Assignment 3
Computer Vision

Name: Muhammad Mubeen

Class: BEE-6D

Reg No.: 33302

Submitted To: Dr. Wajahat Hussain

Date: 18th December, 2017

Question 1

To load AlexNet in Matlab and classify an Image using it.

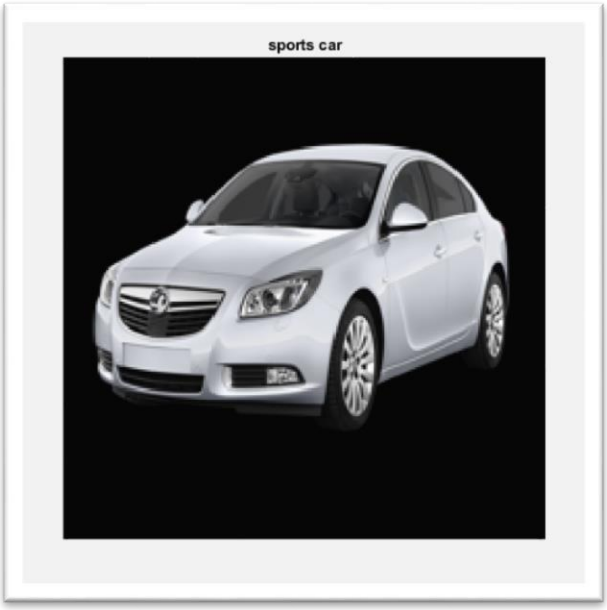
Code

The code is given below: -

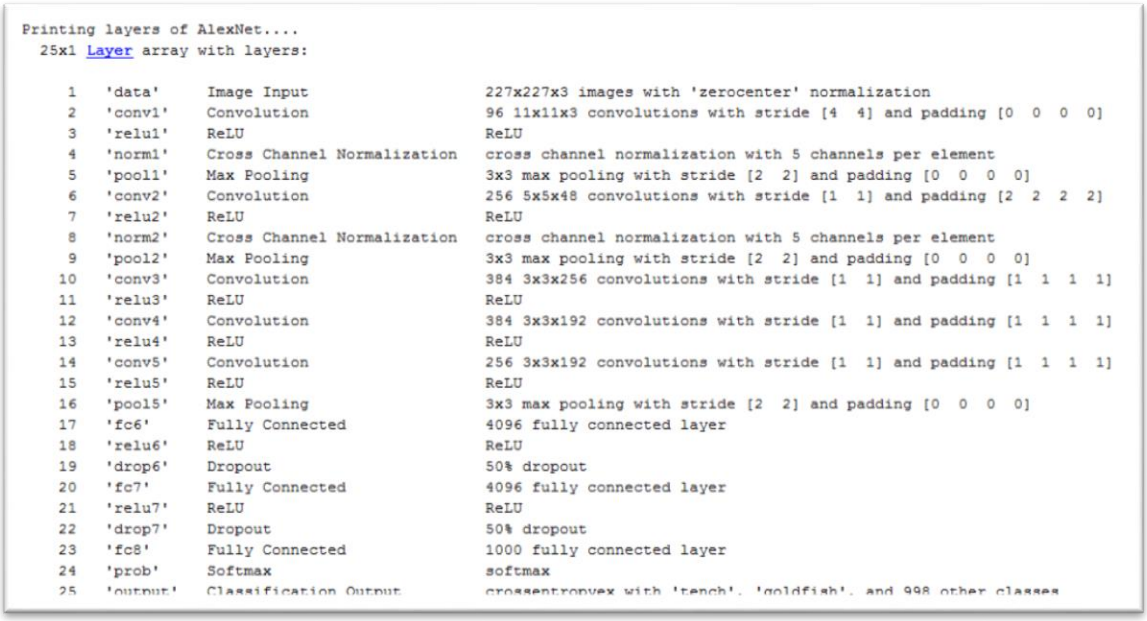
```
clc
clear all

disp('Loading AlexNet....')
net = alexnet;
disp('Network Loaded.')
disp(net);

disp('Printing layers of AlexNet....')
disp(net.Layers);

disp('Printing 10 classes of AlexNet.')
disp(net.Layers(end).ClassNames(100:110));

disp('Reading Image')
img = imread('car_image.png');
img = imresize(img, [227 227]);

disp('Classifying Image')
label = classify(net, img);

disp(strcat('The input Image belongs to the class: ', char(label)));

figure, imshow(img)
title(char(label))
```

