Out Cross Validation

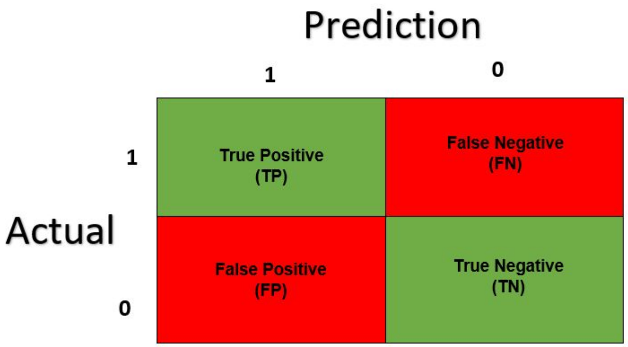
Leave-One-Patient-Out Cross Validation

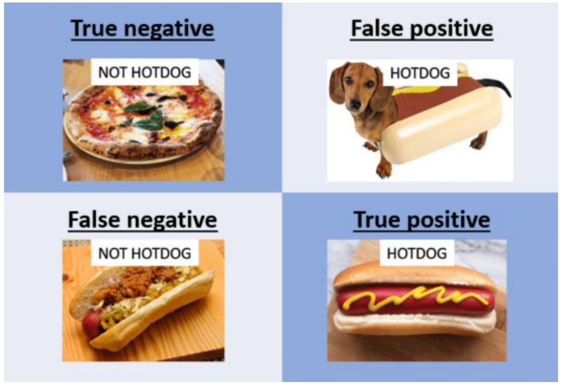
- Process:
  - Divide the dataset into subsets for each patient
  - The number of times we will train and test our network is however many patients there are (takes much longer to run...in our case 384x longer)
  - For each iteration, one subset will be used for testing the network and the rest of the subsets will be used for training
- Reason:
  - Increases the amount of training data. Each subset will be part of the training set k-1 times
  - Gets rid of bias that occurs when a patient’s images are in the test and training sets at the same time



Preliminary Testing Using Leave One Out

- We did tests on four CNNs (resnet18, resnet50, densenet201, shufflenet)
  - Using the UCSD dataset
  - Performed transfer learning using leave one out
  - Performed feature extraction using leave one out and SVM classifier
- For each test we computed
  - Confusion matrix (Of all testing results from 384 models)
  - Overall Accuracy (Accuracy of combined confusion matrix from 384 tests)
  - Overall F_1Score (F_1Score of combined confusion matrix from 384 tests)





Transfer Learning(CNN)

Using the CNN layers:

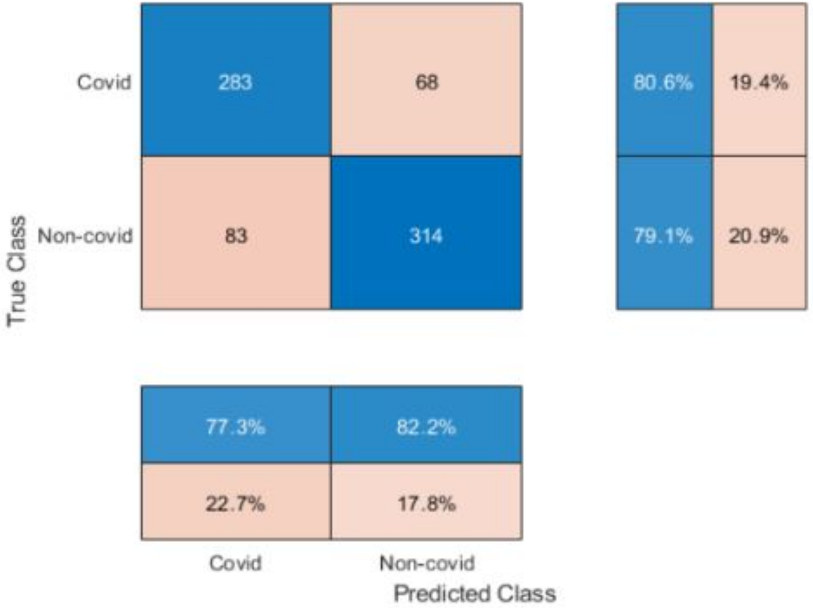
- Trained the CNNs on UCSD dataset
- Classified test images
- Computed accuracy, standardDev, fscore and the confusion matrix

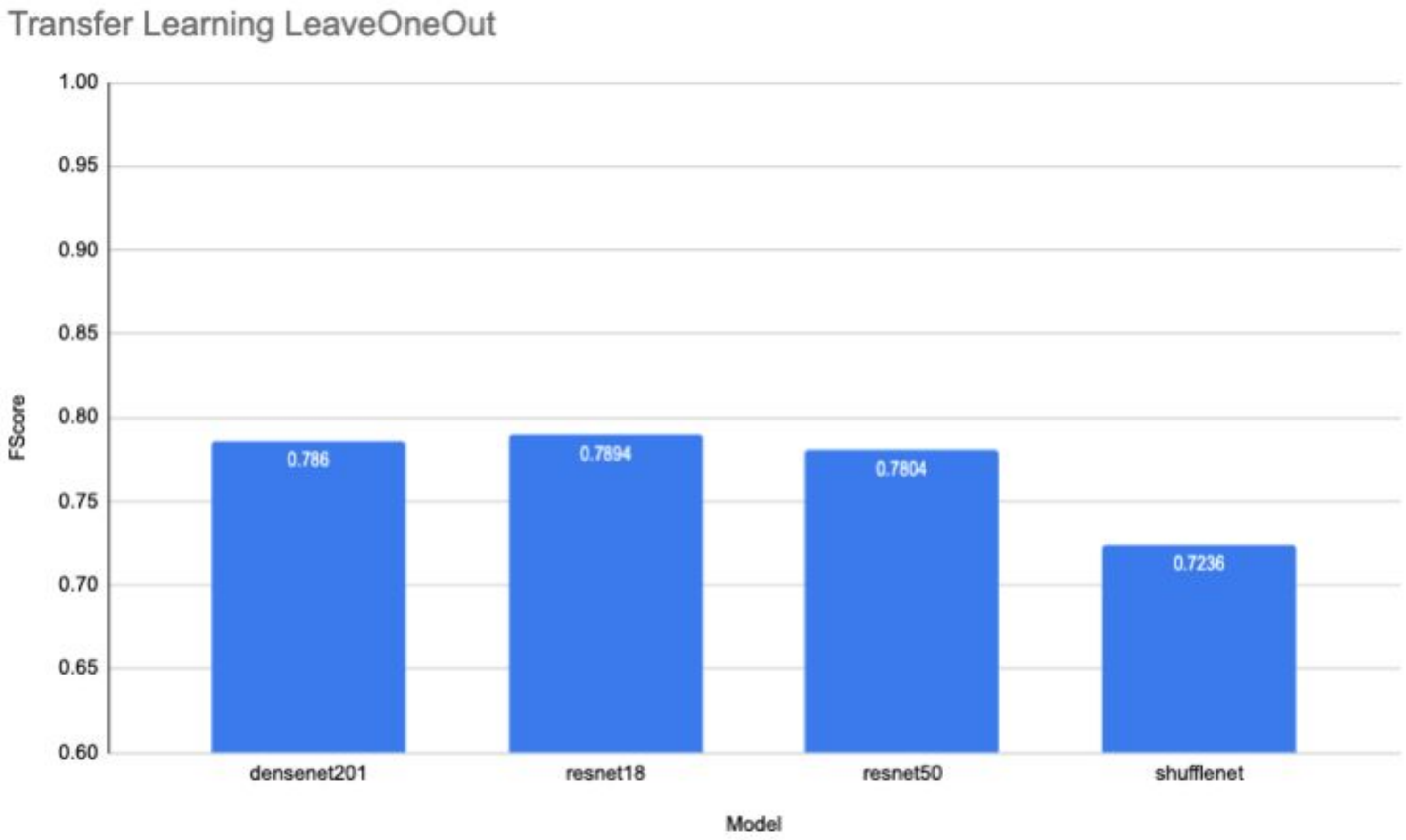
We got the best results with resnet18

- Frozen layer 1
- Solver name ‘sgdm’
- Took roughly 12 hrs to finish

resnet18

Combined Mean ACC	Combined Mean F1	sDev ACC
0.7982	*0.7894*	*0.3436*




Transfer Learning LeaveOneOut
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
FScore
0.786
0.7894
0.7804
0.7236
densenet201
resnet18
resnet50
shufflenet
Model

