Department of Electrical and Computer Engineering
North South University (NSU)

CSE 440: Artificial Intelligence
Section 05

Project

Total Marks: 100
Proposal (20) + Execution (60) + Presentation (20)

Instructor: Dr. Mohammad Mahmudul Alam Semester: Spring 2025

Title:	Real-Time Object Detection Using YOLOv8
Name:	Hasnat Karibul Islam (Alindo)
NSU ID:	2211275042
Name:	Md. Minhajul Islam
NSU ID:	2211022042

Proposal

Title:

Real-Time Object Detection Using YOLOv8.

Introduction:

Object detection is a basic computer vision problem with its use in security, autonomous vehicles, surveillance, and intelligent applications. YOLO (You Only Look Once) is a state-of-the-art deep learning object detection model that is highly recognized for being both fast and accurate. The objective is to implement real-time object detection using YOLOv8, which is one of the latest models in the YOLO line, to identify and classify objects within video streams effectively. The project will lead to AI-powered surveillance and automation innovation, with the precision of detection and processing time being enhanced.

Problem Statement:

The increasing demand for real-time object detection presents serious challenges rooted in computational inefficiencies and accuracy trade-offs. Traditional object detection methods are typically not able to strike a balance between accuracy and speed, and this makes them unsuitable for real-time use cases like security surveillance, traffic monitoring, and autonomous systems. The goal of this project is to develop a high-performance real-time object detection system using YOLOv8 that addresses these challenges by optimizing model performance and deployment.

Objectives:

- Develop a real time object detection system using YOLOv8 model.
- Improve detection accuracy through model fine-tuning and dataset augmentation.
- Optimize the model for efficient real-time processing.
- Evaluate the system's performance in real-world scenarios and refine it based on findings.

Methods:

- **Dataset Collection & Preprocessing:**
  Utilize existing datasets like COCO, Pascal VOC, or custom datasets for model training.
- **Model Selection & Training:** Implement YOLOv8 using a deep learning framework like Ultralytics' YOLO library with PyTorch.
- **Fine-Tuning & Optimization:** Adjust hyperparameters and use techniques like transfer learning to enhance detection accuracy.
- **Real-Time Implementation:** Integrate OpenCV for video processing and implement real-time object detection in live video streams.

Datasets and Tools:

- **Datasets:** COCO, Pascal VOC, or a custom dataset.
- **Tools & Technologies:**
    - Python, PyTorch, Ultralytics YOLOv8
    - OpenCV for video processing
    - GPU acceleration (NVIDIA CUDA / Google Colab(T4 GPU))

Project Update

Title:
Real-Time Object Detection Using YOLOv8

Abstract:
The goal of this project is to create a real-time object detection system with YOLOv8. The goal is to improve accuracy without sacrificing high-speed processing, which can be applied to applications like surveillance and automation. This report outlines the progress achieved, methodologies used, preliminary results, and the way forward to completion.

Methodology:

- **Environment Setup:**
  - Installed necessary dependencies including Python, PyTorch, and Ultralytics YOLOv8.
  - Verified GPU compatibility and configured CUDA for accelerated training.
  - Set up Jupyter Notebook and other development tools for experimentation.
- **Model Exploration:**
  - Studied YOLOv8 architecture, including improvements over previous versions.
  - Tested pre-trained YOLOv8 models on sample images to understand initial performance.
  - Compared different YOLOv8 model sizes (nano, small, medium) for trade-offs in speed and accuracy.
- **Data Preprocessing:**
  - Not yet started; will begin with dataset annotation and formatting in the next phase.

Initial Results:

- Completed the setup of the training environment.
- Conducted preliminary tests with pre-trained YOLOv8 models on sample image

Next Steps:

- Train YOLOv8 on the selected dataset and fine-tune model parameters.
- Implement real-time object detection in video streams using OpenCV.
- Optimize the model for faster inference and improved accuracy.
- Prepare documentation and presentation materials.