Outputs

The predicted class along with image: -

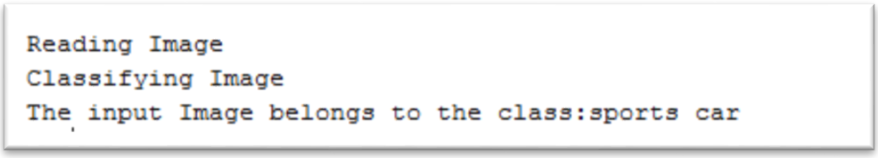


The layers of AlexNet are: -



```
Printing layers of AlexNet....
  25x1 Layer array with layers:

     1   'data'     Image Input                   227x227x3 images with 'zerocenter' normalization
     2   'conv1'    Convolution                   96 11x11x3 convolutions with stride [4  4] and padding [0  0  0  0]
     3   'relu1'    ReLU                          ReLU
     4   'norm1'    Cross Channel Normalization   cross channel normalization with 5 channels per element
     5   'pool1'    Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0  0  0]
     6   'conv2'    Convolution                   256 5x5x48 convolutions with stride [1  1] and padding [2  2  2  2]
     7   'relu2'    ReLU                          ReLU
     8   'norm2'    Cross Channel Normalization   cross channel normalization with 5 channels per element
     9   'pool2'    Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0  0  0]
    10   'conv3'    Convolution                   384 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    11   'relu3'    ReLU                          ReLU
    12   'conv4'    Convolution                   384 3x3x192 convolutions with stride [1  1] and padding [1  1  1  1]
    13   'relu4'    ReLU                          ReLU
    14   'conv5'    Convolution                   256 3x3x192 convolutions with stride [1  1] and padding [1  1  1  1]
    15   'relu5'    ReLU                          ReLU
    16   'pool5'    Max Pooling                   3x3 max pooling with stride [2  2] and padding [0  0  0  0]
    17   'fc6'      Fully Connected               4096 fully connected layer
    18   'relu6'    ReLU                          ReLU
    19   'drop6'    Dropout                       50% dropout
    20   'fc7'      Fully Connected               4096 fully connected layer
    21   'relu7'    ReLU                          ReLU
    22   'drop7'    Dropout                       50% dropout
    23   'fc8'      Fully Connected               1000 fully connected layer
    24   'prob'     Softmax                       softmax
    25   'output'   Classification Output         crossentropyex with 'tench', 'goldfish', and 998 other classes
```

The output of the Network: -



```
Reading Image
Classifying Image
The input Image belongs to the class:sports car
```

