Machine Learning, Spring 2018
Project 1: Image Segmentation via clustering
Due Date: April 5, 2018

Overview.
Image segmentation refers to a class of algorithms that use unsupervised learning to find "interesting" regions in digital images. In this project, you should use clustering algorithms to cluster two types of images: RGB (color) images and Hyperspectral images. You will use the following clustering algorithms:

- K-means
- Self-Organizing Map
- Fuzzy C-means (FCM)
- Spectral Clustering
- Gaussian Mixture Models (GMM).

Code to be written.

Clustering Code. You will write a function or method called "`MyClust`" that take an image (RGB or Hyperspectral) three optioinal Name-Value pairs and returns:

- A labeled image, called `ClusterIm`, with clusters labeled by positive integers ranging from 1 to NumClusters.

In addition, when the input is an RGB image, `MyClust` should return:

- A labeled image, called `CCIm`, with connected components in `ClusterIm` labeled by positive integers ranging from 1 to the Number of Connected Components in `ClusterIm`.

The three Name-Value pair inputs to `MyClust` have names '`Algorithm`', '`ImageType`', and `'NumClusts'` .

'`Algorithm'` can take the following string values

- `'Kmeans'`
- `'SOM'`
- `'FCM'`
- `'Spectral'`
- `'GMM'`

'`ImageType`' can take the string values

- `'RGB'`
- `'Hyper'`

`'NumClusts'` can take integer values `NumClusts` = 1,2,... with an upper bound calculated to be 25% of `N`, where `N` is the name of pixels in the input image. If `NumClusts` is 1, then use the closest integer to `0.05*N` as the value of `NumClusts.`

These functions should use some logic and quantitative measures to try and determine the number of clusters. An example call to MyClust should be of the following form:

```
[ClusterIm, CCIm] = MyClust(Im, 'Algorithm', 'FCM', 'ImType', 'RGB', 'NumClusts', 3);
```

Evaluation Code. You will write 2 functions that evaluate clustering algorithms called "`EvalClustRGB`" and "`EvalClustHyper`". `EvalClustRGB` that takes CCIm and a ground-truth segmentation as input and produces a score using the *Martin index*. `EvalClustHyper` takes `ClusterIm` as an input. These functions are described in more detail in what follows.

Clustering RGB images.

Data and Code Files:
A set of RGB images and segmentations are contained in the .zip file: `ImsAndSegs.zip`
A set of MATLAB scripts are in the directory: `Project1`

The .zip file contains .mat files created in MATLAB. Each file contains an image and 3 possible segmentations of the image. The 3 possible segmentations serve as approximations of the ground truth.

Loading them into MATLAB, e.g. executing the commands:

```
FileName   = 'ImsAndTruths100075.mat';
ImsAndSegs = load(FileName);
```

results in

```
ImsAndSegs = struct with fields:

      Im: [321×481×3 uint8]
    Seg1: [321×481 uint16]
    Seg2: [321×481 uint16]
    Seg3: [321×481 uint16]
```

You can display these images using the function call: `DispImAndSegs(ImAndSegs);` as shown in Fig. 1.

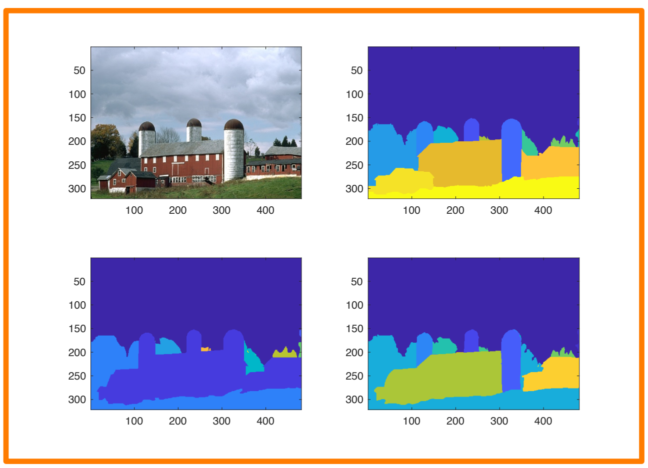

Figure.1 Images Displayed using Function Call `DispImAndSegs(ImAndSegs)`

RGB Image	Segmentation 1
Segmentation 2	Segmentation 3

Names of the displays

You can look at all the images and segmentations using the script: `CheckEm`. Sometimes the 1st segmentation looks better, sometimes the 2nd, and sometimes the 3rd. Sometimes you can't tell. Figure 2 shows some examples.

