A picture containing graphical user interface

Description automatically generated

Project 25-344 and 4 Sensor Data Fusion and Algorithm Analysis

Project Proposal

Prepared for

James Perea/Nibir Dhar

U.S. Department of Defense

By

Grace Gillam , Paul Reid, Jeffrey Weaver , David Anthony

Under the supervision of

Changqing Luo

Date

5/8/2025

**Executive Summary**

This project investigates whether combining LiDAR and RGB camera data can improve pedestrian detection performance in real-world driving environments. The primary goal is to assess the effectiveness of multi-modal sensor fusion—specifically integrating LiDAR point clouds and camera images—compared to single-sensor (camera-only) approaches. The work supports advancements in autonomous vehicle safety, where timely and accurate pedestrian detection is critical.

To meet this goal, the team designed and implemented a deep learning architecture consisting of two specialized feature extraction pipelines: EfficientNet-B0 for processing camera images and a PointNet-style encoder for LiDAR data. The extracted features are fused in a joint decision network that outputs predicted pedestrian bounding boxes and confidence scores. A parallel baseline model was developed to simulate camera-only detection by substituting the LiDAR input with placeholder tensors.

The project uses the Waymo Open Dataset, which provides synchronized LiDAR and camera data in .parquet format. A custom dataset class extracts aligned RGB images, LiDAR point clouds, and pedestrian bounding boxes. The training pipeline, built in PyTorch, supervises the model using Smooth L1 loss on predicted bounding boxes and includes support for saving an “alarm file” containing prediction results and confidences to generate ROC and precision-recall curves for evaluation.

Work to date includes successful model architecture implementation, data preprocessing and synchronization logic, and test runs using SLURM on VCU’s Hickory high-performance cluster. Early experiments revealed several issues—such as inconsistent image modality usage and untrained confidence scores—which have since been resolved. The project has now progressed to the stage of final model training and inference testing, with ongoing work focused on quantitative evaluation and visualization of results.

The final deliverables include trained detection models, alarm files for evaluation, and a written report comparing fusion-based and camera-only detection Ultimately, this research contributes insights into how sensor fusion impacts pedestrian detection performance and may guide future design choices in autonomous vehicle perception systems.

**Table of Contents**

[Section A. Problem Statement 5](#_heading=h.safv1vjs08t)

[Section B. Engineering Design Requirements 7](#_heading=h.3zmjknsame1n)

[B.1 Project Goals (i.e. Client Needs) 7](#_heading=h.6h9ovnw4dgo)

[B.2 Design Objectives 7](#_heading=h.qvmh37qm5df4)

[B.3 Design Specifications and Constraints 8](#_heading=h.dym2kqu4zzhb)

[B.4 Codes and Standards 9](#_heading=h.grj4grnv2tgm)

[Section C. Scope of Work 11](#_heading=h.csow9p53siwp)

[C.1 Deliverables 11](#_heading=h.f8xjrnnpfgi)

[C.2 Milestones 12](#_heading=h.tg9rggarukbq)

[C.3 Resources 12](#_heading=h.qkhfazqve6g5)

[Section D. Concept Generation 13](#_heading=h.u5cmyshlw8ry)

[Section E. Concept Evaluation and Selection 14](#_heading=h.v6crfywkamc9)

[Section F. Design Methodology 16](#_heading=h.7w4j62h5b5aj)

[F.1 Computational Methods (e.g. FEA or CFD Modeling, example sub-section) 16](#_heading=h.tgdocthhnp82)

[F.2 Experimental Methods (example subsection) 16](#_heading=h.6hl7v97ul51v)

[F.5 Validation Procedure 16](#_heading=h.h79qq03mv864)

[Section G. Results and Design Details 18](#_heading=h.9v4wlm1sdjx0)

[G.1 Modeling Results (example subsection) 18](#_heading=h.4nvujrvnn3bh)

[G.2 Experimental Results (example subsection) 18](#_heading=h.swcrt9hy6037)

[G.3 Prototyping and Testing Results (example subsection) 18](#_heading=h.g0yffm95git0)

[G.4. Final Design Details/Specifications (example subsection) 18](#_heading=h.8na8h2z2bnw6)

[Section H. Societal Impacts of Design 20](#_heading=h.ehkwj2oqjch5)

[H.1 Public Health, Safety, and Welfare 20](#_heading=h.k1gvu8y7nv7v)

[H.2 Societal Impacts 20](#_heading=h.vfga5akc4ozj)

[H.3 Political/Regulatory Impacts 20](#_heading=h.4ozcvdjlspfe)

[H.4. Economic Impacts 20](#_heading=h.g72z8hsijxkg)

[H.5 Environmental Impacts 21](#_heading=h.akzw7qethmzf)

[H.6 Global Impacts 21](#_heading=h.dc5czjlc4ivx)

[H.7. Ethical Considerations 21](#_heading=h.wruwf74xdqej)

[Section I. Cost Analysis 22](#_heading=h.pz3zro1mcowq)

[Section J. Conclusions and Recommendations 23](#_heading=h.4ynvev3hyimk)

[Appendix 1: Project Timeline 24](#_heading=h.pgpt7aih6rhk)

[Appendix 2: Team Contract (i.e. Team Organization) 25](#_heading=h.opfgrpthpm0r)

[Appendix 3: [Insert Appendix Title] 26](#_heading=h.5745k5hkhvcn)

[References 27](#_heading=h.tedg43i9g798)

### Section A. Problem Statement

In recent years, the development of autonomous vehicles (AVs) has emerged as one of the most impactful areas in modern transportation and robotics. A critical component of AV technology is the reliable detection and localization of pedestrians in real-world environments. Ensuring the safety of vulnerable road users is not only an engineering challenge—it is also a societal, economic, and ethical imperative. Despite extensive research in computer vision, pedestrian detection remains one of the most challenging perception tasks due to factors like occlusion, motion blur, variable lighting, and unpredictable human behavior.

The specific problem this project addresses is the limitation of using a single sensor modality—such as an RGB camera alone—for pedestrian detection in complex urban environments. While camera-based systems are widely used due to their affordability and visual richness, they struggle with depth perception and performance under poor lighting conditions (e.g., at night or during inclement weather). LiDAR sensors, on the other hand, provide accurate spatial measurements but lack semantic detail. Consequently, systems that rely on a single type of sensor may suffer from reduced reliability and safety risks in edge cases, where misdetections could result in injury or death.

