Spring 2018	CMPEN 497: Applied Computer Vision & Machine Learning for IoT Applications	Homework 2
	Due: (Sunday) April 1st, 11:55 PM	
This assignment represents 12% of the overall course grade		

Team Members Submission Statement:

We confirm that we followed the course policies by contributing equal amount of work to this assignment. Below is the approximate ratios of our contributions.

Name	PSU email	Contribution %	Signature

- *You will submit a separate PDF formatted delivery report for each one of the following questions: Name the reports as follows: TeamName_Question1.PDF and TeamName_Question2.PDF. The first page of the report must include the signed statement shown above.*
- *Although it is team-based project, but each team member may be assigned a different grade according to his/her contribution, understanding, and ability to demonstrate the work or answer insightful questions about the submitted work.*

Tentative Grading rubric

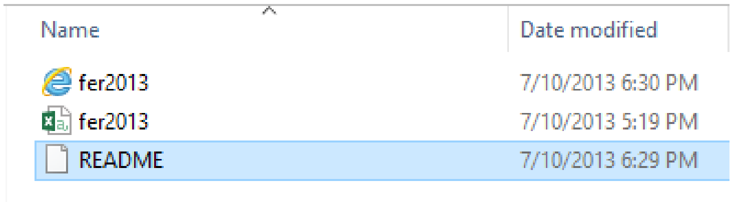
Question	Criteria	Percent	Actual Grade
	Correct & clean implementation code	60	
	Correct answers of questions & ability to explain and demonstrate your work	25	
	Report format & quality	15	
Bonuses			
Overall Total		100	

Question 1 [Expected time to complete (4 to 6) hours per team]

Problem Description

1. Download the dataset of facial expressions (*…) available at the following link:*

- *a.* *https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data*
- *b.* *You will need to create an account on* www.kaggle.com *to download the data, it is free to create an account.*
- *c.* *You will need Tar extension on your Winzip or you will need Linux system (at least bash terminal) in order to unzip the compressed data file.*
  - *i.* *After decompressing the data, you will get the three files shown below. We are interested only in the (.csv) file.*



Name	Date modified
fer2013	7/10/2013 6:30 PM
fer2013	7/10/2013 5:19 PM
README	7/10/2013 6:29 PM

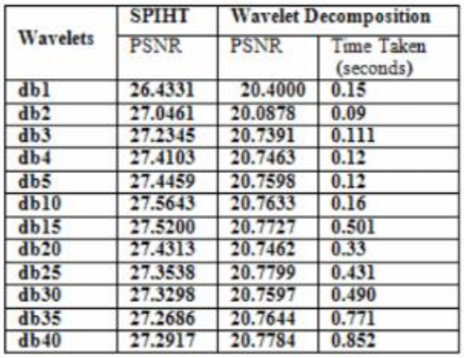
Figure (1)

2. Dataset description:
   - a. The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).
   - *b.* The (train.csv) contains three columns, the "emotion" and "pixels" columns are the ones that really matter. The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column. *Hint: you will need to convert the string to integer array (of type uint8) to prepare the data for any further processing. Only convert to double if needed since double data type uses significantly larger memory space.*
   - c. The training set consists of 28,709 samples for training, 3,589 for testing (public testing) and another 3,589 for (private testing). Feel free to use only the 28,709 samples or the entire dataset (35887 samples). Your code should be able to use the partitioning (2/3 to 1/3) approach to divide the data randomly for training and testing as you desire (as learned at the lab).
   - d. Acknowledgment: this dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project.

