Situation-aware Customized Machine Intelligence for Transportation Safety, Equity, and Resilience

Chenxi Liu

A Dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2024

Reading Committee:
Yinhai Wang, Chair
Edward McCormack
Xuegang Ban

Program Authorized to Offer Degree:
Civil & Environmental Engineering

University of Washington

Abstract

Situation-aware Customized Machine Intelligence for Transportation Safety, Equity, and Resilience

Chenxi Liu

Chair of the Supervisory Committee:

Yinhai Wang

Department of Civil and Environmental Engineering

Urbanization brings significant opportunities for improved quality of life, but it also poses complex challenges in transportation, such as safety, efficiency, equity, and privacy concerns. The widespread deployment of smart city sensors, along with data from mobile devices like onboard sensors and smartphones, has created a substantial ”big data” environment. This dissertation harnesses this vast amount of data to develop a connected and autonomous transportation system that enhances safety, equity, and resilience through the adaptation and customization of machine intelligence. More specifically, the dissertation introduces a three-tiered strategy to customize machine intelligence in transportation scenarios. Firstly, at the data collection stage, it employs cyber-physical collaboration to integrate road environment and situational data into the sensing framework to enhance situational awareness, thereby improving data accuracy and trustworthiness. Secondly, for data processing and modeling, integrated sensing technologies are harnessed, synthesizing inputs from various sensors to provide a detailed and comprehen-

sive understanding of the traffic scene. Finally, at the application level, the dissertation presents a human-machine interaction framework, utilizing advanced communication technologies to design customized traffic warning, control, and management systems responsive to diverse user needs across various scenarios. Additionally, the research enhances the efficiency, utility, reliability, and privacy of machine intelligence by integrating customized systems with cutting-edge distributed computing, extending benefits to a wide array of settings, including under-served rural and low-income areas. In summary, the dissertation introduces an innovative, situation-aware machine intelligence system that utilizes distributed computing technology to deliver real-time, reliable, and personalized traffic services. This system upholds safety, equity, and resilience, ensuring equitable service across the transportation field.

Contents

Listing of figures

List of Tables

Acknowledgments

With profound gratitude, I reflect upon the journey that has culminated in the completion of my Ph.D. program in Transportation Engineering at the University of Washington. This journey has been extraordinary, a confluence of challenges, discoveries, and growth that I could not have navigated without the assistance and support of a constellation of individuals.

First and foremost, I wish to express my deepest gratitude to my advisor, Prof. Yinhai Wang. His continuous guidance, timely encouragement, and unwavering support have been foundational throughout my Ph.D. program. The breadth and depth of his knowledge, coupled with his insightful feedback and passion for transportation engineering, have shaped my research at every turn. Prof. Wang's mentorship has been extraordinary, and I feel privileged to have learned under his tutelage. His generosity in sharing his expertise and his contributions to my academic and personal growth are deeply cherished. He has played a significant role not only in my studies but also in my life. Without him, I could never have achieved the success I have reached today.

In addition, I am grateful to my committee members, Prof. Edward McCormack, Prof. Xuegang (Jeff) Ban, and Prof. Simon Shaolei Du. Their diverse expertise, unique perspectives, and

incisive criticisms have significantly enhanced the quality of my research work. They have provided guidance through the intricacies and complexities of the Ph.D. program, strengthening my resolve and honing my academic focus.

Extending beyond the realm of my committee, I am thankful for the camaraderie, support, and intellectual stimulation provided by my fellow Ph.D. students and colleagues. The enriching discussions and collaborations have rendered my time at the University of Washington an invaluable experience. I extend special gratitude to STAR Lab former members Dr. Hao Frank Yang, Dr. Ruimin Ke, Dr. Ziyuan Pu, Dr. Zhiyong Cui, Dr. Yifan Zhuang, Dr. Meixin Zhu, Dr. Wei Sun, Dr. Dennis Tsai, Mr. Mingjian Fu, Mr. Fengze Yang and current members Dr. Muhammad Monjurul KarimMs, Dr. Shuyi Yin, Mr. Cole Kopca, Mr. Ollie Wiesner, Ms. Nutvara Jantarathaneewat, Mr. Bingzhang Wang, Ms. Yan Shi, Mr. Shucheng Zhang, Ms. Sruangsaeng Chaikasetsin. Furthermore, I would like to extend my gratitude to my collaborators and friends, Dr. Xiangyang Guan, Dr. Xi Zhu, Dr. Feilong Wang, Ms. Yiran Zhang, and Dr. Qiangqiang Guo, as well as all my friends and collaborators outside of STAR Lab. Their insightful feedback, generous assistance, and active support on research projects and coursework have been pivotal to the successful completion of my journey.

I would like to express my sincerest gratitude to my family, whose unwavering love, support, and encouragement have been the cornerstone of my entire program. Their steadfast faith, sacrifice, and understanding during this challenging period have kept me motivated and inspired. Most importantly, I extend my deepest gratitude to my beloved wife, Lu Lingjiu. Despite being in a long-distance relationship for six years, her endless love, encouragement, and support have been my beacon, inspiring me and giving me strength during the most difficult moments. I would also like to thank my father, Zhihong Liu, and mother, Linli Zhang, who have been my

strongest support, enabling me to bravely move forward step by step on the path of exploration.

As I reflect on the broader context of my journey, I owe a significant debt of gratitude to several funding agencies and organizations that played a critical role in my Ph.D. program. Among them, the National Science Foundation, the Federal Highway Administration, the Washington State Department of Transportation, and the Pacific Northwest Transportation Consortium stand out for their substantial support. Their financial backing made it feasible for me to delve into my research work, participate in a range of conferences, and accumulate a wealth of invaluable experience within the diverse and dynamic research community. Simultaneously, I wish to express my immense appreciation to the University of Washington. The institution has not only provided an exceptional academic environment conducive to rigorous study and innovative thinking but has also granted me the opportunity to pursue my passion for transportation engineering fervently. The support I received throughout this journey was unwavering and instrumental in shaping my academic and personal growth. The roles played by my advisor, committee members, staff, colleagues, funding agencies, family, and friends have been immeasurable, each contributing uniquely to my journey. Their support echoed in every academic milestone I achieved and resonated in each personal development I experienced.

At present, as I stand on the threshold of this new phase in my life, the anticipation is both overwhelming and exhilarating. My role as a researcher, transportation engineer, and lifelong learner is set to evolve and expand, and I am eager to engage with new challenges. I look forward to continuing my journey, a journey replete with uncertainties and promises, and I am keen on making meaningful contributions to the field. The future beckons, and I am ready to embrace it, fortified by the knowledge and experiences I gained during my time at the University of Washington.

1

Chapter 1. Introduction

1.1 Research Background

1.1.1 Existing Challenges in Transportation Systems

Urbanization, characterized by the exponential growth of urban populations, has led to an unprecedented strain on transportation infrastructure, resulting in serious safety challenges, mobility constraints, and equity disparities within existing transportation systems. Congested roads

and outdated infrastructure have fueled a concerning surge in accidents and fatalities, with traffic fatalities in the United States reaching 38,824 in 2019, according to the National Highway Traffic Safety Administration (NHTSA) [4]. Active transportation users and vulnerable road users face heightened safety risks due to inadequate infrastructure and lack of protective measures, exacerbated by extreme weather conditions such as heavy snow, rain, or fog [5]. Under-served regions and low-income communities often bear the brunt of transportation safety challenges, experiencing disproportionately higher rates of accidents and fatalities.

Resilience emerges as a paramount challenge within existing transportation systems, underscored by the growing disparity between infrastructure supply and escalating traffic demands [6]. The strain on transportation networks exacerbates vulnerabilities, particularly during periods of increased usage or adverse weather conditions. Chronic traffic congestion due to unbalanced infrastructure supply compromises the reliability and resilience of urban mobility, with inadequate capacity to accommodate fluctuating traffic volumes heightening the risk of system failures and disruptions [7]. Aging infrastructure and inadequate maintenance further leave transportation networks susceptible to damage and prolonged disruptions, emphasizing the need for strategic investments in infrastructure upgrades, capacity expansion, and disaster preparedness measures to ensure continuity of service.

Equity remains a persistent and pressing challenge within transportation systems, as marginalized communities disproportionately bear the burden of inadequate access to transportation services, further exacerbating existing social and economic disparities. Limited affordability and reliability of transportation options create barriers to accessing essential opportunities such as employment, education, and healthcare, perpetuating socioeconomic inequalities [8]. Low-income individuals, people with disabilities, and communities of color face disproportionate

challenges in accessing transportation services, perpetuating systemic inequities [9]. Addressing these equity concerns necessitates targeted interventions to enhance accessibility, affordability, and inclusivity in transportation planning and policymaking, ensuring that all members of society benefit from the opportunities presented by urbanization.

1.1.2 Machine Intelligence in Intelligent Transportation Systems

To tackle safety, resilience, and equity challenges, Intelligent Transportation Systems (ITS) have been introduced, harnessing cutting-edge technologies for transportation applications [10]. Emerging in the late 20th century, ITS has played a pivotal role in addressing concerns such as traffic congestion, safety risks, and environmental impacts [11]. Initially, ITS efforts were focused on developing systems for traffic management, electronic toll collection, and real-time traveler information, laying the groundwork for modern transportation technologies. In recent years, propelled by advancements in multi-modal data inputs and sensing technologies, the scope of ITS has expanded significantly, encompassing a diverse array of innovative applications aimed at addressing complex challenges such as real-time control and prediction within transportation systems [12, 13]. Leveraging the power of big data, machine intelligence has emerged as a critical enabler within ITS, empowering systems to analyze vast amounts of data, extract valuable insights, and optimize transportation operations in a smart, efficient, and strategic manner. Within the transportation domain, machine intelligence is defined as an advanced form of computing that enables machines or devices to interact intelligently with their environment, allowing them to take actions that maximize the likelihood of successfully achieving their objectives. The conceptual framework of machine intelligence in ITS is illustrated in Figure 1.1.

The machine intelligence defined in this thesis can be separated into four key components,

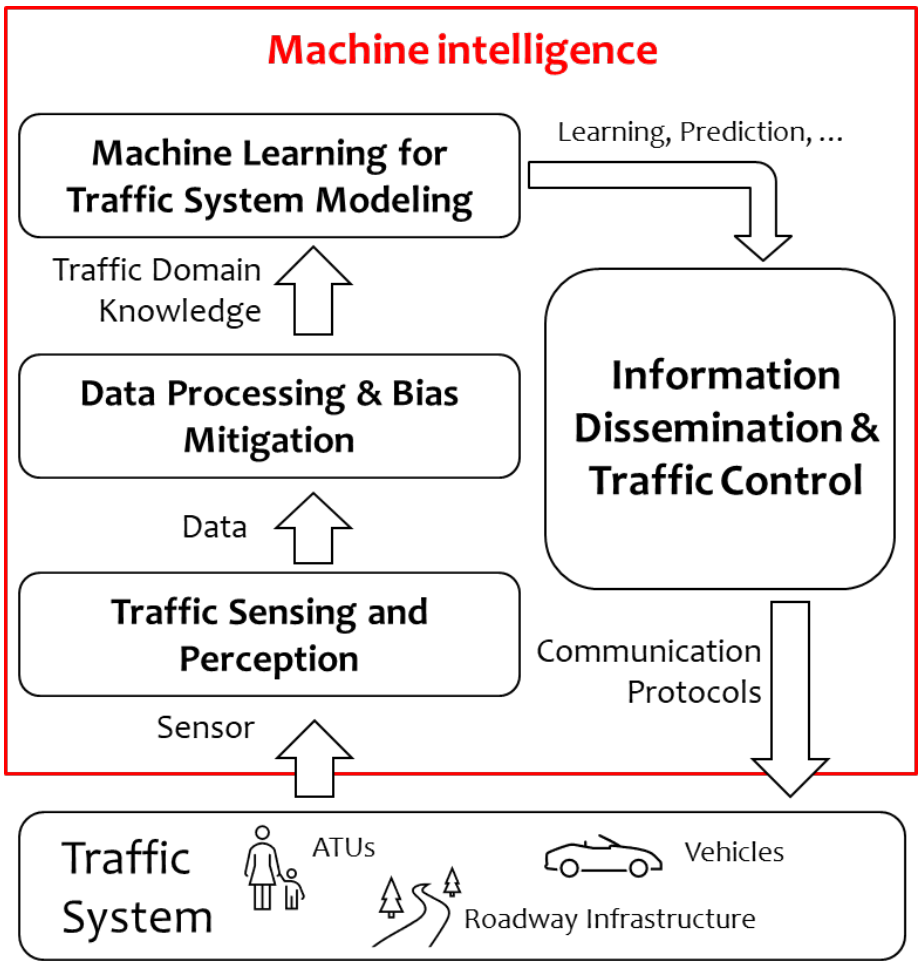

Figure 1.1: The Architecture of Machine Intelligence in Transportation Community

including 1) Traffic Sensing and Perception; 2) Data Preprocessing and Bias Mitigation; 3) Machine Learning and Traffic Modeling; and 4) Information Dissemination and Traffic Control:

- **Traffic Sensing and Perception**: This module harnesses cutting-edge technologies to create a comprehensive sensory network capable of capturing detailed, real-time data on traffic conditions. Advanced sensors, such as LiDAR, radar, surveillance cameras, and acoustic sensors, are strategically deployed to collect diverse data types like image data and point clouds. This real-time sensing is supplemented with data from connected vehicles and mobile devices, which provide additional layers of information through vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications. The collected data serves as the eyes of the ITS, enabling the system to accurately monitor, assess, and react to evolving traffic situations, thereby enhancing responsiveness to emergencies and improving overall traffic management.

- **Data Pre-processing and Bias Mitigation**: This crucial module processes the raw data collected from various sensors to prepare it for analysis. The preprocessing steps involve data cleaning (removing or correcting corrupted or inaccurate records), data integration (combining data from various sources into a coherent dataset), and normalization (scaling inputs to ensure consistent data interpretation across different systems). Bias mitigation is particularly important, as it ensures the reliability and fairness of traffic predictions and management decisions. Techniques such as rebalancing data sets and applying statistical methods to correct for inherent biases in data collection or algorithmic processing are implemented to ensure that the output is not only accurate but also equitable across different regions and populations.

- **Machine Learning and Traffic Modeling**: In this module, sophisticated machine learning techniques are employed to analyze the processed data and develop predictive models. These models utilize both supervised learning (where the model learns from labeled historical data to predict traffic patterns) and unsupervised learning (which may discover new patterns or anomalies without pre-existing labels). The traffic modeling also incorporates real-time data streams to continuously update and refine the predictions. This allows the ITS to adapt to unexpected changes, such as accidents or unusual traffic buildup, and provides foundations for developing simulations and what-if analyses to forecast future traffic scenarios and plan accordingly.

- **Information Dissemination and Traffic Control**: The final module translates the insights gained from traffic modeling into actionable strategies. This involves dynamically managing traffic flow through automated control systems such as adaptive traffic signals, electronic signage, and congestion pricing mechanisms. Information dissemination is multi-channel, targeting not only traditional platforms like radio and traffic light systems but also modern interfaces such as apps and connected vehicle systems, which provide drivers with real-time updates and rerouting options. This proactive information sharing and traffic management help mitigate traffic congestion, optimize route efficiency, and enhance the responsiveness of emergency services, thereby improving the overall safety and efficiency of the transportation network.

1.2 Motivation and Challenges in Machine Intelligence

With the widespread deployment of advanced sensing technologies and AI-related methods in the field, ITS is increasingly capable of managing complex traffic dynamics and enhancing road

safety through real-time data analysis and adaptive decision-making. Despite the considerable benefits that advancements [14, 15, 16, 17] in machine intelligence have brought to ITS, rating concerns about its reliability and trustworthiness have impeded further development in the system. For example, with the rapid expansion of sensor networks and an increase in sensing tasks, the centralized computing framework struggles to keep pace. Additionally, limited cross-sensor cooperation has severely restricted network-scale data collection and information estimation. The machine learning community faces further challenges with multimodality data formats, unbalanced training data, and issues with algorithm reliability and robustness, which can degrade the ability to generalize these algorithms to real-world application scenarios. Moreover, the quality of data in ITS applications can be compromised by adverse conditions that negatively impact traffic sensing. This thesis summarizes the challenges of machine intelligence in the following three main aspects: 1) **Insufficient physical information**; 2) **Isolated sensors and data**; 3) **Lack of interactions with users**

1.2.1 Negligence of Physical Information

Neglecting physical information in Intelligent Transportation Systems (ITS) presents a complex array of challenges that significantly impair the system's effectiveness and safety [18]. This negligence stems primarily from an over-reliance on models and data that do not adequately reflect the current state of the physical environment [19], leading to decisions that are not optimally informed or adapted to real-world conditions.

Physical information, such as real-time weather conditions [20], road surface status [21], and dynamic changes in traffic patterns [22], is critical for the accurate functioning of ITS. When systems fail to incorporate this data, their ability to make sound decisions is compromised. For

example, sensor technologies crucial for the operation of autonomous vehicles or traffic monitoring may falter under varying environmental conditions like fog or heavy rain, which distort the accuracy of the data collected. Similarly, if ITS does not adapt to the abrupt changes in traffic flow caused by temporary events or accidents, they might not manage traffic efficiently or safely.

Moreover, the physical characteristics of the infrastructure itself, such as road quality [23] and layout variations [24], are often underrepresented in the data models used by ITS. This oversight can lead to navigation errors in autonomous driving systems and inefficient routing in traffic management systems, as the algorithms may not be trained to handle the anomalies of less-maintained roads or temporary road signs.

Finally, the behavior of different road users [25], including pedestrians, cyclists, and different types of vehicles, introduces further complexities. A system that overlooks these variations may misinterpret critical cues necessary to predict and react appropriately to the actions of these users. This can lead to increased risk of accidents and inefficiencies in traffic flow, particularly in urban environments where the interaction between various types of road users is frequent and unpredictable.

In essence, the challenge of neglecting physical information is a multi-dimensional problem that affects all aspects of ITS operation—from perception and decision-making to adaptability and compliance [26]. This issue underscores the critical need for systems that can integrate and interpret physical data accurately and in real-time to ensure the safety, efficiency, and reliability of modern transportation networks.

1.2.2 Isolated Sensing Systems

ITS relies heavily on diverse sensing technologies to monitor and manage traffic flows effectively. However, a significant impediment arises from the fact that many of these sensors operate in isolation, without substantial inter-sensor communication or data fusion [27]. This isolated operation can severely limit the systems' ability to form a holistic understanding of traffic scenarios, leading to compromised decision-making and operational inefficiencies.

One of the primary challenges associated with isolated sensors in ITS is the creation and perpetuation of data silos [28]. Each sensor type typically collects specific types of data, optimized for particular tasks. For instance, cameras provide visual records of the environment [20], radar offers robust distance measurements [29], and LiDAR captures precise three-dimensional information [30]. When these sensors operate independently, the data collected remains underutilized for comprehensive scene analysis, as the integration of this diverse data could significantly enhance the accuracy and reliability of traffic assessments.

Furthermore, the lack of cooperation among different sensor systems can lead to biased data interpretations and analytics [31]. For example, a camera-based system might interpret a scene differently from a radar-based system, leading to conflicting data about the same traffic scenario. This discrepancy can introduce bias, particularly in dynamic and complex environments where the interpretation of traffic elements can be highly variable. The consequences of such biases manifest in reduced efficiency of traffic management and control systems, potentially exacerbating traffic congestion and increasing the risk of accidents.

Another critical impact of sensor isolation is the vulnerability to external factors [32], such as weather conditions [20] or sensor malfunctions [33]. Isolated sensors cannot compensate for each other's weaknesses. For instance, cameras might fail to capture clear imagery in foggy

conditions, whereas radar could provide reliable data. Without a mechanism to integrate these complementary data sources, the overall system's effectiveness is diminished, especially under challenging conditions.

In summary, the deployment of diverse sensors in ITS has vastly improved traffic monitoring and management. However, the full potential of these technologies remains limited by issues related to data isolation, which impedes the development of a unified and accurate view of traffic situations. This challenge continues to affect the operational effectiveness of ITS, highlighting the need for improved handling of sensor data within these complex systems.

1.2.3 Limited User Interactions

In ITS, the engagement between the system and its users is critical for optimizing functionality and enhancing user satisfaction. However, one significant challenge that persists within ITS is the lack of meaningful interactions with users. [34] This disconnect not only hampers the ability of machine intelligence to grasp the true needs and preferences of users but also restricts the system's capacity to offer personalized services and targeted assistance to those in need.

ITS systems are primarily designed to manage and improve traffic efficiency at a macro level, focusing on general traffic flow rather than individual user experience [35, 36]. As a result, these systems often fail to capture detailed user-specific data that could inform more personalized service offerings. Without direct and continuous user feedback, ITS cannot effectively adapt to changing user preferences or identify unique user requirements. [12] This lack of personalization can lead to a one-size-fits-all approach where the nuanced needs of different users, especially vulnerable groups such as the elderly or disabled, might be overlooked or inadequately addressed.

Moreover, the absence of interactive mechanisms within ITS means that the systems are of-

ten unable to learn from user behavior in real-time [37]. Machine learning algorithms depend heavily on diverse and dynamic datasets to refine their predictions and functionalities. Without access to ongoing user interaction data, these algorithms miss out on valuable insights that could enhance their accuracy and relevance [38]. This limitation not only affects the system's ability to evolve and improve over time but also reduces its effectiveness in responding to immediate and specific user needs during critical situations.

The lack of user interaction also poses a challenge in emergency and non-standard situations where user input could significantly alter system response [39, 40]. For instance, in scenarios where road conditions suddenly deteriorate or unexpected events occur, real-time user feedback could be instrumental in recalibrating the system's responses more effectively. Without this input, ITS may continue to operate based on outdated or irrelevant data, potentially compromising both safety and efficiency.

In summary, the limited interaction with users within ITS frameworks stands as a substantial barrier to achieving truly responsive and user-centric transportation solutions. This challenge underscores the need for ITS to incorporate more direct and continuous channels for user feedback, ensuring that the system remains adaptable and responsive to the specific requirements and preferences of its users.

1.3 Research Objectives

1.3.1 Integrating Physical Information for Traffic Sensing

This research aims to significantly enhance Intelligent Transportation Systems (ITS) by developing a situation-aware sensing system that effectively integrates physical information from a variety of transportation scenarios. Current ITS implementations often underperform due to their

inability to incorporate contextual physical data—such as environmental conditions and traffic densities—resulting in inaccuracies and system failures with serious implications for traffic management and safety. This project seeks to bridge these critical gaps by employing advanced machine learning algorithms to develop robust models capable of adapting in real-time to diverse transportation conditions. The objective is to ensure optimal performance under various weather, road, lighting, traffic, and community conditions, thereby enhancing the scalability, reliability, resilience, and accuracy of the system. By achieving this, the research aims to establish a scalable, adaptive ITS infrastructure that can effectively handle complex datasets, leading to marked improvements in traffic management and safety outcomes.

1.3.2 Advancing Sensor Cooperation for Complete Scene Understanding

This research focuses on advancing ITS by developing an integrated sensing system that enables comprehensive scene understanding through enhanced sensor cooperation. The objective is to establish a situation-aware cooperative framework among sensors of different types, locations, and times, leveraging distributed machine intelligence to process and analyze data directly at the edge of the network. This approach not only facilitates a holistic view of traffic scenarios but also customizes machine intelligence to tailor responses specifically to current traffic conditions using edge computing. By decentralizing data processing, this system enhances responsiveness and reduces latency, providing deeper insights and a more comprehensive understanding to both road users and transportation agencies. The integration of edge computing allows for more informed and timely decision-making, leading to optimized traffic management, heightened safety measures, and improved efficiency in daily traffic operations and emergency responses. This research aims to transform sensor data into actionable intelligence, fostering a more connected and

intelligent transportation network.

1.3.3 Enhancing Smart and Connected Traffic Infrastructure Systems

This research aims to enhance machine intelligence in ITS by focusing on the dynamic interactions between the system and users, encompassing both road users and transportation agencies. The objective is to develop a smart and connected traffic infrastructure that integrates user inputs and feedback to tailor machine intelligence for customized services. By leveraging these user interactions, the system can adapt and evolve, offering personalized solutions that meet the specific needs of different user groups. Furthermore, the continual integration of user-generated data enables the machine intelligence to refine its algorithms and improve performance across various transportation scenarios. This approach not only enhances user satisfaction and engagement but also ensures that the system's capabilities are finely tuned to the evolving demands of traffic management and safety. Through this user centering innovation, the project seeks to create a more responsive and adaptive ITS that excels in customization and delivers superior performance in diverse traffic environments.

1.4 Thesis Overview and Contributions

This thesis investigates three key tasks for building smart and connected infrastructure systems with customized machine intelligence aiming at transportation safety, resilience, and equity enhancement. The contributions can be divided into three finished components:

- **Proposing Situation-Aware Sensing Systems:** This research centers on the development of situation-aware sensing systems that integrate physical information from transportation scenarios into the sensing framework, enhancing the reliability, resilience, and ac-

curacy of traffic perception. The thesis unfolds through two primary research areas: 1) Integration of road environmental conditions, including weather and road surface states, to construct an adaptive sensing system capable of performing optimally under various weather conditions. 2) Development of a scale-aware perception system designed to detect pedestrians across varying density conditions, effectively addressing challenges such as occlusion, small object detection, and scale variability.

- **Developing a Novel Multimodal Data Representation Learning System:** This research enhances sensor cooperation for comprehensive traffic scene understanding. It introduces a novel multimodal data representation learning system that allows sensors of various types, placed in different locations and times, to collaborate effectively. The thesis explores two primary research areas: 1) Integration of diverse sensors located at a single point to provide a holistic view of traffic conditions. This includes the assimilation of data on weather conditions, visibility, road surface conditions, traffic volume by type, driving speeds, and other pertinent information to forge a detailed understanding of specific traffic scenes. 2) Synchronization of sensors dispersed across different locations to construct an overarching view of the traffic network. The research introduces an innovative framework designed to extract the appearance and orientation details of vehicles. This facilitates the identification of the same vehicles across multiple cameras, enabling the estimation of travel times and trajectories which are critical for assessing the overall traffic network status.

- **Building Demonstrative Cooperative and Equitable Traffic Infrastructure:** This research focuses on the development of demonstrative, cooperative, and equitable traffic infrastructure systems. The thesis introduces research that utilizes real-time traffic vol-

ume data to implement predictive dynamic reversible lane control. This system aims to optimize the traffic flow within a road segment by dynamically adjusting lane directions based on current traffic conditions. The predictive control model captures and utilizes user reactions and interactions as ground truth data.

Chapter 2 Literature Review Representation learning is a machine learning technique that involves automatically learning useful representations or features from raw data, which can then be used for various downstream tasks such as classification, clustering, and prediction. In the field of transportation, representation learning has become increasingly important in recent years as it has been applied to various tasks such as traffic prediction, pattern aggregation, object detection and tracking. In this chapter, the discussion will be focused on the overview and mechanism of representation learning and their applications in transportation, including the representation learning for spatial-temporal data modeling and forecasting, learning visual & LiDAR representations for traffic object recognition, learning visual representations for cooperative perception and learning multi-Modality data representations for unbalanced data distributions, together with edge artificial intelligence in ITS.

Part I. Adaptive System for Resiliency and Equity Enhancement through Cyber-physical Cooperation

Chapter 3: Real-time Multi-task Environmental Perception System for Traffic Safety Empowered by Edge Artificial Intelligence. Traffic safety, reliability, and resilience are significantly influenced by environmental factors such as visibility, road surface, and weather conditions. Yet, current monitoring methods, including weather stations and onboard environmental sensors, often fall short due to their high costs, significant latency, and limited dissemination.

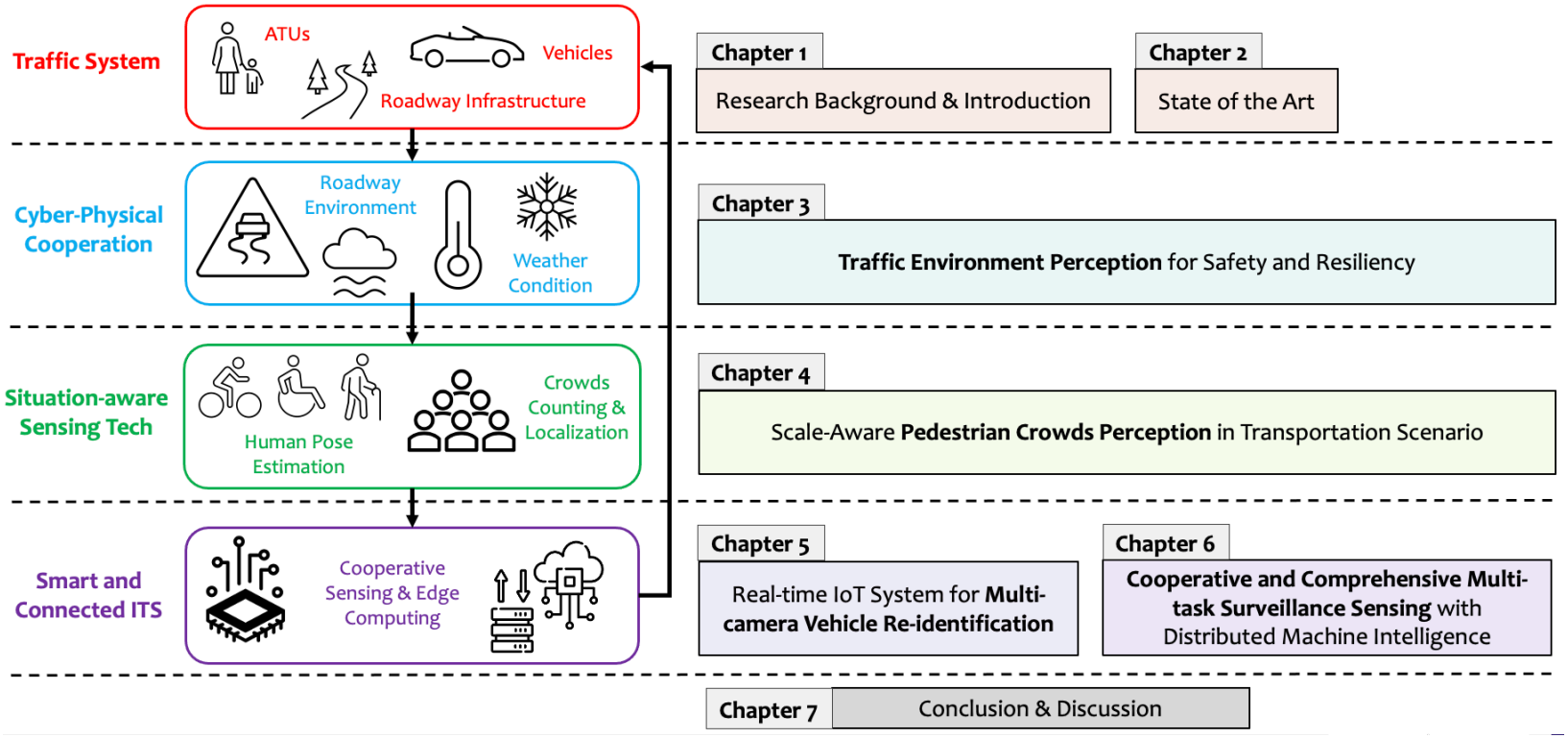

Figure 1.2: Illustration of the Thesis Organization.

This paper presents the Edge-based Multi-task Safety-oriented Environmental (Edge-MuSE) sensing system, designed to address these traffic safety challenges associated with environmental factors. Edge-MuSE departs from traditional single-task sensing methods by performing multi-dimensional traffic environment perception tasks. It estimates key safety-related environmental factors exclusively through camera inputs and incorporates four innovative sensing tasks: visibility estimation, image dehazing, road segmentation, and road surface condition classification. The system is tailored to edge devices to transition computational loads from central servers to distributed nodes, thereby enhancing privacy and reducing latency. Additionally, Edge-MuSE integrates communication functions based on TCP/IP and Wi-Fi protocols, enabling rapid dissemination of sensing results and warning messages to local road users. System structures and data streaming have been optimized to accommodate the constraints of edge devices, ensuring high-efficiency edge computing. Field testing of Edge-MuSE in multiple testbeds in Bellevue (WA, US) and Oslo (Norway) has demonstrated its reliable and precise performance in perception tasks (92.15% accuracy in visibility estimation and 92.25% in road surface condition classification) as well as an impressive processing speed of 21.3 FPS. As such, Edge-MuSE presents a promising solution for enhancing roadway safety, efficiency, and resilience.

Chapter 4: Scale-Aware Representation Learning Empowered Sensing (SARLES) System for Pedestrian Crowds Perception in Complex Transportation Scenarios. Pedestrians are important yet vulnerable users in modern transportation systems, which account for 50% of global fatal road traffic injuries. However, most of the existing sensing models, including detection-based, regression-based, and density estimation methods, are developed based on pre-settings and struggle to cope with complicated transportation scenarios. As a result, the paper proposes an ensemble sensing scheme, Scale-Aware Representation Learning Empowered Sens-

ing (SARLES) system, for detecting and sensing pedestrians in various transportation scenarios. The study addresses the common challenges for accurate pedestrian sensing in transportation scenarios, including occlusion, complex backgrounds, scale variation, diverse distribution, perspective changes, small objects, and blurred regions. The system comprises three innovative modules: Encoder-decoder, Density Map Segmentation & Clustering (DMSC), and Local Patch Refinement (LPR). Firstly, the Encoder-decoder module captures global contextual information and handles the scale variability of pedestrians. Secondly, the DMSC module segments the initial density map into multiple local patches based on the density features and generates patches with stable scale and density distributions. Finally, the LPR module utilizes an ensemble FCN network with various kernel sizes to extract accurate local features from every input patch based on its density and scale features for density map refinements. In summary, the integration of three innovative modules addresses the common challenges, making it a powerful tool for detecting and counting diverse distributed pedestrian groups in complex transportation scenarios. Extensive experiments on three challenging benchmarks indicate superior performance in comparison to the existing state-of-the-art methods.

Part II. Cooperative Sensing Technologies with Distributed Machine Intelligence

Chapter 5: RISTS: Real-time IoT System for Traffic Sensing by Edge Computing and Multi-camera Vehicle Re-identification. Traffic surveillance cameras are like the eyes of the Intelligent Transportation Systems (ITS). However, each camera is isolated and can only extract information from the fixed camera view. To make the cities enjoy the smart surveillance system, the research team proposed RISTS, a Real-time IoT system for traffic sensing. RISTS presents novel

algorithmic and system constructs to push deep learning and multi-camera Re-identification workflow serving for traffic sensing next to IoT devices. On the algorithm side, RISTS proposes a customized edge-based computer vision framework for vehicle detection, tracking, and representation selection in a real-time manner. Then, by only sending objects' representations to the dataset center, the high-bandwidth data transmission and the heavy post-processing system can be abandoned. Furthermore, a customized clip-based vehicle Re-identification pipeline is proposed and integrated into the RISTS, and it significantly outperforms other state-of-the-art methods by 4%-8% on Rank-1 accuracy. Finally, to balance the accuracy level of different camera pairs, a kernel density estimation with kernel smoother is implemented into the traffic information distribution estimation, which can get a precise and reliable result (less than 1.01 KL distance). By maximizing cooperation of the edges and Traffic Management Centers (TMCs) computational resources, orchestrating data transmission, and integrating road network graph features, RISTS can precisely model the network-scale traffic information in a flexible, cost-effective, and easy-scalability workflow, leading to a more advanced and efficient ITS.

Part III. Connected and Smart Infrastructure System for Dynamic Traffic Control

Chapter 6: Cooperative and Comprehensive Multi-task Surveillance Sensing and Interaction System Empowered by Edge Artificial Intelligence. Numerous sensors were introduced to the intelligent transportation system (ITS) in the past decade. As a result, new sensing technologies and the generated data attracted more and more attention, which brought new challenges to ITS like redundant sensors, huge maintenance costs, and data explosion. To satisfy the core demands of traffic agencies in a more effective and efficient way, therefore, the paper

proposes the idea of "Sensing as a Service (SaaS)" and implements it to Cooperative and Comprehensive Smart Edge Node for Sensing and OpeRation (COCO SENSOR) system for practical deployments. COCO-SENSOR is an innovative multi-task sensing system, which is developed to address the key practical applications, including real-time vehicle counting and recognition, road surface condition classification, visibility estimation and live communication among traffic controllers and road users only in one unit. COCO-SENSOR introduced customized cooperative sensing and parallel computation mechanism to increase the perception accuracy and efficiency with limited computation resources on the edge device. In collaboration with the Washington State Department of Transportation and the City of Bellevue, a field experiment was conducted to test the system's performance. The results of the experiment showed that the COCO-SENSOR effectively fills the gap between sensing and traffic services and successfully executed four practical applications, including traffic volume counting by vehicle type, traffic status detection, road visibility estimation, and road surface condition classification, with high accuracy. Additionally, a mobile app was developed for both traffic managers and users to access comprehensive traffic information and live warning messages.

Chapter 7 Final Remarks and Envisioning the Future.

2

Chapter 2. Literature Review

2.1 Machine Intelligence in Transportation Systems

With the rapid development of data science, a lot of sensors are applied in transportation systems, including those embedded in vehicles, infrastructure, and mobile devices [35]. These sensors continuously gather data on various aspects such as traffic flow [41], vehicle speeds [42], road conditions [36], and environmental factors [20]. Additionally, numerous transportation-related mobile apps[43, 44, 45], such as navigation tools and ride-sharing platforms, as well as

social media platforms, generate a vast amount of transportation-related data. This immense volume of data provides a comprehensive, real-time view of the transportation network, essential for effective decision-making and system optimization.

Machine intelligence, defined as advanced computing that enables a machine or device to interact intelligently with its environment, is revolutionizing transportation systems by significantly enhancing their safety, equity, and resilience. By leveraging the data collected from various sources, machine intelligence processes and analyzes this information to extract meaningful insights. These insights drive the development of intelligent systems capable of making autonomous decisions [46], predicting future trends [47], and optimizing operational efficiency [48]. In the transportation field, the introduction of machine intelligence aims to harness the power of big data and translate it into real-world applications that improve traffic safety, ensure equitable access to transportation resources, enhance mobility, and bolster the resilience of transportation networks. For instance, machine intelligence can predict and mitigate traffic congestion [49], optimize public transportation schedules [50], and enhance the reliability and safety of autonomous vehicles. Furthermore, by addressing biases in data processing and ensuring fair representation, machine intelligence helps create more inclusive transportation systems that serve diverse communities equitably [51].

Machine intelligence is implemented in transportation systems from the following four aspects: traffic sensing and perception for data collection, bias mitigation for data pre-processing, machine learning or AI methods for traffic modeling, and information dissemination. Traffic sensing and perception [36, 20] involve the deployment of various sensors to gather real-time data on traffic conditions, vehicle movements, and environmental factors, providing a comprehensive view of the transportation network. Data processing and bias mitigation [52, 53] focus

on cleaning, integrating, and analyzing the collected data while addressing potential biases to ensure accurate and equitable outcomes. Machine learning for traffic system modeling utilizes advanced algorithms to simulate traffic scenarios, predict trends, and optimize traffic management strategies, continuously learning and adapting to new data for improved accuracy and reliability [54, 47, 55]. Communication centers serve as the hubs for disseminating information, integrating data from multiple sources, and providing stakeholders, including transportation agencies, drivers, and pedestrians, with real-time updates and actionable insights to enhance situational awareness and response [35, 9]. Together, these components enable the effective application of machine intelligence in transportation, driving improvements in efficiency, safety, and user experience.

- **Traffic Sensing and Perception for Data Collection**: Traffic sensing and perception are critical for gathering real-time data on traffic conditions [56], vehicle movements [57], and environmental factors [20]. This is achieved through the deployment of advanced sensors such as high-resolution cameras, LiDAR, radar, GPS, and environmental sensors. The comprehensive coverage provided by these sensors allows for continuous data collection, ensuring that transportation systems can monitor traffic flow, vehicle speeds, road conditions, and weather conditions accurately. The effectiveness of this aspect lies in its ability to provide a real-time stream of data, which is crucial for timely decision-making and responsive traffic management. For instance, in smart cities, traffic sensors detect congestion in real-time, enabling dynamic traffic signal adjustments and rerouting of vehicles to prevent bottlenecks and reduce travel time. This capability enhances overall traffic efficiency and contributes to safer and more reliable transportation systems.

- **Bias Mitigation for Data Pre-Processing**: Data processing and bias mitigation are essen-

tial to ensure the accuracy and equity of the data used in transportation systems [52]. Collected data often contains noise, errors, and inconsistencies, which are addressed through data cleaning processes. Integration techniques combine data from multiple sources, providing a unified and reliable dataset for analysis [53]. Bias detection and mitigation techniques, such as re-sampling, re-weighting, and algorithmic adjustments, are implemented to ensure that machine learning models produce fair and equitable results. The performance of this aspect is demonstrated by the ability to create fair and inclusive transportation systems that do not disproportionately impact certain communities or demographic groups [51]. For example, in predictive policing, bias mitigation ensures that traffic enforcement does not unfairly target specific neighborhoods or populations, thereby promoting social equity and trust in the transportation system.

- **Machine Learning or AI Methods for Traffic Modeling**: Machine learning and AI methods are employed to simulate traffic scenarios [49, 58, 59, 60, 54, 47, 61, 62], predict trends, and optimize traffic management strategies. These advanced algorithms create detailed simulations of traffic conditions, considering various factors such as vehicle behavior, road infrastructure, and environmental influences. Predictive models forecast future traffic conditions based on historical and real-time data, allowing transportation agencies to implement proactive measures to mitigate issues such as congestion and accidents. Optimization algorithms enhance traffic management strategies by optimizing traffic signal timings, routing plans, and resource allocation. The continuous learning capability of these models ensures they remain accurate and relevant even as traffic patterns evolve. The effectiveness of this aspect is evident in smart traffic management systems, where AI-driven models predict peak traffic times and dynamically adjust traffic signals

to improve flow, reduce congestion, and enhance road safety.

- **Information Dissemination**: Information dissemination is the process of sharing real-time data and insights with stakeholders, including transportation agencies, drivers, and pedestrians, to enhance situational awareness and response. Communication centers integrate data from various sensors, transportation management systems, and external sources such as weather reports and social media. This integrated data is then disseminated through channels such as traffic management systems, mobile apps, dynamic message signs, and social media platforms. The processed data is transformed into actionable insights, enabling stakeholders to make informed decisions and implement effective strategies. The performance of this aspect is demonstrated by the enhanced situational awareness for all stakeholders, contributing to a more resilient and responsive transportation network. For example, real-time traffic updates provided to drivers through mobile apps help them avoid congested routes and reach their destinations more efficiently, while transportation agencies can promptly respond to incidents, ensuring smoother traffic flow and improved safety.

In summary, these components showcase the performance and effectiveness of machine intelligence in transportation systems, driving improvements in efficiency, safety, and user experience. By leveraging advanced sensing technologies, robust data processing, sophisticated machine learning models, and efficient information dissemination, transportation systems become more intelligent, responsive, and adaptive, ultimately enhancing urban mobility.

2.2 Machine Intelligence for Cyber-Physical Cooperation

Machine intelligence significantly enhances cyber-physical cooperation in transportation systems by integrating advanced sensing technologies and AI algorithms to improve both traffic environment perception and crowd sensing. For example, in traffic environment perception, research has historically focused on the impact of weather conditions on traffic safety, leading to the development of sensors like thermal cameras, intelligent active sensors, and LiDAR systems. These sensors provide real-time data on road and weather conditions, enabling precise identification and differentiation of road types and conditions, such as dry, wet, snow, and ice, through sensor fusion techniques. Similarly, in crowd sensing, detection-based methods using deep learning algorithms like R-CNN and YOLO have become popular for their ability to detect and track pedestrians in complex, high-density scenarios. These methods address challenges such as occlusions and cluttered backgrounds by focusing on local features like head, face, or skeleton, thereby improving detection accuracy. Moreover, edge-based sensing methods, exemplified by the Edge-MuSE system, enhance real-time responsiveness and inclusivity by processing data locally and providing multi-task sensing capabilities, which is especially beneficial for underserved areas. Overall, the implementation of machine intelligence in cyber-physical systems leverages advanced sensing and AI to create a more responsive, efficient, and safer transportation network.

2.2.1 Traffic Environment Perception through Machine Intelligence

Traffic environment sensing has been a cornerstone in transportation safety research for over five decades [63, 35]. Throughout the previous century, a substantial portion of the research was dedicated to understanding the influence of extreme weather conditions on traffic safety

[64] [65]. These studies meticulously outlined critical weather factors impacting traffic safety, which encompassed temperature, rainfall, sunlight, dry spell duration, wind speed, humidity, low visibility, and snowfall. Such research underscored the profound effect of weather on traffic safety, thereby advocating for solutions to tackle associated challenges. As we ventured into the present century, many researchers sought to incorporate weather information from allied fields to gather data on traffic environment for accident analysis and prevention [66, 67]. These pioneering studies yielded significant achievements and paved the way for the adoption of many advanced sensing technologies in the transportation community.

Weather stations, commonly installed by transportation agencies, are a widely used form of traffic environment sensing system [68]. They are equipped with capabilities for monitoring temperature and humidity, forecasting weather, and measuring visibility [69]. The data they collect is publicly accessible and is typically updated every five minutes. However, their main benefits are for transportation agencies in management roles rather than providing direct benefits for road users, owing to high operational costs and deployment challenges. To improve the situation, in recent years, taking advantage of the fast-evolving sensing technologies, more and more advanced sensors have been introduced in the community to provide accurate, timely, and direct environmental information to road users. Thermal cameras [70] can detect temperature differences, which is useful for identifying road surface conditions, but their performance may be compromised in conditions where the temperature is uniformly distributed. Jonson [71] introduced an intelligent active sensor installed on roads to monitor road surface conditions based on freezing point detection. Similarly, sensing systems mounted on vehicle tires have been proposed [72] to estimate road surface conditions via road-tire friction sensing. Kutila et al. [73] proposed a road condition monitoring algorithm that merges a LiARD and a stereo camera.

With sensor fusion techniques, this method achieved a 95% accuracy rate in differentiating between various road types, such as dry, wet, snow, and ice.

The recent advancements in computer vision and AI technologies have introduced image data-based environment sensing methodologies for traffic condition monitoring [25, 74]. Certain research works adopt sensor fusion strategies, integrating inputs from weather information sensing systems and cameras. For instance, Jonsson [75] devised a road condition classification model, harmonizing weather data and road image data, which attained an accuracy exceeding 90%. The primary challenge for such a method lies in reconciling disparate data sources. As a result, these methods rely heavily on the sensors' precise locations, hindering wide deployment due to spatial constraints. To enhance the flexibility of the sensing system, numerous studies proposed methods solely reliant on camera inputs. For example, Mohamed et al. [76] used images from freeway webcams to train detection models based on Convolutional Neural Networks (CNN), a leading-edge deep learning technique, for road condition classification. Similarly, Pan et al. [77] contrasted several deep-learning CNN models, such as ResNet50, Inception-V3, VGG16, and Xception, to address the road surface condition classification problem. In 2021, a modified ResNet18 was proposed [78] for weather and road surface condition detection, achieving accuracy levels of 97% and 99%, respectively. The study emphasized the utility of road surface area segmentation systems in enhancing model performances. Factors derived from images, including the wavelength bands of light reflected from the road surfaces, proved instrumental in classification accuracy. The results underscore the significance of road segmentation in environment sensing, lending further credibility to the proposed Edge-MuSE system.

