DATA 37000 – Final Project

CONVOLUTIONAL NEURAL NETWORKS ON A REAL-WORLD DATASET

Autumn 2025

OBJECTIVE

The final project gives students hands-on experience applying neural network techniques—covered in class—to a real-world image dataset. Students will:

1. Select and justify an image dataset of appropriate scale.
2. Build, train, and evaluate one or more neural-network models.
3. Compare architectures (baseline vs improved NN).
4. Summarize findings in a clear, professional report in your GitHub repository.

Students may choose one of the following three project types:

1. Image **Classification** using CNN c.f. MLP (e.g., ResNet18 fine-tuning)
2. Object **Detection** (e.g., YOLOv8n/s or Faster R-CNN)
3. Image **Segmentation** (e.g., U-Net or YOLOv8-seg)

DATASET REQUIREMENTS

Students must choose one of the following official dataset sources for the final project:

1. **Recommended: Open Images Dataset V4 (Google)**
   A large, high-quality collection of real-world images with labels for classification, detection, and segmentation.
   Students may select a subset (e.g., 5–15 classes) to keep training time manageable.
   Requirements:
   a. Minimum: ~1,000 total images (after choosing your subset)
   b. At least 100 images per class recommended
   c. Must document the subset and classes selected

2. **Optional: ImageNet (via the "31 Free Image Datasets" List)**
   This includes available ImageNet-style datasets from open sources (e.g., Kaggle mirrors, curated subsets, academic mirrors).
   Students must work with legally accessible, properly licensed subsets — not restricted ImageNet downloads.
   Requirements:
   a. Minimum: ~1,000–2,000 total images (subsets allowed)
   b. Minimum ~100/class
   c. All source links and licenses must be cited. ImageNet subset licenses vary by mirror — you must cite the license of your exact source.

General Rules for both options above:

1. No MNIST, Fashion-MNIST, CIFAR, or other toy datasets.
2. No proprietary or private datasets without documented permission.
3. All datasets must be real-world images in JPG/PNG format.
4. Students must clearly document:
   a. dataset source
   b. classes/categories chosen
   c. number of images per class

OPTIONAL (TEAMS)

You may work in teams of 2–3 students. Same rule as the midterm:

- You must use a GitHub repo with visible commit history from all members.
- The github repo should show history of contribution from all members in substantial way.
- If contributions are unbalanced or unclear, the project may receive 25% points taken off.

ANALYSIS REQUIREMENTS

Each project must include the following components.

1. **Problem Statement**
   Explain:
   a. Why this dataset was chosen
   b. Real-world relevance
   c. What the model should achieve (classification, detection, segmentation)

2. **Data Exploration (EDA)**
   Show:
   a. Examples of images
   b. Class distribution (imbalances must be discussed)
   c. Image sizes, quality issues
   d. Annotation format (for detection/segmentation)

3. **Baseline Neural Network**
   Choose one depending on your project type:
   a. Classification: Simple CNN (compare to MLP with comparable parameters if justified, e.g., flattened simple grayscale images)
   b. Object Detection: Pretrained YOLOv8n or torchvision Faster R-CNN with minimal tuning (*You must use a GPU for YOLO training; CPU-only training will not finish*.)
   c. Segmentation: Basic U-Net or YOLOv8-seg

   In all three cases, must also show (i) Architecture summary (ii) Training logs, (iii) Learning curves, and (iv) Initial evaluation metrics.

4. **Improved Model:**
   Improve upon your baseline using at least one of the following:
   a. Transfer learning (e.g., ResNet18 fine-tune)
   b. Deeper network architecture
   c. Data augmentation
   d. Dropout / BatchNorm
   e. Learning rate scheduling
   f. Hyperparameter tuning
   g. early stopping
   h. better optimizer (AdamW, SGD+momentum)
   i. training longer with LR schedules
   j. Better annotation strategy (for detection)

5. **Model Evaluation:**
   a. Discuss overfitting
   b. Explain why the improved model performs differently (or not differently)

c. Compare baseline vs improved NN
d. For **Classification**: Accuracy; Confusion matrix; Per-class metrics; Train/validation curves
e. For **Detection**: Precision / Recall; mAP@50 (or similar metric); Example predictions
f. For **Segmentation**: IoU / Dice score; Overlay visualizations of predictions

6. **Discussion & Interpretation**
   Reflect on:
   a. What worked
   b. What failed
   c. Ethical, practical, or deployment considerations
   d. Limitations of your dataset or labels
   e. How you would improve the system with more time

DELIVERABLES

1. GitHub Repo (can be private, need to add "**physicsland**" and TA "**jiawei-zhang-a**" as collaborators)
   a. When you first create the repo, make sure you have folders like "bigdata", "private", etc, on your .gitignore file . (Use the .gitignore for our class as template.)
   b. Should have coding work in .py files, as well as results/summary on the repo.
   c. The summary of your project can be presented in .ipynb file, or .md (or any variants), pdf, docx formats. It/They should be in a folder named "results".
   d. Make sure you organize your repo in a sensible way. Codes in "src" folder, and graphs/charts in "media" folder. These are just examples. You don't need to use these traditions. But organization is needed.
   e. As mentioned before, if this is team work, every member needs to be contributing substantially as seen in the commit history. Branches, commits, and all the proper git usage will be examined.
2. On canvas, submit your GitHub repo link. Also indicate the names of your teammates if applicable.
3. Make sure proper citations are included in the summary report. (APA or other common styles are accepted.)

GRADING RUBRIC

Your project grade will be determined by:

✓ Dataset selection & documentation (10%)
✓ Problem statement and EDA (15%) – Completeness, correctness, and insight.
✓ Coding, modeling – Correct use of algorithms; reproducibility (25%)
✓ Evaluation & comparison – Appropriate metrics; fair analysis (20%)
✓ Discussion & interpretation – Clarity, reasoning, reflection on results (25%)
✓ Repo organization – Readability, formatting, professionalism (5%)

GUIDELINES FOR USING GENERATIVE ARTIFICIAL INTELLIGENCE (FROM SYLLABUS)

Students are welcome to use AI tools (e.g., ChatGPT, Copilot, Gemini, Claude, Deepseek, DALL·E, etc.) in this course under the following conditions:

- **Transparency**: Any use of AI must be acknowledged and cited, just as you would any other source.
- **Complement, Not Substitute**: AI tools can help brainstorm, debug, or illustrate, but they must not replace your own analysis or critical thinking.
- **Integrity**: Submitting AI-generated work as your own without attribution is considered plagiarism under the University's academic integrity policy.
- **Permission Contexts**: Some assignments will allow AI use (default in this class); others may restrict it to encourage your independent skills. Follow the instructions for each case.

In short: use AI as a collaborator, not a ghostwriter.