Visvesvaraya Technological University

XENSENSE-V1: A DEEP LEARNING-BASED FRAMEWORK FOR VIDEO SEGMENTATION AND SEMANTIC LABELLING IN AUTONOMOUS DRIVING SYSTEMS

Submitted by

Arjun Prabhakaran
Ayman Amjad
Bhanoday Kurma
Bhanu Prakash

Guided by

Ms. Mayanka Gupta

Asst. Professor, Dept. of CSE

B.M.S College of Engineering
Department of Computer Science and Engineering
(Autonomous Institution under VTU)
BENGALURU-560019
Academic Year - 2024-25

B.M.S College of Engineering

Department of Computer Science and Engineering
(Autonomous Institution under VTU)
BENGALURU-560019

Certificate

Certified that the project entitled "XenSense-V1: A Deep Learning-Base Framework for Video Segmentation and Semantic Labelling in Autonomous Driving Systems" is a bonafide work carried out by Arjun Prabhakaran(1BM22CS053), Ayman Amjad (1BM22CS061), Bhanoday Kurma (1BM22CS066), Bhanu Prakash (1BM22CS067) in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Techno- logical University, Belagavi during the academic year 2024-25. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Ms. Mayanka Gupta	Dr. Kavitha Sooda	Dr. Bheemsha
Assistant Professor	Professor and Head	Principal

Examiners:

Examiner 1 *Sign:*__________________*Date:*___________________

Examiner 2 *Sign:*___________________*Date:*___________________

B.M.S College of Engineering

Department of Computer Science and Engineering

Declaration

We, **Arjun Prabhakaran(1BM22CS053), Ayman Amjad(1BM22CS061), Bhanoday Kurma(1BM22CS066), Bhanu Prakash(1BMM2CS067)** students of **6th Semester, B.E, Department of COMPUTER SCIENCE AND ENGINEERING**, BMS College of Engineering, Bangalore, hereby declare that, this AAT Project entitled **"XENSENSE-V1: A Deep Learning-Base Framework for Video Segmentation and Semantic Labelling in Autonomous Driving Systems"** has been carried out in the Department of CSE, BMS College of Engineering, Bangalore during the academic semester **Feb 2025 – June 2025**.

We also declare that to the best of our knowledge and belief, the AAT Project report is not part of any other report by any other students.

Student Name   Student Signature

1. Arjun Prabhakaran

2. Ayman Amjad

3. Bhanoday Kurma

4. Bhanu Prakash M

Acknowledgements

We would like to express our deepest appreciation to our supervisors, Asst. Prof. Mayanka Gupta, Asst. Prof. Dr. Shashikala and Asst. Prof Sneha S Bagalkot, whose expertise, understanding, and patience added considerably to my graduate experience. We also thank them for their encouragement and insightful critiques of my research work.

Additionally, we wish to acknowledge the role of my family for their unwavering support and encouragement throughout my study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Lastly, our thanks also go out to my colleagues and the university staff who have been part of my journey. Their continual support has been invaluable.

Abstract

This project, titled **XenSense-V1**, presents a deep learning-based framework designed for real-time video segmentation and semantic labelling in autonomous driving systems. The approach employs convolutional neural networks and advanced video analysis to accurately identify and classify objects such as vehicles, pedestrians, and road signs from sequential video frames.

XenSense-V1 features a streamlined pipeline involving pre-processing, feature extraction, and pixel-level segmentation, with particular attention to challenging scenarios like low-light and occlusions. Evaluations on benchmark datasets demonstrate improved accuracy and real-time performance over traditional methods.

The results suggest that XenSense-V1 can be effectively integrated into autonomous navigation systems, enhancing vehicle perception and contributing to safer, more reliable self-driving technologies.

