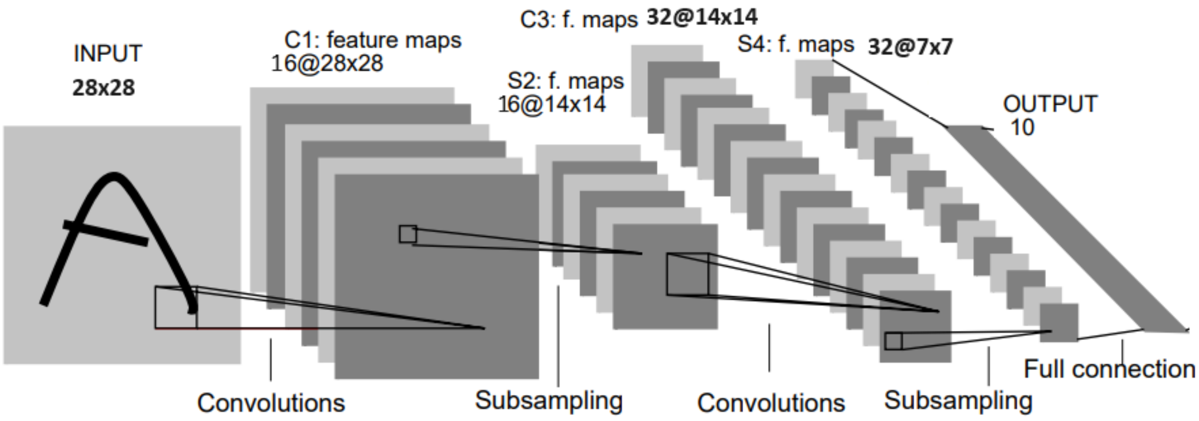
CSCI335 Machine Learning

Homework 12

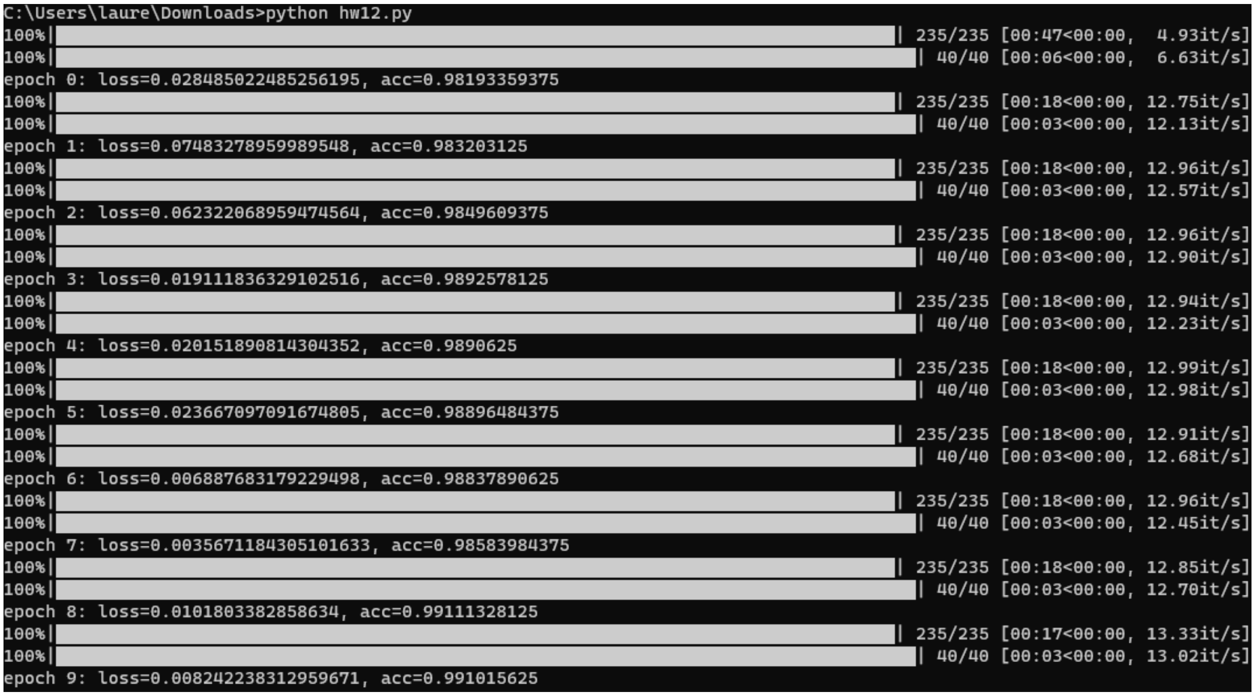
1 (1pt). **Download** MNIST dataset from https://www.kaggle.com/datasets/playlist/mnistzip/

