AI

MDPI

Article

Automated Audible Truck-Mounted Attenuator Alerts: Vision System Development and Evaluation

Neema Jakisa Owor *, Yaw Adu-Gyamfi *, Linlin Zhang and Carlos Sun

Department of Civil and Environmental Engineering, University of Missouri, Columbia, MO 65211, USA; linlinzhang@missouri.edu (L.Z.); csun@missouri.edu (C.S.)
* Correspondence: nodyv@missouri.edu (N.J.O.); adugyamfiy@missouri.edu (Y.A.-G.)

Abstract: Background: The rise in work zone crashes due to distracted and aggressive driving calls for improved safety measures. While Truck-Mounted Attenuators (TMAs) have helped reduce crash severity, the increasing number of crashes involving TMAs shows the need for improved warning systems. Methods: This study proposes an AI-enabled vision system to automatically alert drivers on collision courses with TMAs, addressing the limitations of manual alert systems. The system uses multi-task learning (MTL) to detect and classify vehicles, estimate distance zones (danger, warning, and safe), and perform lane and road segmentation. MTL improves efficiency and accuracy, making it ideal for devices with limited resources. Using a Generalized Efficient Layer Aggregation Network (GELAN) backbone, the system enhances stability and performance. Additionally, an alert module triggers alarms based on speed, acceleration, and time to collision. Results: The model achieves a recall of 90.5%, an mAP of 0.792 for vehicle detection, an mIOU of 0.948 for road segmentation, an accuracy of 81.5% for lane segmentation, and 83.8% accuracy for distance classification. Conclusions: The results show the system accurately detects vehicles, classifies distances, and provides real-time alerts, reducing TMA collision risks and enhancing work zone safety.

Keywords: multi-task learning; work zone safety; Truck Mounted Attenuators (TMA); automated audible alerts; computer vision

Citation: Owor, N.J.; Adu-Gyamfi, Y.; Zhang, L.; Sun, C. Automated Audible Truck-Mounted Attenuator Alerts: Vision System Development and Evaluation. *AI* **2024**, *5*, 1816–1836. https://doi.org/10.3390/ai5040090

Academic Editor: Arslan Munir

Received: 30 July 2024
Revised: 21 September 2024
Accepted: 29 September 2024
Published: 8 October 2024

1. Introduction

The number and severity of work zone crashes has been steadily increasing over the past decade. About 100,000 work zone crashes occurred between 2013 and 2021, resulting in 42,000 injuries and 924 fatalities (an increase of 61%) [1] with a 7% decrease in work zone fatalities in 2022 [2]. In 2022, work zone crashes resulted in 891 fatalities and 37,701 injuries [2]. A 2022 nationwide study on highway work zone safety by the Associated General Contractors of America (AGC) found that 64% of highway contractors experienced work zone crashes [3]. According to Federal and State departments of transportation, distracted and aggressive driving were the key factors perpetuating this unfortunate trend [4]. The FHWA mandated that each highway agency must revise their state safety mobility plan by 2008 to incorporate positive protection measures in work zones [5]. These measures are designed to mitigate the impact of work zone crashes and enhance safety for both workers and the traveling public. The TMA (Truck-Mounted Attenuator) is an example of a protective device used in work zones that provides positive protection to workers by absorbing the energy from rear-end motor vehicle collisions. Studies [6] have confirmed that deploying TMAs in work zones helped reduce the risk of injury or death and cost of a crash by 1.8 and 3.5 times, respectively. The number of TMA hits in recent years has, however, increased; for example, in Virginia, TMA crashes have increased by 52.9%, 26.9%, and 36.4 from 2011 to 2012, 2012 to 2013, and 2013 to 2014, respectively [7]. In Missouri, the number of TMA crashes rose by 20% between 2020 and 2023 [8]. This has necessitated the need for audible alert systems to provide early warning to distracted drivers

who are on collision course with a TMA. These alert systems are manually operated, which presents several challenges—(i) human error: the person responsible for activating the system may forget or become distracted, leading to delays in deploying or no deployment of the alert. (ii) Slower response time: manual systems depend on human reaction, which can be slower under high stress, unlike automated systems that respond instantly. (iii) Operator fatigue: operators may experience fatigue due to the monotonous environment, reducing the effectiveness of the alert system. (iv) High operational cost: manual systems require more personnel, increasing operational costs since someone must always be present to manage the system. (v) Increased risk of injury: If the system is manually operated and a TMA is struck, the operator could be harmed. Automated systems, however, eliminate this risk by removing the need for a human operator in such situations. The goal of the current study is to develop and evaluate an AI-enabled vision system that can be used to automatically trigger an audible alert to drivers on collision course with a TMA.