Previous research collected environment data using various equipment and implemented innovative models for accurate traffic environment sensing. However, some significant limitations

persist. First, many existing sensing methodologies focus solely on single-task sensing, such as road surface condition classification or weather detection. However, traffic safety is influenced by a spectrum of environmental factors, necessitating multi-task sensing for comprehensive traffic environment monitoring and perception. Second, most of these methods are not designed with edge devices in mind, thus failing to meet the latency and reliability requirements of real-time applications. Edge-based sensing methods can process raw data locally, offering results dissemination with minimal latency. Lastly, most current methods cannot support the services to certain demographics, including low-income groups who may be unable to afford sensor costs, and rural areas lacking of internet coverage. To address these three significant limitations, our paper proposes the Edge-MuSE system. It encompasses multi-task sensing, edge-device adaptation, and localized results dissemination, representing a more inclusive and efficient approach to traffic environment sensing.

2.2.2 Scale-aware Machine Intelligence for Crowd Perception

For crowd perception, detection-based methods have gained popularity for pedestrian sensing in transportation applications due to their ability to perform multi-agent sensing, enabling the detection and tracking of various objects in a scene. This is particularly useful in complex transportation scenarios where multiple agents, such as pedestrians, vehicles, and infrastructures, interact with each other, and their interactions need to be studied for traffic safety and efficiency. In 2014, the introduction of Region-based Convolutional Neural Network (R-CNN) [79] accelerated the development of traffic and pedestrian sensing in transportation applications. R-CNN allows for more accurate and efficient object detection in complex scenes through the region suggestion module and the object detection module, which is defined as a two-stage frame-

work. Subsequently, Fast R-CNN [80] and Faster R-CNN [81] algorithms were proposed to further improve detection accuracy and efficiency. However, with the development of intelligent transportation systems, the demand for real-time detection and control has become increasingly important. Light detectors are developed for more efficient video processing, especially on edge devices with limited computing power. One-stage detectors, which directly predict object location through a series of anchors on the feature map, have been proposed for higher processing efficiency. In 2015, the first single one-stage detector, YOLO (You Only Look Once) [82], was introduced, significantly improving processing speed and making real-time video processing possible; its later versions [83, 84, 85] achieved even greater performances in complex scenes for multi-scale object detection. These detectors have enabled more powerful traffic sensing on surveillance systems. However, existing object detection solutions may struggle in high-density, occlusion, cluttered backgrounds, tiny objects, and blurred transportation situations where capturing sufficient features for accurate object detection and classification is challenging. As a result, many methods have been proposed to capture local features of people like head, face, or skeleton to improve the adaptation of detection-based methods to the situations.

Local-feature-based detection methods have proved effective in addressing the above challenges, such as occlusions in pedestrian sensing. One of the popular traditional methods is the scale-space blob detection proposed by [86]. This method is based on the Laplacian of Gaussian (LoG) operator and the Difference of Gaussian (DoG) operator to detect blobs in the image corresponding to the people's heads. Another approach for head detection in crowded scenes is the template matching method, which uses stereo camera inputs [87]. With the recent development of deep learning and AI technologies, CNN-based methods have become popular due to their ability to significantly learn complex features and improve detection accuracy. For instance, Ro-

driguez et al. [88] proposed a density-aware head detection model. In 2013, Sermanet et al. [89] proposed OverFeat, the winner of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013), utilizing a sliding window approach for head localization in crowded scenes. Compared to detecting pedestrians directly, local-feature-based detection methods require fewer features, making the models better equipped to handle complex transportation situations. However, these methods lack global features and spatial information, making it challenging to understand scale and perspective changes in the image. As a result, they generate many false-positive and false-negative detection results in complex scenarios.

In response to the challenges presented by complex and crowded scenes, one promising research direction is to incorporate global and local features into a scale-aware detection algorithm. To achieve this goal, various advanced detectors have been developed. For example, FPN [90] is a multi-scale feature pyramid network that uses feature maps at different scales to detect objects at different scales. Xiaowei et al. [91] propose hybrid convolutional features architecture that extracts and combines information from intermediate layers to support the detection. AugFPN [92] further enhances the multi-scale feature learning ability of the model by narrowing the gap between features at different scales and preserving high-level information. Peng et al. [93] proposed a light decoder for real-time semantic segmentation that uses pooling blocks to fuse multiple feature maps and produce high-quality results. These recently proposed methods have significantly improved the accuracy of detection models based on multi-scale feature extraction and fusion. However, they still struggle with objects that are small or far away from the cameras with few features, which are typically found in higher-density regions. Consequently, there has been a growing interest in tiny object detection methods, which focus on capturing contextual and spatial information from the image rather than directly detecting the object. Xian et

al.[94] collect local and global contextual information to increase the discrimination of tiny object features. Sun et al. [94] collect both local and global contextual information to improve the discrimination of tiny object features. Leng et al. [95] create a Context Reasoning Module to facilitate region proposals to learn challenging objects. Chen et al. [96], Zhang et al. [97], and Sun et al. [98] add extra modules to pass more information into deeper or multi-layers. These approaches demonstrate the potential of incorporating contextual and spatial information into tiny object detection to improve performance in crowded scenarios. However, they still cannot address the challenges of occlusion and cluttered background.

In summary, detection-based methods have been widely used and proven effective in pedestrian detection and sensing, especially in practical transportation applications. These methods detect pedestrians by capturing and classifying their features, which efficiently extracts targets from complex backgrounds and enables real-time sensing of various objects in challenging scenarios. For objects that present sufficient features in the image, detection-based methods can achieve high accuracy detection with low false positive rates. However, there are still some significant challenges when applying these methods to transportation scenarios.

- Firstly, detection-based methods are significantly impacted by occlusion, which is common in busy transportation scenarios such as intersections. Occlusion can result in lack of features for occluded objects, leading to inaccurate detection.

- Secondly, detection-based methods may miss objects located far from the camera, where it is difficult to capture enough features for detection.

- Thirdly, the diverse sizes and scales of transportation components like pedestrians and vehicles in transportation scenarios present a significant challenge for multi-scale object

detection methods. These methods are designed to detect objects at different scales, but the distribution of pedestrians in transportation scenarios may not follow continuous scales, leading to difficulties in capturing their features accurately.

Compared to detection-based methods, density estimation methods are designed to address the challenges of high-density scenarios and directly count pedestrians from the density map. These methods focus on extracting contextual information and successfully overcoming challenges such as occlusion and background clutter in crowded situations. Regression models were first proposed to map the image contextual features to people counting, including global [99] or local features [100, 101] like texture and gradient features from the image, and then matched the features to crowd counting through regression technologies such as linear regression [102] and Gaussian mixture regression [103]. To better understand the spatial information and perspective changes in the image, a series of non-linear mapping approaches have been proposed and achieved great success in the past decade. For example, to handle various features from different regions in the image, Pham et al. [104] introduce multiple random forest models to handle map features for better accuracy. Some methods [105, 106] use multiple SVM and median filters for similar purposes. These methods successfully included spatial information and perspective changes in feature mapping. However, they still use traditional hand-crafted features with low-level information, which are not generalizable.

With the advancement of deep learning and AI technologies, Convolutional Neural Networks (CNNs) have emerged as a popular tool for crowd counting [107, 108] due to their ability to extract hierarchical features from input images. Unlike traditional hand-crafted features, CNN models have significantly improved sensing accuracy through high-level feature extraction. Based on the feature extractors, existing methods can be classified into two categories:

patch-based and image-based. Patch-based methods [109, 110] focus on local features and utilize sliding windows to crop images. These methods extract local features from patches of various sizes, which are then fed to the CNN model for training to predict accurate density maps. However, these methods have difficulty handling images with large perspective or scale changes. To address this challenge, a data-driven method has been proposed [111] to fine-tune the trained CNN model with patches of different scales. Sam et al. [112] proposed a Switch-CNN that can switch between CNNs to adapt to different densities. Compared to patch-based models, image-based models focus on global contextual features to better understand spatial information and perspective changes in the image. Zhang et al. [111] proposed a multi-task deep network that jointly predicts density maps with various densities. Sheng et al. [113] propose a Long-Short Term Memory (LSTM) encoder to handle a series of CNN models adapted to various densities for accurate people counting. Recent developments in AI technologies have led to methods like Contextual Pyramid CNN [114], which encodes both global and local features to generate high-quality density maps. Some researchers [115] generate attention maps in the encoder-decoder network to alleviate the non-uniform distribution issue before generating the density map. These attention maps depict the similarity of all pixel pairs (intra-layer attention map) and encode the relationship between them (inter-layer attention map) [116, 117].

In summary, density estimation methods are promising for detecting and sensing crowded pedestrians through image data. These methods can effectively capture the contextual features and estimate the density of people in a region directly rather than detecting and counting individuals. This makes them well-suited for handling high congestion and cluttered background situations with occlusion. However, they still face some challenges when applying them to complex transportation scenarios.

- The transportation environment is complex, with various objects like vehicles and infrastructure (crosswalks, poles, and traffic cones), which can be easily mistakenly recognized as pedestrians, leading to a large number of false positives.

- Existing density estimation methods are designed to capture continuous changes in densities, scales, and perspectives in images. However, in transportation scenarios, the distribution of pedestrian groups is often discontinuous and diverse. For example, at intersections, a representative transportation scenario, pedestrian groups usually appear in small waiting regions for the red light, while people walking in other directions are going across the road with relatively low density. The two kinds of regions with different densities, scales, and perspectives are fully independent and separated, whose features are hard to be captured by traditional multi-scale methods.

- Density estimation methods struggle in low-density situations, which can be common in transportation scenarios, such as in vehicle lanes where pedestrians are not expected to be present. However, the missing sensing of pedestrians in unexpected regions can result in severe consequences.

2.3 Machine Intelligence for Cooperative Sensing Technologies

2.3.1 Inter-Sensor Cooperation for Cross-Camera Re-Identification

In the traffic area, many surveillance cameras have been installed. It would be advantageous to use these surveillance cameras for traffic information extraction and estimation comparing with other specialized hardware. The data from these cameras have been used extensively to handle vehicle detection problems. Right now, if people want to collect information through different

cameras, a large amount of brute-force human labor work is necessary. However, vehicle Re-ID research has escalated in the past few years, and now they are booming.

Generally, the multi-camera cooperative traffic sensing system includes three cascading components: single-camera multi-object detection, single-camera multi-object tracking, and cross-camera object re-identification. Currently, the deep learning-based approaches are popular for vehicle detection and show promising results. Such models can be divided into two categories, two-stage detector (i.e., Fast R-CNN [80], Faster R-CNN [81] and Mask R-CNN [118]) and single-stage detector (i.e., You Only Look Once (YOLO) [82], Single Shot Detector (SSD) [119]). In general, a two-stage detector can achieve better accuracy with region proposal networks. At the same time, the single-stage algorithms show much faster processing speed and lower false positive error. Considering the balance of edge-capable processing capacity and the real-time detection accuracy, the YOLOv4, TinyYOLOv4 and MobileNet-SSD are propitious and encourage running on edge devices. For single-camera tracking algorithms, the algorithms can be divided into online and offline. Deep SORT [120], and MOANA [121] are well-known as online tracking frameworks with light structure and dependable performance. To achieve more accurate and reliable object tracking results with lower ID switches and better performance in high occlusion areas, Tracklet Net Tracker (TNT) [122] has been proven to be a dependable and high-precision tracking algorithm by many state-of-art frameworks with offline design.

For cooperation perception, object ReID is the fundamental task, which refers to the efforts of associating a particular object across different observations. As for vehicle ReID, the process is to identify and match the target vehicle in different sensors. When a target vehicle appears, vehicle ReID will tell if the vehicle has been observed by other sensors, such as cameras, radars and other wireless sensors. Generally, vehicle ReID methods can be divided into two categories:

sensor-based and vision-based methods [123]. The early-stage vehicle ReID research matches vehicle signatures detected by multiple traffic sensors. The sensor types include magnetic sensors, inductive loop detectors [124], wireless sensors (GPS, RFID, WiFi and Bluetooth MAC address) [125, 126, 127], and even sensor fusion and hybrid methods[128, 129]. So, the vehicle Re-ID technology breaks the ice that each camera installed at different locations works isolated. Besides sensors-based approaches, with the increase in computation power, vision-based methods emerged and have shown a lot of potential. With the vehicle Re-ID, the surveillance cameras can be used together to detect and track the same object at different locations. The emergence and boom of vehicle Re-ID technology are because (1) the increasing public safety and video information extraction needs and (2) the extensive use of surveillance camera networks in the road network, university campuses, parking garages and streets. With the vehicle Re-ID technology, spot a query vehicle or track the vehicle cross multiple cameras in the surveillance networks that can be done accurately and efficiently. In the remaining part of the literature review, I will focus on vision-based approaches, which includes the classic-feature-based methods and deep-feature-based methods.

Visual-based vehicle ReID algorithms based on classical features generally use traditional empirical rules. They extract and identify differentiated features in different images, which are then used to match the same target objects. These traditional features include license plate number, color, texture, size, and the Histogram of Oriented Gradients (HOG). The main advantage of classical feature-based methods is that it is easy to interpret and explain the matching results [130, 131, 132, 133]. However, the accuracy of classical feature-based approaches is limited since traditional features may not be sufficient for vehicle ReID across different cameras. Since different feature extraction approaches may be used for different camera views, the matching

between different features is not a simple linear relationship. When multiple features are used, sophisticated algorithms are needed to fuse them. Also, lots of manual work is needed to label a large number of outline features and key-point features. Currently, traditional feature-based methods have gradually faded away.

The rapid development of Convolutional Neural Networks (CNN) in recent years has dramatically promoted research topics on vehicle recognition. The task of vehicle ReID and retrieval with traffic cameras has always been a challenging subject. The focus of the former researchers tried to extract vehicle features based on the whole image. However, the sizes of vehicles in surveillance cameras are generally not large enough to support these methods, which leads to a bottleneck for vehicle ReID. Therefore, some researchers have started to pay attention to local scales. Commonly used ideas for extracting local features are vehicle key-point localization and region segmentation. Based on the key-point localization and alignment results, some methods extract the features of the key part of the object and make a detailed comparison to achieve good results [130, 134, 135, 136, 137, 138].

Since 2018, metric learning has become more and more popular in vehicle ReID research [139, 140, 141, 142]. Key target of using metric learning for ReID task is to maximize inter-class similarity and minimize intra-class differences. The challenges comes from subtle inter-class differences and significant intra-class differences in vehicles. For example, the same vehicle looks different due to variations in lighting conditions, background, and orientation. Meanwhile, different vehicles with the same brand and color can look similar. Therefore, using appearance features alone may not be enough. To address this, [139, 143] introduced spatial-temporal features and information from roads, routes, trajectories, and vehicle attributes to vehicle ReID research. Specifically, deep networks are used to learn features with the purpose of maximiz-

ing the distance between different classes, while minimizing the distance within the same class. In particular, the triplet constraint is introduced for learning feature embedding, based on the principle that "samples belonging to the same vehicle ID are closer than samples belonging to different IDs." This triplet constraint has been widely used for pedestrian ReID and face recognition tasks. Based on triplet loss, [139] customized the temporal-attention model that fuses the inter-class features (different models, brands, years of manufacture, etc.) as the ranking module to improve the generalization ability of the vehicle representations. Besides, some related works focus on the hybrid features, the combination of deep features, empirical features and related traffic information, and achieve reliable results for vehicle ReID tasks on the public datasets [144].

2.3.2 Intra-Sensor Cooperation for Multi-task Sensing

This section reviews and summarizes the state-of-the-art roadside sensing systems for ITS applications. There is a long history of various types of roadside sensing systems being implemented for purposes such as traffic monitoring, control, enforcement, etc. Transportation agencies operate a traffic management center (TMC) for all transportation-related within the corresponding jurisdiction. Historically, traffic management applications are centralized with the TMCs to coordinate their resources of sensing, processing, and communication technologies [145, 25]. Specifically, the data collected from the roadside sensing systems with diverse sensing technologies such as inductive loop, magnetometer, magnetic, microwave radar, LiDAR, ultrasound, and video detection systems are transmitted to the TMCs. Then, The TMCs aggregate and process the multi-source data in their data centers. Finally, the TMCs deliver services to support traffic operations and applications for safety [74, 146], and mobility enhancement [147, 148, 149].

The advantage of a TMC or a central server is the huge amount of data [150] that it can collect, aggregate, and analyze from different sources to support more accurate decision-making and deliver better services [48]. This is especially true with the wide adoption of complex yet powerful AI algorithms to extract valuable information from roadside sensing systems [151]. However, TMC-based applications still face many challenges, such as high overhead and delays caused by data transmission and heterogeneous data integration [152]. The large amount of data generated from the increasing variety of sensing technologies introduces difficulties for the TMC to process and fully utilize the data for decision-making. The relatively long delay makes TMC-based services struggle to meet the ultra-fast response time requirements of many advanced ITS applications, such as connected and autonomous vehicles, real-time traffic surveillance and warning, short-term traffic prediction, etc. [153, 42, 154]

With the advance of edge computing technology and its clear benefits of low latency and fast response time, high computational efficiency, low bandwidth usage, and privacy, researchers and practitioners in the transportation field have been implementing roadside sensing systems with edge computing technologies. The early stage of deploying edge computing technology for roadside sensing focusing on low-beam LiDAR and traditional image-processing approaches[155, 156, 157]. In recent years, the advance in AI, especially deep learning methods, combined with the rapid development of various sensing technologies greatly accelerated the implementation of edge computing-based ITS applications [158]. Zhou et al. consider edge computing a promising solution to push AI frontier from the cloud to the network edge, or "paving the last mile of AI" [159]. However, one of the key bottlenecks of edge computing is the resource constraints of the edge devices, especially when considering deploying computationally intensive AI models on edge. To address this bottleneck, Song et al. introduced the compression of deep neural net-

works with a three-stage pipeline of pruning, trained quantization, and Huffman coding [160]. Instead of transmitting all the raw data from the roadside sensing systems to the data centers of the TMCs, roadside sensing systems with edge computing capability are able to process data where the data are generated, which can balance the computation load, reduce latency, increase efficiency, lower network bandwidth, and protect privacy. Ferdowsi et al. proposed a novel ITS architecture using edge deep learning to solve many ITS challenges and improve computation, latency, and reliability [161]. Many roadside sensing systems implement CNN-based detection algorithms, such as YOLO-V4 [85] and EfficientNet [162], for traffic data collection and monitoring with high detection accuracy. In addition, there are many studies on lightweight neural network structures to reduce the computational load on the roadside sensing systems while maintaining the detection performance [163][164].

Currently, the majority of roadside sensing systems deploy sensors for individual tasks. For instance, traffic cameras for traffic flow detection, radars for speed detection, and LiDAR for queue and collision detection. While it is recognized that cooperative perception by fusing information from different sensors can increase perception range and accuracy, most studies are about sensor fusion on a single agent. For example, Chen et al. developed a multi-view 3D sensing framework fusing LiDAR point cloud and camera images [165]. In addition, many researchers have been studying multi-agent cooperative perception among connected vehicles, where multiple vehicles share data collected by their onboard sensors such as radar, LiDAR, and camera, via vehicle-to-vehicle (V2V) communication [166][167][168]. Google proposed federated learning in 2016, which is a distributed learning approach that can utilize the computing capabilities of multiple agents for the same sensing task [169]. Nonetheless, research on utilizing roadside sensing systems through vehicle-to-infrastructure (V2I) communication for coopera-

tive perception is still at an early stage. Compared to in-vehicle sensors, roadside sensors can provide an additional field of view, processing resources, communication range, etc. Tsukada et al. proposed a roadside perception unit combining sensors, i.e., LiDAR and camera, and a roadside unit (RSU) for infrastructure-based cooperative perception [170]. Chtourou et al. conducted simulation and analysis to compare vehicles-only with roadside sensors including cooperative perception. The results indicated roadside sensors-based cooperative perception provided up to 8 times more effectively detected objects [171].

Because of the aforementioned issues, existing roadside sensing systems are not well-positioned to fully utilize the large amount of data generated from different sources and support the ITS applications in an effective and efficient manner. Therefore, this paper proposes a well-established application-oriented roadside sensing system with edge computing and cooperative perception capabilities.

2.4 Machine Learning on Edge Artificial Intelligence

Currently, the state-of-the-art development of traffic sensing has always been closely related to IoT technologies. Well-processed summary from the edge detectors present more clear and organized information than raw materials. However, due to the constraints in computing power and communication technology, only limited studies adopted the IoT architecture for traffic sensing. Back in 2014, Jin et al. proposed a Network-Centric IoT architecture with a sensing paradigm used for traffic control and information estimation, which brings the inspiration to researchers using IoT sensors in ITS systems [172]. Li et al. proposed a policy-based secure sensing system, which can hugely improve the safety level and defend the fake alerts generated by attackers [173]. In 2017, Ling et al. proposed an automated object detection algorithm and

fully experimented on the urban surveillance system based on edge computing. Their proposed method help cameras detect object vehicles accurately and can be used to reduce the data volume needed to be transmitted, processed, and managed in the surveillance systems. In 2019, a research group from the University of North Carolina at Chapel Hill proposed a hybrid architecture REVAMP^{2}T using multi-camera pedestrian tracking [174]. In REVAMP^{2}T, each camera is equipped with a computing unit, and hierarchical information extraction and sharing system are made, including single-camera detection, tracking, and multi-camera human Re-ID. This framework achieved network-scale pedestrian tracking with much lower cost and time latency. However, considering real traffic networks, the pedestrians' travel speed, activity range and scale are much lower than vehicles. Ke et al., implemented a hybrid system with edge AI for monitoring parking status using a single camera and achieved promising results [175]. To the author's best knowledge, the authors are pioneers who design and implement video-based hybrid IoT systems for large-scale traffic sensing.

Part I

Adaptive System for Resiliency Enhancement through Cyber-physical Cooperation

3

Chapter 3. Real-time Multi-task Environmental Perception System for Traffic Safety Empowered by Edge Artificial Intelligence

This chapter is modified from the published work:

- C. Liu, H. Yang, M. Zhu, T. Vaa, and Y. Wang*. "Real-time Multi-task Environmental Perception System for Traffic Safety Empowered by Edge Artificial Intelligence", in IEEE Transaction on Intelligent Transportation Systems, 2023. [20]

- C. Liu, H. Yang, R. Ke and Y. Wang*. "Edge-based Automatic Real-time Road Surface Condition Monitoring System (RSCMS) based on Single Monocular Surveillance Camera.", Proceedings of the 102nd Annual Meeting of Transportation Research Board, Washington D.C. USA, Jan. 2023.

- C. Liu, R. Ke, and Y. Wang. "Dark Channel Prior Real-time Visibility Detection Using Monocular Surveillance Cameras". Patent application filed Apr. 15, 2022. [21]

3.1 Challenges and Motivations

Weather conditions always show significant impacts on roadway users, and sometimes the sudden changes are often hard to forecast and can lead to serious safety concerns. Generally, the weather information provided by meteorologists is only 80% accurate within seven days [176]. According to the Federal Highway Administration (FHWA) Report [177], there are over 5,891,000 vehicle crashes yearly. Approximately 21% (1,235,000) of these crashes are adverse weather related, leaving nearly 5,000 people killed and over 418,000 people injured. Further, the report also points out that weather conditions significantly affect driving safety from two perspectives: low visibility and icy/wet/snow-covered road surface conditions. Firstly, low visibility, mainly associated with fog, dust, or smoke, is one of the most hazardous driving safety factors due to its adverse impacts. A study [178] investigated weather-related traffic crashes in Florida and found that low visibility conditions can result in a 32.67% increase in the crash rate. Additionally, they identified that the probability of a crash in fog or smoke is 3.24 times more likely to result in a severe injury and 1.53 times more likely to be a multiple-vehicle crash by odds ratio. Secondly, road surfaces covered by snow, water, or ice can reduce tire friction and extend braking distance, which is a common cause of fatal car crashes. Survey research done in 2009 [179] shows that wet

or icy surface conditions decrease the International Roughness Index (IRI), a significant indicator to evaluate the road surface conditions, by 50%-100%. This could also reduce road level of services (LOS) by two to three levels [180]. Moreover, low visibility and bad road surface conditions usually occur together because of extreme weather conditions, resulting in severe safety hazards.

With the development of Intelligent Transportation System (ITS), many studies have been persistent and have made significant progress in addressing the traffic safety challenges brought by adverse weather [153, 36]. From the perspective of visibility, related research topics like visibility detection and haze removal attracted much attention in the past decades. For example, thermal cameras [181] installed on vehicles can produce a clear view of the surrounding objects for drivers in low visibility conditions. Transportation agencies like the Department of Transportation (DOT) deployed visibility meters along the freeway to monitor the visibility condition in the region. From the road surface condition monitoring perspective, many sensors have been introduced to ITS for this purpose. For example, road friction sensors [72] installed on the wheels can detect the friction on the wheels to determine the road surface condition. And the sensing results can be transmitted to the onboard computers for further processing. Lidar or Radar sensors can estimate the road surface condition through the reflected signal frequency [182]. Rather than install the sensors on the vehicles, some researchers [183, 75] install the sensors directly on the road surface to detect freezing points. The existing sensing technologies and methods can help with traffic environment sensing, however, they cannot achieve the desired effects on traffic safety improvements for the following three reasons:

- **The environmental sensors and algorithms always focus on a single task.** These methods cannot address the traffic safety challenges oriented by complicated environmental

factors. Most of the existing sensing technologies target only one specific sensing task, like visibility meters for visibility estimation, thermal cameras for image de-haze, and friction sensors for road surface condition detection. However, traffic safety improvements are impacted by multiple environmental factors, requiring a multi-task sensing system for comprehensive environmental monitoring and perception.

- **The traditional central-processing architecture fails to meet the latency and reliability requirements.** Nowadays, most of the existing methods transmit the raw data to the back server for central processing, resulting in long latency and low reliability. However, environment perception is a critical sensing task that requires high real-timing performance. In rural areas like mountain roads, weather changes with high frequency. It is necessary to alert drivers before they enter hazardous regions. However, the centralized sensing system cannot provide timely services due to communication and data processing delays. As a result, the sensing results cannot benefit the community in weather-oriented traffic safety improvement.

- **Access to safety information is discriminatory against low-income groups and rural areas.** Information acquisition discrimination remains a significant problem in existing sensing systems. Due to the interactive nature of the transportation system, the safety of every road user is consequential to the traffic system safety. Multiple studies indicate that high-income populations [184] have fewer vehicle accident injuries and deaths. One reason for the huge perception discrimination is primarily disparities in data acquisition, resulting in a lack of affordable and informative traffic data. Advanced onboard sensors [185], like LiDAR, thermal cameras, and tire friction sensors, are too expensive for low-income populations to afford. In addition, the data collected by these sensors only benefit

the vehicles, raising safety and equity concerns for other roadway users due to insufficient perception ability. Public data provided by the government or other organizations do not adequately address this issue. For example, environmental data like temperature and humidity are updated by weather stations at five-minute intervals and provided on the WSDOT website. However, these data lack critical information like the road surface condition of the road segment, leaving drivers inadequately informed. Additionally, the existing weather station system covers limited regions in Washington State, and it is hard to expand the coverage due to its high costs.

This research proposes an Edge-based Multi-task Safety-oriented Environmental (Edge-MuSE) sensing system based on monocular cameras to address weather-oriented traffic safety challenges. Edge-MuSE is a comprehensive environment sensing system that integrates four sub-sensing tasks with only video inputs. Firstly, Edge-MuSE can provide a visibility estimation based on the image or video data captured by the camera sensors. Secondly, Edge-MuSE removes the haze from the original image or video data and reconstructs a haze-free vision for the transportation agents. Thirdly, Edge-MuSE can extract the road segments from the de-hazed image or video data based on the integration of road contour and optical traffic flow. Finally, Edge-MuSE system investigates multiple features, including dark channel value, intensity, color attenuation, and hue disparity values, to identify the light reflection condition and classify the road surface conditions into four categories: dry, wet, snow-covered, and icy.

In Edge-MuSE system, all sensing tasks are deployed on the edge devices, whose perception results can be transmitted to the users with cost-effective, intensive, and reliable local communication protocols with low latency. In the past decade, the quick development of the Internet of Things (IoT) technologies [158] enables the raw data streaming and post-processing on edge

nodes to increase cyber-security and reduce computation loads on the central processor. However, it remains challenging to realize efficient and reliable multi-task sensing on edge devices. To achieve this, the research optimizes the structure of Edge-MuSE for edge computing architecture from two perspectives: 1) Sensing Algorithm: the three sensing tasks are optimized for edge computing to balance the accuracy and efficiency; 2) Multi-task Sensing Architecture: multiple threads are employed in Edge-MuSE to realize parallel computing and make full use of the computation resources.

In summary, the Edge-MuSE system presented in this research offers an edge-based, comprehensive traffic environment sensing solution to enhance transportation safety. The contributions of this research are four-fold:

- **System:** The primary contribution of this research lies in the development of the novel Edge-based Multi-task Safety-oriented Environmental (Edge-MuSE) sensing system. Edge-MuSE incorporates and executes four distinct sensing tasks to offer a comprehensive perception of the traffic environment, enhancing traffic safety measures. A distinguishing feature of Edge-MuSE is its ability to operate solely on video data, thus increasing its adaptability for extensive deployment at a lower cost.

- **Methodology:** Capitalizing on computer vision and AI technologies, Edge-MuSE incorporates four innovative sensing methods: 1) Atmospheric Visibility Estimation, with an average accuracy of 93.17%; 2) Image Dehazing; 3) Road Segmentation, with an accuracy of 92.0%; and 4) Road Surface Classification, exhibiting an accuracy of 93.25%. These methods collectively enhance the comprehensive understanding of the traffic environment.

- **Edge-Adaption:** In this work, the architecture of Edge-MuSE is optimized to accommodate edge computing in two key aspects: 1) Data Streaming Optimization, and 2) Computation Resources Allocation. The refined system is deployed on the Nvidia Jetson Xavier NX device for performance evaluation. The results unveil notable enhancements in processing speed, escalating from 2.3 FPS to 21.3 FPS, and efficiency, improving from 0.106 FPS/Watt to 0.647 FPS/Watt.

- **Implementation:** The research team partnered with the Norwegian Public Roads Administration (NPRA) and the City of Bellevue to establish two testing environments for implementing the Edge-MuSE system. We successfully deployed the system across 13 surveillance cameras, enabling comprehensive environmental sensing to enhance traffic safety. Feedback from the involved agencies indicates that all sensors within the two testbeds effectively accomplished their assigned sensing tasks.

3.2 Multi-task Sensing Technologies

The section introduces the multi-task sensing technologies used in Edge-MuSE. Weather-oriented traffic safety challenges are impacted by many environmental factors like road surface conditions and visibility. Compared to single-task sensing, multi-task sensing methods can integrate various sensing results for comprehensive environment perception. Additionally, multi-task sensing methods can reduce system costs and computation loads by maximizing resource utilization. In the research, Edge-MuSE uses video data as the only input, which can reduce the costs and the computation loads to integrate different environmental sensors.

The architecture of the multi-sensing technologies used in the research for traffic environment perception is shown in Figure 3.1. The sensing technologies used in the research can be divided

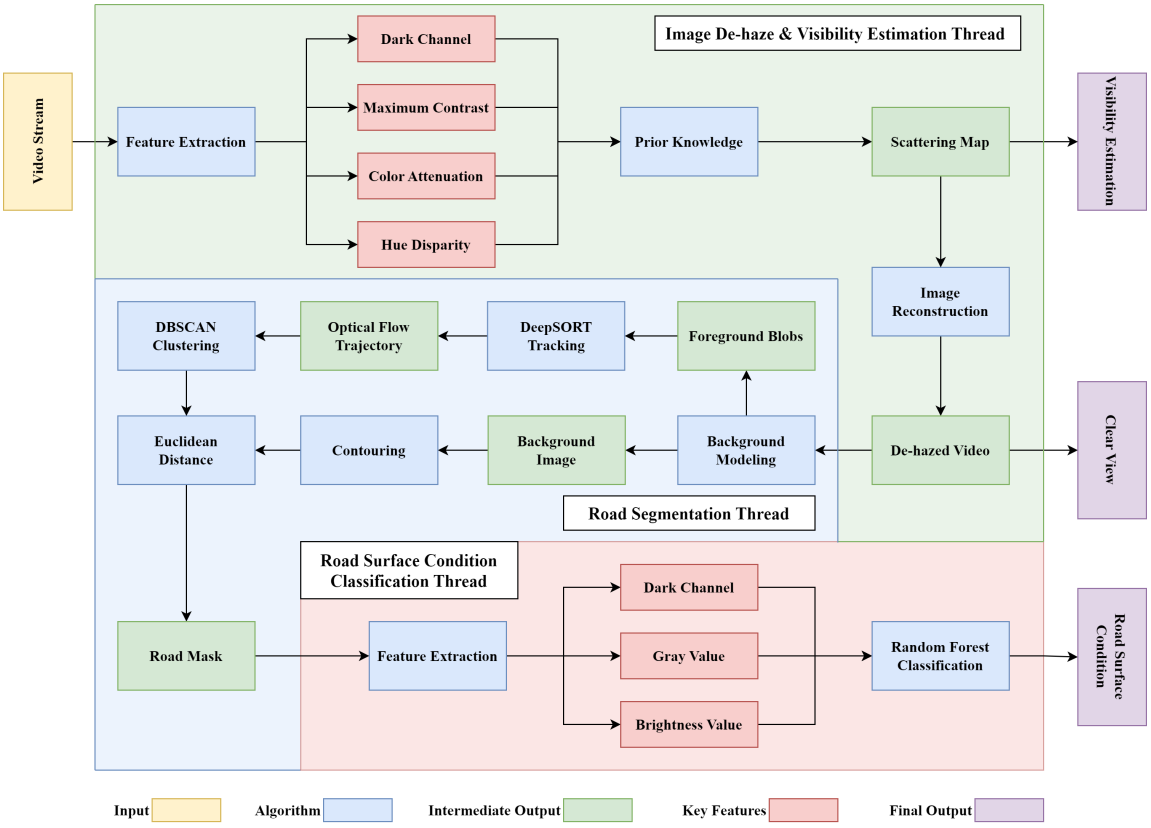

Figure 3.1: Architecture of Edge-MuSE

into three modules: 1) image de-haze, 2) road segmentation and visibility estimation, and 3) road surface condition classification. The three steps are identified through three background colors in Figure 3.1. The following subsections introduce the details of three modules.

3.2.1 Image De-haze & Visibility Estimation

The main task of this step is to estimate the scattering effects caused by the particles in the raw video inputs. To achieve this, the research proposes an innovative feature extraction network to capture the four critical features of the image data: Dark Channel, Maximum Contrast, Color Attenuation, and Hue Disparity. Then, the estimated scattering effects are mapped on the image coordinate to generate a scattering map. Finally, the scattering map can be used to estimate the visibility and reconstruct the haze-free image.

Prior Knowledge

The hazed image captured by the camera $I(x)$ consists of two primary light sources: air light, and light reflected by the surrounding objects. In the research, we assume air light is a homogeneous parallel white light in the scene, noted as A. The actual scene of the object is represented by $J(x)$. For the object captured in the image, the portion of the object reflected light that reaches the camera could be represented as $J(x)t(x)$. Because of the scattering effects, some air lights are scattered to the region having the target object and could be represented as $A(1-t(x))$. Then, the object image captured by the camera is written as:

$$I(x) = J(x)t(x) + A(1-t(x)) \tag{3.1}$$

Where $t(x)$ is the transmission map that can describe the light portion that is not scattered

and reaches the camera. In the research, $t(x)$ is defined as:

$$t(x) = exp(-\beta d(x)) \tag{3.2}$$

Where $d(x)$ is the distance from the scene point to the camera, and β is the scattering coefficient of the atmosphere. Eq.(3.3) suggests that when d goes infinity, the scattering effects t approaches zero. Together with Eq.(3.1), we can get:

$$A = I(x), d(x) \to inf \tag{3.3}$$

In practical imaging of a distance view, d(x) of objects cannot be infinity. However, in a view, the distance from the air light can be regarded as infinity. As a result, together with Eq.(3.3), we have the air light estimation function shown in Eq.(3.4). The air light A is the significant parameter to estimate the depth of image and reconstruct the de-hazed images. More explanations can be found in Section 3.1.3.

$$I_sky(x) = A \tag{3.4}$$

Based on empirical observations, existing image dehazing methods have proposed various critical metrics and prior knowledge for scattering map estimation $t(x)$. The research utilizes four well-proven metrics for haze effects estimation:

- **Dark Channel:** The dark channel is defined as the minimum of all pixel colors in a local patch. The dark channel prior [186] indicates that in the haze-free patches, at least one color channel has a very low and even close to zero intensity value. In other words, if the haze exists, the minimum dark channel value in the patch increases significantly.

Therefore, the dark channel value can be used as a significant indicator in haze removal.

- **Maximum Contrast:** Haze can reduce the contrast of the image. [187] proposed the maximum contrast as the measurement to remove the haze from images. By applying the maximum contrast in the local patch to its neighborhood, the visibility of the image can be enhanced.

- **Color Attenuation:** The color attenuation value is defined as the difference between the brightness value and saturation of the pixel. Color attenuation prior [188] indicates that haze in the image can result in a sharp decrease of saturation value and increase of brightness value. As a result, the difference between the two values can work as the indicator to generate the scattering map.

- **Hue Disparity:** Hue disparity is proposed by [189] for image haze removal. It is defined as the absolute difference between the original image and its semi-inverse value. In most haze-free cases, large hue differences are observed between the two values. And the haze in the image can reduce the difference significantly, making it a good measurement.

Multi-scale Feature Extraction Module

The research designs an innovative feature extraction module to integrate the above four critical metrics for a comprehensive haze removal algorithm. The module's structure is shown in Figure 3.2. The feature extraction process can be divided into four steps.

Firstly, in the convolutional layer (Eq (3.5)), a 5×5 filter is applied to the input image matrix for convolutional operation. The stride of the process is one, and the padding value is zero.

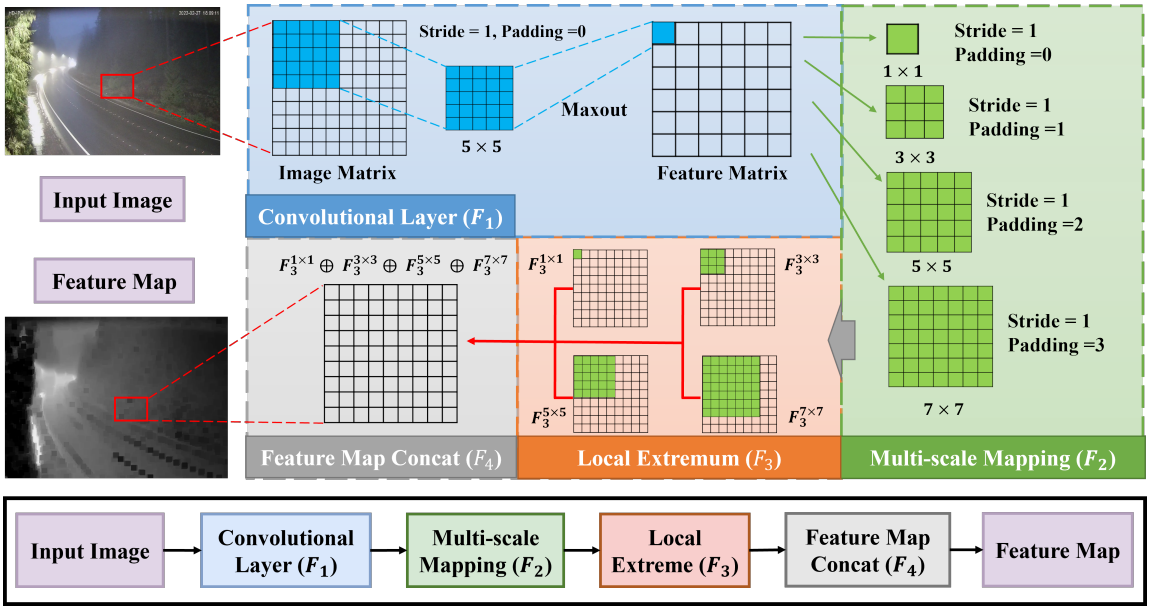

Figure 3.2: Multi-scale Feature Extraction Module Structure

Inspired by [190], The activation function used in the layer is Maxout [191] for non-linear mapping.

$$F_1(x) = \max\{W_1 * I(x) + B_1\} \tag{3.5}$$

Here W_1 and B_1 give the filters and biases respectively. $*$ indicates the convolution operation, and $I(x)$ represents the input image matrix. It is worth mentioning that the filter W_1 is designed for critical features extraction [190]. The three kinds of filters used in the layer are shown in Figure 3.3. The opposite filter (Figure 3.3(a)) with the value of -1 at the center of the kernel is designed for dark channel feature extraction. Cooperated with Maxout activation, the minimum value in three channels of each pixel can be extracted to the feature map. Similarly, the round filter shown in Figure 3.3(b) can capture the intensity difference between the center pixel and surrounding eight pixels, which is identified as visual contrast. Cooperated with Maxout

activation function, the maximum contrast feature can be extracted in output feature map for haze-removal. Finally, if the filter W_1 includes both opposite filter and all-pass filter (Figure 3.3(c)), F_1 is the operation of the color space transformation from RGB to HSV. And then, the color attenuation and hue disparity can be extracted.

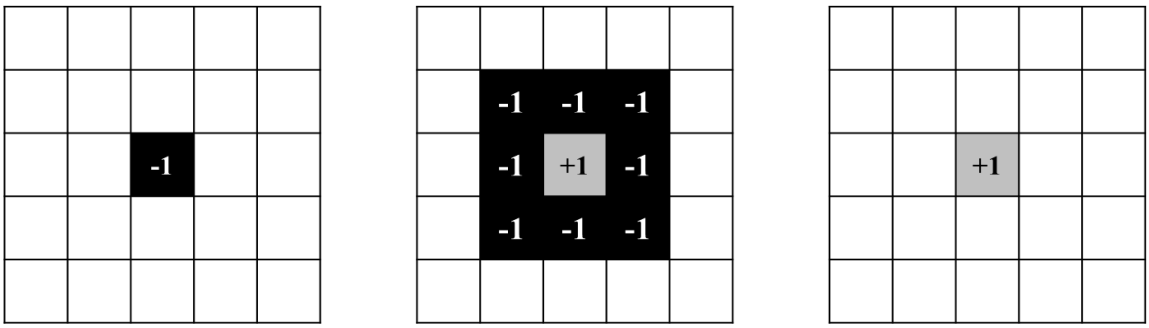

Figure 3.3: Filter Weight in the First Convolutional Layer (F_1)

Secondly, the multi-scale mapping layer is designed to capture the features in different scales. The filters with small kernel size can help the model understand the local features in the patch. And the large-size filters can realize the global features among patches by dropping some details. Some studies have proved that [192] multi-scale features significantly impact haze removal. In Edge-MUsE, we use four filters with various kernel sizes: 1×1, 3×3, 5×5, 7×7. The multi-scale mapping layer can be represented by Eq. (3.6):

$$F_2(x) = W_2 * F_1(x) + B_2, \tag{3.6}$$

with W_2 and B_2 being the weights and biases respectively. Note that the filters used in the layer all follow even distribution. Therefore, the weights in the filters can be represented as $\frac{1}{n \times n}$, where n is the kernel size of the filter ($n = 1, 3, 5, 7$).

The third layer of the feature extraction module (Eq. (3.7) is local extremum. The spatial integration process is an effective method to overcome the local sensitivity and capture the represented features in the patch.

$$F_3(x) = \max_{y \in \Omega(x)} F_2(y), \tag{3.7}$$

where, $\Omega(x)$ is the 12×12 neighborhood centered at x. The max pooling operation are applied to pixels in the feature map, which can preserve the resolution for image reconstruction.

The last layer of feature extraction module, the concatenation layer (Eq. (3.8)), is designed to integrate the multi-scale feature maps.

$$F_4(x) = Avg\{F_3^{1\times1}(x), F_3^{3\times3}(x), F_3^{5\times5}(x), F_3^{7\times7}(x)\} \tag{3.8}$$

The outputs of the feature extraction module are the concated multi-scale feature maps $F_4(x)$ for the four critical features. To integrate the four matrices for the four critical features, the scattering map is introduced in the research to unify the four different feature maps. Based on the prior knowledge mentioned in Section 3.1.1, some pixels can be regarded as haze-free regions to estimate the scattering effects based on the feature maps. For example, dark channel prior [186] assumes that the haze-free pixels should have at least one channel whose intensity value is very low and even close to zero. Therefore, the scattering map of dark channel $t_{dc}(x)$ can be inferred through the normalized dark channel feature map as $1 - F_4^{dc}$. Similarly, the other feature maps can be estimated based on the prior knowledge. The final output scattering map $t(x)$ is the average of all four feature scattering maps:

$$t(x) = Avg\{t_{dc}(x), t_{mc}(x), t_{ca}(x), t_{hd}(x)\}, \tag{3.9}$$

with $t_{dc}(x) = 1 - F_4^{dc}$, $t_{mc}(x) = F_4^{mc}$, $t_{ca}(x) = 1 - F_4^{ca}$, $t_{hd}(x) = 1 - F_4^{hd}$.

De-hazed Image Reconstruction

The estimated scattering map $t(x)$ can work as the input to reconstruct the real scene $J(x)$, which is the haze-free image. Based on Eq.(3.1), $J(x)$ can be represented by Eq.(3.10).

$$J(x) = \frac{I(x)}{t(x)} - \frac{A}{t(x)} + A \tag{3.10}$$

Based on Eq.(3.3) and (3.4), the air light A can be estimated through the pixels where $d \to inf$. And the estimated scattering map $t(x)$ from feature extraction module can infer the depth information in the image through Eq.(3.2). Therefore, in this case, the pixels where $t(x) \to 0$ is regarded as the sky pixels, whose intensity is treated as the atmosphere light A in Eq.(3.10) to reconstruct the haze free scene $J(x)$.

