Masters in Computer Science

Topics in Machine Learning and NN

(COMP-5011)

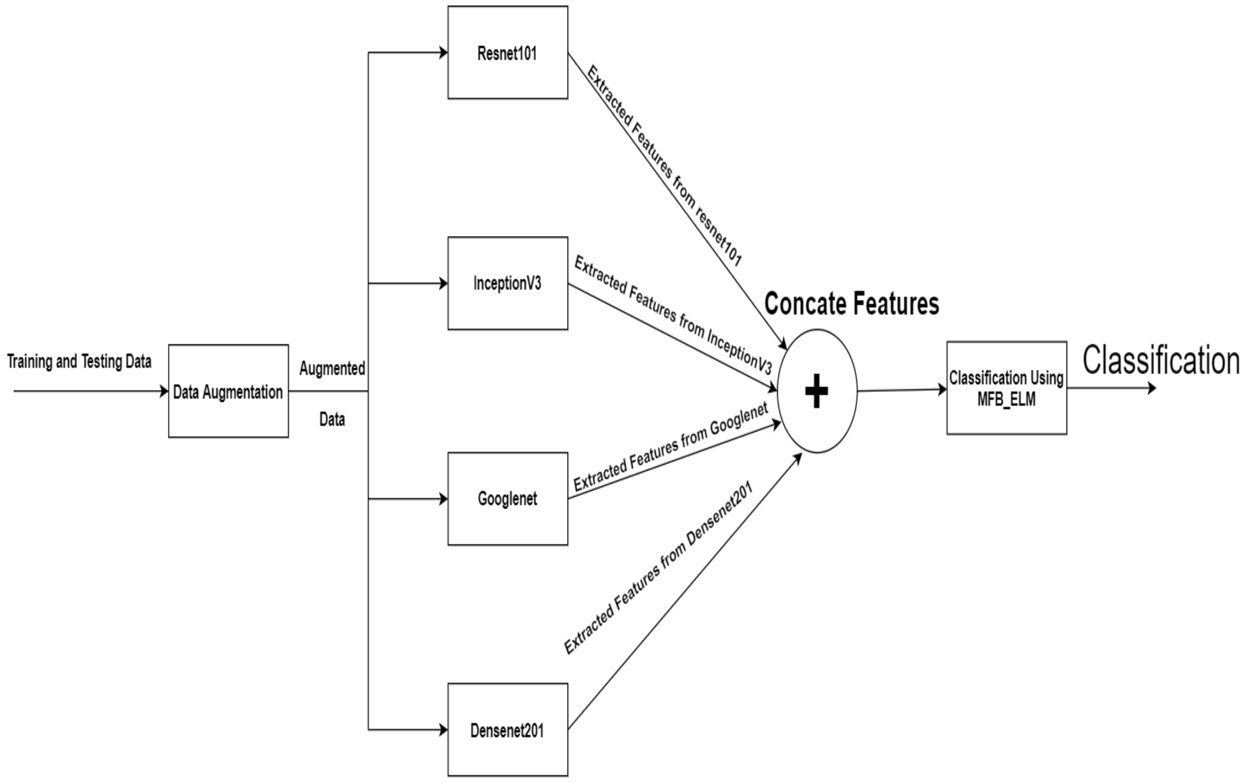
Group:

1. **Jaykumar Nariya (1116571)**

Project:

Task1: Object-centric Image recognition using Caltech101, Caltech256.

Methodology:


Training and Testing Data
Data Augmentation
Augmented
Data
Resnet101
InceptionV3
Googlenet
Densenet201
Extracted Features from resnet101
Extracted Features from InceptionV3
Extracted Features from Googlenet
Extracted Features from Densenet201
Concate Features
+
Classification Using MFB_ELM
Classification

The Following structure or Network is used for classification for all three datasets. We obtained deep features from Resnet101, InceptionV3, Googlenet and Densenet201. These Deep Features are then Combined. I used MFB ELM is used as classifier and trained on using data and used for classification. The Output of the class is provided by the MFB_ELM and As per Project requirement I also test on Elm cuda.

1. Caltech101

Code:

```
tic
% define  folder
Folder = fullfile('C:\Users\Neel\Desktop\DLProject\data', 'caltech101');
disp('steps to output')

% Create ImageDatastore of the dataset for processing in Matlab.
rootFolder = fullfile(Folder, '101_ObjectCategories');
imds = imageDatastore(fullfile(rootFolder), 'LabelSource',
'foldernames','IncludeSubfolders',true);
clear Folder rootFolder;

% Split each label Using 30 images for training and rest for testing
[trainingSet, testingSet] = splitEachLabel(imds, 30);

% Load resnet
disp('1. Loading Pretrained Resnet');
 net = resnet101;

 %set imageSize according to DCNN first input layer
 imageSize = net.Layers(1).InputSize;


%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);
%resetgpu
gpuDevice(1)
```

```
% Get features from resnet
 disp('3. Loading Resnet train features');
 %Extract training features from resnet101 DCNN
 resnet_features_train =
activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

 disp('4. Loading Resnet test features');
 %Extract testing features from resnet101 DCNN
 resnet_features_test =
activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

 %Reshape training and testing features from resnet101
 resnet_features_train =
reshape(resnet_features_train,[1*1*1000,size(resnet_features_train,4)])' ;
 resnet_features_test =
reshape(resnet_features_test,[1*1*1000,size(resnet_features_test,4)])';

 disp('5. Loading Pretrained inceptionv3');
% Load inceptionv3
net = inceptionv3;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb' ,'DataAugmentation',imageAugmenter);

%resetgpu
gpuDevice(1)

disp('6. Loading inceptionv3 train features');
 %Extract training features from inceptionv3 DCNN
inceptionv3_features_train =
activations(net,augImdstraining,'avg_pool','MiniBatchSize',200);

disp('7. Loading inceptionv3 test features');
 %Extract testing features from inceptionv3 DCNN
inceptionv3_features_test =
activations(net,augImdstesting,'avg_pool','MiniBatchSize',200);

%Reshape training and testing features from inceptionv3
inceptionv3_features_train =
reshape(inceptionv3_features_train,[1*1*2048,size(inceptionv3_features_train,
4)])' ;
inceptionv3_features_test =
reshape(inceptionv3_features_test,[1*1*2048,size(inceptionv3_features_test,4)
])';

disp('8. Loading Pretrained googlenet');
% Load inceptionv3
% Load inceptionv3
net = googlenet;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);
%resetgpu
gpuDevice(1)

 disp('9. Loading googlenet train features');
 %Extract training features from googlenet DCNN
  googlenet_features_train = activations(net,augImdstraining,'loss3-
classifier','MiniBatchSize',200);

  disp('10. Loading googlenet test features');
  %Extract testing features from googlenet DCNN
googlenet_features_test = activations(net,augImdstesting,'loss3-
classifier','MiniBatchSize',200);

%Reshape training and testing features from googlenet
googlenet_features_train =
reshape(googlenet_features_train,[1*1*1000,size(googlenet_features_train,4)])
' ;
googlenet_features_test =
reshape(googlenet_features_test,[1*1*1000,size(googlenet_features_test,4)])';

disp('11. Loading Pretrained densenet201');
% Load densenet201
net = densenet201;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%resetgpu
gpuDevice(1)

 disp('12. Loading densenet train features');
 %Extract training features from densenet DCNN
```

```
densenet_features_train =
activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

  disp('13. Loading densenet test features');
  %Extract testing features from densenet DCNN
densenet_features_test =
activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

%Reshape training and testing features from densenet
densenet_features_train =
reshape(densenet_features_train,[1*1*1000,size(densenet_features_train,4)])'
;
densenet_features_test =
reshape(densenet_features_test,[1*1*1000,size(densenet_features_test,4)])';


disp('14. Combining the training features from All DCNN');
% Merge Resnet and googlenet deep features for training
x = horzcat(resnet_features_train,googlenet_features_train);
% Merge densenet and inceptionv3 deep features for training
w = horzcat(inceptionv3_features_train, densenet_features_train);
% Merge all deep features for training
new_F_train = horzcat(x,w);

disp('15. Combining the testing features from All DCNN');
% Merge Resnet and googlenet deep features for testing
y = horzcat(resnet_features_test,googlenet_features_test);
% Merge inceptionv3 and densenet deep features for testing
z = horzcat(inceptionv3_features_test, densenet_features_test);
% Merge all deep features for testing
new_F_test = horzcat(y,z);


%Get Train Label from training dataset
train_labels = grp2idx(trainingSet.Labels);

%Get Test Label from testing dataset
test_labels = grp2idx(testingSet.Labels);

disp('16. creating training and testing dataset for elm');
%Give labels to training and testing features
 training = horzcat(train_labels,new_F_train);
 testing = horzcat( test_labels,new_F_test);



C = 2^-10;
%disp('17. Classification using ELM');
%[TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy] =
ELM(training, testing, 1, 10000, 'sig', C);

%define Number subnetworknode for MFB_ELM
number_subnetwork_node =10;
```

```
disp('18. Classification using MFB_ELM')
[train_time,
train_accuracy,test_accuracy]=MFB_ELM(training,testing,1,1,'sig',number_subne
twork_node,C);
fprintf('Training Time = %f\n',train_time);
fprintf('Training Accuracy = %f\n',train_accuracy);
fprintf('Testing Accuracy = %f\n',test_accuracy);




timeElapsed = toc;
```