Feature Extraction(CNN) Using SVM Classifier

Using the extracted CNN features:

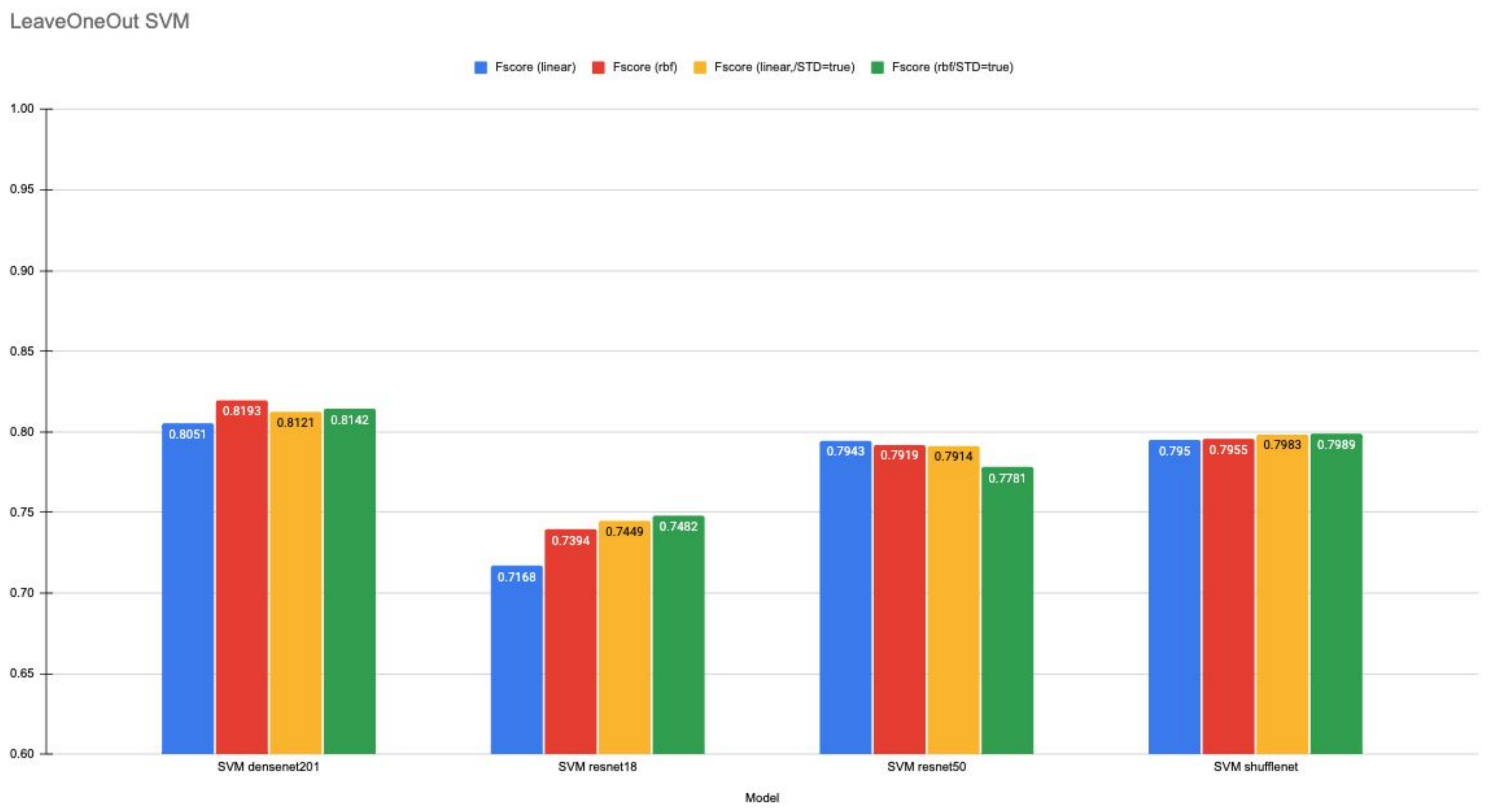
- Classified the test set of images using an SVM classifier
- Computed accuracy, standardDev, fscore and the confusion matrix

We got the best results with densenet201

- Using the 'rbf' Kernel
- Features extracted from layer 'conv5_block32_concat'
- Took roughly 4-5 hrs to finish

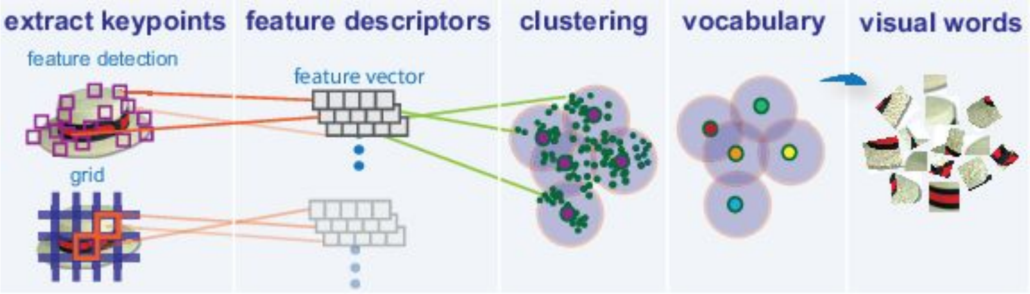
densenet201

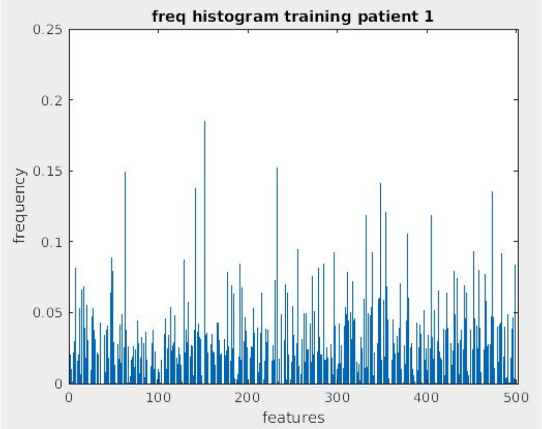
Combined Mean ACC	Combined Mean F1	sDev ACC
0.8302	*0.8193*	*0.3366*



LeaveOneOut SVM
Fscore (linear)
Fscore (rbf)
Fscore (linear,/STD=true)
Fscore (rbf/STD=true)
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.8051
0.8193
0.8121
0.8142
0.7168
0.7394
0.7449
0.7482
0.7943
0.7919
0.7914
0.7781
0.795
0.7955
0.7983
0.7989
SVM densenet201
SVM resnet18
SVM resnet50
SVM shufflenet
Model

Create a Bag of Features (visual vocabulary) with training data





Encoding occurs for EACH training image

Encode each image for the image classifier:

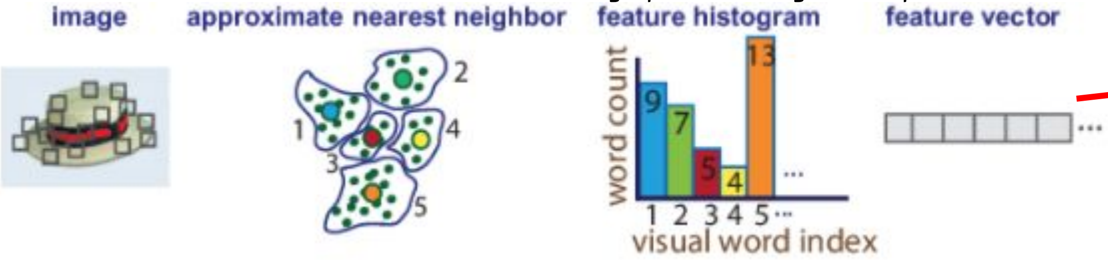
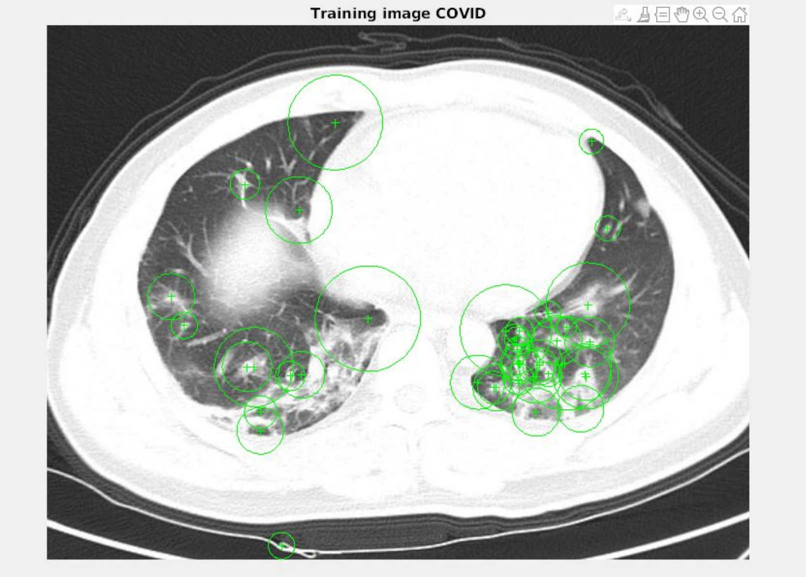


