圖形識別概論Project #3

0516218軒轅照雯

a.1D convolutional result of f and g

```
0.0100
0.0200
0.0300
0.0400
0.0300
0.0200
0.0100
```

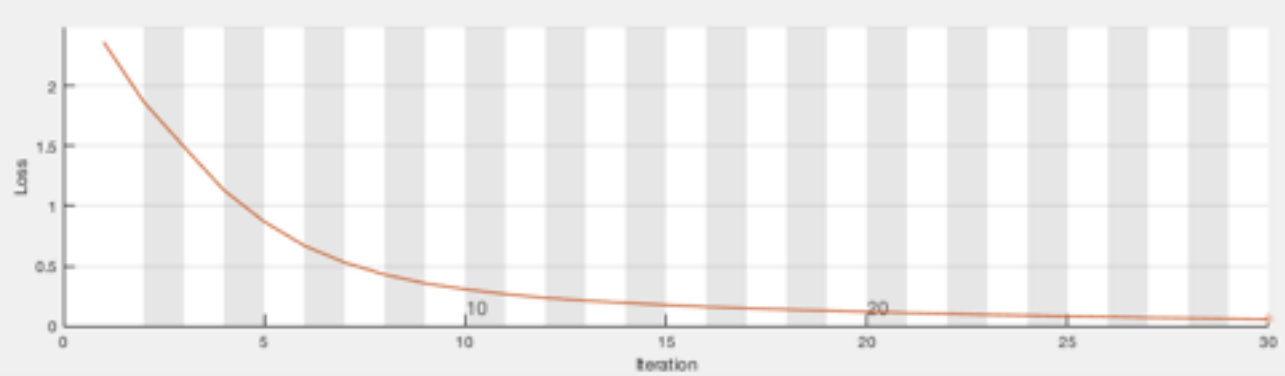
b.2D convolutional result of f and g

```
Columns 1 through 5

   0.0100    0.0200    0.0300    0.0400    0.0300
   0.0200    0.0400    0.0600    0.0800    0.0600
   0.0300    0.0600    0.0900    0.1200    0.0900
   0.0400    0.0800    0.1200    0.1600    0.1200
   0.0300    0.0600    0.0900    0.1200    0.0900
   0.0200    0.0400    0.0600    0.0800    0.0600
   0.0100    0.0200    0.0300    0.0400    0.0300

Columns 6 through 7

   0.0200    0.0100
   0.0400    0.0200
   0.0600    0.0300
   0.0800    0.0400
   0.0600    0.0300
   0.0400    0.0200
   0.0200    0.0100
```