Question 2

To Use AlexNet for feature extraction

Code

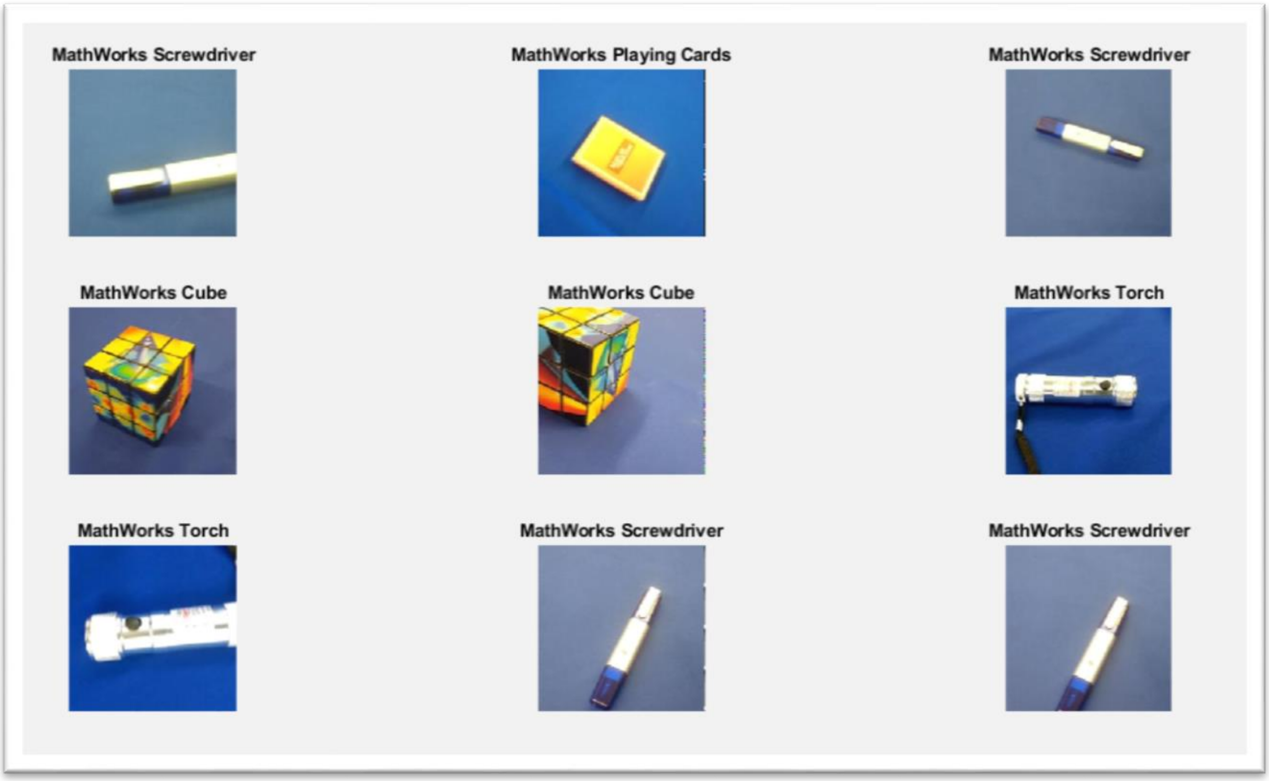
The code for this is given below: -

```
clc
clear all

% Loading a built in dataset
images = imageDatastore('MerchData', 'IncludeSubfolders',true,
'LabelSource','foldernames');

% Splitting the dataset
[trainingImages, testImages] = splitEachLabel(images,0.7,'randomized');

% Showing some of the images in figure
numTrainImages = numel(trainingImages.Labels);
idx = randperm(numTrainImages, 9);
figure
for i = 1:9
    subplot(3,3,i)
    I = readimage(trainingImages, idx(i));
    imshow(I)
    label = trainingImages.Labels(idx(i));
    title(char(label))
end

% Loading Network
net = alexnet;

% Extracting fc7 features for train and test data
layer = 'fc7';
trainingFeatures = activations(net, trainingImages, layer);
testFeatures = activations(net, testImages, layer);

% Getting train and test Labels
trainingLabels = trainingImages.Labels;
testLabels = testImages.Labels;

% Creating an SVM classifier for these features
classifier = fitcecoc(trainingFeatures, trainingLabels);

% Predicting Labels from the classifier of the test Images
predictedLabels = predict(classifier, testFeatures);

% Printing some of the output Images
numTestImages = numel(testImages.Labels);
```

