Graded Assignment 1

Self-Supervised Learning and OOD Generalization

Prepared by Abdulkadir Gokce

Overview

Introduction. In this assignment, you will **implement and explore self-supervised learning** on a downsampled subset of the ImageNet-1k dataset, and evaluate how well your model generalizes **both in-distribution and out-of-distribution (OOD)**. The assignment builds directly on concepts covered in the homeworks and exercise sessions.

Due date: Wednesday 8th October, 2025, 23:59 CET
Compute platform: http://gnoto.epfl.ch

Section 1. Dataset Setup.

You will work with three datasets, including

1. **training dataset:** 200 ImageNet classes, 500 images each (100k total),
2. **validation dataset:** 200 ImageNet classes, 50 images each (10k total),
3. **OOD dataset:** 200 unseen classes, 50 images each (10k total).

Important!

- Use the OOD dataset only for **evaluation**, never for training.
- During grading, your model will be tested on a ***different OOD dataset***, so avoid overfitting to the provided one.

Section 2. Model Implementation.

Load your model from `models.py`. Implement

- an **encoder** network for image representation,
- a **projection head** for the self-supervised loss.

Do not change the input/output dimensions of the provided `ImageEncoder` template. You may use existing architectures, but you must train your models from scratch on the provided datasets. Random checks will be performed.

Section 3. Training and Loss.

Implement

- a `training_step` function,
- an `evaluation_step` function,
- a `custom_loss_function`.

Train your model while saving checkpoints and evaluating periodically.

Section 4. Visualization.

To better understand your model, visualize

- training and validation loss curves,
- sample validation images with predicted labels,
- (*optional*) embedding visualizations using t-SNE or UMAP.

Compare in-distribution and OOD performance.

Section 5. Submission Instructions.

You must submit

- `models.py`: your model implementation,
- `final_model.safetensors`: trained model weights,
- `CS461_Assignment1.ipynb`: completed notebook,
- `report.md`: short report describing your approach (max. 500 words).

by copying your files to `/home/cs461_assignment1_submission`. We will scrape them after the deadline.
Before submission, verify your folder with:

```
python eval.py
```

Grading. The approaches are evaluated using k-NN and linear probes on both the public and private OOD datasets, with each contributing equally to the overall score.

The grades are determined using the following scheme:

- **4.0**: Your code runs and produces a meaningful output.
- **5.0**: Your model performs better than the baselines in Code Notebook 2 (Week 3).
- **6.0**: Your model achieves a substantial improvement over the Week 3 baselines, showing a clear margin of gain on the evaluation metric.

Good luck! And please, double-check your submission!