We have performed the experiments on Google colab and have submitted the notebooks along with the python files.

Description for feature extraction:-
1.importing all libraries required
2.Setting hyperparameters to train the network
3.Normalize and preprocess images in the dataset to match the network requirements
4.Convert dataset images into pytorch ImageFolder format.
5.Define training function.
6.Download the pretrained alexnet model
7.Update the weights of the layer
8.train the model for 10 iterations
9.save the trained model
10.load the test data using pytorch
11.test model accuracy on test data
12.definition of confusion matrix plot function
13.plot the confusion matrix by calling the above function

Description for fine tuning:-
1.importing all libraries required
2.Setting hyperparameters to train the network
3.Normalize and preprocess images in the dataset to match the network requirements and perform data augmentation for the further part.
4.Convert dataset images into pytorch ImageFolder format.
5.Define training function.
6.Download the pretrained alexnet model
7.Update the weights for all the layer
8.Train the model for 10 iterations without data augmentation and 50 iteration with data augmentation.
9.save the trained model
10.load the test data using pytorch
11.test model accuracy on test data
12.definition of confusion matrix plot function
13.plot the confusion matrix by calling the above function

COMPARISON
Feature extraction corresponds to training only the last layer and thus the accuracy is only 89 %.

But our finetuning corresponds to training all the layers and thus the accuracy increased to 91%.

Thus, we can conclude that feature extraction is useful for training a net and using it for classification, but it is made better with fine tuning.

The accuracy of the network decreased to 89% after data augmentation was performed this may be due the networks trivial structure it is not able to adapt to the augmentation of the data. The network also takes a longer time to converge to a particular accuracy when data augmentation is performed.

ADVANTAGES
Also transfer learning is very effective way of training and using a CNN because if our dataset is less, it is not required to train the net from scratch which might increase the complexity and consumes lot of time.

As the network we used is Alexnet that uses the relu as decision function the training and the testing time of the network was low.

In pytorch the pretrained networks can be loaded and modified easily which makes the transfer learning easier to perform on other datasets.

DISADVANTAGES
Transfer learning is not useful for larger sets of data where the accuracy will decrease substantially.

Feature extraction was performed by SAMPRITI NEOG
Fine tuning and data augmentation were done by CHINMAYI SHIVAREDDY