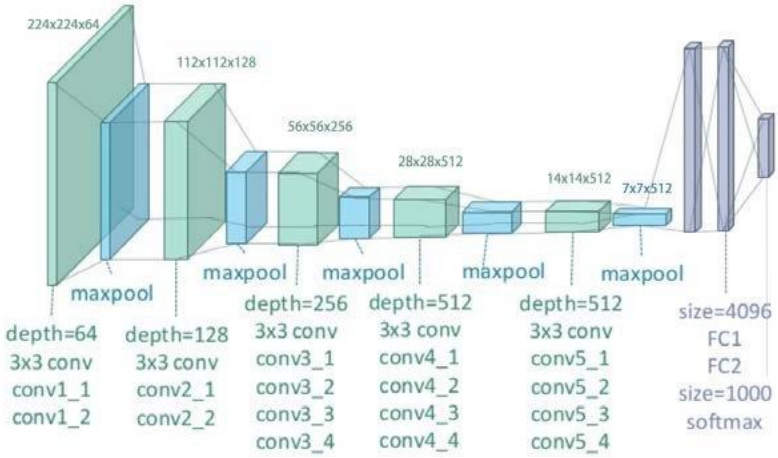
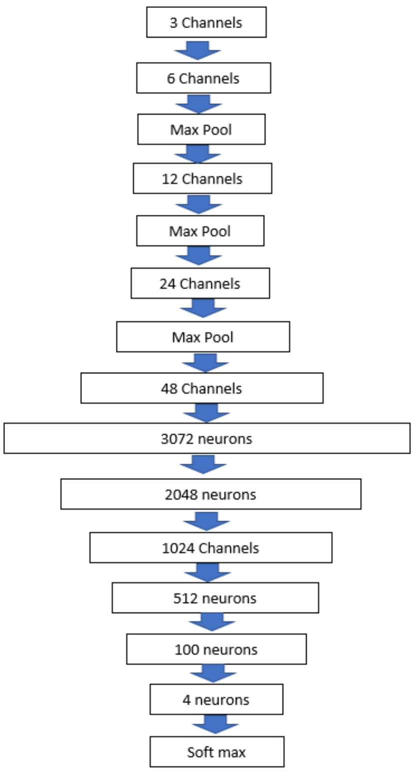
For this multi-class classification problem, we mainly constructed the network in a classical CV deep learning network – VGG (as the below image shows). All our models are having the same structure as VGG, while we only fine-tuned the network by:

- Train by different batch size & epochs.
- The number of neurons in each layer.
- With or without batch normalization layers.



We have totally 4 groups of different types of training datasets, which includes:

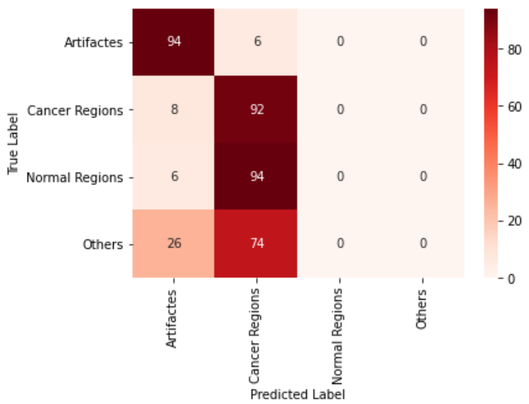
- Data 1: The 6529 images excluding the test images, with imbalanced class and non-divided batch.
- Data 2: The 1087 images excluding the test images, with balanced class and non-divided batch.
- Data 3: The 6529 images excluding the test images, with imbalanced class and batch size 40.
- Data 4: The 1087 images excluding the test images, with balanced class and batch size 40.



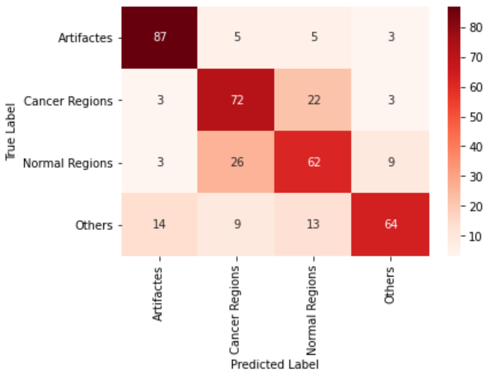
Base CNN Model

The 1st model we constructed is as the left diagrams shows. It is a model with a CNN connected with DNN having the drop-out rate as 0.1.