2 (2pt). **Create** Train and Validation **Dataset** and DataLoader objects.

3 (6pt). Define **the model** as in the image below. Convolution stands for: Conv2d with kernel size 5, stride 1 and padding 2, BatchNorm2d, and ReLU. Subsampling is MaxPool2d. I didn't use activation after the last max-poolong.

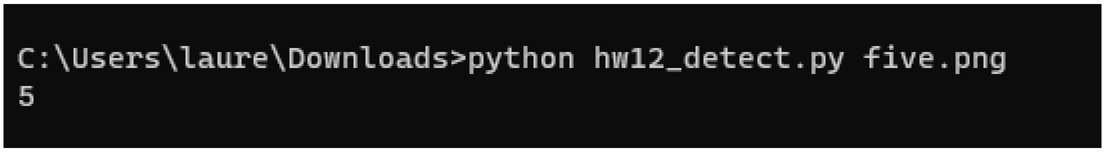


4 (5pt). Train the model and save the weights. The **accuracy** on the validation set should be > 0.99. You may want to use GPU, it took 3 minutes to train it on GPU. It can be Jupyter file or Python file.



```
C:\Users\laure\Downloads>python hw12.py
100%|████████████████████████████████████████| 235/235 [00:47<00:00,  4.93it/s]
100%|██████████████████████████████████████████| 40/40 [00:06<00:00,  6.63it/s]
epoch 0: loss=0.028485022485256195, acc=0.98193359375
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.75it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.13it/s]
epoch 1: loss=0.07483278959989548, acc=0.983203125
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.96it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.57it/s]
epoch 2: loss=0.062322068959474564, acc=0.9849609375
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.96it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.90it/s]
epoch 3: loss=0.019111836329102516, acc=0.9892578125
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.94it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.23it/s]
epoch 4: loss=0.020151890814304352, acc=0.9890625
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.99it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.98it/s]
epoch 5: loss=0.023667097091674805, acc=0.98896484375
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.91it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.68it/s]
epoch 6: loss=0.006887683179229498, acc=0.98837890625
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.96it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.45it/s]
epoch 7: loss=0.0035671184305101633, acc=0.98583984375
100%|████████████████████████████████████████| 235/235 [00:18<00:00, 12.85it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 12.70it/s]
epoch 8: loss=0.0101803382858634, acc=0.99111328125
100%|████████████████████████████████████████| 235/235 [00:17<00:00, 13.33it/s]
100%|██████████████████████████████████████████| 40/40 [00:03<00:00, 13.02it/s]
epoch 9: loss=0.008242238312959671, acc=0.991015625
```

5. (5pt) Write a **Python script** that can be run on the input image and returns the prediction of the hand-written digit. The output should look like this:



```
C:\Users\laure\Downloads>python hw12_detect.py five.png
5
```

(to read from command line use sys.argv)

6 (1pt). Run your script on the images from HW 11. Comment on the performance in the notebook or Python script.

7. **Submit** both training and detection files with saved model weights and the test images in one archived file or as separate files.