The minimum requirements of an automated audible TMA alert system should include the following abilities: (1) detect and classify all vehicles surrounding the TMA; (2) track and determine the direction of each vehicle with respect to the TMA; (3) flag vehicle(s) on collision course based on factors such as speed, acceleration, and time to collision; (4) send an appropriate alert to the driver on collision course. Recent advances in deep machine learning have significantly improved the accuracy of vision systems used for vehicle recognition, lane detection, tracking, and other activities needed to improve real-time situational awareness and safety on roadways. Traditionally, these systems are designed to address a single task, such as vehicle detection, lane segmentation, or driving area segmentation. However, an audible TMA alert system requires a vision system that can perform multiple tasks simultaneously: detect vehicles, track their position with respect to the TMA, identify traveling lanes, and flag vehicles on collision course with the TMA. While multiple, single-task models could be integrated to deploy an alert system, the computational cost of running multiple models is prohibitively expensive and impractical for real-time applications.

To address this challenge, the current study implements the framework of a single model that can multi-task on different but related problems needed to automatically operate an audible TMA alert system. Multi-task learning (MTL) has emerged as a pivotal technique in the realm of machine learning, demonstrating significant advancements across various domains. MTL aims to enhance the performance of multiple related tasks by leveraging shared representations and learning jointly, rather than independently. The foundational principle behind MTL is that learning multiple tasks concurrently can lead to improved generalization and efficiency, particularly when the tasks are related and can benefit from shared information. This makes MTL particularly suitable for deployment on edge devices with limited memory and computing capabilities, where running multiple models simultaneously may not be feasible. The current study takes inspiration from Panoptic models, exemplified by such as YOLOP [9], YOLOP2 [10], and Hybridnets [11], which have demonstrated remarkable capabilities in simultaneously performing car detection, lane detection, and driving area segmentation.

YOLOP [9] was the pioneering model designed to address three tasks (detect cars, segment lanes, and segment driving areas) simultaneously, achieving state-of-the-art (SOTA) performance on embedded devices with end-to-end training. Subsequent models like Hybridnets [11] addressed specific weaknesses, incorporating techniques such as customized anchors, multi-scale bi-directional feature networks, and hybrid loss functions to enhance performance, while YOLOP2 [10] uses data augmentation (mosaic and mixup) and new hybrid loss for enhance the model's generalization. YOLOPX [12] introduced anchor-free detection and attention mechanisms in lane detection to handle the loss of long-range contextual dependencies, while Ehsinet [13] utilized recursive gated convolutions to address high-order spatial interaction loss.

Despite these significant advancements, these existing models face limitations when applied as an Automated Truck-Mounted Alert System. They lack the capability to detect

the distance of oncoming vehicles—a critical feature for alerting distracted drivers to potential hazards. Additionally, they suffer information loss while passing through the successive layers, leading to degradation of the model's performance. To address this gap, we propose the following: (1) A more generalized model that incorporates a distance classification module into the model. This module accurately classifies the vehicles into three zones: a safe zone (>120 m), a warning zone (60–120 m), and a danger zone (<60 m). By providing real-time distance classification alerts, this enhancement aims to improve driver awareness and response times, thereby reducing the risk of TMA collisions. (2) Generalized Efficient Layer Aggregation Network (GELAN) as a backbone. GELAN provides more stable and robust performance, using conventional convolution to achieve higher parameter usage than the depth-wise convolution design based on the most advanced technology. It offers advantages of being light, fast, and accurate. (3) An alert triggering module: This module processes the model's output and activates alarms based on the calculated speed, acceleration, and time to collision of the detected vehicles. This module is activated if the vehicles are within the danger zone or warning zone, ensuring timely warnings. A calibrated camera is used to accurately measure speed, acceleration, and time to collision.

