# Real Time Collision Warning System Using Monocular Vision and Object Tracking

CS5330 Final Project Yunyu Guo

# Introduction

Autonomous vehicles rely heavily on accurate perception of the surrounding environment to ensure safety. A critical component of this perception is the estimation of Time-to-Collision (TTC) with other objects, particularly leading vehicles. This project presents a system that integrates object detection (YOLOv5), tracking (BoT-SORT), and monocular depth estimation (Monodepth2) to calculate TTC for objects in driving scenes. The pipeline processes video input, identifies and tracks objects, estimates their depth frame-by-frame, and computes TTC based on the rate of depth change. The system's performance is evaluated using the KITTI depth completion dataset, assessing both the accuracy of the underlying depth predictions and the final TTC estimates against ground truth data. This provides a practical assessment of using deep learning components for monocular collision warning.

# Related Work

## A. Monocular Depth Estimation

Monodepth2 [1] presents a self-supervised approach for monocular depth estimation that learns from monocular videos without ground truth depth supervision. The method employs minimum reprojection loss, auto-masking of stationary pixels, and multi-scale estimation to produce accurate depth maps. In the project, it leverages Monodepth2's ResNet18-based encoder-decoder architecture to generate per-pixel depth maps from single RGB images, which serve as the foundation for the TTC calculations.

## B. Time-to-Collision Estimation

Kopf et al. [2] introduce a framework for TTC and collision risk estimation using local scale and motion cues from monocular vision. The paper establishes mathematical relationships between optical flow, looming (apparent size changes), and TTC, demonstrating that robust TTC estimates are possible without explicit depth reconstruction. While the project's approach differs by using depth maps, this work provides theoretical foundations for the TTC calculation formula TTC = depth / velocity, where velocity is derived from temporal depth changes.

# Related Work

## A. Multi-Object Tracking

BoT-SORT [3] builds on previous tracking-by-detection frameworks by combining motion prediction via Kalman filtering with re-identification (ReID) features. It introduces camera motion compensation and interaction modeling to achieve state-of-the-art tracking performance. The project pipeline utilizes BoT-SORT with the osnet_x0_25_msmt17 ReID model to maintain consistent object identities across frames, which is critical for accumulating depth histories and computing TTC for each object.

## B. Object Detection

YOLOv3 [4] presents an incremental improvement to the YOLO (You Only Look Once) detection framework, featuring a deeper feature extractor network and multi-scale predictions. Its efficient design allows real-time detection with competitive accuracy. Building upon this, YOLO9000 [5] (also known as YOLOv2) expands the detection capacity to over 9000 object categories through hierarchical classification. YOLOv5 builds upon techniques in both YOLOv3 and YOLOv4 [6]. The project's system employs YOLOv5 [7], a successor to these models, to rapidly detect vehicles and other objects in each video frame, providing the bounding boxes required for tracking and depth extraction.

# Solution

Implementation Pipeline

Step 1: Integration of three core models

- YOLOv5 for object detection
- BoT-SORT for multi-object tracking
- Monodepth2 for single-camera depth estimation

Step 2: Development of TTC calculation pipeline and visualization tools

- Time-to-collision calculation logic based on depth change rate
- Real-time visualization with color-coded warnings

Step 3: Evaluation framework development

- Quantitative evaluation using KITTI depth completion dataset
- Comparison against ground truth depths and derived TTC

# Problem: Bounding Box Color Flickering

- Track consecutive frames where TTC is below threshold (2.0s)
- Only change bounding box to red after 3+ consecutive critical frames
- Immediately reset counter when TTC returns to safe levels
- This prevents false alerts from isolated noisy frames
- Results: Warning visualization is now stable and reliable with minimal flickering, while maintaining quick response to collision risks.

