Introduction to Computer Vision (ECSE 415)
Assignment 4: Neural Networks

DEADLINE: November, 3rd
Please submit your assignment solutions electronically via the **myCourses** assignment dropbox.

The submission should include a single Jupyter notebook. More details on the format of the submission can be found below. Submissions that do not follow the format will be penalized 10%.

The assignment will be graded out of a total of **100 points**. There are *50 points* for accurate analysis and description, *40 points* for bug-free and clean code, and *10 points* concerning the appropriate structure in writing your report with citations and references.

Each assignment will be graded according to defined rubrics that will be visible to students. Check out MyCourses, the "Rubrics" option on the navigation bar. You can use **OpenCV**, **sklearn**, **skimage**, **Numpy**, and **Pytorch** library functions for all parts of the assignment except those stated otherwise. Students are expected to write their own code. (Academic integrity guidelines can be found **here**).

Assignments received late will be penalized by 10% per day.

Submission Instructions

1. Submit a single Jupyter notebook consisting of the solution of the entire assignment.
2. Comment your code appropriately.
3. Give references for all codes which are not written by you. (Ex. the code is taken from an online source or from tutorials)
4. Do not forget to run **Markdown** ('Text') cells.
5. Do not submit input/output images. Output images should be displayed in the Jupyter notebook itself.
6. Make sure that the submitted code is running without error. Add a **README** file if required.
7. If external libraries were used in your code please specify their name and version in the **README** file.
8. We are expecting you to make a path variable at the beginning of your codebase. This should point to your working local (or google drive) folder.
   Ex. If you are reading an image in the following format:

   ```
   img = cv2.imread ( '/content/drive/MyDrive/Assignment1/images/shapes.png' )
   ```

   Then you should convert it into the following:

   ```
   path = '/content/drive/MyDrive/Assignment1/images/'
   img = cv2.imread(path + 'shapes.png')
   ```

   Your path variable should be defined at the top of your Jupyter notebook. While grading, we are expecting that we just have to change the path variable once and it will allow us to run your solution smoothly. Specify, your path variable in the **README** file.
9. Answers to reasoning questions should be comprehensive but concise.

1 Part 1 - CIFAR-10 Classification using Convolution Neural Network (50 points)

For this assignment, you are going to train models on the subset derived from the publicly available CIFAR-10 dataset **source**. The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. For more information, you are encouraged to look at their webpage. In this section, you are expected to implement a Convolution Neural Network(CNN) to classify the images based on their context. GPU is recommended for executing the code in this section.

1. Use Pytorch class torchvision.datasets.CIFAR10 to load the dataset and use the batch size of 32.
2. Implement a CNN with the layers mentioned below.
    - A Convolution layer with 32 kernels of size 3x3
    - A ReLU activation
    - A Convolution layer with 64 kernels of size 3x3
    - A ReLU activation
    - A maxpool layer with kernels size of 2x2
    - A convolution layer with 64 kernels of size 3x3
    - A ReLU activation
    - A convolution layer with 64 kernels of size 3x3
    - A ReLU activation
    - A flattening layer. (This layer resizes the 2D feature map to a feature vector, which should match the square of kernel size).
    - A Linear layer with output size of 10. (Classes should be predicted as numerical values(like 0-9).
3. Create an instance of SGD optimizer with a learning rate of 0.002. Use the default setting for the rest of the hyperparameters. Create an instance of categorical cross entropy criterion.
4. Train the CNN for 10 epochs and show the performance on the test images by displaying the accuracy.
5. Change the kernel size to 5x5 and train the new network with the same hyperparameters.
6. Compare the run time and the results of models under different kernel sizes and briefly discuss the possible factors that affect the performances of a CNN.

2 Part 2 - YOLO Object Detection on Montréal Streets (50 points)

YOLO is a fast object detection technique that is widely used in many research aspects. In this section, you will be asked to take a photo of a street in Montreal and summarize the information in this image by using a trained YOLO model.

1. Use your cellphone or a digital camera to capture of a street scene in Montréal.
2. Implement the YOLOv3 object detection approach to identify what are the types of objects included in the image (such as person, bicycle, vehicle, tree) and count the number of each object.
3. Show your result in a table with categories as the index column and quantity as the value column.
4. Display the original and predicted images in your notebook.

Tips: This section does not ask to implement and train the model. You may find useful information on how to use a well-constructed YOLO model from this website.