This study contributes to the body of knowledge in the following ways:

1. Introducing an AI-enabled vision system that leverages MTL to detect and classify vehicles, perform lane and road segmentation, and determine distance categories (safe zone, warning zone, danger zone) in real-time;
2. Implementing a Generalized Efficient Layer Aggregation Network (GELAN) backbone to enhance model stability, efficiency, and accuracy, addressing the limitations of existing models;
3. Incorporating an alert triggering module that activates alarms based on speed, acceleration, and time to collision, ensuring timely warnings for vehicles in the danger zone or warning zone;
4. Additionally, our research marks the first instance of applying MTL techniques to TMA automatic audible alerts.

2. Related Work

2.1. Multi-Task Learning

Multi-task learning (MTL) in machine learning involves training a single model to perform multiple tasks concurrently. Unlike traditional approaches that train separate models for each task, MTL leverages the interconnectedness of related tasks to enhance its overall performance. The benefits of MTL are numerous. Firstly, it enhances efficiency by reducing computational costs and data requirements through joint learning. This makes MTL especially useful for deployment on edge devices, where it might not be possible to run numerous models at once due to memory limitations. MTL also boosts accuracy by leveraging shared information across related tasks, leading to improved generalization compared to single-task models. Additionally, MTL acts as a regularization mechanism by enabling the model to learn from various tasks simultaneously, mitigating the risk of overfitting. MTL offers a powerful framework that not only enhances computational efficiency and accuracy but also facilitates regularization, thereby contributing to more robust and effective machine learning models [14–16]. Key methodologies in MTL include hard parameter sharing, where layers of neural networks are shared among tasks, and soft parameter sharing, where each task has its model but parameters are regularized to remain similar [17]. MTL has wide-ranging applications across various fields, including natural language processing, computer vision, speech recognition, and biomedical imaging. In the field of computer vision, MTL has achieved significant success, especially with the creation of panoptic models. This study draws inspiration from panoptic models like YOLOP [9], YOLOP2 [10], and Hybridnets [11], which have shown impressive abilities to simultaneously handle tasks such as car detection, lane detection, and driving area segmentation.

2.2. Object Detection

Object detection methodologies are broadly categorized into two main types: two-stage detectors and one-stage detectors. Two-stage detectors employ region proposals followed by feature extraction to detect objects within specific regions. Conversely, one-stage detectors perform end-to-end object detection without region proposals. While two-stage detectors like RCNNs [18] typically offer superior performance, one-stage detectors are preferred due to their faster processing speed. An example of a one-stage detector is YOLO [19], which partitions the image into SxS grids and detects objects within each grid. YOLOv3 [20] introduced the concept of anchors—predefined boxes used within each grid to detect objects; however, tuning anchors for different datasets can be laborious. Recently, there has been a surge in anchor-free detectors, eliminating the need for anchor tuning. Examples include CenterNet [21] and YOLOv8, which have introduced anchor-free detection methods. In addition to CNNs, transformer-based models like DETR [22] have also achieved state-of-the-art results; however, their computational demands preclude their use in real-world applications such as autonomous driving.

2.3. Driving Area Segmentation

Segmentation involves the meticulous labeling of images at the pixel level, a fundamental task in computer vision. The evolution of segmentation methods has been marked by significant advancements in recent years. Initially, a Fully Connected Network (FCN) [23] was employed for segmentation tasks, laying the groundwork for subsequent developments; however, the FCN faced limitations, particularly in addressing the inherent multi-scale nature of segmentation tasks. To overcome this challenge, PSPNet [24] introduced global spatial pooling, enabling the consideration of multi-scale features. DeeplabV3 [25] further refined segmentation techniques with atrous spatial pooling, enhancing the ability to capture multi-scale features effectively. Subsequent advancements, such as SSN [26], focused on incorporating conditional random field units in post-processing to improve segmentation accuracy. ENet [27] introduced unique innovative initialization techniques and bottleneck features to enhance efficiency and performance. DDU-Net [28] adopted two decoders to enhance semantic segmentation.

