EEL5840 Fall 2025
Fundamentals of Machine Learning
Final Project E: Simple Object Tracking

Project E: Simple Object Tracking

Project Due: December 3, 2025, 4:00 PM
Poster Session & Live Demo: December 3, 2025, 4:00 PM - 6:30 PM EST in Robotics High Bay Lab MALA#1030 or recorded video for online/EDGE students
Material Due: poster, presentation, live demo of code, and code implementation deployed to a GitHub repository

1. Description

In this project, you will develop a machine learning system to track a moving object across video frames. Possible categories include: ball, mug, pen, spoon, and notebook. Each group will be responsible for collecting videos in multiple environments using a phone or laptop camera, and the video annotations.

2. Data Set

Each team must record at least 100 videos, each 3-second long, and the object annotations within each video. We recommend collecting evenly number of samples per class, in this case, 20 videos per class (ball, mug, pen, spoon, and notebook). Videos should be stored in *.mov* or *.mp4* format and video frames must be annotated for object label and object location (bounding box). After data collection, all teams' data will be merged and split into training and test sets.

The **moving object labels** are: (1) ball, (2) mug, (3) pen, (4) spoon, and (5) notebook. Each team must collect 100 videos, 20 per class. We recommend you to save your files using a coding system, e.g. **ID-trial-label**. First give a number from 1 to 3 to each team member, this is the ID. Then, for example, when team member with ID 2 is recording hers/his/their 3rd video of a moving spoon, the file name should read "2-3-4.mov".

After collecting all teams' recordings, we will split the data at random into a **training set** and an ***easy* test set**. You will be given the **training set** to train your ML system. We will hold the ***easy* test set** until after you submit your code implementation. This test set will be used for grading, and during your live demo.

We will also create a separate ***hard* test set** that includes out-of-class gestures (e.g., person, car, etc.). This test set will be used for **extra credit contest** - see details below.

Once your team collects the data, you should use the Notebook file "Final Project - Project E Data Collection.ipynb" to store your data and annotations in a pre-defined format. Follow all the instructions in the Notebook file.

You will use `https://www.makesense.ai/` for data annotations (object labeling and localization). Follow the instructions in the Notebook file "Final Project - Project E Data Collection.ipynb".

3. Project Poster, Presentation, and Live Demo Guidelines

Your project will be showcased through three components: a **poster**, an **oral presentation**, and a **live code demonstration**. Together, these should communicate your approach, methodology, and findings clearly and professionally.

3.1. Poster Content and Format

The project should be presented as a single poster measuring **48 inches in height by 36 inches in width**. The poster should follow the structure and tone of an IEEE Transactions report, adapted for a clear and visually engaging single-page format. All text must be written in clear, precise, and grammatically correct English. A digital PDF version of the completed poster must be submitted by the specified deadline.

For full credit consideration, your poster should include the following sections:

- *Abstract.* A short summary of your project's purpose, approach, and key findings (3–5 sentences).
- *Introduction & Background.* Briefly describe the problem, motivation, and any relevant prior work. Include properly formatted references (IEEE style).
- *Implementation.* Explain your technical approach, algorithms, or model structure. Use pseudo-code, equations, or flow diagrams for clarity. If you participated in the extra credit contest, describe your strategy for identifying unknown classes.
- *Experiments & Results.* Describe your dataset, training/testing procedures, and performance metrics. Summarize your results with plots, tables, and concise commentary. This section carries the most weight in grading, show clear, well-designed experiments.
- *Conclusions.* Summarize key takeaways supported by your results. Avoid unsupported claims or copied text from references.
- *References.* List any cited works in IEEE format.

3.2. Presentation and Live Code Demo

During the **poster presentation**, each group will present their poster in a 5-minute oral presentation, highlighting their main contributions, experimental design, and key findings. Presenters should be prepared to clearly explain their approach and to answer questions about their methods, results, and conclusions.

For the live code demo, students will first present their training notebook, briefly explaining the key steps taken to train their final model. Following this, they will be provided with a blind test set and must run their test notebook in real time to evaluate their model's performance. The live code demonstration should also take no longer than 5 minutes, including the live evaluation of the provided blind test set.

All group members are expected to actively participate in the presentation and/or the live code demonstration.

4. Project Code Implementation

You may implement your ML system yourselves or using standard packages/libraries. You can use any packages that come as a default option with Anaconda, TensorFlow, or PyTorch. But, your team must train your ML model from scratch on the provided dataset, i.e. **the use of pre-trained models or transfer learning is not allowed**.

You are welcome to carry any of the following experiments:

- Add more data,
- Apply any data augmentation strategy,
- Design your own model architecture,
- Apply any pre-processing strategy.

Your final code submission should contain 3 files:

- README file - directly edited in your GitHub team repository. A template will be provided.
- train.py or a Notebook (.ipynb) with a function "train". This function should contain the code used to train your final model.
- test.py or a Notebook (.ipynb) with a function "test". This function should load data and labels (in the same format as the training data) and output the predicted labels and a metric score value.
- if you compete in the contest, you can create a separate file for testing on the hard test set (or include it in test.py). This function should receive data and labels (in the same format as the training data) and output the predicted labels and a metric score value.

5. Grading Details

Your grade will be determined using the following rubric:

- 1% Team formation
- 9% Data collection

  - Each team is required to collect 100 videos according to the specifications described in Section 2 and in the Notebook. Grades will be based on data quality, adherence to format requirements, and completeness of the dataset.

- 25% Implementation & Live Demo
  - This component evaluates the functionality and clarity of your implementation, as well as your ability to run and explain it during the live demo.
  - Submitted code must run correctly and easily, with a clear README, well-documented dependencies, and adjustable parameters.
  - All required packages and libraries must be listed, and teams should verify compatibility with the provided computing environment before the due date.
  - During the 5-minute live code demo, teams must:
    * Present their training notebook and explain key steps used to train the final model.
    * Run their test notebook on a blind test set to evaluate performance in real time.
  - All team members must actively participate in either the presentation or the live demo

- 25% Average Intersection-over-Union (IoU) on "easy" blind test data set, computed over the correctly classified samples.
  - Performance will be evaluated using the IoU on the held-out "easy" blind test set.
  - Your code should generate class label and bounding box locations for each test video.
  - Full credit will be awarded for achieving a avverage IoU over correctly classified samples of ≥ 0.55, or at least matching the class average, whichever is lower.

- 40% Poster Presentation
  - Your poster presentation will be evaluated for clarity, completeness, and visual communication of your work.

- The poster must follow the structure outlined in Section 3 and meet formatting and content requirements (Abstract, Introduction, Implementation, Experiments, Conclusions, and References).
- During the 5-minute presentation, teams should clearly explain their contributions, experimental design, and results.
- Grading will consider clarity of explanation, visual organization, experimental soundness, and team participation.

6. Extra Credit Contest

The goal of this project is to implement an end-to-end machine learning system to track objects across frames. The teams with the best average IoU on the "hard" data set will earn **extra credit**. The "hard" data set will also have all 5 labels plus an unknown label (-1). You will want your model to be robust enough to identify out-of-class samples.

7. Submission Details

Turn in your project poster and your code on your group GitHub repository by **Wednesday, December 3rd at 4:00 PM**. In Canvas, you should submit your GitHub URL.

Be sure your repository contains the following files: train.py, test.py, README.txt, and any saved models that will be needed when demoing your work with the test.py file.