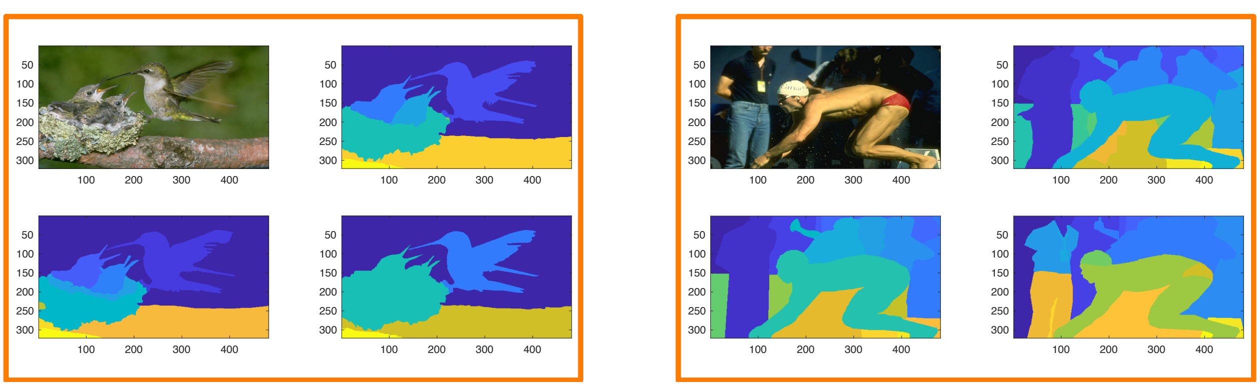

Figure 2. Two examples of segmentations of RGB images. In the left-hand display, segmentations 1 or 2 could be considered the best. In the right-hand side, segmentation 3 could be the display.

Evaluation:
Evaluation plays a large role in Machine Learning. Recall that the "correct" answer in clustering depends on the application. There is no application in this project so you will evaluate your programs by comparing your segmentations to those contained in the files. There are a multitude of methods for evaluating segmentations of images. You should use the *Martin index*, which is defined in the paper:

Mark Polak, Hong Zhang, Minghong Pi, "An evaluation metric for image segmentation of multiple objects", Image and Vision Computing, Volume 27, Issue 8, 2009, Pages 1223-1227.

which can be found in the directory Project1 or at
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.966&rep=rep1&type=pdf

Each segmentation found by your algorithms should be compared to the 3 Segmentations in the .mat file. The best of the resulting 3 values of the Martin index, called OCE, can be used as the score. Each clustering algorithm will have a score for each of the 198 images. The mean and standard deviation of the scores over all the images should be calculated and the results shown in the following tabular form:

self-organizing map

Algorithm:	kmeans mean	kmeans std. dev.	fcm mean	fcm std. dev.	spectral mean	spectral std. dev.	GMM mean	GMM std. dev.
Score:	M1	S1	M2	S2	M3	S3	M4	S4

where Mk and Sk denote scaled means and standard deviations of the scores. Since OCE is between 0 and 1, display the means and standard deviations using integers obtained by Mk = round(1000*mk) and Sk = round(1000*sk) where mk and sk denote the calculated means and standard deviations.