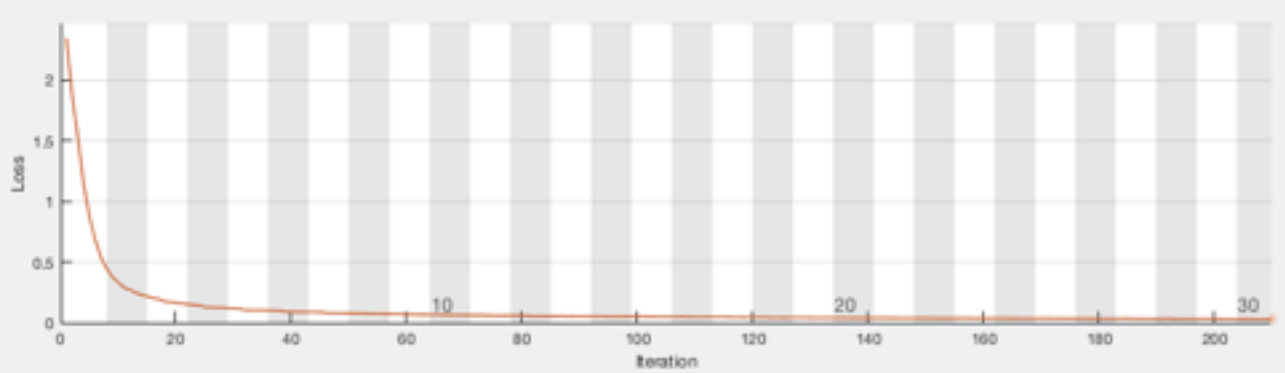
c.CNN

(1)average error vs iteration

1000 training patterns



60000 training patterns



(2)CPU time in learning

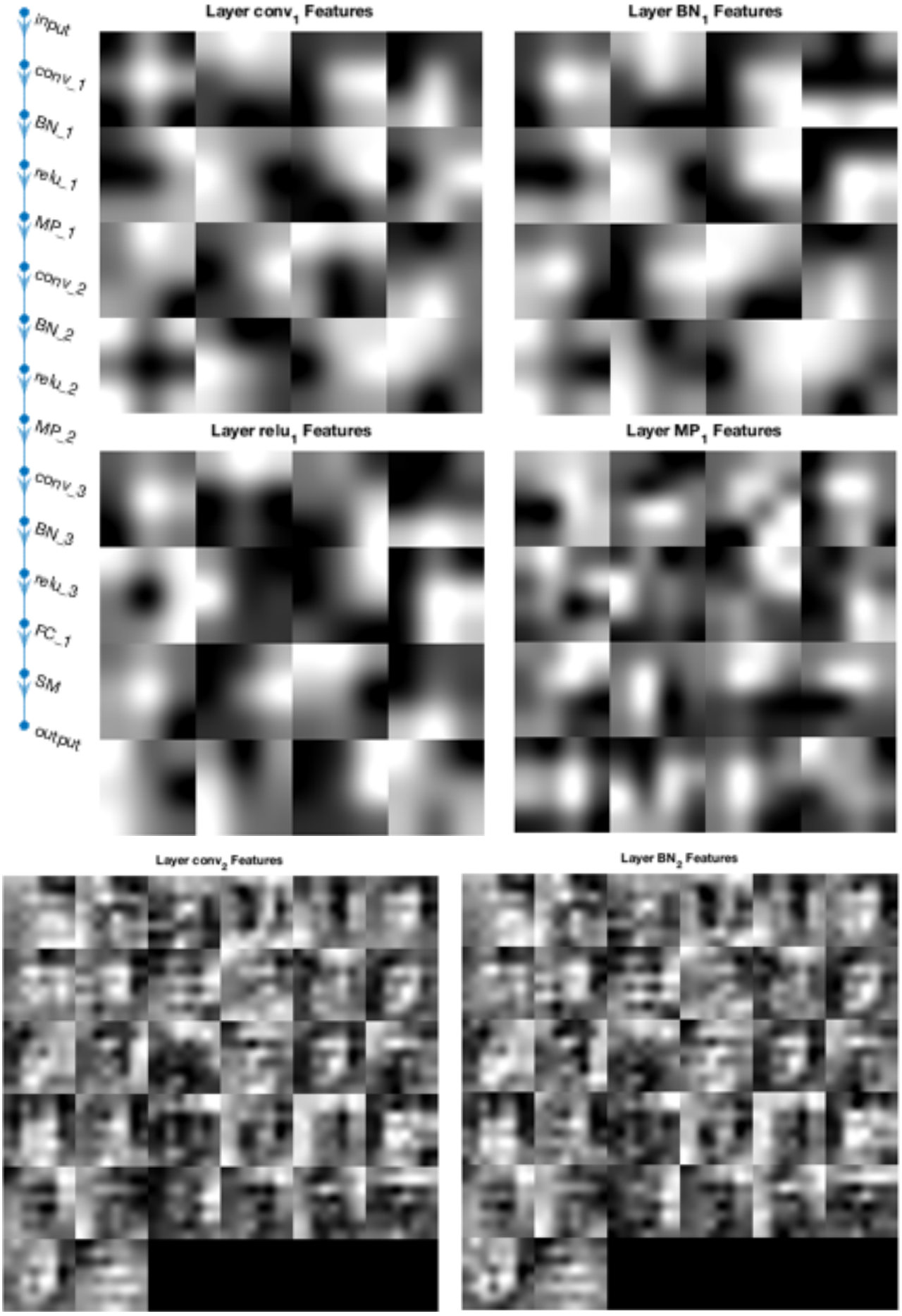
1000 training patterns

Elapsed time: 40 sec

60000 training patterns

Elapsed time: 46 min 59 sec

(3)各層抽取的特徵圖


input
conv_1
BN_1
relu_1
MP_1
conv_2
BN_2
relu_2
MP_2
conv_3
BN_3
relu_3
FC_1
SM
output
Layer conv1 Features
Layer BN1 Features
Layer relu1 Features
Layer MP1 Features
Layer conv2 Features
Layer BN2 Features