Visibility Estimation

The final step of the thread is to estimate the visibility based on the extracted scattering map $t(x)$. In the research, we define visibility as the distance at which an object or light can be clearly observed, which is measured by visual contrast C_v. Visual Contrast C_v is the relative difference between the light intensity of the background and the object. According to the Beer-Lambert law [193], the visual contrast, $C_v(d)$, can be represented as an exponential function with the single variable, distance d:

$$C_v(d) = exp(-\gamma d), \tag{3.11}$$

where γ is the contrast attenuation coefficient to describe the decrease of visual contracts with the increase of distance d. Based on the standards published by the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) [194], the minimum contrast at the eye of a given observer at which an object can be detected is 2%. Therefore, cooperating with the ground truth visibility data d_0 generated in weather stations, γ can be calculated by the Eq.(3.12), where C_v^t is the visual contrast threshold applied (i.e., 0.02).

$$\gamma = -\frac{\ln(C_v^T)}{d_0} \tag{3.12}$$

Similarly, we can set the median value t_m of the local patch Ω in scattering map $t(x)$ to represent the scattering effects in the view. Based on the definition of scattering map in Eq.(3.2), the scattering coefficient β can be represented as:

$$\beta = -\frac{\ln(Med_{x\in\Omega}(t(x)))}{d_0} \tag{3.13}$$

As a result, to map the scattering map $t(x)$ to contrast map $C_v(x)$ for visibility estimation, the function f between the two coefficient β and γ is necessary.

$$\gamma = f(\beta) \tag{3.14}$$

In the process, the ground-truth visibility data can work as the training dataset to minimize the loss function Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\gamma - f(\beta))^2 \tag{3.15}$$

3.2.2 Road Segmentation

In Edge-MuSE system, road segmentation is used for sensing region determination, which impacts the final detection accuracy significantly. As a result, to ensure road segmentation accuracy, Edge-MuSE integrates two road segmentation algorithms: road contour detection and vehicle motion. The structure of the Road Segmentation Thread is marked as green background in Figure 3.1. Both algorithms have their advantages and disadvantages. The contour detection can be applied in all the scenarios for rapid road segmentation, while its detection accuracy is relatively low because of the impacts of many environmental factors. The optical flow methods can realize a very high accuracy road segmentation based on the accumulation of vehicle trajectory. However, it may take a long time in the low traffic volume scenarios like rural areas. Additionally, some parts of the road like shoulders, work zones which are hardly covered by vehicle trajectories are usually excluded from the estimated road mask. Therefore, Edge-MuSE integrates the two methods through euclidean distance [195] for accurate and efficient road segmentation.

It is important to note that the input data of this thread should be the processed de-hazed video data. Firstly, a background modeling algorithm is applied to the video data for background subtraction. In this step, moving objects (foreground blobs) like moving vehicles, and static objects (background images) like roadways, can be split to eliminate the interference between the two kinds of objects in the following steps. In the process, road contour detection will use this background image as the input, and the foreground blobs can work as the input of the vehicle motion detection method. The following paragraphs will introduce these methods in detail.

Contour Detection

In the first step of this method, the lower bound threshold is set as the minimum moving distance between two consecutive frames to filter the background image and foreground blobs. Then Edge-MuSE employs Canny edge detection algorithm [196] [197] to estimate all contours in the processed static background image. Edge-MuSE does contour detection using the following three steps:

- **Pre-processing:** Compared to standard shape objects in the pure background, real-world objects in complicated backgrounds present more challenges for contour detection. For accurate contour detection, erosion and dilation operations are needed before contour detection can occur. The advantage of these operations can be summarized into two points: (1) they can smooth the contour of the object, break the narrow neck and eliminates thin protrusions; and (2) they can bridge narrow discontinuities and slender gullies, eliminate small holes, and fill up the breaks in the contour line.

- **Image Filtering:** The first step of the Canny algorithm is to smooth the image. Canny estimates the first derivative of the Gaussian function, which is the best approximation of the optimal edge detection operator. Then, a convolution operation is performed on the image matrix. Since the convolution operation possesses both commutative and associative properties, the Canny algorithm usually uses a two-dimensional Gaussian function (shown in Eq.(3.16) to smooth the image and remove noise.

$$G(\mathrm{x},\mathrm{y}) = \exp[-(x^2+y^2)/2\sigma^2]/2\pi\sigma^2 \tag{3.16}$$

- **Image Gradient Calculation**: The second step involves calculating the magnitude and direction of the image gradient. The first order partial derivative's approximation on the X and Y directions can be obtained by:

$$E_x[i,j] = (I[i+1,j] - I[i,j] + I[i+1,j+1] - I[i,j+1])/2$$

$$E_y[i,j] = (I[i,j+1] - I[i,j] + I[i+1,j+1] - I[i+1,j])/2 \tag{3.17}$$

Therefore, the magnitude and direction of gradient is shown in Eq.(3.18):

$$\|M(i,j)\| = \sqrt{E_x[i,j]^2 + E_y[i,j]^2} \tag{3.18}$$

The azimuth of the image gradient can be calculated by Eq. (3.19):

$$\theta(i,j) = \arctan(E_y[i,j]/E_x[i,j]) \tag{3.19}$$

Contour detection is developed in a flexible way that overcomes many challenges. However, because of the complexity involved in many real-world transportation scenarios, many unexpected factors like illumination conditions can influence contour detection if only camera inputs are used. Despite the aforementioned series of preventative actions to reduce the impacts of these factors, both false-negative and false-positive can be observed when contour detection methods are applied to actual data. To address the challenges, Edge-MuSE introduces vehicle motion detection methods.

Optical Flow Detection

The second method used for road segmentation is vehicle motion detection. Its first step is to apply the background modeling algorithm to the input video. Therefore, the moving objects in the view can be extracted from the static background. Based on the foreground blobs, Edge-MuSE extracts the road segment in three steps:

- **Pre-processing:** The extracted foreground blob is the collection of all the moving pixels in the image coordinate. However, many static pixels can be detected as moving points, impacted by many practical factors, such as the camera's shake in the wind. The pre-processing step is aimed at dealing with the false-positive and false-negative pixels. The CANNY algorithm [196] is also introduced into the step for contour detection. The area of regions enclosed by the contours can be used to filter the target objects like vehicles, pedestrians, and cyclists from the foreground blobs. The extracted true-positive regions can work as the inputs to the following tracking algorithm.

- **Optical Flow Extraction:** Edge-MuSE uses Simple Online and Realtime Tracking algorithm (SORT) [198] to track the objects and extract the optical flow. In the first step of SORT, a lower bound threshold is set as the minimum intersection between two consecutive frames to filter the optical flow. Then the pixels that have been marked as the optical flow can be represented by feature vector $\vartheta = [x, y, d, v]$. Here, (x, y) is the object location in the pixel coordination, d is the moving direction, and v is the vehicle speed in the image coordinate.

- **Road Segmentation:** The marked pixels accumulate as time progresses. After time T, the marked pixels will cover the major areas where traffic is present. For road segments

with high traffic volume, T can be just a few minutes, while for rural roadways with low density, a larger T is required to extract all the target regions.

3.2.3 Road Surface Condition Classification

The third thread of Edge-MuSE is Road Surface Classification. The thread input is the road mask extracted from the background images. Then, the feature extraction module is applied to the road mask for feature extraction. The structure of the feature extraction module is the same as the one used for the scattering map $t(x)$ in Section 3.1. However, unlike haze removal, road surface condition detection focuses more on the light reflection status of the pavement. As a result, the key features used in the module are dark channel value, gray value, and brightness value to classify the light reflection status on the road surface. Based on the light reflection status, the road surface condition can be categorized into four classes:

- **Dry:** In the dry condition, the light reflection of the road surface is diffused. This allows light from all parts of the road to reflect into the camera, which results in the values of image features like gray value, dark channel, and brightness to be distributed evenly (i.e., low standard deviation) on the roads. This is the most common status of the road surface.

- **Wet:** In wet conditions, water accumulating on the road creates a flat surface where specular reflection occurs. In this case, only a specific road area reflects light directly to the camera, resulting in sharp spiking of the feature value (e.g., dark channel, intensity, brightness). However, in the rest of the road segments, nearly no light reflects into the camera. As a result, a wide range distribution (i.e., large standard deviation) should be observed in the distributions of the features.

- **Snowy:** Similar to dry conditions, the light reflection on the snowy road surface is diffuse reflection. However, the white snow can reflect more light into the camera than dry road pavement. As a result, even though the feature values distribute evenly on the road, they have a relatively higher value than the dry road conditions.

- **Icy:** Similar to wet conditions, the light reflection on the icy pavement should be specular. However, impacted by passing vehicles, the ice on the road surface is usually mixed with snow and dirt, which is much more closer to diffuse reflection than the wet condition. As a result, in the low-light region, all three feature maps can show diffuse reflection characteristics.

Based on the analysis, the extracted feature maps can act as reliable inputs to the model for road surface condition classification. In Edge-MuSE, the inputs are fed to multiple classification methods, including Random Forest (RF), K-Nearest Neighbors (KNN), Support-Vector Machine (SVM), and Naive Bayes (NB) for road surface condition classification. It is found that RF performs the best and thus deployed in Edge-MuSE as the classifier in the practical deployment.

3.3 System Constructs for Edge-Adaption

Edge-MuSE is a comprehensive traffic environment sensing system deployed on edge devices. The advantages of using edge devices can be summarized in the following three points. Firstly, edge-based systems relocate the computation loads from central servers to the edge, which can reduce the time latency caused by high computation loads on the server side. Secondly, only the sensing results instead of the raw data are transmitted to the server side, which can reduce the time latency and communication costs in data transmission. Additionally, the sensing re-

sults produced by edge devices can be disseminated and serve users with local networks for lower response time. Finally, the privacy-sensitive information can be filtered by edge devices before sending back to the server for private protection and cyber security.

However, due to the limited computing ability of edges, it is still a significant challenge to run the multi-task sensing system effectively on edge. Therefore, this section aims to introduce the systematic design for edge device adaption. On edge devices, many factors could become bottlenecks and result in a decrease in processing speed. The systematic design proposed in the section carefully balances the factors and optimizes the entire system. We introduce the systematic design from the following two aspects: 1) data streaming optimization; 2) computing resources optimization.

3.3.1 Data Stream Optimization on Edge

To optimize the data stream, Edge-MuSE separates the algorithm into five parallel threads, each running independently in pre-defined memory spaces. The interactions of the threads are the data storage and retrieval in five predefined caches, which increase the robustness and efficiency of the system running on edge devices. Figure 3.4 shows the data stream, with each thread introduced below.

1. **Video Streaming Thread:** This thread is designed to capture the video data from camera sensors and pre-process the input data. Different camera sensors have different characteristics, and the video streaming thread can unify the input video data to standard size and FPS for further processing. Additionally, the stream can filter the broken images from the raw data. The thread's input is raw video data, and the output is the filtered and standardized image queue with stable FPS. The output image queue can be stored in Cache

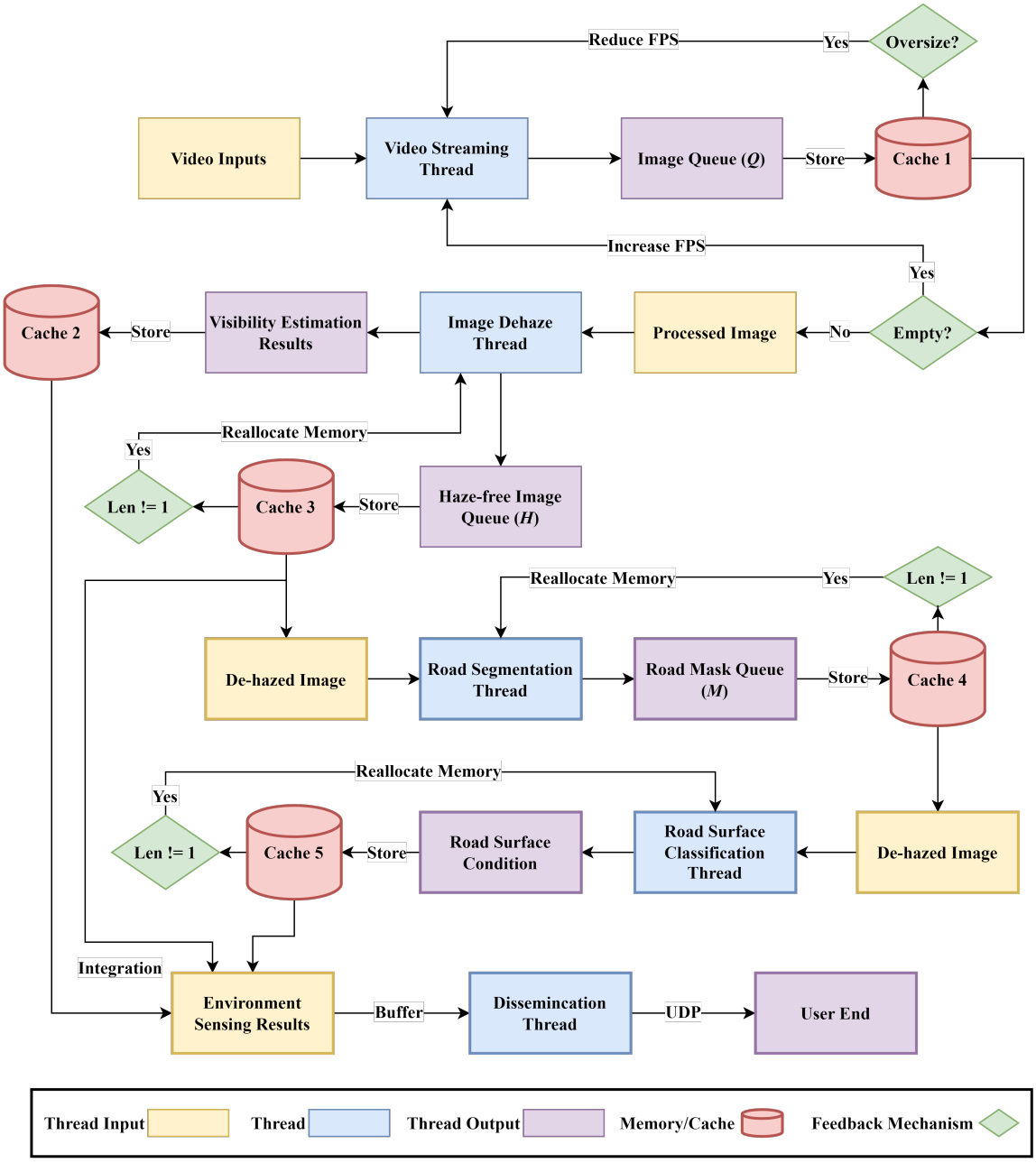

Figure 3.4: Data Streaming Optimization on Edge

1. A feedback mechanism is applied to Cache 1 to check the occupancy status. The check results can feed to the video streaming thread for computation resource adjustments.

2. **Image De-haze Thread:** This thread aims to estimate and remove haze from the image. The input of this thread is the image queue stored in Cache 1. And there are two outputs of the thread: haze-free image queue and visibility estimation results. The visibility estimation results are stored in Cache 2 for dissemination, and the haze-free image queue is stored in Cache 3 for further processing. It is worth mentioning that the feedback mechanism applied to Cache 3 can adjust the memory allocation of the thread to control the processing speed.

3. **Road Segmentation Thread:** Road segmentation Thread is to extract the road mask from the haze-free image queue in Cache 3. For accurate background modeling and optical flow extraction, the batch size is set as 16 in the thread. The output road mask is stored in Cache 4 and updated every iteration. Accordingly, the previous road mask in Cache 4 serves as one of the inputs to the road segmentation thread to increase accuracy.

4. **Road Surface Condition Classification Thread:** This thread can classify the road surface condition into four categories based on the feature extraction module. The input is the latest road mask stored in Cache 4, and the output road surface condition results are stored in Cache 5. Similarly, the feedback mechanism is introduced to adjust the computation resource allocation.

5. **Dissemination Thread:** The input of the threads consists of the de-hazed image queue in Cache 2, visibility estimation results in Cache 3, and the road surface condition classification results in Cache 5. Edge-MuSE uses User Datagram Protocol (UDP) and local

network socket for data transmission to the road users and traffic management agency for information dissemination.

In the process, the research designed the feedback mechanism based on five cache spaces to adjust the distribution of the computation resources automatically. The mechanism can detect the occupancy status of each cache with high frequency. If the cache is overflowed or empty, the mechanism will reallocate the computation resources of the upstream thread to decrease and increase its processing speed correspondingly.

3.3.2 Computation Resources Allocation

To achieve real-time performance on edge devices, the research selected the Nvidia Jetson Xavier NX. NX is equipped with many advanced components to support the computation demands of Edge-MuSE, including a 6-core ARM 64-bit CPU (6 MB L2 and 4 MB L3), two Nvidia Deep Learning Accelerators (NVDLA), and a Volta GPU with 384 NVIDIA CUDA cores and 48 Tensor cores. The assignment of NX computation resources to different threads is shown in Figure 3.5. The five threads are mapped to separate ARM cores (i.e., ARM 0 - 4). The one ARM core (i.e., ARM 5) left free is assigned to handle the OS and background processes running out of the system. The two feature extraction modules in the image de-haze thread and road surface classification thread are assigned to the CUDA cores and Tensor cores of the Volta GPU for matrix convolution operation.

3.4 Experiment and Performance Evaluation

3.4.1 Experiment Configuration

Because the traffic environment varies greatly in different scenarios, the research team built up two testbeds to evaluate Edge-MuSE system comprehensively. In the research, we cooperated with City of Bellevue and Norwegian Public Roads Administration (NPRA) to address the weather-oriented safety challenges. Figure 6.7 shows the sensors prepared for the research and the installation in City of Bellevue and Oslo. The testbeds in the two cities have distinct characteristics. The Bellevue testbed is a local road in the forest on a small hill. As a result, the main challenge in the scenario is the rainy weather and moist climate. The unexpected heavy rain in the winter season results in severe car crashes caused by slippy roads and low visibility. The heavy fog in the morning is also a significant challenge in the scenario. Different from Bellevue, the key challenge in Oslo is the extremely cold winter. The testbed is located on E8 Corridor, one of the main freeways in Europe. The lowest temperature there can reach $-30°C$, resulting into heavy snow and thick ice covering the road surface, which are dangerous for passing vehicles. Therefore, our environment sensing system, Edge-MuSE, is deployed in the two testbeds to address the above weather-oriented traffic challenges. Its performance is highly recognized by the local transportation agencies in both cities. At present, all installed sensors are functioning well and playing a significant role in traffic management.

The sensors set up by the research team consist of three main modules: sensor module for data collection, edge computing module for data processing, and communication module for sensing results dissemination. The Oslo testbed has an add-on heating module to keep all the electronic components warm in low-temperature conditions. The data collection module uses

an advanced IP camera for high-resolution video data collection. The camera is empowered by infrared technology to deal with low light conditions like nighttime or heavy snow. In the edge computing module, we use the Nvidia Jetson Xavier NX device to process the collected video data locally with high efficiency. Finally, the communication module is designed to disseminate the sensing results to transportation agencies and road users for traffic safety improvements.

3.4.2 Data Description

The data we used in the research are from three data sources for different purposes, including the weather data, surveillance data and self-collected video dasta. The **weather data** utilized in this study, which serve as ground-truth for validating the performance of Edge-MuSE, were sourced from weather stations installed by WSDOT. Typically, these stations generate comprehensive weather data that includes weather conditions, air temperature, dew point, relative humidity, atmospheric pressure, wind speed, and more. This data is publicly available, open-source, and can be accessed via the Traveler Information Application Programming Interface (API). For the purposes of this study, we utilized only the basic information from these weather stations, including location (expressed in terms of longitude and latitude), timestamp, weather condition, temperature, humidity, and visibility. It is important to note that this weather data is updated every five minutes. Additionally, the average distance from the testbeds to the nearest weather stations is approximately 1,500 feet, ensuring the relevance and accuracy of the data in the model training and results validation. The **surveillance data** generated by the traffic surveillance system deployed by WSDOT is collected for model training. The research team selected 40 surveillance cameras covering different kinds of transportation scenarios like freeways, rural areas, local roads, and coastal roads to increase the diversity of the training data. The research team collects

250 images for each camera in different weather conditions. And then, the research labeled the images with four kinds of road surface conditions: dry, wet, snow-covered, and icy. Finally, the data are fed to the classifier for model training and testing. The **self-collected video data** were generated by sensors installed in the testbed, containing real-time raw video data and sensing results. Cooperated with the nearest weather stations, the self-collected data is used to evaluate the performance of Edge-MuSE system in the testbed.

3.4.3 Image De-haze & Visibility Estimation

Leveraging the image de-hazing method proposed in Section 3.B, this research extracts four essential features from the original image data and removes the haze (as illustrated in Figure 3.7). Figure 3.7(a) displays the original hazed image captured by the sensor installed at the Bellevue testbed. To exhibit the proficiency of Edge-MuSE, we present the most challenging scenario for haze removal in Figure 3.7: a condition of heavy fog under low lighting with glaring street lights. The de-hazed image, shown in Figure 3.7(b), demonstrates the impressive performance of Edge-MuSE in image de-hazing. Figure 3.7(c) presents the concatenated scattering map estimated from the four critical feature maps displayed at the bottom: dark channel (Figure 3.7(d)), maximum contrast (Figure 3.7(e)), color attenuation (Figure 3.7(f)), and hue disparity (Figure 3.7(g)).

Then the visibility data collected by weather stations are introduced in the step as the ground-truth data to do the model training and testing. In this case, the two coefficients, scattering coefficient β and attenuation coefficient γ, can be calculated based on Eq.(3.12) and Eq.(3.13). Then the regression model f is applied to estimate the relationship between the coefficient with the minimum MSE (Eq.(3.14)). The research uses 5,000 ground truth visibility records to in-

vestigate the relationship between the two coefficients. Figure 3.8 shows the relationship and the regression model trained by the ground truth data, where MSE is used as the loss function.

The estimated relationship between the coefficients can be represented by Eq. (3.20). The value of r^2 and MSE are 0.154 and 0.008, respectively.

$$\beta = 11.579 * \gamma + 0.356 \tag{3.20}$$

One point worth mentioning is that the γ values of all ground truth records are larger than 0.001 in Figure 3.8. From Eq.(3.13), we can see the value of γ varies in a small range in the low haze (i.e., high visibility) condition. Therefore, in this condition, a precise γ has to be obtained to estimate visibility accurately. However, in practical applications, measuring the change of the coefficient γ in low visibility conditions is more meaningful and cost-effective. As a result, existing visibility meters for commercial usage usually set up boundaries for their measurement range. In the research, the visibility meters installed in weather stations by WSDOT set the lowest boundary of γ as 0.001, where the upper boundary of the visibility value is about 4km (2.5 miles). Therefore, the lower bound of γ can be observed in Figure 3.8 as 0.01.

Table 3.1 shows the performance of Edge-MuSE on visibility estimation in different ranges. In low visibility conditions ($V_s < 500m$), the visibility estimation accuracy can reach 99% in $\pm$20% error range. With the increase of visibility, the sensitivity of visual contrast C_v decreases (i.e., C_v/d decreases with d increasing). As a result, the accuracy drops to about 90% in the clear atmosphere (i.e., $V_s \geq 2000m$). However, the overall accuracy can still reach 93% in all situations.

Table 3.1: Performance of Edge-MuSE on Visibility Estimation in Different Conditions

Threshold	±5%	±10%	±20%
$V_s \geq 2000m$	85.29%	89.14%	93.18%
$1000m \leq V_s \leq 2000m$	88.17%	90.25%	95.42%
$500m \leq V_s \leq 1000m$	90.36%	93.22%	97.03%
$V_s < 500m$	91.23%	95.78%	98.75%
Overall	89.27%	92.15%	96.61%

3.4.4 Road Mask Extraction

The research introduces an innovative road mask extraction method by integrating contouring and optical flow to increase the robustness of Edge-MuSE in various scenarios. Figure 3.9 shows the process of road mask extraction. The left side of Figure 3.9 indicates the process of optical flow extraction on foreground blobs. Besides moving vehicles on the road, some false-positive pixels are also included in the foreground due to the camera vibrations. A regional area threshold is introduced into the process to filter the pixels in the moving object tracking algorithm. Finally, the accumulation of the trajectory can represent the road mask in the camera view. Road contouring is shown in the right side of Figure 3.9.

In the experiment, the research team labeled 100 images from various camera views to validate the road mask extraction results from Edge-MuSE. In this case, the Intersection of Union (IOU) method was introduced to quantify the difference between the detected and labeled (i.e., ground-truth) roadway regions. It computes the portion of the roadway that overlaps between the detected region and the labeled region: $IoU = \frac{AreaofOverlap}{AreaofUnion}$. The results are shown in Table 3.2. The average IOU value from the 100 images was 0.92.

Table 3.2: Road Segmentation Results Validation with IOU measurement

IOU Value	Percentile	Accumulated Percentage
less than 0.9	3%	3%
0.9-0.92	31%	34%
0.92-0.95	45%	79%
0.95-0.97	18%	97%
0.97-1.00	3%	100%

3.4.5 Road Surface Condition Classification

The three feature maps, brightness, dark channel, and gray values, are fed to classifiers for road surface condition classification. To eliminate the bias caused by different image inputs from various cameras, all the images are resized to 300×350. Figure 3.10 shows the distributions of the feature values in different surface conditions. The four columns in Figure 3.10 indicate four different road surface conditions: dry, wet, snow-covered, and icy. Row (b), (d), and (f) show the brightness value, dark channel value, and gray value feature maps, respectively. To clearly visualize the feature distributions, we compress the 2D feature maps to 1D feature distributions, as shown in Row (c), (e), and (g). The x-axis indicates the column index of the feature maps, and the y-axis represents the average intensity value of all the pixels in the column.

Differences among the four road surface conditions can be observed from the 1D feature distributions. In dry and snow-covered conditions, diffuse reflection happens on the road surface. Therefore, the three feature values keep stable with the change of x index. The white snow on the road surface can reflect more light than the gray road surface. Therefore, the values in snow-covered conditions are higher than those in dry conditions. For the wet and icy conditions, the distributions of the three features vary extensively. The street lights on the left side of the road result in the rapid rise of three features when $x \in [0, 50]$. In the rest of the image, little light

can be reflected into the camera due to the specular light reflection. As a result, rapid drops can be observed in three feature distributions. However, because ice on the road surface can be impacted by passing vehicles, the mixed ice, dust, and snow can result in diffuse reflection on the road surface. Therefore, the diffuse reflection features can be observed from the icy conditions. Therefore, the icy condition contains a smaller variance in all three feature maps compared to the wet condition. In summary, the features can capture the light reflection status and are useful for classifying road surfaces of different conditions.

In the experiment, the research team labeled 10,000 images captured by 40 cameras in different weather conditions for model training and testing with the 8 : 2 split ratio. The test road surface condition classification results are shown in Table 3.3. The experiment compares the performance of four classifiers on road surface condition classification. And the results show that the Random Forest model gets the best performance and can reach overall 93.25% accuracy. The dry condition has the highest accuracy, which can reach 97%, and the icy condition has the lowest, but still reach 85%.

3.4.6 System Performance Evaluation

To realize reliable and equal traffic environment sensing for safety improvements, the experiment tests Edge-MuSE's performance in data processing and communication efficiency.

Processing Efficiency Evaluation

This subsection aims to evaluate the performance of Edge adaption proposed in Section 4. To improve the processing speed on limited edge devices, the research presents two methods: 1) parallel computing for data streaming optimization; and 2) feedback mechanism for computa-

Table 3.3: Road Surface Condition Classification Results

Model	Accuracy					Mean
RF		Dry	Wet	Snowy	Icy	95.25%
	Dry	0.97	0.01	0.02	0.00	
	Wet	0.01	0.94	0.01	0.04	
	Snowy	0.03	0.01	0.91	0.05	
	Icy	0.01	0.09	0.05	0.85	
KNN		Dry	Wet	Snowy	Icy	83.65%
	Dry	0.92	0.02	0.05	0.01	
	Wet	0.07	0.81	0.03	0.09	
	Snowy	0.09	0	0.87	0.04	
	Icy	0.04	0.19	0.10	0.67	
SVM		Dry	Wet	Snow	Icy	35.82%
	Dry	0.56	0.18	0.20	0.06	
	Wet	0.21	0.34	0.19	0.26	
	Snow	0.38	0.10	0.37	0.15	
	Icy	0.09	0.36	0.22	0.33	
NB		Dry	Wet	Snow	Icy	66.36%
	Dry	0.86	0.03	0.10	0.01	
	Wet	0.08	0.77	0.03	0.12	
	Snow	0.12	0.02	0.79	0.07	
	Icy	0.04	0.18	0.13	0.65	

tion resources allocation. To evaluate the effects of the two methods, we tesed three different structures on Jetson Xavier NX device. The first structure follows the sequential logic flow, indicating that the input of the module is exactly the output of the last module. The second structure introduces the parallel programming method with pre-defined computing resource mapping. Finally, the third one is the structure used in Edge-MuSE, parallel programming with the feedback mechanism for computing resources allocation.

Table 3.4: Processing Efficiency Evaluation

Structures	Sequential	Parallel	Edge-MuSE
Processing Speed (FPS)	2.3	7.4	21.3
CPU Memory Usage (%)	34%	65%	82%
GPU Memory Usage (%)	14%	51%	78%
Power Consumption (W)	21.6W	27.1W	32.9W
Efficiency (FPS/W)	0.106	0.273	0.647

Results are shown in Table 3.4. Five metrics are adopted, including processing speed, CUP and GPU memory usage, Power Consumption, and Efficiency (FPS/Power). The comparison of the first and second columns shows that introducing parallel computing accelerates the processing speed from 2.3 FPS to 7.4 FPS, and the efficiency increases from 0.083 to 0.246. The differences between the second and third columns indicate that introducing the feedback mechanism increases the processing speed and efficiency to 21.3 and 0.647, respectively. The improvements result from the flexible computing resource allocation: the higher usage of CPU and GPU memory indicates more efficient resource allocation for the system.

Communication Efficiency Evaluation

Communication efficiency can impact the time latency significantly in the system, especially in rural areas with unstable internet connections. Therefore, Edge-MuSE deployed the system

on edge devices to realize real-time traffic environment sensing. To test the performance of communication efficiency, the research designs two communication structures shown in Figure 3.11. We use the same multi-task sensing technology but deploy it in edge devices and back servers respectively in the two systems. The top one is the edge-based communication system used in Edge-MuSE, and the bottom one shows the server-based centralized communication system used in traditional sensing systems. Traditional sensing methods send the raw data to the back server for central processing. Then, the raw data are processed by the back server and the sensing results are disseminated to road users through the internet. In rural areas, unstable internet connections can cause long time latency in large-size raw data transmission and sensing results dissemination. However, in edge-based communication systems, like Edge-MuSE, the data is processed and disseminated in local network. The large bandwidth of local network ensures stable and efficient data transmission. Additionally, the sensing results from edge devices are transmitted to transportation agencies for large-scale traffic management. Compared to raw data, the size of sensing results is much smaller, allowing for efficient transmission to back server and thus quick response.

This research evaluates the efficiency of communication under various internet conditions. To achieve this, we set the internet bandwidth at three levels: 2 Mbps, 10 Mbps, and 50 Mbps. We then tested the time latency in both systems, with the results presented in Table 3.5. When using 50 Mbps, we found that real-time raw data can be smoothly delivered from the edge to the processing center, resulting in minimal time latency in both communication systems. However, when we reduced the bandwidth to 10 Mbps, we observed that the slower upload speed did not support the demands for real-time raw data transmission. This caused an increase in time latency in the traditional system. Conversely, the edge-based communication system was

Table 3.5: Communication Efficiency Evaluation

Measurements	Bandwidth	2 Mbps	10 Mbps	50 Mbps
Avg Latency for Users (s)	Centralized Sys	311.06	23.34	1.15
	Edge-based Sys	0.24	0.27	0.21
	Change	-99.92%	-98.84%	-81.74%
Avg Latency for Agencies (s)	Centralized Sys	287.92	20.34	2.18
	Edge-based Sys	5.24	2.26	2.07
	Change	-98.18%	-88.89%	-5.05%
Avg Bytes Utilization (MB/s)	Centralized Sys	0.25	1.36	4.37
	Edge-based Sys	0.25	0.59	0.54
	Change	0.00%	-56.62%	-87.64%
Avg Occupancy	Centralized Sys	96%	95%	69%
	Edge-based Sys	94%	47%	9%
	Change	-1.86%	-50.53%	-86.94%

unaffected by this decrease in bandwidth. This is because the local processing and dissemination were not dependent on the upload speed, and a 10 Mbps bandwidth was sufficient for real-time transmission of sensing results. When we further reduced the bandwidth to 2 Mbps, a standard level in rural areas, a significant latency was observed in the traditional sensing systems. In this case, the time latency increased to five minutes during our experiment, making it unsuitable for real-time applications. However, in the edge-based sensing system, the smaller size of the sensing results made it possible for these results to be transmitted to transportation agencies for traffic management in a timely manner.

3.5 Chapter Summary and Future Works

This research presents the Edge-based Multi-task Safety-oriented Environmental (Edge-MuSE) sensing system as an innovative solution to weather-oriented traffic safety challenges. The system offers comprehensive traffic environment perception through multi-task sensing, enabling

a more efficient improvement of traffic safety compared to traditional single-task sensing methods. Additionally, by reallocating computational loads from central servers to edge devices, Edge-MuSE ensures quicker responses, lower latency, and enhances privacy by transmitting only processed sensing results rather than raw data. The system requires only video data as input, making it versatile and cost-effective for broad deployment. The research's main contributions lie in the creation of this system, the integration of four innovative sensing methods using computer vision and AI technologies, the system's optimization to fit edge computing architecture, and its successful deployment in real-world testbeds. The resulting improvements in traffic safety underscore Edge-MuSE's potential as an effective, wide-scale solution for environmental-related traffic safety issues.

The Edge-MuSE system, while pioneering, presents two significant limitations. First, it is currently designed with fixed surveillance systems in mind, rather than adaptable on-board systems. This design choice has led to the development of embedded algorithms that benefit from stable environmental perception and fixed camera views, such as image de-hazing and road segmentation. However, these are not easily translatable to moving camera systems, posing a challenge for the Edge-MuSE system's versatility. Second, the system's performance is subject to variations in lighting conditions. Particularly in rural areas that often lack artificial lighting at night, the effectiveness of Edge-MuSE may be significantly reduced. This identifies a need for the system to better adapt to a range of lighting scenarios, possibly by incorporating sensor fusion technologies integrating infrared and visual cameras.

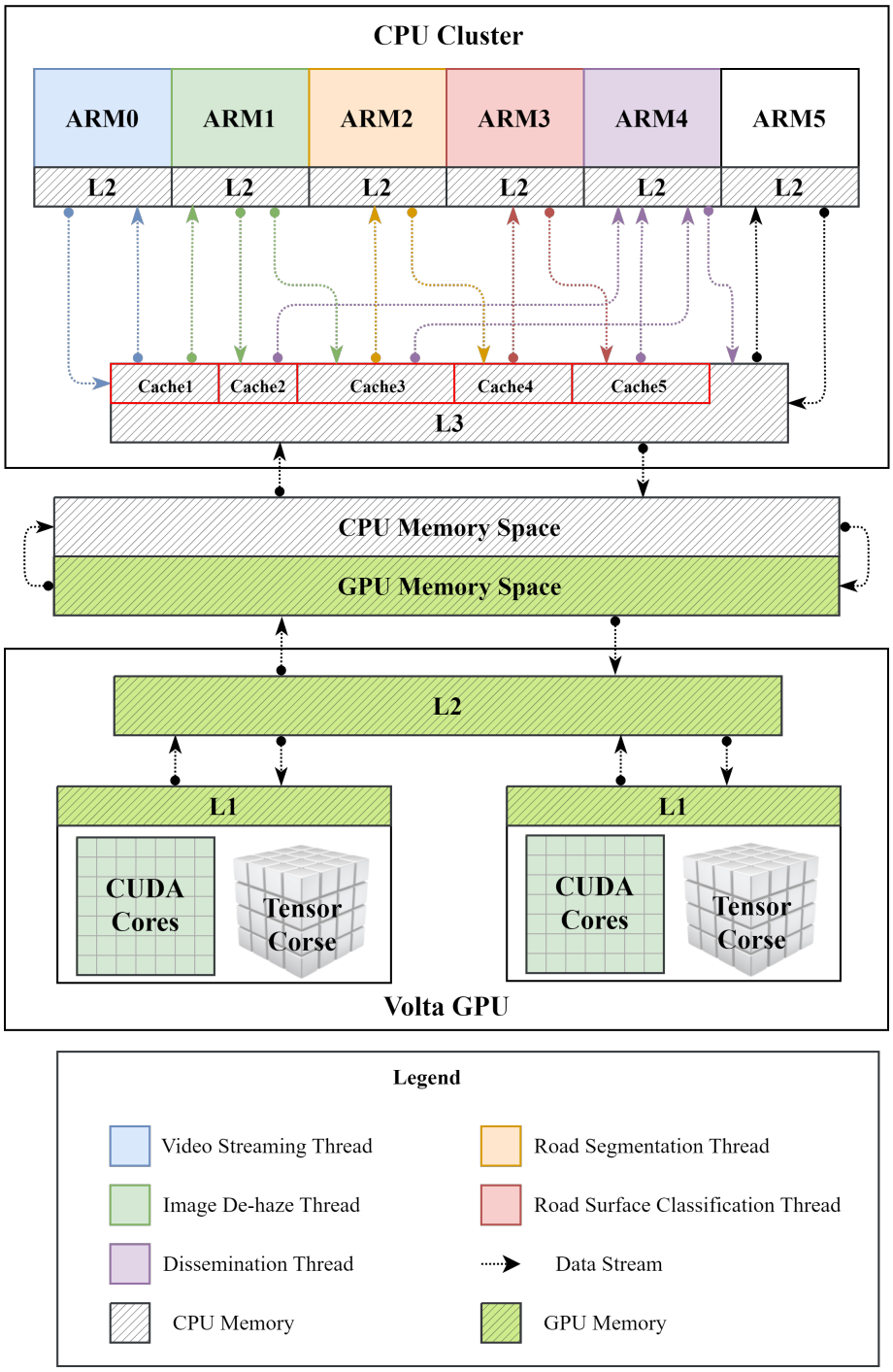

Figure 3.5: Mapping of Threads to Edge Resources

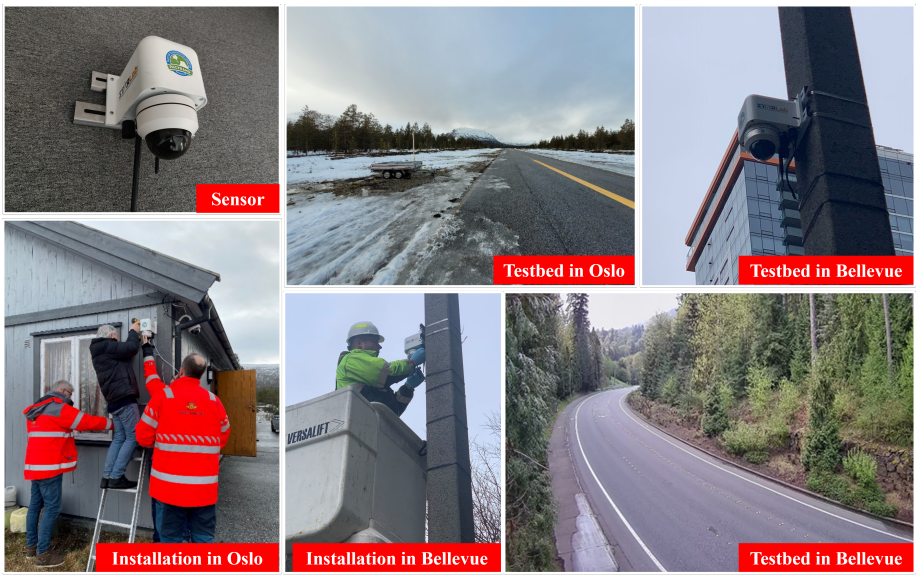

Figure 3.6: Testbeds Setup in City of Bellevue and Oslo

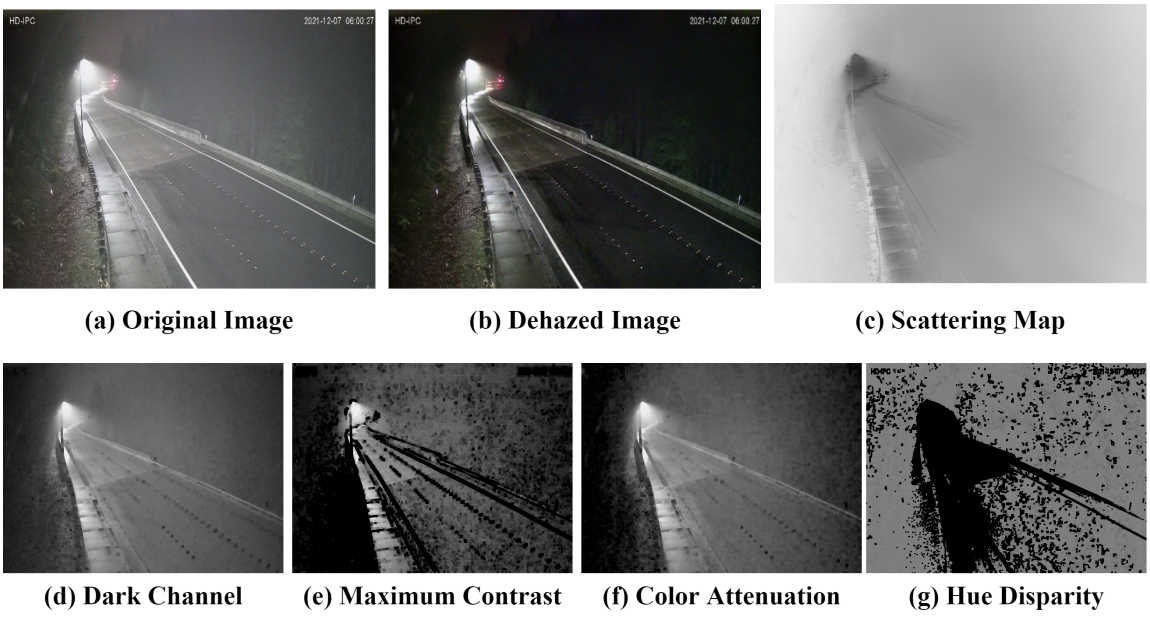

Figure 3.7: Critical Feature Extraction and Image Dehaze

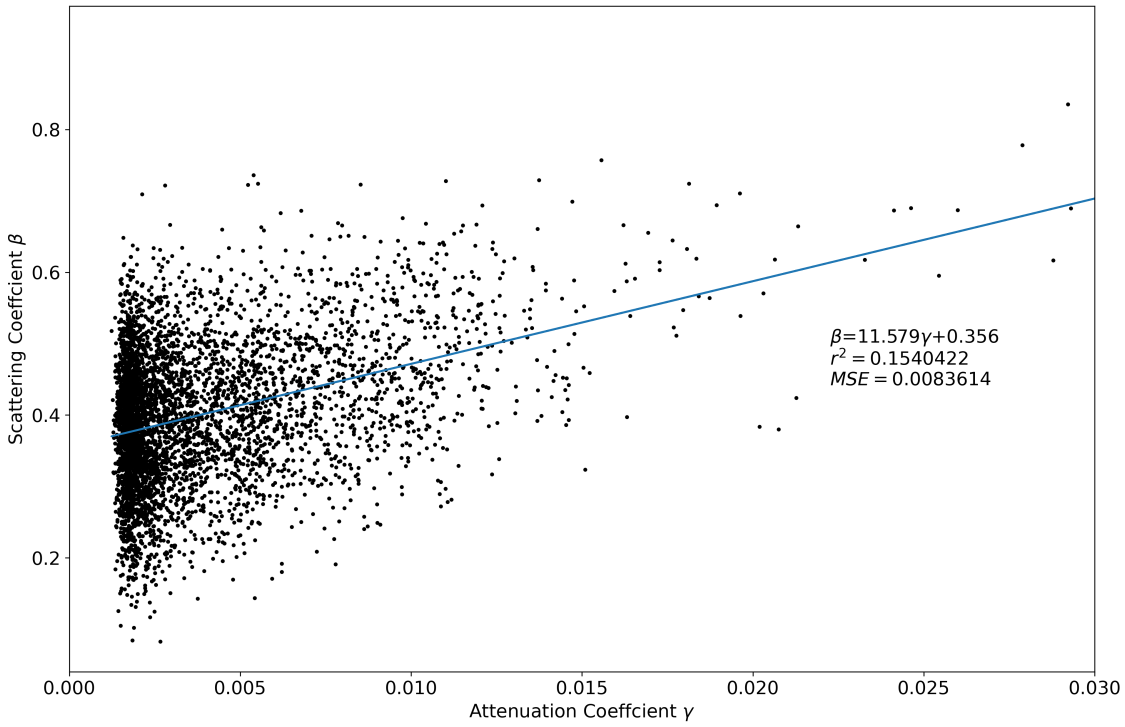

Figure 3.8: Relationship between Scattering Coefficient β and Attenuation Coefficient γ