Contents

List of Figures

List of Tables

List of Acronyms

A2VIS	Amodal Video Instance Segmentation
AGI	Artificial General Intelligence
AI	Artificial Intelligence
BOFP	Bidirectional Occlusion-Guided Feature Propagation
CBAM	Convolutional Block Attention Module
CNN	Convolutional Neural Network
COCOA	Common Objects in Context – Amodal
CRF	Conditional Random Field
CVPR	Conference on Computer Vision and Pattern Recognition
DEVA	Decoupled Video Segmentation for Tracking Anything
DL	Deep Learning
DNN	Deep Neural Network
DR-YOLO	Dehazing and Reasoning YOLO
FPS	Frames Per Second
GUI	Graphical User Interface
HOTA	Higher Order Tracking Accuracy

IDF1	Identification F1 Score
IoU	Intersection over Union
LVOS	Long Video Object Segmentation
mAP	Mean Average Precision
mIoU	Mean Intersection over Union
ML	Machine Learning
MOT	Multi-Object Tracking
NLP	Natural Language Processing
ODFSE	Optical Flow-based Deep Feature Squeeze Estimation
OVIS	Occluded Video Instance Segmentation
ReID	Re-Identification
SAM	Segment Anything Model
SRNet	Space-Time Reinforcement Network
STM	Space-Time Memory Network
TYOLOv8	Temporal YOLOv8
UniVS	Unified Video Segmentation
VIS	Video Instance Segmentation
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Background of the Study

In a variety of applications, including autonomous navigation, traf- fic monitoring, surveillance, and disaster detection, video anal- ysis is essential. Automated video understanding is becoming more and more necessary as the amount of visual data from cameras and Internet of Things devices grows exponentially. Systems are now able to extract structured information from unstructured video streams thanks to techniques like object segmentation and semantic labeling. Furthermore, early emergency response systems are supported by the ability to identify environmental hazards like smoke in video feeds [1, 2]. Real-time detection and tracking of dynamic objects across video frames has been made possible by developments in deep learning and computer vision [3, 4].

1.2 Research Motivation

For large-scale monitoring scenarios, manual video inspection is impractical, time-consuming, and prone to errors. Systems in fields like smart cities, driverless cars, and safety infrastructure depend on the ability to detect moving objects in real time, identify dangerous occurrences like smoke, and comprehend object trajectories. Operational responsiveness and situational awareness are increased when these tasks are integrated into a single framework. The development of intelligent, adaptive, and effective video processing systems is further motivated by the difficulties presented by occlusion, poor visibility, and cluttered environments [5, 6].

1.3 Objective

The purpose of this study is to develop and assess a modular system for semantic labeling and video segmentation with the following goals:

- Identify and label various object classes across video frames with accuracy.
- Recognize the existence and motion of environmental phenomena like smoke.
- Track the position, speed, and direction of objects over time, supporting downstream reasoning.

1.4 Scope

The processing of previously recorded videos in controlled environments for performance assessment and prototyping is the main focus of this study. Although multi-sensor fusion (such as LiDAR or thermal data) and real-time execution are outside the current scope, the framework is modularly designed to accommodate such extensions. The focus is on investigating computationally effective solutions by fusing cutting-edge learning-based methods with traditional computer vision.

1.5 SDG Justification

The Sustainable Development Goals (SDGs) of the UN are in line with the XenSense-V1 project, specifically with:

- **SDG 9: Industry, Innovation and Infrastructure:** By cre- ating real-time, AI-based video segmentation and semantic la- beling systems, this project promotes innovation in intelligent transportation infrastructure. Sustainable industrial solutions for intelligent mobility and traffic management are promoted by the combination of memory-based reasoning, autonomous perception, and hazard detection.

- **SDG 11: Sustainable Cities and Communities:** XenSense- V1 helps create safer urban transportation systems by facili- tating safer, more dependable, and more effective autonomous navigation—particularly in difficult real-world scenarios like occlusion, bad weather, and road imperfections. Due to the system's ability to be deployed on edge devices, expanding urban infrastructures can adopt it widely and affordably.

In addition, the project's emphasis on real-time deployment and modularity facilitates scalable applications in environmental monitoring, public safety, and urban planning, thereby promoting long-term sustainability and resilience in smart city ecosystems.

1.6 Existing System

Present-day video analytics platforms are frequently disjointed; they are either designed for event-specific monitoring (like smoke or anomaly detection) or object detection (like YOLO-based systems). Few solutions combine these tasks, especially when it comes to semantic labeling and real-time object tracking [7, 8]. Furthermore, cutting-edge models frequently have high computational requirements and are not generalizable to unstructured environments. In both academic and commercial settings, hybrid strategies—which combine temporal memory, prompt-guided learn- ing, and effective tracking—remain understudied.