Layer $relu_2$ Features

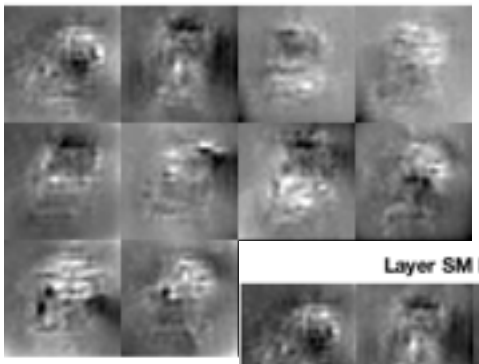
Layer MP_2 Features

Layer $conv_3$ Features

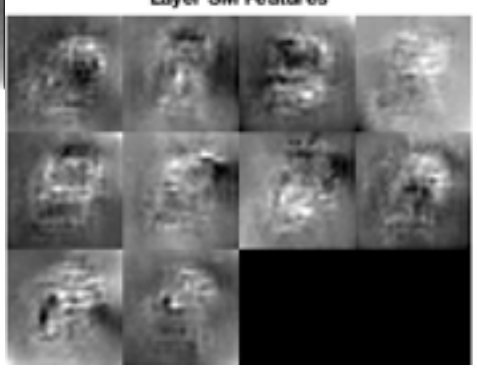
Layer BN_3 Features

Layer $relu_3$ Features

Layer FC_1 Features



Layer SM Features