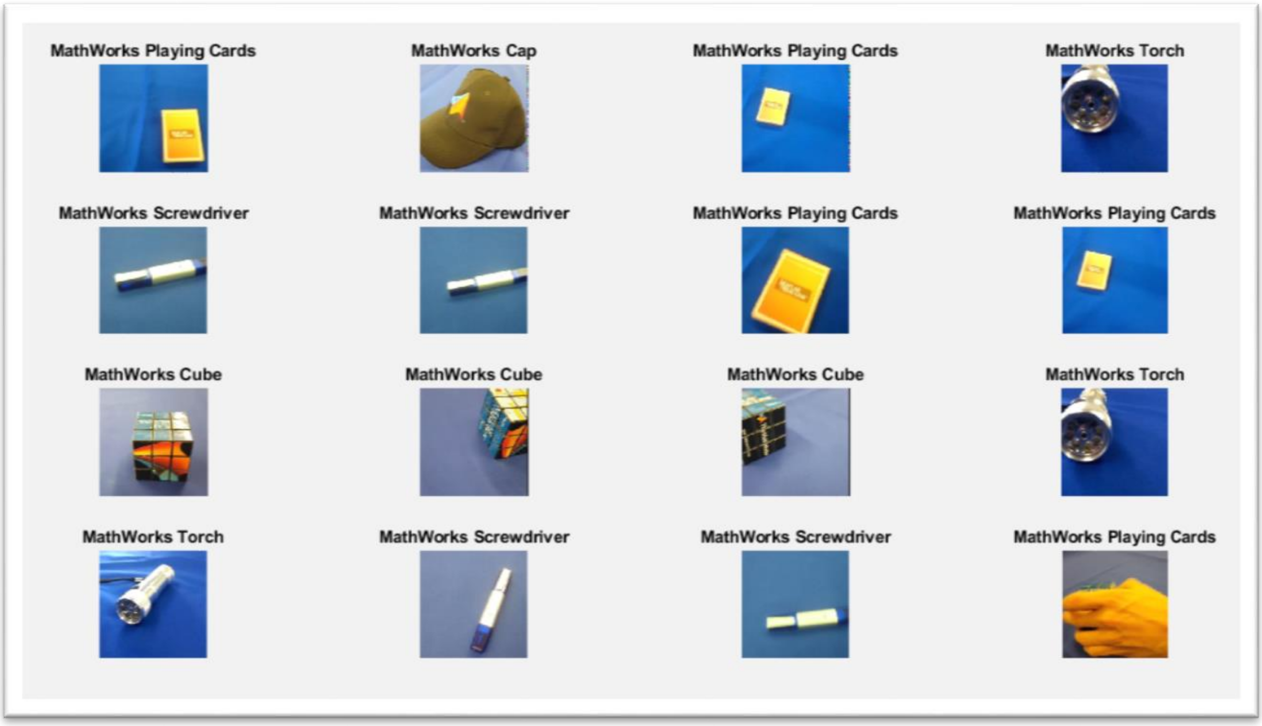
```
idx = randperm(numTestImages, 16);
figure
for i = 1:numel(idx)
    subplot(4,4,i)
    I = readimage(testImages, idx(i));
    label = predictedLabels(idx(i));
    imshow(I)
    title(char(label))
end

% Getting Accuracy of classifier
accuracy = mean(predictedLabels == testLabels);
disp(strcat('The test Accuracy of Classifier is:', num2str(accuracy*100),
'%'))
```

Outputs

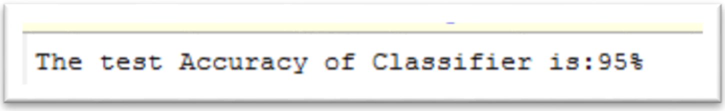
Some of the training images for feature extraction are: -



And the labels of some of the image predicted by classifier are: -



And the accuracy of the classifier: -



```
The test Accuracy of Classifier is:95%
```

