# Introduction

The task assigned was based on transfer learning which is to train a pre-trained model for a new dataset with images less than 1000 images for training. Keras framework is used for the task which has TensorFlow backend and has numerous use-ready tools.

The basic structure of the code has been taken from the code reference mentioned in the references. The model used is MobileNet version 1 which is one of the smallest-sized model trained on the “imagenet” dataset which contains over 14 million images. This model was trained on 2 datasets namely- a subset of “Animal Image Dataset(DOG, CAT, and PANDA)”[1] obtained from Kaggle and a subset of the “Animals-10” dataset also obtained from Kaggle.

Animal Image Dataset subset contained 999 images over 3 classes for training and 600 images over 3 classes for testing. Animals-10 subset contained 996 images over 4 classes for training and 1400 images over 4 classes over 4 classes. The classes are balanced in both datasets.

For training the model, MobileNetV1 was taken without the top layer and appended with a GlobalMaxPooling, 2/1 Dense, and a final Dense prediction layer. Data Augmentation was done using ImageDataGenerator from Keras with rotation, shear, zoom, horizontal flip, MobileNet preprocessing function, and validation split ratio of 0.2.

For optimizer, SGD has been used to have a decaying learning rate because after a few iterations without decay, the loss increases, and accuracy decreases with both datasets.

The two datasets gave satisfactory results with accuracy > 80% in testing. The second dataset shows overfitting unlike the first dataset as the training and validation accuracies are 93% and 95% respectively and the testing dataset gives an accuracy of 83%.

Although the imagenet dataset contains species-specific data of dog and elephant, and also fine-tuning of the pre-trained layers can be done to improve performance and reduce overfitting if any. The datasets had clear distinct features between classes, so was easy to train in very few iterations. A dataset with classes with similar features would require more tweaking to the pre-trained model and/or other techniques to distinguish the feature like model cascading.

# References

Code Reference :

https://github.com/adityaanantharaman/transfer-learning/blob/master/transfer-learning.ipynb

1. <https://www.kaggle.com/alessiocorrado99/animals10>
2. <https://www.kaggle.com/ashishsaxena2209/animal-image-datasetdog-cat-and-panda>
3. <https://keras.io/guides/transfer_learning/>
4. <https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720>
5. <https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1>
6. <https://keras.io/api/optimizers/>
7. <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>
8. <https://www.tensorflow.org/tutorials/images/transfer_learning>
9. <https://www.pluralsight.com/guides/transfer-learning-in-deep-learning-using-tensorflow-2.0>
10. <https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>
11. <https://www.tensorflow.org/guide/keras/train_and_evaluate>
12. <https://medium.com/sciforce/robust-image-classification-with-a-small-data-set-be4de9897495>
13. <https://paperswithcode.com/task/small-data>