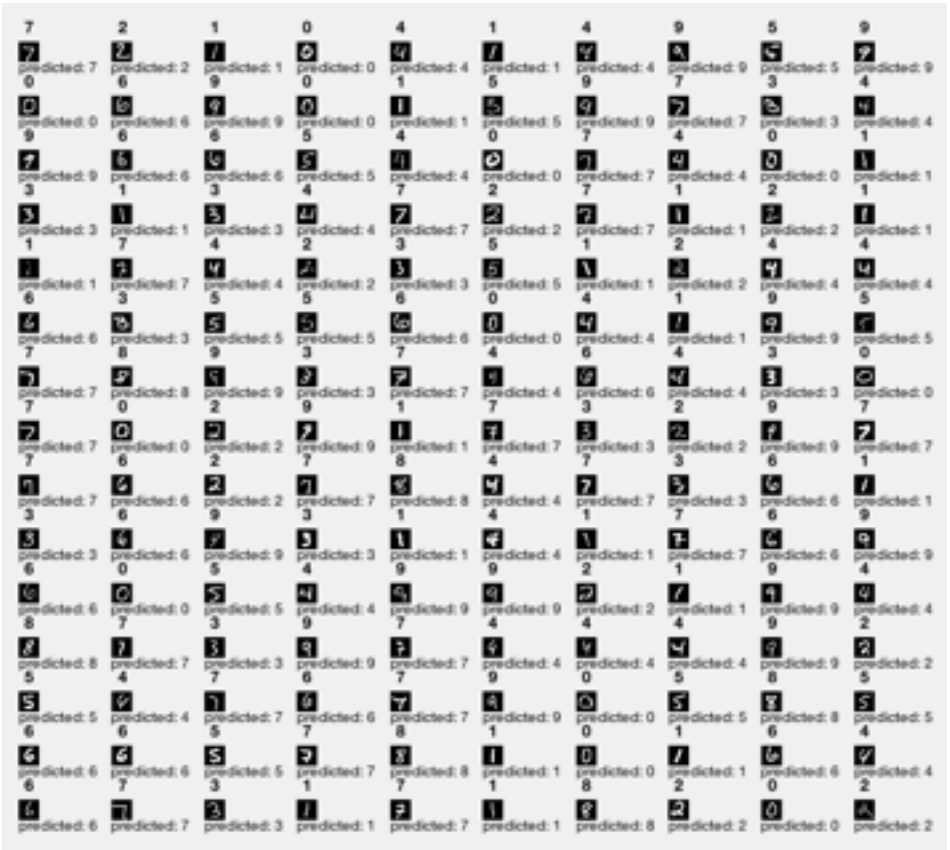


(4)150 testing patterns

1000 training patterns

wrong prediction

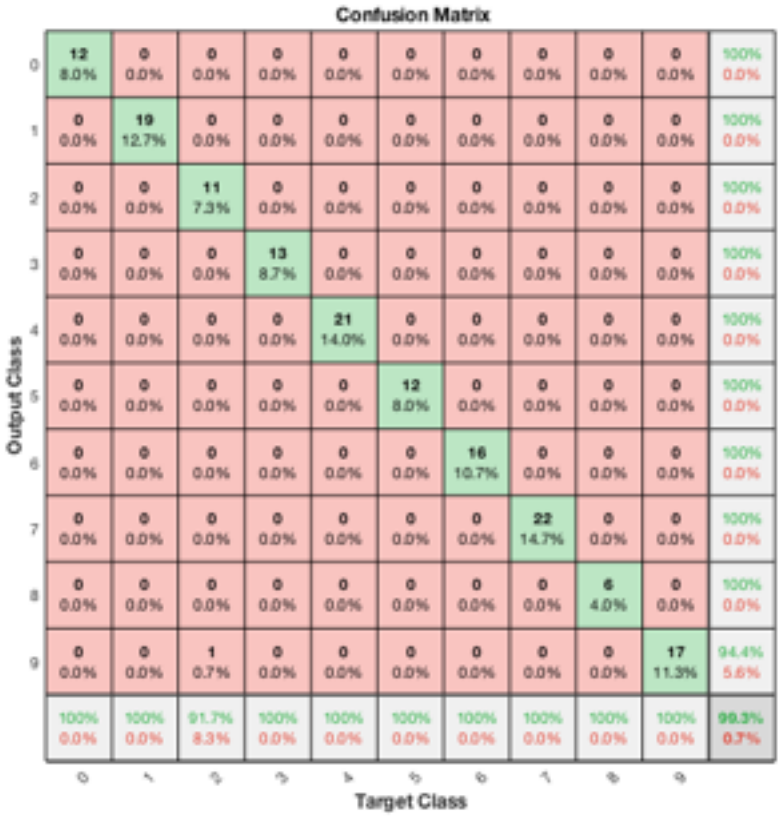
60000 training patterns



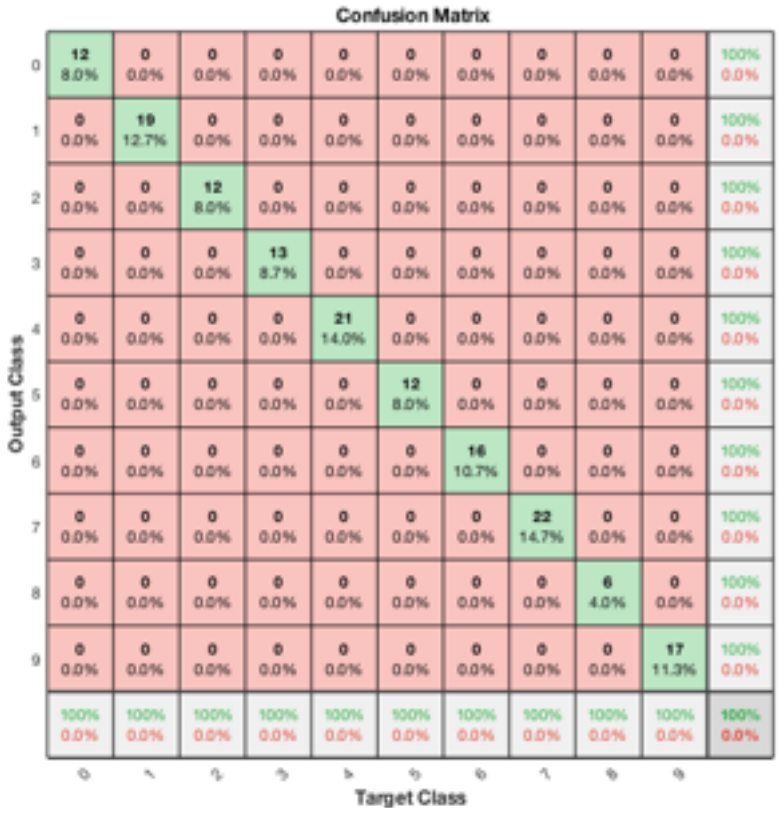
all correct !

(5)confusion matrix

1000 training patterns



60000 training patterns



d. 如何做

(a)(b)Self-programming的部分，我是利用老師上課所教的，還有參考老師給的paper和網路上的資料，主要是根據convolution的定義，先將其中一個函數padding起來成一個新的矩陣，讓我能夠用一般的數學運算，就能夠表示一個函數翻轉後滑過另一個函數的概念。（不過在這次project中的函數，翻轉前後並沒有差異，所以我沒有把翻轉的程式寫上去）

(c)根據助教上課所教的，還有參考網站，去撰寫程式。主要是load data, 給定所需的layers和options，再將training data放進model即完成。

e. references

http://www2.compute.dtu.dk/~pcha/HNO/chap3.pdf
https://www.mathworks.com/examples/computer-vision/community/36153-deep-learning-for-classification-on-the-mnist-dataset
https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/67074/versions/2/previews/demos/Demo1_MNIST/prepareData.m/index.html
https://www.mathworks.com/help/deeplearning/examples/visualize-features-of-a-convolutional-neural-network.html
https://en.wikipedia.org/wiki/Convolution
paper: 2D-Convolution Ch 3 2008 Gonzalez Page 149.pdf

f. Matlab programs

(a)

```
%toolbox
f=[0.1 0.1 0.1 0.1];
g=[0.1 0.1 0.1 0.1];
conv(f, g)
%self
a=[0.1 0 0 0;
  0.1 0.1 0 0;
  0.1 0.1 0.1 0;
  0.1 0.1 0.1 0.1;
  0   0.1 0.1 0.1;
  0   0   0.1 0.1;
  0   0   0   0.1];
b=g';
c=a*b
```

