Problem Statement: Vehicle Risk Assessment System

Objective:
Develop a computer vision system that processes traffic footage to detect vehicles, estimate their speed and proximity, and assign real-time risk scores.

Provided dataset

https://drive.google.com/drive/folders/1PefadTerfdvcBqXkIGLNZAQVhThV5jta?usp=sharing

Technical Requirements:

1. **Data Preparation**
   - Annotate images of Indian road traffic
   - **Vehicle classes:** ['Ambulance', 'Auto', 'BiCycle', 'bike', 'bus', 'car', 'tractor', 'truck', 'person']
   - Use Roboflow for annotation and dataset management
   - Split data into training, validation, and test sets
2. **Model Development**
   - Train YOLOv8n and YOLOv8s models on annotated dataset
   - Compare both models based on detection accuracy and inference speed
   - Optimize models for real-time performance

3. **Risk Assessment Algorithm**
   - Implement composite risk scoring using**:**
   - Risk_Score = Speed_Risk + Distance_Risk + Vehicle_Type_Risk
4. System Integration
   - Process image sequences using actual camera FPS for accurate speed calculation
   - Implement multi-object tracking across frames
   - Calculate real-time speed and distance estimates
   - Generate color-coded risk visualization

Deliverables:

Phase 1: Data & Model Training
- Annotated dataset in YOLO format
- Trained YOLOv8n and YOLOv8s model weights
- Model performance comparison report

Phase 2: Algorithm Development
- Multi-object tracking implementation
- Speed and distance estimation module
- Risk assessment algorithm
- Validation on sample data

Phase 3: System Integration
- End-to-end processing system
- Annotated output with risk assessment
- Exportable detection data with risk scores

Evaluation Metrics:

Model Performance:

- Detection accuracy (mAP50, mAP50-95)
- Precision and recall rates

Inference speed (FPS):

- Risk Assessment

Speed estimation accuracy:

- Distance calculation reliability
- Risk level assignment consistency
- System Performance
- Processing efficiency
- Real-time capability

Submission Requirements:

- Complete source code
- Trained model weights
- Sample input/output demonstrating functionality
- Performance comparison report
- Installation and execution instructions.