1.7 Proposed System

The suggested system, which draws inspiration from the XenSense-V1 architecture, combines motion tracking, smoke identification, object detection, and semantic labeling using both deep learn- ing and classical techniques. It is implemented in Python using OpenCV and prioritizes computational balance, modularity, and reusability. Even in environments with limited resources, scalable performance is achieved through the use of techniques like background subtraction, feature detection, and spatiotemporal pattern recognition.

1.7.1 System Overview

The following steps make up the algorithm pipeline:

- **Frame extraction and preprocessing**: resizing, denoising, and converting video input to frames.
- **Object segmentation and labeling**: Multi-class object detection using region-based methods.
- **Smoke detection**: Using color and temporal cues, detect motion patterns and textures that resemble smoke.
- **Object tracking**: Centroid flow and motion consistency are used to estimate movement vectors, speed, and direction.

To make debugging, extension, and optimization easier, each module is created as a stand-alone component.

1.8 Work Plan

This report's format is set up to give a thorough and well-organized overview of the project. The comprehensive literature survey presented in Chapter 2 includes domain-wise analysis of previous work, identification of important research gaps, and alignment of the XenSense-V1 objectives with unresolved challenges in the field.

The functional and non-functional requirements, stakeholder roles, system specifications, data handling considerations, and pos- sible limitations, presumptions, and hazards are all defined in Chapter 3.

Chapter 4 elaborates on the architectural blueprint and subsystem breakdown, including the data design, module-wise interaction, and overall system architecture.

The work finished in the current phase is finally compiled in the 5, which also lays the groundwork for the subsequent stages of system development. For academic completeness and traceability, a comprehensive list of cited references is included at the conclusion

Chapter 2

Literature Survey

2.1 Overview

Understanding videos is a fundamental task in autonomous navigation and smart surveillance. Real-time dynamic scene detection, segmentation, and semantic interpretation are all part of it. Se- mantic segmentation, occlusion handling, smoke or fog detection, motion estimation, and road anomaly identification are some of the crucial subtasks that support this goal. In order to set the stage for the suggested XenSense-V1 framework, this chapter examines pertinent research developments in each of these sub-domains.

2.2 Detailed Survey

2.2.1 Video Object Segmentation and Semantic Labeling

Finding and separating significant objects between frames is the goal of video object segmentation. These segmented objects are given descriptive tags by semantic labeling. Recent developments include EPCFormer [9], which uses expression prompts and multi-modal input for real-time referring object segmentation, and Semantic-SAM [6], which introduced part-level granularity using vision transformers.

By combining prompt-based input and open-world adaptability, other models like UniVS [10] and DEVA [8] push the envelope and enable dynamic scene understanding and multitask segmentation. Building adaptable and scalable segmentation modules is inspired by these models.

2.2.2 Detection under Adverse Weather Conditions

Fog, rain, and haze are examples of unfavorable weather conditions that reduce object visibility. A number of models have been created to increase resilience in these situations. While Fog-Guard [1] uses perceptual loss to reinforce detection accuracy, DR-YOLO [11] incorporates a dehazing module. TYOLOv8 [12] and D-YOLO [2] use temporal features and domain adaptation to generalize object detection to various weather conditions.

These methods emphasize the significance of environmental condition awareness, which is essential for safe driving in a variety of weather conditions.

2.2.3 Occlusion-Aware Detection and Amodal Segmentation

Recognizing entire objects, even when they are partially or completely obscured, is known as amodal segmentation. A2VIS [13] uses global prototypes and spatiotemporal priors to infer missing object parts, whereas models such as BOFP [14] employ occlusion-guided bidirectional feature propagation.

These methods serve as the foundation for adding occlusion-awareness to video segmentation frameworks and are useful for anticipating concealed cars or pedestrians in situations with heavy traffic.

2.2.4 Memory-Based Video Perception

To help with current frame segmentation, memory-based video perception frameworks like STM [3] and SRNet [4] keep a buffer of previous frames. While SRNet adds prototype matching with auxiliary memory frames for greater temporal consistency, STM links past and present frame features using space-time attention.

For dependable video comprehension, these memory modules make sure the model maintains scene context even in the event of brief occlusions or camera blindness.