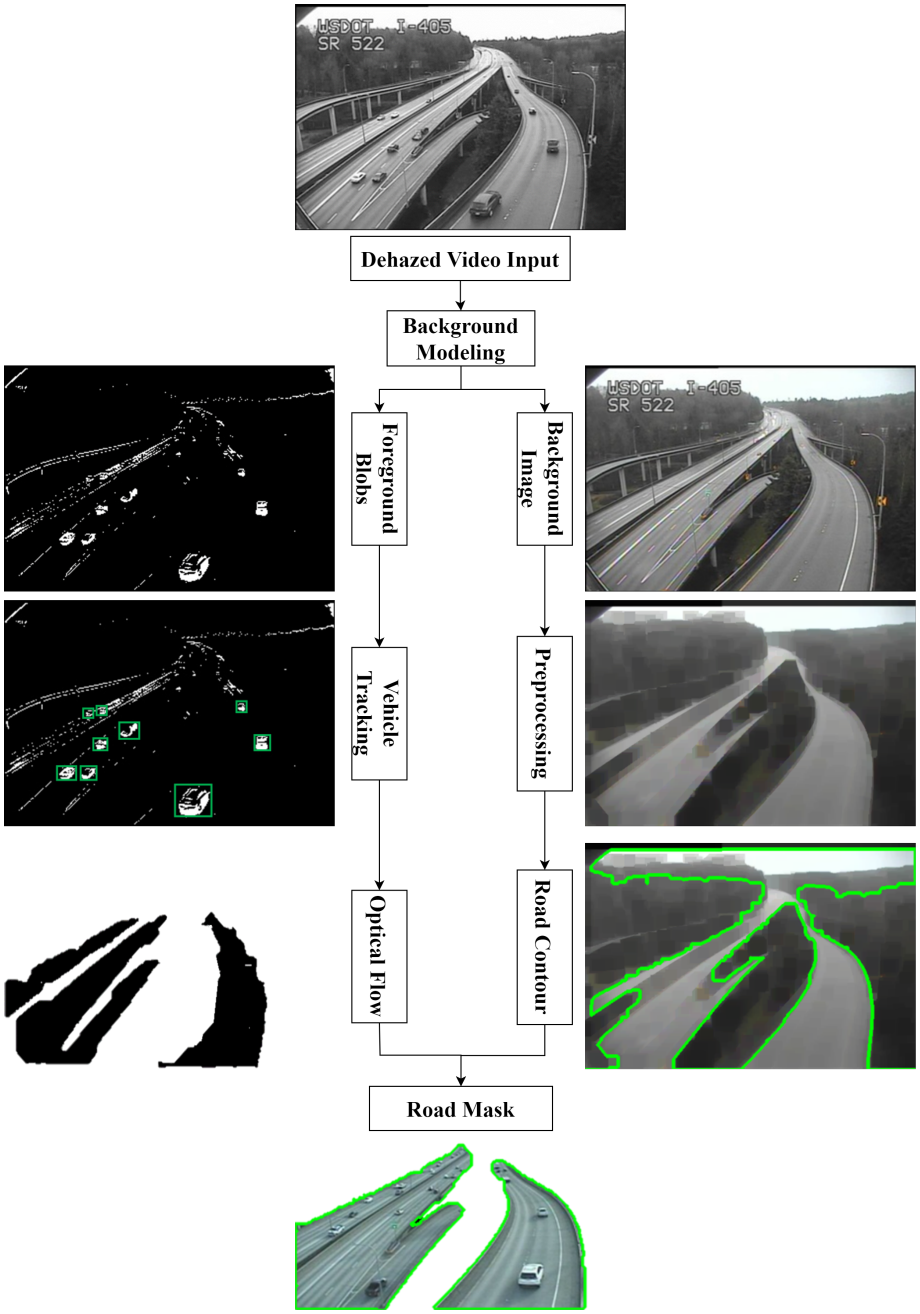

Figure 3.9: Sample Road Mask Extraction Result Demonstration

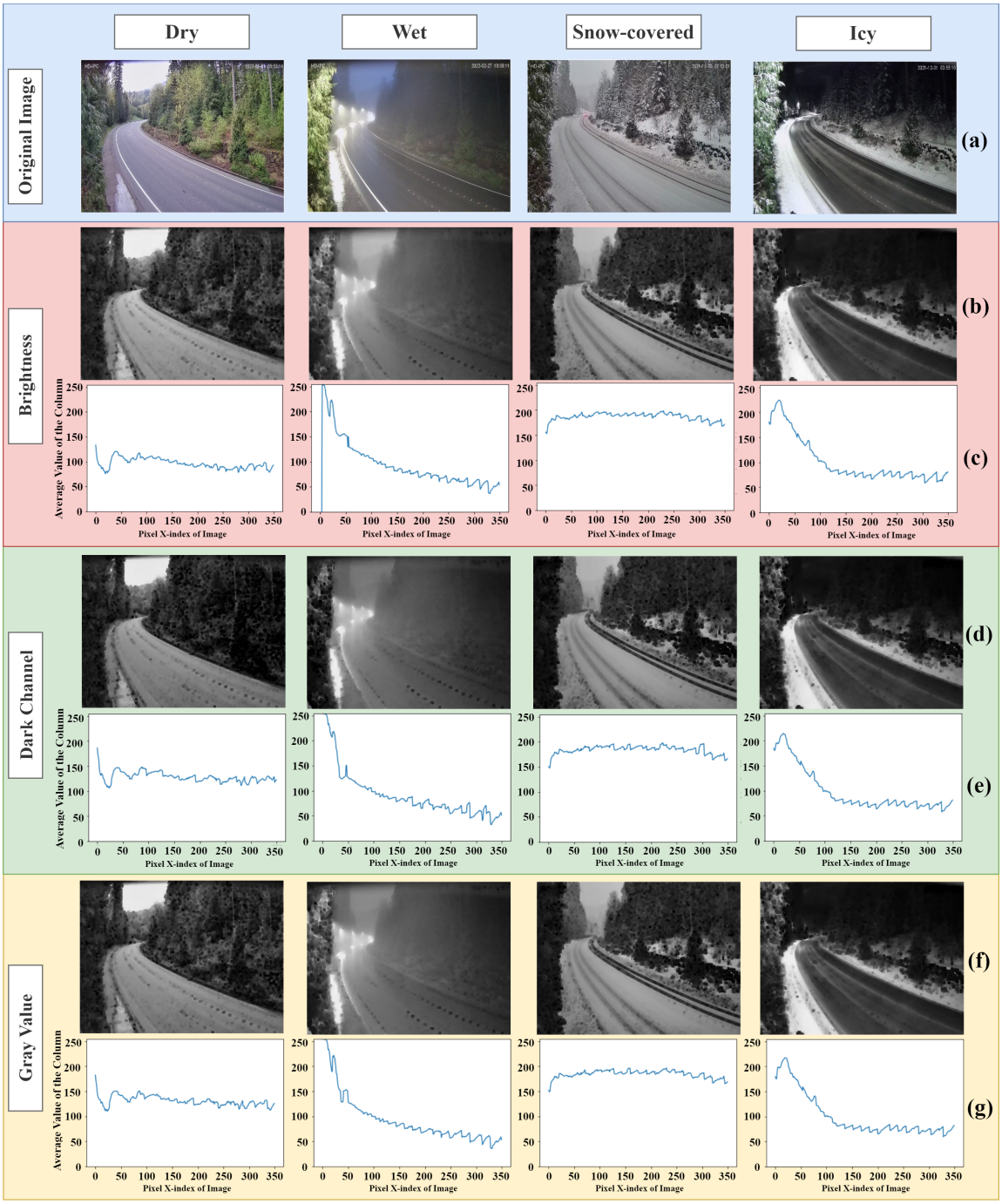

Figure 3.10: Feature Map in Four Road Surface Conditions

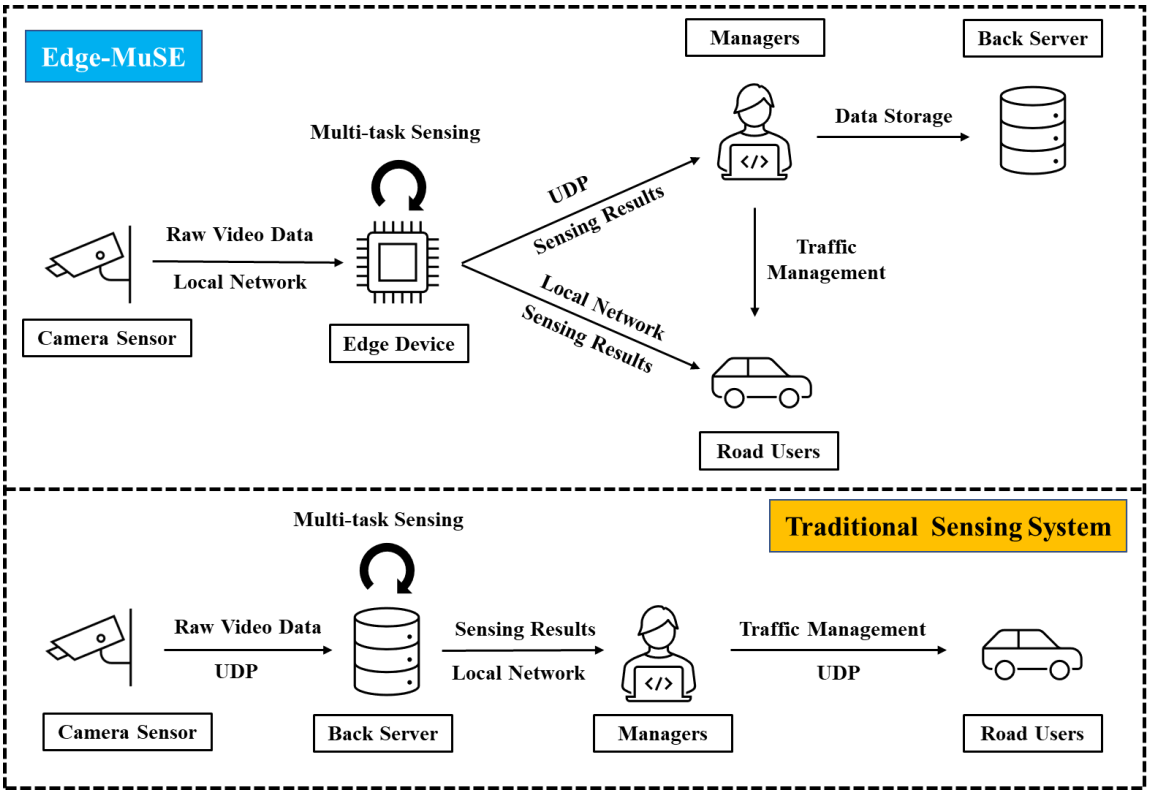

Figure 3.11: Structure of Edge-based and Server-based Communication Systems

Part II

Situation-aware Machine Intelligence for Active Transportation Users

4

Chapter 4. Scale-Aware Representation Learning Empowered Sensing (SARLES) System for Pedestrian Crowds Perception in Complex Transportation Scenarios

This chapter is modified from the published work:

- C. Liu, H. Yang, R. Ke, S. Yin, and Y. Wang*". Scale-Aware Representation Learning Empowered Sensing (SARLES) System for Pedestrian Crowds Perception in Complex Transportation Scenarios", accepted by Transportation Research Part C in Oct 2023.

- C. Liu, H. Yang, K. Ma, X. Jiang, S. Yin, and Y. Wang*. "Scale-Aware Representation Learning Empowered Sensing (SARLES) System for Pedestrian Crowds Perception in Complex Transportation Scenarios.", Proceedings of the 103rd Annual Meeting of Transportation Research Board, Washington D.C. USA, Jan. 2024.

4.1 Challenges and Motivations

Pedestrians are important yet vulnerable users in modern transportation systems. According to the latest National Household Transportation Survey (NHTS) [199], walking accounts for approximately 11% of all surveyed trips and is the second most commonly used mode. However, this popularity brings with it a high risk for pedestrian safety in urban/crowded/busy settings. For example, the World Health Organization (WHO) reports that pedestrians and cyclists, collectively known as Vulnerable Road Users (VRUs), account for 50% of global fatal road traffic injuries, and data from the Fatality Analysis Reporting System (FARS) show that pedestrian fatalities rates have been steadily increasing over the last two decades from 11% (2002) to 17% (2020) in the United States. In 2020 alone, the death toll reached 6,516, meaning one pedestrian is killed in a traffic accident every 81 minutes on average. To address these safety issues, the USDOT published the National Roadway Safety Strategy (NRSS) to outline strategies and encourage technological innovation to reduce serious roadway injuries and fatalities. Developing new sensors and AI-based technologies can play a critical role in addressing these challenges.

Numerous studies have been conducted to enhance road management [48], traffic control [9], and traffic sensing [158] through AI-based technologies. Instead of always installing newer and newer sensors [74], utilizing existing surveillance systems is more cost-effective for cities and agencies. This drives the advancement of smart infrastructure and leads to significant accom-

plishments in the perception of vehicles [200], roads [146, 201], traffic [48], and VRUs [9] based on surveillance cameras. As a result, there has been a growing emphasis on pedestrian sensing technologies in the ITS community, driven by the need for traffic safety and equity. Pedestrian sensing technologies in ITS can be broadly categorized into three types: 1) detection-based methods, 2) regression-based methods, and 3) density estimation methods.

- **Detection-based methods** are essential for sensing objects based on feature extraction and classification and have played a significant role in Intelligent Transportation Systems (ITS). Recent advancements have led to the development of powerful object detectors such as R-CNN [202], YOLO [82], and SSD [119] that have achieved remarkable performances in various scenarios, including traffic control [151], forecasting [148], and management [48]. However, directly applying these detection-based sensing methods to multi-modal transportation scenarios for pedestrian sensing and perception is challenging due to various factors, such as occlusion, cluttered backgrounds, tiny objects, and blurs. Compared to other transportation agents like vehicles, Vulnerable Road Users (VRUs) are small objects with few pixels and limited features, making them difficult to differentiate from the background. Pedestrians often appear in groups, leading to higher occlusion rates than other transportation agents. The complicated transportation backgrounds further reduce the detection accuracy. In summary, it is still a challenging task to implement detection-based methods for perceiving tiny and congested pedestrian groups in complex transportation scenarios.

- **Regression-based methods** aim to capture the global features of the image, such as texture and gradient features, directly for pedestrian sensing. The learned global features are then used for pedestrian perception and sense through various regression techniques,

(a) Highly Congested & Occlusion

(b) Tiny Objects & Scale Change

(c) Complex Background & Blur Regions

(d) Perspective Changs & Diverse Distribution

Figure 4.1: Representative challenges for pedestrian sensing in transportation scenarios include (a) highly congested and occlusions, (b) tiny objects and scale changes, (c) complex background and blur regions, and (d) perspective changes and diverse density distribution.

such as linear regression and Gaussian mixture regression [203]. Regression solutions may be better equipped than detection-based methods to address the challenges of occlusion, tiny objects, and blurs through the ability of global feature extraction. However, changes in perspective and scalability of 2D views can cause a decrease in performance, leading to overestimation in low-density areas and underestimation in high-density areas. The diverse distribution of pedestrian groups in transportation scenarios presents a significant challenge to regression-based methods. Additionally, without considering local features, the performance of regression-based methods is significantly affected by backgrounds, resulting in many false positives in multi-modal transportation scenarios.

- **Density estimation methods** have recently gained popularity in the Computer Vision (CV) community for pedestrian sensing in highly dense scenarios [204, 205]. Unlike detection-based and regression-based models, density estimation methods utilize the powerful representative feature extraction ability of CNN to estimate the crowd numbers through the density map directly. These methods have been applied to many application scenarios, including public safety, public space design, and intelligent crowd monitoring. However, several challenges must be addressed before these methods can be adopted in transportation settings. Firstly, improving VRU safety requires more than just crowd size and counting numbers; the status and locations of these agents are also important, which may be difficult to obtain from density estimation methods. Secondly, VRUs often gather in various density groups in transportation scenarios, which results in unevenly distributed crowds and unsteady scale changes in the camera view. It is difficult for most existing density estimation methods to handle the diverse density as well as the discontinuous scale changes for accurate crowd sensing. Finally, the cluttered background in

transportation scenarios, such as road marks, uneven lighting, passing vehicles, and infrastructures, can significantly impact the accuracy of the density estimation results.

In a vast landscape of pedestrian sensing methods, striking disparities in approaches have been noted across various research sectors. Traditional computer vision tends to narrow its focus, developing models that are tailor-made for challenges within pre-established contexts. Detection-based approaches excel when object features are distinct and discernible by the camera, while density estimation techniques are tailored for densely populated settings where individual object features might be obfuscated due to distance or occlusion. However, the realm of transportation demands a more holistic and adaptive solution, one that can adeptly navigate the multifaceted challenges of real-world scenarios. As depicted in Figure 4.1, transportation scenarios present myriad obstacles for precise pedestrian sensing, such as occlusions, intricate backgrounds, fluctuating scales, varied distributions, perspective shifts, minuscule objects, and blurred zones. Conventional methods, restricted by their predefined confines, fall short of addressing the dynamic requisites intrinsic to genuine transportation contexts.

To address these challenges, we propose a novel ensemble sensing system: Scale-Aware Representation Learning Empowered Sensing (SARLES). SARLES extracts multi-scale representations for enhanced pedestrian sensing accuracy, and thus is designed for heterogeneous transportation scenarios. The system is divided into three modules:

The initial phase employs a robust fourth-order encoder-decoder structure, rendering a preliminary density map of the entire image, facilitating the capture of global contextual cues and accommodating the scale disparities inherent to pedestrian figures. Recognizing the potential inadequacies of this map due to the inherent distortions of 2D imaging, the subsequent Density Map Segmentation and Clustering (DMSC) module is introduced. This innovative seg-

ment clusters the preliminary map into congruent density feature patches, employing iterative clustering to ensure consistency in scale and density. Culminating the system is the ensemble scale-adaptive Local Patch Refinement (LPR) module. Leveraging an ensemble FCN network with diverse kernel dimensions, this module refines each patch based on its unique density and scale characteristics. Consequently, SARLES possesses the dexterity to discern both macro and micro features of pedestrian assemblies, refining its output to produce superior quality density maps tailored for precise pedestrian detection and enumeration.

SARLES is versatile to detect pedestrians robustly in different scenarios, especially those challenging cases identified previously: obstructions, busy backgrounds, and uneven distributions. The contributions of this study is three-fold:

- **System Design**: SARLES's primary contribution is scale adaptability for detecting diverse pedestrian group distributions. This robust design ensures SARLES is equipped to tackle the multifaceted challenges pervasive in transportation environments, such as occlusions, bustling backgrounds, minuscule entities, irregular distributions, and the intricacies of perspective shifts.

- **Modules and Methodology**: The main framework consists of three modules, each encapsulating distinct methodological advancements.

  - The Encoder-Decoder module is adept at harnessing multi-scale contextual information. It encodes the spatial interdependencies within the entire image. It also innovatively introduces a residual module tailored for feature extraction. This enhancement ensures optimal feature capture across varied encoding resolutions.

  - The DMSC module, through its iterative clustering approach, segments the preliminary density map into congruent patches based on density characteristics. This strategic segmentation counters the distortions due to perspective variations in 2D representations, yielding patches with consistent scale and density metrics, priming them for refined processing in following modules.
  - With the LPR module, SARLES presents an ensemble FCN network, specifically curated for nuanced local feature extraction. This innovation produces an impeccably precise density map integral for pedestrian sensing. The eclectic mix of kernel dimensions ensures that the module adaptively refines the local patches proposed by the DMSC module, granting SARLES the finesse to discern both the overarching and intricate features of pedestrian conglomerations amidst the complexities of transportation backdrops.

- **Experiment Results**: SARLES outperforms five baseline models across three benchmark datasets. Its ability to detect heterogeneous pedestrian group distributions proves itself a state-of-the-art (SOTA) model.

4.2 Architecture

The architecture of the SARLES system is illustrated in Figure 6.1. The system comprises three innovative modules: an encoder-decoder module, a Density Map Segmentation and Clustering (DMSC) module, and a Local Patch Refinement (LPR) module. Firstly, the fourth-order encoder-decoder module takes the whole image as input and performs multi-scale contextual feature extraction for an accurate initial density map generation. This initial density map is suitable

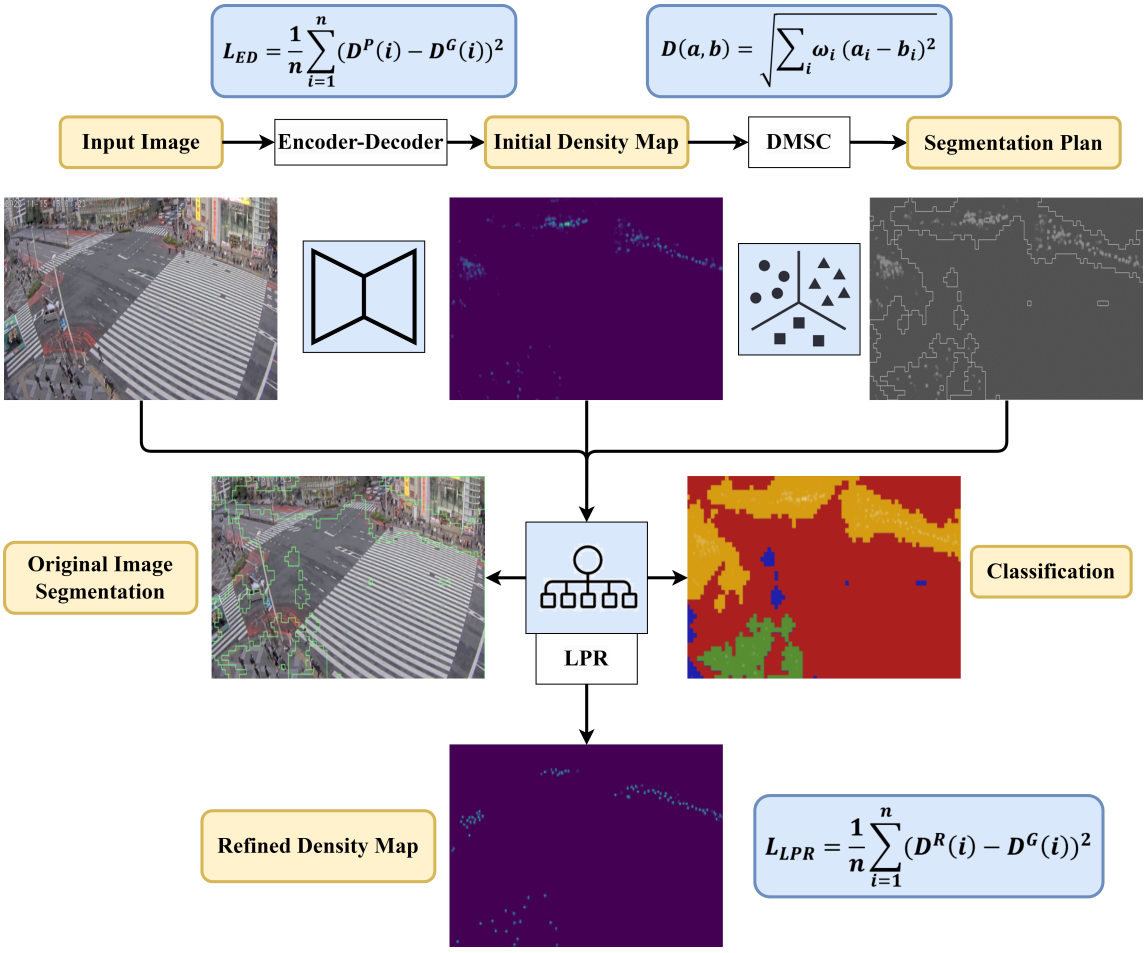

Figure 4.2: The architecture of SARLES system.

[The architecture of SARLES system. The input image is fed to the Encoder-decoder module for multi-scale feature extraction and initial density map generation. Then the Density Map Segmentation and Clustering (DMSC) module segments the initial density map into several patches based on density features. Finally, the patches are classified and finetuned in the Local Patch Refinement (LPR) module for an accurate density map generation. The key metrics used in the three modules are also listed accordingly.]

for basic sensing tasks such as crowd counting, while it may lack the details needed for higher-level sensing tasks, particularly in transportation scenarios like pedestrian localization. Therefore, to fit the demands, DMSC module is proposed in the paper to cluster and segment the initial density map into multiple small local patches based on the density features. The LPR module then uses a CNN model with two fully-connected (FC) layers to classify the local patches into five categories ranging from high density to low density. The segmentation plan from DMSC and classification labels are then fed to an ensemble network for patch density map refinement. Finally, by integrating all the local patches, the system can generate a precise density map for accurate pedestrian detection, sensing, and localization. Further details about the SARLES can be found in the following subsections.

4.2.1 Encoder-decoder Module for Initial Density Map

The fourth-order encoder-decoder structure used in the paper can capture four levels of features across the input whole image (shown in Figure 4.3). The extracted multi-scale contextual features from the image can be used to generate an initial density map. The proposed encoder-decoder module is a symmetric network consisting of two parts: the encoder and the decoder module. The encoder module aims to reduce the feature matrix dimension from high resolution to low resolution, and each encoder layer consists of a residual layer for feature extraction and a max pooling layer for encoding and dimension reduction. In the decoder part, the network introduces the nearest neighbor interpolation for up-sampling and integrates the features across scales through the residual layer. The decoder part allows the network to understand both critical features and spatial information for a more accurate density map generation. The encoder-decoder module is an end-to-end architecture. Therefore, the loss function used in the training

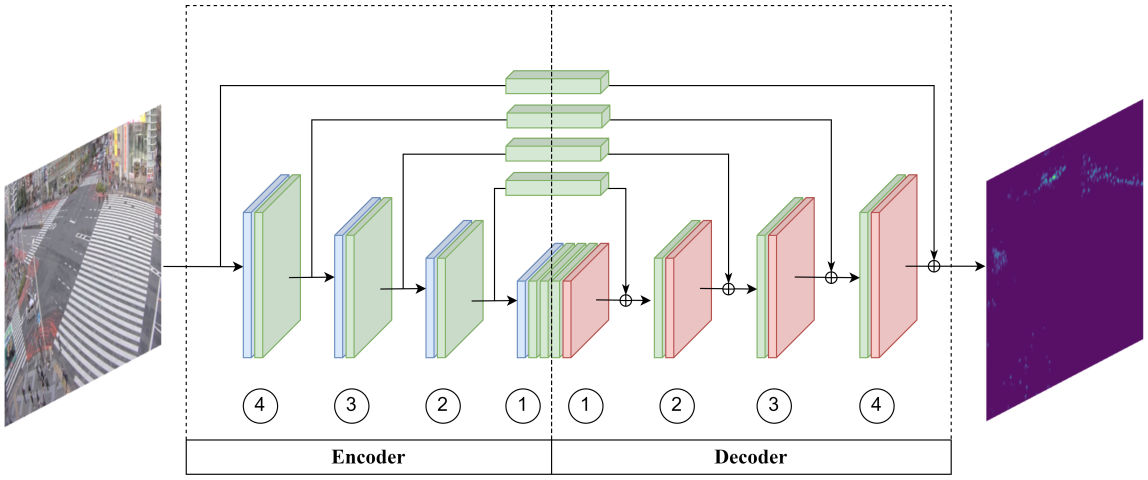

Figure 4.3: The structure of the symmetric fourth-order Encoder-decoder module for multi-scale feature extraction and initial density map generation.

[The structure of the symmetric fourth-order Encoder-decoder module for multi-scale feature extraction and initial density map generation. The green blocks indicate the residual module (see Figure 4.4) used for feature extraction. And the blue and red blocks represent the down-sampling and up-sampling operations, respectively. Please note that before each encoding operation, the network branches off and applies a residual module at the original pre-pooled resolution, and this can bring significant multi-scale information to the decoder part for spatial feature extraction.]

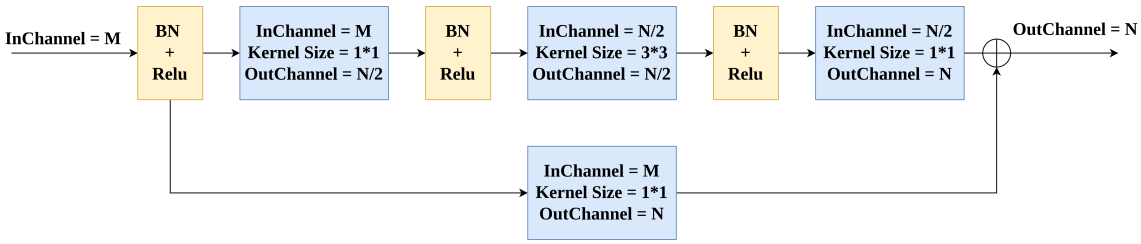

Figure 4.4: The structure of residual module designed for multi-scale features extraction.

process is standard Mean Squared Error (MSE) (shown in Eq 4.1) which is aimed at minimizing the differences between the estimated and ground-truth density map.

$$L_{ED} = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} (D^P(i,j) - D^G(i,j))^2 \tag{4.1}$$

Where (i,j) indicates the pixel coordinate. D^P and D^G represent the predicted density map and ground truth density map, respectively. The ground truth density map D^G is generated by applying a Gaussian blur kernel to each annotated head point. The generation of D^G is shown in Eq 4.2, where δ represents the discrete delta function; p represents the location of the annotated head point in the image coordinate; H indicates the number of annotation head points in the image; G represents the Gaussian kernel; and σ_k indicates the variance of the Gaussian kernel applied to annotation point k. In ShanghaiTech dataset [1], σ_k is defined as $\sigma_k = 0.3\bar{d}_k$, where $\bar{d}_k$ is the average distance of point k to its r nearest neighbors, which can be used to customize the kernel size of Gaussian filter to fit the points in various density regions. This paper follows the rule and uses the customized Gaussian filter to generate the ground truth density map for model training and validation.

$$D^G(p) = \sum_{k=1}^{H} \delta(p - p_k) * G(\sigma_k) \tag{4.2}$$

It is worth noting that the innovative residual module (shown in Figure 4.4) play a pivot role in the ED module. The structure begins with an input having M channels, which first undergoes a batch normalization (BN) followed by a Rectified Linear Unit (ReLU) activation function. This is succeeded by a 1×1 convolution operation that reduces the channel size to $N/2$. After another round of BN and ReLU, a 3×3 convolution is applied, maintaining the same number of channels ($N/2$). A subsequent BN + ReLU and 1×1 convolution expand the channels back to N. In parallel, there is a skip connection that performs a direct 1×1 convolution on the initial input, converting its channels from M to N. The outputs from both the main sequence and the skip connection are then element-wise added together, resulting in an output with N channels. This module plays a pivotal role in encoder-decoder architectures as it allows the network to retain critical information from previous layers, mitigating the vanishing gradient problem. The residual connection ensures that, if certain layers don't contribute to the final performance, their weights can be set in such a way that they'll effectively perform identity mapping, making the network easier to optimize.

4.2.2 Density Map Segmentation and Clustering (DMSC) Module

The DMSC module is meticulously crafted to tackle the challenges of discontinuous scale shifts and varied density distributions in transportation contexts, ensuring precise pedestrian sensing, detection, and localization. While traditional crowd counting techniques offer numerous density estimation methods for multi-scale sensing, they predominantly lean on CNN, FCN, or regression models. These methods capture spatial details under the presumption of continuous shifts in scales, densities, and perspectives within images. Contrarily, transportation settings often see pedestrians clustering in distinct small groups within specific zones, leading to irregu-

lar scale transitions and a myriad of density distributions. Addressing this, the DMSC module adeptly segments the preliminary density map into distinct patches, each characterized by its unique density feature. This ensures a uniform density within each patch while preserving the diversity across them. These differentiated patches are then channeled into the Local Patch Refinement (LPR) module, facilitating the creation of an enhanced, precision-tuned density map optimized for accurate pedestrian sensing.

The DMSC module integrates the renowned DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [206]. A staple in computer vision and pattern recognition, DBSCAN stands out as an unsupervised clustering technique, distinguished by its adaptability and efficiency. Its edge lies in its ability to operate without pre-defined information regarding cluster count or configuration. Within the scope of this research, DMSC harnesses DBSCAN to cluster pixels of proximate values while distinguishing them from significantly varying ones. Parameters like minimum pixel count and neighborhood radius allow DBSCAN to discern and segment local clusters of diverse densities within a density map. Nonetheless, a single DBSCAN layer can be influenced by noise present in the initial map, often leading to an excessive creation of clusters or superpixels. To remedy this, DMSC incorporates an iterative strategy (refer to Algorithm 1) for refined density map segmentation. The process commences by partitioning the initial map into 3600 small standard superpixels, with each superpixel comprising 576 pixels. Subsequent DBSCAN layers then group these superpixels based on density similarities, yielding larger, feature-rich superpixels for advanced processing. The distance between two such superpixels, a and b, is computed using Eq 4.3.

$$d(a,b) = \sqrt{\sum_{i=1}^{n} \omega_i \cdot (a_i - b_i)^2} \tag{4.3}$$

Algorithm 1 Initial Density Map Segmentation & Clustering (DMSC)

Input: Image I, pixel p, neighbourhood threshold ε, minimum set size M, maximum number of sets N, labeled set L

Output: Superpixel label sets $L(p)$

Set initial pixel label is unvisited for each pixel $p \in I$
while TRUE **do**
  while $\forall p \in I$ is visited **do**
    randomly find an unvisited pixel p_s
    set p_s as visited
    find all pixels $C_s = \{p_1, p_2, ..., p_m\}$ in the neighbourhood ε of seed pixel p_s
    if $m > M$ **then**
      Create a new label $l \in L$
      for every $p_i \in C_s$ **do**
        if p_i is unvisited **then**
          set p_i is visited and add it to label l
          if there are more than m pixels in the neighbourhood ε of p_i **then**
            add the pixels into C_s
          end if
        end if
      end for
    else
      mark p_s as noise
    end if
  end while
  if $length(L) \leq N$ **then**
    break
  else
    Set the average value of $l \in L$ as the new pixel $p' = avg(l)$ in the next iteration
    Empty label set L and set all new pixels p' as unvisited
  end if
end while

Where a_i and b_i indicates the i^{th} attribute of pixel a and b, respectively. The ω_i indicates the weight of i^{th} attribute in the distance measurement. In the paper, we use three attributes ($n = 3$) including 2D coordinate of the pixel in initial density map and the pixel gray value. In each layer of DBSCAN, the superpixels generated by the previous layer are regarded as new inputs whose value is the mean value of all the pixels in the superpixel. Finally the iteration will stop until the number of clusters are reduced no more than ten and the number of pixels which are taken accounted as noises are no more than 10%. This process results in the clustering of small patches with similar density features and the formation of larger patches for further processing. This iterative DBSCAN procedure provides a flexible and efficient approach to segmenting the density map into meaningful patches without the need for prior knowledge of the number or shape of the clusters.

It is worth mentioning that the DMSC module stands out for its strategic design, aimed at overcoming the challenges of random initialization and ensuring consistently stable clustering outcomes. Central to its functionality is the incorporation of DBSCAN, a density-based spatial clustering algorithm that doesn't rely on initial seed points, but rather grows clusters founded on point density. Enhancing its efficacy, DMSC utilizes an iterative process, refining clustering results by iterating over the density map multiple times. Before clustering, the initial density map undergoes segmentation into standardized superpixels, offering a more uniform basis for DBSCAN. Additionally, the DMSC approach excels in noise handling; with the inherent properties of DBSCAN identifying and isolating noise from primary clusters, the module maintains that noise remains below 10% of the dataset. Finally, the DMSC's use of weighted distance measurement ensures a balanced clustering, uninfluenced by individual attributes. Cumulatively, the DMSC module showcases a harmonious blend of techniques, from DBSCAN's capabili-

ties to iterative refinement and noise management, making it a reliable solution for challenges in density map segmentation and clustering.

4.2.3 Local Patch Refinement (LPR) Module

The local Patch Refinement (LPR) Module aims to build an ensemble network to handle the patches with various density distributions and generate a more detailed and accurate density map for precise pedestrian sensing. The LPR module is comprised of two parts: 1) a CNN model for patches classification based on density features; 2) an ensemble network for density map refinement. The classification result for each patch serves as one input to the ensemble network, which determines key parameters in model concatenation. The structure of LPR module is shown in Figure 4.5.

In the classification part, a CNN model with a 1×1 convolutional layer and the following two FC layers are designed to classify the patches from high to low density. The CNN model is trained to categorize patches into one of five predefined density classes: very high density, high density, median density, low density, and very low density. In the training process, a certain number of regions of interest (ROIs) are generated randomly from the input initial density map, and their sizes must cover more than 1/16 of the whole image to ensure effective feature learning. Based on the features extracted by the convolutional layer, the FC layers assign density labels for the ROIs. The ground truth density labels used in the training process are defined as the ratio of people number to the region of the image. The loss function is the standard cross-entropy loss shown in Eq 4.4, where t_i is the ground truth label and p_i is the Softmax probability for i^{th} class.

$$L_{Classification} = -\sum^{i} t_i * log(p_i) \tag{4.4}$$

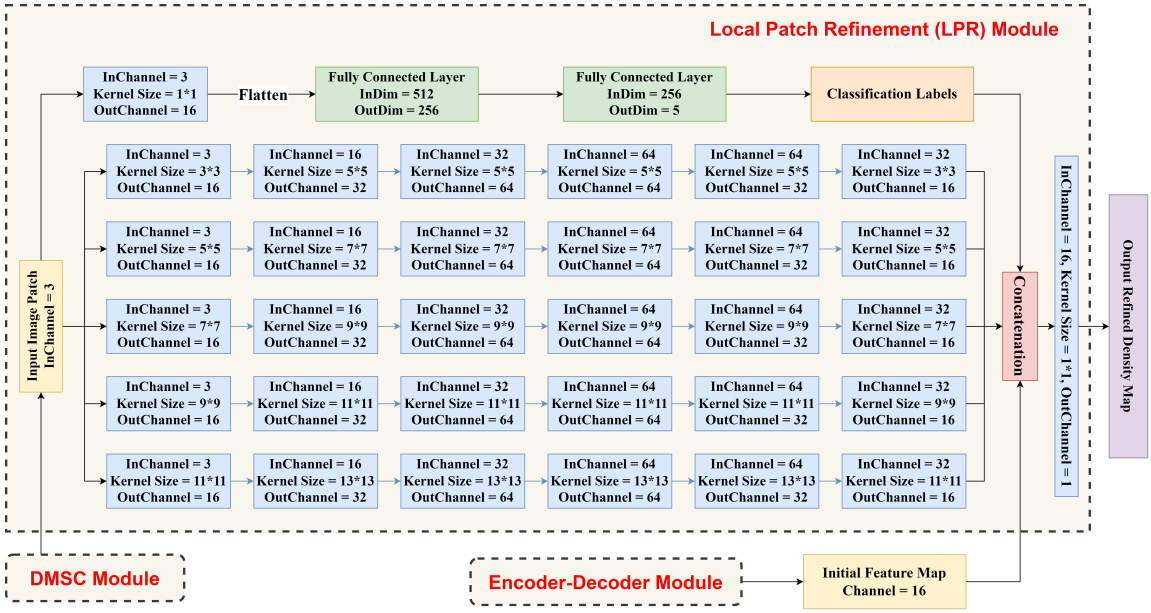

Figure 4.5: The structure of Local Patch Refinement (LPR) module.

[The structure of Local Patch Refinement (LPR) module. The input image patches are input to the classification model and ensemble network for density map refinement. The three dashed boxes indicate the three modules proposed in the SARLES system. The blue boxes and green boxes in the figure indicate the CNN layers and FC layers, respectively. The yellow and purple boxes indicate the inputs and outputs of LPR module.]

The second part of LPR module is the ensemble network, which aims to generate a more detailed and accurate density map by handling patches with various densities. The network consists of five Fully Convolution Networks (FCNs) with the same structure but different kernel sizes to extract features at different scales. The structure of the ensemble network is shown in Figure 4.5. Every FCN has six layers of convolutional layers for feature extraction. The features extracted from the five FCN models as well as the encoder-decoder module are concatenated based on the weights determined by the classified density labels. The density classification labels play a significant role in the concatenations process by determining the weights of different levels of features for generating the final density map. For high-density patches, the high-level features extracted from large kernel sizes FCN models have a higher weight in generating final density maps. Conversely, low-level features have a higher weight in processing low-density patches.

The ensemble network is meticulously trained for comprehensive pedestrian detection, striving to accurately localize each pedestrian within the scene rather than merely counting them. While many traditional density estimation methods prioritize crowd counting, endeavoring to minimize discrepancies between the actual and estimated number of individuals, our method deviates. Instead, it harnesses the standard MSE loss function (shown in Eq 4.5) to minimize the difference between the refined density map output $D^R(i,j)$ and the ground-truth density map $D^G(i,j)$ (see Eq 4.2) for the input training image I_i.

$$L_{LPR} = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} (D^R(i,j) - D^G(i,j))^2 \tag{4.5}$$

4.3 Experiment & Results Evaluation

4.3.1 Experiment Design & Implementation

The system is trained in an end-to-end way, and the overall loss function is defined as the minimum MSE between the estimated density map and the ground-truth map (shown in Eq 4.6), where L_{LPR} and L_{ED} are defined in Eq 4.1 and Eq 4.5, respectively.

$$Loss = L_{LPR} + L_{ED} \tag{4.6}$$

In the training process, The learning rates are set as 10^{-4} initially and multiplied by 0.98 every 1K iterations. The batch size is fixed as 4. All images are resized to 1080×1920, and the labels are generated under the same size. Finally, the Adam algorithm [207] is introduced in the paper to optimize the SARLES system and obtain the best results on NVIDIA GTX 2080Ti GPU using Pytorch framework.

4.3.2 Evaluation Metrics

The conventional evaluation metrics for crowd counting, such as MAE and MSE, are not suitable for evaluating SARLES, which aim to perform pedestrian sensing tasks, including counting and localization. Therefore, four metrics are introduced in the paper to evaluate the system performance.

For counting performance evaluation, the paper uses MAE and MSE metrics which are defined in Eq 4.7 and Eq 4.8. Where N is the number of images in the test batch. C_i and C_i^G are the estimated counting and ground truth counting of pedestrians, respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^G| \tag{4.7}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - C_i^G)^2} \tag{4.8}$$

In addition, we introduce two metrics, Peak Signal-to-Noise Ratio (PSNR) metric and Structural Similarity in Image (SSIM) [208], to evaluate the quality of density maps, which is significant for pedestrian sensing. The two metrics are well-known in image similarity calculation metrics. The paper follows the rules given by [114] to calculate the two metrics for performance evaluation.

For monochrome images like density maps, PSNR metric is shown in Eq 4.9, where MAX_I represents the maximum possible pixel value in the image. D and D^G indicate the estimated density map and the ground truth density map generated from Eq 4.2. (i,j) represents the coordinate of the pixel, and N indicates the test batch size.

$$PSNR = \frac{1}{N}\sum^{N} 10log(\frac{MAX_I^2}{\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(D(i,j) - D^G(i,j))^2}) \tag{4.9}$$

The SSIM metric is represented by Eq 4.10. Where μ indicates the mean value of the density map, and *sigma* indicates the variance or co-variance. D and D^G indicate the estimated density map and the ground truth density map generated from Eq 4.2. N indicates the test batch size. C_1 and C_2 indicate the two constant.

$$SSIM = \frac{1}{N}\sum^{N} \frac{(2\mu_D\mu_{D^G} + C_1)(2\sigma_{D,D^G} + C_2)}{(\mu_D^2 + \mu_{D^G}^2 + C_1)(\sigma_D^2 + \sigma_{D^G}^2 + C_2)} \tag{4.10}$$

Table 4.1: Detailed information about the used datasets

Datasets	Number of Images	Average Resolution	Count Statistics		
			Total	Avg	Max
ShTech-A	482	589×868	241,677	501.4	3,139
ShTech-B	716	768×1024	88,488	123.6	578
UCF-QNRF	1535	2013×2902	1251642	815	12865
CityStreet	500	2704×1520	-	~110	-
Self-collect Dataset	500	1080×1920	42,138	834.7	1836

4.3.3 Data Description

In the experiment, four datasets, ShanghaiTech [1], UCF-QNRF[2], and CityStreet [3], are used in the paper to evaluate the performance of SARLES system in different application scenarios. The properties of the datasets used in the research are shown in Table 4.1.

To supplement the previous three datasets for model training and testing, the research team collected a self-made dataset using live camera streaming from several busy intersections and walking streets in Tokyo. This dataset covers various challenges mentioned earlier, including complicated backgrounds, diverse density distribution, scale, and perspective changes. The live camera dataset includes 500 annotated images, consisting of 400 training images and 100 test images. It contains different situations in transportation scenarios, including high-density crowds gathering in four waiting zones with different scales for red lights, large volumes of pedestrians going across the street during peak hours, and a few pedestrians hanging on the street in low-light conditions at midnight. The dataset provides a diverse range of transportation pedestrian crowds scenarios for model training and testing.

4.3.4 Encoder-decoder Performance Evaluation

The fourth-order encoder-decoder module used in the paper can extract four levels of features for accurate initial density map generation. The output initial density map plays a crucial role in the final pedestrian sensing. Therefore, the section aims to evaluate the performance of the encoder-decoder module by visualizing the effect of each encoder-decoder layer when generating the initial density map. It is worth noting that the feature matrices extracted from different levels of encoder-decoder layers have different resolutions. To visualize and evaluate the features extracted from various levels of encoder-decoder layers, the research team implements customized up-sampling methods to restore the feature matrices to their original resolution. Figure 4.6 shows the sample results generated from the four levels of encoder-decoder layers, from the lowest to highest level encoder-decoder layer. The density map generated by Decoder 4 is the final output of the Encoder-decoder module, the initial density map, which has the same resolution as the original image. From Figure 4.6, we can observe that the global features are easily extracted from the encoded low-level layers (first and second level layer). With the resolution recovery process in decoders, the density maps gradually incorporate more details about local features for a more accurate density map. Therefore, we can observe that more details are included in the density maps generated from high-level layers (third and four layers).

To evaluate the performance of each encoder-decoder layer, the research team utilized two evaluation metrics, PSNR and SSIM, to compare the density maps generated from each layer with the ground-truth density map. The comparison results are presented in Table 4.2. The increase in PSNR and SSIM from Decoder 1 to Decoder 4 indicates continuous improvements in the density map generation during the encoding-decoding process. Additionally, it can be observed that the rate of increase for both metrics decreases as the number of layers in the encoder-

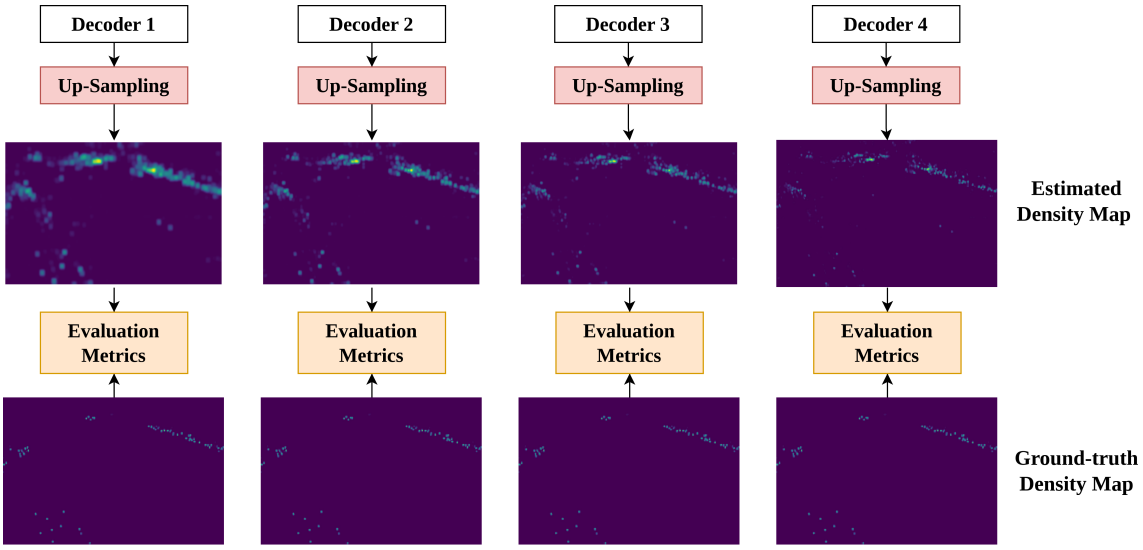

Figure 4.6: Ablation study for Encoder-decoder Module performance evaluation.

[Ablation study for Encoder-decoder Module performance evaluation. The density maps on the top are the sample results generated from each Encoder-decoder layer, from the lowest (Decoder 1) to the highest (Decoder 4). The density maps on the bottom are the ground-truth density map generated by Eq 4.2. The comparison between the estimated density map and the ground-truth density can be used to evaluate the effects of the encoder-decoder module layer by layer.]

Table 4.2: Encoder-decoder module performance evaluation results by comparing the density maps generated from each decoder layer with PSNR and SSIM metrics on ShanghaiTech Part A, Part B, and self-collected datasets. The last column shows the improvements of each layer compared to the last layer.

Layers	Metrics	Shanghai-Tech A	Shanghai-Tech B	Self-collected Data	Avg Improvements
Decoder 1	PSNR	12.74	14.23	18.36	–
	SSIM	0.32	0.41	0.46	–
Decoder 2	PSNR	18.04	19.95	24.14	+30.62%
	SSIM	0.55	0.60	0.68	+46.23%
Decoder 3	PSNR	21.95	23.56	27.39	+12.81%
	SSIM	0.68	0.76	0.81	+18.22%
Decoder 4	PSNR	23.31	25.15	29.63	+6.01%
	SSIM	0.73	0.82	0.86	+6.95%

decoder module increases, indicating that deeper encoder-decoder modules with more layers may have little impact on performance improvement.

