

## CS 5970/6970

Fall 2025

### Assignment 4 – Build, Improve, and Pretrain: CNNs, Residuals, and SSL

Due: 11/21/2025 11:59 pm

Design a small CNN from scratch (5–10 layers), train it on CIFAR-10, then upgrade it with residual connections and compare. Finally, pretrain one of your models with a self-supervised objective and fine-tune on labels. You'll analyze accuracy, data efficiency, convergence, and learned features.

#### Task 1: Small CNN (from scratch)

[15 Points]

**Goal:** Implement and train a compact CNN classifier on CIFAR-10 with the following specifications:

- 5–10 learnable layers (conv/normalization/activation/linear count toward depth; pooling doesn't).
- $\leq 1.5\text{M}$  parameters (show a param count).
- Use standard training (cross-entropy) and reasonable augments (random crop/flip, optional color jitter).
- Train to reasonable test accuracy (don't overtrain; ~20–30 epochs or early stopping is fine).

#### Deliverables

- Model diagram (text or figure), param count, training curves.
- Test accuracy and confusion matrix.

#### Task 2: Improve your CNN

[10 Points]

**Goal:** Modify your CNN to include identity (skip) connections where shapes match (e.g., two convs per block), and projection when they don't ( $1\times 1$  conv). Make sure to follow the below specifications:

- Keep overall depth and parameter budget comparable ( $\pm 20\%$  is fine).
- Reuse the same optimizer/augments unless you justify changes.
- Train and compare against Part A.

#### Deliverables:

- Residual block design (1–2 sentences/figure).
- Training curves overlay (A vs. B), final test accuracy.

- Brief analysis: optimization stability (loss smoothness), convergence speed (epochs to 90% of final acc), and any generalization differences.

### **Task 3: Self-Supervised Pretraining + Fine-tuning [25 points]**

**Goal:** Pretrain either your vanilla CNN or residual CNN without labels, then fine-tune on labeled CIFAR-10, with the following specifications:

- Choose one SSL strategy from the below list:
  - Predictive: Rotation prediction (0/90/180/270) or patch jigsaw.
  - Contrastive: SimCLR-style instance discrimination (two strong views).
  - Masked modeling / reconstruction: Mask random patches and predict pixels/features.
- Follow the below training protocol:
  - Pretrain on the entire unlabeled CIFAR-10 train set (labels hidden).
  - Evaluation B (Fine-tune): Unfreeze and fine-tune end-to-end with 10% labeled data (make sure to sample uniformly across classes!).

#### **Additional Deliverables for this section:**

- Brief description of your SSL objective and augmentations (1 paragraph) and a discussion of it helps to train it with SSL or not.

#### **Deliverables:**

1. An IPYNB notebook that can run on Google Colab.
  - a. You can use the same notebook for each part. Make sure to label them correctly with a description using a text cell.
2. Your code that you used to perform all three tasks.
  - a. Provide proper documentation on how to run your code, any dependencies, etc. as a text cell *before* your code cell.
3. All your data that is needed to run and reproduce your code and analysis. You can embed the link as Google Drive folder or upload the images with your code in a single ZIP file.
4. A description of your approach, what worked and what didn't and GenAI usage declaration, and a statement of contribution for each member in your team. If it is an individual project, then just say "This was an individual project and I did everything in it."