Results:

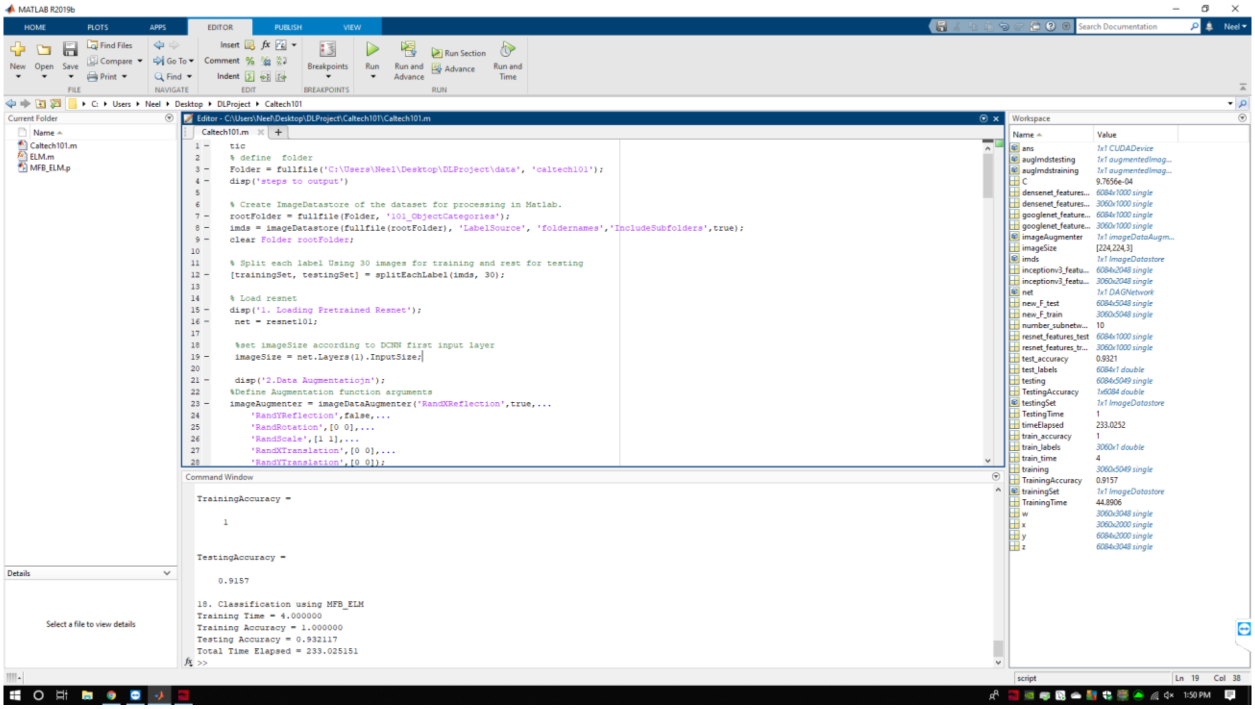
- **Top 1-Accuracy: 93.23%**
- **Accuracy in each Trial:**

Classifier	Run 1	Run 2	Run 3	Avg
ELM(Cuda)	0.9124	0.9159	0.9265	0.9265
ELM(CPU)	0.9157	0.9065	0.9246	0.9246
MFB_ELM	0.9321	0.9323	0.9330	0.9323

- **GPU: NVIDIA QUADRO RTX 8000 48GB**
- **Total time: = 22.1547minutes each run**

Outputs:

1.Run (Testing Acc: - 93.2117%)

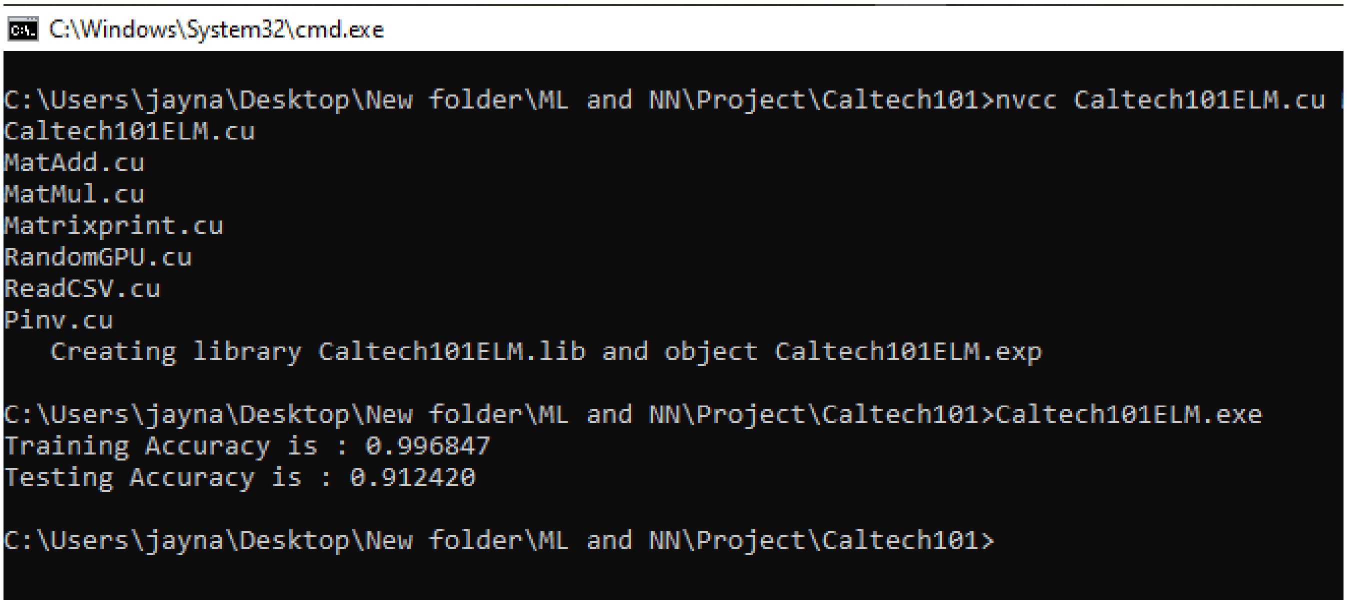


```matlab
tic
% define  folder
Folder = fullfile('C:\Users\Neel\Desktop\DLProject\data', 'caltech101');
disp('steps to output')

% Create ImageDatastore of the dataset for processing in Matlab.
rootFolder = fullfile(Folder, '101_ObjectCategories');
imds = imageDatastore(fullfile(rootFolder), 'LabelSource', 'foldernames','IncludeSubfolders',true);
clear Folder rootFolder;

% Split each label Using 30 images for training and rest for testing
[trainingSet, testingSet] = splitEachLabel(imds, 30);

% Load resnet
disp('1. Loading Pretrained Resnet');
 net = resnet101;

 %set imageSize according to DCNN first input layer
 imageSize = net.Layers(1).InputSize;

 disp('2.Data Augmentatiojn');
%Define Augmentation function arguments
imageAugmenter = imageDataAugmenter('RandXReflection',true,...
    'RandYReflection',false,...
    'RandRotation',[0 0],...
    'RandScale',[1 1],...
    'RandXTranslation',[0 0],...
    'RandYTranslation',[0 0]);
```

```
TrainingAccuracy =

     1


TestingAccuracy =

    0.9157

18. Classification using MFB_ELM
Training Time = 4.000000
Training Accuracy = 1.000000
Testing Accuracy = 0.932117
Total Time Elapsed = 233.025151
>>
```