Additionally, the ablation study is implemented to Encoder-decoder to measure the impact of the module on the performance of SARLES system. The paper replaces the initial density map generated by the original Encoder-decoder module with the intermediate density map generated by Decoder 1, 2, and 3. The output of Decoder 4 is the output of Encoder-decoder module. We did the tests on all three datasets based on the two metrics, PSNR and SSIM. The comparison results are shown in Table 4.4. It is observed that the quality of the initial density map is significant for the final output refined density map. Compared to the results shown in Table 4.2, high-quality initial density map can be improved better in the following DMSC and LPR modules (PSNR: 20.11% and SSIM:12.79% improvements for the initial density map generated from Decoder 4). However, if the initial density map is in low quality, the improvements are very limited (PSNR: 9.49% and SSIM:7.21% improvements for the initial density map generated from

Table 4.3: Ablation study for Encoder-decoder module. The four rows show the results for different decoder outputs with the DMSC and LPR modules. The output of Decoder 4 is the initial density map we used in SARLES system.

Models	Datasets	Shanghai-Tech A	Shanghai-Tech B	Self-collected Dataset
Decoder 1 + DMSC + LPR	PSNR	13.43	15.33	20.71
	SSIM	0.34	0.45	0.50
Decoder 2 + DMSC + LPR	PSNR	19.94	21.52	23.95
	SSIM	0.60	0.65	0.73
Decoder 3 + DMSC + LPR	PSNR	23.78	25.91	32.23
	SSIM	0.74	0.82	0.89
Decoder 4 + DMSC + LPR (SARLES System)	PSNR	24.07	29.45	35.59
	SSIM	0.81	0.94	0.97

Decoder 1).

4.3.5 DMSC Performance Evaluation

This section aims to visualize and evaluate the effects of DMSC module in SARLES system. The initial map segmentation is significant for the refinement process. Figure 4.7 shows some sample results for initial density map segmentation. In column (a), the input initial density map is displayed. All the initial density maps are monochrome images and have been resized to a scale of 1080×1920 for further processing. In column (b), the initial $60 \times 60 = 3600$ superpixels are displayed. These superpixels are defined using the k-means clustering method. The value and location of each initial superpixel in the image is the mean value of all the included pixels. This step reduces the calculation load of the module, and the reduced resolution is sufficient for accurate diverse density patches clustering. In column (c), the intermediate results of the third iteration are shown. Most of the superpixels are well-clustered into patches, but some super-

Table 4.4: Ablation study for Encoder-decoder module. The four rows show the results for different decoder outputs with the DMSC and LPR modules. The output of Decoder 4 is the initial density map we used in SARLES system.

Models	Datasets	Sh-A	Sh_B	Self-collect
ED with residual module	PSNR	23.31	25.15	29.63
	SSIM	0.73	0.82	0.86
ED without residual module	PSNR	19.52	20.95	23.19
	SSIM	0.53	0.61	0.65
Changes	PSNR	-16.26%	-16.70%	-17.01%
	SSIM	-27.40%	-25.61%	-21.73%

pixels on the edges are still not because their values and locations are between two groups. The clustering methods with small search radius are not able to group them and have to regard them as noises. The targets of the following iterations are to cluster these superpixels into surrounding patches. Finally, in column (d), the output density map segmentation is presented. All the superpixels are well-clustered into proper patches, and the patches are well-separated based on the density features.

To quantitatively evaluate the effect of the DMSC module on the SARLES system, an ablation study was conducted, comparing the proposed system with and without the DMSC module using four evaluation metrics: MAE, MSE, PSNR, and SSIM. The results are presented in Table 4.5. Without the DMSC module, the initial density map is directly fed into the LPR module without undergoing the clustering process. This means that the LPR module is not able to fully utilize the prior knowledge about the density classification. As a result, the performance of the SARLES system drops significantly after the DMSC module is removed. It is worth noting that the performance drop is smaller on the ShanghaiTech A dataset, which has many scenarios with consistent density distributions. In such cases, there is little difference between the initial

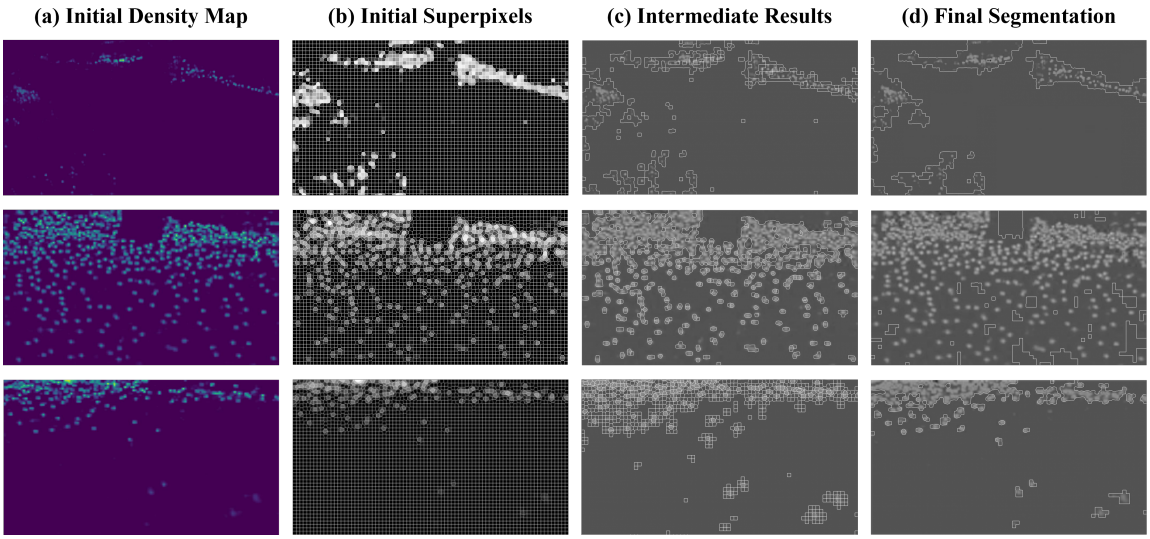

Figure 4.7: Some sample results for initial density map segmentation.

density map and the clustered density map. However, in scenarios with diverse density distributions, such as those in the ShanghaiTech B and self-collected datasets, the DMSC module proves to be essential in accurately clustering the image into multiple patches based on their density features, resulting in a more consistent density distribution within each patch and different density distributions among different patches.

4.3.6 Model Complexity

The complexity analysis of the SARLES system, as outlined in the provided table, presents a concise overview of the system's architectural details. The SARLES model consists of approximately 38 million parameters, indicating a sophisticated yet efficient design capable of handling complex tasks. The model's total size is measured at 147.55 megabytes, which reflects its substantial capacity for processing and analysis without being excessively large for practical applications. Furthermore, the SARLES system requires 5.8 billion floating-point operations per sec-

Table 4.5: Ablation study for DMSC module. Compared to SARLES system, removal of DMSC can result in the increase of MAE and MSE, as well as the decrease of PSNR and SSIM.

Models	Datasets	Shanghai-Tech A	Shanghai-Tech B	Self-collected Datasets
ED+LPR	MAE	78.91	10.23	11.34
	MSE	143.89	15.83	22.27
	PSNR	21.85	23.52	25.14
	SSIM	0.76	0.83	0.84
ED+DMSC+LPR (SARLES)	MAE	69.08	8.33	9.07
	MSE	116.12	12.12	16.42
	PSNR	24.07	29.45	35.59
	SSIM	0.81	0.94	0.97
Changes	MAE	+14.23%	+22.81%	+25.03%
	MSE	+23.91%	+30.61%	+35.63%
	PSNR	-9.22%	-20.14%	-29.36%
	SSIM	-6.17%	-11.70%	-13.40%

Table 4.6: SARLES system Complexity Analysis

Model Name	**# Parameters**	**Model Size**	**FLOPs**
SARLES	38M	147.55MB	5.8G

ond (FLOPs) to function, showcasing its computational intensity and the intricate calculations it performs to achieve its objectives. This analysis underscores the balance SARLES maintains between complexity and efficiency, making it a powerful tool in its domain.

4.3.7 Results Evaluation & Analysis

The initial density as well as the segmentation results are input to the Local Patch Refinement (LPR) module for final refined density map generation. To evaluate the performance of the entire SARLES system, the paper compares the proposed SARLES system with five recently published state-of-the-art methods, CSRNet [204], FCN [209], PCCNet [210], DM-Count [211], and DISSINet [212] on ShanghaiTech, UCF-QNRF, CityStreet and self-collected datasets. Table 4.7 shows the detailed comparison results.

ShanghaiTech Part A and UCF-QNRF focus on high-density situations, making them inherently challenging. Methods such as DM-Count and DISSINet, which are explicitly designed for pedestrian counting in these situations, demonstrate superior performance. These specialized methods cater to the unique challenges posed by high-density scenarios. Although SARLES is not the best-performing model, its design principles are optimized for transportation-oriented pedestrian sensing.

ShanghaiTech Part B, consisting of situations with lower pedestrian density to Sh-A and UCF-QNRF, is notable for its variation in density. Such diversity necessitates a model maintaining a reasonable balance between accuracy and the fidelity of density maps to the ground truth. This is SARLES's tru strength. Achieveing the highest scores in PSNR and SSIM, SARLES proves that its generated density maps are very close to the ground truth. This aptitude in handling diverse densities and maintaining high fidelity distinguishes SARLES from other traditional models.

Table 4.7: The Detailed Comparison of Proposed SARLES System and the SOTA Methods on ShanghaiTech [1] Part A, Part B, UCF-QNRF [2], CityStreet [3] and Self-collected Datasets

Methods		CSRNet	FCN	PCC Net	DM-Count	DISSINet	SARLES
Sh-A	MAE	68.17	126.51	73.52	**59.7**	60.63	69.08
	RMSE	115.04	173.47	124.03	**95.7**	96.04	116.12
	PSNR	23.79	22.18	22.78	**24.74**	24.18	24.07
	SSIM	0.76	0.65	0.74	**0.85**	0.84	0.81
Sh-B	MAE	10.56	23.81	11.03	7.4	**6.85**	8.33
	RMSE	15.95	33.10	18.96	11.8	**10.34**	12.12
	PSNR	27.02	21.36	23.80	27.94	28.76	**29.45**
	SSIM	0.86	0.79	0.90	0.91	0.93	**0.94**
UCF-QNRF	MAE	266.14	153.18	148.71	**85.6**	99.1	108.64
	RMSE	397.53	256.5	247.30	**148.3**	159.2	181.54
	PSNR	12.40	17.22	17.64	23.17	22.84	**23.31**
	SSIM	0.24	0.37	0.41	**0.79**	0.72	0.76
City Street	MAE	14.93	24.86	21.71	12.24	11.62	**9.37**
	RMSE	22.87	57.19	43.64	17.27	15.41	**12.05**
	PSNR	26.46	21.85	22.31	26.23	26.54	**38.15**
	SSIM	0.84	0.72	0.76	0.82	0.85	**0.95**
Self-collected Dataset	MAE	13.27	41.91	18.55	9.88	10.05	**9.07**
	RMSE	22.30	63.74	31.28	**13.52**	14.87	16.42
	PSNR	24.58	17.12	19.97	26.92	27.06	**35.59**
	SSIM	0.76	0.46	0.67	0.82	0.89	**0.97**

CityStreet and Self-collected Datasets capture urban scenarios where pedestrians interact within complex transportation environments. To evaluate SARLES's performance in these environments, our research team amalgamated images from two camera perspectives within the CityStreet dataset with our self-collected datasets. This approach aimed to comprehensively evaluate the performance of the proposed SARLES system. SARLES demonstrates its efficacy by outperforming other methods across all metrics in these urban contexts. The inherent challenges presented by these datasets, including diverse pedestrian density and intricate transportation scenes, adversely affect competing methods. Their failure to accurately differentiate between true pedestrians and false positives, which emerge from the complex background, leads to decreased performance. This highlights the importance of SARLES's design principles, specifically tailored to adeptly navigate the challenges of urban environments.

In summary, with its specialized design catering to transportation-oriented pedestrian sensing, the SARLES system proves especially effective in datasets representing urban contexts. While it may not be the top performer in high-density scenarios, its ability to handle diverse densities and maintain high structural and visual fidelity makes it more robust. The shortcomings of other methods in urban settings further emphasize the robustness and applicability of SARLES in real-world transportation scenarios.

The visualized comparison among MCNN, CSRNet, SCAR, and SARLES systems on ShanghaiTech part A, part B, and self-collected datasets are shown in Figure 4.8, Figure 4.9, and Figure 4.10.

Figure 4.8 shows the sample results from comparison on ShanghaiTech Part A. Most of the scenarios in the datasets are highly congested, but the distribution of density is continuous, and the scale changes in the image are steady. Therefore, CSRNet, SCAR, and SARLES (ours) can

all handle this situation well and perform similarly well on the dataset.

Figure 4.9 shows the sample results from comparison on the ShanghaiTech Part B. The dataset is known for having outdoor scenes with low or medium density and diverse crowd distributions. From the presented images, it is observed that SARLES outperforms all the other methods in handling high-density, low-density, diverse-density, and complicated backgrounds. To highlight the strengths of SARLES, we have marked four red boxes on the image. Boxes 1, 2, and 3 show local low-density areas with uneven distribution, and SARLES can accurately capture these features in the generated density map. Box 4 demonstrates the impact of complex backgrounds on these methods. SARLES can correctly identify a false negative, while other methods exhibit false positives. The results illustrate the excellent performance of SARLES in dealing with diverse density crowds and complex backgrounds.

Figure 4.10 shows sample results from comparison on the self-collected dataset. The three columns of figures depict a busy intersection with pedestrian groups concentrated in the four waiting areas or going across the street, resulting in dynamic density distribution. This scene also contains complicated backgrounds, including road signs, traffic facilities, passing vehicles, and clothing models in street shops, which can negatively impact crowd detection and lead to many false positives. Three red boxes are drawn on the figures to highlight the challenges. Box 1 shows traffic facilities on the roadside, box 2 shows a street shop model, and box 3 shows the passing vehicles as well as road markings. It is observed that SARLES outperforms all other methods on this dataset by handling these challenges. These results demonstrate the excellent performance of SARLES for pedestrian sensing in transportation scenarios.

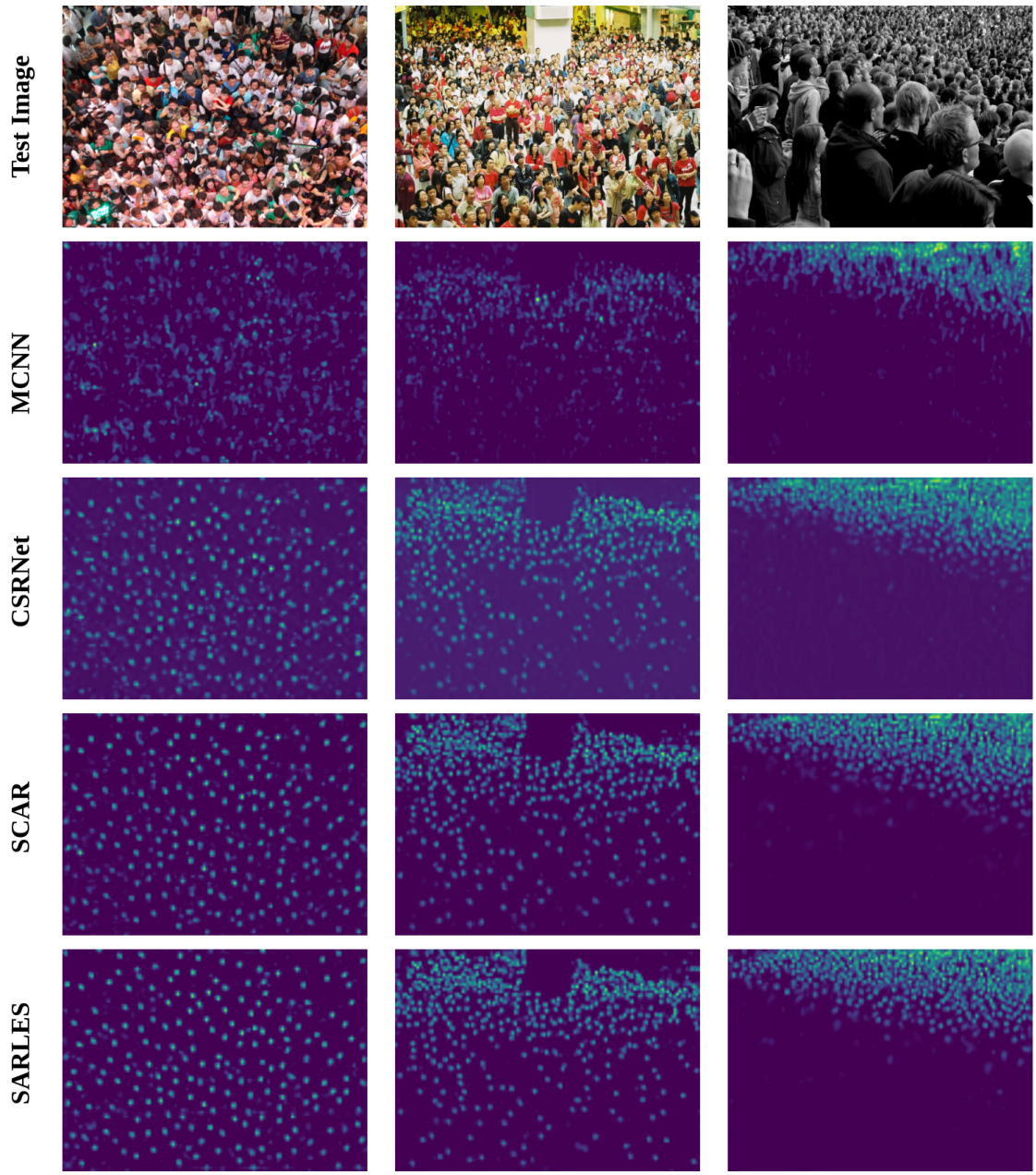

Figure 4.8: Sample results from comparison on ShanghaiTech Part A

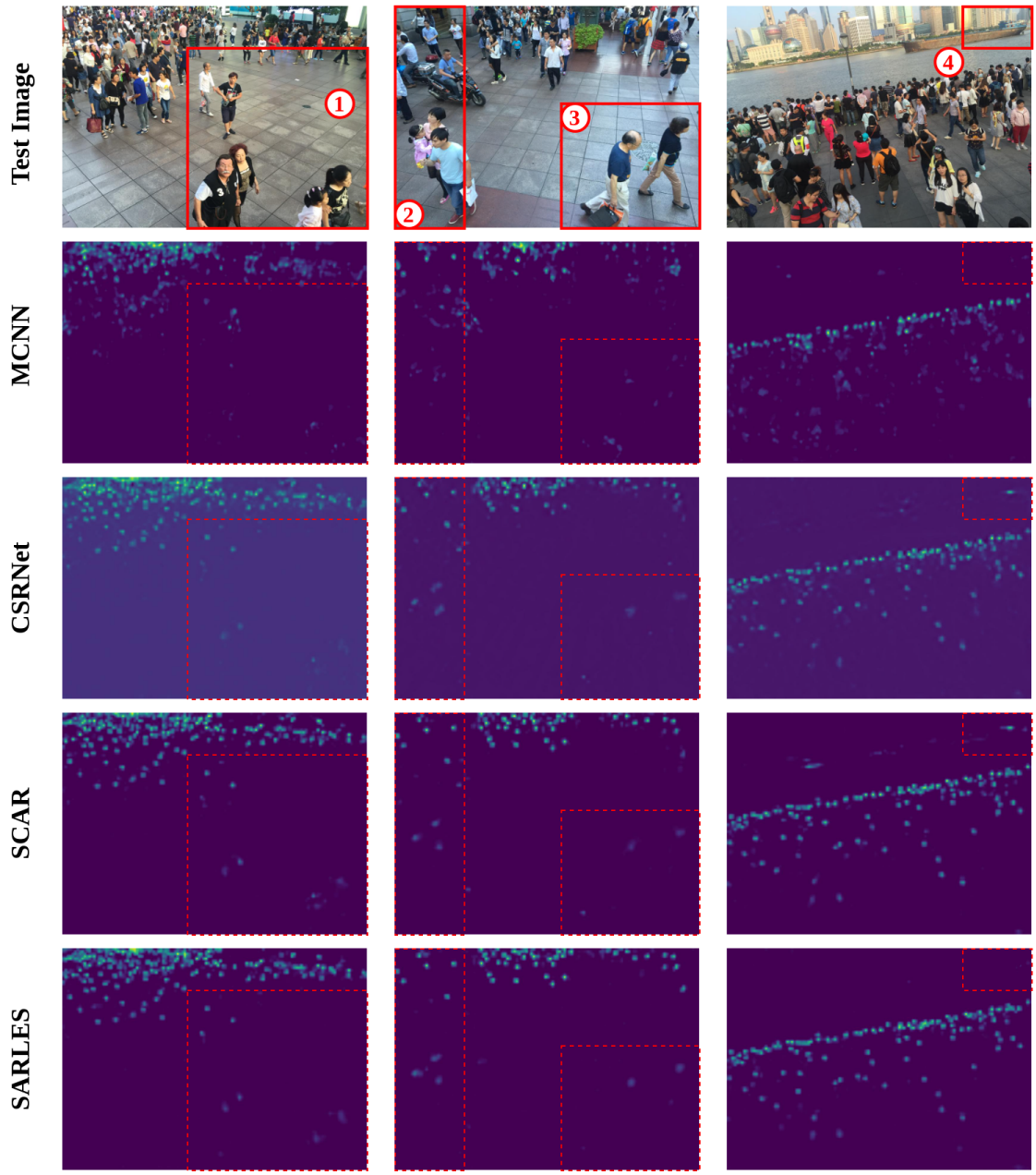

Figure 4.9: Sample results from comparison on the ShanghaiTech Part B

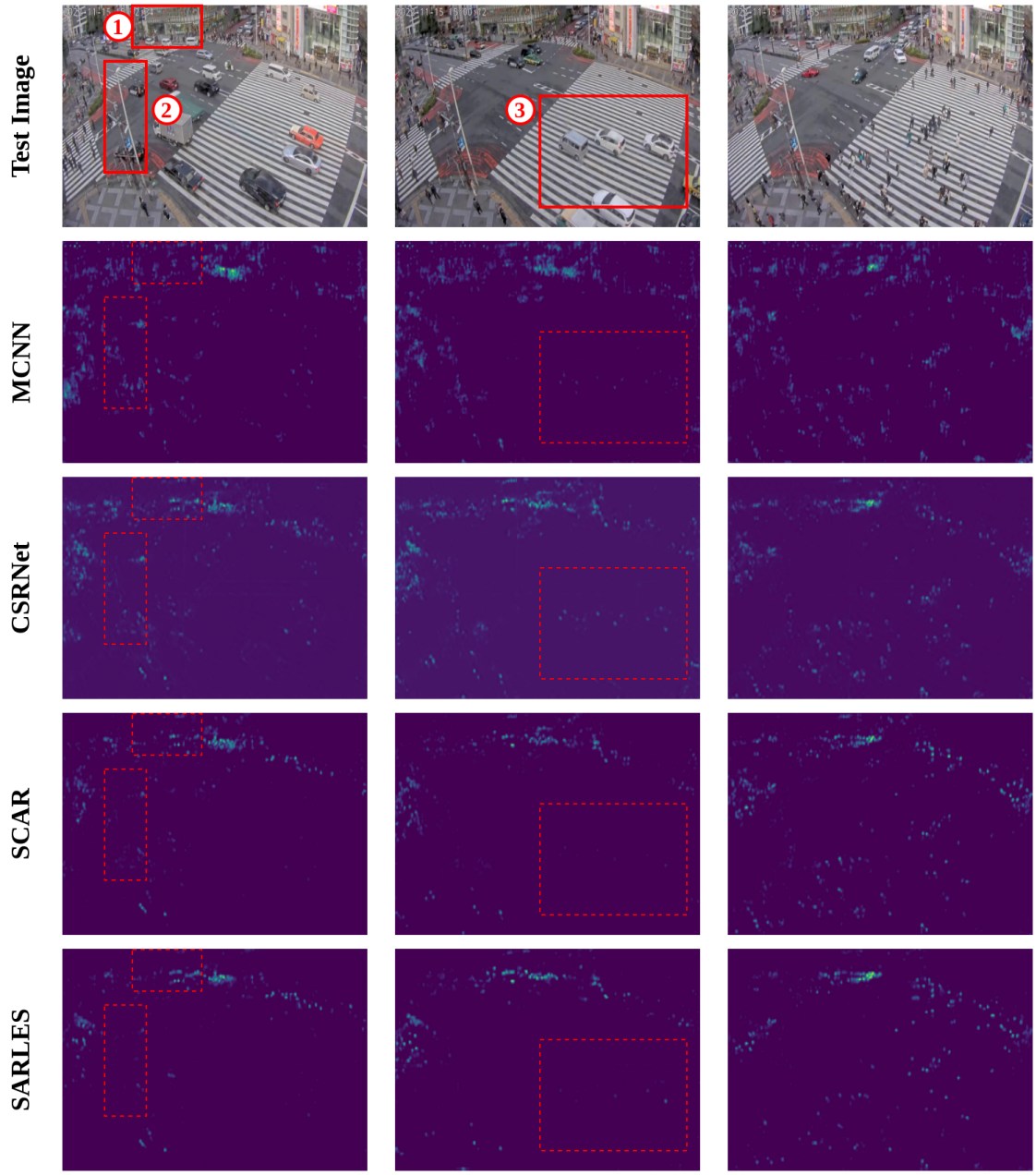

Figure 4.10: Sample results from comparison on the self-collected dataset

4.4 Conclusion

In summary, this study addresses the common challenges for accurate pedestrian sensing in transportation scenarios, including occlusion, complex backgrounds, scale variation, diverse distribution, perspective changes, small objects, and blurred regions. Then the proposed SARLES utilizes a scale-aware approach to capture multi-scale features and perceive representation information for more accurate pedestrian sensing. The system comprises three innovative modules: Encoder-decoder, DMSC, and LPR. Firstly, the Encoder-decoder module captures global contextual information and handles the scale variability of pedestrians, improving the quality of the final outputs by 98.27% in the ablation experiment. Secondly, the DMSC module segments the initial density map into multiple local patches based on the density features and generates patches with stable scale and density distributions. This module results in an average improvement of 20.54% on MAE, 29.17% on MSE, 18.94% on PSNR, and 8.71% on SSIM in the ablation experiment. Finally, the LPR module utilizes an ensemble FCN network with various kernel sizes to extract accurate local features from every input patch based on its density and scale features for density map refinements. In the ablation experiment, the LPR module improved the average performance by 21.08% on MAE, 26.85% on MSE, 13.29% on PSNR, and 7.34% on SSIM. Integrating three innovative modules addresses common pedestrian detection challenges, making SARLES a powerful tool for detecting and counting diversely distributed pedestrian groups in complex transportation scenarios.

Part III

Cooperative Sensing Technologies with Distributed Machine Intelligence

5

Chapter 5. Real-time IoT System for Multi-camera Vehicle Re-identification

This chapter is modified from the published work:

- C. Liu, H. Yang, R. Ke, M. Zhu and Y. Wang*. "Real-time IoT System for Traffic Sensing by Edge Computing and Multi-camera Vehicle Re-identification (ICTD32-RISTS)", Podium Presentation of AI & Big Data Track, Sensing and Data Analytics, Proceedings of 2022 ASCE International Conference on Transportation and Development (ICTD),

May 2022. Seattle, USA.

- H. Yang, J. Cai, M. Zhu, C. Liu and Y. Wang*, 2022. "Traffic-Informed Multi-Camera Sensing (TIMS) System Based on Vehicle Re-Identification", in IEEE Transactions on Intelligent Transportation Systems. [213, 25]

5.1 Challenges and Motivations

With the rapid development of video processing and communication technology, surveillance cameras are widely implemented in Intelligent Transportation Systems (ITS). Today, the surveillance cameras deployed in the road network are not only for transmitting the real-time traffic data, but also for traffic sensing, transportation data analysis, road surface condition estimation, and security management [214, 215, 216, 217]. Currently, in each city, hundreds of cameras are distributed in the road network. However, the system has not been fully exploited: cameras are isolated and can only extract information from their own Field Of Views (FOVs). To transmit the raw video to the Traffic Management Centers (TMCs), huge data streams needs to be transmitted, and large volume storage resources need to be allocated to the surveillance system. Although video-based methods have been well-investigated by previous researchers, however, most of them focus on single-camera information analyses, and only can be correctly deployed on the powerful servers in TMCs. After post-processing by algorithms, the cross-camera information (i.e., the same vehicle captured by various cameras) needs to be manually checked and summarized by labors, which is unacceptable and unaffordable for most of agencies.

To extend the traffic manager's eyes from a single camera to a camera system and facilitate traffic sensing on a network level, Multi-Target Multi-Camera Tracking (MTMCT) and multi-camera vehicle Re-identifiction (vehicle Re-ID) related works are emerged [139, 143]. In the pra-

mary MTMCT framework, the system is feasible to detect and track objects across various cameras with overlapping/non-overlapping views. Specifically, the MTMCT technology workflow includes two steps: 1) single camera-based information extraction, generally including Multi-Object Detection (MOD) and Single Camera Tracking (SCT); 2) extracting and matching the same objects captured by distinct cameras and further associating them from various FOVs by Vehicle Re-ID; With the booming of vehicle Re-ID algorithms development, researchers can make the cameras to a real system and extract the information from multiple selected sensing nodes together.

However, to apply the cutting-edge complicated researches for systematical traffic sensing, challenges are almost everywhere. The first and foremost one is limited computing storage resources. As mentioned above, MTMCT and vehicle Re-ID inputs are based on every single camera multi-object detection and tracking. Running detection and tracking algorithms for tens even hundreds of cameras simultaneously is too expensive to implement for TMCs. Secondly, using the video generated by various cameras as inputs, the individual differences of them cannot afford to ignore. The resolution, lighting condition, frame rate, and orientation for each camera deployed in the road network sometimes have huge dissimilarities. In this case, without individual features, using a unified Re-ID algorithm deploying on cloud or center servers for all inputs can inevitably lead to a significant decrease in accuracy. Thirdly, even obtaining multi-camera vehicle tracking and Re-ID results, the accuracy levels vary in different camera viewpoints, and at different locations. How to efficiently use the outputs to sense and summarize traffic parameters is becoming a significant hurdle.

With the increasing computing power on the edge devices, the researchers see the light at the end of the tunnel. With the popularize of IoT technology, processing the raw video data on

distributed edge nodes, then selecting the representations of the objects is becoming a potential way to achieve the network scale traffic sensing. Instead of streaming raw video data to TMCs 24/7, edge computing resources allowed each node to filter the original input, transmit the useful object images to servers, and then use them as the input for the vehicle Re-ID algorithm. Under the circumstances, each distributed edge node can contribute to the whole framework and the pressure of computing resources for TMCs can be decomposed a lot. Also, much more communication bandwidth can be saved for use in the V2X tasks. A huge amount of raw video storage space can be economized by only collecting helpful recording clips, instead of raw data.

However, to build a hybrid IoT system for traffic sensing based multi-camera information extraction is beset with difficulties. The first and foremost is how to boost the objects candidates selection algorithm in real-time with high accuracy. Although real-time MOD has been verified to be feasible in many edge devices [218], however, how to make online tracking affordable is an obstacle that needs to be overcome. Algorithm optimization, efficiency improvement and consideration of differences in various FOVs are necessary. Secondly, after obtaining the tracking results with different object tracks, suitable represent images of each object need to be further investigated. Additionally, reliable data transmission for multi-camera tracking and Re-ID is indispensable, especially for such a time-sensitive (vehicles passing a camera normally in several seconds) task. Furthermore, different from well investigated image-based and video-based Re-ID, automatically cross camera objects matching is highly related with the Rank-1 accuracy. Re-design and improve SOTA Re-ID frameworks and customize an object Re-ID algorithm based on short reorientation clips need to be developed. Besides, instead of visual features, find a possible solution for utilizing the roadway geometric features, road network graph node and link connectivity, and spatial-temporal constraints to aid the limited clip information for cross-

camera objects Re-ID process is a topic for further exploration.

In summary, the authors proposed a comprehensive hybrid IoT system for network-scale traffic sensing solution by edge computing and Vehicle Re-ID – Real-time IoT System for Traffic Sensing (RISTS). The team claims the technical contributions as follows:

- A comprehensive hybrid IoT workflow RISTS for network-scale traffic information estimation has been investigated and tested. Four components are included: edge nodes objects representations selection; cross-camera objects Re-identifiction, traffic information estimation; and TMCs traffic manager interactions. This is the first IoT system for cross camera traffic information estimation to the research team best knowledge.

- To boost the edge objects representations selection framework works in a real-time manner, each edge node is integrated with the NVIDIA AGX Xavier embedded edge-computing platform, running the adapted and optimized deep learning based video processing framework, to achieve the vehicle detection, tracking and representation selection over the streaming pixels in a real-time manner. Based on the experiment on the real dataset collected in midsize USA city, the edge achieved average MOT IDF1 at 81.84%, 76.63%, 73.91% and 71.42% with 34.4, 33.6, 34.2 and 33.9 FPS at the light, normal, busy and heavy traffic conditions, respectively.

- To match the edge inputs and get precise and reliable Rank-1 result, a novel Re-ID framework is proposed and integrated into the RISTS workflow. Four hierarchical levels of features, including frame-level, clip-level, identity-level (vehicle attributes information) and network connectivity information (road graph constraint) are integrated into the process. An attributes-aware multi-query to candidate re-ranking mechanism is designed to

improve the best candidate match performance. Based on our extensive experiments on two datasets, our Re-ID algorithm significantly outperforms other SOTA methods with 90.14% and 92.43% on Cityflow and Freeway Sensing Video (FSV) dataset, respectively.

- With RISTS, the traffic sensing scale extended from point-based to the link and network scales. To extract the accurate information from various camera pairs with difference Re-ID accuracy, a precision-aware kernel density estimator is integrated in RISTS instead of using the Re-ID result directly. By comparing with the metadata, the link travel time and speed distribution estimation can achieve less than 1.01 KL distance. The area network OD extraction is within 2% error.

- The RISTS is an affordable, easy to scale up, less bandwidth and power consuming solution. By maximizing the potential of the hybrid IoT architecture, RISTS can save much more money on the server hardware, including both GPUs (only with 25%) and data storage (10% of data volume). Considering hundreds of traffic sensing cameras are necessary in a city, the total cost will be only 30% or less with a much more comprehensive network sensing ability.

5.2 Methodology

5.2.1 Overall Framework Architecture

To be brief, four components are involved: edge nodes, communication system, edge server and TMC control center. TMC is the control center of the whole system for admitting the input from traffic managers to visualize the results. Including n edge nodes are separated in the road network. The object detection, single-camera tracking, clips selection and generation are simul-

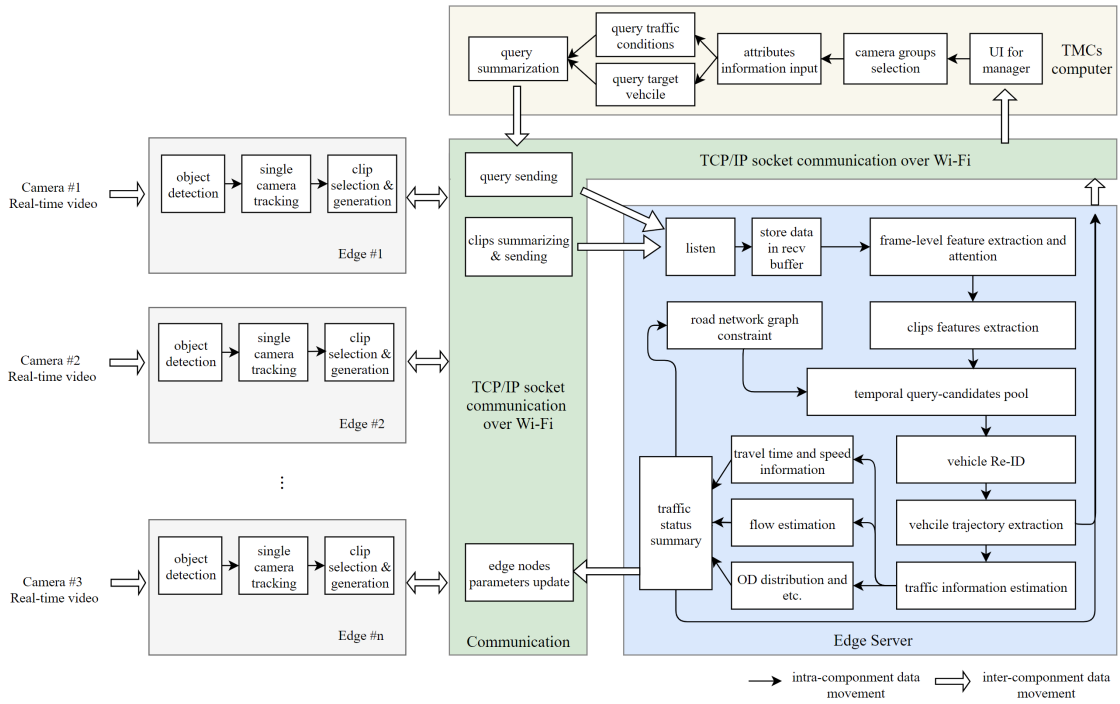

Figure 5.1: Workflow of RISTS

taneously running on the edge devices. Via the TCP/IP socket-based on WiFi, clips are summarized and sent to the edge server. Then, the edge server is used to finish heavy computation tasks, including the cross camera objects Re-ID, traffic information estimation and send the necessary parameters to all edges and the TMCs.

In general, there are four components are involved: multiple edge nodes, communication system based on TCP/IP socket, the edge server and TMC control server. The FIGURE 1. shows the workflow of RISTS. In RISTS, TMC are the control center of the whole system. The system can admit the input from traffic managers, including attributes information (weather condition, road type), choosing camera groups and also use to visualize the traffic network information estimation results. Then, N edge nodes are separated in different locations of the road

network. Each of them are equipped with an NVIDIA Jetson AGX Xavier to support real-time object detection, single-camera tracking, tracker clips selection and generation. Via the TCP/IP socket-based on WiFi, tracker clips and the spatial-temporal information are summarized and sent to the edge server. The last component in the RISTS is the edge server, which is the central node for listening and summarizing the belonging edge nodes' information and finishing the cross camera tasks. The tasks include clips features extraction, candidates filter and selection, cross-camera vehicle Re-ID, traffic information estimation, and communicating the necessary parameters to all edges and the TMCs.

5.2.2 Edge-side Methodology

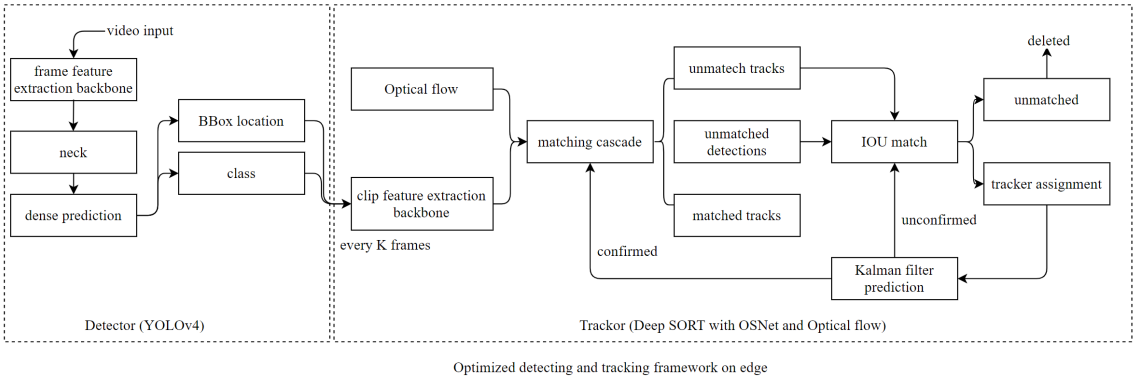

Figure 5.2: Optimized detecting and tracking framework on edge device. The YOLOv4 detector and Deep SORT tracking with OSNet and Optical flow are customized for real-time processing.

Real-time Multi-object Detection and Tracking

Deep learning feature extractors are usually the bottleneck in online tracking algorithms, which makes them unscalable for real-time applications, not to mention running on the edge devices. To achieve faster processing on the edge side, we choose to use YOLOv4 [85] and Deep SORT

as basic structure. YOLOv4 [120] is a famous single stage detector with a CSPDarknet53 backbone, SPP neck [219] and YOLOv3 [84] Head, which can achieve the balance of processing speed and accuracy. In the experiment, we retrained the detector by two datasets: MIO-TCD [220], and LHTV [221].

Inspired by the current state-of-art joint detection and embedding [222], Tracktor [223] and Fast MOT [224], the researchers designed a mechanism for the tracking feature extractor. As shows in the FIGURE 2. in the practice, the feature only extracted every K frames and then using as the frame-level features. Since the surveillance cameras are always mounted at a certain point with a stable background, optical flow [225] is then used to fill in the gaps. Towards a lightweight and reliable feature extractor for clips, the team chooses to implement the omni-scale network (OSNet) [226]. In OSNet, to efficiently capture the spatial dependencies and avoid introducing many parameters, the building block is operated through pointwise and depthwise convolutions. A unified Aggregation Gate (AG) is integrated into the tracking framework for dynamically fusing multi-scale features with various input-dependent channel-wise weights. Based on the customized design, OSNet achieves impressive feature extraction results with a featherlight architecture.

Clip Selection and Generation

After the real-time detection and online tracking procedures, the object are cut into different tracks. The nest step is to select $\mathscr{N}$ frames of a track with the high quality representativeness and consist into a clip $\mathscr{C}$. Here, the selection rules are related with several factors, including the object detection confidence (θ), object size ($\mathscr{S}$) and the attributes input (weather condition and road type). A size normalization is integrated into the selection process by regarding the

biggest size frame of a track as unit. Then, a confidence θ_i is used as the bottom value to filter the low-confidence frames. Define the object frame k score as ρ^k, Then calculate the score for each frame for each tracker:

$$\rho^k = \{\theta^k * \mathscr{S}^k | \theta^k > \theta_i\} \tag{5.1}$$

With the value of ρ^k, the next step is to select top $\mathscr{N}$ frames with largest result of ρ^k as representative frames to consist a $\mathscr{C}$. The last task for the edge is to send the $\mathscr{C}$ for each track to the sever for further processing by TCP socket protocol. It is worth noting that for different attribute inputs, the θ_i and $\mathscr{N}$ can be changed by the real situation. The detail parameter settings are in the experiment section.

5.2.3 Server-side Methodology

After receiving the the clips and attributes information from edge devices, the server keeps running four threads simultaneously to match the information among multiple cameras. The whole processes show in FIGURE 3.

Road Network Graph Constraint

In order to save computing resources, before extracting vehicle representations, the researchers use road graph restrictions and travel time restrictions to filter the set of potential matching targets. The first step is to build the adjacent matrix $\mathbf{A}$ for the installed cameras. Based on the connection information of $\mathbf{A}$, the second step is setting up the query and gallery camera pair $\mathbf{P}_{ij}$, which represents matching the objects using the camera i as query set and j as candidate set. The third step is to set the match time constraint. Based on the transportation domain knowledge

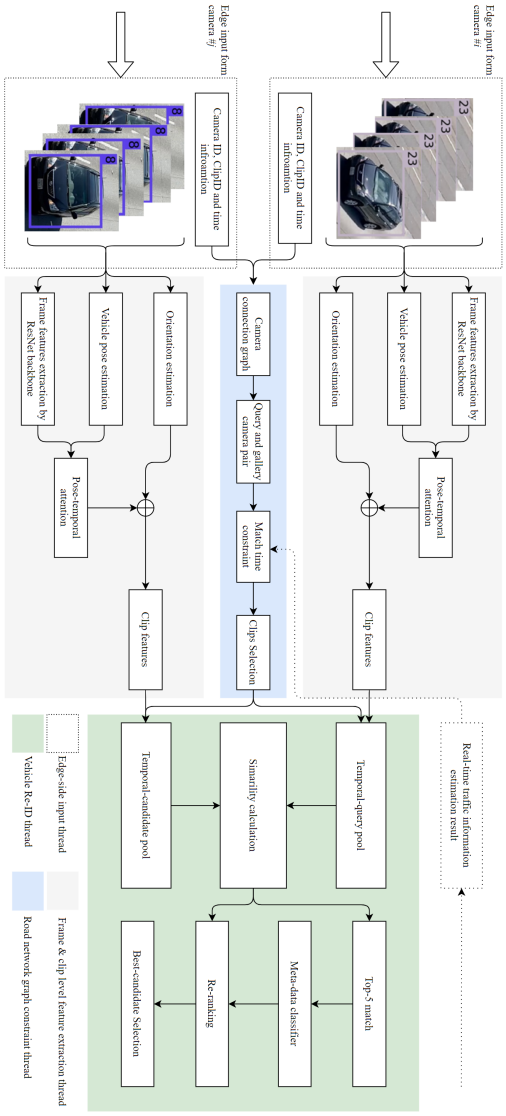

Figure 5.3: RISTS_Re-ID workflow on edge servers visualization

support, the research team using a matching time window by considering the real-time travel time reliability. Considering the average travel time at time t is t_{ij}, considering a buffer index (b_{ij},

varies with time of a day), and then the matching time window length is

$$T_{P_{ij}} = (t_{ij} \cdot (1 - b_{ij}),\ t_{ij} \cdot (1 + b_{ij})). \tag{5.2}$$

The detail value of the b_{ij} is various for different cities at different time of a day. In this paper, the author using the b_{ij} by referring the city's congestion level and the travel time reliability summarized in [227].

Frame-level Feature Extraction

After receiving the clips (I) from edge nodes, the frame-level features extraction is designed for capturing necessary information using for vehicle Re-ID. Here, the research team extracted the features of each single frame from the ResNet50 [228] backbone pre-trained on the ImageNet [229] and CityFlow [230] datasets. The 2048 dimensional vector of the fully-connected layer is treated to represent the appearance features ($f^I_{a^i}$) for each frame.

From previous research, pose and orientation shows significant impact for the human and vehicle Re-ID result. So, the team integrates the car key points estimation to represent the pose and structural features. In detail, by using the car key points network estimation proposed by [231], 18 vehicle surfaces can be obtained by each input frame. Then, by concatenating the nodes of surfaces in order, the researchers can convert the 3D information into a 72-dimensional vector. Then, embedding the obtained vector to a 2048 dimensional vector to represent the vehicle pose features ($f^I_{p^i}$).