2.2.5 Motion Estimation using Optical Flow

Optical flow analyzes changes between successive frames to estimate an object’s motion. Deep learning-based techniques like ODFSE [17, 18] improve on the widely used classical algorithms like Lucas-Kanade and Farneback by combining semantic features with flow vectors for increased accuracy.

In autonomous navigation, motion estimation is essential for path planning, object trajectory prediction, and speed estimation.

2.2.6 Pothole and Road Defect Detection

Finding irregularities in the road surface, such as potholes and speed bumps, is essential to protecting both the health of the vehicle and

Table 2.1: Summary of Occlusion-Aware and Long-Term Video Models

Paper / Model	**Task**	**Key Feature**	**Dataset(s)**	**Results**
Baselizadeh et al. (2025) OOSIS	Instance Seg. + Occlusion Order	CRF-based joint mask and order labeling	KINS, COCOA	mAP: 21.6, WCS: 77.4
Tran et al. (2025) A2VIS [13]	Amodal VIS + Tracking	Spatiotempora l priors, global object prototypes	SAILVOS	AP: 23.1, IDF1: 25.9, HOTA: 32.1
Baghbaderani et al. (2024) BOFP [14]	Video Semantic Seg.	Bidirectional occlusion-guid ed propagation	Cityscapes, VSPW	mIoU: 76.5, mTC: 70.6
Yue et al. (2023) [15]	Multi Object Tracking	Occlusion modeling with adaptive features	MOT16/17	HOTA: 57.7, MOTA: 57.7
Qi et al. (2022) OVIS [16]	Occluded VIS Occluded VIS Benchmark	Heavy occlusion dataset + stratified AP	OVIS	AP: 16.9, APHO: 4.6

its occupants. To achieve high detection accuracy on custom datasets, Nissimagoudar et al. [19] and Rubin et al. [7] used YOLO-based architectures enhanced with attention modules. These systems show how crucial visual surface analysis is for hazard avoidance and route planning.

2.3 Research Gap Identification

Even though these models are a major advancement, there are still important gaps that drive the suggested work:

- Lack of real-time, unified pipelines that combine segmentation, motion estimation, occlusion handling, and hazard detection.
- Limited adaptability of existing models to unstructured and chaotic road conditions common in India.
- Memory-based models are not modular and require high computational resources, making them unsuitable for edge devices.
- Prompt-based segmentation models lack consistency in the presence of occlusions or ambiguous context.
- Most pothole and weather detection systems are image-based and do not leverage temporal video cues.

2.4 Mapping of Your Objective with Research Gap Identification

These drawbacks are intended to be addressed by the suggested XenSense-V1 system, which provides a comprehensive autonomous perception solution:

- **Unified Architecture:** Motion tracking, anomaly detection, and semantic segmentation will all be integrated into a single, real-time video analysis pipeline with XenSense-V1.

- **Modular Integration of Memory:** In order to manage transient occlusion and preserve context, we intend to deploy a lightweight memory recall module that draws inspiration from STM and SRNet.
- **Ready Edge Deployment:** Compact models (such as MobileNet or pruned YOLOv8) will be used to optimize the system for platforms such as Jetson Nano.
- **Detection with Temporal Awareness:** In order to increase the accuracy of weather and pothole detection beyond single-frame snapshots, our design integrates temporal filters and flow cues.
- **Adaptation to Indian Context:** Every component, including synthetic occlusion and weather scenarios if needed, will be tested using data that reflects the unpredictability of Indian roads.

Chapter 3

Requirement Analysis and Specification

3.1 Stakeholders

3.1.1 Internal Stakeholders

The internal stakeholders involved in the design and implementation of the XenSense-V1 system include:

- **Development Team:** Responsible for building and integrating components such as segmentation, tracking, and smoke detection.
- **Academic Supervisors:** Guide the project direction and validate progress against research goals.
- **Testers:** Validate system performance across modules and ensure adherence to functional requirements.

3.1.2 External Stakeholders

These are beneficiaries or users of the system once deployed:

- **Urban Planners and Traffic Authorities:** Can use the framework to monitor and analyze traffic environments.
- **Autonomous Vehicle Developers:** May adopt the architecture for deployment in low-resource automotive hardware.
- **Research Community:** Interested in benchmarking or extending the model.