3. **Important Tips**: Reading the entire (CSV) file and convert it into arrays of double or integers is a complex loading and parsing issue. The Matlab built-in functions to do this use very poor algorithm that has (high time and memory complexity). Hence, you will need to parse the elements as smaller chunks, preferably between 5000 and 10000 rows at a time. I provided you with a template code to do so. It is the file named (hw2_template). The script calls another Matlab function named (importfileAsColVectors) that I generated. Make sure that all files in the same directory with the (.csv) data file in order for this to work. *Note that the (emotions) vector is your data labels.*

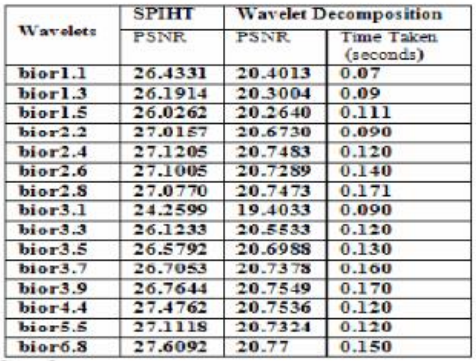
4. Develop your own Matlab code to extract features vector based on 2D-wavelets analysis from the dataset as explained in the wavelets features extraction supplement at the end of this document.
    a. Notice that I am not asking you to just generate level (1) of the wavelets tree then concatenate the wavelets coefficients and use them as they are. But you have unlimited possible combinations that you can come up with to choose which coefficients (for example which quarter of the four quarters (LL, LH, HL, HH) to keep or for how many levels you would like to generate wavelet coefficients to add to your features.
    b. Although you have infinite options and possibility to extract as many levels as you want, remember that we use the wavelets so that you do not use the original image. So your new features vector per each data sample cannot be of a higher dimension than the input image total pixels. If your system does not provide at least 50% reduction in the input dimensionality, the CTO of the company to buy your product will surely drop the deal and never deal with your employer.
    c. Once decided, explain in details which features you have chosen. Provide a simple illustration , preferably visual representation and add it into your report
5. Repeat the process using 2 different independent types of wavelets. Figure 2 gives a summary of a set of the most famous wavelets. Notice that wavelets that have higher PSNR should help you maintain more information of the original input, then it is expected to give you better detection results if your neural network is structured properly. However, you must choose a technique that runs fast enough so that you don't spend days generating the new wavelets data from your original dataset of grayscale images.

Table 1. Debauchies Wavelets Analysis



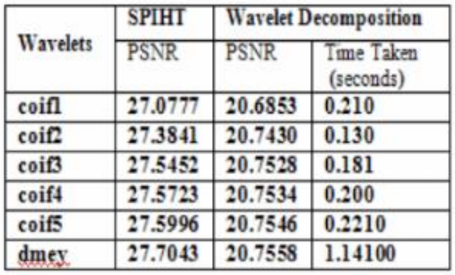
Wavelets	SPIHT	Wavelet Decomposition	
	PSNR	PSNR	Time Taken (seconds)
db1	26.4331	20.4000	0.15
db2	27.0461	20.0878	0.09
db3	27.2345	20.7391	0.111
db4	27.4103	20.7463	0.12
db5	27.4459	20.7598	0.12
db10	27.5643	20.7633	0.16
db15	27.5200	20.7727	0.501
db20	27.4313	20.7462	0.33
db25	27.3538	20.7799	0.431
db30	27.3298	20.7597	0.490
db35	27.2686	20.7644	0.771
db40	27.2917	20.7784	0.852

Table 2. Biorthogonal Wavelets Analysis



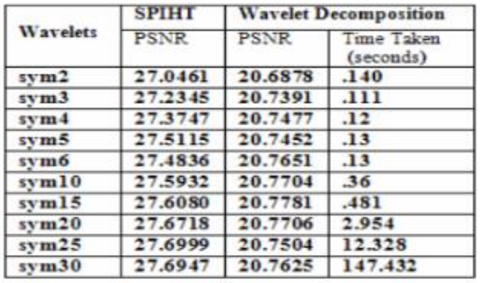
Wavelets	SPIHT	Wavelet Decomposition	
	PSNR	PSNR	Time Taken (seconds)
bior1.1	26.4331	20.4013	0.07
bior1.3	26.1914	20.3004	0.09
bior1.5	26.0262	20.2640	0.111
bior2.2	27.0157	20.6730	0.090
bior2.4	27.1205	20.7483	0.120
bior2.6	27.1005	20.7289	0.140
bior2.8	27.0770	20.7473	0.171
bior3.1	24.2599	19.4033	0.090
bior3.3	26.1233	20.5533	0.120
bior3.5	26.5792	20.6988	0.130
bior3.7	26.7053	20.7378	0.160
bior3.9	26.7644	20.7549	0.170
bior4.4	27.4762	20.7536	0.120
bior5.5	27.1118	20.7324	0.120
bior6.8	27.6092	20.77	0.150