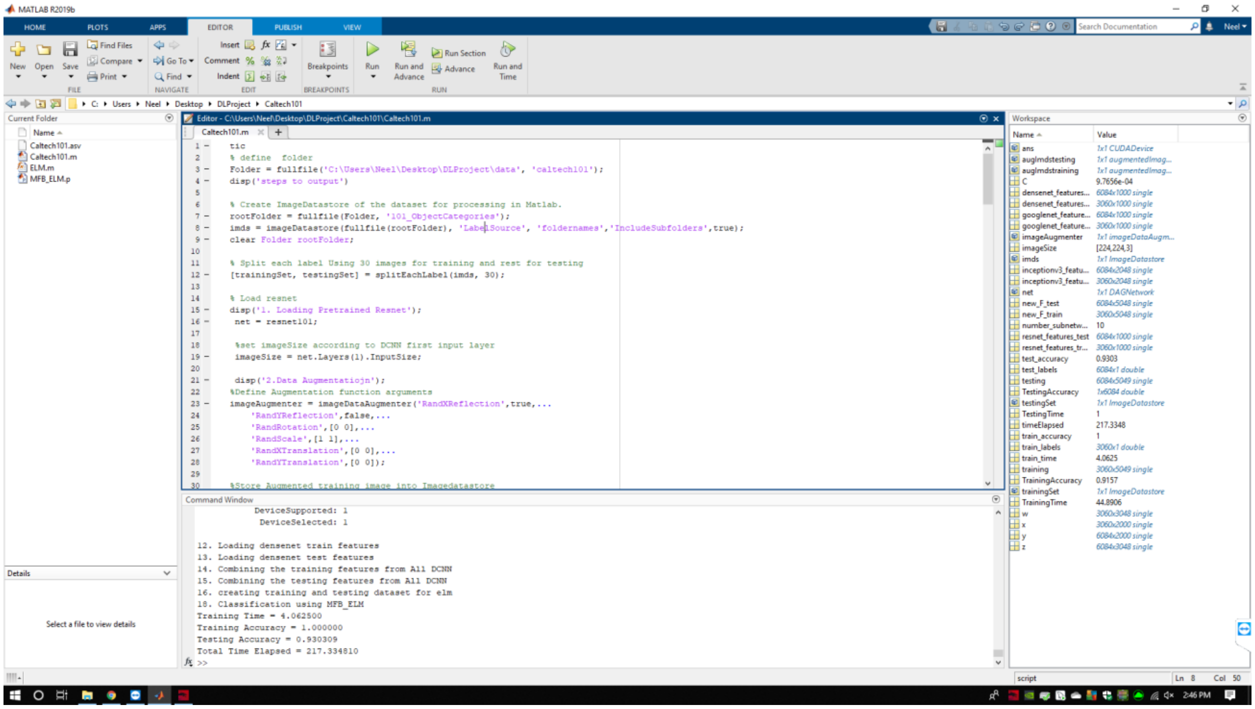
Name	Value
ans	1x1 CUDADevice
augImdstesting	1x1 augmentedImag...
augImdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	6084x1000 single
densenet_features...	3060x1000 single
googlenet_feature...	6084x1000 single
googlenet_feature...	3060x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	6084x2048 single
inceptionv3_featu...	3060x2048 single
net	1x1 DAGNetwork
new_F_test	6084x5048 single
new_F_train	3060x5048 single
number_subnetw...	10
resnet_features_test	6084x1000 single
resnet_features_tr...	3060x1000 single
test_accuracy	0.9321
test_labels	6084x1 double
testing	6084x5049 single
TestingAccuracy	1x6084 double
testingSet	1x1 ImageDatastore
TestingTime	1
timeElapsed	233.0252
train_accuracy	1
train_labels	3060x1 double
train_time	4
training	3060x5049 single
TrainingAccuracy	0.9157
trainingSet	1x1 ImageDatastore
TrainingTime	44.8906
w	3060x3048 single
x	3060x2000 single
y	6084x2000 single
z	6084x3048 single

1.Run (Testing Acc: - 91.2420%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>nvcc Caltech101ELM.cu
Caltech101ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech101ELM.lib and object Caltech101ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>Caltech101ELM.exe
Training Accuracy is : 0.996847
Testing Accuracy is : 0.912420

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>
```

2.Run (Testing Acc: - 93.0309%)

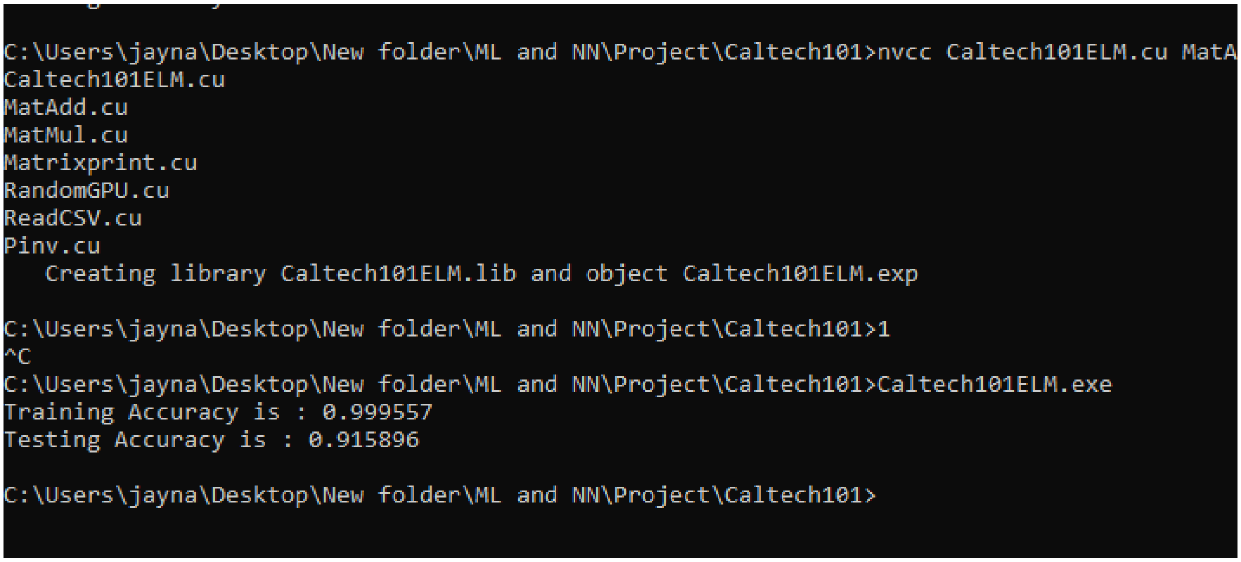


```
tic
% define  folder
Folder = fullfile('C:\Users\Neel\Desktop\DLProject\data', 'caltech101');
disp('steps to output')

% Create ImageDatastore of the dataset for processing in Matlab.
rootFolder = fullfile(Folder, '101_ObjectCategories');
imds = imageDatastore(fullfile(rootFolder), 'LabelSource', 'foldernames','IncludeSubfolders',true);
clear Folder rootFolder;

% Split each label Using 30 images for training and rest for testing
[trainingSet, testingSet] = splitEachLabel(imds, 30);

% Load resnet
disp('1. Loading Pretrained Resnet');
 net = resnet101;

 %set imageSize according to DCNN first input layer
 imageSize = net.Layers(1).InputSize;

 disp('2.Data Augmentatiojn');
%Define Augmentation function arguments
imageAugmenter = imageDataAugmenter('RandXReflection',true,...
    'RandYReflection',false,...
    'RandRotation',[0 0],...
    'RandScale',[1 1],...
    'RandXTranslation',[0 0],...
    'RandYTranslation',[0 0]);

