# CS 639: Deep Learning for Computer Vision, Spring 2025
# Problem Set 2

Instructor: Yong Jae Lee (yongjaelee@cs.wisc.edu)
TA: Harris Zhang (harrisz@cs.wisc.edu)
TA: Jaden Park (jadenpark@cs.wisc.edu)

Due: Thursday, March 13th, 11:59 PM

**Instructions**

1. Download the zipped assignment file from Canvas.

2. Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the `*.ipynb` file. No installation or setup is required. For more information, please see this tutorial on using Colab.

3. Next, we recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. Work through the notebook, executing cells and implementing the codes in the `*.py` file as indicated. You can save your work, both `*.ipynb` and `*.py`, in Google Drive (click "File" -> "Save" or press "Ctrl/Cmd + s") and resume later if you don't want to complete it all at once.

4. While working on the assignment, keep the following in mind:

    - The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks**.
    - **Do not add or delete cells from the notebook**. You may add new cells to perform scratch computations, but you should delete them before submitting your work.
    - **Run all cells, and do not clear out the outputs, before submitting**. You will only get credit for code that has been run.

5. Once you have completed a notebook, download the completed `uniqueid_PS2.zip` file, which is generated from your last cell of the `convolutional_networks.ipynb` file. Submit this to Canvas. Note that only one person from the group will need to do this.

6. You may complete the assignment individually or with a partner (i.e., maximum group of 2 people). If you worked with a partner, provide the name of your partner in the `*.ipynb` file. **We will be using MOSS to check instances of plagiarism/cheating.**

7. **Double check your submission (in particular, that all outputs are printed in the notebook in the zip file) before submitting!** We will **NOT** regrade your assignment because of mistakes due to not double checking your work.

The goal of this assignment is for you to learn how to build a convolutional neural network so that you can

- Develop a deep understanding of convolution, pooling, batch normalization, and initialization.
- Gain experience using forward function and backward propagation for convolutional neural networks.

The notebook `convolutional_networks.ipynb` will walk you through those pipelines in PyTorch. You are required to write code on `convolutional_networks.py`.[1]

## [OPTIONAL] Extra credit short answer problems [up to 0.5% added to final class score]

Save your answers into a PDF, and submit it together with your code to Canvas.

1. We learned about activation functions such as the relu function, and regularization techniques such as dropout. When designing your own neural network, the order in which these components appear matters.

   Prove that it is not the case when using dropout and relu. In other words, show that

   $$dropout(relu(x)) = relu(dropout(x)) \quad \forall x \in \mathbb{R}^d$$

2. Given an input tensor of shape $X \in \mathbb{R}^{C \times m \times n}$, we want to perform AveragePool2D operation with a kernel size of (k, k). For the sake of simplicity, assume that m and n are divisible by k.

   We can actually implement this ourselves using Conv2D operation. Write down the convolution kernel that we need to use to perform AveragePool2D with Conv2D. You must include the dimensions and the values of its weights.

[1] The coding assignment is adapted from Stanford CS 231n.