

# PROGRAMMING ASSIGNMENT-3

## PART-II

ROBOT VISION

### Notes

Programming language for the assignment is **Python** and you will use **PyTorch** framework for deep learning. You can use standard python built-in IDLE, or other IDLEs such as CANOPY, PyCharm Community Edition, PyScripter, CodeSculptor, Eric Python, Eclipse plus PyDev, etc.

Following libraries can be used when necessary:

- PIL (The Python Imaging Library), Matplotlib, NumPy, SciPy, LibSVM, OpenCV, VLFeat, python-graph.

Submit part-1 by **30th April 2021**, 11.59pm.

### Question 2: Convolutional Neural Networks (CNN) [5 pts]

Your goal in this assignment is to train convolutional neural networks for image classification. You will use CIFAR-10 dataset, which has 60K color images (each has size 32x32 pixels) from 10 classes. You will be provided the template code for this assignment and you have to make some changes to the network and analyze the results after these changes. *For each of these task, use learning rate of 0.1 and batch size of 100 and train them for 10 epochs each.*

#### Your tasks:

- 2 pt **Simple CNN:** In this task, your goal is to design a convolutional neural network with 2 convolutional layers (*Conv2d*) layers and 2 pooling layers, followed by 2 fully connected layers. Both *Conv2d* layers should have 10 filters (output channels). The second *Conv2d* layer's input channels should match first *Conv2d* layer's output channels. Use a kernel size of 3 for all convolutional layers. Apply *ReLU* activation to each *Conv2d*. Each *Conv2d* layer should be followed by *max\_pool2d* layer with kernel size of 2. The output features from convolution after flattening will be 360, so set the input features in fully connected layer accordingly (You can use *fc1\_model1*) for this). There are 10 classes in CIFAR-10 dataset, so this will be a 10-way classification network. You can use *model\_0* from the template and modify that to fit this task.
- 1.5 pt **Increase filters:** In this task, you will increase the filters in each *Conv2d* layer in your network. Learn 20 kernels for the first *Conv2d* layer (set output channels to 20). For the second *Conv2d* layer learn 40 kernels (set the output channels to 40). Match input channels for second *Conv2d* layer with output channels of first *Conv2d* layer. Since this will change output feature size after second *Conv2d* layer, use *fc1\_model2* with 1440 input features for this task.
- 1.5 pt **Large CNN:** In this task, your goal is to increase the size of the network. Take the network from previous task and add one more *Conv2d* layer with 40 filters (set both input and output channels to 40). Do not add a max pooling layer after this third convolution layer. Use *fc1\_model3* with 640 input features for this task.

**What to submit:**

- Code: The completed code for all the tasks.
- A short write-up about the results and your observations from each tasks. For each task, you should report the training/testing accuracy for the best model. Analyze the variation in training/testing loss as you train your network and discuss what you observe. Also, discuss the time required for training your network.