%Store Augmented training image into Imagedatastore
```

```
          DeviceSupported: 1
           DeviceSelected: 1

12. Loading densenet train features
13. Loading densenet test features
14. Combining the training features from All DCNN
15. Combining the testing features from All DCNN
16. creating training and testing dataset for elm
18. Classification using MFB_ELM
Training Time = 4.062500
Training Accuracy = 1.000000
Testing Accuracy = 0.930309
Total Time Elapsed = 217.334810
>>
```

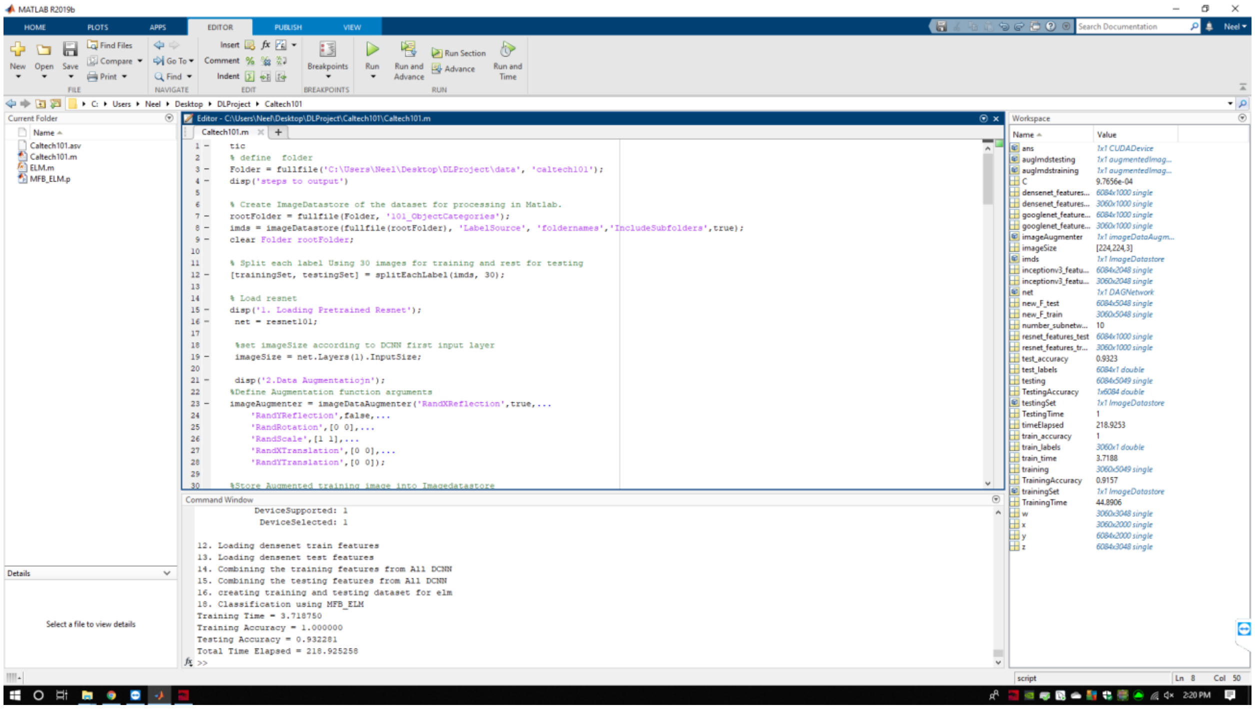
Name	Value
ans	1x1 CUDADevice
auglmdstesting	1x1 augmentedImag...
auglmdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	6084x1000 single
densenet_features...	3060x1000 single
googlenet_feature...	6084x1000 single
googlenet_feature...	3060x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	6084x2048 single
inceptionv3_featu...	3060x2048 single
net	1x1 DAGNetwork
new_F_test	6084x5048 single
new_F_train	3060x5048 single
number_subnetw...	10
resnet_features_test	6084x1000 single
resnet_features_tr...	3060x1000 single
test_accuracy	0.9303
test_labels	6084x1 double
testing	6084x5049 single
TestingAccuracy	1x6084 double
testingSet	1x1 ImageDatastore
TestingTime	1
timeElapsed	217.3348
train_accuracy	1
train_labels	3060x1 double
train_time	4.0625
training	3060x5049 single
TrainingAccuracy	0.9157
trainingSet	1x1 ImageDatastore
TrainingTime	44.8906
w	3060x3048 single
x	3060x2000 single
y	6084x2000 single
z	6084x3048 single

2.Run (Testing Acc: - 91.5896%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>nvcc Caltech101ELM.cu MatA
Caltech101ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech101ELM.lib and object Caltech101ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>1
^C
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>Caltech101ELM.exe
Training Accuracy is : 0.999557
Testing Accuracy is : 0.915896

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>
```

3.Run (Testing Acc: - 93.2281%)

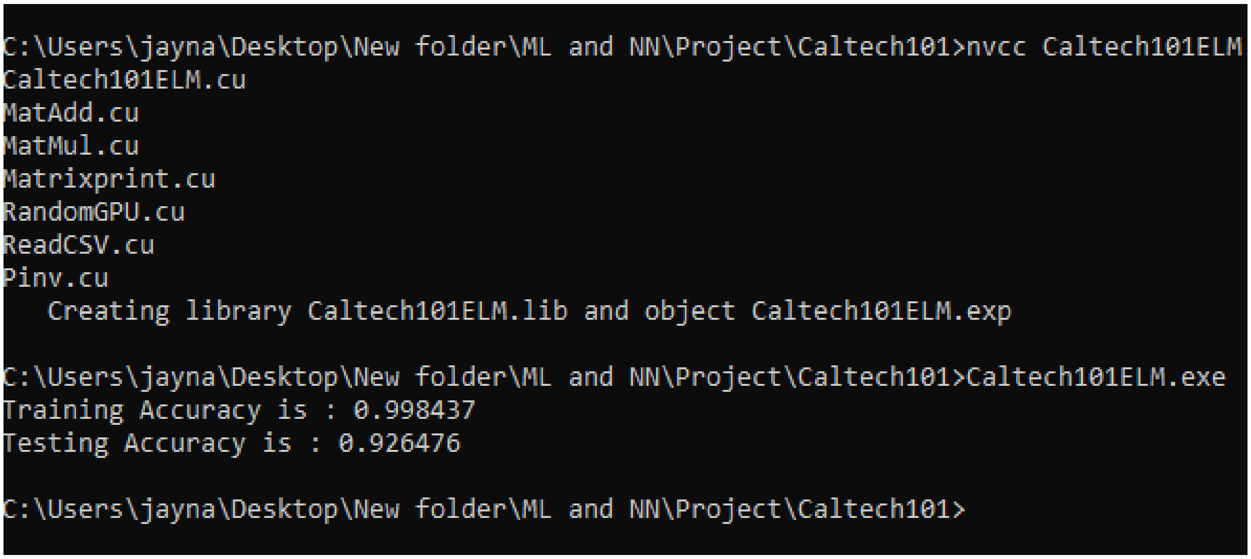


```
tic
% define  folder
Folder = fullfile('C:\Users\Neel\Desktop\DLProject\data', 'caltech101');
disp('steps to output')

% Create ImageDatastore of the dataset for processing in Matlab.
rootFolder = fullfile(Folder, '101_ObjectCategories');
imds = imageDatastore(fullfile(rootFolder), 'LabelSource', 'foldernames','IncludeSubfolders',true);
clear Folder rootFolder;

% Split each label Using 30 images for training and rest for testing
[trainingSet, testingSet] = splitEachLabel(imds, 30);

% Load resnet
disp('1. Loading Pretrained Resnet');
 net = resnet101;

 %set imageSize according to DCNN first input layer
 imageSize = net.Layers(1).InputSize;

 disp('2.Data Augmentatiojn');
%Define Augmentation function arguments
imageAugmenter = imageDataAugmenter('RandXReflection',true,...
    'RandYReflection',false,...
    'RandRotation',[0 0],...
    'RandScale',[1 1],...
    'RandXTranslation',[0 0],...
    'RandYTranslation',[0 0]);

%Store Augmented training image into Imagedatastore
```

```
           DeviceSupported: 1
            DeviceSelected: 1

12. Loading densenet train features
13. Loading densenet test features
14. Combining the training features from All DCNN
15. Combining the testing features from All DCNN
16. creating training and testing dataset for elm
18. Classification using MFB_ELM
Training Time = 3.718750
Training Accuracy = 1.000000
Testing Accuracy = 0.932281
Total Time Elapsed = 218.925258
>>
```

