Introduction to Computer Vision (ECSE 415)
Assignment 3

Due: November 2^{nd}, 11:59PM

Please submit your assignment solutions electronically via the myCourses assignment dropbox. Attempt all parts of this assignment. The assignment will be graded out of total of **33 points**. Students are expected to write their own code. (Academic integrity guidelines can be found at `https://www.mcgill.ca/students/srr/academicrights/integrity`). Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be graded.

Submission Instructions

1. Prepare and submit two separate Google Colab notebooks for the two questions.
2. Comment your code appropriately.
3. Do not submit input/output images. Assume image folders are kept in a same directory as the codes.
4. Make sure that the submitted code is running without error. Add a README file if required.
5. Answers to reasoning questions should be comprehensive but concise.
6. Submissions that do not follow the format will be penalized 10%.

Note that you can use any of the OpenCV, sklearn, skimage and Pytorch functions shown during the tutorial sessions.

1 Image Classification using RF and SVM

For this task, you are given a dataset of flower images[1]. The dataset contains images of 9 types of flowers. You can read the images and the corresponding labels as follows.

[1] The dataset is derived from the 102-Category Flower dataset[1].

```
train_images = np.load('flower_subset.npz')['train_images']
train_labels = np.load('flower_subset.npz')['train_labels']
test_images = np.load('flower_subset.npz')['test_images']
test_labels = np.load('flower_subset.npz')['test_labels']
```

The arrays `train_images` and `test_images` are stacks of 1556 and 90 gray-scale images of size 128×128, respectively.

- Resize the train/test images to 64 × 64 and compute HoG features using cells of 8×8 pixels, blocks of 4×4 cells and 4 bins. This should yeild a feature vector of size 1600 per image. **(3 points)**
  (Suggestion: Make a function which takes list of images as arguments and delivers list of HoG features as output. The same function can be used for train and test set.)

- Fit a non-linear SVM classifier (use RBF kernel with gamma='auto' and C=1) on the features and the class labels of the training images. **(1 points)**

- Predict labels of the test images by feeding the test features to the trained classifier and calculate classification accuracy. **(2 points)**

- Tune values of hyperparameters 'gamma' and 'C' to achieve test accuracy greater than 50%. **(2 points)**

- Fit a Random Forest(RF) classifier (set n_estimators=10, max_depth=5 and criterion='entropy') on the features and the class labels of the training images. **(1 points)**

- Predict labels of the test images by feeding the test features to the trained classifier and calculate classification accuracy. **(2 points)**

- Tune values of hyperparameters 'n_estimators' and 'max_depth' to achieve test accuracy greater than 50%. **(2 points)**

- Compare results of SVM and RF classifiers. Which one provides better results? Experiment training both classifiers with a range of random states (try different random values for the argument 'random_state') and measure classification accuracy of the test set. Which classifier is more stable or robust to the change in random state? **(3 points)**

2 Image Classification with Convolution Neural Network (CNN).

In this part, you will classify MNIST digits [2] into 10 categories using a CNN. You may chose to run the code on GPU.

1. Use Pytorch class `torchvision.datasets.MNIST` to (down)load the dataset. Use batch size of 32. **(3 points)**

2. Implement a CNN with the layers mentioned below. **(5 points)**

   - A convolution layer with 32 kernels of size 3×3.
   - A ReLU activation.
   - A convolution layer with 64 kernels of size 3×3.
   - A ReLU activation.
   - A maxpool layer with kernels of size 2×2.
   - A convolution layer with 64 kernels of size 3×3.
   - A ReLU activation.
   - A convolution layer with 64 kernels of size 3×3.
   - A ReLU activation.
   - A flattening layer. (This layer resizes 2D feature map to a feature vector. The length of this feature vector should be 4096.)
   - A Linear layer with output size of 10.

   (Suggestion: you can start with the code from Tutorial 6 and adapt it for the current problem.)

3. Create an instance of SGD optimizer with learning rate of 0.001. Use the default setting for rest of the hyperparameters. Create an instance of categorical cross entropy criterion. **(1 point)**

4. Train the CNN for 10 epochs. **(5 points)**

5. Predicts labels of the test images using the above trained CNN. Measure and display classification accuracy. **(3 points)**

References

[1] Nilsback, Maria-Elena, and Andrew Zisserman. "Automated flower classification over a large number of classes." 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. IEEE, 2008.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.