German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Mohammed A.-M. Salem
DMET1067 Deep Learning in Computer Vision, Spring Semester 2022

Assignment 1

Total Points 100 - Due to Thursday April 7th 2022

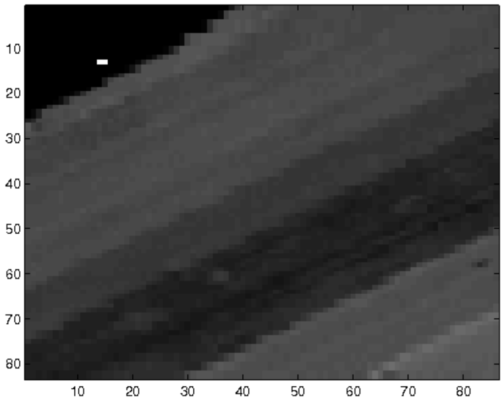
Regulation:

1- You may work in teams up to 2 students.
2- The deadline is on April 7th 2022 till 11:59 pm.
3- A deduction of 5% for the late submissions till 5:59 am of the next morning.
4- A deduction of 30% for the late submissions after 6:00 am of the next morning.
5- Submit your work to DLCV.GUC@gmail.com.
6- The Subject of the Email is Assign1-[ID1_ID2]
7- The deliverables are (in one zip file named as the email subject, rar files are not accepted):
    a. Source Code (Use comments to describe the code lines). Add your name, Uni-Id and emails inside the source code
    b. Report on MS-Word or MS-Powerpoint. The report should contain names of all team members, their Uni-IDs and emails, a detailed description of the steps. It should include all results and tables, and figures.
    c. The title page of the report should include your name, id and email.
    d. Notepad file (.txt) includes your names, IDs and emails. Let the file be named as "names.txt".

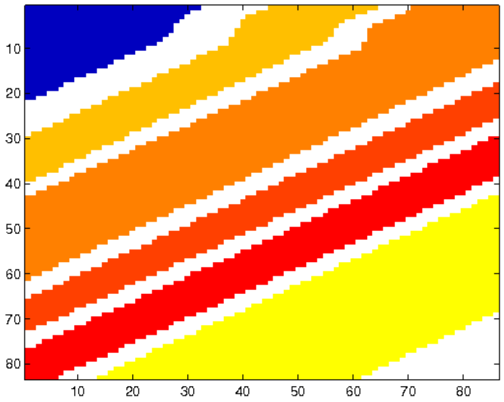
Test images: source:

Dataset 1

http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Salinas-A_scene



Sample band of Salinas-A dataset

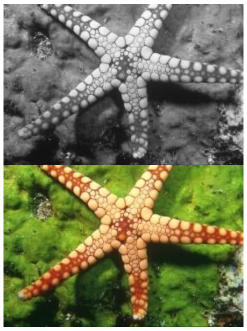


Ground truth of Salinas-A dataset

Dataset 2

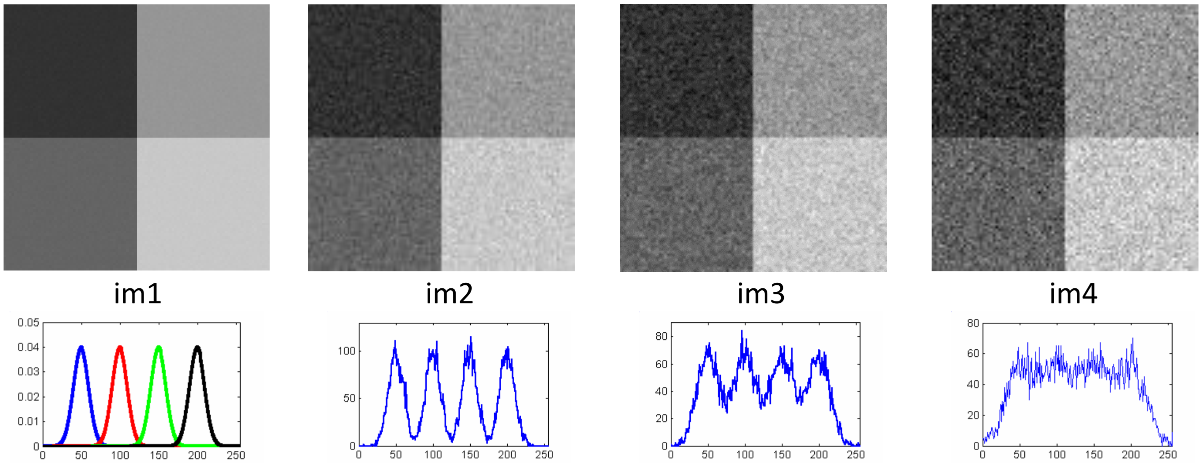
https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images/

Some examples of simple gray and color images.