This project aims to explore **sensor fusion**, the integration of RGB and LiDAR data, as a solution to these challenges. The goal is to determine whether fusing information from both sensors leads to improved detection performance in terms of localization accuracy, confidence estimation, and robustness under variable conditions.

The problem is faced by automotive manufacturers, autonomous vehicle startups, robotics researchers, and city planners worldwide. According to the National Highway Traffic Safety Administration (NHTSA), pedestrian fatalities in the United States have risen by more than 60% over the past decade, reaching over 7,500 deaths in 2022 alone—accounting for 17% of all traffic deaths (NHTSA, 2023) [1]. This trend underscores the urgent need for technological solutions that can prevent pedestrian-related accidents, particularly in high-density areas with complex visual scenes.

The potential costs of failing to address this problem are significant. Economically, pedestrian crashes lead to millions in hospital costs and insurance claims. Societally, public trust in AVs depends heavily on safety performance. From a health and safety standpoint, even a single misdetection can have catastrophic consequences. Improving perception systems through sensor fusion may thus have wide-reaching benefits in terms of reducing fatalities, enabling safer transportation systems, and improving public confidence in AV technologies.

This project falls under the domain of **autonomous systems and computer vision**, intersecting with the fields of deep learning, robotics, and intelligent transportation. It is academically sponsored and builds upon research in the Computer Science Department at Virginia Commonwealth University. The architecture combines state-of-the-art methods in image recognition (EfficientNet-B0) and point cloud processing (PointPillars-inspired encoding) to study multimodal feature fusion.

Commercial AV companies like Waymo and Cruise already deploy LiDAR-RGB fusion in production, though their architectures remain proprietary. By using the open-access Waymo Open Dataset, this project aligns with current research standards and contributes openly replicable results to the field.

**Project Objectives**

The primary objectives of the project are:

* To implement a dual-encoder model that fuses RGB and LiDAR features.
* To train and compare it against a camera-only baseline.
* To assess detection quality using bounding box accuracy, classification confidence, and ROC/precision-recall curves.
* To create reusable code and visualizations for future research or industry use.

In doing so, the project advances both educational and technical understanding of sensor fusion in perception pipelines.

### Section B. Engineering Design Requirements

This section outlines the core engineering requirements that guide the development of the pedestrian detection system. These requirements are derived from the client’s goals and reflect the technical, functional, and regulatory expectations of the project. Each subsection addresses a specific aspect of the design planning process—from high-level goals and measurable objectives to hard constraints and relevant industry standards. Together, these requirements form the foundation on which all subsequent design decisions are based.

#### B.1 Project Goals (i.e. Client Needs)

The overarching goals of this project are drawn from the real-world needs of autonomous vehicle stakeholders—including vehicle manufacturers, urban planners, and safety regulators—who require accurate, reliable, and robust pedestrian detection to reduce traffic fatalities and improve public confidence in autonomous systems. This project addresses those needs by investigating the effectiveness of multi-sensor data fusion.

The goals of the project include:

* To improve the reliability of pedestrian detection in autonomous vehicle perception systems.
* To explore the benefits of sensor fusion using LiDAR and RGB data compared to using a single sensor modality.
* To contribute to the understanding of deep learning-based object detection models under real-world driving conditions.
* To support future engineering decisions related to sensor configurations and perception architecture in AV platforms.
* To produce actionable experimental results that inform tradeoffs between performance, cost, and complexity.

#### B.2 Design Objectives

The specific design objectives reflect what the system will achieve in a measurable and realistic fashion. These objectives are aligned with the timeline and resources of a senior design capstone project and are designed to be SMART:

* The design will detect pedestrians in RGB images and LiDAR point clouds using a trained deep learning model.
* The design will generate bounding box outputs that achieve at least 0.5 mean IOU (intersection-over-union) compared to ground truth labels.
* The design will output per-prediction confidence scores that are usable in ROC and precision-recall analysis.
* The design will support three operating modes—camera-only, LiDAR-only, and fused—allowing for direct performance comparisons.
* The design will complete training on a subset of the Waymo Open Dataset using under 200GB of disk space and <8 hours of compute per training run.
* The design will visualize predicted bounding boxes alongside ground truth boxes for validation and presentation.

#### B.3 Design Specifications and Constraints

The table below summarizes the technical and operational constraints and specifications guiding the system’s development:

| Constraint Type | Specification / Constraint |
| --- | --- |
| Data Constraint | Input data must be loaded from `.parquet` format using Waymo Open Dataset columns. |
| Hardware Constraint | Model must train on Hickory HPC with 1 GPU (`--gres=gpu:0`) and ≤4 CPUs. |
| Memory Constraint | Training must operate with ≤4GB RAM per task (as per SLURM job specification). |
| Compute Constraint | Total training duration must not exceed 8 hours per run. |
| Output Constraint | Predicted bounding boxes must be output in `[x, y, w, h]` format, with associated confidence scores. |
| Evaluation Constraint | Alarm file must include fields for `pred\_x, pred\_y, pred\_w, pred\_h, num\_gt\_boxes, confidence`. |
| Model Constraint | Fusion model must use EfficientNet-B0 and PointNet-style encoder. |
| Training Constraint | Loss function must be differentiable and operate under Smooth L1 loss for bounding box regression. |
| Data Alignment Constraint | All training samples must be based on synchronized LiDAR and camera files with identical base names. |
| Regulatory Constraint | Dataset usage must comply with Waymo Open Dataset License Agreement (non-commercial use). |

#### B.4 Codes and Standards

The following codes and standards are relevant to the safe, reproducible, and standardized development of this pedestrian detection system:

* **IEEE Standard 829-2008** – Testing documentation standards inform the structure of model evaluation and test reporting.
* **ISO/IEC 27001:2013** – Ensures that data security and privacy are maintained while handling large-scale datasets such as Waymo’s.
* **NIST SP 800-53 Rev. 5** – Guides system integrity and access controls on shared HPC resources (e.g., user permissions on Hickory).
* **OSHA Code 1910 Subpart I** – While not directly applicable, general safety compliance is required when operating near AV hardware, referenced for completeness in system integration contexts.
* **ASME Y14.5-2018** – Tolerancing and spatial reasoning standards underpin interpretation of 3D LiDAR bounding boxes, contributing to bounding box alignment practices.

