

# GNR638: Mini Project 1

Avadhoot Gorakh Jadhav(210050027)  
Gohil Megh Hiteshkumar(210050055)

February 2024

## Preprocessing of data:

We first split the data into test and train as per the README file of CUB\_200\_2011. Then for preprocessing, we used the following:

- Resize every image to  $224 \times 224 \times 3$
- The grayscale images were made RGB by concatenating the channels
- The images were normalized using mean and standard deviation.

## Comparison of different models :

In this project, the first task we did was to compare different ImageNet models with different number of parameters and corresponding test accuracy. The models we have tested are,

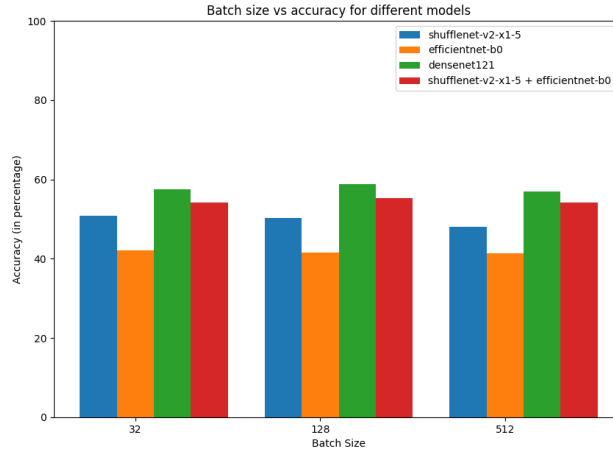
Model Name	Number of Params
Shufflenet_v2_x1_5	2683624
EfficientNet-B0	4263748
DenseNet121	7158856
ShuffleNet + EfficientNet	6947372

## Loss, optimizer :

Loss used for all models is cross entropy loss and optimizer used for all models is adam optimizer. The models were trained for 100 epochs.

## Accuracy vs batch size for all models :

Batch size played important role in accuracy, as batch size increased we can see change in accuracy for all models in . One more key observation regarding the models and accuracy is that higher number of parameters gives better performance.



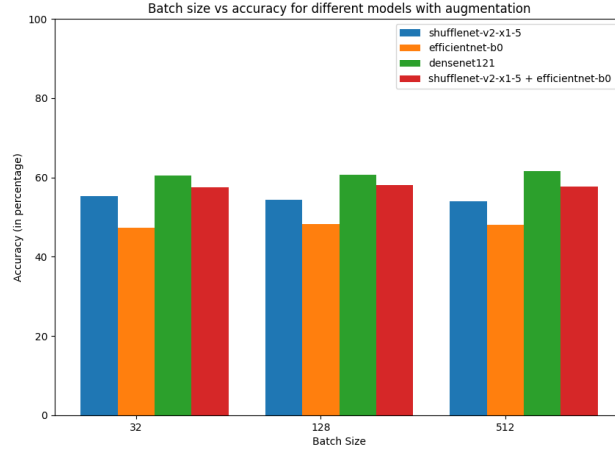
Here **DenseNet121** with **7.1 million parameters** gave the **best performance among these models with accuracy around 61.53 %**. In this project, we tried to use two pretrained models simultaneously (with less than 10M params). For this purpose, sequentially used ShuffleNet\_v2\_x1\_5 and EfficientNet\_B0. The results were better than just training two individual models but not at par with the model with same number of parameters.

## 1 Data Augmentation

In order to increase the accuracy further we used the data augmentation as follows:

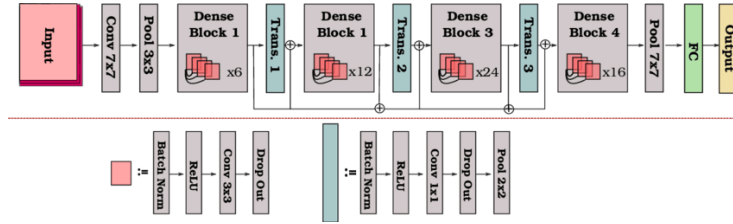
- Random rotation angle selected randomly between -10 to 10
- Random horizontal flip

Use of data augmentation increased the accuracy for all the models which can be seen in the following graph.



## 2 DenseNet121 model

The architecture of our final Model is as follows:



This model was trained for 100 epochs, with learning rate 0.001 and using Adam optimizer. The Data used here was augmented data and hence there were double size of training data. after every 10 epochs the accuracy of the model was noted and loss of model was calculated in each of the epoch. The graphs for loss and accuracy of model with epochs is as follows:

