基于CNN、SVM的图像分类

检查系统配置

```
deviceInfo = gpuDevice;

computeCapability = str2double(deviceInfo.ComputeCapability);
assert(computeCapability > 3.0, ...
    'This example requires a GPU device with compute capability 3.0 or higher.')
```

数据集

数据集使用李飞飞等人收集的数据库caltech101，该数据库有101个类别，和1个背景图片文件夹。

```
datasetsFolder = '../../datasets/caltech101'; % define output folder
url = 'http://www.vision.caltech.edu/Image_Datasets/Caltech101/101_ObjectCategories.tar.gz';
if ~exist(datasetsFolder, 'dir') % download only once
    disp('Downloading 126MB Caltech101 data set...');
    untar(url, datasetsFolder);
end

% 加载数据集
rootFolder = fullfile(datasetsFolder, '101_ObjectCategories');
categoriesFolders=dir(rootFolder);
categoriesFolders(1:3)=[];% 去除背景文件夹
categories= {categoriesFolders(:).name}';
% categories=categories(randperm(length(categories),20));
% categories=categories(1:10);

imds = imageDatastore(fullfile(rootFolder, categories), 'LabelSource', 'foldernames');
```

显示类别和数量

```
tbl = countEachLabel(imds);
tbl(1:5,:)
```

```
ans =
        Label        Count
    ----------     -----

    Faces          435
    Faces_easy     435
    Leopards       200
    Motorbikes     798
    accordion       55
```

为了使各类样本数量平衡，选取数量最少的为基准抽取样本

```
minSetCount = min(tbl{:,2});
imds = splitEachLabel(imds, minSetCount, 'randomize');
tbl = countEachLabel(imds);
tbl(1:5,:)
```

```
ans =
      Label       Count
    __________    _____

    Faces         31
    Faces_easy    31
    Leopards      31
    Motorbikes    31
    accordion     31
```

加载**AlexNet CNN**网络

```
cnnURL = 'http://www.vlfeat.org/matconvnet/models/beta16/imagenet-caffe-alex.mat';
cnnMatFile = fullfile('../../alexnet', 'imagenet-caffe-alex.mat');
if ~exist(cnnMatFile, 'file') % download only once
    disp('Downloading pre-trained CNN model...');
    websave(cnnMatFile, cnnURL);
end
convnet = helperImportMatConvNet(cnnMatFile)
```

```
convnet =
  SeriesNetwork (具有属性):

    Layers: [23×1 nnet.cnn.layer.Layer]
```

展示**CNN**结构

```
convnet.Layers
```

```
ans =
  23x1 Layer array with layers:

     1   'input'               Image Input                   227x227x3 images with 'zerocenter' normal
     2   'conv1'               Convolution                   96 11x11x3 convolutions with stride [4  4
     3   'relu1'               ReLU                          ReLU
     4   'norm1'               Cross Channel Normalization   cross channel normalization with 5 channe
     5   'pool1'               Max Pooling                   3x3 max pooling with stride [2  2] and pa
     6   'conv2'               Convolution                   256 5x5x48 convolutions with stride [1  1
     7   'relu2'               ReLU                          ReLU
     8   'norm2'               Cross Channel Normalization   cross channel normalization with 5 channe
     9   'pool2'               Max Pooling                   3x3 max pooling with stride [2  2] and pa
    10   'conv3'               Convolution                   384 3x3x256 convolutions with stride [1
    11   'relu3'               ReLU                          ReLU
    12   'conv4'               Convolution                   384 3x3x192 convolutions with stride [1
    13   'relu4'               ReLU                          ReLU
    14   'conv5'               Convolution                   256 3x3x192 convolutions with stride [1
    15   'relu5'               ReLU                          ReLU
    16   'pool5'               Max Pooling                   3x3 max pooling with stride [2  2] and pa
    17   'fc6'                 Fully Connected               4096 fully connected layer
    18   'relu6'               ReLU                          ReLU
    19   'fc7'                 Fully Connected               4096 fully connected layer
    20   'relu7'               ReLU                          ReLU
    21   'fc8'                 Fully Connected               1000 fully connected layer
    22   'prob'                Softmax                       softmax
    23   'classificationLayer'   Classification Output       cross-entropy with 'n01440764', 'n0144353
```

```
% 展示第一层结构
```

```
convnet.Layers(1)
```

```
ans = 
  ImageInputLayer (具有属性):

                Name: 'input'
           InputSize: [227 227 3]

   Hyperparameters
    DataAugmentation: 'none'
       Normalization: 'zerocenter'
```

```
% 展示最后一层结构
convnet.Layers(end)
```

```
ans = 
  ClassificationOutputLayer (具有属性):

            Name: 'classificationLayer'
      ClassNames: {1000×1 cell}
      OutputSize: 1000

   Hyperparameters
    LossFunction: 'crossentropyex'
```

```
% 原始CNN网络的输出类别数
numel(convnet.Layers(end).ClassNames)
```

```
ans = 1000
```

图像预处理

AlexNet CNN以227 227 3的RGB图像作为输入 这里把将样本拉伸到227*227并转换为RGB图像的函数 作为imageDatastore的读取时调用的函数

```
imds.ReadFcn = @(filename)readAndPreprocessImage(filename);
```

分割样本

将样本随机分为训练集和测试集

```
[trainingSet, testSet] = splitEachLabel(imds, 0.3, 'randomize');
```

选取**CNN**的**fc7**层输出作为特征向量

```
featureLayer = 'fc7';
trainingFeatures = activations(convnet, trainingSet, featureLayer, ...
    'MiniBatchSize', 32, 'OutputAs', 'columns');
```

训练多类别**SVM**

```
trainingLabels = trainingSet.Labels;
```

一个svm分类器最多对两类进行分类，旧版本matlab没有集成多分类器的svm工具箱，只能借助Libsvm。这里利用新的fitcecoc函数，可以获得基于纠错输出编码和svm的多分类器。典型的使用二分类器构造多分类器的编码方案是训练与类别数目相同的二分类器，每个分类器对应一个类别，以输出最值所指类别为多分类器的结果，即one versus all编码。fitcecoc提供了多种编码方案，这里选择简单的one versus all编码，加快训练速度

```
% 选择线性svm
classifier = fitcecoc(trainingFeatures, trainingLabels, ...
    'Learners', 'svm', 'Coding', 'onevsall', 'ObservationsIn', 'columns');
```

测试分类器

测试训练集

```
predictedLabels = predict(classifier, trainingFeatures');

% 获取训练集混淆矩阵
confMat = confusionmat(trainingLabels, predictedLabels);
% 转换为百分比
confMat = bsxfun(@rdivide,confMat,sum(confMat,2));
% 显示精度
mean(diag(confMat))
```

```
ans = 1
```

测试测试集

```
% 获取测试集特征向量
testFeatures = activations(convnet, testSet, featureLayer, 'MiniBatchSize',32);
predictedLabels = predict(classifier, testFeatures);
testLabels = testSet.Labels;
confMat = confusionmat(testLabels, predictedLabels);
confMat = bsxfun(@rdivide,confMat,sum(confMat,2));
mean(diag(confMat))
```

```
ans = 0.7826
```

保存SVM分类器

```
save('classifier','classifier');
```

预处理函数

```
function Iout = readAndPreprocessImage(filename)

I = imread(filename);

% 把灰度图像转换为RGB图像
if ismatrix(I)
    I = cat(3,I,I,I);
end

% 拉伸到277*277
```

```
Iout = imresize(I, [227 227]);
end
```