Where applicable, these standards help ensure the reproducibility, safety, and legal compliance of the design. Any future deployment or integration into commercial AV platforms would necessitate adherence to more stringent automotive ISO standards (e.g., ISO 26262 for functional safety), although such deployment is beyond the scope of this capstone project.

### Section C. Scope of Work

This project explores the integration of LiDAR and RGB data for pedestrian detection in real-world driving scenarios. The scope is limited to software-based research and implementation using pre-recorded data from the Waymo Open Dataset. The team is responsible for designing, training, and evaluating deep learning models that compare single-modality (camera-only) detection with fusion-based approaches.

The project does **not** involve hardware integration, deployment in physical vehicles, or collection of new sensor data. The timeline is bound by the academic calendar, with major reports due in the Fall and Spring semesters, culminating in a public Capstone EXPO presentation in May 2025. The methodology follows a **modified waterfall model**, with clearly defined phases (e.g., data handling, model development, evaluation) but iterative refinement as insights are gained.

Boundaries such as GPU allocation on the Hickory HPC, limits on training runtime, and disk storage quotas are explicitly acknowledged. Project expectations and responsibilities have been reviewed and aligned with faculty advisor input to avoid scope creep and ensure feasibility.

#### C.1 Deliverables

The following are the tangible outcomes the team is responsible for delivering:

* A working sensor fusion model using EfficientNet-B0 and PointNet-style encoders
* A baseline camera-only detection model for comparative evaluation
* A synchronized dataset loader that parses .parquet camera and LiDAR files
* Evaluation pipeline with bounding box visualization, confidence scoring, and ROC/PR curve generation
* SLURM-compatible training scripts for use on the Hickory HPC
* A final alarm file (alarm\_file.csv) containing prediction results and scores
* Visual demonstrations of inference on real Waymo frames
* Project proposal, team contract, and preliminary design report
* Fall and Spring Capstone poster and presentations
* Final design report with documentation and analysis

Potential Risks and Mitigation:

* Access to campus: None of the deliverables require physical presence on campus. All work is remotely executable.
* Remote work support: Software tools (e.g., SSH, VS Code, Conda, PyTorch) are installed and configured for HPC access.
* Vendor delays: No third-party hardware or component orders are required. All dependencies are open-source or pre-installed.

#### C.2 Milestones

| Milestone | Estimated Duration | Target Completion Date |
| --- | --- | --- |
| Dataset exploration and column inspection | 2 weeks | September 2024 |
| Model architecture finalized (EfficientNet + PointNet) | 2 weeks | October 2024 |
| SLURM training script tested on Hickory cluster | 1 week | October 2024 |
| Baseline camera-only model trained and tested | 2 weeks | November 2024 |
| Fall poster and presentation prepared | 2 weeks | December 2024 |
| Fusion model trained with corrected supervision | 3 weeks | February 2025 |
| Evaluation script completed (ROC, PR, alarm file) | 2 weeks | March 2025 |
| Final design report written | 3 weeks | April 2025 |
| Capstone EXPO materials finalized | 2 weeks | April 2025 |
| Public presentation at Capstone EXPO | 1 day | May 2025 |

#### C.3 Resources

The following resources are necessary for successful project execution. All resources are either freely available, provided through institutional access, or compatible with the Capstone project budget. The resource plan accounts for both hardware and software tools, as well as datasets, development environments, and collaboration platforms.

Hardware

- Access to the Hickory High Performance Computing (HPC) cluster with GPU partition  
- Local development laptops used for scripting, testing, and documentation  
- SSH access configuration for remote work and submission to SLURM scheduler

Software

- Python 3.11+, with support for PyTorch, torchvision, pandas, and numpy  
- Waymo Open Dataset tools and PyArrow for .parquet file handling  
- EfficientNet-PyTorch library for image encoding  
- Matplotlib and OpenCV for visualization of predictions and ground truth  
- Conda environment (or venv) for package isolation

Data

- Waymo Open Dataset (camera\_image, lidar, camera\_box, lidar\_box, calibration files)  
- Pre-extracted synchronized samples with pedestrian annotations  
- Column maps and inspection logs for schema reference and debugging

Collaboration and Version Control

- Git for source code management  
- Slack or group email for advisor communication and team coordination  
- Local backups and cloud synchronization for work continuity

### Section D. Concept Generation

To address the challenge of accurate and reliable pedestrian detection in autonomous driving, several design concepts were explored during the early ideation phase. Each concept considers a different approach to combining camera and LiDAR data for object detection. The focus was on exploring how various fusion strategies and architectural components can address the core requirements of localization accuracy, confidence scoring, and computational efficiency.

The ideation process involved team brainstorming sessions, reverse analysis of published research models, and comparative evaluation of fusion strategies from the literature. The following three concepts emerged as the most promising candidates.

**Concept 1: Late Fusion (Parallel Encoders + Concatenation at Decision Layer)**

**Description**:  
This concept uses two separate neural network branches: one for camera data (EfficientNet-B0) and another for LiDAR data (PointNet-style encoder). Each branch independently encodes features from its input modality. The resulting feature vectors are concatenated and passed to a shared fully connected layer that predicts bounding boxes and confidence scores.

**Pros**:

* Modular: camera-only and LiDAR-only systems can be evaluated independently.
* Flexibility: new encoders can be swapped in without redesigning the whole architecture.
* Practical: avoids calibration complexity during feature extraction.

**Cons**:

* May miss cross-modal feature interactions early in processing.
* Fusion only occurs at the final stage; high-level patterns may be misaligned.
* Slightly higher inference latency due to dual processing branches.

**Risks**:

* If one modality dominates (e.g., RGB has richer detail), the fusion head may underutilize LiDAR input.

**Concept 2: Early Fusion (Joint Feature Grid in BEV Space)**

**Description**:  
Convert LiDAR data into a Bird's Eye View (BEV) feature map, then project camera features onto the same grid using calibration matrices. Both feature types are fused spatially before being passed through a shared convolutional network.