We have applied the Data 1 to train this model by 400 epochs. But due to the high imbalance of the dataset, the result is quite poor (as below confusion map shows).



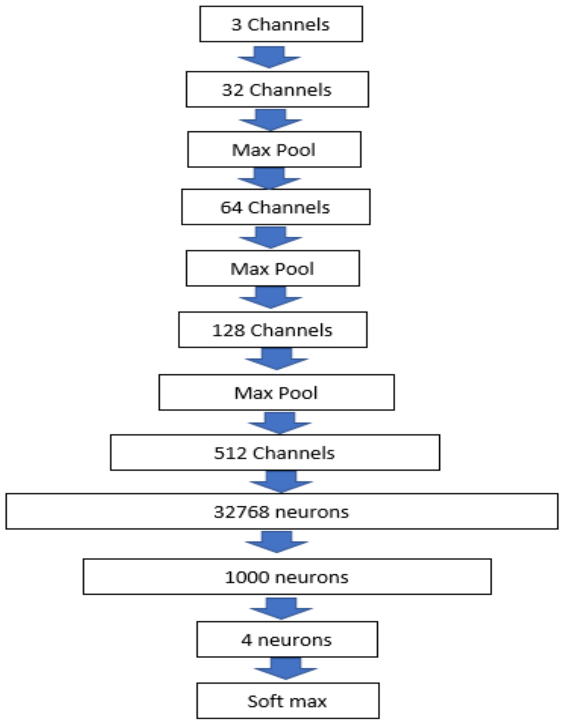
So, we then applied the Data 2, which have the balanced class data set for training, using the epochs as 500. As the



below confusion map shows, the result is significantly improved than the previous one, even the accuracy is still not qualified enough.

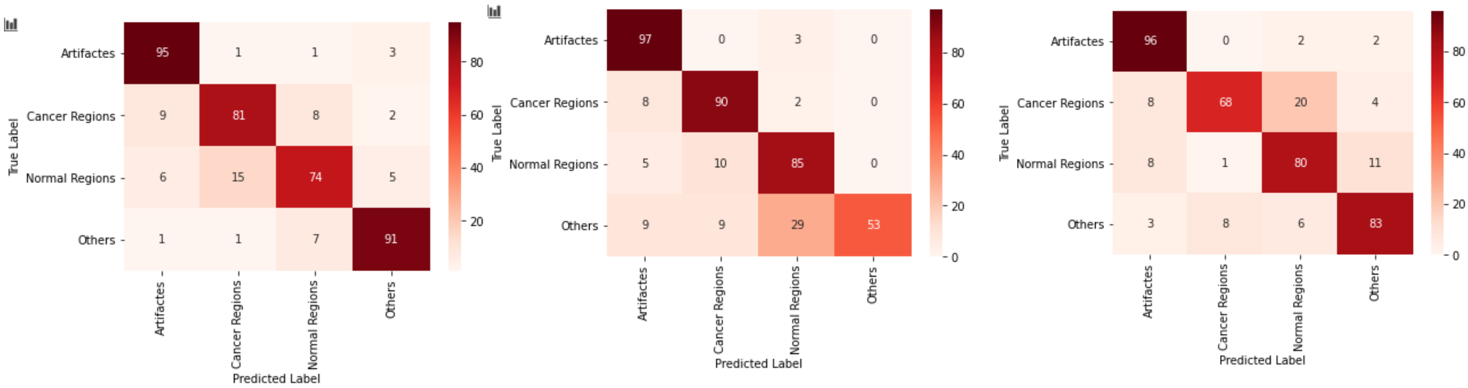
VGG(Without batch normalization)

We then tried a typical VGG net for this classification problem. The structure is as shown below. Here, we have increased the channels (features) in the CNN part, and reduced the layers in the DNN part. The drop-out rate is still 0.1.