Image classifier

SIFT and SURF feature extraction compared

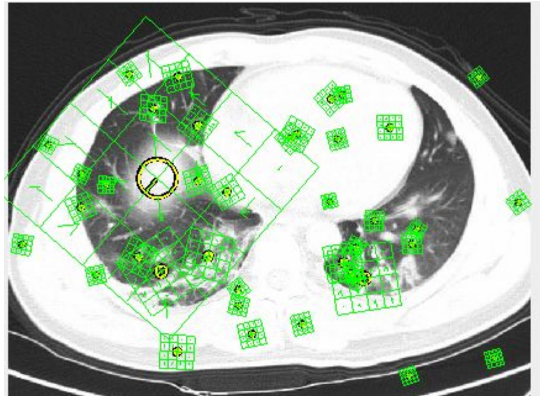
SURF:



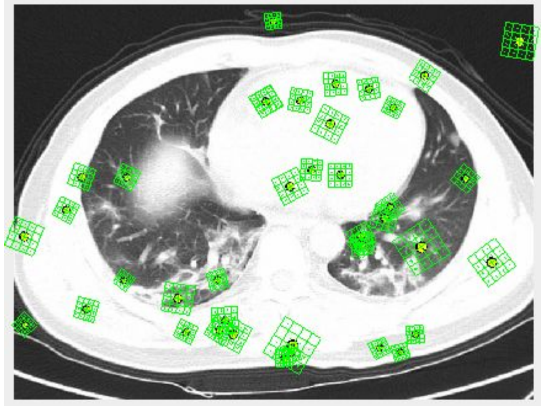
Accuracy = .7537 fscore = .7135

SIFT :

Edge thresh 8



Edge thresh 4



Preliminary Testing Results

- Fairly good results for best performing networks
  - Best networks had ~ 80% accuracy and ~ .8 F_1 Scores
- No where near the same results as the Nature Paper
  - Nature paper had ~ 90% accuracy on average and ~.9 F_1 Score for the same four networks, with and without data augmentation
  - Shows that even with data augmentation, the bias from same patient imaging still has a profound effect

CNN model	Accuracy (%)	Sensitivity (%)	Specificity (%)	F_1 score	AUC
ShuffleNet	86.13 ± 10.16	83.54 ± 19.89	89.05 ± 5.77	0.86 ± 0.12	0.93 ± 0.06
ResNet-18	90.16 ± 2.36	89.45 ± 7.31	90.95 ± 9.29	0.91 ± 0.02	0.96 ± 0.05
ResNet-50	92.62 ± 4.19	91.14 ± 3.35	94.29 ± 5.15	0.93 ± 0.04	0.98 ± 0.01
DenseNet-201	91.72 ± 6.52	88.61 ± 8.86	95.24 ± 4.36	0.92 ± 0.07	0.97 ± 0.03

Figure[2] : Classification results from different CNNs with data augmentation in the Nature Paper

[2] Tuan D. Pham. *A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks*

Final Testing

- We did final tests on four CNNs (resnet18, resnet50, densenet201, shufflenet)
  - Using the UCSD dataset to train
  - Testing on Coronacase & Radiopaedia dataset
  - Tested with transfer learning
  - Tested with feature extraction and SVM classifier
  - Gathered results with and without data augmentation on the training set
- For each test we computed
  - Confusion matrix
  - Overall Accuracy
  - Overall F_1 Score
  - *For data augmentation the mean of 5 runs and calcualted standard deviation

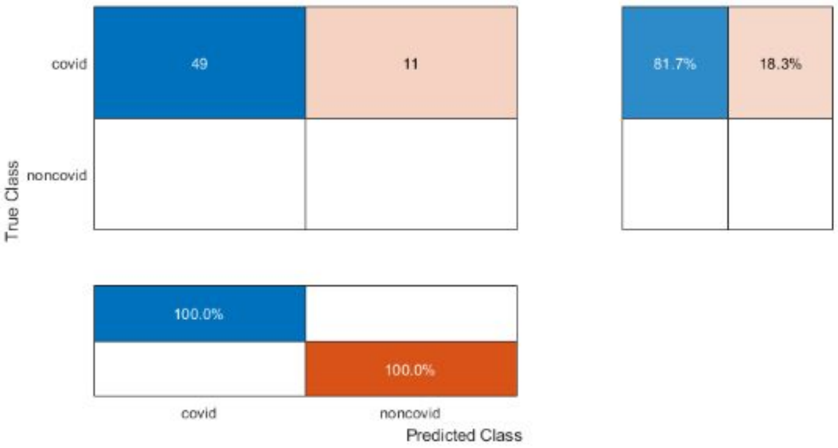
Transfer Learning Results

Final data split

Transfer Learning (resnet18)

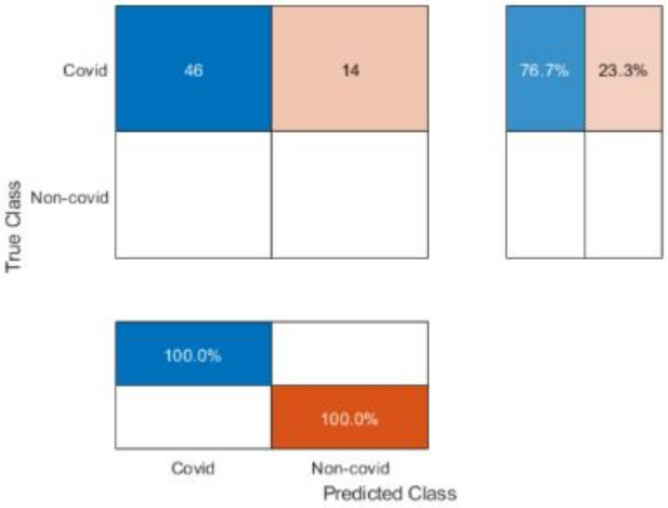
resnet18	
Combined Mean ACC	Combined Mean F1
0.8167	*0.8991*

17/20 Patients classified correctly



resnet18 with data augmentation	
Combined Mean ACC	Combined Mean F1
0.77 +/- 0.2994	*0.8367 +/- 0.2473*

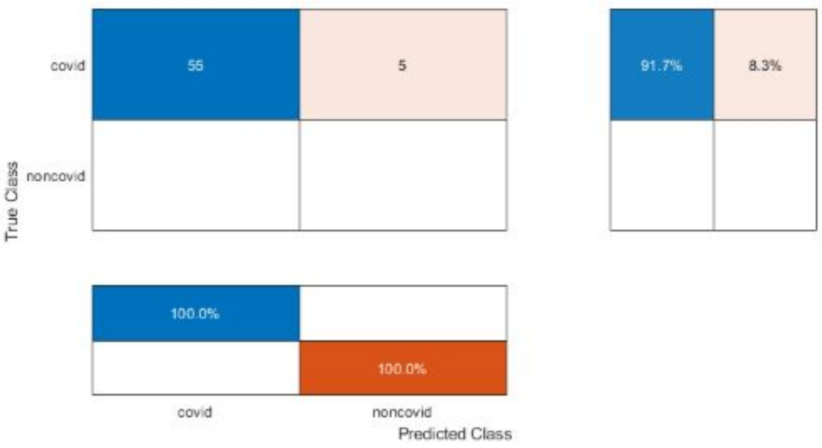
16/20 Patients classified correctly



Transfer Learning (resnet50)