3.1.3 Roles and Responsibilities

Role	**Responsibility**
Developer	Integrate detection, segmentation, and tracking models.
Supervisor	Validate design goals and academic rigor.
End User	Provide feedback on system performance in real environments.

Table 3.1: Stakeholder Roles and Responsibilities

3.2 Functional Requirements

- Use YOLOv8 and transformer backbones [6, 9] to segment and detect vehicles, pedestrians, obstacles, and smoke.
- Perform semantic labeling using prompt-driven architectures [8, 10].
- Estimate speed/direction with optical flow and Deep SORT [17, 18].
- Detect smoke/fog using CNN-based classifiers [1].
- Display bounding boxes, labels, and motion overlays.
- Enable modular toggling of functionalities.

3.3 Non-Functional Requirements

- **Real-Time:** Achieve 20 FPS on HD.
- **Scalable:** Allow modular plug-and-play model integration.
- **Portable:** Deploy on RTX GPUs and Jetson Nano.
- **Reliable:** Long-term runtime with no crashes.
- **Accurate:** mIoU greater than 70 percent, IDF1 greater than 60, Smoke accuracy greater than 85 percent [2, 3].
- **User-Friendly:** Clean, togglable GUI.

3.4 System Requirements

3.4.1 Hardware Requirements

- Development**:** RTX 3080 GPU.
- Deployment**:** Jetson Nano/Xavier, Coral TPU.
- Optional Input**:** HD webcam or CSI camera.

3.4.2 Software Requirements

- OS: Ubuntu 20.04 / Windows 10
- Language: Python 3.9+
- Libraries: PyTorch/TensorFlow, OpenCV, YOLOv8, Deep SORT, CUDA/cuDNN
- Tools: VS Code, Jupyter

3.5 Data Requirements

- **Input:** Live/recorded video for segmentation, detection, tracking, and condition recognition.
- **Output:** Annotated frames with bounding boxes, labels, motion vectors, and hazard indicators.

3.6 Constraints and Limitations

- **Hardware Budget:** Limited to open-source tools and affordable edge devices.
- **Computational Load:** Transformer models require GPU acceleration.
- **Dataset Scope:** Limited availability of India-specific annotated datasets.
- **Deployment Timeframe:** Bound by academic calendar.

3.7 Assumptions

- Public models and datasets will remain available.
- Developers have access to required hardware and libraries.
- Users will provide video input via webcam or file.
- Ground truth labels from benchmark datasets are reliable for model tuning.

3.8 Risks and Challenges

3.8.1 Technical Risks

- Heavyweight models may fail on Jetson Nano.
- Visual features may be insufficient for accurate smoke detection in all cases.

3.8.2 Operational Risks

- Difficulty in accessing real-world driving videos from Indian roads.
- Feature toggling modules may introduce synchronization bugs.

3.8.3 Mitigation Strategies

Risk	Mitigation Strategy
Edge inference failure Limited data availability Module conflicts	Compress model or switch to quantized version Augment using synthetic occlusions/weather Adopt unit testing for individual toggles

