**ASSIGNMENT - 4**
**Computer Vision**
**Topic: Deep Learning**

Total Marks: 100 Deadline: April 30, 2023.

---

**Instructions :**

1. Use Google Colab or Kaggle Notebook with GPU runtime to implement this assignment.
2. Do not copy from other students. Any case of plagiarism will result in zero marks.
3. You can refer to codes online (e.g., Github, Kaggle) but do not copy-paste. The resource must be cited in the report if referred.
4. Strictly follow the submission guidelines.
5. Allowed languages: python
6. You can use any inbuilt library if not mentioned to code from scratch.

**Submission Guidelines :**

1. Submit .py python files and .ipynb for all the questions.
2. Strictly submit a single report (.pdf) for all the questions. No .doc, .docx file will be accepted.
3. If you are using Colab, then attach your Colab link in the report (preferred)
4. Submit a single zip file containing all Python files and report.
5. The name of the zip file should be roll_no.zip, and python files should have the name, e.g roll_no_qu1.py or roll_no_qu2.py, etc. The report should have the name roll_no.pdf. If the naming convention is not followed, we will award zero marks.

---

Q1. Perform image classification using CNN on the MNIST dataset. Follow the standard train and test split. Design an 8-layer CNN network (choose your architecture, e.g., filter size, number of channels, padding, activations, etc.). Perform the following tasks:

1. Show the calculation of output filter size at each layer of CNN.
2. Calculate the number of parameters in your CNN. Calculation steps should be clearly shown in the report.
3. Report the following on test data: (should be implemented from scratch)
    a. Confusion matrix
    b. Overall and classwise accuracy.
    c. ROC curve. (you can choose one class as positive and the rest classes as negative)
4. Report loss curve during training.
5. Replace your CNN with resnet18 and compare it with all metrics given in part 3. Comment on the final performance of your CNN and resnet18.

Q2. Download the Flicker8k dataset [images, captions]. Implement an encoder-decoder architecture for Image Captioning. For the encoder and decoder, you can use resnet/densnet/VGG and LSTM/RNN/GRU respectivly. Perform the following tasks:

1. Split the dataset into train and test sets appropriately. You can further split the train set for validation. Train your model on the train set. Report loss curve during training.
2. Choose an existing evaluation metric or propose your metric to evaluate your model. Specify the reason behind your selection/proposal of the metric. Report the final results on the test set.

Q3. Use the dataset from Assignment 3 (Q. 4). Train YOLO object detection model (any version) on the train set. Compute the AP for the test set and compare the result with the HOG detector. Show some visual results and compare both of the methods.