Labeller Assignment

Abstract

This report documents the development of an end-to-end computer vision pipeline for semantic/instance segmentation and multi-object tracking focused on vehicles and pedestrians. The project covers data collection via Unsplash API, annotation with Labellerr's model-assisted labeling, YOLO-based training and evaluation, ByteTrack integration for tracking, and a Streamlit demo for deployment and QA. The work simulates a real-world ML lifecycle from dataset creation to results review and delivery, culminating in a reproducible demo and structured tracking JSON export.

Dataset and Annotation Workflow

Images were collected through the Unsplash developer API to ensure variety in scenes, lighting, and scales while adhering to platform usage guidelines and API practices. Annotation was performed in Labellerr with polygon masks; initial manual efforts were accelerated using model-assisted click-to-outline, which improved throughput while preserving mask quality. Labeled data were exported in COCO-compatible format suitable for YOLO-based training and iterative quality checks.

Training and Evaluation Setup

The YOLO-Seg model was fine-tuned on the exported dataset with standard augmentations and validation monitoring, prioritizing stability on a relatively small custom split. Best checkpoints were selected based on validation metrics and later reused for video tracking to ensure consistency between image evaluation and video performance.

Model Performance Results

Metric	Value
mAP@0.5	0.550
mAP@0.5:0.95	0.344
Precision	0.561
Recall	0.616

F1-Score	0.588

Performance Analysis

From the confusion matrix analysis, the model demonstrates reasonable performance in distinguishing between vehicles, pedestrians, and background. The normalized confusion matrix shows that vehicles are correctly classified 71% of the time, while pedestrians achieve 66% correct classification. Some confusion occurs between pedestrians and background (68% misclassification rate), indicating opportunities for improvement in pedestrian detection sensitivity.

Precision-Recall Curves

The precision-recall curves for both bounding box detection and mask segmentation show distinct performance characteristics between classes. Vehicle detection achieves better AP scores (vehicles: 0.645 for boxes, 0.622 for masks) compared to pedestrians (0.454 for boxes, 0.414 for masks), reflecting the typical challenge of detecting smaller, more variable pedestrian shapes.

Tracking System and Application

ByteTrack was integrated via Ultralytics' tracking API using tracker="bytetrack.yaml", enabling ID assignment for detections across frames and producing a stable stream of tracked objects. A Streamlit app accepts video upload, runs YOLO+ByteTrack, overlays bounding boxes and IDs frame-by-frame, and exports both a rendered video and a structured tracking_results.json for reproducibility.confusion_matrix.jpg

Challenges and Resolutions

- **Labeling efficiency vs. quality**: Manual polygon masks were initially slow; switching to Labellerr's model-assisted click-to-outline reduced effort while maintaining accuracy.confusion_matrix.jpg
- **Class imbalance**: Pedestrians showed lower performance than vehicles, likely due to size variation and occlusion patterns in the dataset.
- **Background confusion**: The model occasionally misclassifies pedestrians as background, indicating need for more diverse pedestrian samples and better augmentation strategies.

- **Tracking stability**: Tuned ByteTrack parameters (confidence=0.5, IoU=0.7) to reduce ID switches while maintaining detection sensitivity

Key Learnings

- Model-assisted labeling in Labellerr significantly accelerates dataset creation while maintaining editorial control for final mask quality.confusion_matrix.jpg
- Class-specific performance varies significantly: vehicle detection benefits from more consistent shapes and sizes, while pedestrian detection requires more sophisticated approaches for handling variation.

Implementation Architecture

Core Components

- **Inference Pipeline**: YOLO model loading, ByteTrack integration via Ultralytics tracking API, and per-frame JSON export
- **Video Processing**: OpenCV VideoWriter with codec fallbacks (mp4v → XVID → avc1) for cross-platform compatibility
- **User Interface**: Streamlit app with progress tracking, video preview, and dual download functionality

Step-by-Step Reproduction Guide

1. **Dataset Collection**: Use Unsplash API for diverse street scene images; manage API keys with environment variables
2. **Annotation**: Create Labellerr project, utilize AI-assisted click-to-outline, export COCO segmentation format
3. **Training**: Fine-tune YOLO-Seg with augmentations, monitor validation metrics, select best weights
4. **Evaluation**: Generate confusion matrices, PR curves, and confidence analysis using validation data
5. **Integration**: Implement ByteTrack via Ultralytics tracking API with appropriate thresholds
6. **Deployment**: Build Streamlit app with upload, processing, and download capabilities
7. **Testing**: Validate on diverse video content, tune confidence/IoU thresholds for optimal performance

Results Summary

The system demonstrates a functional end-to-end pipeline with documented performance metrics (mAP@0.5: 0.550), visual evaluation through multiple curve analyses, and a deployable demo application. The project successfully simulates a practical computer vision development cycle from data creation through deployed validation, with clear performance characteristics and improvement pathways identified.BoxPR_curve.jpg+2

Technical Specifications

- **Model**: YOLO-Seg (YOLOv8 segmentation variant)
- **Tracking**: ByteTrack via Ultralytics integration
- **Deployment**: Streamlit web application
- **Output Formats**: MP4 video with overlays, JSON tracking data
- **Performance**: mAP@0.5 = 0.550, F1-Score = 0.588

Future Improvements

Based on the evaluation results, priority improvements

- Enhanced pedestrian detection through targeted data augmentation and class-specific loss weighting
- Improved background-foreground separation to reduce pedestrian misclassification
- Hyperparameter tuning for confidence thresholds based on deployment scenarios
- Extended evaluation on longer video sequences to assess tracking stability over time