Name	Value
ans	1x1 CUDADevice
augImdstesting	1x1 augmentedImag...
augImdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	6084x1000 single
densenet_features...	3060x1000 single
googlenet_feature...	6084x1000 single
googlenet_feature...	3060x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	6084x2048 single
inceptionv3_featu...	3060x2048 single
net	1x1 DAGNetwork
new_F_test	6084x5048 single
new_F_train	3060x5048 single
number_subnetw...	10
resnet_features_test	6084x1000 single
resnet_features_tr...	3060x1000 single
test_accuracy	0.9323
test_labels	6084x1 double
testing	6084x5049 single
TestingAccuracy	1x6084 double
testingSet	1x1 ImageDatastore
TestingTime	1
timeElapsed	218.9253
train_accuracy	1
train_labels	3060x1 double
train_time	3.7188
training	3060x5049 single
TrainingAccuracy	0.9157
trainingSet	1x1 ImageDatastore
TrainingTime	44.8906
w	3060x3048 single
x	3060x2000 single
y	6084x2000 single
z	6084x3048 single

3.Run (Testing Acc: - 92.6476%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>nvcc Caltech101ELM
Caltech101ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech101ELM.lib and object Caltech101ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>Caltech101ELM.exe
Training Accuracy is : 0.998437
Testing Accuracy is : 0.926476

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech101>
```

2. Caltech256

Code:

```
tic
% define  folder
Folder = fullfile('C:\Users\Neel\Desktop\DLProject\data', 'caltech256');
disp('steps to output')

% Create ImageDatastore of the dataset for processing in Matlab.
rootFolder = fullfile(Folder, '256_ObjectCategories');
imds = imageDatastore(fullfile(rootFolder), 'LabelSource',
'foldernames','IncludeSubfolders',true);
clear Folder rootFolder;

% Split each label Using 30 images for training and rest for testing
[trainingSet, testingSet] = splitEachLabel(imds, 30);

% Load resnet
disp('1. Loading Pretrained Resnet');
 net = resnet101;

 %set imageSize according to DCNN first input layer
 imageSize = net.Layers(1).InputSize;


%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);
%resetgpu
gpuDevice(1)
```

```
% Get features from resnet
 disp('3. Loading Resnet train features');
 %Extract training features from resnet101 DCNN
 resnet_features_train =
activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

 disp('4. Loading Resnet test features');
 %Extract testing features from resnet101 DCNN
 resnet_features_test =
activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

 %Reshape training and testing features from resnet101
 resnet_features_train =
reshape(resnet_features_train,[1*1*1000,size(resnet_features_train,4)])' ;
 resnet_features_test =
reshape(resnet_features_test,[1*1*1000,size(resnet_features_test,4)])';

 disp('5. Loading Pretrained inceptionv3');
% Load inceptionv3
net = inceptionv3;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb' ,'DataAugmentation',imageAugmenter);

%resetgpu
gpuDevice(1)

disp('6. Loading inceptionv3 train features');
 %Extract training features from inceptionv3 DCNN
inceptionv3_features_train =
activations(net,augImdstraining,'avg_pool','MiniBatchSize',200);

disp('7. Loading inceptionv3 test features');
 %Extract testing features from inceptionv3 DCNN
inceptionv3_features_test =
activations(net,augImdstesting,'avg_pool','MiniBatchSize',200);

%Reshape training and testing features from inceptionv3
inceptionv3_features_train =
reshape(inceptionv3_features_train,[1*1*2048,size(inceptionv3_features_train,
4)])' ;
inceptionv3_features_test =
reshape(inceptionv3_features_test,[1*1*2048,size(inceptionv3_features_test,4)
])';

disp('8. Loading Pretrained googlenet');
```

```
% Load inceptionv3
net = googlenet;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);
%resetgpu
gpuDevice(1)

 disp('9. Loading googlenet train features');
 %Extract training features from googlenet DCNN
  googlenet_features_train = activations(net,augImdstraining,'loss3-
classifier','MiniBatchSize',200);

  disp('10. Loading googlenet test features');
  %Extract testing features from googlenet DCNN
googlenet_features_test = activations(net,augImdstesting,'loss3-
classifier','MiniBatchSize',200);

%Reshape training and testing features from googlenet
googlenet_features_train =
reshape(googlenet_features_train,[1*1*1000,size(googlenet_features_train,4)])
' ;
googlenet_features_test =
reshape(googlenet_features_test,[1*1*1000,size(googlenet_features_test,4)])';

disp('11. Loading Pretrained densenet201');
% Load densenet201
net = densenet201;

%set imageSize according to DCNN first input layer
imageSize = net.Layers(1).InputSize;

%Store Augmented training image into Imagedatastore
augImdstraining = augmentedImageDatastore(imageSize,trainingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
   'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%resetgpu
gpuDevice(1)

 disp('12. Loading densenet train features');
```

```
%Extract training features from densenet DCNN
  densenet_features_train =
activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

  disp('13. Loading densenet test features');
  %Extract testing features from densenet DCNN
densenet_features_test =
activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

%Reshape training and testing features from densenet
densenet_features_train =
reshape(densenet_features_train,[1*1*1000,size(densenet_features_train,4)])'
;
densenet_features_test =
reshape(densenet_features_test,[1*1*1000,size(densenet_features_test,4)])';


disp('14. Combining the training features from All DCNN');
% Merge Resnet and googlenet deep features for training
x = horzcat(resnet_features_train,googlenet_features_train);
% Merge densenet and inceptionv3 deep features for training
w = horzcat(inceptionv3_features_train, densenet_features_train);
% Merge all deep features for training
new_F_train = horzcat(x,w);

disp('15. Combining the testing features from All DCNN');
% Merge Resnet and googlenet deep features for testing
y = horzcat(resnet_features_test,googlenet_features_test);
% Merge inceptionv3 and densenet deep features for testing
z = horzcat(inceptionv3_features_test, densenet_features_test);
% Merge all deep features for testing
new_F_test = horzcat(y,z);


%Get Train Label from training dataset
train_labels = grp2idx(trainingSet.Labels);

%Get Test Label from testing dataset
test_labels = grp2idx(testingSet.Labels);

disp('16. creating training and testing dataset for elm');
%Give labels to training and testing features
 training = horzcat(train_labels,new_F_train);
 testing = horzcat( test_labels,new_F_test);



C = 2^-10;
% disp('17. Classification using ELM');
% [TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy] =
ELM(training, testing, 1, 10000, 'sig', C);