Task 1: Dataset 3	10 Points

[8 Points] Create images similar to the following figures test images for evaluating the segmentation process.



im1 im2 im3 im4

Histogram im1 Histogram im2 Histogram im3 Histogram im4

1- All images are 512x512 gray level.
2- The first image is an ideal image with no noise added. (im1 and im5).

3- The following three images are with added noise. Add three level of noise to create smooth image, noisy image and very noisy image. Name them as testGrayImage_low, testGrayImage_mid, and testGrayImage_hi. It is not expected that you get identical images or histograms to the ones displayed above.

4- Include the images in your submission.

5- Add full description on the procedure followed to create the images and how the noise level was chosen.

Task 2		26 Points

Use Python or MATLAB to complete the following tasks:

1. [10 Points] Implement the K-means algorithm.
    a. Don't use a built-in function for that. Your algorithm should accept as inputs:
        i. Data: a vector of ***n*** sample points with dimension ***D***. For example: gray level images (one dimensional data), RGB images (3-dimensional data) or even multi-spectrum images (*n*-dimensional data).
        ii. The number of classes ***K***.
        iii. The initial class centers $C^0=\{C_1^0 , C_2^0, \ldots, C_K^0\}$. Note that the data and the initial class centers have the same dimension and are of the same type such unsigned 1 byte integer.
2. Test the K-means algorithm on images of Dataset 3.
    a. Apply the algorithm on two of the images of Dataset 3.
    b. [10 Points] Construct a confusion matrix for each gray level test image. Report the results in your report.
3. Test the K-means algorithm on images of Dataset 2.
    a. Select a color and the corresponding gray-level images.
    b. Apply the algorithm on the selected images.
    c. [5+1 Points] Show the images before and after the K-means algorithm and comment of the results in your report.
4. [Optional]Apply K-means for hyperspectral image classification
    a. Load the hyperspectral image and the segmentation ground truth saved in .mat files. You can use the following lines:

```
#MAT files created with Matlab up to version 7.1 can be read using the
mio module part of scipy.io. Reading structures (and arrays of structur
es) is supported, elements are accessed with the same syntax as in Matl
ab
from scipy.io import loadmat
x = loadmat('/content/drive/My Drive/Colab Notebooks/SalinasLine.mat')
data = x['salinasA_corrected_line']
```

b. Reshape the image to be one vector of pixels where each pixel has 204 subbands.
c. Apply the k-means algorithm on this image. The image contains 7 classes in total.
d. Report your results and comment on.

Task 3	27 Points

1- Write an implementation of the EM algorithm for gray-level images. Thus the diminssion of the input data is 1.

2- The program should accept as inputs the algorithm parameters (number of classes, the initial estimatoins of the Gaussian Mixture Model parameters, and the tolerance level to be used on the stoping condition ε) and the input data in form of a matrix.

3- It should return the classification matrix with the same size as the input matrix, the predicted GMM parameters, and the number of iterations.

4- Use the images of Dataset 3. The following inputs should be used:
   a. The number of classes $K = 4$.
   b. The weight of each distribution in the mixture $p_k^0 = \frac{1}{K} : k = 1,\ 2, \dots K$
   c. Initial guess of the distributions' parameters: $\{[\mu_1^0 = 25, \sigma_1^0 = 1],\ [\mu_2^0 = 90, \sigma_2^0 = 1],\ [\mu_3^0 = 160, \sigma_3^0 = 1], [\mu_4^0 = 240, \sigma_4^0 = 1]$.
   d. $\varepsilon = 0.01$.

5- [8 Points] For each test construct the confusion matrix and compute the overall accuracy, and the classification accuracy of each class.

6- [8 Points] Report the results. For each case show the input image, the noise level, the resulted classification matrix, the corresponding confusion matrix and the accuracies, and the number of iterations.