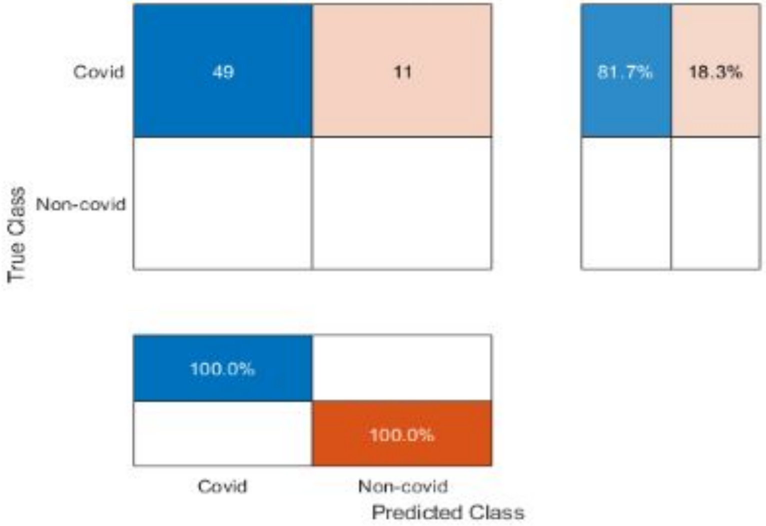
resnet50	
Combined Mean ACC	Combined Mean F1
0.9167	*0.9565*

19/20 Patients classified correctly



resnet50 with data augmentation	
Combined Mean ACC	Combined Mean F1
0.8133 +/- 0.085	*0.8941 +/- 0.05*

17/20 Patients classified correctly



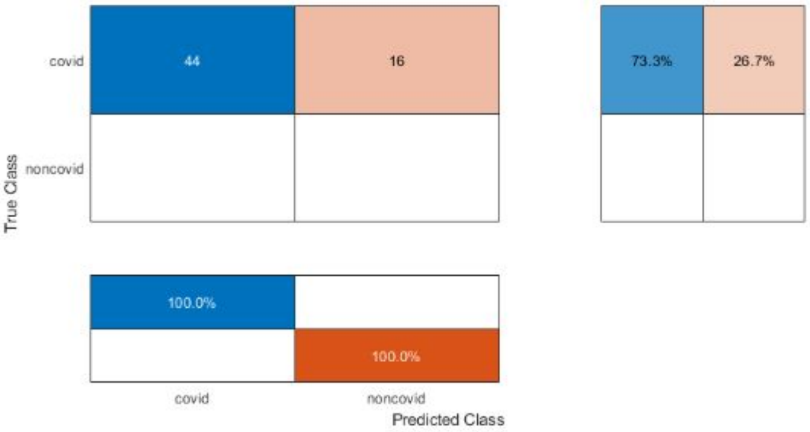
Transfer Learning (shufflenet)

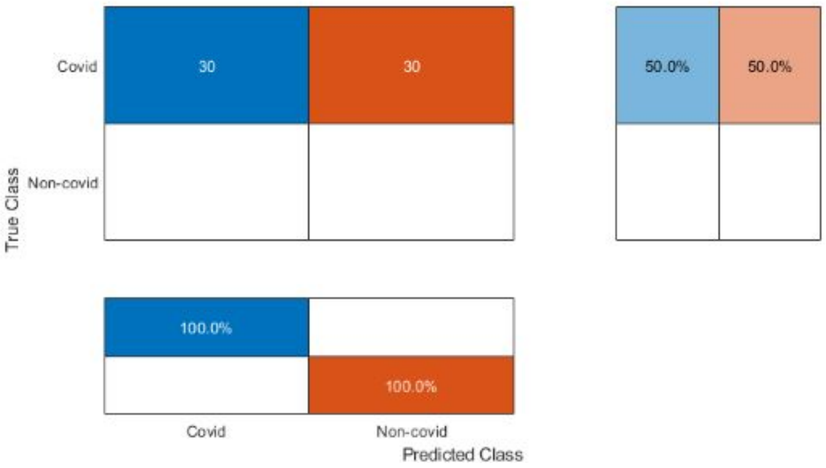
shufflenet	
Combined Mean ACC	Combined Mean F1
0.7333	0.8462

16/20 Patients classified correctly

shufflenet with data augmentation	
Combined Mean ACC	Combined Mean F1
0.4967 +/- 0.1227	0.6565 +/- 0.1051

10/20 Patients classified correctly





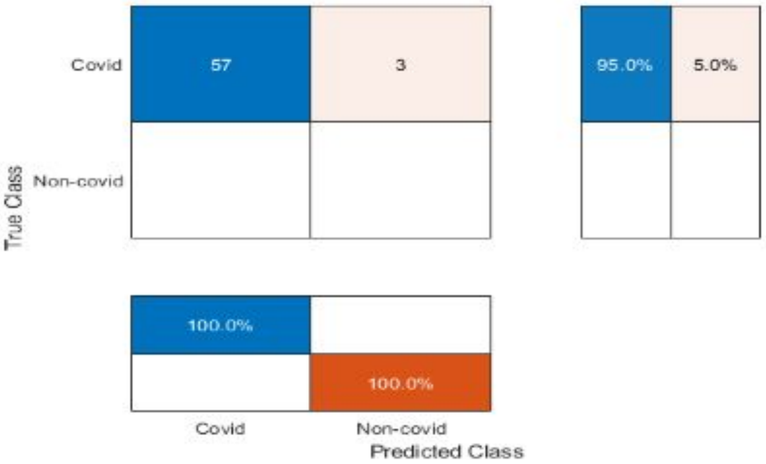
Transfer Learning (densenet201)

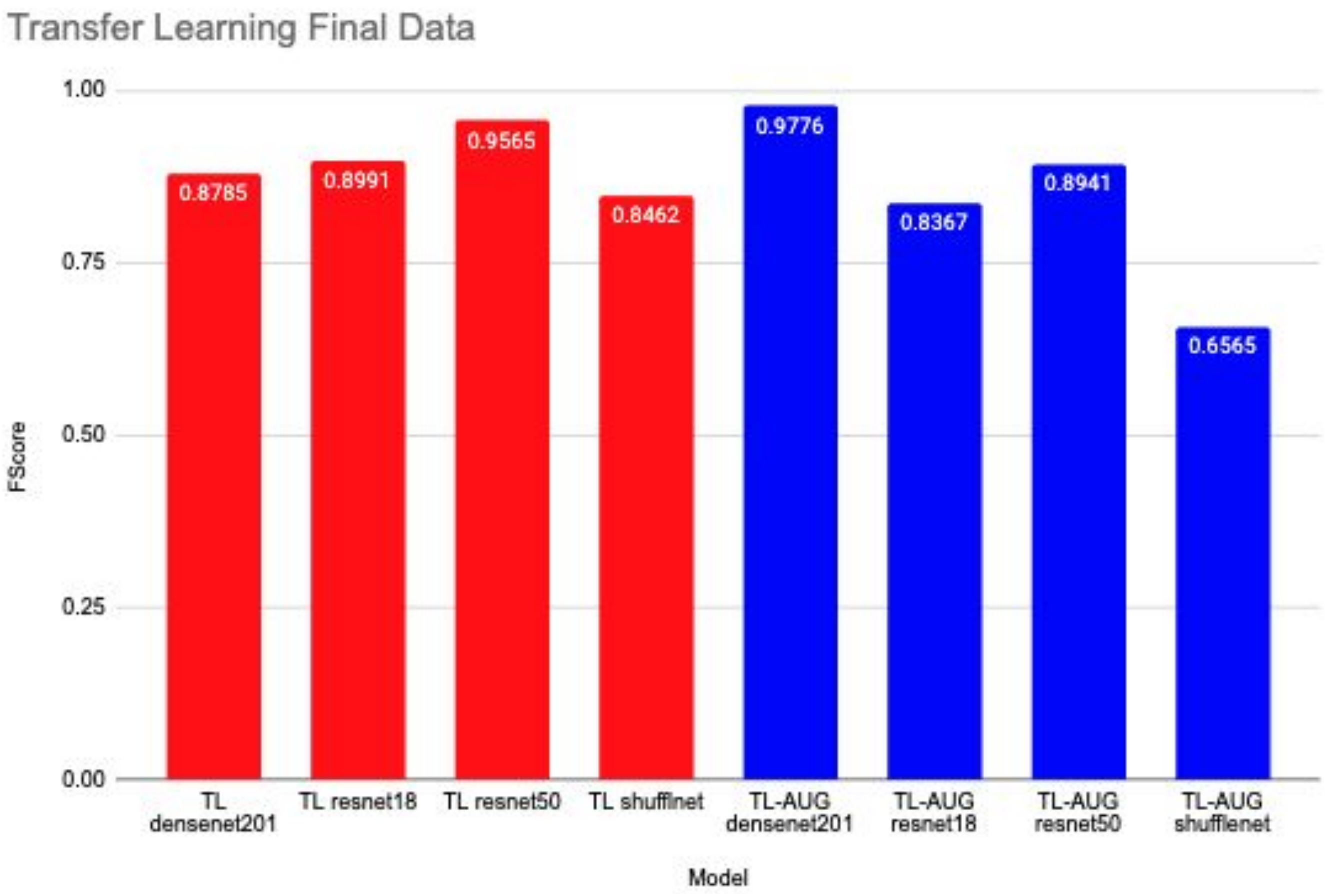
densenet201	
Combined Mean ACC	Combined Mean F1
0.7833	0.8785