Question 3

To finetune AlexNet

Code

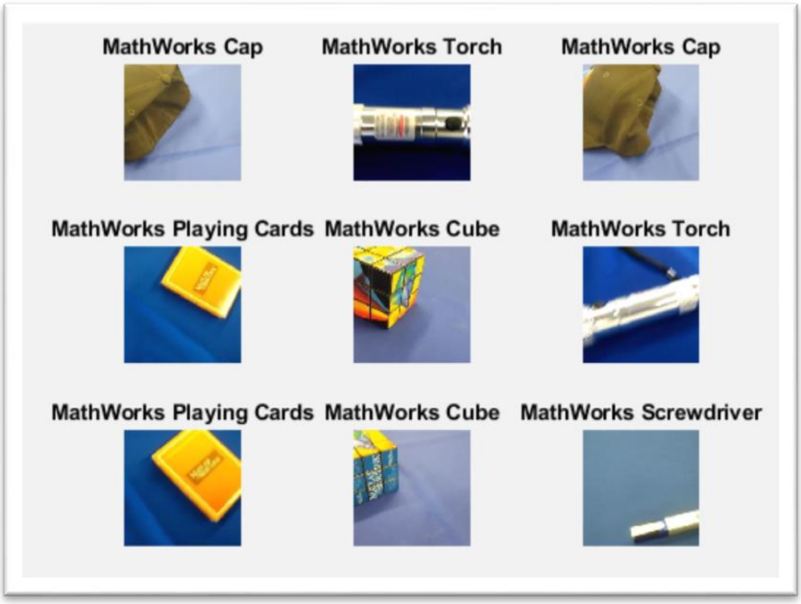
The code is given below: -

```
clc
clear all

% Loading a built in dataset
images = imageDatastore('MerchData', 'IncludeSubfolders',true,
'LabelSource','foldernames');

% Splitting the dataset
[trainingImages, testImages] = splitEachLabel(images,0.7,'randomized');

% Showing some of the images in figure
numTrainImages = numel(trainingImages.Labels);
idx = randperm(numTrainImages, 9);
figure
for i = 1:9
    subplot(3,3,i)
    I = readimage(trainingImages, idx(i));
    imshow(I)
    label = trainingImages.Labels(idx(i));
    title(char(label))
end

% Loading Network
net = alexnet;

% Loading all layers of AlexNet except the last 3 ones
layersTransfer = net.Layers(1:end-3);

% Adding new layer add end of extracted layer having numClasses
numClasses = numel(categories(trainingImages.Labels));
layers = [
    layersTransfer

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor
',20)
    softmaxLayer
    classificationLayer];
```

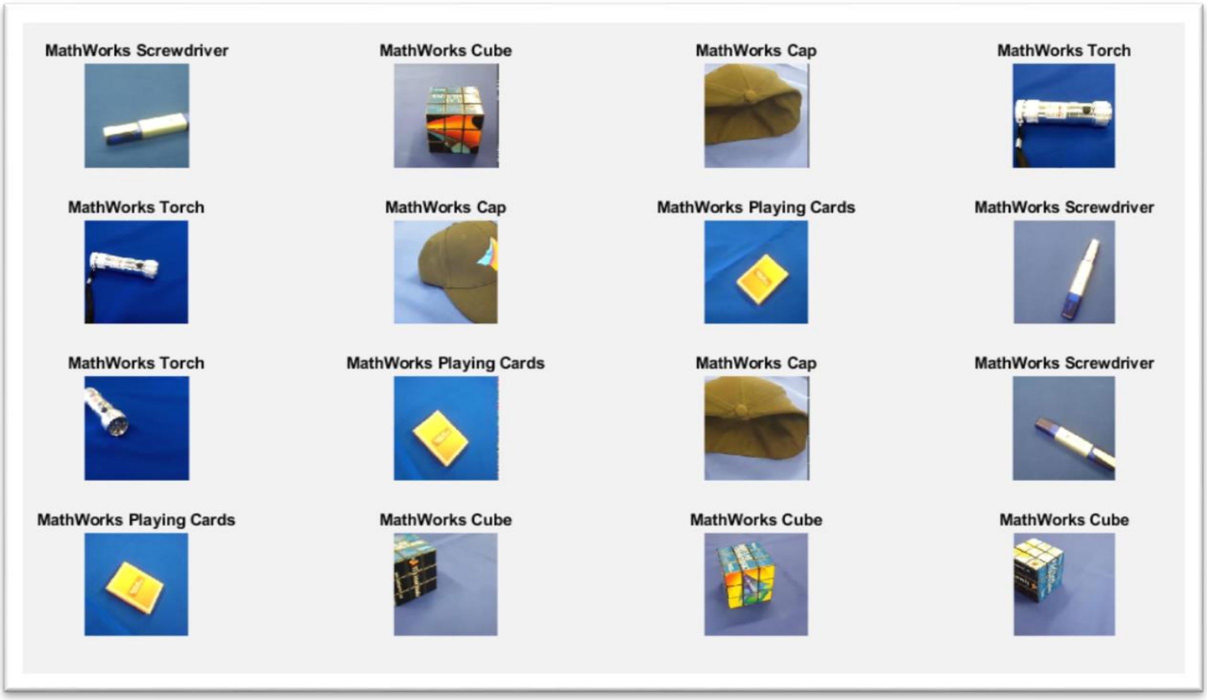
```
% Specifying training parameters
miniBatchSize = 10;
numIterationsPerEpoch = floor(numel(trainingImages.Labels)/miniBatchSize);
options = trainingOptions('sgdm',...
    'MiniBatchSize',miniBatchSize,...
    'MaxEpochs',4,...
    'InitialLearnRate',1e-4,...
    'Verbose',true,...
    'Plots','training-progress',...
    'ValidationData',testImages,...
    'ValidationFrequency',numIterationsPerEpoch);

% Training the network
netTransfer = trainNetwork(trainingImages,layers,options);

% Prediciting the output labels on test set
predictedLabels = classify(netTransfer,testImages);

% Printing some of the output Images
numTestImages = numel(testImages.Labels);
idx = randperm(numTestImages, 16);
figure
for i = 1:numel(idx)
    subplot(4,4,i)
    I = readimage(testImages, idx(i));
    label = predictedLabels(idx(i));
    imshow(I)
    title(char(label))
end

% Getting Accuracy of classifier
testLabels = testImages.Labels;
accuracy = mean(predictedLabels == testLabels);
disp(strcat('The test Accuracy of Classifier is:', num2str(accuracy*100),
'%'))
```