Table. 3. Coiflets Wavelets Analysis



Wavelets	SPIHT	Wavelet Decomposition	
	PSNR	PSNR	Time Taken (seconds)
coif1	27.0777	20.6853	0.210
coif2	27.3841	20.7430	0.130
coif3	27.5452	20.7528	0.181
coif4	27.5723	20.7534	0.200
coif5	27.5996	20.7546	0.2210
dmey	27.7043	20.7558	1.14100

Table 4. Symlets Wavelets Analysis



Wavelets	SPIHT	Wavelet Decomposition	
	PSNR	PSNR	Time Taken (seconds)
sym2	27.0461	20.6878	.140
sym3	27.2345	20.7391	.111
sym4	27.3747	20.7477	.12
sym5	27.5115	20.7452	.13
sym6	27.4836	20.7651	.13
sym10	27.5932	20.7704	.36
sym15	27.6080	20.7781	.481
sym20	27.6718	20.7706	2.954
sym25	27.6999	20.7504	12.328
sym30	27.6947	20.7625	147.432

Figure (2)

6. After generating and saving the two different sets of features by using two different discrete wavelets analysis categories. You should now have two sets of features that you can use for machine learning with your properly designed Neural Networks. For example, you have a first set named (HaarData.mat) and a second set named (SymletData.mat).
7. Modify the (NN_lab1.m) code and develop additional Matlab code to construct, train and test two neural networks (HaarNet) and (SymletNet), which receive the HaarData and the SymletData as their training and testing vectors.
8. The new Neural Networks MUST be capable to distinguish between the 7 facial expressions categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). (Hint: Matlab (nnstart) call allows you to use the NN GUI to generate the code of your NN structure. Also, here are some helpful details:
   a. Create the network such that it uses the wavelets features
   b. The network will consist of three or more layers (input, hidden layer #1, … , output). YES, I am allowing you this time to add more than one hidden layer to handle any possible non-linearity in your features. However, I recommend using no more than two layers.
   c. For the hidden layers, you may need to sweep the # of neurons to decide which one to choose. Instead of sweeping using the entire size of the dataset, you may use (10%-15%) portion of the dataset for sweeping (7,000 samples for example). Then pick a reasonable size to run the full experiment and collect your results.
   d. Train and test the network, then generate the ROC curve for each one of the (7) facial expressions categories. *Add the ROC curves with short explanation of each one to your delivery report. Make sure to clearly mention which ROC curve belongs to which expression.*
   e. Also, *plot the final system accuracy and add it to your delivery report as well.*
9. Repeat the process for each network until you find the network with the best accuracy.
10. Provide a performance comparison between the two final networks in your report covering the following (Add into your Delivery report):
    a. The accuracy chart resulting from sweeping scenarios that you run for each network. If you use two hidden layers, then this plot must have 3 axes. This X-axis is the # of neurons in hidden layer (1), the Y-Axis is the # of neurons in hidden layer (2), and the Z-axis is the accuracy value.
    b. The ROC Curve and confusion matrix of each facial expressions for the final configurations that you have chosen (this is the configurations that provided you with the best results = highest accuracy).
    c. Clearly emphasize which network generated higher accuracy, then try to explain based on your insights about wavelets and neural networks why this design and this family of wavelets worked better? *Hint: try some research online to find out the answer to why in terms of wavelets families.*

Deliverables:

1. Two separate set of Matlab files that implement steps (3-4) and steps (6-9) separately. You can use more files if you plan to divide your code into functions. I do not want to control how you use your good programming skills.

    a. Full Matlab code submission through a designated folder on the team Git Repository. I will use Matlab 2017a for testing your codes, this is our standard environment.