7- [3 +3 Points] Test the algorithm for different values for ε and for each test report the performance (number of iterations *N*) and accuracy (Classification Accuracy) of the algorithms. Let $\varepsilon = 0.01,\ 0.05,\ and\ 0.1$ and report the results similar to point 6.

8- [5 Points] In your report discuss the following points:
   a. What happens when we change the noise level in the input image on the results of the segmentation visually and on the accuracy?
   b. What is the effect of changing the epsilon parameter?
   c. How the GMM complexity increases with the increase of the dimension of the input data?

Task 4	**14 Points**

1. To achieve good results with supervised classification techniques when dealing with color images, a quantization step may be used on the input image before applying the algorithm. The objective is to reduce the number of different cases. This will reduce the size of the lookup tables too. If the color depth of a color channel or the gray level is 8 bits, a simple quantization step could be implemented by reducing it into 4 bits. In other words, to represent the entire color channel or the gray level range using 16 colors only instead of 256. In the following equation the input gray level color is x, y is the corresponding gray level after quantization, QI is the quantization interval, in this case IQ = 16. The operator $\lfloor x \rfloor$ returns the greatest integer less or equal to x.

$$y = IQ * \left\lfloor \frac{x}{IQ} \right\rfloor$$

2. Apply the discussed quantization step on a selected color and gray level images.
   a. Select two gray level images from Dataset 2.
   b. Apply the quantization step.
   c. [5+2 Points] Report the images before and after the quantization.
   d. From the Dataset 2 select the corresponding color images of the previously selected gray level images and apply the quantization step.
   e. [5+2 Points] Report the images before and after the quantization.

Task 5	**25 Points**

Use Python or MATLAB to complete the following tasks:

3. Implement a binary class Naïve Bayes classifier algorithm. Name it ***BM* = *BayesModel(dta, gt)***
   a. Don’t use a built-in function for that. Your algorithm should accept as inputs the training dataset:
      i. ***dta***: a vector of ***n*** sample points with dimension ***l***. For example: gray level images (one dimensional data), RGB images (3-dimensional data) or even multi-spectrum images (204-dimensional data).

         The recommended format for this input is a matrix with length ***n*** and width ***l***. Thus the rows represent the sample points and the columns in one row represent the features of this point.

ii. The labels vector ***gt*** (ground truth). This vector has the same length of the data vector. The number of distinguished values in this vector represents the number of classes. The recommended format for this input is a column vector with length ***n***.

b. The output of the algorithm is a trained model that is ready for testing.

```
% The corresponding Matlab method is:

Mdl =  fitcnb(X,Y) returns a multiclass naive Bayes model (Mdl), trained by predictors X and class labels Y.
```

4. Implement a prediction function to test the trained Bayes classifier model. Name it ***lbl*** = ***BayesPredict(BM, td)***

   a. The inputs are the trained Bayes Model and the test dataset. The ***td*** is expected to have the same format as the ***dta***.

   b. The output of this function ***lbl*** is a column vector contains the predicted labels of the test sample points. The length of ***td*** and ***lbl*** is the same.

```
% The corresponding Matlab method is:

YPred = predict(net,X) predicts responses for the image data in X using the trained model.
```

5. Implement a function to compute the confusion matrix. Name it ***Mtrx*** = ***ConfMtrx(gt, lbl)***.

6. [5 Points] The model should be trained on the 80% of the data, to save 20% for testing. The sample points for training and testing should be selected randomly.

7. Apply the algorithm on test gray level images.

   a. Select a color image from Dataset 2 and apply the quantization step (Task 4).

   b. Use a simple built-in thresholding technique to binarize the corresponding gray-level image. Use the resulted binary image to serve as the ground-truth of the classification of the color images.

   c. Apply the Naïve Bayes classifier algorithm on the selected color image.

   d. [15+5] Report the input image with the ground truth, the resulted predicted binary image and the confusion matrix.

To load the hyperspectral image and the segmentation ground truth saved in .mat files. You can use the following lines:

```
#MAT files created with Matlab up to version 7.1 can be read using the mio module part of scipy.io. Reading structures (and arrays of structures) is supported, elements are accessed with the same syntax as in Matlab
from scipy.io import loadmat
x = loadmat('/content/drive/My Drive/Colab Notebooks/SalinasLine.mat')
data = x['salinasA_corrected_line']
```

Good luck