Outputs

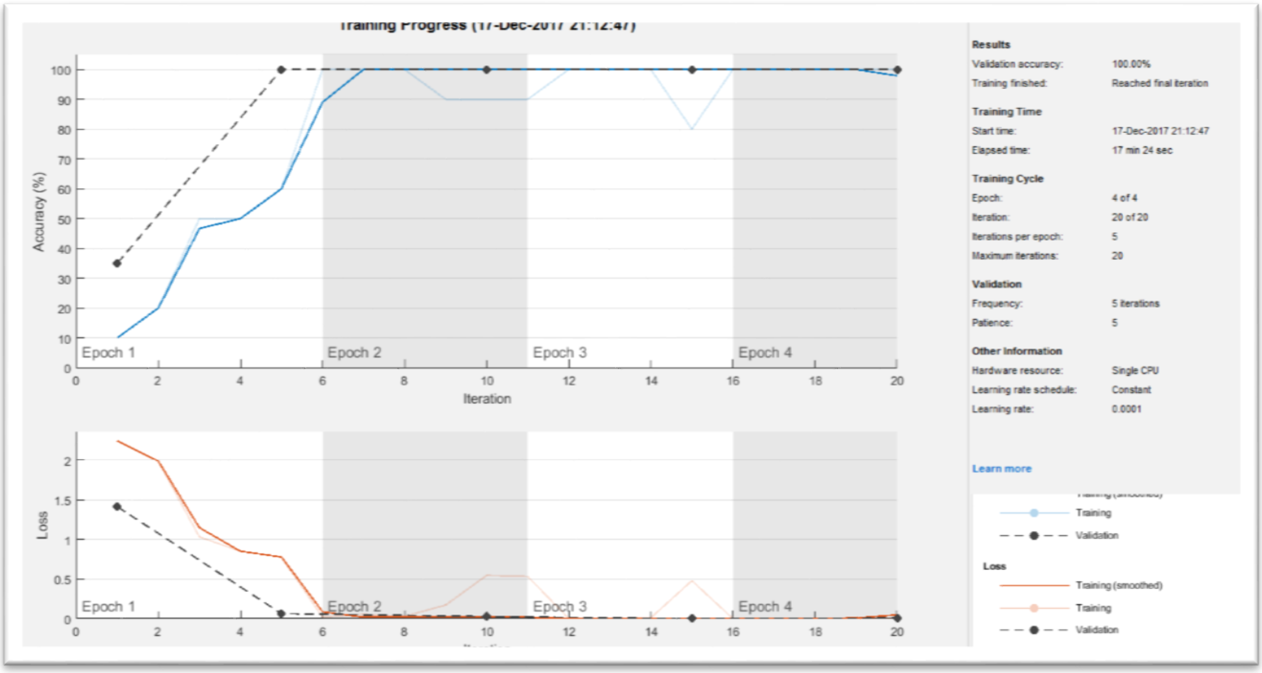
Some training images with labels include: -