16/20 Patients classified correctly

densenet201 with data augmentation	
Combined Mean ACC	Combined Mean F1
0.9567 +/- 0.033	*0.9776 +/- 0.018*

19/20 Patients classified correctly



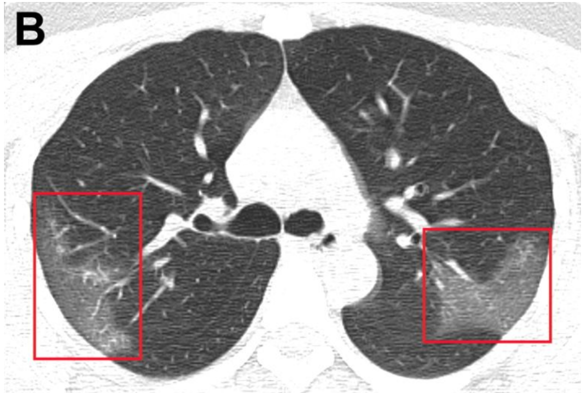


Transfer Learning Final Data
1.00
0.75
0.50
0.25
0.00
FScore
0.8785
0.8991
0.9565
0.8462
0.9776
0.8367
0.8941
0.6565
TL densenet201
TL resnet18
TL resnet50
TL shufflnet
TL-AUG densenet201
TL-AUG resnet18
TL-AUG resnet50
TL-AUG shufflenet
Model

What are the CNNs looking at?

Class Activation Mapping (CAM):

- Heat-Map of what areas of an image are being used more/less for a classification decision
    - Summary of how it works:
        1. Take feature maps(matrices) from the final convolutional-layer of the CNN
        2. Multiply the weights in the following fully connected layer by the feature maps
        3. This gives you a grid of numbers that represent the heat-map
    - To get the classification score you would instead multiply the respective weight with the average of each feature map.

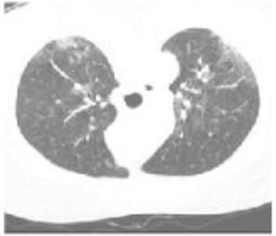


The above boxes are areas of viral pneumonia and ideally would be what we want to be used as classifying features

