Deep Learning Model I: AlexNet

In this exercise, we will practice to use one pre-trained model, AlexNet, to recognize objects in a few images, and extract features to examine image similarities.

Please use AlexNetEx.m for the exercise.

After installing MATLAB deep learning toolbox and AlexNet. You can first load the pre-trained model.

```
% load pre-trained model
net = alexnet;
net.Layers
inputSize = net.Layers(1).InputSize;
```

First, we will practice to let the pre-trained model recognize objects in a few images.

```
testimg = imread('img1.png');
testimginput =
testimg(1:inputSize(1),1:inputSize(2),1:inputSize(3));
[label_testimg,scores] = classify(net,testimginput);
```

The first output argument shows the label of object category that produces the highest score for recognition. The second output argument includes the recognizing scores (the output of the CNN model) for the 1000 categories.

Second, we will use some Matlab build-in functions to extract features in different layers, and then use MDS to show the locations of images in a 2D space.

(1) Use “activations” function to extract feature vectors for each test image
(2) Use “pdist” function to compute the cosine distances of pairwise feature vectors, which can yield the distance matrix for all test images
(3) Run nonmetric MDS to display images in a 2D space. Use “mdscale” function.
(4) Show the MDS result plot.

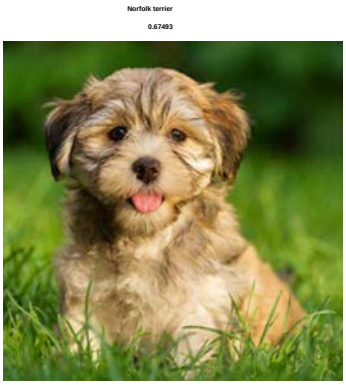
In-class Exercise:

Show MDS results for three layers, “conv2”, “conv5”, and “fc7”. Study the differences from the MDS results.

If your computer can not run AlexNet, feature vectors from the three layers were provided in the three mat file. You can load in these feature vectors from the mat files to complete the in-class exercise.

Results

Step 1: object recognition from images



Step 2: Feature extraction and MDS