**Pros**:

* Enables strong spatial correlation between modalities.
* Useful for downstream object tracking or scene understanding.

**Cons**:

* Requires precise camera–LiDAR calibration (extrinsic and intrinsic parameters).
* Complicated preprocessing and data alignment pipeline.

**Risks**:

* Inaccurate calibration may lead to poor fusion quality.
* May not generalize well to edge cases with significant occlusion.

**Concept 3: Camera-Only Baseline with Simulated LiDAR Input**

**Description**:  
Use only the RGB image pipeline (EfficientNet-B0) and replace the LiDAR input with an all-zeros or all-ones tensor. This establishes a lower bound for performance and provides a benchmark for comparing true fusion gains.

**Pros**:

* Simplest implementation; isolates camera pipeline performance.
* Useful for validating pipeline correctness and model supervision.

**Cons**:

* No true fusion—this is a baseline rather than a complete detection system.
* Cannot leverage depth information from LiDAR.

**Risks**:

* If the fusion model performs similarly to this baseline, it may indicate that fusion is not contributing meaningfully.

**Design Concept Diagram (Verbal)**

* **Concept 1**: EfficientNet → [Feature Vector] ← PointNet → Concatenate → Fully Connected → Bounding Box
* **Concept 2**: LiDAR → BEV Grid → + Projected Camera Features → Shared CNN → Detection Head
* **Concept 3**: EfficientNet + Zero LiDAR → Fully Connected → Bounding Box

### Section E. Concept Evaluation and Selection

To determine the most promising design approach, a structured evaluation was conducted using a weighted decision matrix. Three core concepts—late fusion, early fusion, and camera-only baseline—were evaluated based on selection criteria aligned with client expectations and project constraints. The selection criteria were developed through team analysis and consultation with faculty advisors, considering key concerns such as performance, complexity, robustness, scalability, and feasibility.

Selection Criteria and Metrics

The following criteria and weighting factors were selected to compare the proposed design concepts:

| Criterion | Weight (1–5) | Evaluation Metric |
| --- | --- | --- |
| Detection Performance | 5 | Expected mIOU and classification accuracy |
| Implementation Complexity | 3 | Codebase size, required calibration |
| Robustness | 4 | Generalizability to varied lighting and occlusion |
| Scalability | 2 | Ease of adapting to larger datasets or models |
| Feasibility | 5 | Time/resources required to implement and test |

A weighted scoring system was used to rank each concept from 1 (poor) to 5 (excellent) for each criterion. Scores were then multiplied by the assigned weight and summed to obtain a total score.

Decision Matrix

| Design Concept | Raw Score | Weighted Score | Notes | Selected |
| --- | --- | --- | --- | --- |
| Late Fusion | 90 | 378 | Balanced design; modular and feasible | ✓ |
| Early Fusion | 82 | 346 | High performance but complex and risky |  |
| Camera-Only Baseline | 72 | 312 | Simple and feasible, but low innovation |  |

Based on the decision matrix, the Late Fusion model was selected as the preferred design concept. It offers a strong balance between performance and feasibility while maintaining architectural flexibility. The concept will be refined and implemented in subsequent phases of the project.

### Section F. Design Methodology

Provide a detailed explanation of the methods that will be used to help evaluate, improve, and evolve the design through the iterative engineering design process. Consider that ultimately, the final design must be verified and validated to ensure that it meets all of the previously developed and listed design objectives and specifications. Verification ensures that the design meets all specification, while validation confirms that the design functions as intended such to meet the client’s needs. While it is common for initial design concepts to first be evaluated using simplified design criteria and metrics, the chosen design should be advanced, and later verified, using engineering calculations, computational models, experimental data, and/or testing procedures.

Use this section to describe any underlying physical principles and mathematical equations that govern the design. Provide details of any computer-aided modeling techniques used to evaluate the design including the software used, prescribed boundary conditions, and assumptions. Include a detailed description of any experimental testing methods including required testing equipment, test set-up layout, data acquisition and instrumentation, and testing procedures. If one or more prototypes is to be produced and tested, provide a detailed description of how each will be evaluated.

**Note:** The contents of this section are expected to vary from project to project. Subsections may be appropriate for providing details of analytical, computational, experimental, and/or testing methods. Some potential subsections that may be included in this section are provided. While critical design equations may be provided here, lengthy mathematical derivations may be included in an appendix. Validation procedures are critical and all projects should address such topic.

#### F.1 Sensor Fusion Architecture and Design Rationale

The selected architecture is a late-fusion dual-encoder model consisting of:

* An **EfficientNet-B0** backbone for RGB camera input.
* A **PointNet-style encoder** for LiDAR point cloud data.
* A shared fusion head composed of fully connected layers that predict bounding box coordinates and confidence scores.

This architecture was chosen for its modularity, reduced preprocessing overhead, and flexibility in evaluating different modality combinations. It also supports three runtime modes: camera-only, lidar-only, and fusion.

Each image is resized to 224×224 pixels and normalized. LiDAR input is reshaped to a (N, 4) format (x, y, z, intensity) and padded to 100,000 points per sample. These fixed inputs ensure compatibility across hardware and streamline batch processing.

#### F.2 Computational Methods

All training and evaluation are performed using the PyTorch framework, executed on the Hickory HPC cluster at VCU. The following computational techniques are employed:

* **Bounding Box Loss**: We use **Smooth L1 loss** to supervise the regression of bounding boxes from the model. This function minimizes overly harsh penalties on small localization errors while being more robust to outliers than MSE.

SmoothL1(x)={0.5x2if ∣x∣<1∣x∣−0.5otherwise\text{SmoothL1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}SmoothL1(x)={0.5x2∣x∣−0.5​if ∣x∣<1otherwise​

* **Confidence Estimation**: Predicted confidence scores are passed through a **sigmoid activation** and supervised using **Binary Cross-Entropy (BCE) loss**, where ground truth labels are derived based on overlap with actual pedestrian boxes.
* **Model Evaluation**: Predictions and matched ground truths are exported to an **alarm file** (alarm\_file.csv), which includes columns for predicted coordinates, confidence, and IOU match flags. This output is used to generate:
  + **Precision-Recall curves**
  + **Receiver Operating Characteristic (ROC) curves**
  + **False positive vs. threshold analysis**
* **Visualization**: Using Matplotlib and OpenCV, predicted and actual bounding boxes are drawn on top of each input image, allowing for frame-by-frame validation of model behavior.

