

CV Assignment-2

COVID19 Classification Challenge

Report

Submitted by:
Soham Patil
16CO249
Bidyadhar Mohanty
16CO212

Experiments conducted:

- The numpy array was given had images of size 512x512 and local ram as well as RAM on Kaggle was unable to process due to lack of resources.
- We resized the images to 256x256.
- While trying to implement custom model with 10 layers the output accuracy was low.
- There is a huge imbalance in dataset. We have very few images in covid19 class.
- We performed augmentation for better results.
- After performing augmentation, we got an accuracy of 92%.
- But when we visualized the output through grad cam visualization. We figured that our model was detecting the alphabets R ,B1 etc.
- Reading the literature of covid19 we came to a conclusion that the heatmap or features should be concentrated more on chest of the x-ray image.
- So, we chose to use transfer learning. VGG19 was used for training.
- Imagenet weights were used.
- The output of the grad cam showed that the model detecting from the chest region of the image.

Results:

Model: Transfer Learning on VGG

VGG19 pretrained model was used in the experiment.

Loss: categorical_crossentropy

Optimizer: Adam

Model weights with best validation accuracy was saved

Epochs =40

Base Weights are taken from imagenet

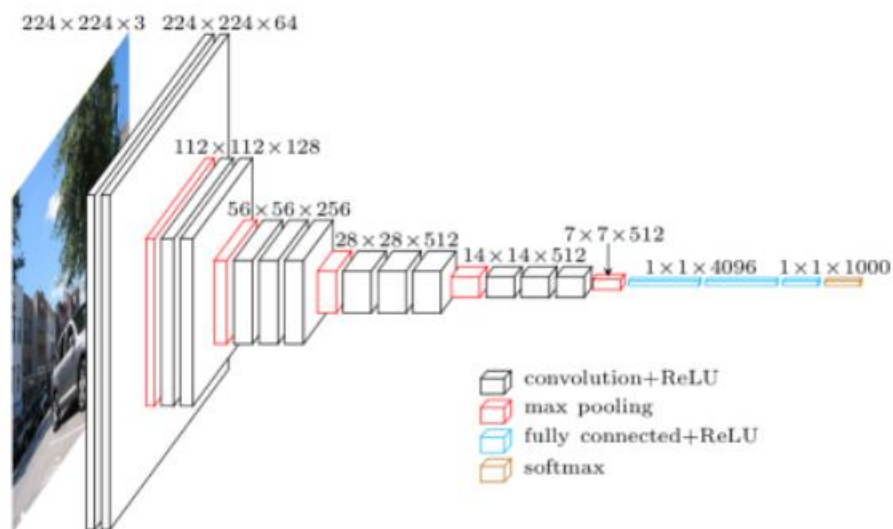


Figure 1: A visualization of the VGG architecture ([source](#)).

Performance

Accuracy – 0.97198

	precision	recall	f1-score	support
Covid-19	0.94286	0.97059	0.95652	34
Normal	0.97642	0.96279	0.96956	215
Pneumonia	0.97235	0.98140	0.97685	215
Macro Avg	0.96387	0.97159	0.96764	464
Weighted Avg	0.97207	0.97198	0.97198	464

Screenshot of Results

```
>
      precision    recall  f1-score   support

0      0.94286      0.97059      0.95652         34
1      0.97642      0.96279      0.96956        215
2      0.97235      0.98140      0.97685        215

   accuracy                   0.97198         464
  macro avg      0.96387      0.97159      0.96764         464
 weighted avg      0.97207      0.97198      0.97198         464
```

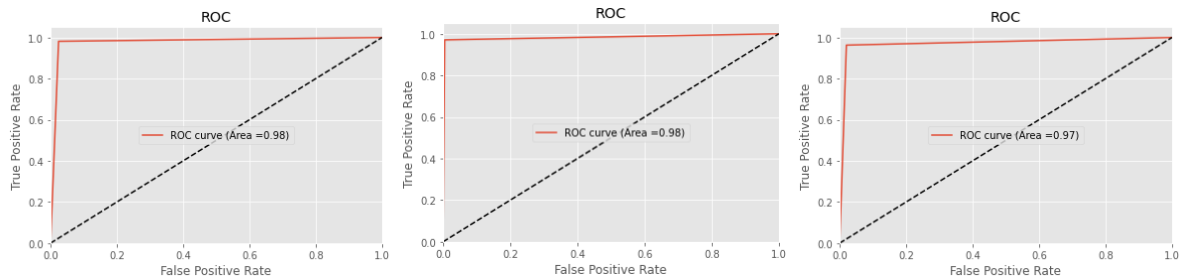
Confusion Matrix

```
# report = metrics.classification_report(true_classes, predicted_classes, target_names=class_labels)
print(matrix)
```

```
>
[[ 33  1  0]
 [ 2 207  6]
 [ 0  4 211]]
```

ROC value:

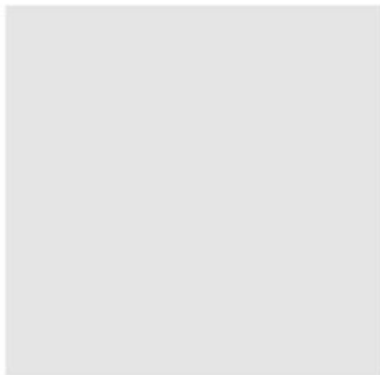
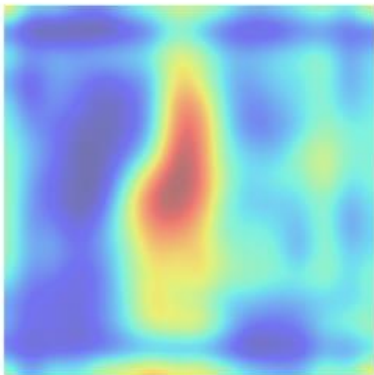
{0: 0.98296853625171, 1: 0.9713551881946391, 2: 0.9786494816475203}



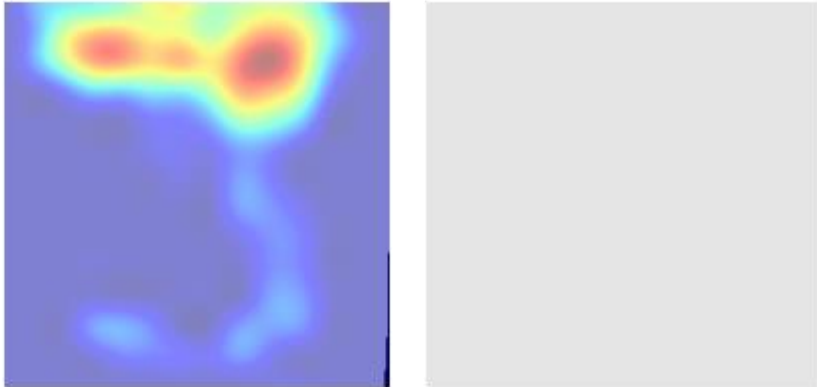
Output Visualization with GradCam (GradCAM-with-keras)

Heatmap of Covid

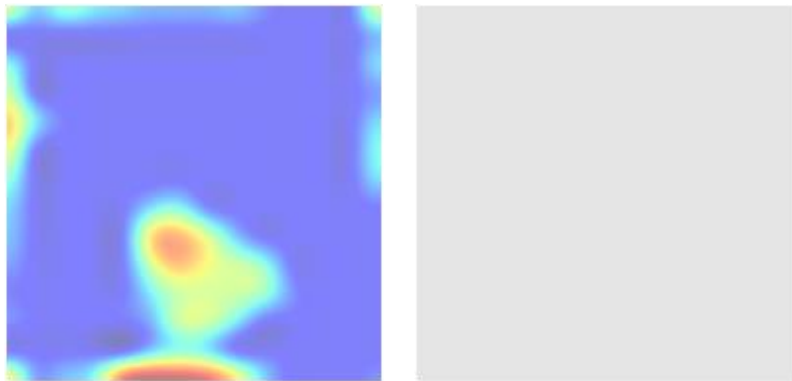
```
[49] f, ax = plt.subplots(**subprot_args)
      for i in range(len(cam)):
          heatmap = np.uint8(cm.jet(cam[i])[..., :3] * 255)
          # ax[i].imshow(images[i])
          ax[i].imshow(heatmap, cmap='jet', alpha=0.5)
      plt.tight_layout()
      plt.show()
```



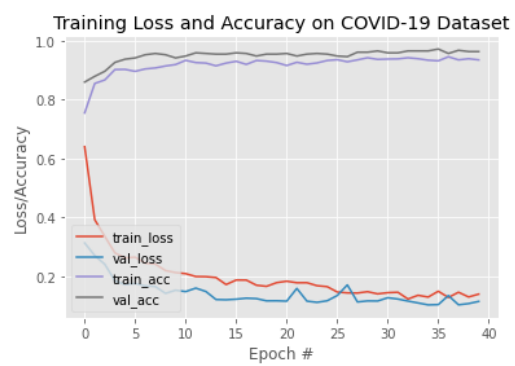
Heatmap of Normal



Heatmap of pneumonia



Training Loss



Libraries

All necessary installs are written in the jupyter notebook.

Tensorflow 2.0.2

Other requirements:

RAM \geq 13GB

(if there is collision please contact us at sohampatil798@gmail.com)

Observations:

- Augmentation of covid class images is needed to get good accuracy.
- Images in the Dataset contains letters.
- Smaller models learn these letters instead.
- Through gradCam we can stop treating deep learning model like a black box. GradCam visualization helps in identifying if the features vector extracted at fully connected layer contains features which we need to analyze.
- Grad Cam source link given in the readme doesn't work with tensorflow >2.0.
- We need to use [this](#) library.