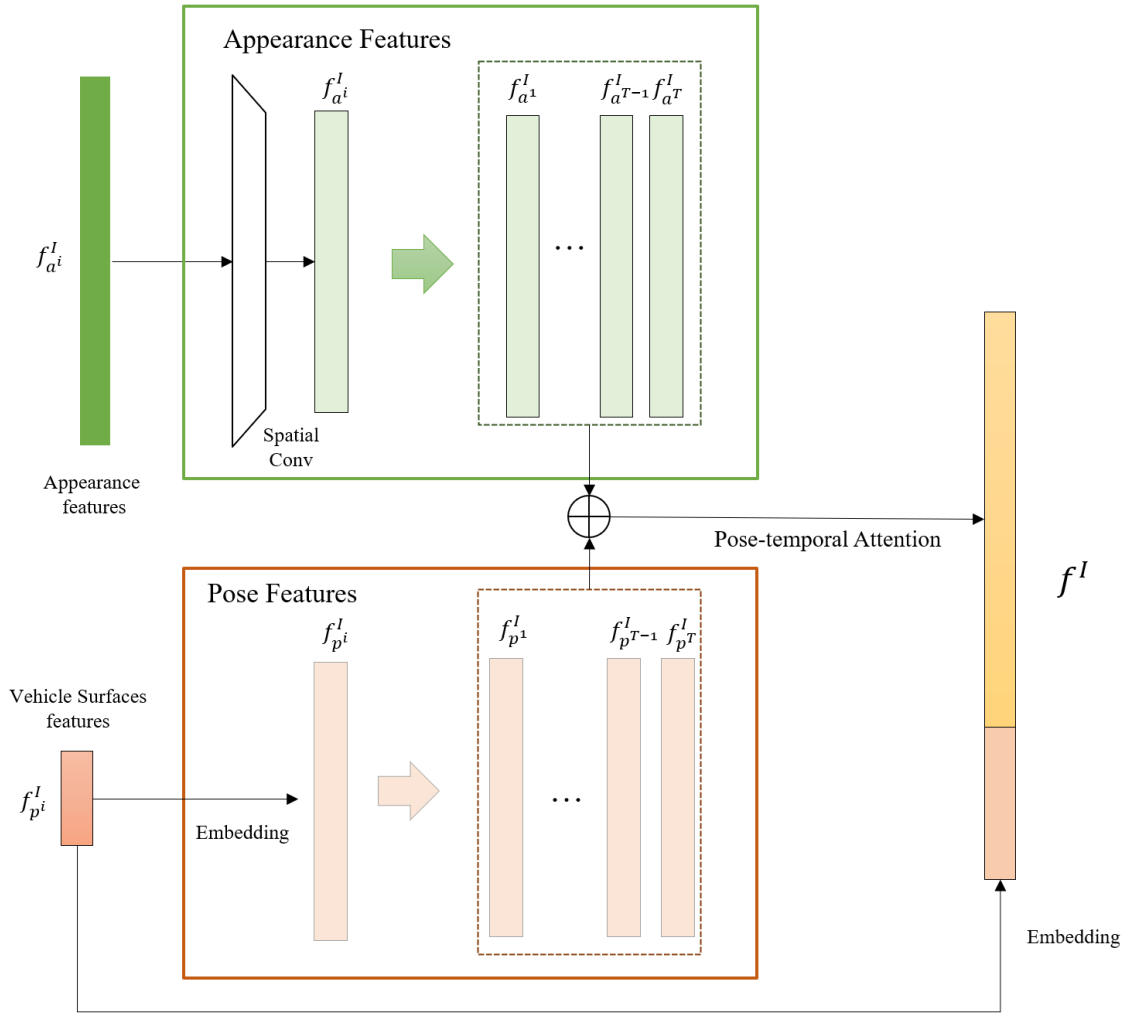

Figure 5.4: The pose-aware clip-level feature extraction component visualization

Clip-level Feature Extraction

After obtaining the frame-level features, the team uses the pose-temporal attention mechanism to fuse $\mathscr{C}$ frames features into a clip-level feature (f_c^I). The detailed process is shown in FIGURE 4. For the appearance features fusion, a 2D convolution operation is used to capture the spatial correlations of neighborhood frames. Meanwhile, for better utilizing the vehicle pose features,

an embedding process is designed by multiplying a parameter matrix of $\mathbf{W} \in \mathscr{R}^{72*2048}$. Then, apply pose-attention to the two vector sets with the same size of $\mathscr{R}^{2048*T}$ and fuse into a final vector. Then, to leave the raw pose information for further comparison, another smaller embedding is used to map the $(f_{p^i}^I)$ into a 256 dimension vector and then concentrate with the output of the pose-attention. Both vectors constant the final clip feature (f_c^I) of clip I.

Vehicle Re-ID by Clips

After obtained the features of each clips, then the similarity calculation can be measured by the Euclidean distance. Only top five closest targets will be used as the alike candidates sets $(\mathscr{A})$ and then considered for further re-ranking.

The loss function of the vehicle Re-ID is consists of two parts. Considering the task somehow contains classifications, the cross-entropy (*Xent*) loss is included.

$$\mathscr{L}_{Xent} = -\sum_{i=1}^{P} \log\left(p(i))q(i)\right),\ q(i) = \delta_{i,j}, \tag{5.3}$$

Meanwhile, inspired by metric learning used in human Re-ID, triplet loss is widely used for distinguishing intra-class and inter-class differences. Batch hard (BH) triplet loss [232] shows promising results. However, a significant fundamental difference between vehicle and face Re-ID tasks, which is no people with the same appearance in the world, but there are cars with exactly the same appearance features. If training the vehicle Re-ID model using BH loss, the process would be too tough to obtain reliable results if the same vehicle exists. So, in this research, the team uses batch sample (BS) instead [233, 139] of BH triplet loss into model training. In BS loss, a mini-batch $\mathscr{B}$ is defined in the following equation:

$$\mathscr{L}_{BS}(\theta;\mathscr{B}) = \sum_{\text{all batches}} \sum_{a\in B} l_{tri}(a), \tag{5.4}$$

Where

$$l_{tri}(a) = [m + \sum_{p\in P(a)} w_p D_{ap} - \sum_{n\in N(a)} w_n D_{an}]_+, \tag{5.5}$$

In equation (5), w_p are the weights of positive samples, w_n are the weights of negative samples, D_{ap} are the distances of anchor sample to the positive samples, D_{an} are anchor samples to the negative samples, and m represents the predefined margin value.

The final loss function of the proposed RISTS_Re-ID methodology is a combination of both BS triplet loss and cross-entropy loss, as shown in the following,

$$\mathscr{L}_{ReID} = \lambda \mathscr{L}_{BS} + (1-\lambda)\mathscr{L}_{Xent}. \tag{5.6}$$

Attribute-aware Re-Ranking

To obtain cross-camera traffic information, the toughest part is a reliable matching result with precise top-one accuracy. In this research, the team integrates a light attributes classification of the final alike candidates' sets ($\mathscr{A}$) to choose the best one. Since the appearance features and pose features have already been well used in previous steps, object attributes features, including vehicle types, brand, color and module are included in the process. For the balance of computing time and classification accuracy, Light-CNN, proposed by Wu et al. [234], is adopted into this process. The MFM_fc1 layer with 256 dimensions, and then expanded into 512 by a fully-connected layer is used for final classification features. Using p_i and p_j as the metadata probabilities, for samples i and j from classes c_i and c_j, then the meta-data distance can be shown

in the following equation:

$$\mathscr{D}_m(q,gi) = \sum_n d_n(q,g_i), \tag{5.7}$$

where q represents a query clip and g_i represent the i_{th} candidate. In detail,

$$d_n(q,g_i) = c(p_i) \cdot c(p_j) \cdot (-log_n P(c_i = c_j | p_i, p_j)). \tag{5.8}$$

And the $c(p_i)$ is the KL distance between p_i and the uniform distribution. The n is the number of classes defined in the dataset.

Finally, the candidates are re-ranked by the distance in the equation (9)

$$\mathscr{D} = \mathscr{D}_e(q,gi) + \beta \cdot \mathscr{D}_m, \tag{5.9}$$

where the $\mathscr{D}_e$ is the euclidean distance obtained from the clips Re-ID. The β a hyperparameter that can be tuned in the experiment.

5.2.4 Traffic Sensing by Vehicle Re-ID

Point-level Traffic Information Extraction

Through each node deployed in RISTS, point-based vehicle counts can be obtained by the detection and tracking algorithms in real-time. Meanwhile, based on the general definition that the traffic flow rate as the number of vehicles passing a point in a given time period usually expressed as an hourly flow rate. RISTS can easily obtained the flow rate by summarizing the count value every per time unit (i.e., per hour).

Link-level Traffic Information

Cross camera information extraction enable RISTS can be used for estimate link-level traffic information (i.e., link travel time, speed) in a high penetration rate. Firstly, several parameters need to bed defined: T_r, T_l represents the thread time length for vehicle Re-ID to select the best match candidate and time length for extracting cross camera link information from the best candidate match pool. In this research, T_l is consists of $\mathscr{M}$ continuous T_r.

For the link information estimation, the input are the clips pairs in the best candidate pool and their attributes information. Assume the time for object vehicle v passing the camera i is t_i^v and then passing the camera j is t_j^v, then the travel time of v from i to j is t_j^v minus t_i^v. Suppose we have $\mathscr{V}$ vehicles in the pool, then calculated all the vehicles' travel time and speed (link distance over travel time). Based on the values, estimates the real-time cross camera travel time distribution based on Kernel Density Estimation (KDE) and Kernel Smoother (KS) [235]. In the traditional way, researchers always use a parametric probability approach to estimate the cross-camera traffic information (i.e., travel time, speed, etc.) distributions. The basic assumption of the proposed approach is the traffic information distributes similarly to a specified distribution and then estimation necessary parameter based on the observed data. However, the large errors and limited capability of handling complex scenarios make these methods unsuitable for cross-camera scenes. Here, the research team introduces KDE and KS as a nonparametric approach for estimating probability distributions. The proposed approach enables the construction of travel time distribution and speed distribution with the advantages of fewer assumptions and a more flexible fitting structure with fast processing speed. In detail, the density of the evaluated variable ($\mathscr{X}$) by the KDE and KS is shown mathematically as:

$$f(\mathscr{X}_{ij}) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_o}{h}). \tag{5.10}$$

In equation (10), $\mathscr{X}_{ij}$ are variable need to estimated, including speed (ls_{ij}) and travel time (t_{ij}). n is the number of observations and h is the kernel bandwidth that controls the smoothing level of PDF. K represents the kernel function and the researchers using the Gaussian kernel here. With the $f(t_{ij})$ and $f(s_{ij})$, information like link speed distribution and travel time reliability, congestion level can be obtained.

Network-level Traffic Information

After obtaining the vehicle Re-ID information between adjacent cameras, combined with the graph of the road network, the trajectory of the vehicles can be extracted. Accumulated the trajectory by every time period (network information extraction thread time length, T_n), area Origination and Destination (OD) information can be extracted with the channelized characteristics of roads. Subsequently, the utility level, the traffic flow distribution, and the OD distribution can be extracted in realtime. To the authors' best knowledge, this is the first reliable traffic flow and OD distribution extraction framework in real-time with a high penetration rate.

5.3 Experiment

5.3.1 Dataset Description

Cityflow

The Cityflow Vehicle Re-ID dataset is proposed by NVIDIA in 2019 [230], which was captured by 40 traffic cameras, including the scenario of intersections, street roads and highways. In this

research, the Cityflow dataset is used to train and evaluate the detection and vehicle Re-ID algorithm performance deploying on the edge server. In detail, a total of 666 vehicles are annotated with distinct vehicle IDs are used for training and testing the cross camera vehicle Re-ID based on clips. License plates are masked in black for privacy consideration. Due to the testing query set are only a single image, is not well suited into the clip-based scenario. So we use the original 20% of the training set as the testing set.

Freeway Sensing Video (FSV) Dataset

Due to the time length of each surveillance video clip in Cityflow dataset is too short (average length is only 4.88 min) and not included the ground truth timestamp, a whole new dataset Freeway Sensing Video (FSV) dataset is proposed by authors to demonstrate the traffic sensing and information estimation evaluation.

FSV dataset includes 4.31 hours (258.35 minutes) video and includes four different cameras on Interstate 5 freeway, with the video format of 1080p/30 fps. The illustration of the FSV dataset is shown in the FIGURE 5, with the FSV cameras (from camera #1 to camera 4) location, orientation and view. All the image frames include ground truth time and the camera location GPS information. Among the camera links, the longest distance is 3216 m, and the shortest is 405 m. In this research, a total 40.12 minutes of video is used to fine-tune the detection, tracking and Re-ID algorithms, and 218.23 minutes of video are used to evaluate the traffic sensing framework performance. In evaluation, using the long video as input of the edge devices for real-time detection and tracking, and then send the clips to the server with spatial-temporal information for cross camera vehicle Re-ID and traffic information estimation.

Figure 5.5: The detail information of FSV dataset, including 4 camers locations and view point

5.3.2 Edge-side Experiment

Edge-side Workflow and Algorithm Adaptation

The Nvidia Jetson AGX Xavier is choose as an embedded edge platform for real-time vehicle detection, tracking and clip selection, integrated with an 1080P camera. The Xavier has eight

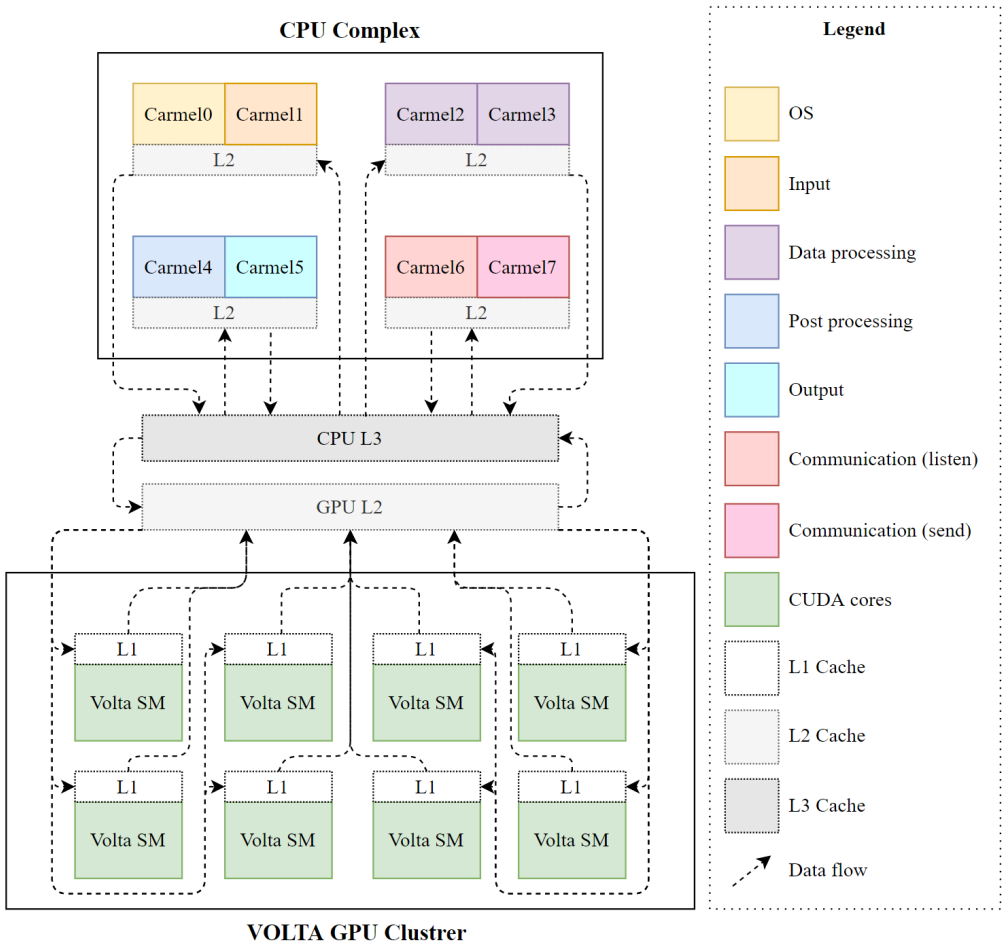

Figure 5.6: The combination of RISTS edge nodes workflow with Jeston Xavier computational resources

Carmel CPU and eight Volta SM GPU to consist the CPU and GPU cluster. To fully stimulate the computing potential on edge nodes, the entire work flow running on the edge side of RISTS is combined with the reasonable Xavier resources without overlap in one cycle. As FIGURE 6. shows, the CPUs are divided into seven groups for handling OS system, input preparation, data processing for algorithms, output preparation and output control, then one for communication receiving orders from TMC and another is used to transmit to data to edge servers. Then, the

GPU cluster is incharged for handle the detection and tracking script. The CPU and GPU cluster share the information through the cache for each cycle. To speed up the whole edge framework, the detector and feature extractor use the TensorRT backend and perform asynchronous communication inference. In addition, all code finished by Python language, including Kalman filter, optical flow, and data association, are optimized using Numba.

Edge-side Parameters

The parameters using on the edge nodes are as follows:

- In the multi-object tracking, the algorithm extract features every K frames, and add the associations by optical flow. Here the K is equal to four.
- $\mathscr{N}$ frames of a track with the high quality representativeness is five.
- the object detection confidence (θ) base line is equal to 0.75.
- object size ($\mathscr{S}$) here is no less than 500 pixel.

Edge-side Detection and Tracking Performance

For the real-time multi-object detection, the authors pre-define three classes, including truck, bus and car. The average precision of detection are 81.40%, 78.33% and 96.92% for the four camera views together, respectively. Set the IoU threshold as 50%, then the Area-Under-Curve (AUC) for each unique recall mean average precision (mAP@0.50) is 85.55%.

The team evaluated the real-time multi objects tracking result separately under different traffic conditions (light traffic (less than 50v/min), normal (50v/h to 85v/h) and busy (85v/h to 120v/h)

heavy (more than 120v/h), with the accurate result and the computational performance. The results are summarized in the Table 2.

Table 5.1: The Multi-object detection and tracking performance on four RISTS edge nodes

Traffic condition	Camera #1			Camera #2			Camera #3			Camera #4		
	IDF1	MOTA	FPS	IDF1	MOTA	FPS	IDF1	MOTA	FPS	IDF1	MOTA	FPS
Light	81.76	64.31	34.4	82.12	65.43	33.6	78.24	61.52	34.2	85.24	67.45	33.9
Normal	76.23	61.61	26.1	77.07	63.05	25.4	73.42	56.62	25.2	79.78	63.92	26.9
Busy	74.47	60.01	19.3	75.04	61.78	18.2	68.97	54.81	18.7	77.14	62.13	20.1
Heavy	71.23	59.22	16.3	74.05	60.63	17.1	66.17	51.48	16.8	74.23	60.61	17.4

As shown in Table 2, The scripts are deployed on edges can perform real-time MOD, MOT and candidates selections process with reliable MOT IDF1 score. The camera #4 performs best, which gets 85.24, 79.78, 77.14 and 74.23 on the light, normal, busy and heavy traffic conditions with 33.9, 26.9, 20.1 and 17.4 FPS, respectively. Even the camera 3 performs lowest among four nodes, the average IDF1 still scores 71.70. Considering the general surveillance video input always use 10-15 FPS as standards, such a performance can satisfy all kinds of real time traffic multi-object detection and tracking with post processing.

5.3.3 Vehicle Re-ID Experiment

Vehicle Re-ID Environmental Setting

After the vehicle representations are selected and sent to the edge server by the internet; the clip-based Vehicle Re-ID framework is triggered. In RISTS, the edge server is equipped with A CPU of Intel Core-i9 9900K and two GPUs manufactured by NVIDIA. The cost of the server is around $3500. The server can support up to eight RISTS edge nodes input. The operating system is Linux system and the RISTS_Re-ID is implemented by PyTorch.

RISTS Vehicle Re-ID Parameters

The parameters using on the edge nodes are as follows:

- b_{ij} in this work is used as the travel time reliability index in the data collection region, which is open source and provided on Digital Roadway Interactive Visualization and Evaluation Network (DRIVE Net) platform developed by the Washington State of Department of Transportation and University of Washington.

- The clip level features (f_c^I) is fused by $\mathscr{C}$ frames, and here the $\mathscr{C}$ is equal to four.
- Each vehicle can be obtained two clip level features (f_c^I) by fuse frame zero to three and one to four, then used for re-ranking together.
- The λ in equation (6) is 0.5.
- The β in equation (9) is 0.4.

Vehicle Re-ID Results Summarization and Comparison

Figure 5.7: The RISTS_Re-ID Framework Result Visualization on FSV and Cityflow dataset

To show our proposed RSITS_Reid's potential in vehicle Re-ID tasks, the research introduced four SOTA methods.

Table 5.2: The RISTS_Re-ID Framework Result Summarization and Comparison with SOTA Methods

Type	Method	Accuracy			
		Rank-1 ↑	Rank-5 ↑	mAP ↑	Infer time (s/b) ↓
FSV dataset	BoT	80.03	82.92	69.58	**0.201**
	AGW	80.55	84.05	68.22	0.226
	BoT_ibn	82.57	85.23	71.68	0.202
	SBS	82.08	85.95	72.62	0.317
	RISTS_Re-ID	**90.14**	**93.72**	**77.01**	0.309
Cityflow2021	BoT	85.45	87.86	70.11	**0.142**
	AGW	86.74	88.98	72.31	0.165
	BoT_ibn	88.79	90.58	75.57	0.214
	SBS	88.13	90.72	75.04	0.251
	RISTS_Re-ID	**92.43**	**94.87**	**80.09**	0.242

BoT [236]. Bag of Tricks (BoT) is a well-known baseline for deep human Re-ID. Here, the author re-train the model by a 256*256 ResNeXt50 as a backbone, with Global Average Pooling (GAP). Cross-entropy and triplet loss are integrated into the BoT.

AGW [237, 238]. Attention Generalized mean pooling with Weighted triplet loss (AGW) is proposed in 2020. Here, the author re-train the model by a 256*256 ResNeXt50 as backbone, with Attention Generalized Mean Pooling (AGMP). Cross-entropy and weighted triplet loss are used in the model.

BoT_ibn [236, 239]. The Instance-Batch Normalization (IBN) network is integrated into the upgrade version of BoT and called BoT_ibn in this research as a baseline.

SBS [237]. SBS is proposed in the Fast Re-ID framework by integrating the SOTA tricks used in the Re-ID tasks. Here, the author re-train the model by a 256*256 ResNeXt50 as the backbone, with on-local block (NL). The Generalized Mean Pooling (GMP) is used to down-sample the vector. Circle Softmax is used instead of traditional linear classification layers. Both Cross-entropy and weighted triplet loss are integrated into the BoT.

The results are summarized into Table 3 and visualized on the FIGURE 7., including the Rank-1, Rank-5, mAP and inference time (second per batch). RISTS_Re-ID framework significantly outperforms other SOTA methods on both datasets, especially on the Rank-1 accuracy. On the FSV dataset, the RISTS achieves 90.14% Rank-1 and 3.72 Rank-5, with the best mAP accuracy. As mentioned before, to extract link traffic information, Rank-1 accuracy is the most important measurement need to be improved. The customized attributes-aware multi-query and re-ranking mechanism play an essential role in enhancing the Rank-1 accuracy. Such a result indicates that the RISTS Re-ID results can sufficiently support traffic information estimation.

5.3.4 Traffic Information Estimation Evaluation

The final target of RISTS is to sense the information at multi-level of traffic networks, including node, link and local area. Here, the research team mainly evaluate the link and network level of traffic information estimation, including link travel time and speed, as well as network OD flow distribution.

Link Information

The link speed and travel time are two key parameters to be estimated. Different from the traditional method, for RISTS, the research team estimates the distribution instead of the average value. The FIGURE 8. shows the box plot of metadata and estimated distribution. From the result, both the travel time and speed are obtained a very close distribution (with less than 1.01 KL distance) for three camera link. For the link travel time, the distribution of Cam1-2 and Cam3-4 are obtained 0.09, 0.11 KL distance respectively, which can be treated as the same. For the Cam2-3, due to two exits are located in the road section, as well as the road network graph constraint,

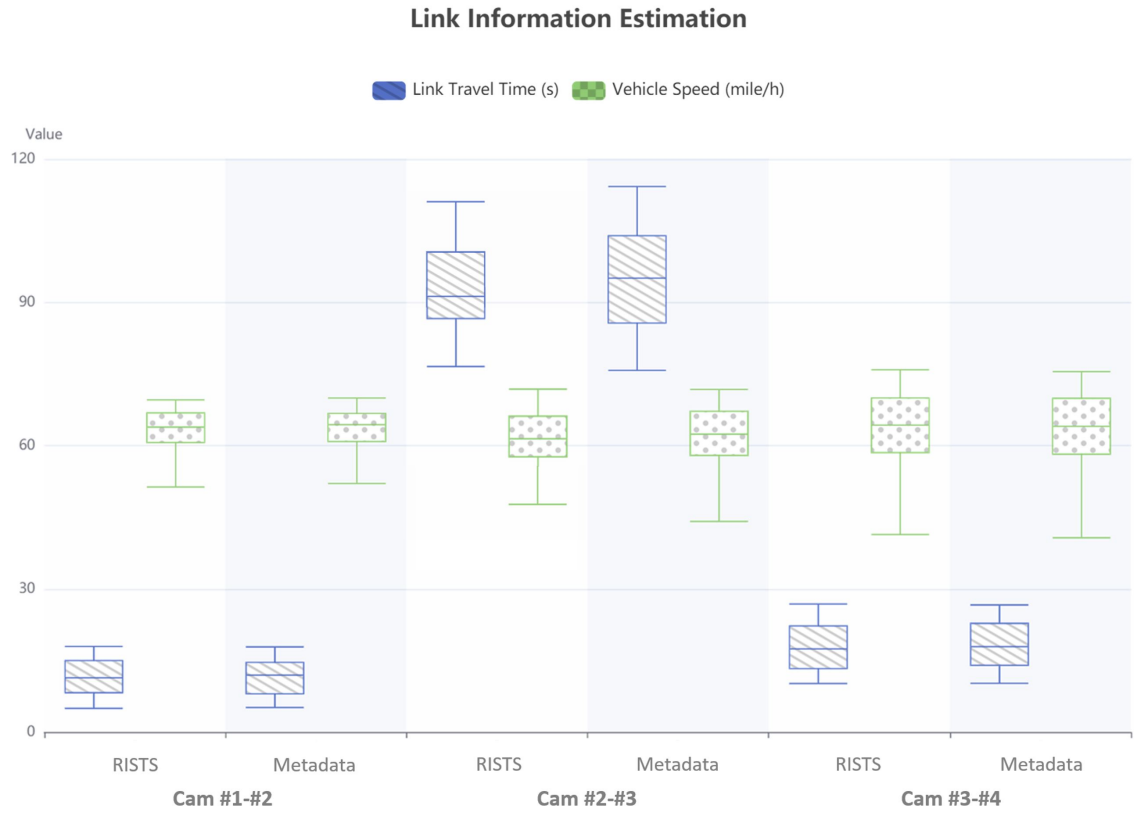

Figure 5.8: Link travel time and average speed distribution estimation comparison

the estimated distribution are more concentrated compared with the metadata. However, the general distribution is quite close (KL distance 1.01 for link travel time distribution and 0.94 for link average speed distribution). Such an accuracy level can satisfy all kinds of challenging traffic-related services inducing travel time reliability analysis, road network resilience analysis and etc.;

Area OD Flow Distribution

Area OD estimation is a long-lasting challenge for traffic sensing. The previous methods mainly rely on vehicle GPS data captured by the onboard device. However, the GPS tracking equipment

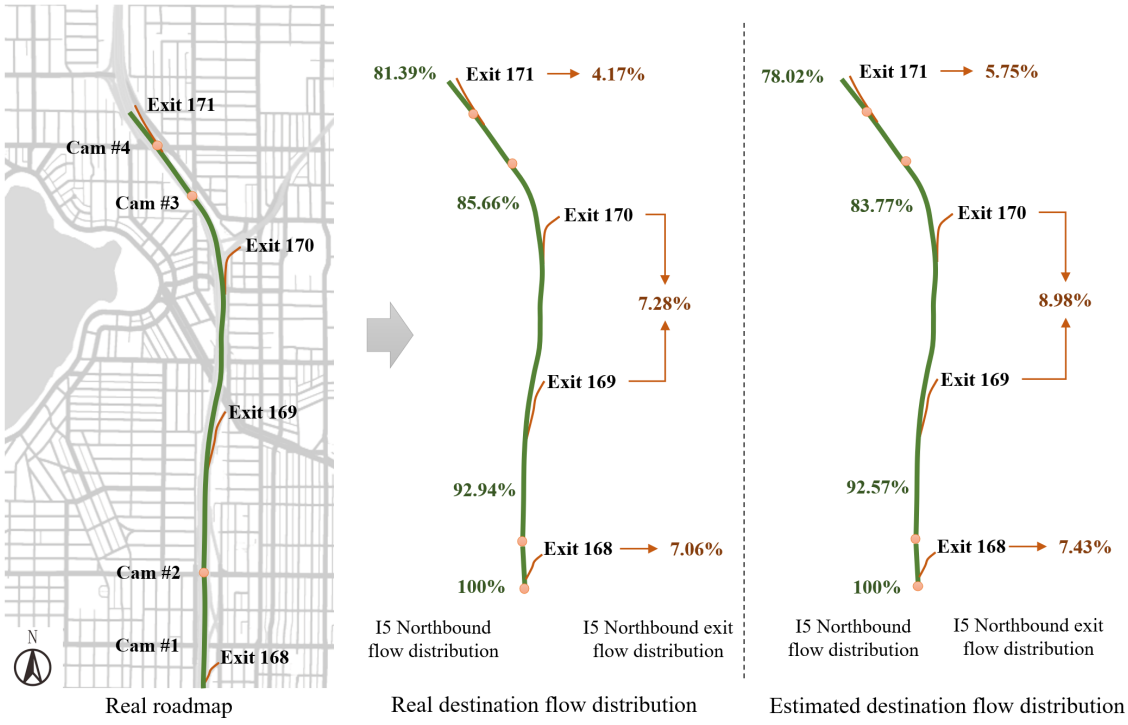

Figure 5.9: Area OD estimation and comparison by RISTS

always with the complex installation procedure. The GPS shifting in the urban area is also a serious challenge, which decreases the accuracy of network OD estimation [240]. However, with RISTS, the researchers can kill two birds with one stone. The RISTS can not only answer how many vehicles are passed on each node but also where they will go in with less than 2% distribution error. As shown in FIGURE 9., the RISTS can obtain the route trajectory data by continuous Re-ID vehicles and then summarize the traffic flow distribution. With the road graph constraint combination, the traffic engineers can obtain the area OD with high precision.

5.3.5 System Evaluation

Due to the RISTS is the first proposed IoT system for network-level traffic sensing framework by multi-camera Re-ID, we mainly compared with two other current frameworks, video-based framework and motion-trigger-based framework. For the previous framework, the basic structure is the cameras collect and transmit the original real-time video to the TMCs. Then the TMCs deal with the original video and then extract the traffic information by implementing MOD, MOT and multi-camera Re-ID. Such a framework is the current most popular structure and implemented by pioneer researchers [143]. The motion-trigger-based methods are proposed in recent years [241, 242]. By running basic motion-driven detection or tracking methods on the edge node, the TMCs can receive video clips with target objects (i,e, vehicles, pedestrians and etc.). Then post-processes the video clips by the same workflow. Such a framework can reduce the communication stream, especially in rural areas.

In this research, we conducted a comparison with video-based and motion-trigger frameworks by conducting same workflow in MOD and MOT. For the vehicle Re-ID component, due to the RISTS_Reid is customized and can not suit for the video based Re-ID framework, we using the SOTA video-based SBS Re-ID framework in the workflow. The detail system information can be found in TABLE 4.

Communication Efficiency

The communication band-with requirements are always the traffic engineers worried about when anywhere is needed to install surveillance cameras. Even with motion-trigger transmission, the volume of data is still quit high when the traffic condition is quit busy and the stream need high performance transmission system. Such video-based framework significantly limited the vehi-

Table 5.3: System information of three multi-camera traffic sensing architecture

Name	Video-SBS	Motion-SBS	RISTS
Hardware	two Titan Xp	two Titan Xp	four Jeston Xavier and one Titan Xp
Archt.	online MOD and MOT, Re-ID	online MOD, offline MOT, Re-ID	online MOD and MOT, Re-ID
Sys Power	846w	825w	4*34.2w+402w
Cameras	4	4	4
Latency	198.98s	42.74s	10.21s

cles Re-ID utility in transportation information collection. However, with the RISTS based on IoT structure, useful objects representations can be well selected by edge nodes, only small amount of data need to be send to TMCs for cross camera Re-ID. From our experiment, the average data stream is only 8.1% of video method and 24.4% of motion-triggered IoT system. The RISTS makes traffic data in the rural and low communication band-width area also can enjoy better transportation services by SOTA IoT technologies.

Power and Computational Consumption

With the help of edge and center IoT structure, the power consummation, computational latency can be much more saved based on the experiment. Here, we evaluated the computational latency from the video clip input to obtain the traffic information estimation results, and then average the latency on each vehicle. The team found that the traditional video-based method are with the highest latency due to the server is in charge of multiple video workflow simultaneously. Even two Titan Xp is installed, such a framework is the highest latency with the maximum power consummation. Due to selecting the video clips with vehicles, the motion-SBS can significantly reduce the computational time, but the power consummation is still close to the video-based method. The RISTS performs only 5% of the latency, 63% of the power consummation, with

the highest Re-ID accuracy.

Scalability

When the traffic managers consider actual deployment, scalability becomes one of the most significant factors. For the video based system, the department not only need to pay for the hardware system, but also the fee of database, communication services, power supply, and periodical maintenance. Here, the research team compared the RISTS with traditional frameworks in TABLE 5. From the comparison, it is easy to find that RISTS has multiple advantages. For the original video processing framework, one server can only deal with two video streams. A huge amount of cost needs to be spent on GPUs and data storage. The motion-trigger method can reduce half of the communication and data volume, bu the cost is higher when nodes are limited. Compared with the previous two frameworks, RISTS has four advantages. 1) Due to the hybrid Iot architecture, RISTS can save much more money on the server hardware, including both GPUs (only 30%) and data storage (5%). 2) Easy to scale up. If the manager installs more nodes, much more money will be saved. 3) The communication requirement is easy to reach, and flexible. Only daily bandwidth can fit into the system. 4) The power consummation is much more user-friendly. The total cost will be only 35% or less if twenty nodes are included in the RISTS. Considering the number of cameras installed in the current surveillance system for traffic monitoring, the RISTS is a solution with high reliability, low price, and easy management.

5.4 Conclusions and Future Work

In this research, the authors proposed a hybrid IoT system – RISTS for extracting the comprehensive network-level traffic information by edge artificial intelligence and multi-camera Re-identifiction. As whole new traffic sensing architecture, RISTS provides network-level vehicle Re-identifiction and traffic information estimation without using the steaming video. By maximizing cooperation of the edges and Traffic Management Centers (TMCs) computational resources, orchestrating data transmission and integrating road network graph features, RISTS can precisely model the network-scale traffic information in a flexible, cost-effective and easy-scalability IoT workflow.

Future research of RISTS is planned as follows. For the edge artificial intelligence, especially the multi-camera vision information extraction, the team will integrate and integrate various smart edge nodes with various types of sensors, such as UAV cameras and on-board cameras, as well as thermal cameras to build up a more inclusive Re-ID and tracking algorithms. Also, exploring the graph representation features extraction model and fuse the spatial-temporal information with graph information, and then automatically generate the camera loop based on the algorithms will be another necessary point very necessary.

6

Chapter 6. Cooperative and Comprehensive Multi-task Surveillance Sensing and Interaction System Empowered by Edge Artificial Intelligence

This chapter is modified from the following published works:

- C. Liu, H. Yang, R. Ke, W. Sun, J. Wang and Y. Wang*. "Cooperative and Comprehensive Multi-task Surveillance Sensing and Interaction System Empowered by Edge Artificial Intelligence." Best Paper Award of TRB AED30 Committee, Lectern Session, Proceedings of the 102nd Annual Meeting of Transportation Research Board, Washington D.C. USA,

Jan. 2023.

- C. Liu, H. Yang, R. Ke, W. Sun, J. Wang and Y. Wang*, 2023. "Cooperative and comprehensive multi-task surveillance sensing and interaction system empowered by edge artificial intelligence." Transportation research record, 2677(9), pp.652-668. [35]

- C. Liu, H. Yang, Z. Cui, R. Ke, and Y. Wang*. "Cooperative and Comprehensive Multitask Surveillance Sensing and Interaction System Empowered by Edge Artificial Intelligence" in Transportation Research Record, 2023.

6.1 Challenges and Motivations

Modern society faces severe challenges in the transportation systems, including, but not limited to, traffic congestion, injuries, and lack of perception fatalities. To overcome these challenges, taking advantage of the development of sensing technologies in the past decade, a variety of sensors have been introduced to the transportation community and used in the development and evolution of ITS. Combinations of advanced sensors, data process algorithms, and communication systems are applied in different transportation applications. Based on the National ITS Reference Architecture published by the United States Department of Transportation (USDOT) [243], the target clients of ITS services can be summarized into four categories: vehicles, travelers, infrastructures, and control centers. Therefore, in the past decade, thousands of researchers studied and exploited new combinations of sensing systems (i.e., the combination of sensors, algorithms, and communication systems) for various kinds of clients and applications. A report [244] published by USDOT in 2018 reviewed multiple ITS case studies and proved the effects of ITS on advancing traffic safety, mobility, and environmental sustainability.

While these technologies have the potential to revolutionize the way in which transportation systems operate, they also raise new questions and concerns. [14]. The studies related to ITS prioritize technological advancements over the practical needs and demands of users. This has resulted in a situation where many studies focus more on the development of cutting-edge technologies oriented systems rather than on addressing the real-world problems faced by road users or managers. Focusing too heavily on technologies while ignoring users' needs can result in bloated and redundant sensing systems in ITS. As we know, transportation is a complex system that involves various components, including travelers, vehicles, infrastructure, environment, and control centers. As a result, the demands for similar sensing tasks may vary among different transportation system users, leading to a diversity of solutions. Set speed measurement as an example, while transportation agencies tend to use loop detectors to gather speed information for traffic flow management purposes [48, 154], individuals such as drivers prefer to use advanced onboard sensors, such as radars, to capture the speed of their and surrounding vehicles for the purpose of travel safety and efficiency. This disparity in preferred sensors highlights the varying demands and needs among different components of the transportation system and leads to the sensor and data explosion in the field [36]. The vast amount of multi-source data generated from different sensors is overwhelming data centers and hindering the full potential of ITS. Estimates suggest that all data centers globally can only handle approximately 20 ZB, while approximately 850 ZB are generated daily in 2021 [159]. Furthermore, the design of most sensors to provide sensing services exclusively to their target users often leads to an uncooperative sensing system where information is not shared among different components. This lack of collaboration results in the generation of a large amount of redundant data, which then leads to increased costs and decreased efficiency due to repetitive processing. In addition to efficiency

issues, the limitations of uncooperative systems also include potential risks from blind spots and the inability to achieve systematic optimization due to a lack of global information. In conclusion, ignoring user needs and only focusing on technologies in the design of the sensing system leads to a decrease in safety, as well as an increase in redundancies and costs.

To address the above challenges, the research team proposed the idea of "Sensing as a Service (Saas)," aiming to develop a user-demand-oriented sensing system. The team collaborated with the Washington State Department of Transportation (WSDOT) and the City of Bellevue to set up two real-world testbeds to evaluate the system's performance. Based on the actual needs of the road users and transportation agencies, the team developed a transportation application-specified Cooperative and Comprehensive Smart Edge Node for Sensing and OpeRation (COCO SENSOR) system. The architecture of the COCO SENSOR system, shown in Figure 1, consists of four levels. The first level is the application level, which determines the needs of the different users in the application scenarios. The team focused on two well-defined transportation applications: traffic status for transportation agencies and real-time safety warnings for road users. The second level, sensing technologies, is designed to fulfill the needs defined in the first level and utilizes parallel computing to perform multiple sensing tasks efficiently. The third level is responsible for identifying the necessary data to support the sensing technologies in level two, while the fourth level involves the sensors used in the COCO SENSOR system to provide data inputs to the third level. The four levels are connected through communication systems, both local and global.

COCO SENSOR can address the above challenges in the following three perspectives. **Firstly, the COCO SENSOR introduces an edge computing mechanism in the system to process the raw data close to where the data is generated.** With the assistance of edge computing,

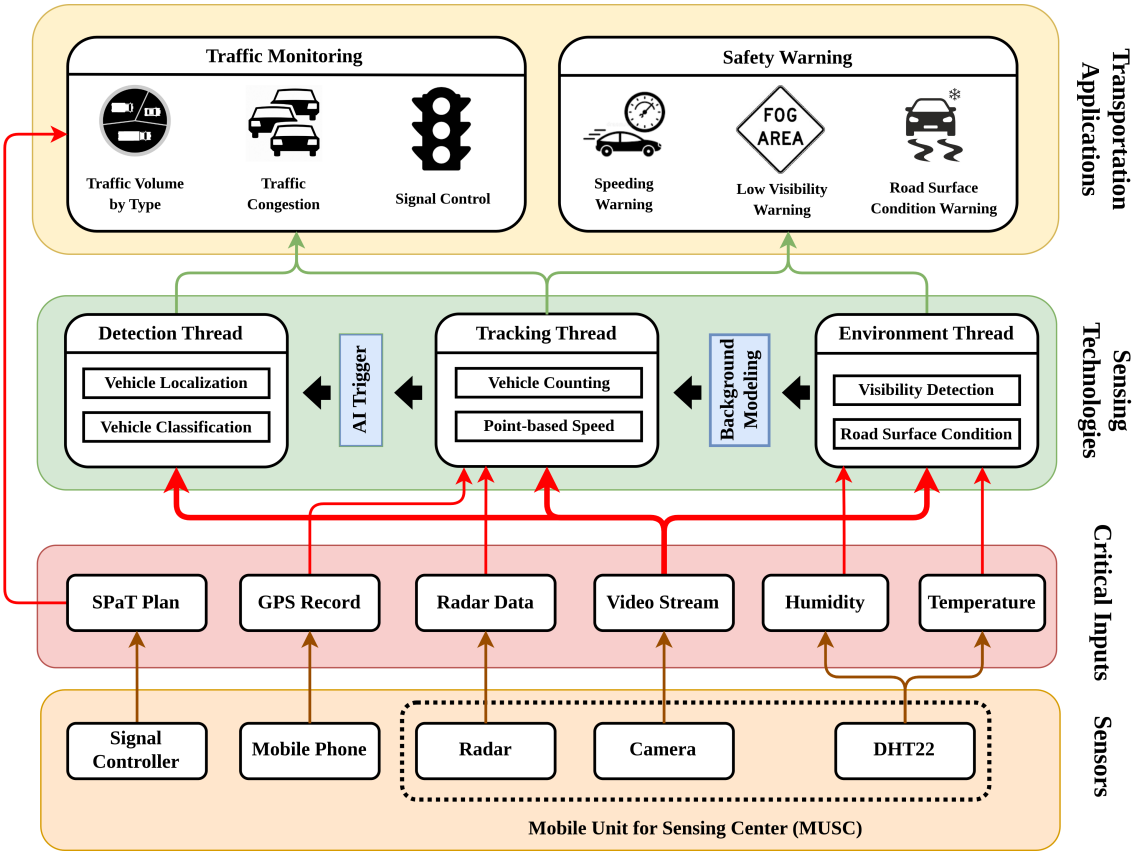

Figure 6.1: COCO SENSOR System Architecture Illustration.

most of the data can be processed locally, and only sensing results are transmitted back to the server. Therefore, integrating edge computing with customized algorithms can address data explosion challenges by significantly reducing the computation loads of central processing units. **Secondly, intra-unit cooperative sensing is the approach proposed in COCO SENSOR to connect all the sensors in the system and share the sensing results with all the demanded users.** The paper introduces multiple cooperative sensing models for various applications. **Finally, achieving "Sensing as a Service (SaaS)" is always the key target throughout the customization of COCO SENSOR.** Traditional sensing system focuses more on sensors and sensing technologies. However, COCO SENSOR treats sensing as a tool to serve the users in ITS. The first level in COCO SENSOR is transportation application, which determines the following three levels: sensing technologies, sensing data, and sensor selection. Therefore, COCO SENSOR is customized and developed, especially for the user demands in real application scenarios. The contributions of the paper can be summarized in the following four parts:

- **Idea:** The first and main contribution of the paper is the idea of "Sensing as a Service (SaaS)." The paper implements the idea and develops an application-oriented Cooperative and Comprehensive Smart Edge Node for Sensing and OpeRation (COCO SENSOR) system to address practical transportation applications.

- **Sensing Technologies:** COCO SENSOR system introduces the cooperative sensing mechanism and coordinates the computation resources on the edge device for multi-task sensing including visibility detection (92% accuracy), road surface condition detection (91% accuracy), lane-based volume counting (97% accuracy), average speed detection (90%), and object detection (95% accuracy).

- **Parallel Computing:** Due to the limited computation resources on the edge device, COCO SENSOR system introduced three independent threads to coordinate the computation loads for each thread.

- **System Implementation:** Cooperating with WSDOT and the City of Bellevue, the team set up testbeds COCO SENSOR system implementation. The team targets four critical applications in the testbed: traffic volume by vehicle type, traffic status detection, low visibility warning, and road surface condition warning. Additionally, a mobile APP is developed for both traffic managers and users to obtain comprehensive traffic information and live warning messages anytime, anywhere.

The architecture of the sensing technologies is shown in Figure 6.2. Unlike the other architectures described in Section 2 that balance the computation loads between edge and cloud servers, COCO SENSOR system is designed to operate fully on the edge device. To make efficient use of the limited resources on the edge devices, including CPU, memory, and disk space, the system includes three parallel and independent threads for multiple sensing tasks: 1) Environment Thread; 2) Vehicle Tracking Thread; 3) Object Detection Thread. The reasons for designing three threads instead of one are summarized in three points. These threads allow for efficient resource allocation based on the varying computation demands of different sensing algorithms and enhance the system's robustness in real-world applications. The specifics of each thread are discussed in the following section.

