# **Final Project**

## **Introduction to Deep Learning**

### Jay Rathod(23B3973)

#### Project ID: 147

Fine-grained classification:

1. Download the dataset: <https://data.caltech.edu/records/65de6-vp158/files/CUB_200_2011.tgz?download=1>

this dataset is a .tgz file, of 1.2 GB, which contains 200 bird classes, in total containing around 12000 images. We are given the splitting data, few images in each class are for test, rest for train data-frames.

2. Downscale the images to (224, 224, 3).

3. The parameters should be less than 10M.

3. We would be rescaling the image later to less than 1 and flipping it horizontally.

6. Please use: tensorflow(latest).keras, pandas, sklearn, numpy, os, matplotlib and tensorflow libraries to process the image and data evaluation also use models and optimizers in the packages.

7. We would be extracting the .tgz dataset file(1.2 GB), so import tarfile.

8. Also import os for joining the paths further in the code,

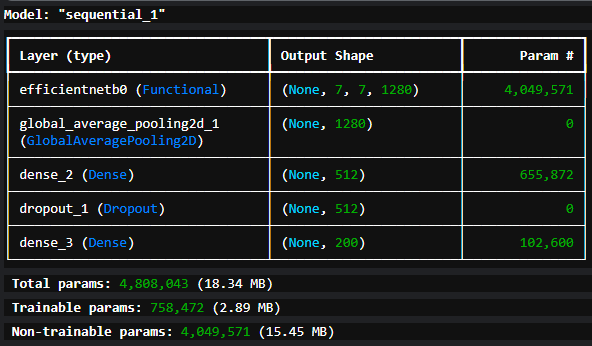
9. We will use tranfer learning using a pre-trained model of efficientNetB0.

Link to Saved Checkpoint and GitHub Repo Link to GitHub Repository here Link to Report and Checkpoint here

Model Structure:

• Download the train dataset from the above link. And add the input file(.tgz) in Kaggle. Name it ‘CUB\_200\_2011’ and copy its path to paste in tgz\_path variable in the code.

• Clone/Download the Repo git clone <github link>

• Extract the dataset file to working path of Kaggle, named as ‘extracted\_dataset’ using tarfile library of python.

• Load the data, as images\_path and add labels to it, provided in the dataset.

• Process the data/images, like rescaling it and reading its path.

• Using the train\_test\_split.txt file in the dataset to split the data into train and test, it labels the respective image as 1(if its for training) and 0(for test).

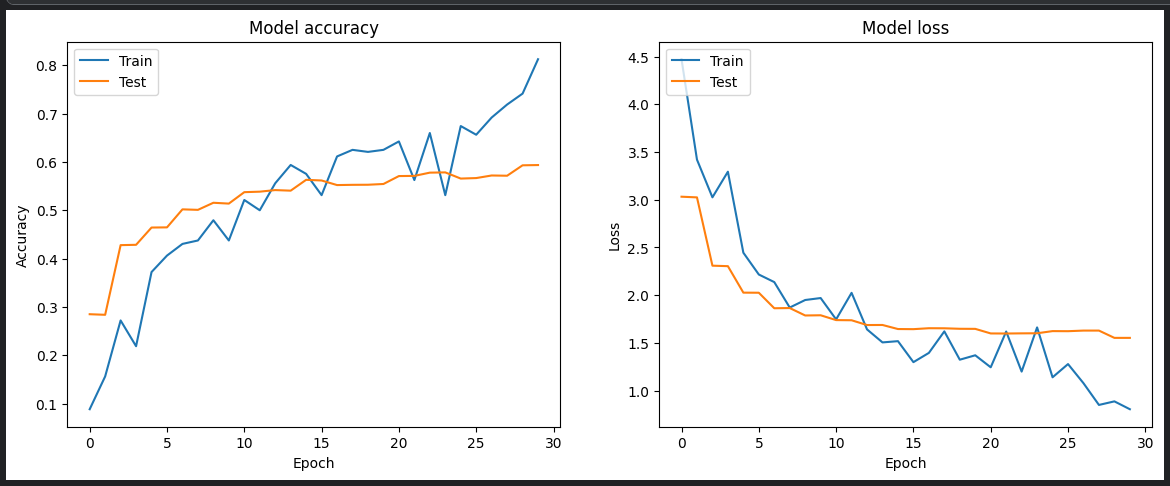
• Define the base model of efficientNetB0, after downloading the .h5 file of the pre-trained model in the same directory and uploading on Kaggle, as a model. then construct the layers for this model.

running model.summary() shows the model in tabular form, as shown in the above image.

• Ensure that the max params limit is 10M.

• Compile the model, and train it using model.fit(…), with batch size of 32 and 30 epochs.

• Using matplotlib library of python to plot the graph of model accuracy and model loss.



Device Specification

• 12 Gen Intel® Core™ i5-1230U CPU

• no GPU, since we used Kaggle.com for training the model

• 8 GB 4200 MHz RAM

Training Details

• train\_test\_split.txt file in the dataset, which labels each images to 1(for train) and 0(for test).

• Total Number of image = 11,788

• Total Number of parameters Model = 4.8M

• Batch Size = 32

• Epochs Trained = 30

• Optimizer Used - Adam (lr=0.001)

• Total Time Taken = 4.5hr

• Train accuracy = 81.25%(approx.)

• Test accuracy = 59.35%(approx.)

• Train Iter Batches per second = 187(For 5.2 params)