Table 3.2: Risk Mitigation Strategies

Chapter 4

High-Level Design

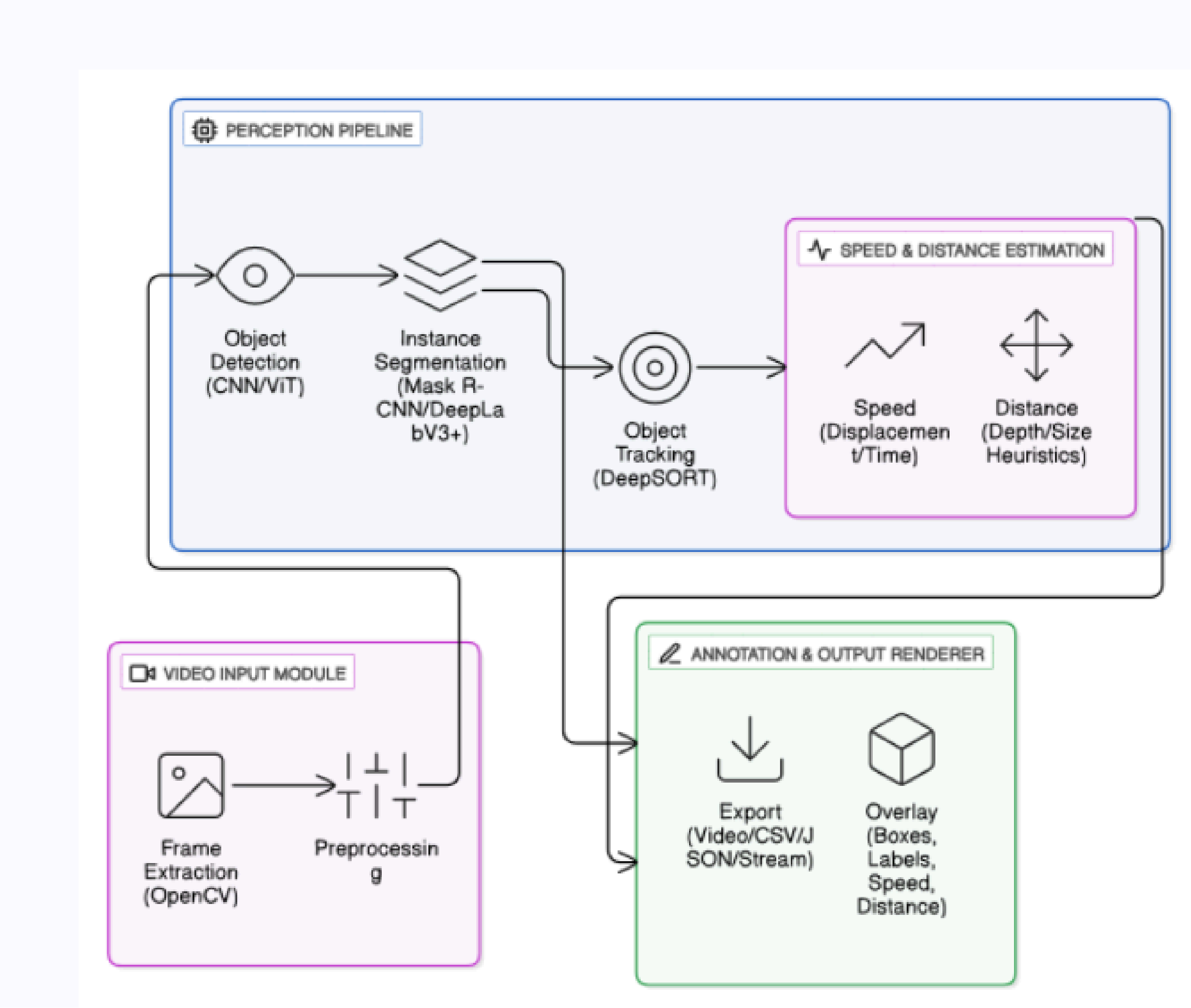
4.1 System Architecture Overview

Real-time, modular video analysis for autonomous driving and smart surveillance settings is made possible by the XenSense-V1 architecture. High throughput, effective visual perception, and the smooth integration of essential elements like segmentation, tracking, condition detection, and motion analysis are the main goals of the architecture.

A multi-stage pipeline is used by the core system to process input in the form of recorded or live video feeds. First, a YOLOv8 model, which is renowned for its accuracy and speed, is used for object detection and instance segmentation. The segmented frames are then divided into parallel processing units, including additional modules for identifying potholes and surface anomalies, classifiers for smoke and fog detection, and DeepSORT for tracking. The last phase entails real-time visual annotation and rendering of insights onto the video frames.

The architecture was created with extensibility and modularity in mind. Future scaling, like adding LIDAR data, prompt-based vision, or multi-sensor fusion, depends on developers being able to quickly add, swap out, or deactivate particular modules.

Figure 4.1: System Architecture: High-Level Design of the Video Processing Pipeline


PERCEPTION PIPELINE
Object Detection (CNN/ViT)
Instance Segmentation (Mask R-CNN/DeepLabV3+)
Object Tracking (DeepSORT)
SPEED & DISTANCE ESTIMATION
Speed (Displacement/Time)
Distance (Depth/Size Heuristics)
VIDEO INPUT MODULE
Frame Extraction (OpenCV)
Preprocessing
ANNOTATION & OUTPUT RENDERER
Export (Video/CSV/JSON/Stream)
Overlay (Boxes, Labels, Speed, Distance)

