German University in Cairo
Faculty of Media Engineering and Technology
Winter 2021

CSEN 1022 – Machine Learning

Assignment #1

(Due on November 10 at mid-night)
(This assignment can be done in teams of maximum 2 students)

You are required to design a Fisher's Linear Discriminant classification algorithm that can recognize images of three classes (airplane, automobile and frog). These classes are part of the machine learning benchmark CIFAR-10 dataset. The data is available on the GitHub assignment (https://classroom.github.com/a/vi-OLYlS). In the Data folder, you will find two folders: "train" and "test". The "train" folder contains 3 sub-folders named "airplane", "automobile" and "frog", with each one containing 5000 different images of the respective classes. The "test" folder also contains the same 3 subfolders with 1000 different images of each class. The images in the "train" folder should be used to train a classifier for each class using the method given at the bottom of slide 18 in Lecture 3.pdf. After the classifiers are trained, test each classifier using the images given in the "test" folder. Use the following equation for Fisher's Linear Discriminant $\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$. The bias term should be computed as explained in the lecture and Problem Set 3 using the equation $w_0 = -\mathbf{w}^T(\mathbf{m}_1 + \mathbf{m}_2)/2$. You will examine Fisher's Linear Discriminant twice. Once dealing with the provided RGB images, and once after converting them to gray-scale.

Deliverables:

- Your code.
- A confusion matrix using RGB images showing the number of images of the "test" folder of each class that were classified to belong to different classes (For example: Number of images of airplane that were classified as airplane, automobile and frog, and so on for the other classes).
- The classification accuracy achieved using RGB images.
- A confusion matrix using gray-scale images showing the number of images of the "test" folder of each class that were classified to belong to different classes (For example: Number of images of airplane that were classified as airplane, automobile and frog, and so on for the other classes). To convert the images from RGB to gray scale, implement the conversion using the equation:
  $I_{gray}(p) = 0.3 I_R(p) + 0.59 I_G(p) + 0.11 I_B(p)$
- The classification accuracy achieved using gray-scale images.
- Comparing the outcomes achieved using RGB images and using gray-scale images, which image format gives higher accuracy? Also, which of the 3 classes shows the biggest change in accuracy? What are your interpretations for these results?

Important Notes:

- Divide the R, G and B values by 255 to scale them between 0 and 1.
- If the inverse of $\mathbf{S}_W^{-1}$ results in a singular matrix, use the pseudoinverse function (np.linalg.pinv()).
- **Do not use Python built-in functions for covariance or the Fisher's linear discriminant function. You have to implement your own version of all needed functions. You are only allowed to use the mean function, functions that load images into Python and the libraries already imported in the first cell in the code provided on GitHub (numpy and matplotlib).**