2. README.txt file that describes in details how to use and run your code to regenerate the exact results you have successfully acquired. This is very important file because it facilitates grading your code. Upload the README file to the repository folder and upload another copy to the Canvas folder when uploading your delivery report.
3. Delivery Report (PDF Format)
    a. You should write your report as very neat documentation that allows me to understand everything you developed so that I can run the code to generate the same exact outputs on my own.

Features Extraction Supplement

- This guide depends heavily on the slides from Week 9 lecture on Wednesday. Figure (3) shows the basic stages to generate the wavelet features then save it. *HINT: if the wavelets concatenated dataset is too big when saved to single file, you can split the data into 2 to 4 smaller size matrices, save them to separate files. Then, when you intend to load the data for training and testing your future NNs, you can load each file separately into a separate matrix variable before you concatenate the variables back into one large matrix.*

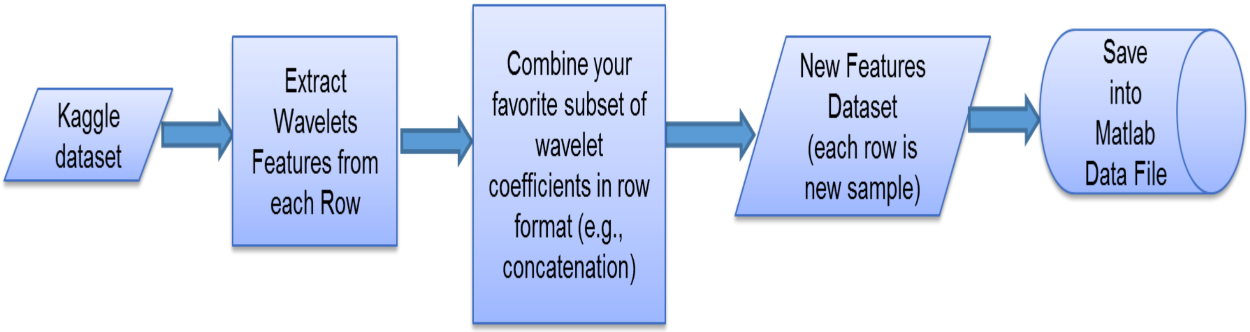

Kaggle dataset
Extract Wavelets Features from each Row
Combine your favorite subset of wavelet coefficients in row format (e.g., concatenation)
New Features Dataset (each row is new sample)
Save into Matlab Data File

Figure (3)

- Figure (4) summarizes the stages to train and test a new neural network based on the pre-saved wavelets data file.

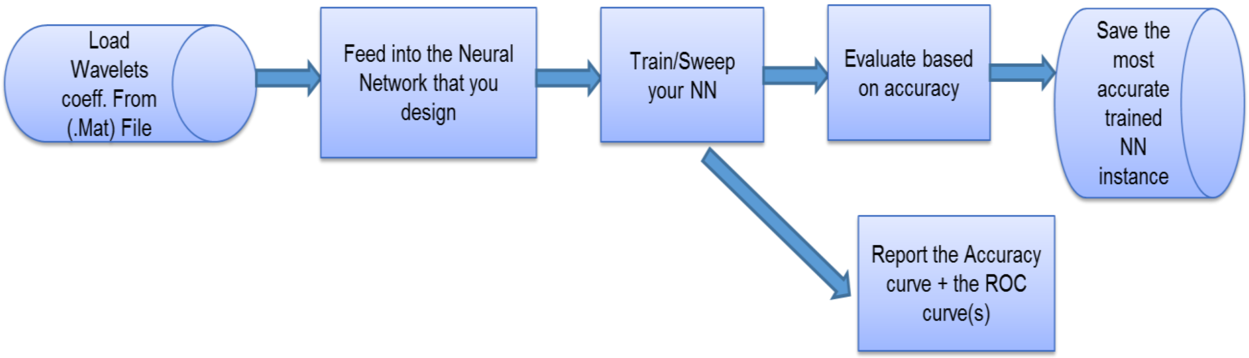

Load Wavelets coeff. From (.Mat) File
Feed into the Neural Network that you design
Train/Sweep your NN
Evaluate based on accuracy
Save the most accurate trained NN instance
Report the Accuracy curve + the ROC curve(s)

Figure (4)