4.2 Module/Component Design

The proposed XenSense-V1 system comprises the following interdependent components:

- **YOLOv8 Segmentation Module:** In charge of identifying and classifying objects in every frame, such as cars, pedestrians, and obstructions. Using transformer-enhanced backbones enhances accuracy and contextual understanding [6, 9]. Confidence scores, class labels, and bounding boxes are all included in the segmentation output.
- **DeepSORT Tracking Module:** Creates motion trajectories by using the bounding box outputs. To assign unique IDs and

guarantee continuity across frames, DeepSORT combines cosine distance-based appearance embeddings with Kalman filtering [18]. This makes it possible to estimate direction and speed and aids in tracking fast-moving or obscured objects.

Environmental Hazard Detection:

  - **Smoke/Fog Detection:** To identify smoke or fog, a lightweight CNN classifier examines temporal patterns and frame-level texture variations. It employs temporal filtering to lower false positives and is trained on datasets such as RESIDE
and RTTS [1].
  - **Pothole and Speed Bump Detection:** An attention-enhanced YOLO variant (CBAM-YOLO) is used to identify surface anomalies, providing high accuracy even in noisy road conditions [7]. Proactive path planning is aided by this feature.

- **Object Analytics Module:**
  - Uses bounding box centroid shifts to calculate speed by tracking pixel displacement across frames. The motion history of DeepSORT is enhanced by this module.
  - Camera-specific calibration matrices or object size priors are used in monocular methods for distance estimation.
- **Output Renderer:** Creates a final visual output by combining the outcomes from every module. Bounding boxes, class labels, speed vectors, condition warnings (such as smoke detected), and trajectory paths for tracked objects are superimposed on top of each frame. Both offline video writing and live preview are supported by the output renderer.

Each of these modules is contained within a separate service with distinct input/output requirements, parallel processing, making future enhancements possible without requiring a complete system redesign.

4.3 Data Design

To meet the demands of real-time processing, XenSense-V1's data design places a strong emphasis on structured flow, consistency, and extensibility. An overview of the main elements of the data architecture is provided below:

1. **Input Data Format:** The system receives live streams from camera modules or raw video input (such as .mp4 or .avi files). For synchronization, frames are extracted at 25–30 frames per second and stored in a buffer with a timestamp and frame index.

2. **Intermediate Data Representations:**

- Bounding boxes, object class, and confidence values are the results of segmentation.
- Tracking outputs include velocity estimates, motion vectors, and object IDs.
- Environmental tags: anomaly coordinates and binary smoke/fog flags.
- Metadata: frame dimensions, processing latency, and timestamps.

By using shared memory or message queues (in multi-threaded configurations) to transfer these representations between modules, asynchronous processing is made possible without bottlenecks.

3. **Output Format:** The finished product is a saved video file or composite video stream that includes:

- Annotated objects with bounding boxes and IDs
- Motion vectors and direction indicators
- Alert tags for hazards (e.g., “Smoke Detected”)
- Summary logs for post-processing analysis

4. **Data Storage and Export:** For analytical purposes, processed results may be optionally saved in CSV or JSON logs. It is also possible to export motion maps and segmentation masks for additional training or visualization. Real-time data streaming to

Cloud-based dashboards for remote monitoring will be supported in future iterations.

The XenSense-V1 system is guaranteed to be not only real-time ready but also furnished for long-term analytical use and modular expansion, thanks to this thorough data flow design.

Chapter 5

Summary

The conceptualization, design, and system-level specification of XenSense-V1, a modular deep learning-based video segmentation and labeling framework meant for real-time deployment in smart surveillance and autonomous driving applications, are presented in this report. The study starts by pointing out the increasing demand for accurate visual perception in difficult situations like occlusion, poor visibility, and uneven roads, particularly in unstructured settings like Indian roads.

The review of the literature covers developments in a number of fields, including road hazard detection, occlusion-resilient modeling, prompt-based semantic labeling, optical flow motion estimation, and video object segmentation. The fragmented nature of existing solutions and the absence of unified, edge-deployable pipelines are ultimately exposed by each section, which highlights key contributions and limitations in the body of existing research.