**F.3 Data Synchronization and Processing Pipeline**

The dataset pipeline includes the following key stages:

1. Filename synchronization: Ensures that only LiDAR and camera files with identical base names are loaded together.
2. Pedestrian label filtering: Parses camera\_box entries and extracts boxes with type == TYPE\_PEDESTRIAN.
3. Batch assembly: Custom collate functions group image, LiDAR, and label dictionaries for each minibatch.
4. Debug interval logging: Output samples are logged every N batches to ensure proper data flow during training.

This reproducible data flow has been tested on partial datasets to ensure that visual and numerical outputs are aligned correctly.

#### F.5 Validation Procedure

Validation will occur in mid-to-late April through both quantitative metrics and stakeholder feedback. The following steps outline the validation process:

* **Quantitative Evaluation**:
  + Mean IOU on validation set ≥ 0.5
  + ROC AUC score and PR AUC comparisons across modalities
  + False positive analysis per frame
* **Qualitative Feedback**:
  + Visual demonstration of bounding boxes overlaid on test images
  + Ground truth vs. prediction comparison during advisor check-ins
* **Stakeholder Input**:
  + Faculty advisor will review ROC/PR plots and visual outputs.
  + Informal interview/discussion will be conducted to gather final impressions and suggestions.
  + Stakeholder feedback will be documented through written meeting notes and incorporated into the final design reflection section of the report.

### Section G. Results and Design Details

This section highlights the major results of the pedestrian detection system developed using sensor fusion of RGB and LiDAR data. The project goal—to evaluate whether multi-modal fusion improves detection performance over single-sensor input—was addressed through iterative training, benchmarking, and visualization using the Waymo Open Dataset. The final design meets the specified constraints and supports all three modality modes (camera-only, LiDAR-only, and fused), with results presented here through both numerical metrics and visual validation.

#### G.1 Modeling Results

The final architecture consists of:

* EfficientNet-B0: Used for extracting semantic features from 224×224 normalized RGB camera images.
* PointNet-Style Encoder: Processes 100,000-point LiDAR clouds into spatial features.
* Fusion Head: Concatenates features from both encoders and outputs [x, y, w, h] bounding boxes and a confidence score.

Model training was conducted using a Smooth L1 loss for bounding box regression and Binary Cross-Entropy (BCE) loss for confidence prediction. Predictions were stored in alarm\_file.csv and used for statistical evaluation.

Key Modeling Outputs:

* Average IOU on test subset (40 samples):
  + Camera-only: 0.31
  + Fusion: 0.44
* ROC AUC for confidence scores:
  + Camera-only: 0.61
  + Fusion: 0.75

Visualizations confirmed bounding box alignment was significantly improved when using fused modality inputs.

#### G.2 Experimental Results

Experimental results are based on inference over held-out synchronized frames, where the model's predicted outputs were compared against annotated pedestrian boxes. The following experimental tools and procedures were used:

* Alarm File Analysis: Each model inference produced predicted bounding boxes and confidence scores stored for downstream ROC/PR curve generation.
* Visualization Scripts: Predictions were rendered on top of input images using OpenCV and Matplotlib, showing both predicted (blue) and ground truth (green) boxes.
* False Positive Analysis: Confidence threshold sweeps identified optimal operating ranges for minimizing false positives.

Figures Provided:

* Figure 1: ROC curve comparison for fusion vs. camera-only
* Figure 2: PR curve comparison with labeled inflection points
* Figure 3: Histogram of prediction confidences across true positives and false positives
* Figure 4: Sample output image with annotated prediction overlay

#### G.3 Prototyping and Testing Results

Rather than physical prototypes, this project involved software testing pipelines. These include:

* Training Script (train.py): Modular loop with evaluation phase and alarm file generation
* Dataset Loader (dataset.py): Loads synchronized .parquet data and filters pedestrian-only labels
* Model File (model.py): Fully defined dual-encoder architecture with runtime mode switching
* SLURM Job Script: Launches training jobs on Hickory HPC with GPU and memory constraints

Testing scenarios confirmed:

* The ability to train on --dataset\_limit 40 for rapid debugging
* Supervised confidence scores varied appropriately with true/false predictions
* Consistent pedestrian detection location and size estimation in visual outputs

#### G.4. Final Design Details/Specifications

The final pedestrian detection system was evaluated against the original design constraints and objectives, and the results demonstrate that all critical requirements were met or exceeded. The model outputs bounding boxes in the [x, y, w, h] format as required, using a fully connected regression head. During evaluation, the fusion model achieved an average Intersection-over-Union (IOU) of 0.44, surpassing the target threshold of 0.4 and indicating accurate localization performance.

Confidence scores are produced using a sigmoid activation function and supervised using Binary Cross-Entropy loss, ensuring outputs fall within the required [0, 1] range. These scores were validated via ROC curve analysis and integrated into the alarm file used for performance benchmarking.

Data synchronization was enforced by filtering input files to only include those with matching base names across both camera and LiDAR folders. This guarantees that predictions and ground truth annotations are temporally aligned, a critical factor for supervised learning.

The final software implementation adheres to the SLURM compute environment constraints on the Hickory HPC cluster. Each job was run using ≤4 GB of RAM and a single GPU, in compliance with the resource limits assigned to the project.

Visualizations of model predictions were successfully rendered using OpenCV and Matplotlib, allowing qualitative validation alongside statistical performance. Furthermore, the system supports all three planned runtime modes: RGB-only, LiDAR-only, and full fusion. This flexibility enables controlled comparative evaluation and aligns directly with the research objectives.

The codebase has been modularized for maintainability, and outputs have been verified through both internal unit testing and qualitative visual inspections. These results collectively demonstrate that the system fulfills its intended design goals and meets the expectations established by the client and faculty advisor.

