Exercise 2: TKO_3120 Machine Learning and Pattern Recognition

Goal: Developing and evaluating a deep learning model for one of the most important problems of visual recognition- image classification. For models, Convolution Neural Networks (CNNs) have been successfully used in a variety of computer vision tasks. This type of projects would involve understanding the state-of-the-art vision models, and building a CNN model.

Grading: The reports are evaluated on 0-3 points scale (0 means that the work is rejected). The points are counted as extra points for the exam. This project must be done alone.

Deadline: Tue 22 May, 2018 at 23:55.

Tools: Keras, Matlab, TensorFlow, PyTorch, Caffe

> **Tip:** I highly recommended you to use Keras in IPython notebook server with the `jupyter notebook` command.

Data set: The extended Yale Face Database B (same dataset in Exercise1).
The data set consists of face image of 38 different persons, each photographed in several conditions:

- Intensity of illumination
- Position of light source
- But only one pose, unlike in the original non-cropped database

All test image data used in this exercise are aligned, cropped, and then re-sized to 168x192 images. There are also some non-cropped images, you are allowed to ignore them.

> **Tip:** You can find more details about data set in the exercise.pptx file in moodle.

Download Data set:
The dataset is available in http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html -> (cropped Images)

OR
try https://seafile.utu.fi/f/2d94c45d1c214f4da640/.

Instruction for Submissions: Return the project report to Moodle in PDF format.

Contact info: fahimeh.farahnakian@utu.fi.

You are welcome to come to my office hours to brainstorm and suggest your project ideas.

Office hours: Wednesday 14:00-16:00, Agora 4th floor, room 452C

Tasks:

1. Create a CNN model with the following architecture:

Conv layer 1:

- Filter size = 5
- Number of filters = 16
- Activation function = relu

Pooling layer 1:

- max pooling = 2

Conv layer 2:

- Filter size = 5
- Number of filters = 36
- Activation function = relu

Pooling layer 2:

- max pooling = 2

Fully connected layer 1:

- Number of neurons: 512
- Activation function = relu

Fully connected layer 2:

- Number of neurons: number_Of_Classes
- Activation function = softmax

Regularisation: dropout (prob = 0.5)

2. Train the model on 80% of data for training and 20% of data for test and display a plot of the loss function and accuracy as shown below.

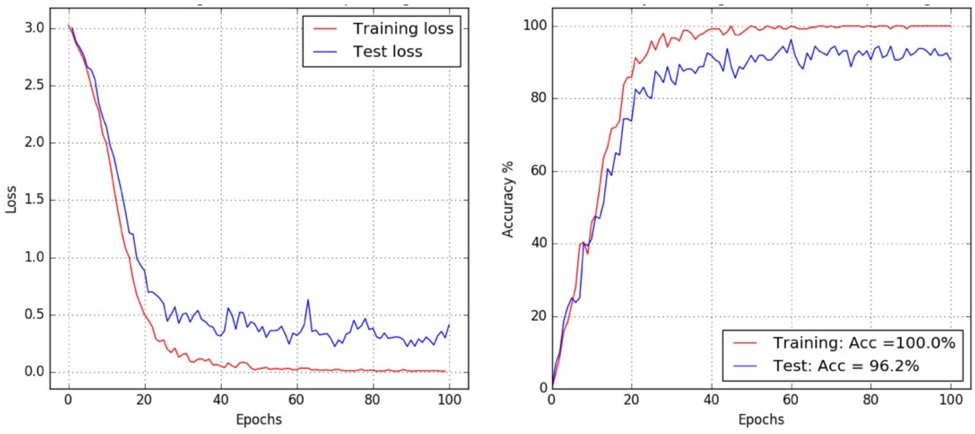

Figure1: This figure shows only an example

3. Evaluate the model on the test data and report the overall classification accuracy and the confusion matrix
4. Make predictions on some image from test dataset by using the trained model and display the predicted class and class probability on the image

Tip 1: The desired prediction accuracy (>70%) can be achieved by the model performance improvement (hyper-parameter optimization).

Tip 2: You should use the feature extraction and preprocessing tasks from the Exercise 1 in order to get a high classification accuracy in this exercise.

Report: Write a compact project report (maximum 5 pages). The report should consist of the following parts:

- Name and student number.
- Brief description of the implementation.
- Report the classification accuracy and loss plots, the overall classification accuracy, confusion matrix and a few example images of predicted class.
- Source code for implemented software, as well as brief descriptions of any used external software packages, and the commands and parameters used to run them.