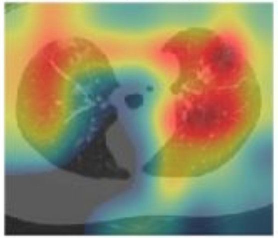
Heatmaps

resnet18 - without data augmentation

Real: covid
Pred: covid 98.9%

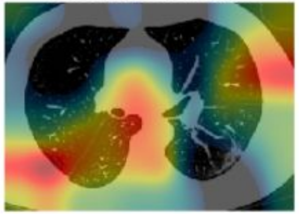


Real: covid
Pred: covid 98.9%



Real: covid
Pred: noncovid 67.4%

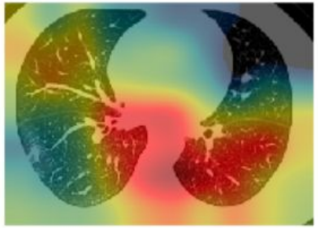
Real: covid
Pred: noncovid 67.4%



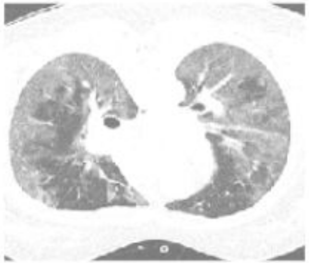
resnet18 - with data augmentation

Real: covid
Pred: covid 80.1%

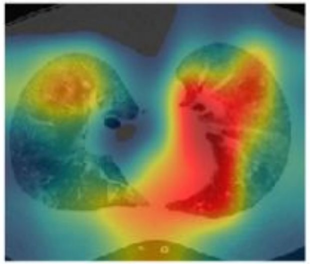
Real: covid
Pred: covid 80.1%



Real: covid
Pred: noncovid 90.7%



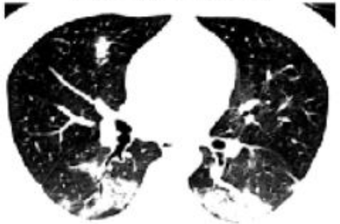
Real: covid
Pred: noncovid 90.7%



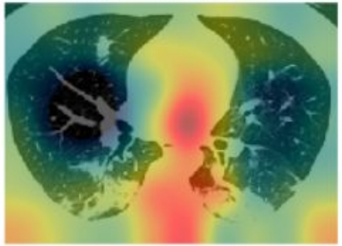
Heatmaps

resnet50 - without data augmentation

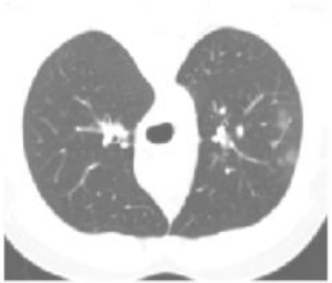
Real: covid
Pred: covid 100%



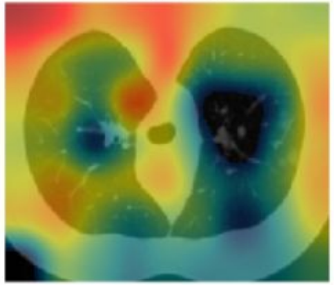
Real: covid
Pred: covid 100%



Real: covid
Pred: noncovid 70.7%

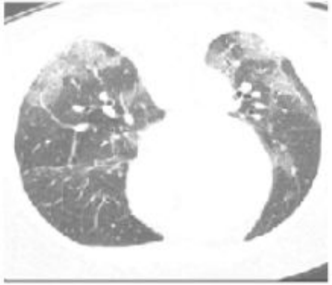


Real: covid
Pred: noncovid 70.7%

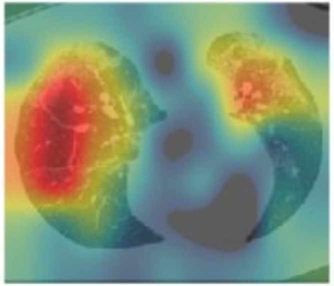


resnet50 - with data augmentation

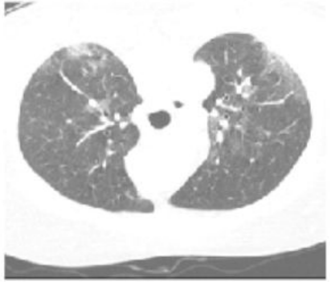
Real: covid
