MSML/DATA603

Project 2 **Due 02/08/2021**

1 Introduction

This is a group project. Each group should consist of 2 or 3 students. The aim of this project is to implement neural networks and use them for image recognition.

2 Datasets

POSE The same dataset as in project 1: cropped images of 68 subjects under 13 different poses. You can access the i_{th} pose of the j_{th} subject as `pose(:, :, i, j)`. Form the training set out of the first 10 poses for each subject, and the test set out of the remaining three poses for each subject.

MNIST Download the dataset MNIST of handwritten digits 0, 1, ..., 9 from http://yann.lecun.com/exdb/mnist/. The description of Dataset is also on the website. The training set has 60000 28 x 28 grayscale images of handwritten digits (10 classes) and a testing set has 10000 images. To read this dataset in Matlab, you can download the function

```
[imgs labels] = readMNIST(imgFile, labelFile, readDigits, offset)
```

written by Siddharth Hegde from Matlab file exchange. The script below reads the training images and the test images and saves them in a mat file `mnist.mat`:

```
imgFile = 'train-images.idx3-ubyte';
labelFile = 'train-labels.idx1-ubyte';
readDigits = 60000;
offset = 0;
[imgs_train, labels_train] = readMNIST(imgFile, labelFile, readDigits, offset);
labels_train = categorical(labels_train);
%
imgFile = 't10k-images.idx3-ubyte';
labelFile = 't10k-labels.idx1-ubyte';
readDigits = 10000;
offset = 0;
[imgs_test, labels_test] = readMNIST(imgFile, labelFile, readDigits, offset);
labels_test = categorical(labels_test);
%
save('mnist.mat','imgs_test','imgs_train','labels_test','labels_train');
```

3 Classifying tasks

1. Identifying subjects in the `pose.mat` dataset.
2. Identifying handwritten digits in the `MNIST` dataset.

Experiment with different architectures of neural networks and choices of parameters in them aiming at maximizing the percentage of correctly classified images.

4 Programming language

Matlab or Python with appropriate deep learning packages.

1. Matlab: Deep Learning Toolbox. If you do not have it already, download Deep Learning Toolbox from mathworks.com. Open Matlab help by clicking at the question mark icon at the top right corner of the Matlab's command window or editor window. Search for `trainNetwork`. Click on the first item that comes up: `trainNetwork`. There is a very helpful example that you can open as a Live Script. Next, to understand how to setup an architecture of a neural network, click on any word `layers` in Section Description. Then click on `List of Deep Learning Layers`. You might find helpful the following resources on convolutional layers:
   this site in mathworks, or
   this site in Machine Learning Mastery.
   Also this mathworks web page on AlexNet may be helpful.

2. Python with library PyTorch or TensorFlow.

Every student must develop an individual set of all codes. However, please feel free to communicate with each other and learn from each other.

5 Submission Guidelines

Please submit one report per working group of 2–3 students with descriptions of what has been done, choice of neural networks architectures, parameters, summaries of your observations, figures, tables, and conclusions. Every group member should link her/his codes to the report pdf. These can be e.g. Dropbox links, GitHub links, Google drive links, etc. Please DO NOT upload your codes on ELMS, only pdf files with links to your codes.

Reports must be prepared in LATEX or any other text editor. *Handwritten reports will not be graded.*