Hyperspectral Images:

You are given two hyperspectral images, both taken with airborne imagers (imaging spectrometers). An image over Pavia, Italy and an image over Santa Barbara, CA. The pixels in these images have dimensions 178 and 103 respectively so segmenting these images requires working in a much higher dimensional space than with RGB images. Example pixels are shown in Fig. 3

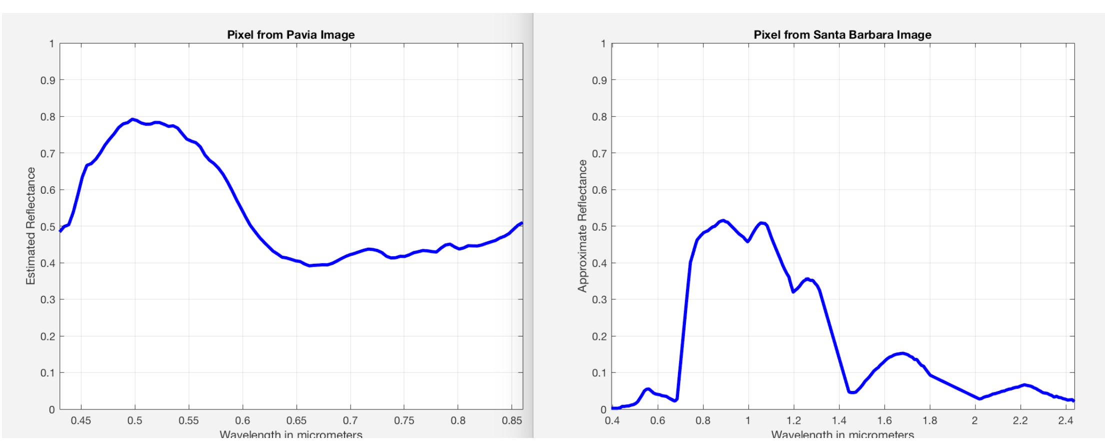

Figure 3. A pixel from the Pavia image and a pixel from the Santa Barbara Image. The Pavia and Santa Barbara pixels can be treated as 103 and 178 dimensional vectors, respectively.

The evaluation is not easy in this case. You should just evaluate the Santa Barbara image subjectively using the RGB image as a guide. The RGB image is shown in Figure 4 and is also in the directory Project1.

Figure 4. An RGB depiction of the Santa Barbara image.

The Pavia image has partial ground truth. The Pavia has an labeled image of the ground-truthed regions and a mask that is 1 at those points at which there is ground truth and 0 otherwise. The images are shown in Figure 5.

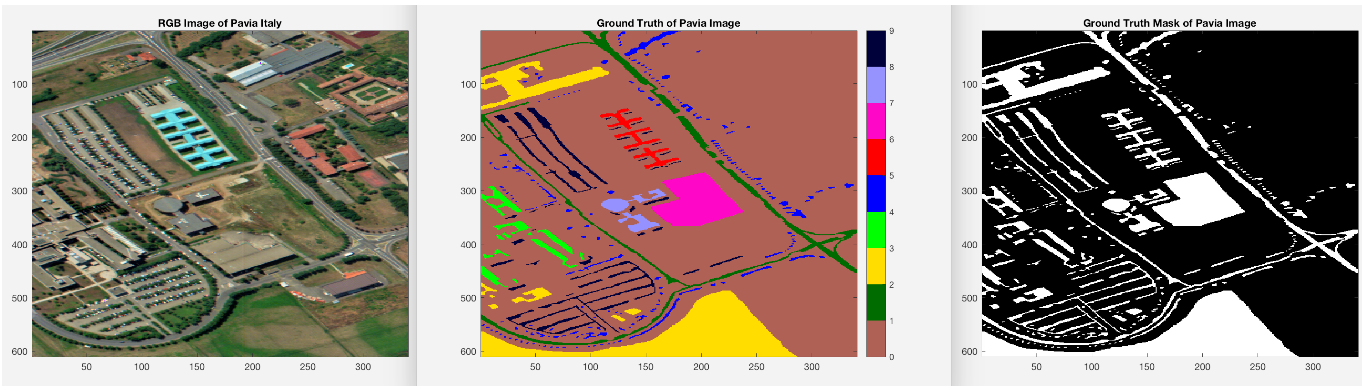

Figure 5. An RGB depiction of the Pavia Hyperspectral image, the ground image with labels shown in the color bar, and the grouth truth mask. The label 0 in the labeled ground truth image means there is no ground truth at that pixel.