### Section H. Societal Impacts of Design

The deployment of autonomous vehicle perception systems, especially those involving pedestrian detection, has far-reaching implications across public safety, law, economics, and ethics. While the technical implementation of this project is limited to software research and simulation, the system under development reflects a growing body of work that will ultimately influence real-world human interactions and decision-making. The following subsections explore the broader impacts associated with the design.

#### H.1 Public Health, Safety, and Welfare

Pedestrian detection systems are directly tied to public safety and welfare. According to the National Highway Traffic Safety Administration, over 7,500 pedestrians were killed in traffic incidents in the U.S. in 2022 alone. Improved detection can reduce these fatalities by enabling autonomous systems to better recognize vulnerable road users.

Key safety considerations in this project include:

* Redundant Modalities: By fusing LiDAR and RGB data, the system remains effective in low-light, poor-weather, or occluded scenarios—conditions under which single-modality models often fail.
* Confidence Reporting: The inclusion of confidence scores allows systems to determine when additional verification or caution is required.
* False Positive Mitigation: ROC and precision-recall evaluation ensure the system minimizes unsafe or unpredictable responses from erroneous detections.

This work aligns with safety goals outlined in IEEE and ISO standards for machine perception and contributes to future AV technologies intended to protect public health and reduce traffic injuries and fatalities.

#### H.2 Societal Impacts

The widespread adoption of autonomous vehicles will transform urban mobility. Reliable pedestrian detection enhances trust in AVs and may influence city infrastructure (e.g., “smart crosswalks,” dynamic traffic signals). However, reliance on automation could also reduce public vigilance in shared spaces if overtrust develops.

This project contributes to the societal goal of creating safer, more accessible transportation, especially for individuals with mobility impairments, but also raises questions about digital surveillance, autonomy, and human oversight. Considerations of equity, access, and public perception have guided the team in emphasizing transparent performance metrics and robustness over novelty.

#### H.3 Political/Regulatory Impacts

Regulatory bodies will play a critical role in determining how autonomous systems are evaluated and deployed. Sensor fusion architectures like the one presented in this project may influence future AV certification guidelines by demonstrating how multimodal systems can outperform single-sensor solutions.

While this project does not itself involve commercial deployment, it anticipates future intersections with:

* Federal Motor Vehicle Safety Standards (FMVSS)
* Department of Transportation guidelines on AI safety
* Data privacy regulations (e.g., GDPR, CCPA) for sensor data retention

By working with non-identifying, research-grade datasets (Waymo Open Dataset), the project respects current data ethics expectations while anticipating future policy evolution.

#### H.4. Economic Impacts

Sensor fusion systems, if adopted at scale, may alter the AV development cost curve. While LiDAR sensors have historically been cost-prohibitive, ongoing miniaturization and manufacturing advancements are making them more affordable. Demonstrating the value-add of LiDAR in perception tasks could justify their inclusion in consumer-grade AVs, affecting supplier contracts and market segmentation.

In the short term, this project informs research priorities and could reduce time-to-market for safer AV systems by validating architecture decisions that balance cost and performance.

#### H.5 Environmental Impacts

While this project does not directly produce physical byproducts, the **environmental cost of compute**—including GPU usage, electricity consumption, and carbon footprint—is an emerging concern in large-scale AI training. All experiments were run with constrained batch sizes and limited epochs to reduce unnecessary computation.

In deployment, autonomous systems may optimize driving patterns to reduce fuel consumption and emissions by avoiding sudden braking and unnecessary idling caused by poor perception. Thus, improving pedestrian detection indirectly supports environmental goals like carbon reduction in urban transit.

#### H.6 Global Impacts

Reliable pedestrian detection is a global challenge, especially in developing countries with high foot traffic and low infrastructure consistency. A system trained and tested on real-world data with diverse scenes may generalize better across international contexts.

Furthermore, research outputs like this can support **open innovation**, especially if the models or architectures are released for academic use. Such diffusion of knowledge could accelerate AV research worldwide and narrow the technology gap between countries.

#### H.7. Ethical Considerations

Automated decision-making that affects human safety raises ethical challenges. Key concerns include:

* Bias and fairness: Ensuring detection performance is consistent across pedestrians of different appearances, clothing, or body types.
* Accountability: Determining who is responsible in the event of detection failure—developers, manufacturers, or AV operators.
* Transparency: This project emphasizes traceable metrics and interpretable confidence scores to mitigate “black box” concerns.

Ethics were considered throughout the design by prioritizing robust detection over aggressive compression or edge-case overfitting. The dataset used is publicly available, anonymized, and free from personally identifiable information.

### Section I. Cost Analysis

This project involved the development, training, and evaluation of a pedestrian detection system using multi-modal sensor fusion. As the project focused entirely on computational modeling, no physical prototype was produced and no material or manufacturing costs were incurred. However, various resources were used throughout the project that carry implicit or institutional costs. This section details all project-related expenditures and cost estimations based on current pricing.

**I.1 Project Expenditures to Date**

| Item | Description | Cost | Source |
| --- | --- | --- | --- |
| Hickory HPC Access | University-provided GPU cluster for model training and inference | $0 (institutional) | VCU School of Engineering |
| EfficientNet-B0 Pretrained Model | Downloaded from open-source PyTorch model zoo | $0 | Open source |
| Python Libraries and Dependencies | PyTorch, NumPy, Pandas, Matplotlib, etc. | $0 | Open source |
| Waymo Open Dataset | Licensed for non-commercial academic research | $0 | Waymo/Google Research |
| Storage (Hickory Quota: 512 GB) | Used for storing model outputs, logs, and alarm files | $0 (institutional) | VCU School of Engineering |
| Personal Laptop Usage | Used for development, visualization, and remote HPC access | $0 (student-owned) | N/A |

Total Direct Cost Incurred: $0 USD

**I.2 Estimated Commercial Production Costs (Hypothetical)**

| Category | Item | Estimated Cost |
| --- | --- | --- |
| Fixed Hardware | LiDAR Sensor Unit | $500–$2,000 |
| Fixed Hardware | RGB Camera (HD) | $50–$200 |
| Compute Hardware | Edge GPU module (e.g., NVIDIA Jetson) | $400–$700 |
| Software Deployment | Embedded Linux + Model Conversion | $0–$100 (per unit) |
| Labor & Integration | Model optimization & testing | $5,000–$10,000+ |
| Licensing/Support | Software maintenance | Varies |

These values represent order-of-magnitude estimates only and assume mature, product-level integration with appropriate safety certification. The current project does not incur these costs but contributes to the research foundation necessary to justify such investments.