(b)

```
%toolbox
f=0.1*ones(4, 4);
g=0.1*ones(4, 4);
conv2(f, g)
%self
a=[zeros(3, 10);
  zeros(4, 3) f zeros(4, 3);
  zeros(3, 10);];
b=g;
c=zeros(7, 7);
for i=1:7
    for j=1:7
        for x=1:4
            for y=1:4
                c(i, j)=c(i, j)+a(i+x-1, j+y-1)*b(x, y);
            end
        end
```

```
    end
end
display(c);

(c)
%load data
filename1_1 = '/Users/Winnie/Downloads/三上/圖形識別/matlabprac/train-images.idx3-ubyte';
fp1_1=fopen(filename1_1,'r');
magic = fread(fp1_1, 1, 'int32', 0, 'ieee-be');
ntrain = fread(fp1_1, 1, 'int32', 0, 'ieee-be');
numRows = fread(fp1_1, 1, 'int32', 0, 'ieee-be');
numCols = fread(fp1_1, 1, 'int32', 0, 'ieee-be');
rawImgDataTrain = uint8(fread(fp1_1, ntrain * numRows * numCols, 'uint8'));
fclose(fp1_1);
rawImgDataTrain = reshape(rawImgDataTrain, [numRows, numCols, ntrain]);
rawImgDataTrain = permute(rawImgDataTrain, [2,1,3]);
imgDataTrain(:,:,1,:) = uint8(rawImgDataTrain(:,:,:));
filename1_2 = '/Users/Winnie/Downloads/三上/圖形識別/matlabprac/train-labels.idx1-ubyte';
fp1_2=fopen(filename1_2,'r');
magic = fread(fp1_2, 1, 'int32', 0, 'ieee-be');
ntrain = fread(fp1_2, 1, 'int32', 0, 'ieee-be');
labelsTrain = fread(fp1_2, ntrain, 'uint8');
fclose(fp1_2);
labelsTrain = categorical(labelsTrain);
filename2_1 = '/Users/Winnie/Downloads/三上/圖形識別/matlabprac/t10k-images.idx3-ubyte';
fp2_1=fopen(filename2_1,'r');
magic = fread(fp2_1, 1, 'int32', 0, 'ieee-be');
ntest = fread(fp2_1, 1, 'int32', 0, 'ieee-be');
numRows = fread(fp2_1, 1, 'int32', 0, 'ieee-be');
numCols = fread(fp2_1, 1, 'int32', 0, 'ieee-be');
rawImgDataTest = uint8(fread(fp2_1, ntest * numRows * numCols, 'uint8'));
fclose(fp2_1);
rawImgDataTest = reshape(rawImgDataTest, [numRows, numCols, ntest]);
rawImgDataTest = permute(rawImgDataTest, [2,1,3]);
imgDataTest = uint8(zeros(numRows, numCols, 1, ntest));
imgDataTest(:,:,1,:) = uint8(rawImgDataTest(:,:,:));
filename2_2 = '/Users/Winnie/Downloads/三上/圖形識別/matlabprac/t10k-labels.idx1-ubyte';
fp2_2=fopen(filename2_2,'r');
magic = fread(fp2_2, 1, 'int32', 0, 'ieee-be');
ntest = fread(fp2_2, 1, 'int32', 0, 'ieee-be');
labelsTest = fread(fp2_2, ntest, 'uint8');
fclose(fp2_2);
labelsTest = categorical(labelsTest);
layers = [
    imageInputLayer([28 28 1], 'Name', 'input')
    convolution2dLayer(3,16,'Padding',1, 'Name', 'conv_1')
    batchNormalizationLayer('Name', 'BN_1')
    reluLayer('Name', 'relu_1')
    maxPooling2dLayer(2,'Stride',2,'Name', 'MP_1')
    convolution2dLayer(3,32,'Padding',1,'Name', 'conv_2')
    batchNormalizationLayer('Name', 'BN_2')
    reluLayer('Name', 'relu_2')
    maxPooling2dLayer(2,'Stride',2,'Name', 'MP_2')
    convolution2dLayer(3,64,'Padding',1, 'Name', 'conv_3')
    batchNormalizationLayer('Name', 'BN_3')
    reluLayer('Name', 'relu_3')
    fullyConnectedLayer(10, 'Name', 'FC_1')
    softmaxLayer('Name', 'SM')
    classificationLayer('Name', 'output')];
miniBatchSize = 8192;
```

```
options = trainingOptions( 'sgdm',...
    'MiniBatchSize', miniBatchSize,...
    'Plots', 'training-progress');
%%%plot layers
lgraph=layerGraph(layers);
figure;
plot(lgraph);
%%%1000 train
net1 = trainNetwork(imgDataTrain(:,:,1,1:1000), labelsTrain(1:1000), layers, options);
%%%1000 test
predLabelsTest1 = net1.classify(imgDataTest(:,:,1,1:1000));
accuracy = sum(predLabelsTest1 == labelsTest(1:1000)) / 1000
%150 testing patterns
figure;
hold on;
target=zeros(150, 1);
for i = 1:150
      subplot(15, 10, i);
      img=imgDataTest(:,:,1,i);
      imshow(img);
      title(char(labelsTest(i)));
      text(0, 35, sprintf('predicted: %c',predLabelsTest1(i)));
end
%confusion
figure;
plotconfusion(labelsTest(1:150), predLabelsTest1(1:150));
%plot feature map
for layer = 2:15
    name = net1.Layers(layer).Name;
    if strfind(name, "conv")==1
        channels=net1.Layers(layer).NumFilters;
    else if strfind(name, "FC")==1
            channels=net1.Layers(layer).OutputSize;
        end
    end
    I = deepDreamImage(net1,layer,1: channels, 'Verbose',false,'PyramidLevels',1);
    figure;
    I = imtile(I,'ThumbnailSize',[64 64]);
    imshow(I);
    title(['Layer ',name,' Features'])
end
%%%60000 train
net2 = trainNetwork(imgDataTrain, labelsTrain, layers, options);
%%%10000 test
predLabelsTest2 = net2.classify(imgDataTest);
accuracy = sum(predLabelsTest2 == labelsTest) / numel(labelsTest)
%150 testing patterns
figure;
hold on;
target=zeros(150, 1);
for i = 1:150
      subplot(15, 10, i);
      img=imgDataTest(:,:,1,i);
      imshow(img);
      title(char(labelsTest(i)));
      text(0, 35, sprintf('predicted: %c',predLabelsTest2(i)));
end
%confusion
figure;
plotconfusion(labelsTest(1:150), predLabelsTest2(1:150));
```