Speed detection of vehicles from video clip using a single model in different extreme weather conditions

Abhinav P T

Project Guide: P. Balamurugan, IEOR, IIT Bombay
Team Name: QuantaQuotient (22B0656)
IE643 Course Project

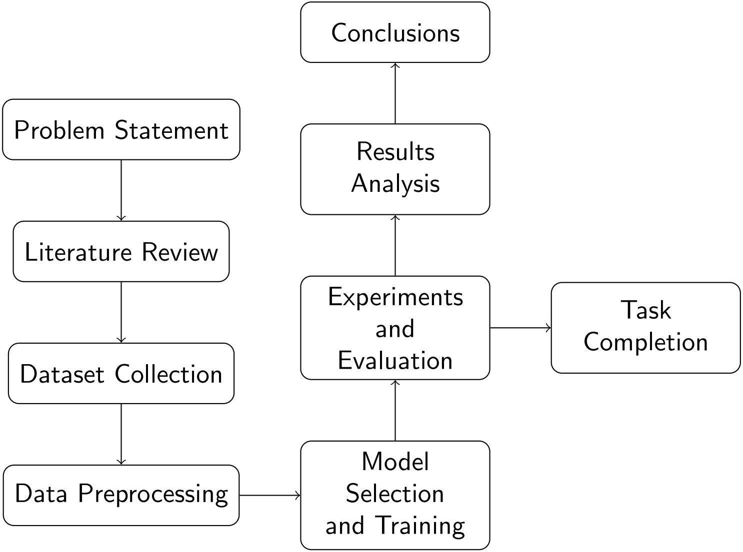
September 25, 2024

Overview

Problem Statement

This project aims to develop a vehicle speed detection system using video footage from still cameras, ensuring accuracy in both normal and extreme weather conditions (fog, rain, snow). Accurate speed monitoring is essential for traffic safety and management, especially under adverse weather that hinders visibility.

Workflow


Conclusions
Problem Statement
Results Analysis
Literature Review
Experiments and Evaluation
Task Completion
Dataset Collection
Data Preprocessing
Model Selection and Training

Work done before prep-presentation review

- **Literature Reviewed:**
  - Gollapalli et al. (2023) - Utilized YOLOv8 and Deep SORT for vehicle detection and speed estimation.
  - Neamah and Karim (2023) - Implemented YOLOv8 for real-time traffic monitoring with high accuracy.
- **Model Identification:**
  - Selected YOLOv8 as the primary model for vehicle detection.

Work done before prep-presentation review

- **Speed Estimation Process:**
  - **Source and Target Identification:** Defines starting and ending positions of vehicles in the frame.
  - **Region of Interest (ROI):** Focuses computation on specific areas for improved accuracy.
  - **Calculation Methodology:**
    - **Distance Calculation:** Based on the absolute difference between vehicle start and end positions.
    - **Time Calculation:** Determined from frame count and frame rate (FPS).
    - **Speed Formula:** Speed $=$ Distance / Total Time $\times$ Conversion Factor (e.g., km/h).

Work Done After Pre Presentation Review

- Data Collection and Processing
- Model Development
- Model Training and Fine-Tuning
- Experimental Setup
- Conducting Experiments
- Result Analysis and Evaluation
- Documentation of Findings

Proposed Approach: YOLOv8 Model

- **YOLOv8 Overview:**
  - YOLOv8 (You Only Look Once version 8) is a state-of-the-art, real-time object detection model.
  - Designed for efficiency with a single-stage architecture, allowing simultaneous detection and classification.
- **Network Structure:**
  - Improved feature extraction techniques enhance accuracy and speed.
  - Incorporates context aggregation to better understand the spatial relationships between objects.
  - Capable of detecting multiple objects within a frame with high precision.

Proposed Approach: Source and ROI Identification

- **Source and Target Identification:**
  - Identifies the starting (source) and ending (target) positions of vehicles within the video frame.
  - Essential for tracking vehicle movement and speed calculation.
- **Region of Interest (ROI):**
  - Focuses on specific areas within the frame where vehicle detection and speed estimation occur.
  - Reduces computational load and enhances detection accuracy by concentrating on relevant segments.

Proposed Approach

- **Perspective Transformation:**
  - Uses a transformation matrix M to map points from the source plane to the target plane.
  - Essential for correcting perspective distortions and ensuring accurate position tracking.
- **Speed Calculation:**
  - **Distance Calculation:**
    - Computes the absolute difference between vehicle start and end positions within the ROI.
  - **Time Calculation:**
    - Elapsed time is derived from the number of frames and the video frame rate (FPS).
  - **Speed Formula:**
    - Speed is calculated using:

$$\text{Speed} = \frac{\text{Distance}}{\text{Total Time}} \times \text{Conversion Factor}$$

    - Converts speed from pixels/frame to desired units (e.g., km/h).