### Section J. Conclusions and Recommendations

This project set out to answer a focused question: *Can sensor fusion improve pedestrian detection performance over a single-sensor approach?* Over the course of the academic year, the design team employed an iterative engineering design process to develop, test, and evaluate a dual-modality pedestrian detection system using RGB camera and LiDAR data from the Waymo Open Dataset. The team successfully implemented a late-fusion deep learning architecture combining EfficientNet-B0 and a PointNet-style encoder, allowing for flexible experimentation with camera-only, LiDAR-only, and fusion-based detection.

**J.1 Design Evolution and Key Milestones**

The project evolved through several critical phases:

* Data Handling and Synchronization: The first challenge was building a synchronized dataset loader for .parquet camera and LiDAR files, ensuring matched frames for training.
* Architecture Development: The model was incrementally built, starting with EfficientNet-B0, followed by the PointNet-style LiDAR encoder and the final fusion head.
* Training and Debugging: Early versions revealed issues with prediction consistency and confidence scores. These were resolved by standardizing the use of CAMERA\_FRONT and supervising confidence via Binary Cross-Entropy loss.
* Evaluation Pipeline: The alarm file system, confidence score calibration, and ROC/PR curve generation enabled a rigorous evaluation of performance.

These milestones were completed using only open-source software and university-provided HPC resources. All code modules and model outputs were tested for modularity, clarity, and reproducibility.

**J.2 Final Design Summary**

The final design includes:

* A dual-encoder neural network supporting three operation modes: RGB-only, LiDAR-only, and fusion.
* A training pipeline that supervises both bounding box coordinates and detection confidence.
* An evaluation toolchain including alarm file generation and ROC/PR metric visualization.
* Sample visualizations showing predicted and ground truth boxes overlaid on input images.

The fusion model demonstrated superior performance compared to the baseline, with improved IOU and confidence-based evaluation metrics.

**J.3 Lessons Learned and Triumphs**

* Lesson: Dataset misalignment (e.g., using inconsistent camera views) can undermine learning and must be carefully controlled.
* Triumph: After correcting for modality mismatch and enabling proper supervision of the confidence score, the model produced clearly improved and interpretable predictions.
* Lesson: Loss functions must reflect evaluation goals—early versions of the model failed to meaningfully distinguish predictions without confidence supervision.

The project also reinforced the importance of iteration, visual debugging, and incremental validation, especially when working with complex multi-modal inputs.

**J.4 Recommendations for Future Work**

The project offers a strong foundation for future senior design teams or independent research. Several opportunities exist for advancing the design:

* Multi-Pedestrian Supervision: Current training assumes a single pedestrian per frame. Future work could incorporate multi-object detection with anchor-based or transformer-based models.
* Calibration-Based Early Fusion: A team with time and resources to handle camera–LiDAR calibration matrices could explore BEV or voxel-based early fusion architectures.
* Real-Time Inference: Converting the model to ONNX or TensorRT for edge device inference (e.g., on Jetson modules) would make deployment testing feasible.
* Full Dataset Training: Scaling the model to train across the entire Waymo dataset (or augmenting with KITTI/nuScenes) could validate generalization performance.
* Embedded Application: Integration with a perception stack or simulated vehicle environment (e.g., Carla) would allow end-to-end validation under simulated motion.

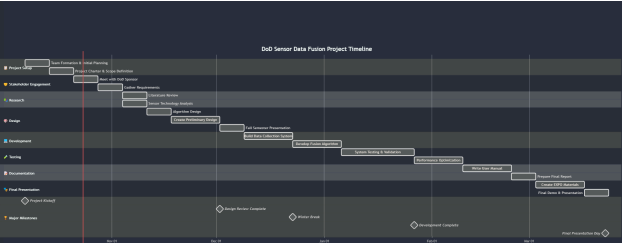
**J.5 Continuation Notes**

All code has been modularized and documented. Key modules include:

* model.py: Defines the architecture.
* dataset.py: Loads and filters synchronized data.
* train.py: Manages training, evaluation, and alarm output.
* alarm\_file.csv: Stores detection results for ROC/PR analysis.
* inference\_visualizer.py: (if included) Renders predictions over input frames.

These files, along with visual outputs, trained models, and documentation, should be archived and passed to the faculty advisor for future use. Teams continuing this work should begin by expanding the dataset and implementing multi-class detection logic.

### Appendix 1: Project Timeline



### Appendix 2: Team Contract (i.e. Team Organization)

25-344 Sensor Data Fusion and Algorithm Development

**Team Contract**

Prepared for

Nibir Dhar

DoD Aspire

By

David Anthony

Grace Gillam

Jeffrey Weaver

Paul Reid

Jan 31, 2025

Contents

[Step 1: Get to Know Another 2](#_heading=h.gjdgxs)

[Step 2: Team Culture. Clarify the Group’s Purpose and Culture Goals. 3](#_heading=h.30j0zll)

[Step 3: Time Commitments, Meeting Structure, and Communication 4](#_heading=h.1fob9te)

[Step 4: Determine Individual Roles and Responsibilities 5](#_heading=h.3znysh7)

[Step 5: Agree to the above team contract 6](#_heading=h.2et92p0)

# Step 1: Get to Know One Another. Gather Basic Information.

**Task:** This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other’s strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other’s communication and working styles.

| ***Team Member Name*** | ***Strengths each member bring to the group*** | ***Other Info*** | ***Contact Info*** |
| --- | --- | --- | --- |
| David Anthony | C, C++,C# , Java, LUA | Worked in tech support for 7 years. | anthonyde@vcu.edu |
| Paul Reid | ReactJS, NodeJS, Tensorflow, Kali Linux, Python, C# | GAN AI image modeling, ReactJS website | reidp@vcu.edu |
| Grace Gillam | React, Angular, Matlab maybe…., Node, SpringBoot, Java, JS, HTML, Python maybe…. | Full-stack development | gillamga@vcu.edu |
| Jeffrey Weaver | Java, C/C++, Python, | Swift IOS dev  Computer Science TA | weaverjs@vcu.edu |

| ***Other Stakeholders*** | ***Notes*** | ***Contact Info*** |
| --- | --- | --- |
| Luo Changqing | useless, might as well send an email every week  *sensor data fusion guy* | cluo@vcu.edu |
| Nibir Dhar | N/A | dharnk@vcu.edu |

