Task 1: Image classification using a MLFFNN with two hidden layers, and with Deep CNN features for an image as the input to the MLFFNN, with (a) VGGNet as Deep CNN and (b) GoogLeNet as Deep CNN.

Hyper parameters: Learning rate - 0.001 Epochs - 10 FC Layer Hidden nodes - 256 Number of FF layer - 2

VGGNet as Deep CNN

Experiment:

VGGNet is used as the convolutional network to extract feature maps - low dimensional representation of input image which are then fed into a Fully connected neural network to classify the images. 5 classes of images are there in the training set. VGGNet is imported directly from PyTorch Package. The input image is resized to 224 x 224 and normalized with mean [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224, 0.225] before feeding into Convolutional layers as the images are normalized in this way while training the VGGNet.

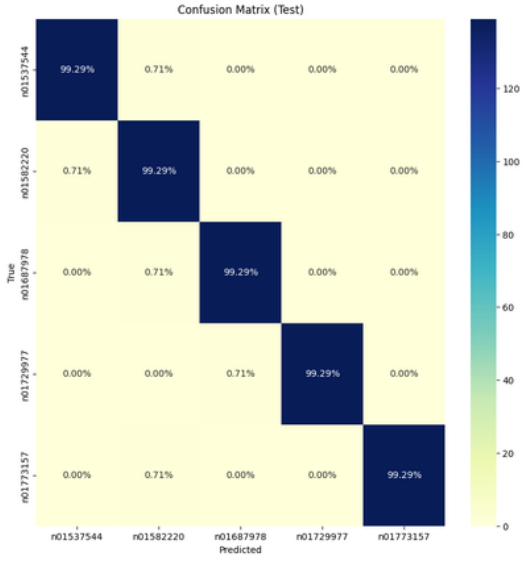
Sample input image after normalization

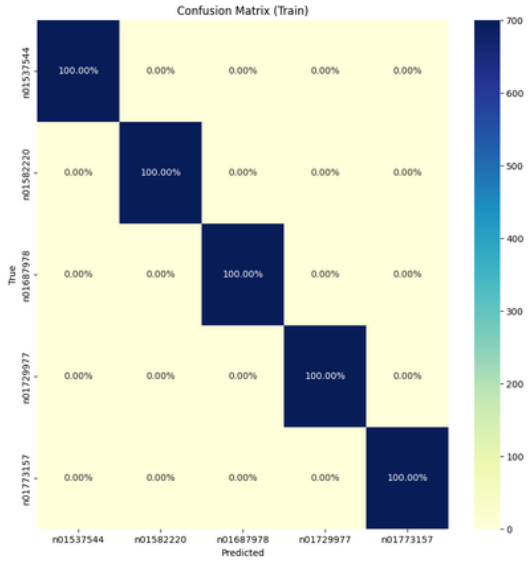
Features:

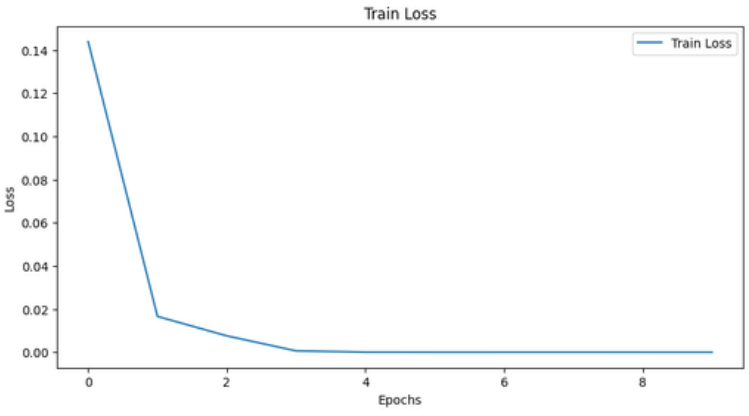
- **Number of feature maps** - 512 with dimension of 7X7
- **Number of inputs node of MLFFNN** - 512 x 7 x 7 = 25088

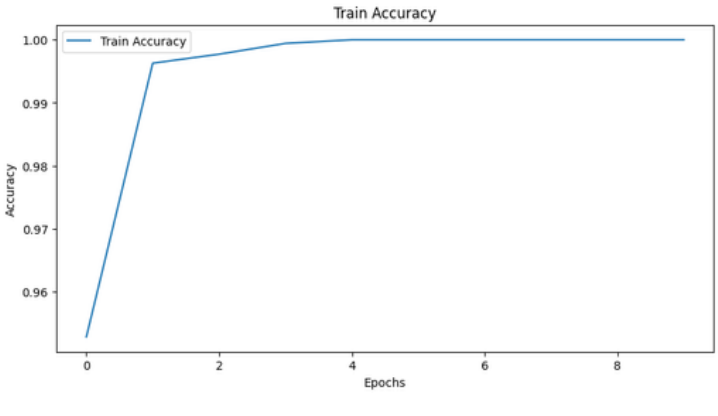
Observations :

- **Train Loss**: 0.0000
- **Train Accuracy** : 100%
- **Test Loss** : 0.0423
- **Test accuracy** : 99.29%









GoogLeNet as Deep CNN.

