ECE 5332-011: Deep Learning for Medical Signal/Image data

Assignment 1: Introduction to Image Processing and Machine Learning Techniques.

Spring 2019 Due: 02/10/2019

The purpose of this assignment is to brush up the basic concepts and fundamentals required in this course. It is recommended to use MATLAB for this assignment as the dataset, images and functions are readily available in MATLAB. Python can be used as well provided you are using the same dataset and images.

1. <u>Basic Image processing</u>

To get a brush up on image processing using MATLAB, go through the video tutorial playlist at: 'https://www.youtube.com/watch?v=-cSVGwAwZZ4&list=PLEo-jHOqGNyUWoCSD3l3V-FjX9PnHvx5n' .

a. Loading and saving an image

From the MATLAB image dataset, load the image 'onion.png'. Notice that this is a color image and has 3 color planes. Convert this to a grayscale image and save it as 'onion_gray.png'.

b. Introduction to the 2D Fourier space

Load the image 'cameraman.tif' from the MATLAB image database and obtain the 2D Fourier transform of the image. Use the *'fftshift()'* function to bring the DC component at the center and the logarithmic intensity transform to display the 2D Fourier space. The following commands can be used to achieve that:

```
I = imread('cameraman.tif');
If = fftshift(fft2(I));
If_log = log(1 + abs(If));
If_disp = If_log/max(If_log(:));
imshow(If_disp,[])
```

Next, reconstruct the image using different regions of the Fourier space. Create two binary masks, one to keep just the center region, with some fixed radius around it, of the Fourier space and second to keep everything but the center region of the Fourier space. Reconstruct the image using masked Fourier space images and comment on the reconstructed images. Also try to vary the radius of the binary mask and report the changes in the reconstruction. Use the *'ifft2( )'* function to reconstruct the images. The sample code is given below for a 256 x 256 image:

```
[x,y] = meshgrid(-128:127,-128:127);
Z = sqrt((x.^2) + (y.^2));
R = 25; % Radius
```

```
% low freq mask
Zl = zeros(size(Z));
Zl(Z < R) = 1;
If_low = If.*Zl;
I_rec_low = ifft2(ifftshift(If_low));

% High freq mask
Zh = zeros(size(Z));
Zh(Z > R) = 1;
If_high = If.*zh;
I_rec_high = ifft2(ifftshift(If_high));
```

c. Spatial filtering and convolution

2D convolutions are used to apply different transformations to the image. The convolution kernel decides the output of the convoluted image and hence selection of a kernel becomes of prime importance. We would be testing the effect of a spatial low pass kernel (integrator) and a spatial high pass kernel (differentiator).

Load the 'coins.png' image from the MATLAB dataset. First convolve the image with 3x3 Sobel kernel and obtain the gradient of the image. What do we observe in the output image? Secondly, use the 5x5 box blur kernel. What do we observe in the output image? Classify and explain which one of the following acts as a low pass and a high pass kernel?

2. <u>Clustering</u>

Clustering refers to a process of grouping similar feature vectors into same class. The two main types of unsupervised algorithms are 'k-means' and 'hierarchical' clustering. Load the '*kmeansdata.mat*' dataset. Notice that the dataset contains 560 feature vectors each of which are 4 dimensional. Plot any 2 dimensions against each other and try to estimate the number of clusters that might be present in the dataset. Use MATLAB's 'kmeans' and 'cluster' function to classify the dataset into the desired number of clusters using k-means and hierarchical clustering respectively. What happens when we try to cluster the dataset into more or a smaller number of clusters than the actual number of clusters present? Also mention how the two clustering algorithms are different from each other with advantages and limitation of both.

3. <u>Simple Image classification problem.</u>

"*It is better to have a simple classifier and complex features rather than a complex classifier and simple features.*" Based on that, we would be creating a simple classifier that would take complex image features as input and classify a given image based on image features.

a. Feature detector and descriptor

An Image feature refers to a key-point or an area of interest unique to that image. A feature detector detects such unique features in an image and a descriptor describes such feature point in a unique way. Use the functions *'detectSURFFeatures()'* and *'extractFeatures()'* to detect SURF features and descriptors in the 'cameraman.tif' image. Observe different parameters like feature location, orientation, dimensionality, strength etc. Display the original image along with 10 strongest features highlighted.

b. Creating the training feature matrix

For this part, we would be using image dataset provided by Columbia University. Download the dataset here: 'http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php'.

The dataset contains images of simple objects in different angle of rotation. For this project we would be using objects 3,4,5,6,9,10,12,13,14 and 19. We would be using projections 0 to 11 and 60 to 71; i.e. 24 images per object. We would use some of the 24 images from each object as training images and the rest as test images.

Find SURF features for each training image of a given object. Select and store feature descriptor for N strongest features. That would form a feature matrix for that particular object. Form a feature matrix for each object.

c. Testing

For testing, find SURF features for any given test image and use the 'matchFeatures()' function to match the features between test image and the training feature matrix. The test image is assigned to the object whose training matrix has the highest number of matches.

Compute the accuracy once all test images are assigned to one of the objects. Vary the number of training images from 5 to 20 in steps of 5 and compare the accuracy with respect to it. Also change the number of features used to create the training matrix to N = {16, 8, 4, 2} and see the effect of it on accuracy.

Submission:

You are expected to work in groups of three. Please upload a single zip file containing a brief report either in word or PDF and a single MATLAB or python code. Please include necessary figures and tables in the report with proper captions.