Laboratory Assignment No. 6

Aim: Object detection using Transfer Learning of CNN architectures

a. Load in a pre-trained CNN model trained on a large dataset
b. Freeze parameters (weights) in model's lower convolutional layers
c. Add custom classifier with several layers of trainable parameters to model
d. Train classifier layers on training data available for task
e. Fine-tune hyper parameters and unfreeze more layers as needed

Objective: To load a pre-trained model and improve its performance by Transfer Learning architecture.

Infrastructure: Computer/ Laptop/ Virtual Machine

Software used: Jupyter Notebook/Google Colab, Tensorflow, Kearas

Theory
What Is Transfer Learning?

Transfer learning generally refers to a process where a model trained on one problem is used in some way on a second related problem. One or more layers from the trained model are then used in a new model trained on the problem of interest. Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error.

The weights in re-used layers may be used as the starting point for the training process and adapted in response to the new problem. This usage treats transfer learning as a type of weight initialization scheme. This may be useful when the first related problem has a lot more labelled data than the problem of interest and the similarity in the structure of the problem may be useful in both contexts.

How to Use Pre-Trained Models

Some of these usage patterns as follows:

- **Classifier**: The pre-trained model is used directly to classify new images.
- **Standalone Feature Extractor**: The pre-trained model, or some portion of the model, is used to pre-process images and extract relevant features.
- **Integrated Feature Extractor**: The pre-trained model, or some portion of the model, is integrated into a new model, but layers of the pre-trained model are frozen during training.
- **Weight Initialization**: The pre-trained model, or some portion of the model, is integrated into a new model, and the layers of the pre-trained model are trained in concert with the new model.

It may not be clear as to which usage of the pre-trained model may yield the best results on your new computer vision task, therefore some experimentation may be required.

Ways to Fine tune the model

1. **Feature extraction** – We can use a pre-trained model as a feature extraction mechanism. What we can do is that we can remove the output layer and then use the entire network as a fixed feature extractor for the new data set.
2. **Use the Architecture of the pre-trained model –** What we can do is that we use architecture of the model while we initialize all the weights randomly and train the model according to our dataset again.
3. **Train some layers while freeze others** – Another way to use a pre-trained model is to train is partially. What we can do is we keep the weights of initial layers of the model frozen while we retrain only the higher layers. We can try and test as to how many layers to be frozen and how many to be trained.

Building A Deep Learning-Based Object Detection Model

Training a performing deep learning model for object detection takes a lot of data and computing power. To facilitate the development, we can use transfer learning by fining tuning models pre-trained based on other relevant datasets.

Since there are multiple backend logistics such as paths and hyper parameters to take care of when training a full-scale deep learning model, we can create a central dictionary to store these configuration parameters, including setting up the different paths, installing relevant libraries, and downloading the pre-trained models.

Conclusion: We conclude from the experiment, how to develop a model for a specific application with the help of transfer learning architecture in deep learning.