Experiment:

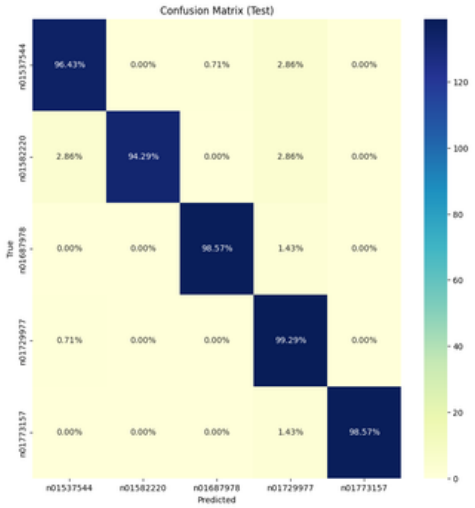
GoogLeNet is used as the convolutional network to extract feature maps - low dimensional representation of input image which are then fed into a Fully connected neural network to classify the images. The same training set and same normalization as in VGGNet is used. The pretrained GoogLeNet is imported from PyTorch package. Since the number of feature maps is almost double as in the case of VGGNet , average global pooling is performed which converts the 2D feature maps to a single number.

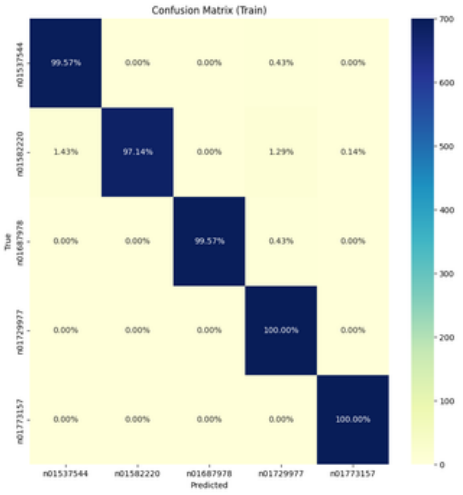
Features:

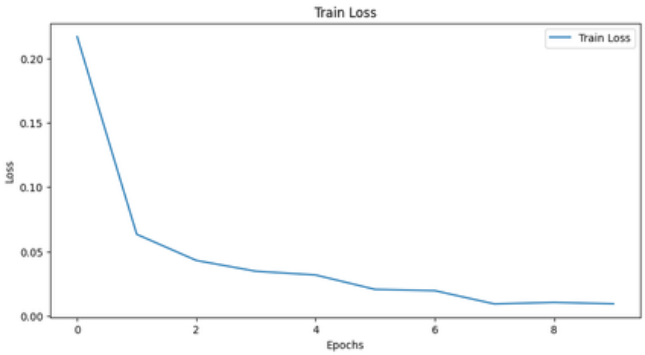
- **Number of feature maps** - 1024 with dimension of 7X7
- **Number of inputs node of MLFFNN** - 1024 x 1 = 1024

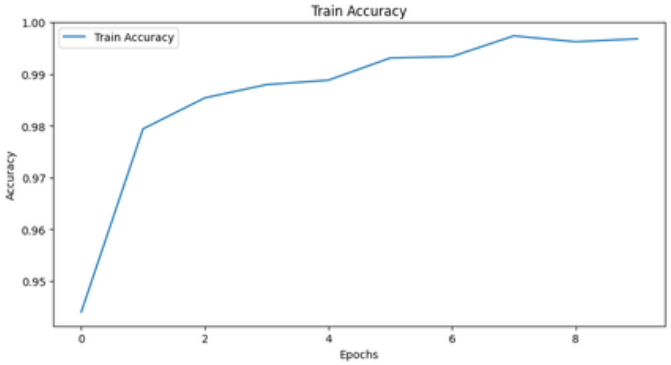
Observations :

- **Train Loss**: 0.0094
- **Train Accuracy** : 99.96%
- **Test Loss :** 0.1203
- **Test accuracy** : 97.43%









Inferences:

- The performance metrics of VGGNet is marginally higher than GoogLeNet.
- From the graphs , it is also observed that VGGNet network converged quickly.
- The zero loss and 100% accuracy in VGGNet suggests that the model has overfit the data but the train accuracy is up to mark to infer that it can generalize well.
- Since the number of features from GoogLeNet is reduced to 1024 by average pooling which is far below the feature dimension of 512 x 7 x 7 in VGGNet, it was able to converge well