Pred: covid 87.3%



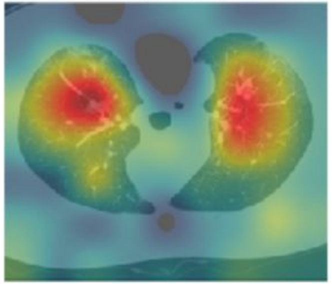
Real: covid
Pred: covid 87.3%



Real: covid
Pred: noncovid 99.5%

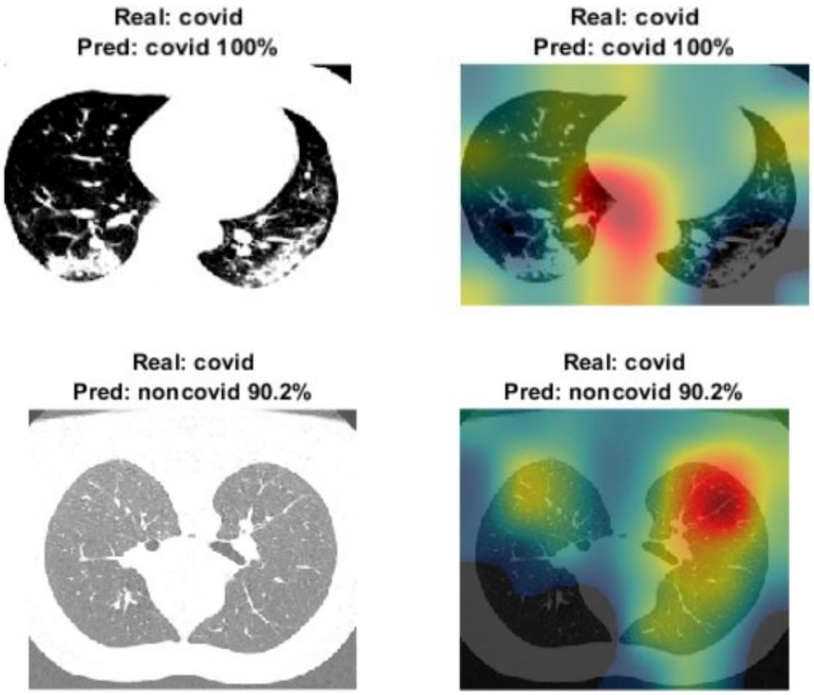


Real: covid
Pred: noncovid 99.5%

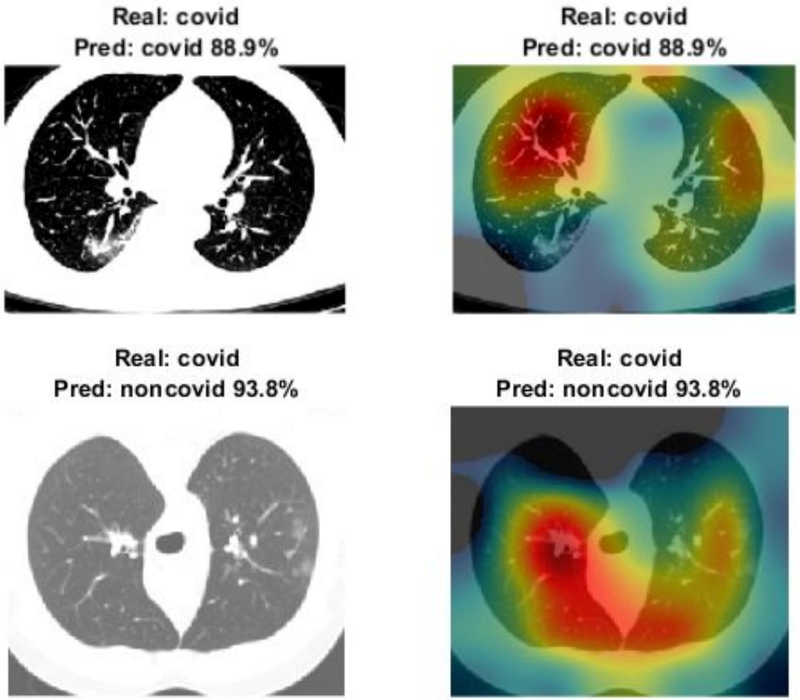


Heatmaps

densenet201 - without data augmentation

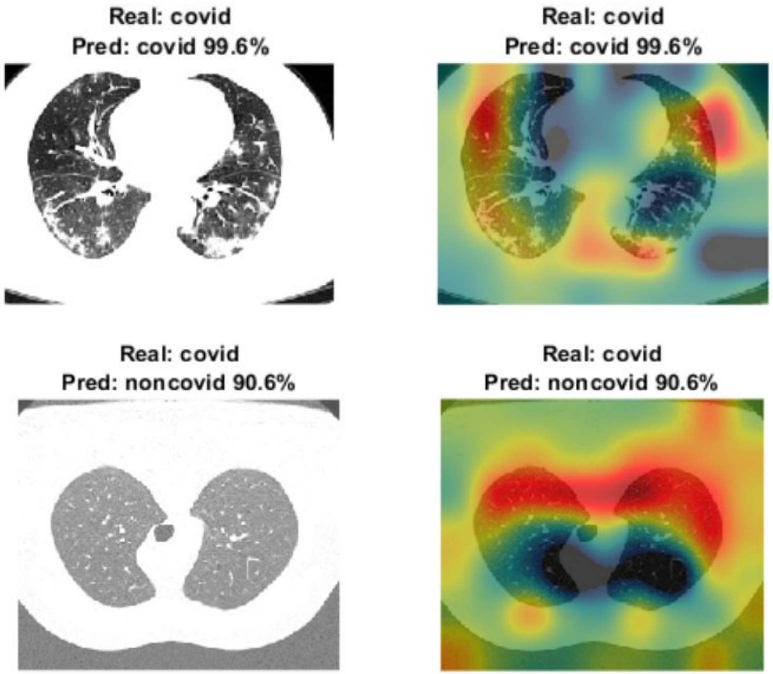


densenet201- with data augmentation

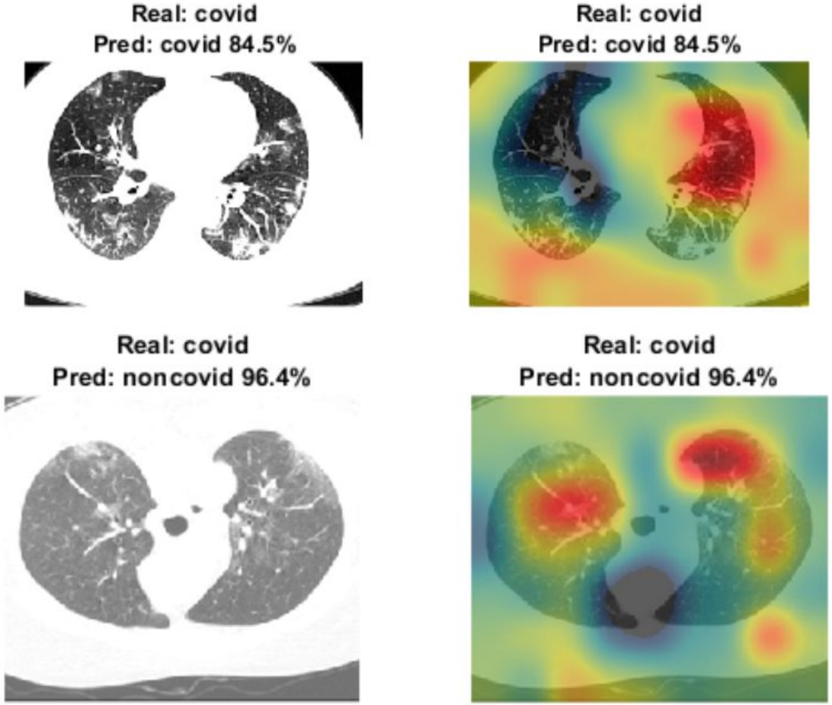


Heatmaps

shufflenet - without data augmentation



Shufflenet - with data augmentation



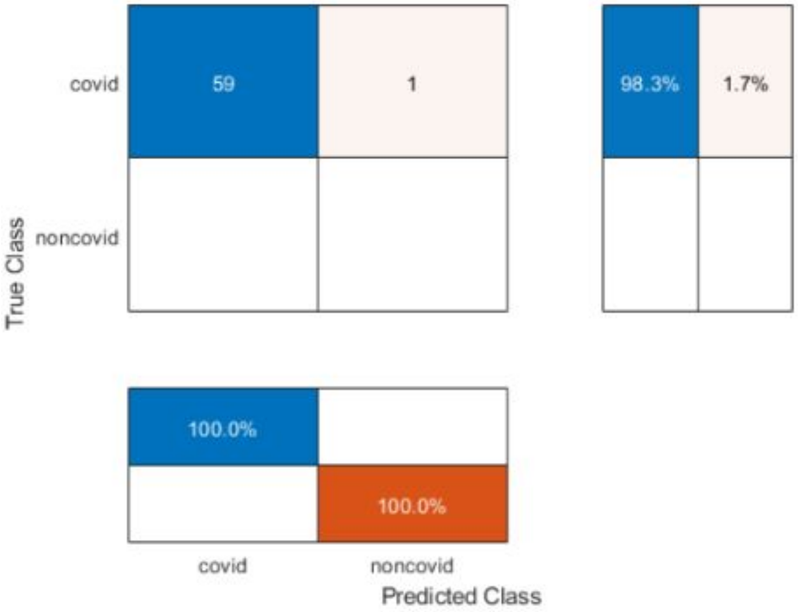
Feature Extraction & SVM Results

Final data split

Feature Extraction(CNN) Using SVM Classifier

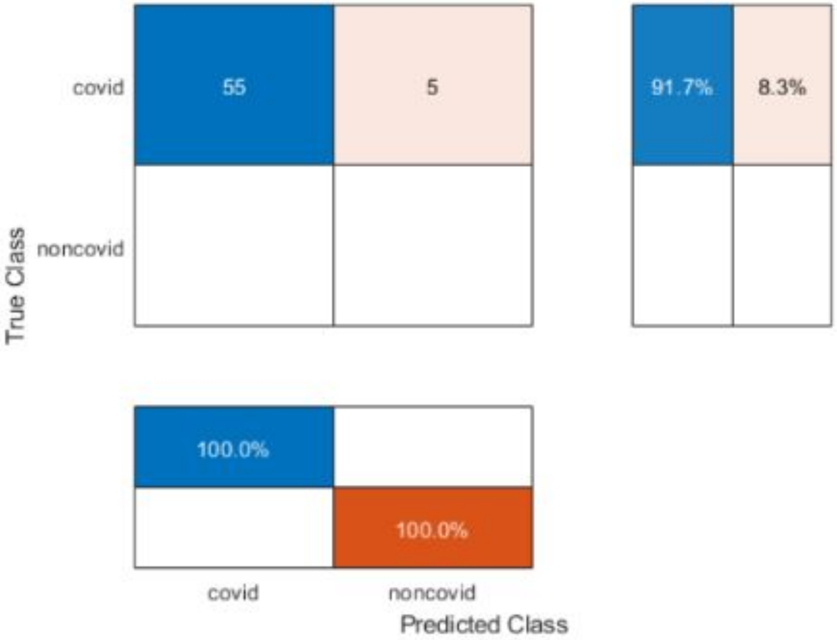
resnet18	
Combined Mean ACC	Combined Mean F1
0.9833	*0.9916*

19/20 Patients classified correctly



resnet50	
Combined Mean ACC	Combined Mean F1
0.9167	*0.9565*

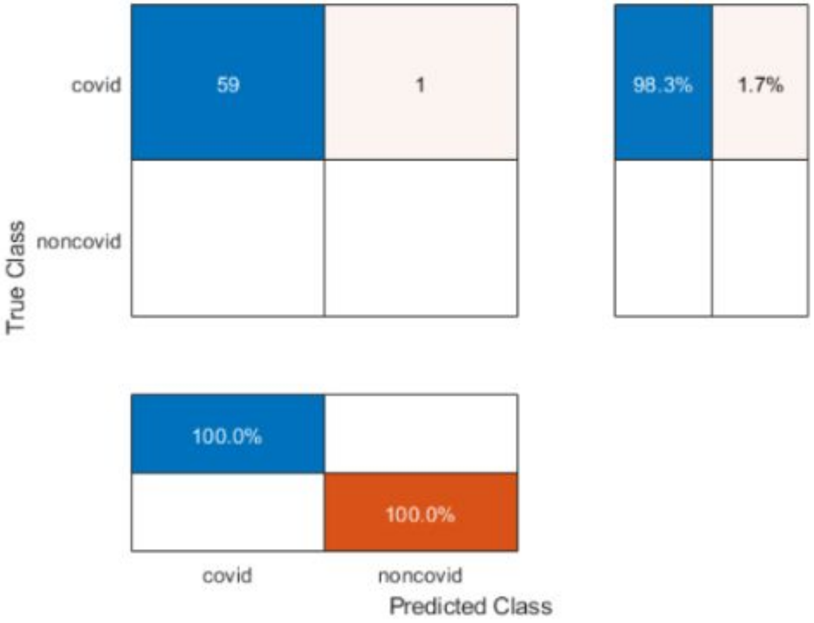
18/20 Patients classified correctly



Feature Extraction(CNN) Using SVM Classifier

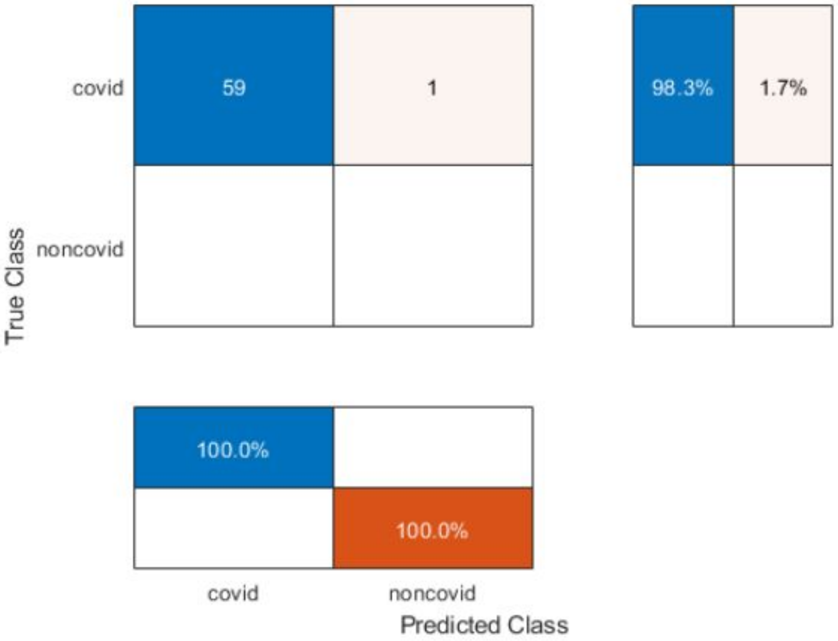
densenet201	
Combined Mean ACC	Combined Mean F1
0.9833	*0.9916*