The project employs a mixed methodology, quantitatively benchmarking findings on publicly available datasets and qualitatively synthesizing scholarly innovations. The architectural innovations and practical limitations of models like STM, SRNet, UniVS, and CBAM-YOLO are identified through a critical analysis. These results guide the development of a modular system that integrates adaptive inference logic, real-time analytics, and memory-based perception in order to be ready for the future.

The high-level design describes a multi-phase architecture that unifies analytics subunits, DeepSORT tracking, smoke and anomaly detection modules, and YOLOv8-based segmentation under a common data pipeline.

Future components like depth estimation, LI- DAR fusion, or dynamic prompting can be easily integrated thanks to clear interface definitions and modular abstraction.

Stakeholder roles, hardware/software dependencies, performance benchmarks, and design constraints are all formally defined in the requirement analysis and specification chapter. On low-resource edge platforms such as Jetson Nano, special attention is paid to attaining scalable, interpretable, and real-time system behavior.

All things considered, this report establishes the fundamental framework for developing XenSense-V1 into a strong and flexible visual perception system. It combines the latest developments in motion tracking, segmentation, and hazard detection into a unified pipeline that is best suited for implementation in high-variability, real-world settings. As autonomous navigation and video comprehension continue to advance, the system's adaptability and strategic design guarantee its continued relevance.

References

[1] S. Gharatappeh, S. Neshatfar, S. Y. Sekeh, and V. Dhiman, "Fogguard: guarding yolo against fog using perceptual loss," *arXiv preprint arXiv:2403.08939*, 2024.

[2] Z. Chu, "D-yolo: A robust framework for object detection in adverse weather conditions," *arXiv preprint arXiv:2403.09233*, 2024.

[3] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9226–9235.

[4] Y. Chen, W. Zhu, Z.-X. Yang, and E. Wu, "Space-time reinforcement network for video object segmentation," *arXiv preprint arXiv:2405.04042*, 2024.

[5] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing, "Putting the object back into video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 3151–3161.

[6] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, C. Li, J. Yang, L. Zhang, and J. Gao, "Semantic-sam: Segment and recognize anything at any granularity," *arXiv preprint arXiv:2307.04767*, 2023.

[7] R. Rubin, C. Jacob, S. Sundar, G. Stoian, D. Danciulescu, and J. Hemanth, "Pothole detection and assessment on highways using enhanced yolo algorithm with attention mechanisms," *Advances in Civil Engineering*, 2025.

[8] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," *arXiv preprint arXiv:2309.03903*, 2023.

[9] J. Chen, J. Lin, G. Zhong, H. Fu, K. Nai, K. Yang, and Z. Li, "Expression prompt collaboration transformer for universal referring video object segmentation," *arXiv preprint arXiv:2308.04162*, 2025.

[10] M. Li, S. Li, X. Zhang, and L. Zhang, "Univs: Unified and universal video segmentation with prompts as queries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[11] F. Zhong, W. Shen, H. Yu, G. Wang, and J. Hu, "Dehazing & reasoning yolo: Prior knowledge-guided network for object detection in foggy weather," *Pattern Recognition*, vol. 156, p. 110756, 2024.

[12] M. V. Lier *et al.*, "Evaluation of spatio-temporal small object detection in real-world adverse weather conditions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[13] J. Zhou, L. Zhang *et al.*, "Amodal video instance segmentation with spatiotemporal priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] R. K. Baghbaderani *et al.*, "Temporally-consistent video semantic segmentation with bidirectional occlusion-guided feature propagation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[15] Y. Yue, Y. Yang *et al.*, “Improving multi-object tracking by full occlusion handle and adaptive feature fusion,” *IET Image Processing*, 2023.

[16] J. Qi, Y. Gao *et al.*, “Ovis: Occluded video instance segmentation—a benchmark,” *International Journal of Computer Vision*, 2022.

[17] A. Papazoglou and V. Ferrari, “Fast object segmentation in unconstrained video,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[18] K. Liu, Y. Ye, X. Li, and Y. Li, “A real-time method to estimate speed of object based on object detection and opti- cal flow calculation,” *Journal of Physics: Conference Series*, 2018.

[19] P. C. Nissimagoudar, S. R. Miskin, V. N. Sali, R. SK, D. SK, G. HM, and N. CI, “Detection of potholes and speed breaker for autonomous vehicles,” *Procedia Computer Science*, vol. 237, pp. 675–682, 2024.