Transfer learning using Pre-trained model as Feature Extractor

We use ResNet50 deep learning model as the pre-trained model for feature extraction for Transfer Learning.

* To implement Transfer learning, we will remove the last predicting layer of the pre-trained ResNet50 model and replace them with our own predicting layers. FC-T1 and FC\_T2 as shown below
* Weights of ResNet50 pre-trained model is used as feature extractor
* Weights of the pre-trained model are frozen and are not updated during the training

Diagram

Description automatically generated

We do not want to load the last fully connected layers which act as the classifier. We accomplish that by using “**include\_top=False**”. We do this so that **we can add our own fully connected layers on top of the ResNet50 model for our task-specific classification.**

**We freeze the weights of the model by setting trainable as “False”**. This stops any updates to the pre-trained weights during training We do not want to train ResNet layers as we want to leverage the knowledge learned by the deep neural network trained from the previous data set which in our case is “imagenet”

Graphical user interface, text

Description automatically generated with medium confidence

Table

Description automatically generated

We now create our model using Transfer Learning using Pre-trained ResNet50 by adding our own fully connected layer and the final classifier using **softmax** activation function.

Table

Description automatically generated

Fine-tune the pre-trained models

* To implement Transfer learning with fine-tuning, we remove the last predicting layer of the pre-trained model and replace them with our own predicting layers. FC-T1 and FC\_T2 as shown below.
* Initial lower layers of the network learn very generic features from the pre-trained model. To achieve this initial layers weights of pre-trained models frozen and not updated during the training
* Higher layers are used for learning task-specific features. Higher layers of pre-trained models are trainable or fine-tuned
* Improves performance with less training time

Diagram

Description automatically generated

Table

Description automatically generated

We see that we have set the training for early layers of ResNet50 to false and the last few layers of ResNet50 are now trainable.

we now add our own fully connected layer and classifier on top of the ResNet50. We have already removed the last fully connected layer and the classifier layer from ResNet50

Table

Description automatically generated

Future Steps:

* Image data augmentation
* Unfreezing early layers to make it trainable
* Use cross validation [k-fold]
* Use keras optimizer [other than adam]
* Hyperparameter tuning