We have applied Data 2, Data 3 & Data 4 to this model for comparison. Finally, all the result of this 3 different trained models are having the accuracy more than 80%. **So, we may conclude that in the current case, the more the features CNN extracted, the easier the model will reach a good performance on classification.** What's more, **the Data 2, which is the non-divided batch balanced dataset has the best performance overall, having the accuracy as 85%.**

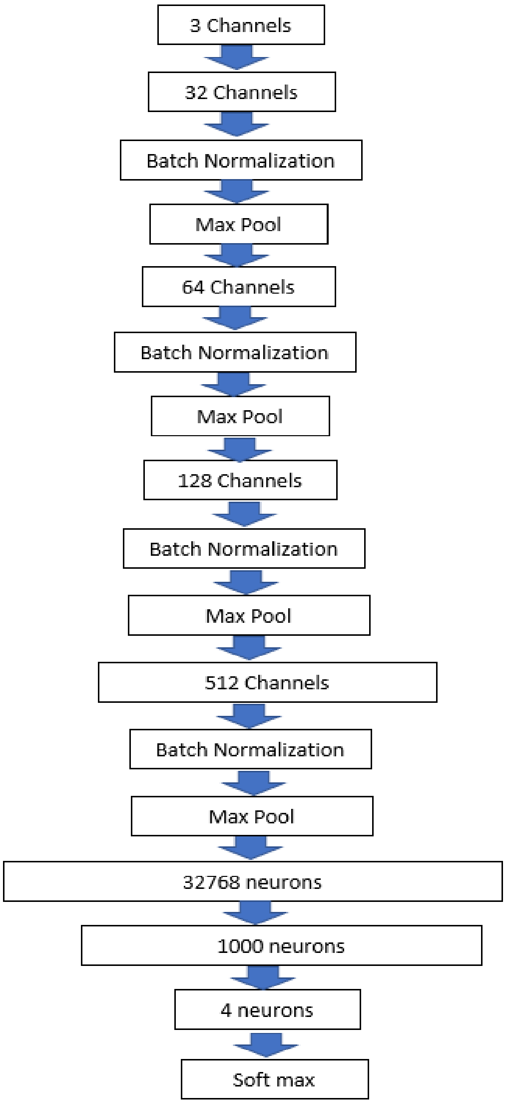
The below 3 confusion maps are the classification result for the model trained by Data 2, Data 3 and Data 4 respectively, from left to right.

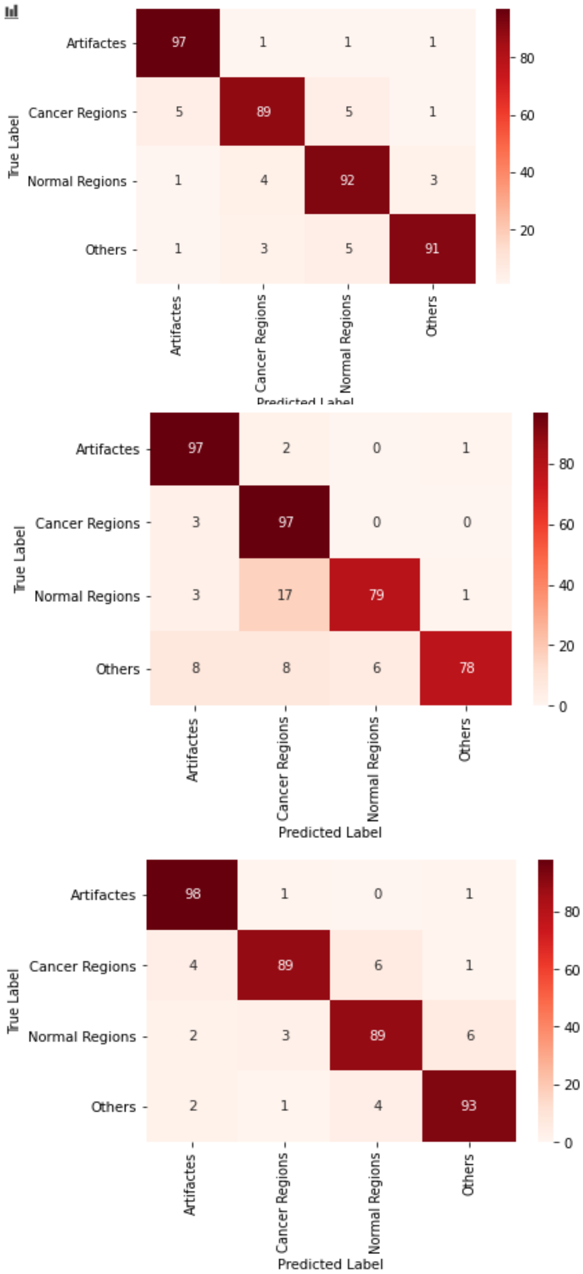


VGG (With Batch Normalization)

We then constructed a model with the batch normalization layers in the CNN part, while we kept all the other structures and parameters as the same as the previous VGG. The structure is as shown in the below diagram.