The evaluation should be performed by multiplying your segmentation by the ground truth Mask. Don't use connected components; just use cluster labels. You can set the number of clusters to 9 since there are 9 ground truthed regions.

Summary:

You should turn in functions (written in either MATLAB or Python):

```
MyClust<GroupNumber>
MyKmeans<GroupNumber>
MyFCM<GroupNumber>
MySpectral<Group Number>
MyGMM<GroupNumber>

MyMartinIndex<GroupNumber>
MyClustEvalRGB<GroupNumber>
MyClustEvalHyper<GroupNumber>
```

They should not depend on the particular files you are given, just to the file format and directory structure. I should be able to run on different images stored using the same file format and directory structures.

Team Member Roles:

There are four people on a team. Two people should work on the clustering algorithm. The other two should work on the evaluation code. The two people who work on the clustering algorithm should test the evaluation code and the two that work on the evaluation code should test the clustering algorithm. Similarly, the two that work on the clustering (evaluation) algorithm should write the parts of the report associated with the clustering (evaluation) algorithm experiments and observations. The two that write the the clustering (evaluation) part should edit the evaluation (clustering part). One person is in charge of uploading the deliverables.

Deliverables:

The **Code** listed above should be turned in through Canvas. A Readme.txt file should be turned in that defines any dependencies on other tools (such as functions/methods to find connected components).

A **Report** should be turned in as a .pdf file, with single column, 1.5 line spacing and with the following format. Use the exact format!

Project 1 Report
<Group Members>

1. Member Roles

A table describing who did what in the following form with Membern replaced by a name.

	Cluster Alg	**Eval Alg**	**Cluster Test**	**Eval Test**	**Cluster Write**	**Eval Write**	**Cluster Edit**	**Eval Edit**
Member1	1			1	1			1
Member2	1			1	1			1
Member3		1	1			1	1	
Member4		1	1			1	1	

2. Experiments

Test: program correct
reasonable result

2.1. RGB Experiments and Tabular Results (1 page)

Explain experiments conducted using pseudo-code. Include table of Martin Indices defined above.

2.2. Hyperspectral Experiments (1 page)

Describe experiments using pseudo-code. For Pavia, describe the evaluation algorithm using pseudo-code. Perform experiments by reducing the dimensionality of the pixels using Principal Components. For the Pavia image, describe results in a table

3. Observations (<= 4 pages)

For all image types, describe any observations you have regarding the advantages/disadvantages of each algorithm and their relative performance. You can use pictures of results to illustrate your points. Rank the algorithms by your assessment of their usefulness/accuracy for image segmentation using two separate tables of the form.

3.1. K-means advantages and disadvantages
3.2. FCM advantages and disadvantages
3.3. Spectral Clustering advantages and disadvantages
3.4. Gaussian Mixture Models advantages and disadvantages
3.5. Relative performance of algorithms

Table X. Algorithm Ranking Table.

Ranks	RGB Rankings	Hyperspectral Rankings
1 (Best)	`AlgRGB1`	`AlgHyper1`
2	`AlgRGB2`	`AlgHyper2`
3	`AlgRGB3`	`AlgHyper3`
4	`AlgRGB4`	`AlgHyper4`

Identify differences between RGB (low-dimension) and hyperspectral (high-dimension) and the effects of reducing dimensionality using PCA.

Grading. Your grade will be based on

Code (50%)

Correct format	(15%)
Runs with no errors	(20%)
Reasonable Results	(15%)

Report (50%)

Correct format	(15%)
Succinctness and Clarity	(10%)
Result reasonableness	(10%)
Objective analysis in Observations	(15%)