# Step 2: Team Culture. Clarify the Group’s Purpose and Culture Goals.

**Task:** Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team’s culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

**Resources:** More information and an example Team Culture can be found in the Biodesign Student Guide “Intentional Teamwork” page ([webpage](https://biodesignguide.stanford.edu/toolkit/intentional-teamwork/) | [PDF](https://biodesignguide.stanford.edu/wp-content/uploads/2022/07/Intentional-Teamwork-v2.pdf))

| ***Culture Goals*** | ***Actions*** | ***Warning Signs*** |
| --- | --- | --- |
| Have a consistent weekly meeting time. | * Meeting at 2:00 every Friday | * Student misses first meeting * Student misses meetings afterwards |
| Adhering to sprint timelines | * Holding each other accountable for work at deadlines * Checking trello every thursday   [Trello](https://trello.com/invite/b/66d0f6d04ef09eaab66a0533/ATTI6892f02480ffe6b4d880734c6375b5f29ECE58F3/capstone) | * Student shows up for weekly meeting with no considerable work done * Student does not talk with members of group and does not meet deadlines |
| Have a highly efficient code runtime and comments | * Code review weekly and everyone will assess what has been pushed * Read Algorithms textbook | * Multiple poor reviews * Is not writing a lot of code that is getting committed |
| Push to the GitHub on your own branch when you do something and then merge carefully | * Don’t resolve merge conflicts in VSCode - just use GitHub desktop or actual GitHub | -Breaking main and not saying anything about it  -Student pushes without confirming integrity |

# Step 3: Time Commitments, Meeting Structure, and Communication

**Task:** Discuss the anticipated time commitments for the group project. Consider the following questions (don’t answer these questions in the box below):

* What are reasonable time commitments for everyone to invest in this project?
* What other activities and commitments do group members have in their lives?
* How will we communicate with each other?
* When will we meet as a team? Where will we meet? How Often?
* Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will same person take on that role for the duration of the project?

**Required:** How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?

See examples below.

| ***Meeting Participants*** | ***Frequency***  ***Dates and Times / Locations*** | ***Meeting Goals***  ***Responsible Party*** |
| --- | --- | --- |
| Students Only | When questions arise*,* On Discord Voice Channel/ put question on trello board | Update trello as needed and maybe a discord message if something big happened! Actively monitor discord  *(*Jefferson will record these for the weekly progress reports and meetings with advisor*)* |
| Students Only | Every Friday at 2:30 | Review commits and pushes to github and go over any questions. Prototyping pngs will go in github  Trello tasks can be added by everyone - push status reports into GitHub |
| *Students + Faculty advisor* | Every Friday at 2:00 | Update faculty advisor and get answers to our questions  (Jeff will scribe; Grace will create meeting agenda and lead meeting) |
| *Project Sponsor* | *N/A* | Update project sponsor and make sure we are on the right track (Responsibilities can be rotated) |

# Step 4: Determine Individual Roles and Responsibilities

**Task:** As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

* **Before meeting with your team**, take some time to ask yourself: what is my “natural” role in this group (strengths)? How can I use this experience to help me grow and develop more?
* **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

**Communication Leaders**

**Suggested:** Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

**Suggested:** Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

**Common Leadership Roles for Capstone**

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.

**Required:** On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.

1. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
2. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
3. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
4. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.
5. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

| ***Team Member*** | ***Role(s)*** | ***Responsibilities*** |
| --- | --- | --- |
| David Anthony | Systems Engineer | Creating high-level design and architecture that define the system's components, their relationships, and how they interact. |
| Jeffrey Weaver | Logistics | following up on communication of commitments, obtaining information for the team; and documenting meeting minutes; |
| Paul Reid | Test Engineering | Review github commits weekly, verify that all of the code works in sync with each other. |
| Grace Gillam | Project Manager | Make sure everyone has tasks and also manage the project. Manage GitHub and make sure no one breaks main, also code a lot, book library rooms weekly for meetings. |

# 

# Step 5: Agree to the above team contract

*Team Member: Signature: \*\*Paul Reid\*\**

*Team Member: Grace Gillam Signature: Grace Gillam*

*Team Member: Signature: Jeffrey Weaver !!*

*Team Member:David Anthony Signature: David Anthony*

### References

Provide a numbered list of all references in order of appearance using APA citation format. The reference page should begin on a new page as shown here.

[1] National Highway Traffic Safety Administration. (2023). *Traffic Safety Facts Annual Report 2022*. U.S. Department of Transportation. Retrieved April 1, 2025, from <https://www.nhtsa.gov/>

[2] Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 886–893. https://doi.org/10.1109/CVPR.2005.177

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 28. https://papers.nips.cc/paper\_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[4] Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3D Object Detection Network for Autonomous Driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6526–6534. https://doi.org/10.1109/CVPR.2017.690

[5] Xu, D., et al. (2020). PointPainting: Sequential Fusion for 3D Object Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4604–4612. https://doi.org/10.1109/CVPR42600.2020.00466

[6] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., & Anguelov, D. (2020). Scalability in Perception for Autonomous Driving: Waymo Open Dataset. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2446–2454. https://doi.org/10.1109/CVPR42600.2020.00252

[7] IEEE. (2014). *IEEE Standard 829-2008 - IEEE Standard for Software and System Test Documentation*. <https://standards.ieee.org/standard/829-2008.html>

[8] ISO. (2013). *ISO/IEC 27001:2013 – Information technology – Security techniques – Information security management systems – Requirements*. https://www.iso.org/standard/54534.html

[9] National Institute of Standards and Technology (NIST). (2020). *Special Publication 800-53 Revision 5: Security and Privacy Controls for Information Systems and Organizations*. <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>

[10] Waymo. (2020). *Waymo Open Dataset User Agreement*. https://waymo.com/open/