2.4. Lane Detection

The narrow configuration of lanes, the fragmented pixel distribution, and occlusion constitute three distinct characteristics that provide challenges for lane segmentation. Until recently, conventional lane line identification algorithms—in particular, the Hough transform [29]—were widely used. However, the advent of LaneNet [30] marked a paradigm shift by treating individual lane lines as separate segmentation instances. Subsequently, SCNN [31] introduced a slice-by-slice convolution to enhance information propagation across channels within each layer, thereby improving segmentation accuracy. VPGNet [32] employs vanishing point guidance to detect both lane and road markings, leveraging geometric cues for enhanced detection performance. RESA [33] adopts a recurrent approach, iteratively shifting sliced feature maps in both vertical and horizontal directions to capture global pixel information effectively. Enet-SAD [34], on the other hand, makes use of a self-attention rectification technique to facilitate the learning of low-level features from high-level characteristics, thereby improving performance without compromising the model's lightweight architecture.

Independent lane detection, lane segmentation, and drivable area detection face limitations such as high computational costs, inconsistent results, increased latency, and resource inefficiency. MTL addresses these issues by sharing feature extraction layers, which reduces computational load and memory usage, improves consistency across tasks, lowers latency by processing tasks in parallel, and enhances overall performance by leveraging the interrelatedness of tasks.

3. Methodology

Introduction

The methodology is divided into three main parts: data collection and processing, model training and evaluation, and the alert triggering system. These parts are illustrated in Figure 1. Data collection and processing involves gathering video and point cloud data using LiDAR and processing it for model input. Model training and evaluation focuses on training the model to output bounding box coordinates, distance classifications (safe zone, warning zone, danger zone), lane segmentation, and road segmentation. The alert triggering system triggers alarms based on speed, acceleration, and time-to-collision data from a calibrated camera. The details of these parts are discussed in the sections that follow.