Dataset and Data Preprocessing

- **Dataset: UA Detrac**
  - Contains 100 hours of video footage divided into training and testing splits.
  - Labels include class, x_c, y_c (center coordinates), height, and width for each detected object.
- **Data Selection:**
  - Sampled 1 out of every 15 frames from the video.
  - Resulted in around 5,000 training images and 3,000 validation images.
- **Data Augmentation:**
  - Applied augmentations to simulate extreme weather conditions: fog, rain, and snow.
  - Original images conserved, resulting in a total of $4\times$ the number of images.
  - Each label corresponds identically to the original images.

Experimental Setup and Details

- **Environment:**
  - Utilized Kaggle with GPU P100 for model training and evaluation.
- **Training Configuration:**
  - **Epochs:** 50 epochs for training.
  - **Patience:** Early stopping with a patience of 10 epochs to prevent overfitting.
- **Model:** YOLOv8x for object detection.
- **Metrics:**
  - Evaluated using accuracy, precision, recall, and mAP@50-95.
- **Data Handling:**
  - Implemented data augmentation techniques to enhance model robustness against weather variations.

Experimental Results

Metric	Fine-Tuned YOLOv8x
Mean Average Precision (mAP@50)	0.630
Mean Average Precision (mAP@50-95)	0.475
Precision	0.684
Recall	0.600

Table: Performance metrics for fine-tuned YOLOv8x model

Experimental Results

- **Mean Average Precision (mAP@50):**
  - Achieved mAP@50 of 0.630, indicating high accuracy in object detection.
- **Mean Average Precision (mAP@50-95):**
  - mAP@50-95 of 0.475 reflects consistent detection across varying IoU thresholds.
- **Precision:**
  - Precision of 0.684 indicates a strong ability to minimize false positives.
- **Recall:**
  - Recall of 0.600 shows good detection capability but suggests room for improvement.

Experimental Results

Class	mAP@50
Bus	0.819
Car	0.817
Truck	0.592
Van	0.290

Table: Class-wise mAP@50 for fine-tuned YOLOv8x model

Conclusions

- **Effective Model Selection:**
  - The fine-tuned YOLOv8x model demonstrates strong performance in vehicle detection and speed estimation across various weather conditions.
- **Robust Performance Metrics:**
  - Achieved mAP@50 of 0.630 and precision of 0.684 highlight the model's accuracy and reliability.
- **Class-Specific Insights:**
  - High accuracy for buses and cars suggests effective training, while lower metrics for trucks and vans indicate areas for improvement.
- **Future Work:**
  - Explore additional data augmentation techniques and model adjustments to enhance performance for underrepresented vehicle classes.

Plan for Novelty Assessment

- **Automation of Traffic Speed Assessment:**
  - The plan is to introduce an automated approach to traffic speed assessment, moving away from traditional methods such as radar guns and manual data collection.
- **Key Traffic Metrics:**
  - Focuses on essential metrics like design speed (98th percentile), safe speed (85th percentile), median speed (50th percentile), and minimum speed (15th percentile).
- **Efficiency Improvement:**
  - By utilizing advanced object detection and tracking techniques, the project aims to streamline data collection processes, reducing time and resource expenditure.

References

- **Shaqib, S.M., Alo, A.P., Ramit, S.S., Rupak, A.U.H., Khan, S.S., & Rahman, M.S.** (2024). Vehicle Speed Detection System Utilizing YOLOv8: Enhancing Road Safety and Traffic Management for Metropolitan Areas. *arXiv*. `https://arxiv.org/html/2406.07710v1`
- **Parwateeswar Gollapalli.** (2023). Speed Sense: Smart Traffic Analysis with Deep Learning and Machine Learning. *ResearchGate*. `https://www.researchgate.net/publication/379497600_Speed_sense_Smart_traffic_analysis_with_deep_learning_and_machine_learning`
- **Saif B. Neamah & Abdulamir A. Karim.** (2023). Real-time Traffic Monitoring System based on Deep Learning and YOLOv8. *ARO: The Scientific Journal of Koya University*. `https://aro.koyauniversity.org/index.php/aro/article/view/1327/35`

References

- **JuanPedro.** (2024). Detailed Explanation of YOLOv8 Architecture. *Medium*. https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-p
- **Official YOLOv8 Documentation.** (2023). YOLOv8: State-of-the-Art Real-Time Object Detection https://docs.ultralytics.com/models/yolov8/
- **OpenCV Documentation.** (2023). Open Source Computer Vision Library. Accessed: [Date you accessed the link]. https://docs.opencv.org/4.x/
- **Github.** (2024).Github Repository of the project. https://github.com/AbhinavPT/DeepLearningSpeedDetection.git