Semi Supervised Learning

Choosing Model

Though ResNet gave a higher test accuracy on while running supervised learning, CNN gave a better final test accuracy when we carried out semi-supervised using pseudo labelling. The ResNet model overfitted the dataset extremely fast.

Data Augmentation

Augmenting the data by adding transformations like rotation, flip and colour jitter helped reduce overfitting. Data Augmentation along with adding Dropout layers helped boost val_accuracy from 0.61 to 0.66.

Choosing Threshold

I tried multiple threshold's ranging from 0.7 all the way to 0.96, the best result on the test accuracy was given by 0.93. This can be justified by the fact that a threshold of 0.93 would produce pseudo labels on a fairly large number of images from the unlabelled data set while maintaining a high level of accuracy. (13,925 to be exact)

Choosing Epochs

Rather than choosing the number of epochs manually I decided to utilise the earlyCallback method in .fit() function of Keras. I set the monitor to be val_loss and ended the training if the val_loss stopped decreasing in value

Overfitting

One drawback of the semi supervised model is that it's overfitting. If we put the test data as the validation set for the fitting process it is even more clear. When we train the model on the train dataset we get an val_accuracy of approximately 0.66 and a vaL_loss of about 0.95. However when we add pseudo labels to the untrained dataset and train the model on that, accuracy falls to 0.65 and val_loss increases to 1.02. Though this shift is minor, train_loss continues to decrease along with an increase in train_accuracy thus showcasing overfitting

Conclusion

After running a semi-supervised model on both CNN and ResNet it is evident that the model is clearly overfitting and thus a supervised learning based on ResNet is the optimum solution for the given dataset getting us an val_accuracy of over 0.7 and a val_loss of 0.94