%define Number subnetworknode for MFB_ELM
number_subnetwork_node =10;
```

```
disp('18. Classification using MFB_ELM')
[train_time,
train_accuracy,test_accuracy]=MFB_ELM(training,testing,1,1,'sig',number_subne
twork_node,C);
fprintf('Training Time = %f\n',train_time);
fprintf('Training Accuracy = %f\n',train_accuracy);
fprintf('Testing Accuracy = %f\n',test_accuracy);



timeElapsed = toc;
```

Results:

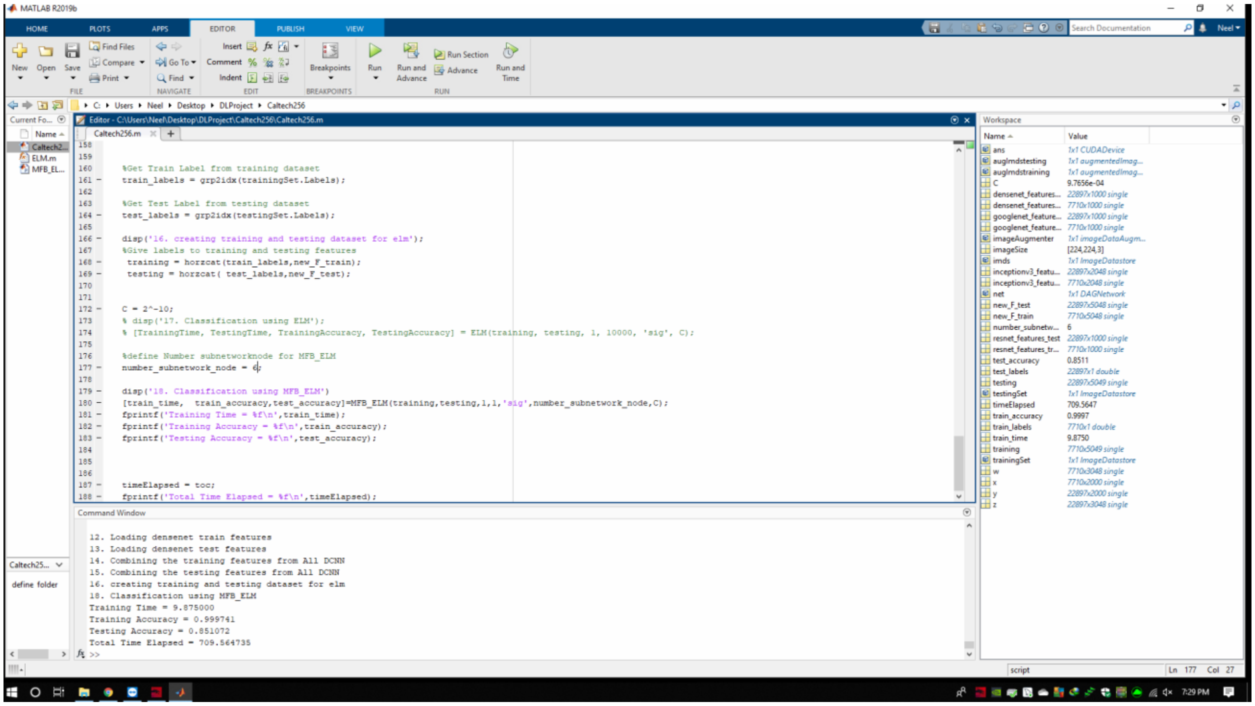
- **Top 1-Accuracy:** **85.14%**
- **Accuracy in each Trial:**

Classifier	Run 1	Run 2	Run 3	Avg
ELM(Cuda)	0.8394	0.8457	0.8346	0.8457
ELM(CPU)	0.8248	0.8496	0.8392	0.8496
MFB_ELM	0.8511	0.8514	0.8505	0.8514

- **GPU:** **NVIDIA QUADRO RTX 8000 48GB**
- **Total time:** **60.421minutes**

Outputs:

1.Run (Testing Acc: - 85.10%)



```
%Get Train Label from training dataset
train_labels = grp2idx(trainingSet.Labels);

%Get Test Label from testing dataset
test_labels = grp2idx(testingSet.Labels);

disp('16. creating training and testing dataset for elm');
%Give labels to training and testing features
 training = horzcat(train_labels,new_F_train);
 testing = horzcat( test_labels,new_F_test);


C = 2^-10;
% disp('17. Classification using ELM');
% [TrainingTime, TestingTime, TrainingAccuracy, TestingAccuracy] = ELM(training, testing, 1, 10000, 'sig', C);

%define Number subnetworknode for MFB_ELM
number_subnetwork_node = 6;

disp('18. Classification using MFB_ELM')
[train_time,  train_accuracy,test_accuracy]=MFB_ELM(training,testing,1,1,'sig',number_subnetwork_node,C);
fprintf('Training Time = %f\n',train_time);
fprintf('Training Accuracy = %f\n',train_accuracy);
fprintf('Testing Accuracy = %f\n',test_accuracy);



timeElapsed = toc;
fprintf('Total Time Elapsed = %f\n',timeElapsed);
```

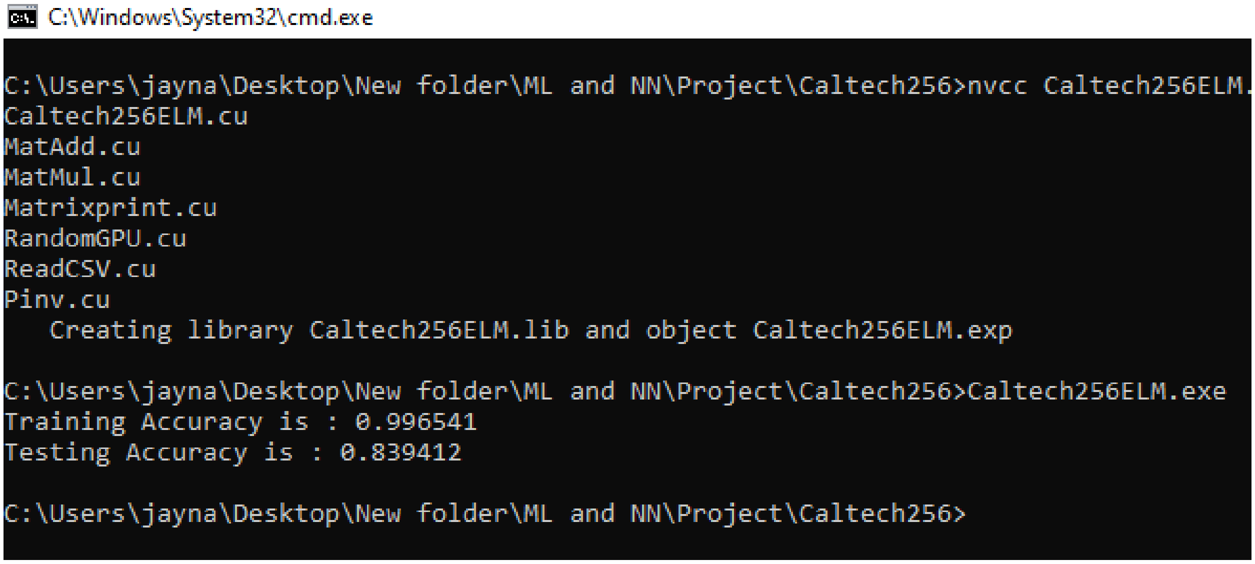
Command Window