# Results

TABLE I. DEPTH ESTIMATION PERFORMANCE

| Metric | Value |
|---|---|
| AbsRel | 0.0732 |
| SqRel | 0.3586 |
| RMSE | 3.0595 |
| RMSElog | 0.1134 |
| $\delta < 1.25$ | 0.9510 |
| $\delta < 1.25^2$ | 0.9895 |
| $\delta < 1.25^3$ | 0.9962 |

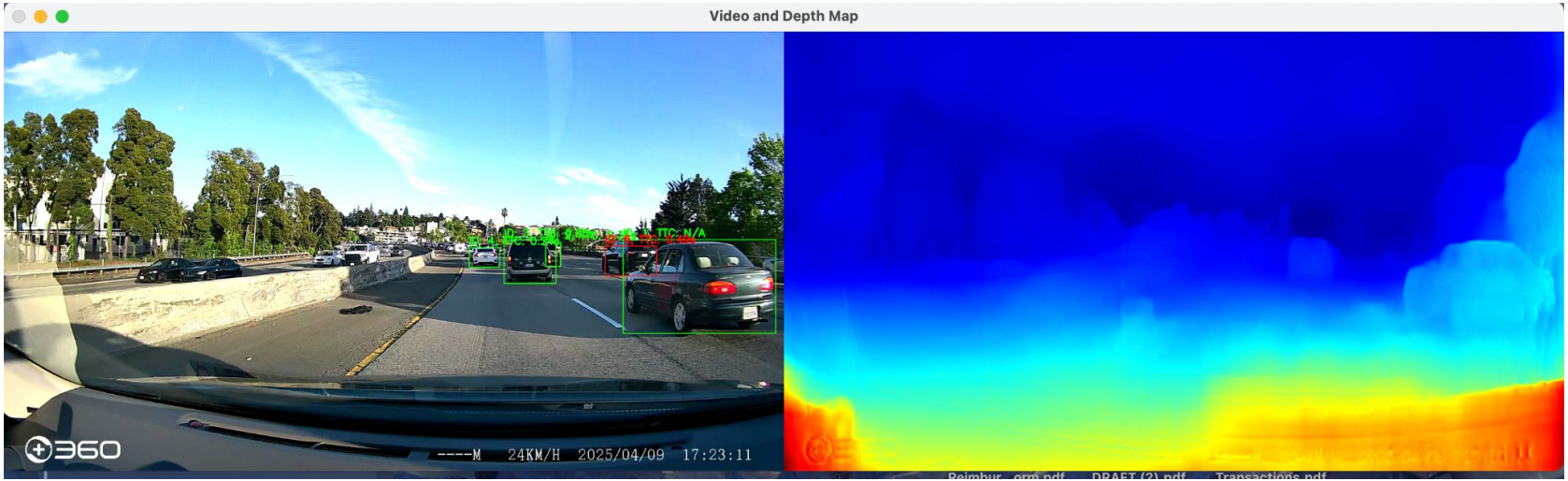
TABLE II. TTC ESTIMATION ERRORS

| Metric | Value |
|---|---|
| Total TTC Calculation | 40 |
| Mean Absolute TTC Error | 3.24s |
| Median Absolute TTC Error | 0.79s |
| Mean Relative TTC Error | 163.7% |
| Median Relative TTC Error | 61% |

The results indicate strong performance for a monocular depth estimation approach, with over 95% of pixels having an error less than 25% of the ground truth value. This provides a reliable foundation for subsequent TTC calculations. The substantial difference between mean and median errors suggests the presence of outliers in TTC estimation. While the median absolute error of 0.79 seconds indicates reasonable accuracy for many cases, the high mean relative error (163.7%) highlights challenges in handling certain scenarios.

# Results

Fig. 1.Example of frames with detected objects, their tracked IDs, and TTC estimates on the left, and the depth map on the right




When applied to driving video sequences, the system successfully detects vehicles, estimates their depth, and calculates TTC. The system also provides visual warnings when TTC falls below a safety threshold (2.0 seconds) for multiple consecutive frames, highlighted by changing bounding box colors from green to red.

The depth maps reveal the system's ability to accurately capture the relative depth of the scene, with closer objects appearing in warmer colors (red/yellow) and distant objects in cooler colors (blue/purple).

These qualitative results demonstrate the potential of the monocular approach for cost-effective collision warning systems.