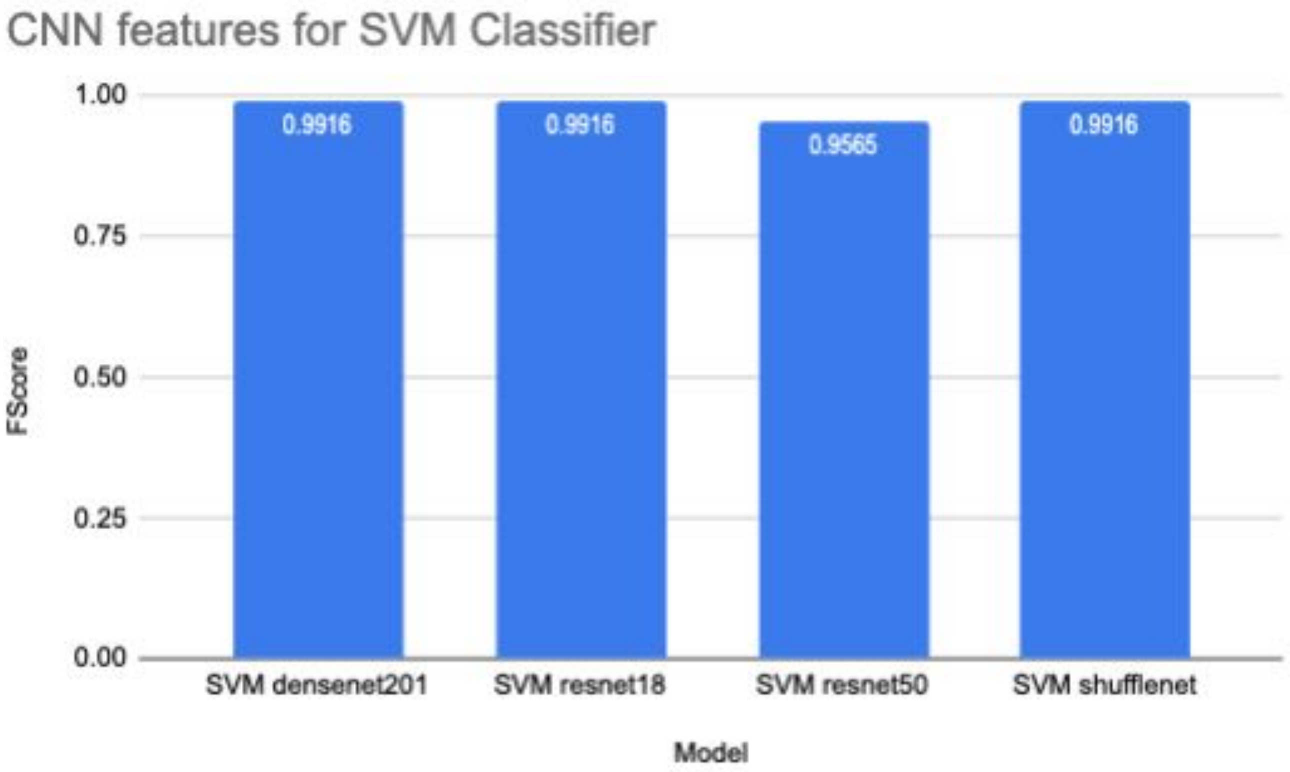
19/20 Patients classified correctly



shufflenet	
Combined Mean ACC	Combined Mean F1
0.9833	*0.9916*

19/20 Patients classified correctly

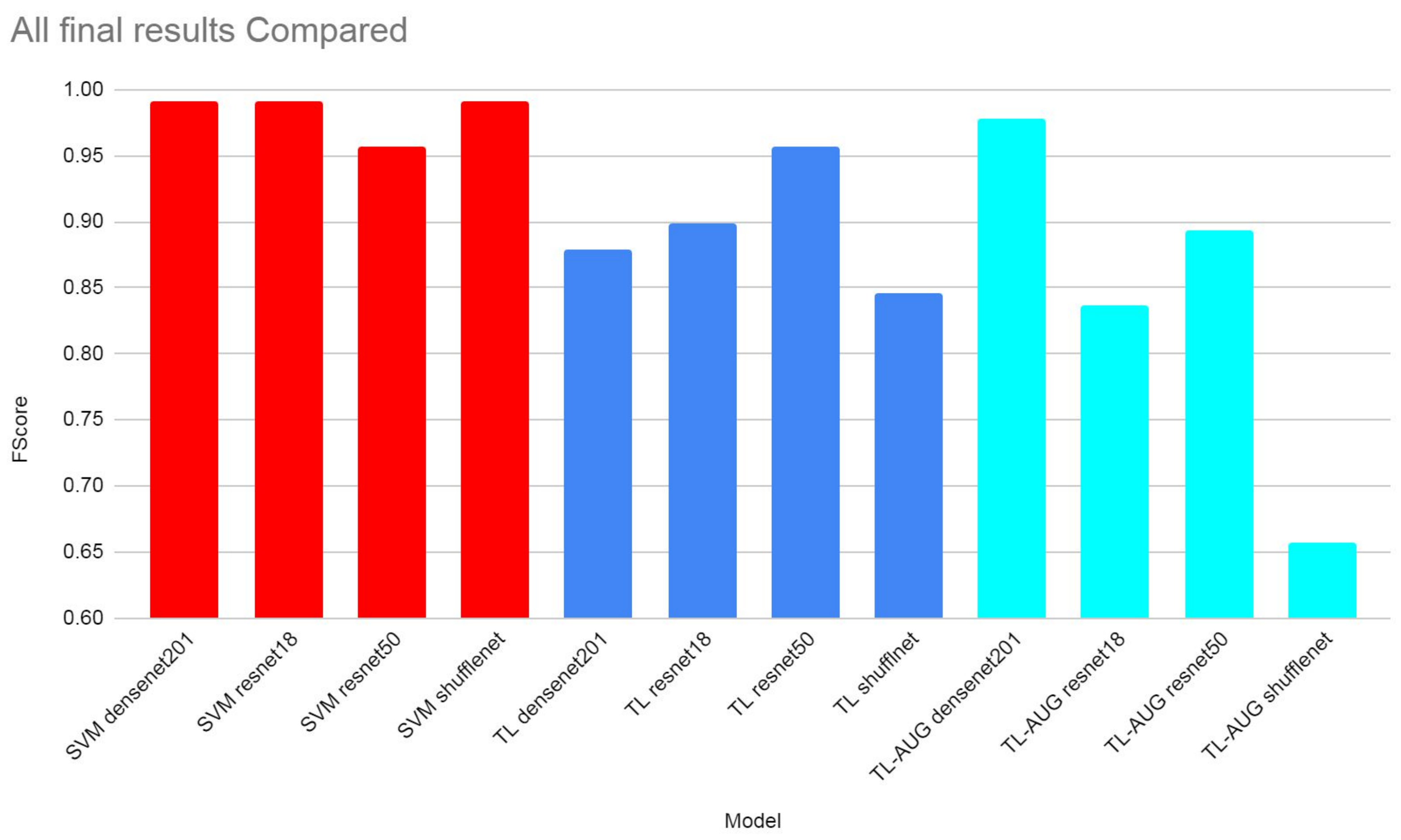




CNN features for SVM Classifier
1.00
0.75
0.50
0.25
0.00
FScore
0.9916
0.9916
0.9565
0.9916
SVM densenet201
SVM resnet18
SVM resnet50
SVM shufflenet
Model

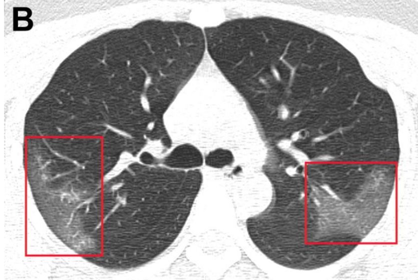
Final Testing Results

- Good results overall, with some models and methods performing very well
- Feature extraction with SVM
  - All networks with SVM performed very well, high 90s for accuracies and high .90s for F_1Scores
- Transfer learning
  - Variability between performance
  - resnet50 and densenet201 performed the best with low std deviations and fairly good accuracies and F_1Scores
- Heat maps
  - Data augmentation helped the CNNs find more relevant features for Covid19
  - Without data augmentation many networks classified on non Covid19 related features
  - resnet50 and densenet201 tended to find the important areas of the lungs more often

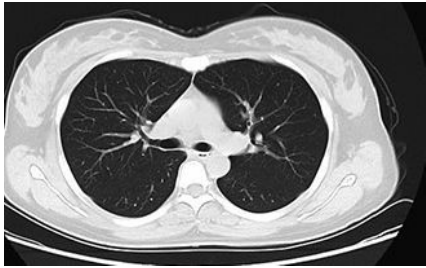


All final results Compared
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
FScore
SVM densenet201
SVM resnet18
SVM resnet50
SVM shufflenet
TL densenet201
TL resnet18
TL resnet50
TL shufflnet
TL-AUG densenet201
TL-AUG resnet18
TL-AUG resnet50
TL-AUG shufflenet
Model

Conclusion and Discussion



- Results were very good and showed that CT scans could be useful in diagnosing Covid19 but…
- Covid19 can't actually be purely diagnosed from the lungs
  - Covid19 can cause pneumonia but not always
  - Pneumonia is really what we are trying to identify in the lungs



- Issues with the datasets…
  - We know somewhat what pneumonia looks like due to annotations on some images
  - No pneumonia features in some UCSD Covid19 CT images
  - Pneumonia features in some UCSD non-Covid19 CT images

Questions?

Additional Data

resnet50			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
0.8249	*0.8385*	*0.3521*	*0.3434*
resnet18			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
0.7942	*0.8105*	*0.3728*	*0.3637*
densenet201			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
0.817	*0.8331*	*0.3543*	*0.3432*
shufflenet			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
0.7738	*0.7931*	*0.3823*	*0.3726*

resnet50			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
92.62	*0.93*	*4.19*	*0.04*
resnet18			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
90.16	*0.91*	*4.19*	*0.02*
densenet201			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
91.72	*0.92*	*6.52*	*0.07*
shufflenet			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
86.13	*0.86*	*10.16*	*0.12*

resnet50			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
91.4094	*0.91*	*6.6845*	*0.0746*
resnet18			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
92.7517	*0.9326*	*4.7076*	*0.0445*
densenet201			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
94.36	*0.9464*	*4.7362*	*0.0451*
shufflenet			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
89.7987	*0.9019*	*6.1584*	*0.0617*

resnet50			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
93.62	*0.94*	*6.17*	*0.06*
resnet18			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
95.44	*0.96*	*8.02*	*0.07*
densenet201			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
96.2	*0.96*	*4.95*	*0.05*
shufflenet			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
95.97	*0.88*	*6.45*	*0.06*

resnet50			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
95.9732	*0.9644*	*6.8935*	*0.0603*
resnet18			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
95.5705	*0.9607*	*6.0141*	*0.0512*
densenet201			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
96.7785	*0.9697*	*5.3775*	*0.0504*
shufflenet			
Mean ACC	Mean Avg F1	sDev ACC	sDev Avg F1
95.5705	*0.9586*	*6.0514*	*0.0571*