6.1.1 Environment Thread

Environment Thread is indicated with a yellow background in Figure 6.2. Firstly, the dehaze algorithm is applied to the input video stream from camera sensors to estimate and remove the

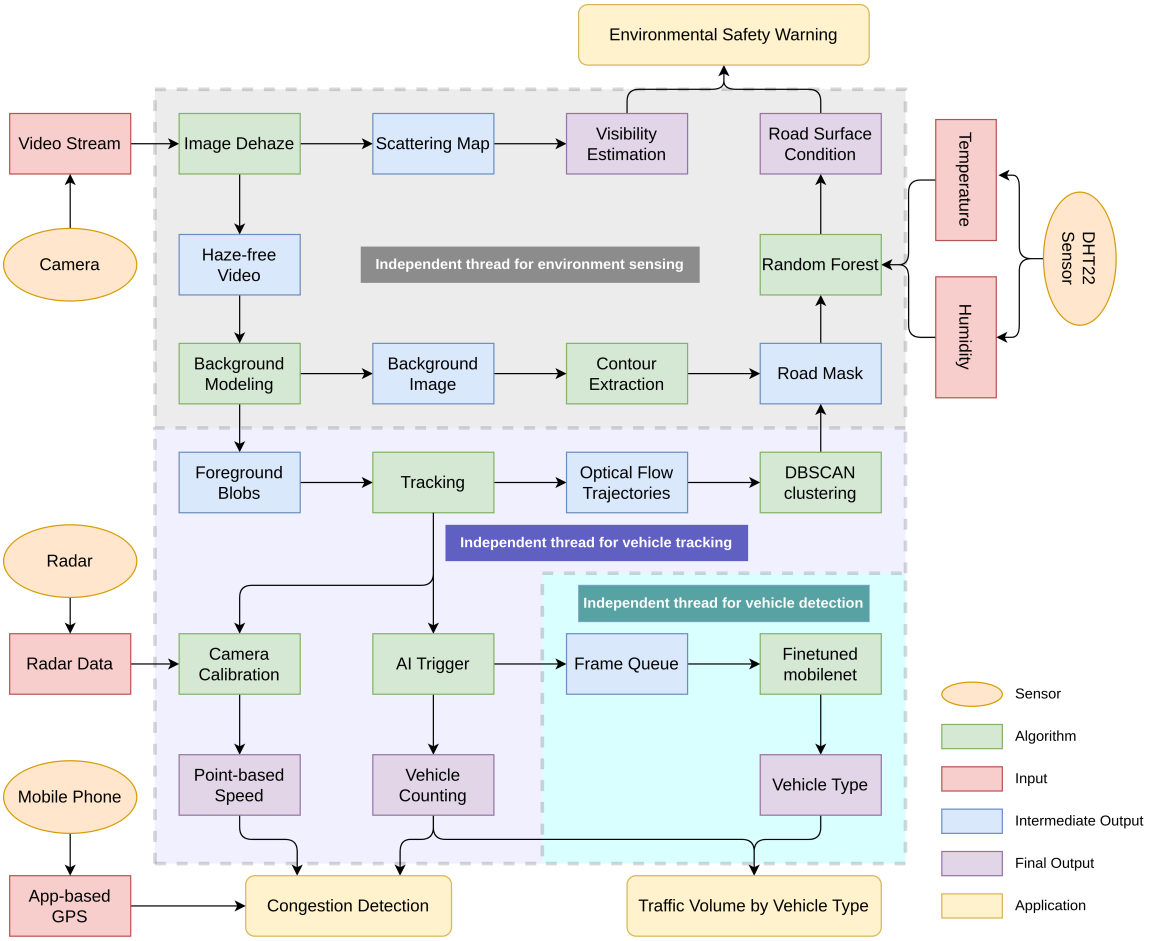

Figure 6.2: Sensing Technologies Architecture

haze for visibility estimation and haze-free video, respectively. Then the background modeling is implemented in the haze-free video to generate background and foreground images. The contour extraction model and tracking algorithms are applied to background and foreground images respectively for road mask extraction. Finally, four features, including two image features (i.e., intensity and black channel) are extracted from the road mask and two environment features (i.e., temperature and humidity) from DHT22 sensors are taken by a random forest model for the road condition classification.

Image Dehaze and Visibility Detection

The hazed image captured by the camera $I(x)$ consists of two primary light sources: air light and light reflected by the surrounding objects. In the paper, we assume air light is a homogeneous parallel white light in the scene, which is represented by A. The actual scene of the object is represented by $J(x)$. For the object captured in the image, the portion of the object reflected light that can reach the camera could be represented as $J(x)t(x)$. Because of the scattering effects, some portions of air light can be scattered to the region where the target object is located. And this portion of air light can be represented as $A(1-t(x))$. Therefore, the object image captured by the camera can be represented as the sum of $J(x)t(x)$ and $A(1-t(x))$. And the target of haze removal is to estimate the scattering effects $t(x)$.

Dark channel is a concept proposed by [186] in 2011, indicating the smallest value in the three channels of Red, Green, and Blue. The Dark Channel Prior (DCP) is based on the broad observation of outdoor haze-free images. In most of the haze-free patches, at least one color channel has some pixels whose intensity values are shallow and even close to zero. Therefore, based on the

Therefore, if we apply the dark channel operation to Eq. (1), we can get the following:

$$D(\frac{I(x)}{A}) = t(x)D(\frac{J(x)}{A}) + (1 - t(x)) \tag{6.1}$$

Based on DCP, the dark channel value of the real scene $J^c(y)$ should be close to 0. Therefore, we can get the following:

$$J^{dark}(x) = D(\frac{J(x)}{A}) = 0 \tag{6.2}$$

Therefore, based on Eq. (2) (3) (4), we can get the transmission map $t(x)$, which indicates the scattering effects in the scene. Based on the transmission map, we can realize image dehaze and produce visibility estimation to warn the road users.

$$t(x) = 1 - D(\frac{I^c(x)}{A}) = 0 \tag{6.3}$$

Road Mask Extraction

The team proposed a method to improve road mask generation in the Environment Thread by integrating the results of contour detection and motion detection. The method takes advantage of the strengths of both techniques to overcome their limitations. Vehicle motion detection tracks moving objects to extract the region of interest in the scene, however, struggles in regions with low traffic volume. Road contour detection quickly detects road segments, but accuracy is impacted by environmental factors such as lighting. The proposed integration enhances the system's adaptability and accuracy. A detailed explanation of the process can be seen in Figure 6.3 and the following paragraphs.

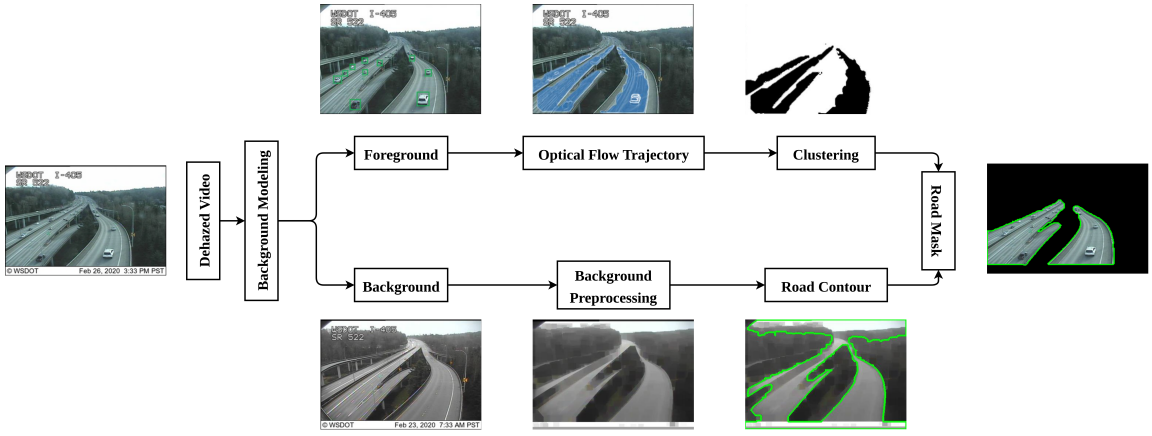

Figure 6.3: Road Mask Generation by Integrating Contour Detection and Optical Trajectory Flow

Object contour detection is a research topic with a long history. The paper employs Canny edge detection algorithm [196, 197] to estimate all the contours in the processed background image. Compared with standard shape objects in the pure background, the real-world objects in the complicated backgrounds bring more challenges to contour detection. Therefore, the paper applies both erosion and dilation operations to the image to smooth the edges of the image. Then, the Canny algorithm is implemented to generate the first derivative of the Gaussian function, which is the best approximation of the optimal edge detection operation. Finally, the calculated magnitude and direction image gradients can realize road contour detection.

The second method used for road segmentation is vehicle motion detection. In the method, firstly, a lower bound threshold is set up as the minimum moving distance between two consecutive frames to filter the optical flow. Therefore, the moving vehicles can be extracted from the static background. Then the pixels which have been marked as the optical flow can be represented by feature vector $[x, y, d, v]$, where (x, y) indicates the location of the vehicle in the pixel coordination, d represents the moving direction, and v indicates the travel speed of the vehicle. The marked pixels accumulate as time goes on. After time T, the marked pixels will cover the

major areas where the traffic driving through. For the road segment with high traffic volume, T can be just a few minutes, while for the rural roadways with low density, a large T is required to extract all the target regions.

To integrate the road mask generated by contour detection and optical trajectory flow, we introduce the cosine similarity method to quantify the distance between contour points and the edge points of the accumulated optical flow. The method can eliminate the outliers contour points can produce an accurate road mask.

Road Surface Condition Detection

We have selected four features, two from the image data and two from the environment, for road surface condition classification. The two image features are the intensity value and dark channel value. Based on DCP theory, for most objects in nature, the dark channel value is pretty small, otherwise, the object will be in white. This is also the case for most road surfaces. However, for rainy or snowy conditions where there are a lot of reflections or white pixels, the dark channel can be relatively higher. The intensity value, on the other hand, can differentiate between snowy and non-snowy conditions, as the intensity values are high in snowy conditions. Figure 6.4 and Figure 6.5 show examples of intensity and dark channel histograms of road regions in Washington State, respectively, as captured by a surveillance camera. The difference between the intensity and dark channel features of roads with snow and without snow is still significant, which can be used for classification in our method. The addition of temperature and humidity sensors will further increase the accuracy and reliability of the classification.

In addition to the image features, two environmental features, temperature and humidity, collected from the DHT22 sensor, are also considered. Temperature is an indicator of snowy

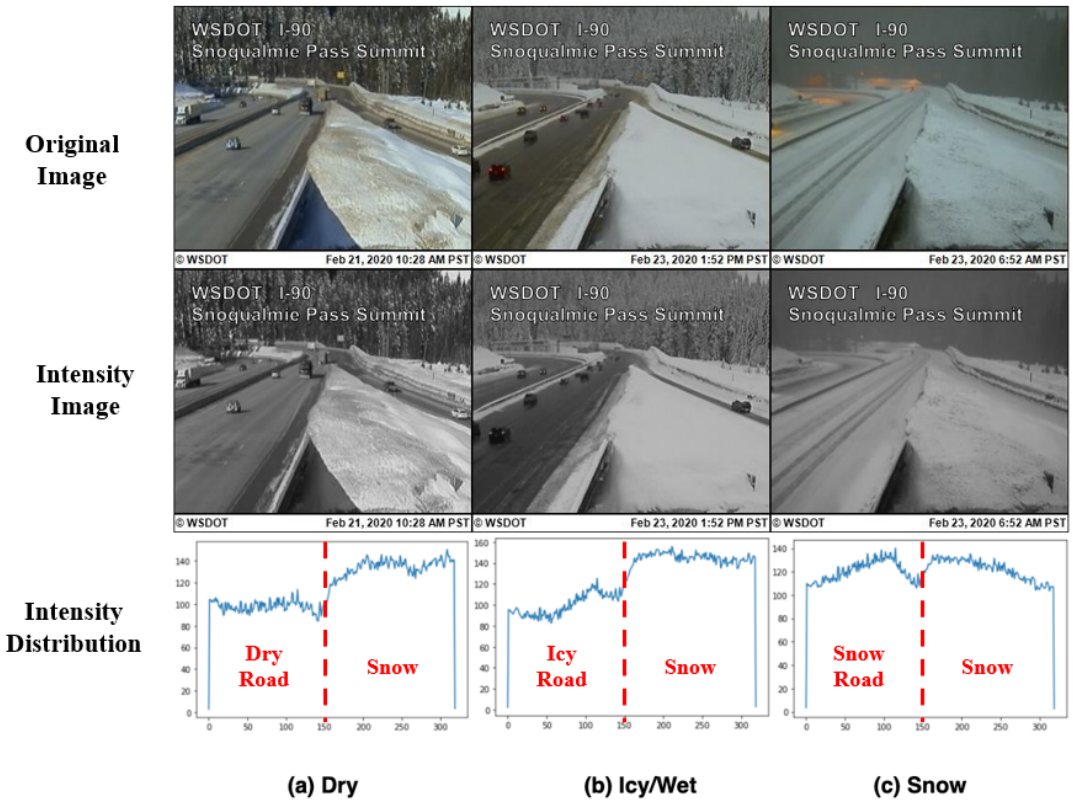

Figure 6.4: Image Intensity Value Distribution in Different Road Surface Conditions

conditions and humidity is an indicator of rainy conditions. The selected features are simple, representative, and form a feature space suitable for successful classification. The feature vector is expressed as $[I, K, T, H]$, where I is the median intensity, K is the median dark channel value, T is the temperature, and H is the humidity.

Based on the four features motioned before, COCO SENSOR can classify the road surface conditions into four categories:

- **Dry:** Diffuse reflection occurs on the road surface. In this case, the gray value should be steady and the dark channel value should be close to 0.

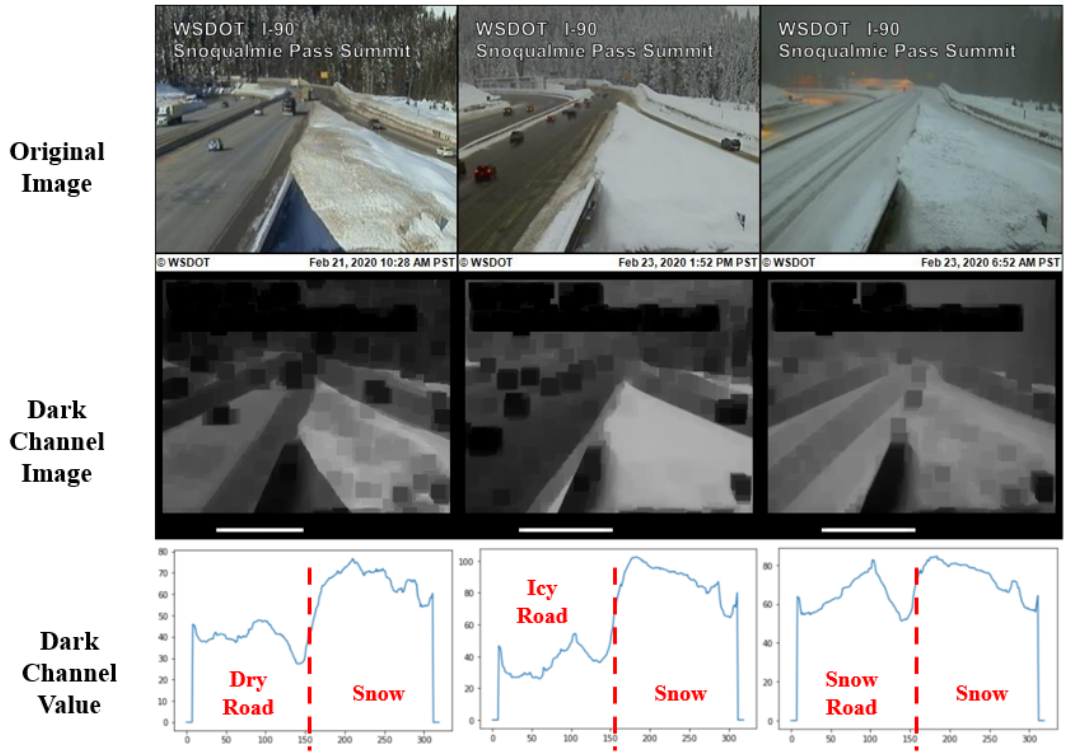

Figure 6.5: Dark Channel Value Distribution in Different Road Surface Conditions

- **Wet:** Specular reflection is the main approach to light transmission on the road surface. Impacted by surrounding light sources, the brightness varies a lot on the road surface, which results in a large variance of gray values. The increased portion of specular reflection results in the increase of dark channel value.

- **Icy:** Icy condition has similar intensity and dark channel values to the wet condition because they both result from specular light reflection on the road surface. However, in this case, environmental features like temperature can help the system judge if the road surface is covered by ice or water.

- **Snow:** Diffuse reflection occurs on the snow-covered road surface. Compared with dry conditions, the reflection rate of snow is much higher. As a result, both image intensity values and dark channel values are higher.

6.1.2 Tracking Thread

The Tracking Thread is indicated with a red background in Figure 6.2. In COCO SENSOR, Tracking Thread is the main thread that connects the other two threads and produces multiple sensing results to support the transportation applications. In the thread, SORT tracking is applied to the foreground images extracted by background modeling from traffic video. The tracking results are used for three sensing tasks: 1) optical flow trajectory extraction for road mask; 2) AI trigger for object detection and vehicle counting; 3) cooperative camera calibration for speed measurement. The optical flow trajectory extraction is introduced in Section 3.2.1 for road mask generation. Therefore, the section only introduces the other two sensing tasks in Tracking Thread.

AI Trigger

COCO SENSOR system is an edge-based system, whose performance highly depends on the limited computation resources of edge devices. In Tracking Thread, AI trigger is designed to filter the frames containing the interested objects for further sensing tasks. In the system, AI trigger controls the inputs of multiple algorithms including volume counting and object detection. AI trigger can reduce estimated 50% computation loads for the system.

First of all, we need to define a region as the region of interest, which triggers further processing using the AI classifier when there are moving objects. This can be done by either manually labeling one area for each traffic direction or using the extracted road mask in the previous step as the region of interest. Then, the background modeling and SORT tracking run in real-time in the main thread to process every video frame. In case there is any object detected and tracked in the region of interest, it will activate the procedure that extracts and stores the intermediate information into a global variable queue Q.

Each element in the Q has three major elements: the current time T, the current frame F, and a list L that contains objects' information in the current frame. Note that the current time T is needed as a time label because the frame may be processed later, depending on how many elements are left in Q. The list can be any length depending on the number of moving objects in the region of interest. There are three sub-components for each object in L: the bounding box *bb*, the tracking id *tid*, and the direction *id* of the road area. Note that the *bb* and *tid* are obtained from the background modeling and SORT tracking.

For traffic volume detection and classification, it constantly checks if the queue Q is empty. If Q is not empty, it will pop the first element from Q, then it will run the Mobilenet V2 classifier for road user classification and use other information we store in Q for traffic volume counting

for each type of road user and each traffic direction. Note that the moving object may not be any road users we are interested in, e.g., an animal, or just some false detections due to sudden light change of camera vibration. The AI classifier can filter out those detections in addition to classifying road users into different types.

Radar Cooperative Camera Calibration

Camera calibration is aimed at converting the 2D image coordinate to a 3D real-world coordinate, which is critical for video-based speed measurement. In recent years, many solutions strategies are proposed in calibration equations, however, most methods require the placement of calibration points or calibration objects in the field of view. In transportation scenarios, it does bring difficulties in placing the calibration objects or targets in the roadway. Additionally, re-calibration operations are required every time the camera internal parameters like focal length and angle demands are adjusted. Therefore, the system introduces the radar sensor to provide the ground truth spatial information to calibrate the camera system.

Radar is the common sensor for high accuracy and efficiency speed sensing. However, in transportation scenarios, especially in the freeway system, there are multiple lanes in each direction. Therefore, limited by the beam width, a single radar sensor cannot cover all the lanes in the road to realize lane-based speed measurement. Therefore, the system proposes an innovative fully-automatic radar cooperative camera calibration method for lane-scale speed measurements.

6.1.3 Object Detection Thread

The Object Detection Thread is indicated with a blue background in Figure 6.2. The AI trigger in Tracking Thread is designed to select, store, and process just a small portion of the original

video frames, which are the frames of interest. Specifically, when moving objects are defined and tracked in predefined regions on the road, the frame will be stored in a queue waiting to be processed. As a result, even in some limited IOT devices like Raspberry Pi, the independent threads can ensure the system works in a consistent way.

In the paper, to generalize the system to limited IOT devices, the research team uses Mobilenet V2 [245], a light convolutional neural network architecture specified for mobile devices, to complete object detection tasks. And the one-stage model Single Shot Detection (SSD) [119] is applied to the architecture. The model MobileNet V2 SSD is not the most advanced model in the object detection task, however, it has the best performance on accuracy and efficiency balancing. The research team pre-trained the model with COCO dataset[246] and finetuned the model with MIO-TCD traffic surveillance dataset [220] for eleven specific classes of transportation agent detection. The model is light enough to be deployed on limited IoT devices without GPUs.

6.2 Communication System

The Figure 6.6 shows the communication synchronization among COCO Sensor, signal controller and the user person identifier devices (PIDs) through WiFi or cellular network. All the communication algorithms can be divided into two parts: interaction with the controller and interaction with user PIDs. The communication workflow of COCO Sensor with the signal controller (sending the pedestrian information and obtaining the waiting time for each phase) can be finished by cable connection or local wireless network. Based on the standard of in NTCIP, the communication script can capture the real-time signal phase and timing information by the API and then store it in the buffer. Then, the signal and the sensing information can

be broadcast to the user PIDs (cell phones, wearable devices) with the COCO cooperative user application. The COCO sensor is also integrated with an independent thread for licensing the request (i.e., crossing request) or messages (accident information) generated by roadway users. The detailed communication scheme is illustrated in Figure 6.6.

6.3 Experiment & Sensing Results Evaluation

6.3.1 System Configuration and Settings

In the research, COCO SENSOR system is implemented on Raspberry Pi 4 for multi-task sensing. Raspberry Pi 4 is a limited edge device without the support of advanced hardware like GPU for video processing. However, its low price and ease of operation make it particularly suitable for large-scale deployments in transportation systems. Additionally, if COCO SENSOR can realize real-time video processing on Raspberry Pi 4, it is easy to be transferred to other edge devices like Nvidia Jetson series. Therefore, in the test, we employed Raspberry Pi 4 as the central unit to connect all the other components in the traffic scene. The other hardware components used in the test are customized based on the applications in the scenario, including the environment sensor (i.e., DHT22), PTZ camera, radar sensor, communication kit, and protective shell. To test the performance of the system, the team prepared some devices including a test vehicle, a cell phone with COCO SENSOR application installed, a portable radar gun, and a laptop.

The research team cooperates with Washington State Department of Transportation (WSDOT), City of Bellevue, and Pacific Northwest Transportation Consortium (PacTrans) to build the testbed for COCO SENSOR system experiment and test. We installed MUSCs in multiple transportation scenarios including mountain roads, local streets, freeways, and intersections for various application tests. Figure 6.7 shows the deployment of COCO SENSOR system in the

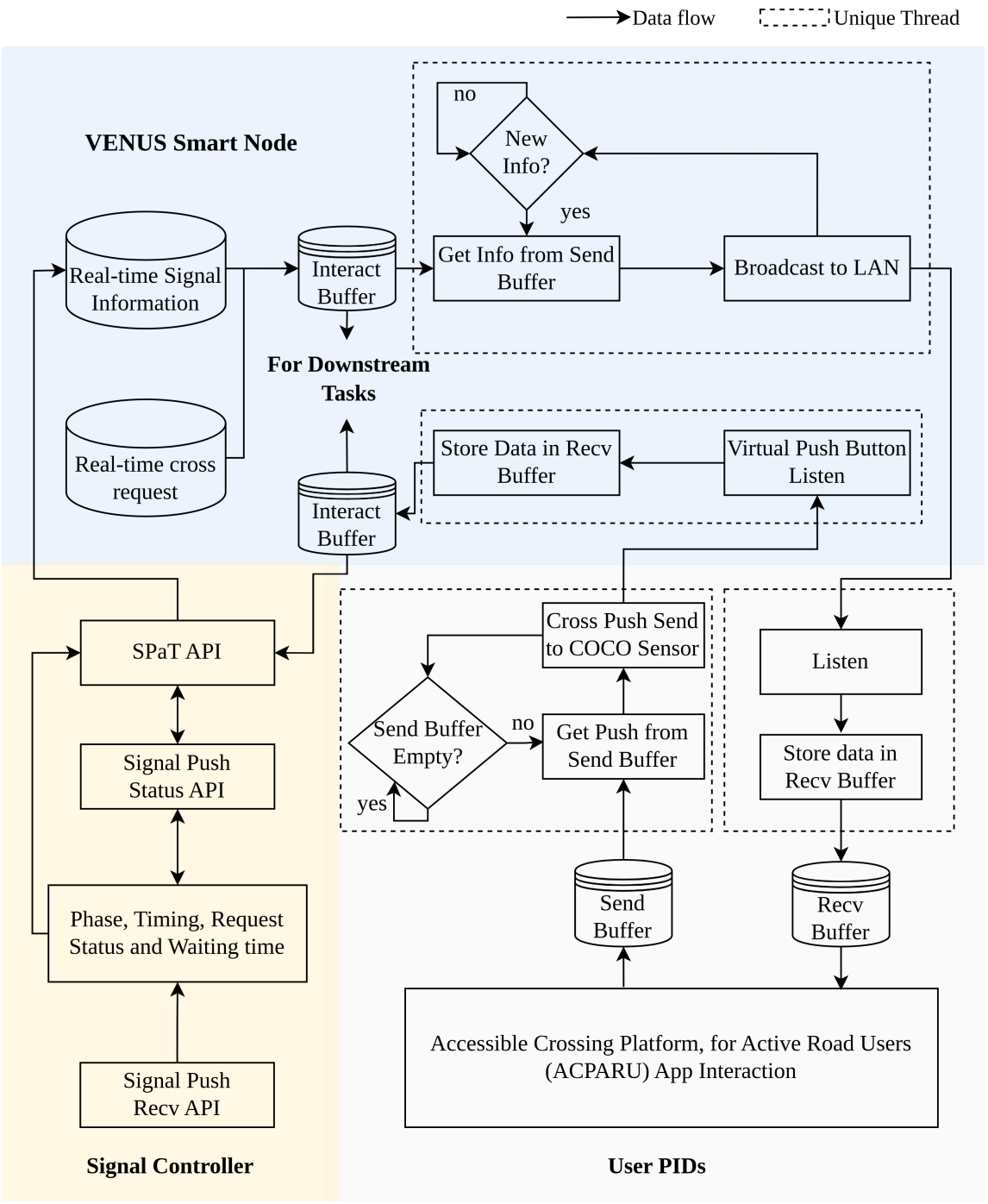

Figure 6.6: The communication workflow across user PIDs, signal controllers and the COCO Sensor. The signal phase and timing information, user request, and other messages can be disseminated via COCO SENSOR to users and the roadside control unit.

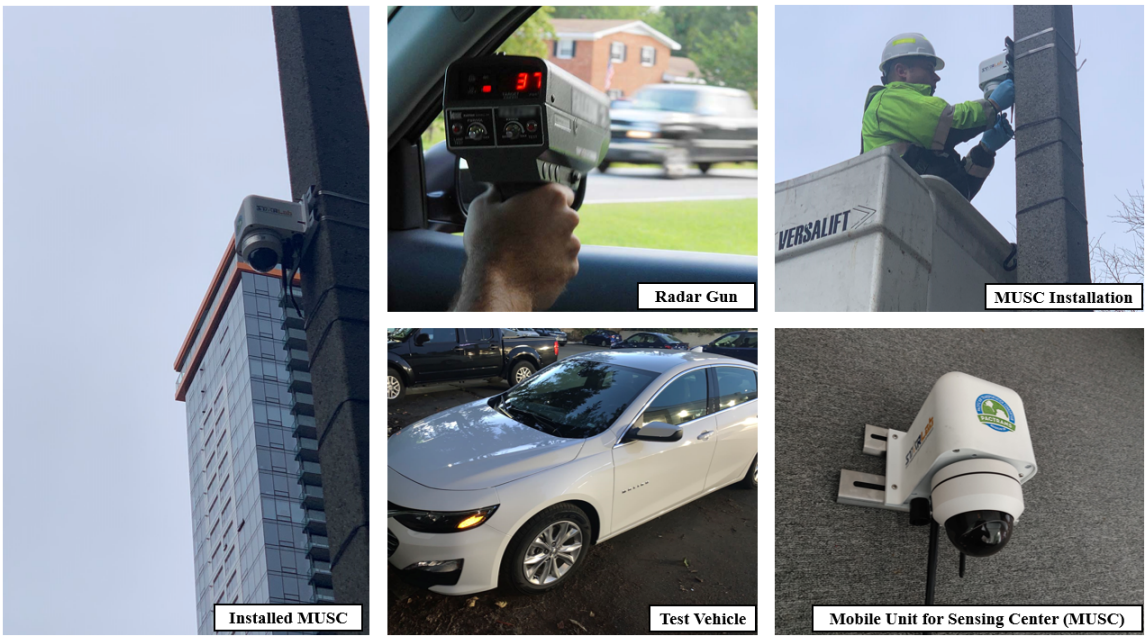

Figure 6.7: COCO SENSOR Deployment in Bellevue Testbed

testbeds. The system has been deployed and tested in the testbed since 2021. And all the data used in the paper to validate the performance of the system is collected during the test period.

6.3.2 Environment Sensing

To validate the performance of environment sensing, the team introduced the public weather data collected by WSDOT weather stations near the testbeds as the ground truth data. The ground truth data used in the paper includes real-time weather conditions and visibility conditions.

The results of visibility detection are shown in Table 1. The overall accuracy of the visibility detection can reach 92.15% (threshold = 10%). The results in Table 1 show that the detection accuracy increases when the visibility condition turns better. In extreme conditions like thick fog or snow storms, the visibility detection accuracy drops to about 89.14%. The image dehaze

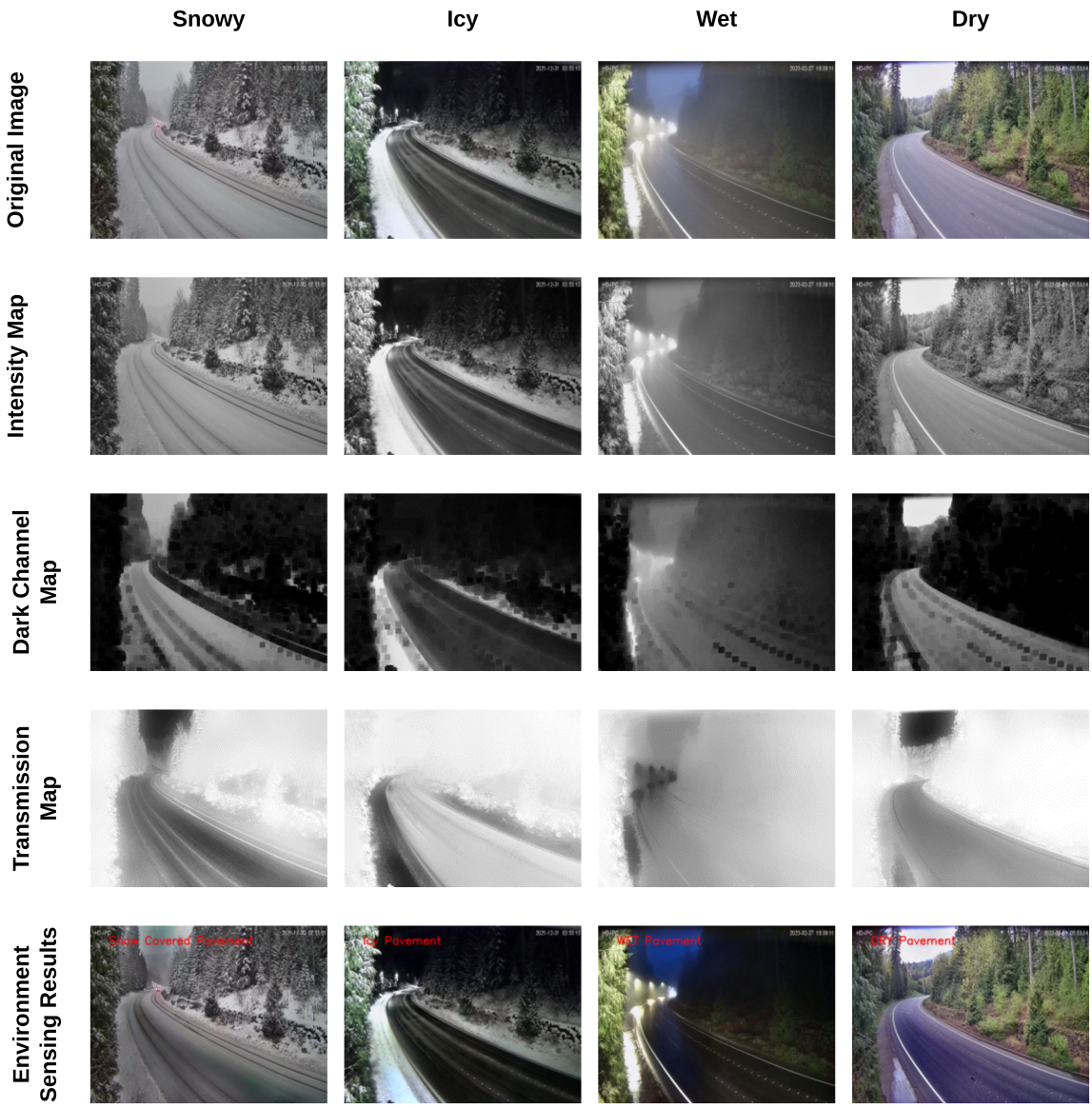

Figure 6.8: Environment Sensing Results Visualization

Threshold	±5%	±10%	±20%
$V_s < 500m$	85.29%	89.14%	93.18%
$500m \leq V_s < 1000m$	88.17%	90.25%	95.42%
$1000m \leq V_s < 2000m$	90.36%	93.22%	97.03%
$V_s \geq 2000m$	91.23%	95.78%	98.75%
Overall	89.27%	92.15%	96.61%

Table 6.1: Visibility Estimation Performance

effects are shown in Figure 6.8. The difference between the original images (i.e., the first row) and the de-hazed images (i.e., the last row) indicates the effects of the haze removal algorithm proposed in COCO SENSOR.

The results of road surface condition classification are shown in Table 2. We can see that the accuracy can reach 96%, 92%, 90%, 86% for dry, wet, snowy, and icy road surfaces. It is no doubt that the dry and wet conditions have higher accuracy because they are the two most common road conditions, which means they have more data to feed the model for training. However, snowy and icy conditions only happen for a short period during the winter season in Washington State. As a result, the lack of data can result in the drop in accuracy. Additionally, in the data annotation, it is difficult to distinguish the icy and snowy conditions in some cases, which may confuse the model in dealing with the two situations. However, in summary, the overall results (95%) are still good enough to support the piratical weather condition warning system. To visualize the process of road surface condition classification, Figure 6.8 shows the intensity map (second row), dark channel map (third row), and final results (the last row).

6.3.3 Lane-scale Volume Counting

The demo of lane-scale vehicle counting is shown in Figure 6.9. The lane information is extracted from the road mask in Tracking Thread. Cooperating with SORT algorithm, the vehicle

Surface Condition	Dry	Wet	Snowy	Icy
Dry	0.96	0.01	0.01	0.02
Wet	0.01	0.92	0.04	0.03
Snowy	0.03	0.00	0.90	0.07
Icy	0.00	0.05	0.09	0.86

Table 6.2: Road Surface Condition Classification Performance

can be tracked and counted in the lane. It is worth mentioning the situation about lane change. The lane change behavior has not impact on the total traffic volume counting, however, it is difficult to determine which lane the vehicle is in. Therefore, the paper introduces the counting zone design in the scene. The vehicles can only be counted in the lane where they pass the corresponding zone.

Because there is no public traffic volume data in the test road segment, the ground truth data is collected through manually counting. The team counted four hours of traffic volume in total covering 16 time periods in one day. The results show that the counting accuracy can reach 97% in the test road segment.

6.3.4 Vehicle Speed Measurement

Figure 6.10 shows the radar cooperated camera calibration results. Cooperated with a radar sensor, spatial information including speed and distance can work as the ground truth reference for camera calibration. The yellow dash lines in Figure 6.10 determines four parallel lines with the same distance. The space between the first line and the last line (marked in red) is selected as the speed measurement region. The speed detected in the system is calculated by the travel time for the vehicles passing the two red lines. The blue lines extracted from the road mask are used to determine lane distribution in the scene for lane-based speed measurement. To validate

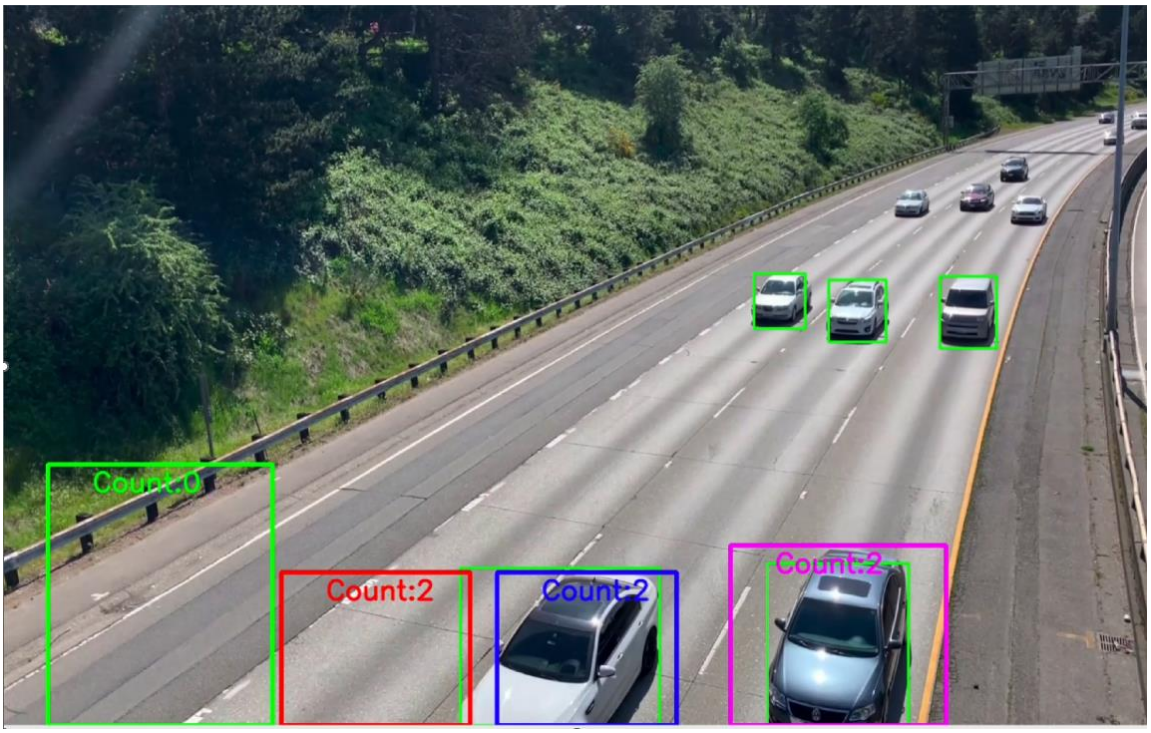

Figure 6.9: Lane-scale Vehicle Counting Demo on Freeway Scenario

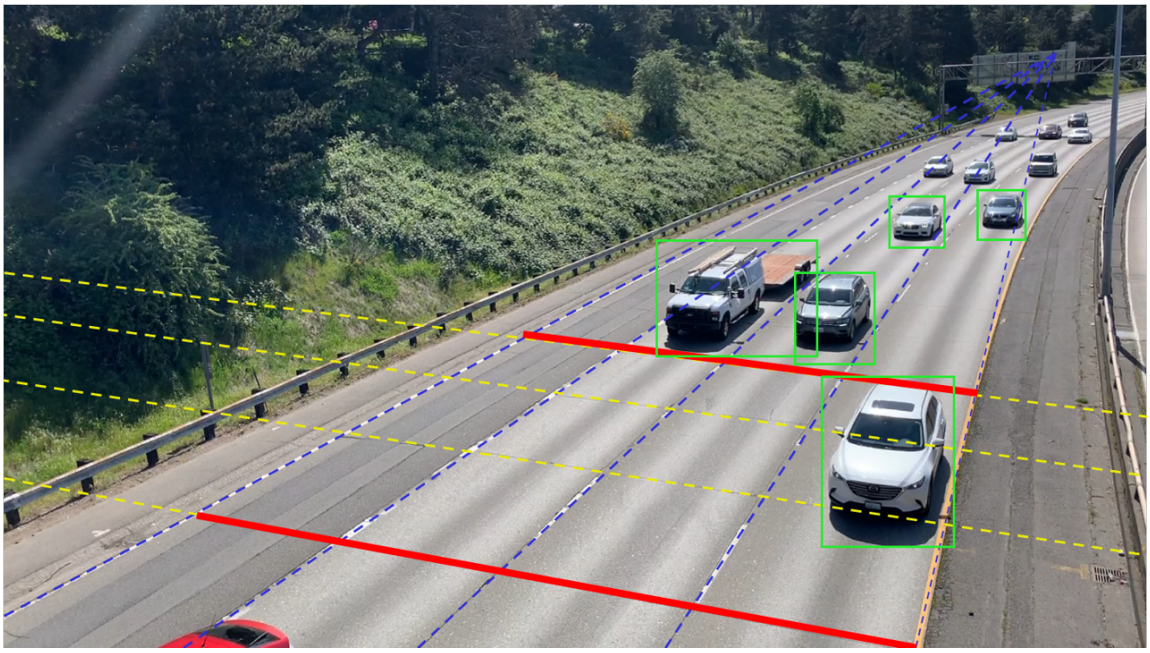

Figure 6.10: Radar Cooperated Camera Calibration for Vehicle Speed Measurement

the speed measurement results, the team employed a portable radar gun to detect the speed of passing vehicles. And the results show that the average errors in $\pm 10\%$.

6.3.5 Object Classification

The object detection model is retrained by MIO-TCD dataset which consists of ten classes of road users. Based on the demands of the applications in the testbed, COCO SENSOR combined the class labels into four road user categories: car, truck, bus, and cyclists and the background. In the new classifications: truck consists of single-unit truck and articulated truck; cyclists consists of bicyclist and motorcyclist; and car consists of car, work van, and pickup truck; bus is still the original bus. Figure 6.11 show some sample object detection results in various conditions. Table 6.3 shows the object detection results in the test period. In the test, bus and

Detection Results	Car	Truck	Bus	Cyclist
Car	91%	5%	2%	2%
Truck	6%	87%	4%	3%
Bus	1%	2%	96%	1%
Cyclist	2%	2%	1%	95%

Table 6.3: MobileNet Object Detection results on MIO-TCD dataset

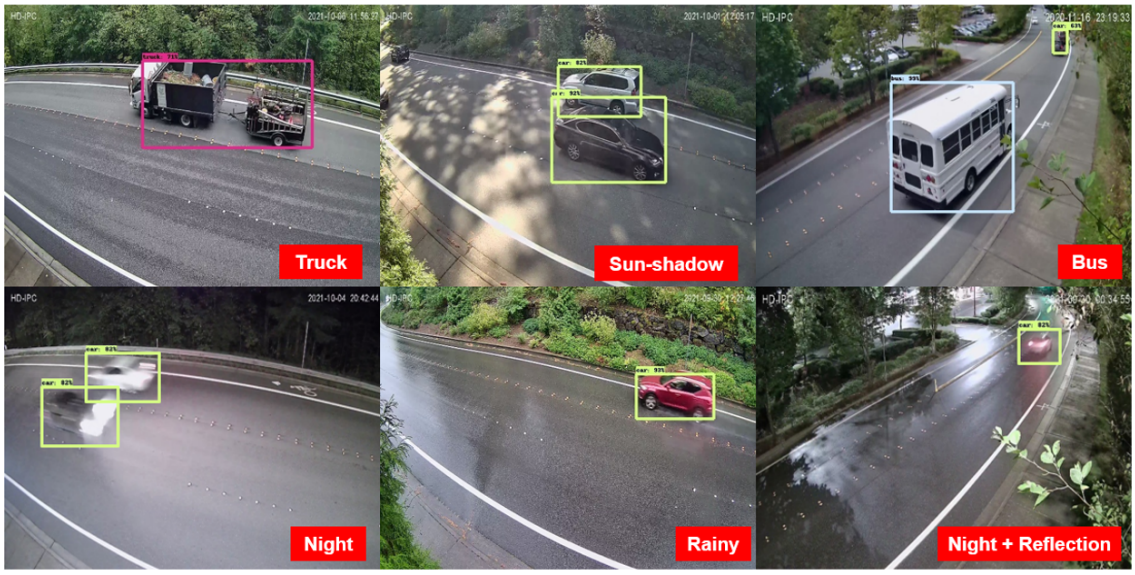

Figure 6.11: Sample Object Detection Results

cyclists have relatively higher accuracy and reach 96% and 95% respectively. However, the car and truck categories have relatively low accuracy, which may be due to the common features of the pickup truck.

6.3.6 Edge Adaption Performance Evaluation

This subsection is aimed at evaluating the performance of Edge adaption of COCO Sensor system. To improve the processing efficiency on edge devices, COCO Sensor introduces two methods, parallel computing and AI trigger, in Section 3. For parallel computing, we designed three

Table 6.4: Processing Efficiency Evaluation

Structures	Sequential Architecture	Parallel Architecture	COCO SENSOR
Processing Speed (FPS)	1.3	3.2	11.3
CPU Memory Usage (%)	34%	65%	82%
Power Consumption (W)	2.2	4.1	5.6
Efficiency (FPS/W)	0.591	0.780	2.018

independent parallel threads, Environment Thread, Tracking Thread, and Detection Thread. Based on the computation resources consumption in each independent thread, the system can adjust the processing speed and resource allocation to realize systematic optimization. Secondly, AI trigger is an innovative trigger mechanism designed by COCO SENSOR to filter the frames without the objects of interest for matrix computation loads reduction. Both methods can improve the system performance on edge devices significantly.

To evaluate the performance of the two methods on edge devices, the research team deployed the same system with different architectures. The first architecture follows the sequential logic flow, which indicates the input of the module is exactly the output of the last module. The second architecture introduces three independent threads for parallel computing without the AI trigger mechanism. Finally, the third architecture is COCO SENSOR system, parallel programming with the AI trigger mechanism for computation loads reduction and resource allocation. The three systems are tested on a popular but limited IoT device, Raspberry Pi4. The Raspberry Pi 4B has a Broadcom BCM2711 system-on-chip, and it runs on a 1.5-GHz quad-core 64-bit ARM Cortex-A72 CPU @ 1.5 GHz and no GPU. The memory size we used in the project is 4 GB. The experiment compares the processing speed of three architectures with the same input and processing model.

The comparison results of three architectures are shown in Tabel 6.4. We use four metrics,

processing speed, CUP memory usage, Power Consumption, and Efficiency (FPS/Power), to measure the performance of three structures. By comparing the first and second columns, the introduction of parallel computing can increase the processing speed from 1.3 FPS to 3.2 FPS, and the efficiency is increased from 0.591 to 0.780. And the differences between the second and third columns indicate the introduction of the AI trigger mechanism can increase the processing speed and efficiency to 11.3 and 2.018, respectively. The improvements from sequential architecture to COCO SENSOR system show the effects of parallel computing and AI trigger mechanisms.

6.4 System Implementation & Application Development

COCO SENSOR is a transportation application-oriented sensing system. Therefore, this section introduces how to apply the sensing results in Section 4 to practical application. Based on Figure 6.1, the system builds up four different kinds of applications, including traffic volume by type, traffic congestion detection, low visibility warning, and road surface condition warning. In the system, the first two applications are developed for traffic status monitoring, and the other two applications are for real-time safety warnings. As a result, the design of the two kinds of applications is totally different.