```
12. Loading densenet train features
13. Loading densenet test features
14. Combining the training features from All DCNN
15. Combining the testing features from All DCNN
16. creating training and testing dataset for elm
18. Classification using MFB_ELM
Training Time = 9.875000
Training Accuracy = 0.999741
Testing Accuracy = 0.851072
Total Time Elapsed = 709.564735
fx >>
```

Workspace

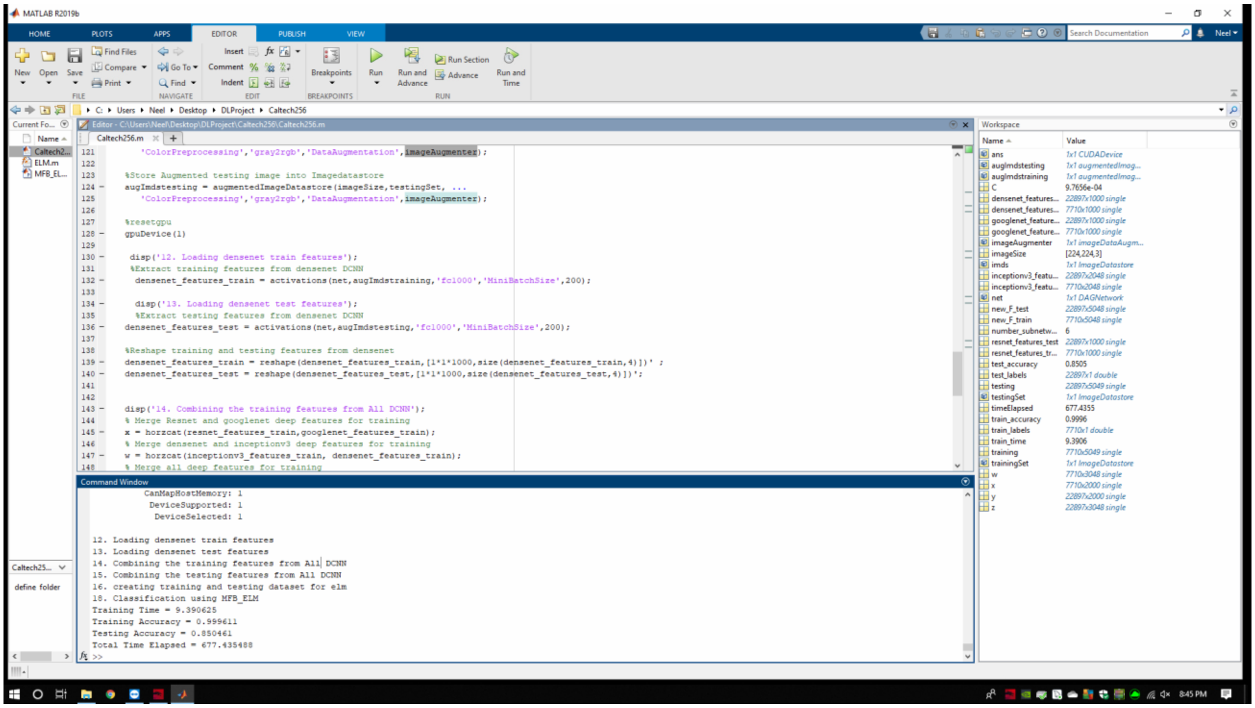
Name ▲	Value
ans	1x1 CUDADevice
auglmdstesting	1x1 augmentedImag...
auglmdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	22897x1000 single
densenet_features...	7710x1000 single
googlenet_feature...	22897x1000 single
googlenet_feature...	7710x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	22897x2048 single
inceptionv3_featu...	7710x2048 single
net	1x1 DAGNetwork
new_F_test	22897x5048 single
new_F_train	7710x5048 single
number_subnetw...	6
resnet_features_test	22897x1000 single
resnet_features_tr...	7710x1000 single
test_accuracy	0.8511
test_labels	22897x1 double
testing	22897x5049 single
testingSet	1x1 ImageDatastore
timeElapsed	709.5647
train_accuracy	0.9997
train_labels	7710x1 double
train_time	9.8750
training	7710x5049 single
trainingSet	1x1 ImageDatastore
w	7710x3048 single
x	7710x2000 single
y	22897x2000 single
z	22897x3048 single

1.Run (Testing Acc: - 83.94%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>nvcc Caltech256ELM.
Caltech256ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech256ELM.lib and object Caltech256ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>Caltech256ELM.exe
Training Accuracy is : 0.996541
Testing Accuracy is : 0.839412

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>
```

2.Run (Testing Acc: - 85.04%)

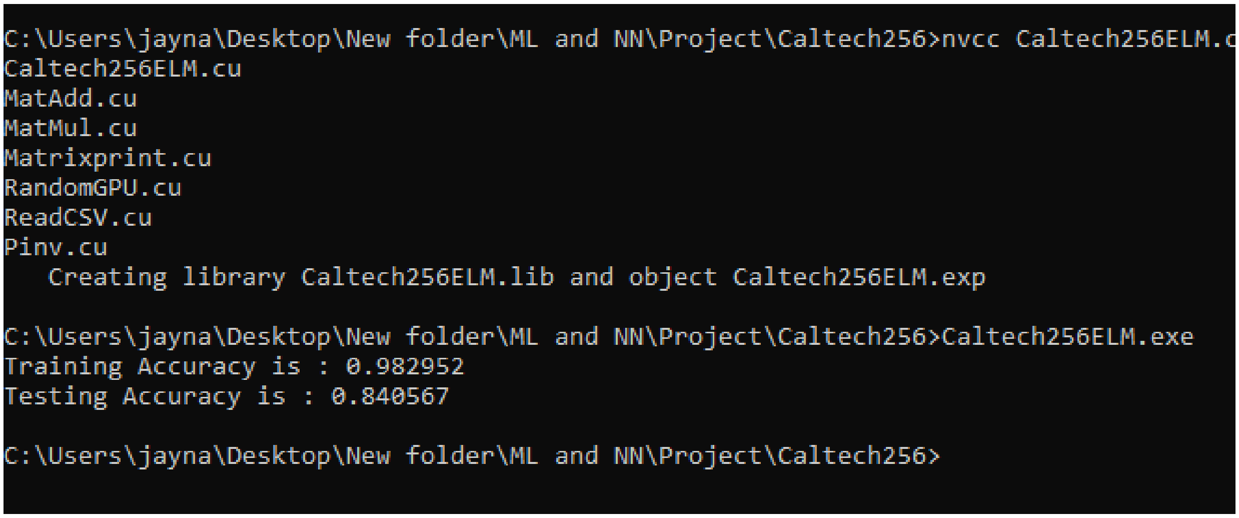


```
    'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%Store Augmented testing image into Imagedatastore
augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
    'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

%resetgpu
gpuDevice(1)

 disp('12. Loading densenet train features');
 %Extract training features from densenet DCNN
  densenet_features_train = activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

  disp('13. Loading densenet test features');
  %Extract testing features from densenet DCNN
densenet_features_test = activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

%Reshape training and testing features from densenet
densenet_features_train = reshape(densenet_features_train,[1*1*1000,size(densenet_features_train,4)])' ;
densenet_features_test = reshape(densenet_features_test,[1*1*1000,size(densenet_features_test,4)])';


disp('14. Combining the training features from All DCNN');
% Merge Resnet and googlenet deep features for training
x = horzcat(resnet_features_train,googlenet_features_train);
% Merge densenet and inceptionv3 deep features for training
w = horzcat(inceptionv3_features_train, densenet_features_train);
% Merge all deep features for training
```

```
          CanMapHostMemory: 1
           DeviceSupported: 1
            DeviceSelected: 1

12. Loading densenet train features
13. Loading densenet test features
14. Combining the training features from All DCNN
15. Combining the testing features from All DCNN
16. creating training and testing dataset for elm
18. Classification using MFB_ELM
Training Time = 9.390625
Training Accuracy = 0.999611
Testing Accuracy = 0.850461
Total Time Elapsed = 677.435488
>>
```

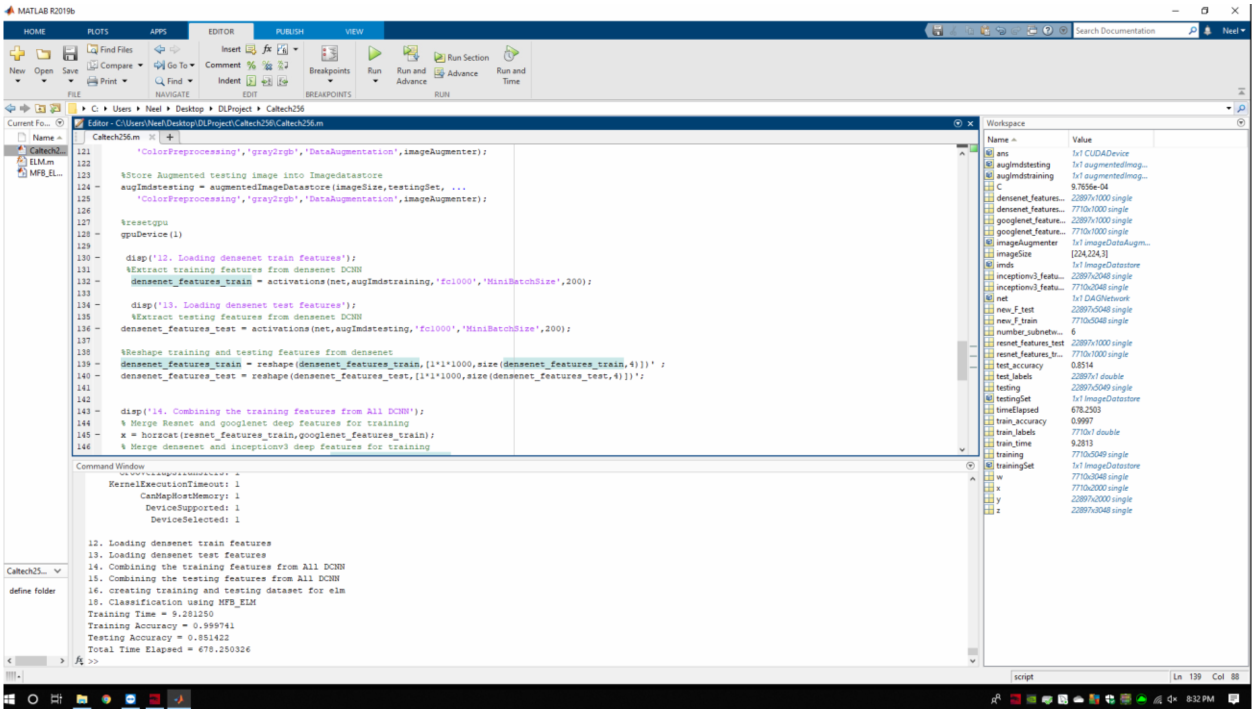
Name	Value
ans	1x1 CUDADevice
augImdstesting	1x1 augmentedImag...
augImdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	22897x1000 single
densenet_features...	7710x1000 single
googlenet_feature...	22897x1000 single
googlenet_feature...	7710x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	22897x2048 single
inceptionv3_featu...	7710x2048 single
net	1x1 DAGNetwork
new_F_test	22897x5048 single
new_F_train	7710x5048 single
number_subnetw...	6
resnet_features_test	22897x1000 single
resnet_features_tr...	7710x1000 single
test_accuracy	0.8505
test_labels	22897x1 double
testing	22897x5049 single
testingSet	1x1 ImageDatastore
timeElapsed	677.4355
train_accuracy	0.9996
train_labels	7710x1 double
train_time	9.3906
training	7710x5049 single
trainingSet	1x1 ImageDatastore
w	7710x3048 single
x	7710x2000 single
y	22897x2000 single
z	22897x3048 single