For this model, we have applied the Data 2, Data 3 and Data 4 for separate training. The result for those models are illustrates at the below right side, from top to bottom, corresponding to the model trained by Data 2, Data 3, and Data 4 respectively.

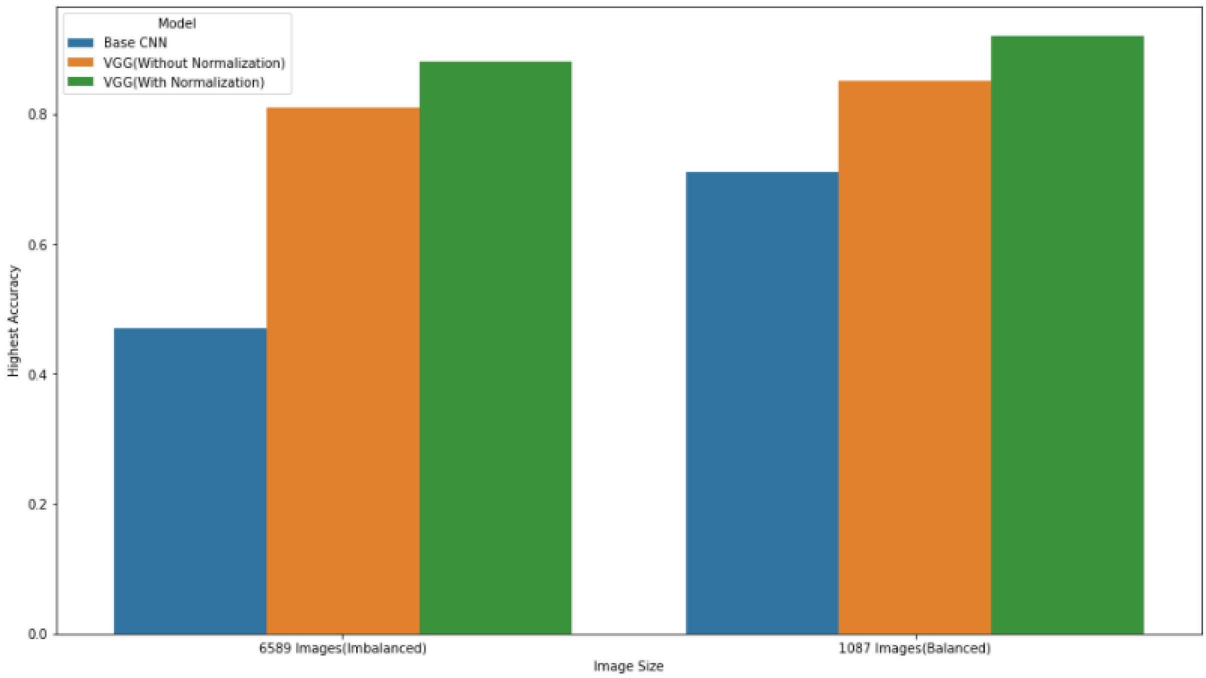




The accuracy here has been significantly improved once again comparing with the result generated by the VGG net without batch normalization, having the accuracy as 92%, 89% and 92% respectively.

Overall Conclusion

- In this scenario, the more the feature extracted by CNN, the easier the model to be trained to have a better performance.
- The more the balanced training dataset, the better result the model will generate.
- The batch normalization performs as an essential role in image classification problem by dealing with outlier with Z-score, which can reduce the influence from the noise in the image to our result.
- By principle, training the data by batch can reduce the impact from the outliers, but in this scenario, it seems that this has not improved our result.
- The below bar plot illustrates the performance among different kinds of models and different corresponding data set trained model's result.


Model
Base CNN
VGG(Without Normalization)
VGG(With Normalization)
Highest Accuracy
0.0
0.2
0.4
0.6
0.8
6589 Images(Imbalanced)
1087 Images(Balanced)
Image Size