For the second kind of application, the team used visibility sensing results and road surface classification as the two inputs to generate the real-time warning message for traffic safety. In the test, if the visibility is lower than 1,000 m or the road surface condition is wet, icy, or snowy, the system will generate warning messages and broadcast them to all the clients.

To disseminate the information including traffic status reports and safety warning messages, the team developed a mobile APP for the clients. Figure 6.12 shows the design of the App.

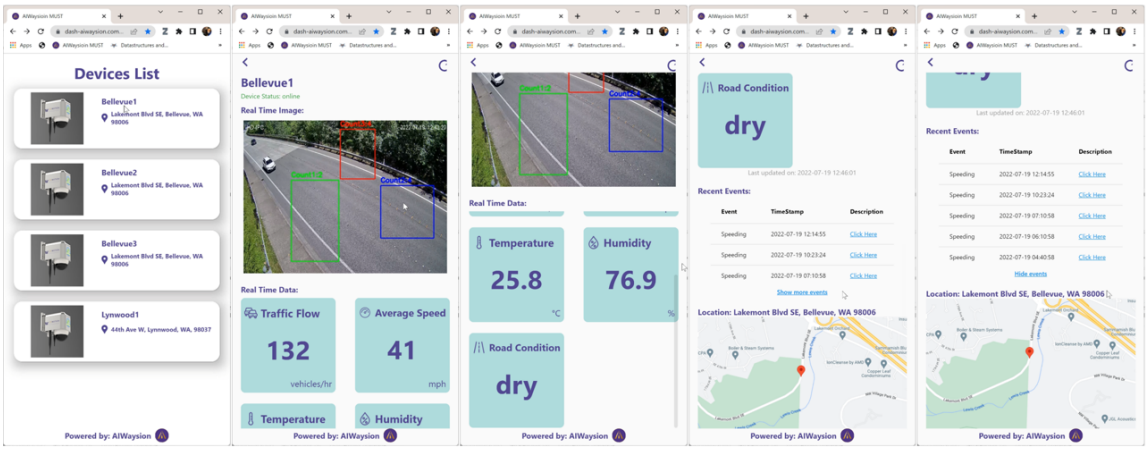

Figure 6.12: Mobile Application User Interface Visualization for the Warning System

The user can select the device they want to access on the home page. After a click, the detailed information including real-time surveillance view, traffic volume (15 min), average speed (15 min), temperature, humidity, visibility, and road surface condition are presented on the App. Users can also turn on the notification and customize the information they would like to receive. And the historical events including traffic congestion, speeding events, wet/snowy/icy surface conditions, and low visibility can be checked by the users.

6.5 Chapter Summary

The paper presents the concept of "Sensing as a Service (SaaS)" aimed at creating a user-centric sensing system. The research team partnered with the Washington State Department of Transportation (WSDOT) and the City of Bellevue to set up a real-world testbed to gather actual user and transportation agency demands. Based on these demands, the team developed the Cooperative and Comprehensive Smart Edge Node for Sensing and OpeRation (COCO SENSOR) system specifically for transportation applications.

COCO SENSOR has a four-layer structure as shown in Figure 6.1. The system includes four different transportation applications, including traffic volume counting by vehicle type, traffic status detection, road visibility estimation, and road surface conditions detection, which were proposed in the testbeds. The second layer has three independent threads designed to handle five challenging sensing tasks - visibility detection, road surface classification, lane-scale traffic volume counting, vehicle speed measurement, and object detection - on the edge device. The third layer supplies the crucial data for the sensing tasks in the second layer, and the final layer consists of the sensors used in the system, providing the inputs to the sensing algorithms. These final two layers demonstrate how the cooperative mechanism works in the COCO SENSOR system.

The contributions of the paper can be summarized in the following four aspects:

- **The "Sensing as a Service (SaaS)" concept**: The paper presents a novel idea of "Sensing as a Service (SaaS)" which focuses on developing a transportation application-oriented system that addresses real-world demands. Based on the idea, the paper proposes a Cooperative and Comprehensive Smart Edge Node for Sensing and OpeRation (COCO SENSOR), which was tested in collaboration with the Washington State Department of Transportation and the City of Bellevue.

- **Sensing Technologies:** COCO SENSOR system introduces the cooperative sensing mechanism and coordinates the computation resources on the edge device for multi-task sensing, including visibility detection (92% accuracy), road surface condition detection (91% accuracy), lane-based volume counting (97% accuracy), average speed detection (90%), and object detection (95% accuracy).

- **Parallel Computing:** To handle the limited computation resources on the edge device, COCO SENSOR system introduced three independent threads to coordinate the computation loads for each thread.

- **Real-world Implementation:** Collaborating with WSDOT and the City of Bellevue, Washington, the team set up testbeds to support the implementation of COCO SENSOR system. The team implemented and evaluated four critical applications in the testbed: traffic volume by vehicle type, traffic status detection, low visibility warning, and road surface conditions detection and warning. Additionally, the team developed a mobile APP for information dissemination.

The deployment of COCO SENSOR in various tests highlights its strength in using the "Sensing as a Service (SaaS)" approach but also reveals some limitations. Firstly, while cooperative sensing in COCO SENSOR provides more comprehensive traffic sensing, it lacks effective human interaction. Users can benefit from the system, but their feedback cannot directly influence its customization for better services. Secondly, while COCO SENSOR offers real-time traffic services, it does not fully utilize high-speed local networks for efficient data dissemination. Hence, the research team plans to focus on improving the communication components of the system in the future to enhance its human interaction.

7

Chapter 7. Final Remarks and Envisioning the Future

7.1 Research Contributions and Findings

The advancement of Intelligent Transportation Systems (ITS) hinges on the integration of cutting-edge technologies and customized machine intelligence to address safety, equity, and resilience challenges. This dissertation contributes to these advancements by developing innovative systems that enhance traffic perception, data representation, and infrastructure adaptability. The research contributions are categorized into three key areas, each of which has been thoroughly

explored and documented.

7.2 Part I: Contributions on Situation-Aware Sensing Systems

The foundation of accurate and reliable traffic management lies in the effective acquisition and processing of contextual data. This part of the research focuses on developing adaptive sensing systems that dynamically integrate physical information from diverse transportation scenarios to enhance traffic perception.

- **Key Contribution 1: Integration of Contextual Physical Information for Enhanced Traffic Perception** This research introduces situation-aware sensing systems that significantly improve the reliability, resilience, and accuracy of traffic data by incorporating environmental factors such as weather conditions, road surface states, and traffic densities. These systems are designed to adapt in real-time to changing conditions, leading to more precise and informed decision-making processes. The innovative integration of this contextual information not only enhances traffic management and safety outcomes but also ensures that the system performs optimally across a wide range of scenarios.

- **Key Contribution 2: Scale-Aware Perception for Improved Pedestrian Detection** In addressing the challenges of pedestrian detection, such as occlusion, small object detection, and scale variability, this research develops a scale-aware perception system that adjusts dynamically to varying density conditions. By enhancing the precision of traffic data collection, this system contributes to a safer and more equitable transportation infrastructure, ensuring that vulnerable road users are accurately detected and protected.

7.3 Part II: Contributions on Multimodal Data Representation Learning

To fully leverage the potential of diverse sensor data in ITS, this research focuses on the development of advanced data representation systems that facilitate comprehensive traffic scene understanding and improved sensor cooperation.

- **Key Contribution 3: Enhanced Sensor Cooperation through Multimodal Data Representation** The research presents a novel multimodal data representation learning system that enables a holistic understanding of traffic conditions through the integration of data from various sensors across different types, locations, and times. This system captures critical details such as weather, visibility, road surface states, and traffic volumes, which are essential for optimizing traffic management and ensuring the system's responsiveness in diverse scenarios.
- **Key Contribution 4: Synchronization of Distributed Sensors for Comprehensive Traffic Network Analysis** Innovating further, this research synchronizes sensors dispersed across multiple locations, constructing an overarching view of the traffic network. This framework allows for the identification and tracking of vehicles across different cameras, enabling the accurate estimation of travel times and trajectories. The result is a more connected and intelligent transportation system that enhances both safety and efficiency.

7.4 Part III: Contributions on Demonstrative Cooperative and Equitable Traffic Infrastructure

This part of the research is dedicated to improving transportation equity and safety by developing systems that are responsive to the needs of diverse user groups, including vulnerable road

users and underserved communities.

- **Key Contribution 5: Predictive Dynamic Reversible Lane Control for Traffic Optimization** The research introduces a predictive dynamic reversible lane control system that optimizes traffic flow by adjusting lane directions based on real-time traffic volume data. By incorporating user reactions and interactions as ground truth data, the system ensures that traffic management strategies are both effective and equitable, addressing the needs of a diverse range of road users.

- **Key Contribution 6: User-Centered Innovation for Adaptive Traffic Management** This research emphasizes the importance of user-centered approaches within ITS, focusing on dynamic interactions between the system and its users, including both road users and transportation agencies. By leveraging user inputs and feedback, the system tailors machine intelligence to offer personalized solutions that enhance user satisfaction and engagement. This ensures that the system's capabilities evolve to meet the changing demands of modern urban environments, providing equitable and responsive transportation services.

7.5 Future Research Directions

7.5.1 Intelligent and Cooperative Infrastructure Systems

These future research directions are designed to push the boundaries of current intelligent transportation systems and urban management practices. By focusing on human-system cooperation, decentralized computing, and resilience, my research aims to create smarter, safer, and more

equitable urban environments capable of adapting to the evolving challenges of modern cities. The following three potential research directions summarize my future research objectives:

7.5.2 Intelligent and Cooperative Infrastructure Systems

Urban cyber-physical systems (CPS) represent intricate, interconnected networks requiring seamless integration of physical structures (like intersections and curbsides), digital infrastructure, and human elements (encompassing users, workers, policymakers). Central to this integration is a human-system cooperative approach, essential for creating transportation solutions that are both effective and widely adopted. My future research will explore the dynamics of human interaction with these systems, aiming to advance technologies that facilitate infrastructure-vehicle cooperation. This includes developing sophisticated sensing and data collection methods, enhancing traffic-vehicle control systems, and addressing broader concerns like traffic equity, accessibility, cybersecurity, and privacy. A significant emphasis will be placed on improving infrastructure for vulnerable road users and underserved communities through innovative technology and methodologies. This research aims to create a more inclusive urban environment where all users can benefit from advancements in intelligent transportation systems, ensuring that technological progress leads to equitable and accessible outcomes for everyone.

7.5.3 Edge Computing and Federated Learning for Equitable Smart Cities

The proliferation of edge-computing devices in urban CPS, such as cameras and LiDAR, presents challenges in data centralization and model training due to latency, cost, and privacy issues. My future research will delve into novel distributed computing and machine learning techniques for collaborative and decentralized urban data collection, with a particular emphasis on ubiq-

uitous computing and federated learning. Federated learning, which allows devices to collaboratively learn a shared model while keeping data localized, shows great promise in enhancing model performance, reducing server load, and bolstering user privacy in CPS. This research will investigate the application of federated learning in smart city environments, developing algorithms and frameworks that ensure efficient and secure data processing at the edge. The goal is to create smart city infrastructures that are not only intelligent but also equitable, ensuring that the benefits of technological advancements are distributed fairly across all urban areas and populations.

Cyber-Physical Resilience Modeling and Enhancement

As IoT and ICT technologies become integral to urban management, their reliance on robust infrastructure highlights the risks of cyber failures, necessitating a comprehensive study of network resilience in interconnected smart city infrastructures. My future research will focus on developing an attack-resilient operational framework designed for proactive resource allocation and adaptive response to failures across various urban scenarios, including sensor networks, smart homes, and connected intersections. This framework will identify and address network vulnerabilities, minimize system instability, and provide targeted retrofitting recommendations to strengthen overall resilience. By simulating a range of cyber-physical threats and failure scenarios, the research will aim to enhance the robustness of smart city systems, ensuring that they can withstand and quickly recover from disruptions. This will involve interdisciplinary approaches combining cyber-physical systems engineering, cybersecurity, and urban planning to create resilient, secure, and sustainable urban environments.

References

[1] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[2] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–546, 2018.

[3] Qi Zhang and Antoni B Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8297–8306, 2019.

[4] David Randall Peterman. Federal highway traffic safety policies: Impacts and opportunities. *Congressional Research Service. Report*, 2019.

[5] Michalis Diakakis, Efthymis Lekkas, Iraklis Stamos, and Evangelos Mitsakis. Vulnerability of transport infrastructure to extreme weather events in small rural catchments. *European journal of transport and infrastructure research*, 16(1), 2016.

[6] Nanxi Wang, Min Wu, and Kum Fai Yuen. Modelling and assessing long-term urban transportation system resilience based on system dynamics. *Sustainable Cities and Society*, 109:105548, 2024.

[7] Sören Groth. Multimodal divide: Reproduction of transport poverty in smart mobility trends. *Transportation Research Part A: Policy and Practice*, 125:56–71, 2019.

[8] Rodney Tolley and Brian John Turton. *Transport systems, policy and planning: a geographical approach*. Routledge, 2014.

[9] Hao Frank Yang, Yifan Ling, Cole Kopca, Sam Ricord, and Yinhai Wang. Cooperative traffic signal assistance system for non-motorized users and disabilities empowered by

computer vision and edge artificial intelligence. *Transportation Research Part C: Emerging Technologies*, 145:103896, 2022.

[10] Yinhai Wang, Zhiyong Cui, and Ruimin Ke. *Machine learning for transportation research and applications*. Elsevier, 2023.

[11] Fabio Arena, Giovanni Pau, and Alessandro Severino. A review on ieee 802.11 p for intelligent transportation systems. *Journal of Sensor and Actuator Networks*, 9(2):22, 2020.

[12] Chenxi Liu, Chenlu Pu, Lili Du, and Yinhai Wang. Potentials and challenges of ai-empowered solutions to urban transportation infrastructure systems: Nsf ai-transportation workshop phase i. *Journal of Transportation Engineering, Part A: Systems*, 150(9):02524001, 2024.

[13] Chenlu Pu, Chenxi Liu, Yinhai Wang, and Lili Du. Frontiers of emerging ai technologies best practices and workforce development in transportation: Nsf ai–transportation workshop phase ii. *Journal of Transportation Engineering, Part A: Systems*, 150(9):02524002, 2024.

[14] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. Data fusion in intelligent transportation systems: Progress and challenges–a survey. *Information Fusion*, 12(1):4–10, 2011.

[15] Matthew Veres and Medhat Moussa. Deep learning for intelligent transportation systems: A survey of emerging trends. *IEEE Transactions on Intelligent transportation systems*, 21(8):3152–3168, 2019.

[16] Ammar Haydari and Yasin Yılmaz. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):11–32, 2020.

[17] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and Alex Kai Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1544–1561, 2020.

[18] Dietmar PF Möller and Hamid Vakilzadian. Cyber-physical systems in smart transportation. In *2016 IEEE international conference on electro information technology (EIT)*, pages 0776–0781. IEEE, 2016.

[19] Danda B Rawat, Chandra Bajracharya, and Gongjun Yan. Towards intelligent transportation cyber-physical systems: Real-time computing and communications perspectives. In *SoutheastCon 2015*, pages 1–6. IEEE, 2015.

[20] Chenxi Liu, Hao Yang, Meixin Zhu, Feilong Wang, Torgeir Vaa, and Yinhai Wang. Real-time multi-task environmental perception system for traffic safety empowered by edge artificial intelligence. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[21] LIU Chenxi, KE Ruimin, and Yinhai Wang. Determining a visibility measure based on an image of an environment, February 1 2024. US Patent App. 18/359,607.

[22] Yuepeng Cui, Hao Xu, Jianqing Wu, Yuan Sun, and Junxuan Zhao. Automatic vehicle tracking with roadside lidar data for the connected-vehicles system. *IEEE Intelligent Systems*, 34(3):44–51, 2019.

[23] Ammar Sohail, Muhammad Aamir Cheema, Mohammed Eunus Ali, Adel N Toosi, and Hesham A Rakha. Data-driven approaches for road safety: A comprehensive systematic literature review. *Safety science*, 158:105949, 2023.

[24] Hauer Ezra. On the relationship between road safety research and the practice of road design and operation. *Accident Analysis & Prevention*, 128:114–131, 2019.

[25] Hao Frank Yang, Jiarui Cai, Chenxi Liu, Ruimin Ke, and Yinhai Wang. Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning. *Transportation research part C: emerging technologies*, 148:103982, 2023.

[26] Michael Gerlich. Perceptions and acceptance of artificial intelligence: A multi-dimensional study. *Social Sciences*, 12(9):502, 2023.

[27] Chao Xiang, Chen Feng, Xiaopo Xie, Botian Shi, Hao Lu, Yisheng Lv, Mingchuan Yang, and Zhendong Niu. Multi-sensor fusion and cooperative perception for autonomous driving: A review. *IEEE Intelligent Transportation Systems Magazine*, 2023.

[28] Abhishek Thakur and Sudhansu Kumar Mishra. An in-depth evaluation of deep learning-enabled adaptive approaches for detecting obstacles using sensor-fused data in autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 133:108550, 2024.

[29] Xinyi Li, Yinlong Liu, Venkatnarayanan Lakshminarasimhan, Hu Cao, Feihu Zhang, and Alois Knoll. Globally optimal robust radar calibration in intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(6):6082–6095, 2023.

[30] Markus Münzinger, Nikolas Prechtel, and Martin Behnisch. Mapping the urban forest in detail: From lidar point clouds to 3d tree models. *Urban Forestry & Urban Greening*, 74:127637, 2022.

[31] Efi Dvir, Mark Shifrin, and Omer Gurewitz. Cooperative multi-agent reinforcement learning for data gathering in energy-harvesting wireless sensor networks. *Mathematics*, 12(13):2102, 2024.

[32] Rongxin Zhu, Azzedine Boukerche, and Qiuling Yang. An efficient secure and adaptive routing protocol based on gmm-hmm-lstm for internet of underwater things. *IEEE Internet of Things Journal*, 2024.

[33] Numan Senel, Klaus Kefferpütz, Kristina Doycheva, and Gordon Elger. Multi-sensor data fusion for real-time multi-object tracking. *Processes*, 11(2):501, 2023.

[34] Philipp Markert, Leona Lassak, Maximilian Golla, and Markus Dürmuth. Understanding users' interaction with login notifications. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2024.

[35] Chenxi Liu, Hao Yang, Ruimin Ke, Wei Sun, Julia Wang, and Yinhai Wang. Cooperative and comprehensive multi-task surveillance sensing and interaction system empowered by edge artificial intelligence. *Transportation research record*, 2677(9):652–668, 2023.

[36] Ruimin Ke, Chenxi Liu, Hao Yang, Wei Sun, and Yinhai Wang. Real-time traffic and road surveillance with parallel edge intelligence. *IEEE Journal of Radio Frequency Identification*, 6:693–696, 2022.

[37] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[38] Mengchi Liu and Dongmei Yu. Towards intelligent e-learning systems. *Education and Information Technologies*, 28(7):7845–7876, 2023.

[39] Robertas Damaševičius, Nebojsa Bacanin, and Sanjay Misra. From sensors to safety: Internet of emergency services (ioes) for emergency response and disaster management. *Journal of Sensor and Actuator Networks*, 12(3):41, 2023.

[40] Xiaoliang Xie, Linglu Huang, Stephen M Marson, and Guo Wei. Emergency response process for sudden rainstorm and flooding: Scenario deduction and bayesian

network analysis using evidence theory and knowledge meta-theory. *Natural Hazards*, 117(3):3307–3329, 2023.

[41] Di Chen, Meixin Zhu, Hao Yang, Xuesong Wang, and Yinhai Wang. Data-driven traffic simulation: A comprehensive review. *IEEE Transactions on Intelligent Vehicles*, 2024.

[42] Hao Yang, Chenxi Liu, Meixin Zhu, Xuegang Ban, and Yinhai Wang. How fast you will drive? predicting speed of customized paths by deep neural network. *IEEE transactions on intelligent transportation systems*, 23(3):2045–2055, 2021.

[43] C Ludwig, J Psotta, A Buch, N Kolaxidis, S Fendrich, M Zia, J Fürle, A Rousell, and A Zipf. Traffic speed modelling to improve travel time estimation in openrouteservice. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:109–116, 2023.

[44] Haoran Chen, Xuedong Yan, Xiaobing Liu, and Tao Ma. Exploring the operational performance discrepancies between online ridesplitting and carpooling transportation modes based on didi data. *Transportation*, 50(5):1923–1958, 2023.

[45] Xuan Feng, Qinping Lin, Ning Jia, and Junfang Tian. The actual impact of ride-splitting: An empirical study based on large-scale gps data. *Transport Policy*, 147:94–112, 2024.

[46] Meixin Zhu, Xuesong Wang, and Yinhai Wang. Human-like autonomous car-following model with deep reinforcement learning. *Transportation research part C: emerging technologies*, 97:348–368, 2018.

[47] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143*, 2018.

[48] Chenxi Liu, Hao Yang, Ruimin Ke, and Yinhai Wang. Toward a dynamic reversible lane management strategy by empowering learning-based predictive assignment scheme. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23311–23323, 2022.

[49] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044, 2015.

[50] Enjian Yao, Tong Liu, Tianwei Lu, and Yang Yang. Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society*, 52:101862, 2020.

[51] Samuel Ricord and Yinhai Wang. Investigation of equity biases in transportation data: a literature review synthesis. *Journal of transportation engineering, Part A: Systems*, 149(11):03123004, 2023.

[52] Feilong Wang, Xuegang Ban, Peng Chen, Chenxi Liu, and Rong Zhao. Mitigating biases in big mobility data: a case study of monitoring large-scale transit systems. *Transportation Letters*, pages 1–14, 2024.

[53] Samuel Ricord and Yinhai Wang. Understanding the potential equity issues of loop detector data. In *International Conference on Transportation and Development 2022*, pages 207–218, 2022.

[54] Zhiyong Cui, Shen Zhang, Kristian C Henrickson, and Yinhai Wang. New progress of drive net: An e-science transportation platform for data sharing, visualization, modeling, and analysis. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–2. IEEE, 2016.

[55] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

[56] Shafiza Ariffin Kashinath, Salama A Mostafa, Aida Mustapha, Hairulnizam Mahdin, David Lim, Moamin A Mahmoud, Mazin Abed Mohammed, Bander Ali Saleh Al-Rimy, Mohd Farhan Md Fudzee, and Tan Jhon Yang. Review of data fusion methods for real-time and multi-sensor traffic flow analysis. *IEEE Access*, 9:51258–51276, 2021.

[57] Christoph Wiesmeyr, Carmina Coronel, Martin Litzenberger, Herbert Josef Döller, Hans-Bernhard Schweiger, and Gaetan Calbris. Distributed acoustic sensing for vehicle speed and traffic flow estimation. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pages 2596–2601. IEEE, 2021.

[58] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

[59] Cong Ma, Changshui Yang, Fan Yang, Yueqing Zhuang, Ziwei Zhang, Huizhu Jia, and Xiaodong Xie. Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. *arXiv preprint arXiv:1804.04555*, 2018.

[60] Zhiyong Cui, Ruimin Ke, and Yinhai Wang. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In *6th International Workshop on Urban Computing (UrbComp 2017)*, 2016.

[61] Zhiyong Cui, Longfei Lin, Ziyuan Pu, and Yinhai Wang. Graph markov network for traffic forecasting with missing data. *Transportation Research Part C: Emerging Technologies*, 117:102671, 2020.

[62] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118:102674, 2020.

[63] Athanasios Theofilatos and George Yannis. A review of the effect of traffic and weather characteristics on road safety. *Accident Analysis & Prevention*, 72:244–256, 2014.

[64] Dariush Haghighi-Talab. An investigation into the relationship between rainfall and road accident frequencies in two cities. *Accident Analysis & Prevention*, 5(4):343–349, 1973.

[65] Lasse Fridstrøm, Jan Ifver, Siv Ingebrigtsen, Risto Kulmala, and Lars Krogsgård Thomsen. Measuring the contribution of randomness, exposure, weather, and daylight to the variation in road accident counts. *Accident Analysis & Prevention*, 27(1):1–20, 1995.

[66] Akram Khaleghei Ghosheh Balagh, Farnoosh Naderkhani, and Viliam Makis. Highway accident modeling and forecasting in winter. *Transportation research part A: policy and practice*, 59:384–396, 2014.

[67] Magda Bogalecka and Krysztof Kotowrocki. Prediction of critical infrastructure accident losses of chemical releases impacted by climate-weather change. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 788–792. IEEE, 2018.

[68] Bertrand D Tanner. Automated weather stations. *Remote Sensing Reviews*, 5(1):73–98, 1990.

[69] Michele Citterio, Dirk van As, Andreas P Ahlstrøm, Morten L Andersen, Signe B Andersen, Jason E Box, Charalampos Charalampidis, William T Colgan, Robert S Fausto, Søren Nielsen, et al. Automatic weather stations for basic and applied glaciological research. *Geological Survey of Denmark and Greenland Bulletin*, 33:69–72, 2015.

[70] Martin Hendel, Morgane Colombert, Youssef Diab, and Laurent Royon. Improving a pavement-watering method on the basis of pavement surface temperature measurements. *Urban Climate*, 10:189–200, 2014.

[71] Patrik Jonsson. Road status sensors: A comparison of active and passive sensors. In *Proc. 16th ITS World Congress and Exhibition on Intelligent Transport Systems and Services*, 2009.

[72] Gurkan Erdogan, Lee Alexander, and Rajesh Rajamani. Estimation of tire-road friction coefficient using a novel wireless piezoelectric tire sensor. *IEEE Sensors Journal*, 11(2):267–279, 2010.

[73] Matti Kutila, Pasi Pyykönen, Werner Ritter, Oliver Sawade, and Bernd Schäufele. Automotive lidar sensor development scenarios for harsh weather conditions. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 265–270. IEEE, 2016.

[74] Meixin Zhu, Hao Frank Yang, Chenxi Liu, Ziyuan Pu, and Yinhai Wang. Real-time crash identification using connected electric vehicle operation data. *Accident Analysis & Prevention*, 173:106708, 2022.

[75] Patrik Jonsson, Johan Casselgren, and Benny Thörnberg. Road surface status classification using spectral analysis of nir camera images. *IEEE Sensors Journal*, 15(3):1641–1656, 2014.

[76] Adham Mohamed, Mohamed Mostafa M Fouad, Esraa Elhariri, Nashwa El-Bendary, Hossam M Zawbaa, Mohamed Tahoun, and Aboul Ella Hassanien. Roadmonitor: An intelligent road surface condition monitoring system. In *Intelligent Systems' 2014*, pages 377–387. Springer, 2015.

[77] Guangyuan Pan, Liping Fu, Ruifan Yu, and Matthew Muresan. Evaluation of alternative pre-trained convolutional neural networks for winter road surface condition monitoring. In *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, pages 614–620. IEEE, 2019.

[78] Md Nasim Khan and Mohamed M Ahmed. Weather and surface condition detection based on road-side webcams: Application of pre-trained convolutional neural network. *International Journal of Transportation Science and Technology*, 2021.

[79] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[80] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[81] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[82] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[83] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[84] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[85] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[86] Robert T Collins. Mean-shift blob tracking through scale space. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–234. IEEE, 2003.

[87] Tim Van Oosterhout, Sander Bakkes, Ben JA Kröse, et al. Head detection in stereo data for people counting and segmentation. In *VISAPP*, pages 620–625, 2011.

[88] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *2011 International Conference on Computer Vision*, pages 2423–2430. IEEE, 2011.

[89] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[90] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[91] Xiaowei Hu, Yun Liu, Kai Wang, and Bo Ren. Learning hybrid convolutional features for edge detection. *Neurocomputing*, 313:377–385, 2018.

[92] Chaoxu Guo, Bin Fan, Qian Zhang, Shiming Xiang, and Chunhong Pan. Augfpn: Improving multi-scale feature learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[93] Chengli Peng, Tian Tian, Chen Chen, Xiaojie Guo, and Jiayi Ma. Bilateral attention decoder: A lightweight decoder for real-time semantic segmentation. *Neural Networks*, 137:188–199, 2021.

[94] Xian Sun, Peijin Wang, Cheng Wang, Yingfei Liu, and Kun Fu. Pbnet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:50–65, 2021.

[95] Jiaxu Leng, Yihui Ren, Wen Jiang, Xiaoding Sun, and Ye Wang. Realize your surroundings: Exploiting context information for small object detection. *Neurocomputing*, 433:287–299, 2021.

[96] Zuyao Chen, Qianqian Xu, Runmin Cong, and Qingming Huang. Global context-aware progressive aggregation network for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10599–10606, 2020.

[97] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[98] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. *arXiv preprint arXiv:2105.12555*, 2021.

[99] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *Bmvc*, volume 1, page 3, 2012.

[100] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[101] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[102] Nikos Paragios and Visvanathan Ramesh. A mrf-based approach for real-time subway monitoring. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[103] Yan Tian, Leonid Sigal, Hernán Badino, Fernando De la Torre, and Yong Liu. Latent gaussian mixture regression for human pose estimation. In *Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III 10*, pages 679–690. Springer, 2011.

[104] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3253–3261, 2015.

[105] Hua Yang, Yihua Cao, Shuang Wu, Weiyao Lin, Shibao Zheng, and Zhenghua Yu. Abnormal crowd behavior detection based on local pressure model. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–4. IEEE, 2012.

[106] Norhaida Hussain, Halimatul Saadiah Md Yatim, Nor Liza Hussain, Jasy Liew Suet Yan, and Fazilah Haron. Cdes: A pixel-based crowd density estimation system for masjid al-haram. *Safety Science*, 49(6):824–833, 2011.

[107] Ge Yang and Dian Zhu. Survey on algorithms of people counting in dense crowd and crowd density estimation. *Multimedia Tools and Applications*, 82(9):13637–13648, 2023.

[108] Junyu Gao, Yuan Yuan, and Qi Wang. Feature-aware adaptation and density alignment for crowd counting in video surveillance. *IEEE transactions on cybernetics*, 51(10):4822–4833, 2020.

[109] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302, 2015.

[110] Min Fu, Pei Xu, Xudong Li, Qihe Liu, Mao Ye, and Ce Zhu. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43:81–88, 2015.

[111] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 833–841, 2015.

[112] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5744–5752, 2017.

[113] Biyun Sheng, Chunhua Shen, Guosheng Lin, Jun Li, Wankou Yang, and Changyin Sun. Crowd counting via weighted vlad on a dense attribute feature map. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1788–1797, 2016.

[114] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *Proceedings of the IEEE international conference on computer vision*, pages 1861–1870, 2017.

[115] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[116] Congcong Wen, Xiang Li, Xiaojing Yao, Ling Peng, and Tianhe Chi. Airborne lidar point cloud classification with global-local graph attention convolution neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:181–194, 2021.

[117] Weihang Kong, He Li, Guanglong Xing, and Fengda Zhao. An automatic scale-adaptive approach with attention mechanism-based crowd spatial information for crowd counting. *IEEE Access*, 7:66215–66225, 2019.

[118] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[119] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[120] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3645–3649. IEEE, 2017.

[121] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[122] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019.

[123] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 182:50–63, 2019.

[124] Rene O Sanchez, Christopher Flores, Roberto Horowitz, Ram Rajagopal, and Pravin Varaiya. Vehicle re-identification using wireless magnetic sensors: Algorithm revision, modifications and performance analysis. In *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*, pages 226–231. IEEE, 2011.

[125] Yegor Malinovskiy, Yao-Jan Wu, Yinhai Wang, and Un Kun Lee. Field experiments on bluetooth-based travel time data collection. Technical report, 2010.

[126] Yegor Malinovskiy, Un-Kun Lee, Yao-Jan Wu, and Yinhai Wang. Investigation of bluetooth-based travel time estimation error on a short corridor. Technical report, 2011.

[127] Michael Abbott-Jard, Harpal Shah, and Ashish Bhaskar. Empirical evaluation of bluetooth and wifi scanning for road transport. In *Australasian Transport Research Forum (ATRF), 36th*, page 14, 2013.

[128] Hyunggi Cho, Young-Woo Seo, BVK Vijaya Kumar, and Ragunathan Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1836–1843. IEEE, 2014.

[129] Ryan A Kerekes, Thomas P Karnowski, Mike Kuhn, Michael R Moore, Brad Stinson, Ryan Tokola, Adam Anderson, and Jason M Vann. Vehicle classification and identification using multi-modal sensing and signal learning. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2017.

[130] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016.

[131] Dominik Zapletal and Adam Herout. Vehicle re-identification for automatic video traffic surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–31, 2016.

[132] Qi Zheng, Chao Liang, Wenhua Fang, Da Xiang, Xin Zhao, Chengping Ren, and Jun Chen. Car re-identification from large scale images using semantic attributes. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5. IEEE, 2015.

[133] Rogerio Schmidt Feris, Behjat Siddiquie, James Petterson, Yun Zhai, Ankur Datta, Lisa M Brown, and Sharath Pankanti. Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Transactions on Multimedia*, 14(1):28–42, 2011.

[134] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2017.

[135] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[136] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, pages 869–884. Springer, 2016.

[137] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1900–1909, 2017.

[138] Yi Zhou, Li Liu, and Ling Shao. Vehicle re-identification by deep hidden multi-view inference. *IEEE Transactions on Image Processing*, 27(7):3275–3287, 2018.

[139] Tsung-Wei Huang, Jiarui Cai, Hao Yang, Hung-Min Hsu, and Jenq-Neng Hwang. Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019.

[140] Kai Lv, Weijian Deng, Yunzhong Hou, Heming Du, Hao Sheng, Jianbin Jiao, and Liang Zheng. Vehicle reidentification with the location and time stamp. In *Proc. CVPR Workshops*, 2019.

[141] Hao Chen, Benoit Lagadec, and Francois Bremond. Partition and reunion: A two-branch neural network for vehicle re-identification. In *Proc. CVPR Workshops*, pages 184–192, 2019.

[142] Ming-Ching Chang, Jiayi Wei, Zheng-An Zhu, Yan-Ming Chen, Chan-Shuo Hu, Ming-Xiu Jiang, and Chen-Kuo Chiang. Ai city challenge 2019–city-scale video analytics for smart transportation. In *Proc. CVPR Workshops*, pages 99–108, 2019.

[143] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019.

[144] Xiao Tan, Zhigang Wang, Minyue Jiang, Xipeng Yang, Jian Wang, Yuan Gao, Xiangbo Su, Xiaoqing Ye, Yuchen Yuan, Dongliang He, Shilei Wen, and Errui Ding. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[145] Soufiene Djahel, Ronan Doolan, Gabriel-Miro Muntean, and John Murphy. A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Communications Surveys & Tutorials*, 17(1):125–151, 2014.

[146] Chenxi Liu, Hao Yang, Meixin Zhu, Feilong Wang, Torgeir Vaa, and Yinhai Wang. Real-time multi-task environmental perception system for traffic safety challenges empowered by edge artificial intelligence. *Available at SSRN 4265369*.

[147] Lawrence A Klein. *ITS sensors and architectures for traffic management and connected vehicles*. CRC Press, 2017.

[148] Meng-Ju Tsai, Zhiyong Cui, Chenxi Liu, Hao Yang, and Yinhai Wang. An incremental learning-based framework for non-stationary traffic representations clustering and forecasting. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3237–3242. IEEE, 2022.

[149] Hao Yang, Chenxi Liu, Christopher Gottsacker, Xuegang Ban, Chao Zhang, and Yinhai Wang. Cell-speed prediction neural network (cpnn): A deep learning approach for trip-based speed prediction. Technical report, 2019.

[150] Yinhai Wang, Wei Sun, and Chenxi Liu. Data for cooperative perception of road-side unit and on-board equipment with edge artificial intelligence for driving assistance [supporting dataset]. 2021.

[151] Chenxi Liu. Bi-level optimization algorithm for dynamic reversible lane control based on short-term traffic flow prediction. Master's thesis, University of Washington, 2020.

[152] Allan M De Souza, Celso ARL Brennand, Roberto S Yokoyama, Erick A Donato, Edmundo RM Madeira, and Leandro A Villas. Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13(4):1550147716683612, 2017.

[153] Xuan Zhou, Ruimin Ke, Hao Yang, and Chenxi Liu. When intelligent transportation systems sensing meets edge computing: Vision and challenges. *Applied Sciences*, 11(20):9680, 2021.

[154] Xin Fu, Hao Yang, Chenxi Liu, Jianwei Wang, and Yinhai Wang. A hybrid neural network for large-scale expressway network od prediction based on toll data. *PloS one*, 14(5):e0217241, 2019.

[155] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):831–843, 2000.

[156] Peyman Babaei. Vehicles tracking and classification using traffic zones in a hybrid scheme for intersection traffic management by smart cameras. In *2010 International Conference on Signal and Image Processing*, pages 49–53. IEEE, 2010.

[157] David Felguera-Martín, José-Tomás González-Partida, Pablo Almorox-González, and Mateo Burgos-García. Vehicular traffic surveillance and road lane detection using radar interferometry. *IEEE transactions on vehicular technology*, 61(3):959–970, 2012.

[158] Yinhai Wang, Wei Sun, Chenxiao Liu, Zhiyong Cui, Meixin Zhu, Ziyuan Pu, et al. Cooperative perception of roadside unit and onboard equipment with edge artificial intelligence for driving assistance. 2021.

[159] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

[160] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[161] Aidin Ferdowsi, Ursula Challita, and Walid Saad. Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview. *ieee vehicular technology magazine*, 14(1):62–70, 2019.

[162] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[163] Xiao Ling, Jie Sheng, Orlando Baiocchi, Xing Liu, and Matthew E Tolentino. Identifying parking spaces & detecting occupancy using vision-based iot devices. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2017.

[164] Junhao Zhou, Hong-Ning Dai, and Hao Wang. Lightweight convolution neural networks for mobile edge computing in transportation cyber physical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(6):1–20, 2019.

[165] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[166] Michael Aeberhard and Nico Kaempchen. High-level sensor data fusion architecture for vehicle surround environment perception. In *Proc. 8th Int. Workshop Intell. Transp*, volume 665, 2011.

[167] Andreas Rauch, Felix Klanner, and Klaus Dietmayer. Analysis of v2x communication parameters for the development of a fusion architecture for cooperative perception systems. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 685–690. IEEE, 2011.

[168] Michael Aeberhard, Sascha Paul, Nico Kaempchen, and Torsten Bertram. Object existence probability fusion using dempster-shafer theory in a high-level sensor data fusion architecture. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 770–775. IEEE, 2011.

[169] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[170] Manabu Tsukada, Takaharu Oi, Masahiro Kitazawa, and Hiroshi Esaki. Networked roadside perception units for autonomous driving. *Sensors*, 20(18):5320, 2020.

[171] Ameni Chtourou, Pierre Merdrignac, and Oyunchimeg Shagdar. Collective perception service for connected vehicles and roadside infrastructure. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–5. IEEE, 2021.

[172] Jiong Jin, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things journal*, 1(2):112–121, 2014.

[173] Wenjia Li, Houbing Song, and Feng Zeng. Policy-based secure and trustworthy sensing for internet of things in smart cities. *IEEE Internet of Things Journal*, 5(2):716–723, 2017.

[174] Christopher Neff, Matías Mendieta, Shrey Mohan, Mohammadreza Baharani, Samuel Rogers, and Hamed Tabkhi. Revamp 2 t: Real-time edge video analytics for multicamera privacy-aware pedestrian tracking. *IEEE Internet of Things Journal*, 7(4):2591–2602, 2019.

[175] Ruimin Ke, Yifan Zhuang, Ziyuan Pu, and Yinhai Wang. A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on iot devices. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[176] Bruce Rose and Eric Floehr. Analysis of high-temperature forecast accuracy of consumer weather forecasts from 2005–2016. *Forecast Watch Report.(Dublin, OH, USA)*, 2017.

[177] FHWA (Federal Highway Administration). How do weather events impact roads? 2011.

[178] Mohamed Abdel-Aty, Al-Ahad Ekram, Helai Huang, and Keechoo Choi. A study on crashes related to visibility obstruction due to fog and smoke. *Accident Analysis & Prevention*, 43(5):1730–1737, 2011.

[179] Ashraf M Rahim, Gregg Fiegel, Khalid Ghuzlan, and Dan Khumann. Evaluation of international roughness index for asphalt overlays placed over cracked and seated concrete pavements. *International Journal of Pavement Engineering*, 10(3):201–207, 2009.

[180] Highway Capacity Manual. Highway capacity manual. *Washington, DC*, 2(1), 2000.

[181] Rituraj Neog, Shukla Acharjee, and Jiten Hazarika. Spatiotemporal analysis of road surface temperature (rst) and building wall temperature (bwt) and its relation to the traffic volume at jorhat urban environment, india. *Environment, Development and Sustainability*, 23:10080–10092, 2021.

[182] Mohammad Aldibaja, Naoki Suganuma, and Keisuke Yoneda. Robust intensity-based localization method for autonomous driving on snow–wet road surface. *IEEE Transactions on industrial Informatics*, 13(5):2369–2378, 2017.

[183] Amedeo Troiano, Eros Pasero, and Luca Mesin. New system for detecting road ice formation. *IEEE Transactions on Instrumentation and Measurement*, 60(3):1091–1101, 2010.

[184] Daniel Kravetz and Robert B Noland. Spatial analysis of income disparities in pedestrian safety in northern new jersey: is there an environmental justice issue? *Transportation research record*, 2320(1):10–17, 2012.

[185] Ke Ma and Hao Wang. How connected and automated vehicle–exclusive lanes affect on-ramp junctions. *Journal of Transportation Engineering, Part A: Systems*, 147(2):04020157, 2021.

[186] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.

[187] Robby T Tan. Visibility in bad weather from a single image. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[188] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE transactions on image processing*, 24(11):3522–3533, 2015.

[189] Codruta O Ancuti, Cosmin Ancuti, Chris Hermans, and Philippe Bekaert. A fast semi-inverse approach to detect and remove the haze from a single image. In *Asian Conference on Computer Vision*, pages 501–514. Springer, 2010.

[190] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[191] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013.

[192] Ketan Tang, Jianchao Yang, and Jue Wang. Investigating haze-relevant features in a learning framework for image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2995–3000, 2014.

[193] Donald F Swinehart. The beer-lambert law. *Journal of chemical education*, 39(7):333, 1962.

[194] Brian Clearman. *International Marine Aids to Navigation*. Mount Angel Abbey, 2010.

[195] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[196] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[197] Weibin Rong, Zhanjing Li, Wei Zhang, and Lining Sun. An improved canny edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation*, pages 577–582. IEEE, 2014.

[198] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[199] Adella Santos, Nancy McGuckin, Hikari Yukiko Nakamoto, Danielle Gray, Susan Liss, et al. Summary of travel trends: 2009 national household travel survey. Technical report, United States. Federal Highway Administration, 2011.

[200] Ruimin Ke, Zhiyong Cui, Yanlong Chen, Meixin Zhu, Hao Yang, and Yinhai Wang. Edge computing for real-time near-crash detection for smart transportation applications. *arXiv preprint arXiv:2008.00549*, 2020.

[201] Hao Yang, Chenxi Liu, Meixin Zhu, Wei Sun, and Yinhai Wang. Hybrid data-fusion model for short-term road hazardous segments identification based on the acceleration and deceleration information. In *International Conference on Transportation and Development 2020*, pages 313–326. American Society of Civil Engineers Reston, VA, 2020.

[202] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[203] Xiyang Liu, Jie Yang, Wenrui Ding, Tieqiang Wang, Zhijin Wang, and Junjun Xiong. Adaptive mixture regression network with local counting map for crowd counting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 241–257. Springer, 2020.

[204] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018.

[205] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):2141–2149, 2020.

[206] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[207] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[208] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[209] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017.

[210] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3486–3498, 2019.

[211] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. *Advances in neural information processing systems*, 33:1595–1607, 2020.

[212] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. Crowd counting with deep structured scale integration network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1774–1783, 2019.

[213] Hao Yang, Jiarui Cai, Meixin Zhu, Chenxi Liu, and Yinhai Wang. Traffic-informed multi-camera sensing (tims) system based on vehicle re-identification. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17189–17200, 2022.

[214] Ruimin Ke, Zhibin Li, Sung Kim, John Ash, Zhiyong Cui, and Yinhai Wang. Real-time bidirectional traffic flow parameter estimation from aerial videos. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):890–901, 2016.

[215] Guohui Zhang, Ryan P Avery, and Yinhai Wang. Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. *Transportation research record*, 1993(1):138–147, 2007.

[216] Yegor Malinovskiy, Yao-Jan Wu, and Yinhai Wang. Video-based vehicle detection and tracking using spatiotemporal maps. *Transportation research record*, 2121(1):81–89, 2009.

[217] Álvaro González, Miguel Ángel Garrido, David Fernández Llorca, Miguel Gavilán, J Pablo Fernández, Pablo F Alcantarilla, Ignacio Parra, Fernando Herranz, Luis M

Bergasa, Miguel Ángel Sotelo, et al. Automatic traffic signs and panels inspection system using computer vision. *IEEE Transactions on intelligent transportation systems*, 12(2):485–499, 2011.

[218] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[219] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[220] Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018.

[221] Hao Frank Yang. Novel traffic sensing using multi-camera car tracking and re-identification (mcctri). Master's thesis, University of Washington, 2020.

[222] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3038–3046, 2017.

[223] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 941–951, 2019.

[224] GeekAlexis. Fast mot, 2020.

[225] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[226] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3702–3712, 2019.

[227] David Schrank, Tim Lomax, and Bill Eisele. 2017 urban mobility report. *Texas Transportation Institute,[ONLINE]. Available: http://mobility. tamu. edu/ums/report*, 2019.

[228] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[229] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[230] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[231] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J Krishna Murthy, and K Madhava Krishna. The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8404–8410. IEEE, 2018.

[232] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[233] Ratnesh Kuma, Edwin Weill, Farzin Aghdasi, and Parthasarathy Sriram. Vehicle re-identification: an efficient baseline using triplet embedding. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2019.

[234] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[235] Meiping Yun and Wenwen Qin. Minimum sampling size of floating cars for urban link travel time distribution estimation. *Transportation Research Record*, 2673(3):24–43, 2019.

[236] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[237] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 6(7):8, 2020.

[238] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *arXiv preprint arXiv:2001.04193*, 2020.

[239] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018.

[240] Dinh-Van Nguyen, Fawzi Nashashibi, Trung-Kien Dao, and Eric Castelli. Improving poor gps area localization for intelligent vehicles. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 417–421. IEEE, 2017.

[241] Moein Shakeri and Hong Zhang. Moving object detection in time-lapse or motion trigger image sequences using low-rank and invariant sparse decomposition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[242] Zi Yang and Lilian SC Pun-Cheng. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. *Image and Vision Computing*, 69:143–154, 2018.

[243] ITS America et al. The national architecture for its: A framework for integrated transportation into the 21st century. Technical report, United States. Department of Transportation, 1996.

[244] Liz Greer, Janet L Fraser, Drennan Hicks, Mike Mercer, Kathy Thompson, et al. Intelligent transportation systems benefits, costs, and lessons learned: 2018 update report. Technical report, United States. Dept. of Transportation. ITS Joint Program Office, 2018.

[245] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[246] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.