Some of the predicted labels of classifier are: -



The training summary is: -



Results

Validation accuracy: 100.00%

Training finished: Reached final iteration

Training Time

Start time: 17-Dec-2017 21:12:47

Elapsed time: 17 min 24 sec

Training Cycle

Epoch: 4 of 4

Iteration: 20 of 20

Iterations per epoch: 5

Maximum iterations: 20

Validation

Frequency: 5 iterations

Patience: 5

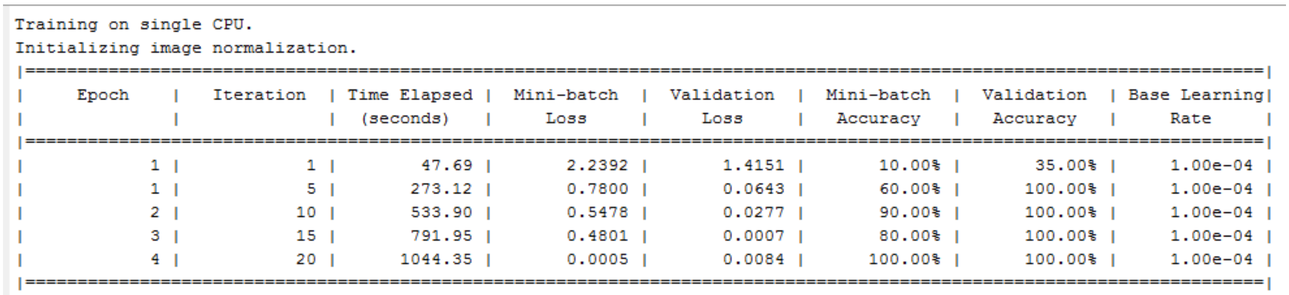
Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

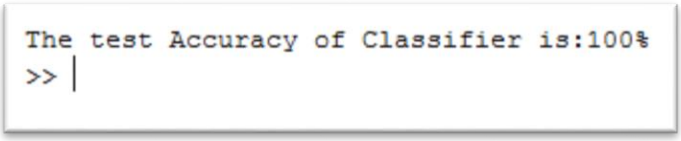
Learning rate: 0.0001

Learn more



```
Training on single CPU.
Initializing image normalization.
|=========================================================================================================================|
|     Epoch    |   Iteration  | Time Elapsed |  Mini-batch  |  Validation  |  Mini-batch  |  Validation  | Base Learning|
|              |              |  (seconds)   |     Loss     |     Loss     |   Accuracy   |   Accuracy   |     Rate     |
|=========================================================================================================================|
|            1 |            1 |        47.69 |       2.2392 |       1.4151 |       10.00% |       35.00% |     1.00e-04 |
|            1 |            5 |       273.12 |       0.7800 |       0.0643 |       60.00% |      100.00% |     1.00e-04 |
|            2 |           10 |       533.90 |       0.5478 |       0.0277 |       90.00% |      100.00% |     1.00e-04 |
|            3 |           15 |       791.95 |       0.4801 |       0.0007 |       80.00% |      100.00% |     1.00e-04 |
|            4 |           20 |      1044.35 |       0.0005 |       0.0084 |      100.00% |      100.00% |     1.00e-04 |
|=========================================================================================================================|
```

And the accuracy of classifier is: -



```
The test Accuracy of Classifier is:100%
>> |
```