2.Run (Testing Acc: - 84.06%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>nvcc Caltech256ELM.c
Caltech256ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech256ELM.lib and object Caltech256ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>Caltech256ELM.exe
Training Accuracy is : 0.982952
Testing Accuracy is : 0.840567

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>
```

3.Run (Testing Acc: - 85.14%)



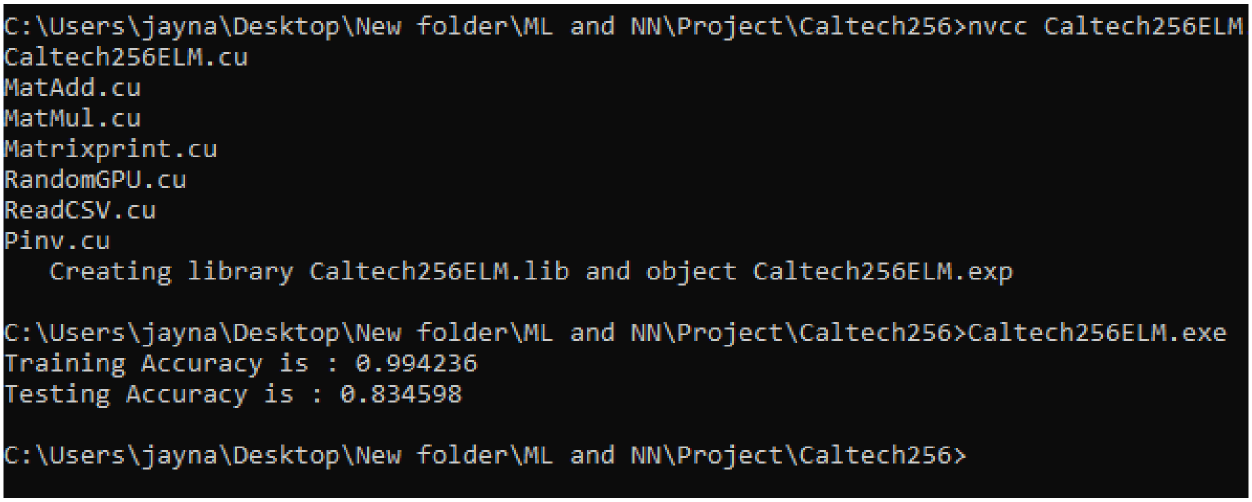
```
      'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

   %Store Augmented testing image into Imagedatastore
   augImdstesting = augmentedImageDatastore(imageSize,testingSet, ...
      'ColorPreprocessing','gray2rgb','DataAugmentation',imageAugmenter);

   %resetgpu
   gpuDevice(1)

    disp('12. Loading densenet train features');
    %Extract training features from densenet DCNN
     densenet_features_train = activations(net,augImdstraining,'fc1000','MiniBatchSize',200);

     disp('13. Loading densenet test features');
     %Extract testing features from densenet DCNN
   densenet_features_test = activations(net,augImdstesting,'fc1000','MiniBatchSize',200);

   %Reshape training and testing features from densenet
   densenet_features_train = reshape(densenet_features_train,[1*1*1000,size(densenet_features_train,4)])' ;
   densenet_features_test = reshape(densenet_features_test,[1*1*1000,size(densenet_features_test,4)])';


   disp('14. Combining the training features from All DCNN');
   % Merge Resnet and googlenet deep features for training
   x = horzcat(resnet_features_train,googlenet_features_train);
   % Merge densenet and inceptionv3 deep features for training
```

```
    KernelExecutionTimeout: 1
          CanMapHostMemory: 1
           DeviceSupported: 1
            DeviceSelected: 1

12. Loading densenet train features
13. Loading densenet test features
14. Combining the training features from All DCNN
15. Combining the testing features from All DCNN
16. creating training and testing dataset for elm
18. Classification using MFB_ELM
Training Time = 9.281250
Training Accuracy = 0.999741
Testing Accuracy = 0.851422
Total Time Elapsed = 678.250326
>>
```

Workspace

Name	Value
ans	1x1 CUDADevice
augImdstesting	1x1 augmentedImag...
augImdstraining	1x1 augmentedImag...
C	9.7656e-04
densenet_features...	22897x1000 single
densenet_features...	7710x1000 single
googlenet_feature...	22897x1000 single
googlenet_feature...	7710x1000 single
imageAugmenter	1x1 imageDataAugm...
imageSize	[224,224,3]
imds	1x1 ImageDatastore
inceptionv3_featu...	22897x2048 single
inceptionv3_featu...	7710x2048 single
net	1x1 DAGNetwork
new_F_test	22897x5048 single
new_F_train	7710x5048 single
number_subnetw...	6
resnet_features_test	22897x1000 single
resnet_features_tr...	7710x1000 single
test_accuracy	0.8514
test_labels	22897x1 double
testing	22897x5049 single
testingSet	1x1 ImageDatastore
timeElapsed	678.2503
train_accuracy	0.9997
train_labels	7710x1 double
train_time	9.2813
training	7710x5049 single
trainingSet	1x1 ImageDatastore
w	7710x3048 single
x	7710x2000 single
y	22897x2000 single
z	22897x3048 single

3.Run (Testing Acc: - 83.46%)



```
C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>nvcc Caltech256ELM
Caltech256ELM.cu
MatAdd.cu
MatMul.cu
Matrixprint.cu
RandomGPU.cu
ReadCSV.cu
Pinv.cu
   Creating library Caltech256ELM.lib and object Caltech256ELM.exp

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>Caltech256ELM.exe
Training Accuracy is : 0.994236
Testing Accuracy is : 0.834598

C:\Users\jayna\Desktop\New folder\ML and NN\Project\Caltech256>
```

Average Accuracy:

- **Caltech101+Caltech256 /2 = (93.23+85.14)/2 ≈ <u>90</u>**

Reference:

- **Yimin Yang, Q.M.Jonathan Wu. Autoencoder with invertible functions for dimension reduction and image reconstruction. IEEE Transactions on Neural Networks and Learning Systems. 2018.- MFB_ELM**
- http://www.yiminyang.com/code.html
- **All Together Now! The Benefits of Adaptively Fusing Pre-trained Deep Representations. Yehezkel S. Resheff1, Itay Lieder2 and Tom Hope2.**