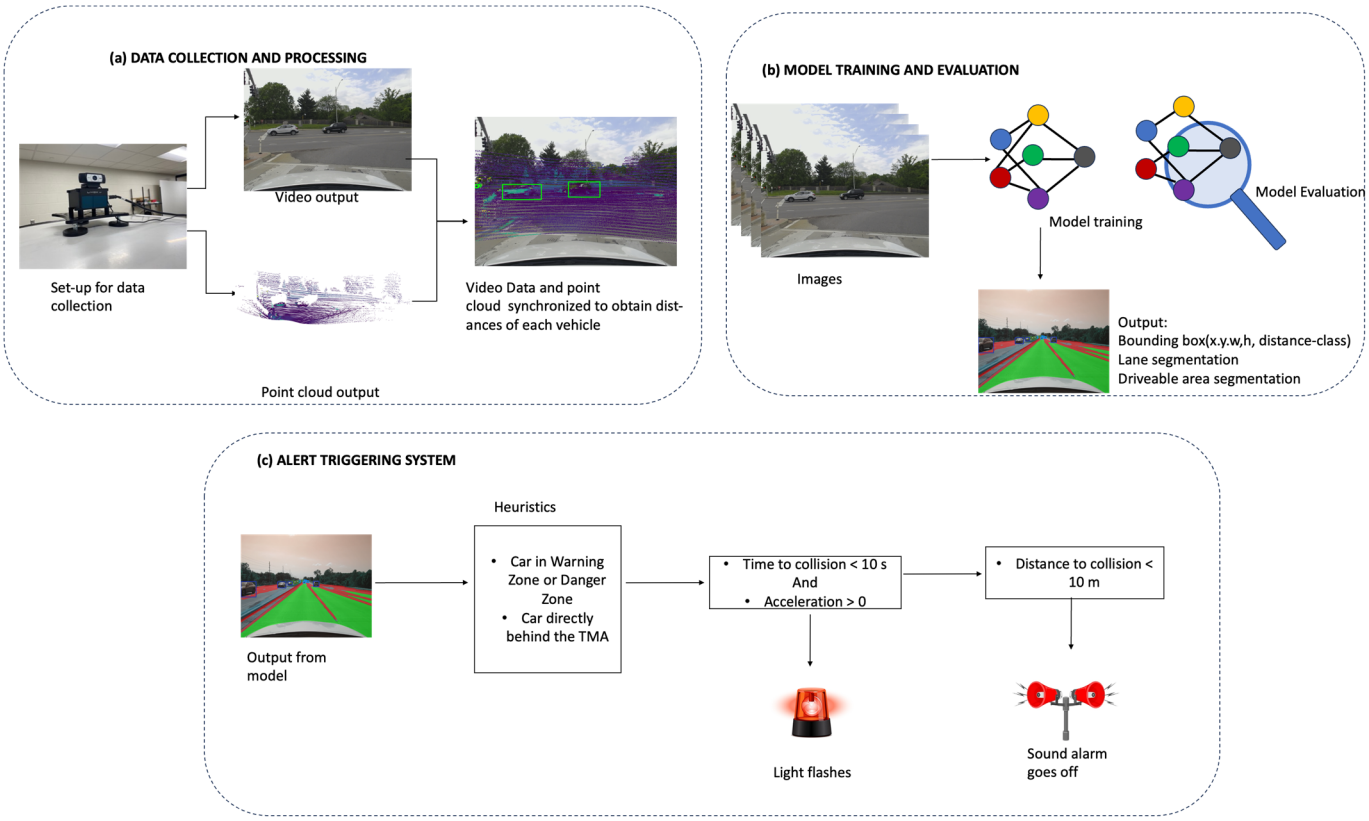

Figure 1. Methodology.

4. Data Collection and Processing

4.1. Sensor Setup and Data Collection

Figure 2 showcases the data acquisition setup, featuring a high-definition 1080p Logitech webcam (Logitech, HongKong, China) mounted above a Livox LiDAR sensor (Livox, Shenzhen, China), both affixed to the top of a vehicle. The Livox LiDAR HAP (T1) employed in this setup boasts a 120° horizontal field of view (FOV) and a 25° vertical field of view. This LiDAR is capable of detecting objects up to a range of 150 m with 10% reflectivity, providing comprehensive 3D spatial information. A Logitech HD 1080 webcam was used for video streaming. This webcam offers a 78° diagonal field of view, capturing high-definition video footage. To prevent any obstruction of the LiDAR's view, the webcam was strategically mounted on top of the LiDAR sensor. This vertically aligned configuration ensures that both sensors have an unobstructed field of view, enabling them to simultaneously capture complementary data streams.

Figure 2. Data acquisition setup: a high-definition 1080p Logitech webcam mounted above a Livox LiDAR with a 120-degree horizontal field of view, both installed on top of a vehicle to gather spatial and video data.

The Logitech webcam was selected for its sharp 1080p HD resolution, outstanding performance in low-light settings, autofocus capability, and automatic light adjustments. Logitech is a reputable brand in the webcam market, known for delivering durable, high-quality products that are both user-friendly and competitively priced compared to other brands. The Livox HAP was chosen for its ability to endure harsh environments while maintaining reliable performance. It offers a detection range of 150 m, which is ideal for calibrating our model, and generates a dense point cloud with a unique non-repetitive scanning pattern, enhancing point cloud resolution over time. Compared to other premium LiDAR systems, the HAP is also more cost-efficient.

The entire sensor assembly was securely installed on the vehicle, as depicted in Figure 2. This mobile setup was employed to gather data from a diverse range of environments, such as urban streets and highways. The chosen data collection routes included Providence, Grindstone Parkway, Broadway, and I-70W in Columbia, Missouri, covering both urban and freeway driving conditions. This variety ensures a robust dataset that captures different driving conditions and environments.

The Logitech camera recorded video at a resolution of 640 $\times$ 480 pixels, operating at a frame rate of 30 frames per second (FPS). Meanwhile, the Livox LiDAR operated at a frame rate of 10 Hz, capturing spatial data ten times per second.

4.2. Data Processing

In our data processing pipeline, the integration of camera and LiDAR data is critical for accurate analysis and application; however, the differing frame rates and spatial perspectives of these sensors present challenges. To address these, we performed two key steps: data synchronization and data calibration. These steps ensure that the data from both sensors are temporally and spatially aligned, enabling precise and meaningful integration.

4.3. Data Synchronization

The first step in our pipeline was to achieve temporal alignment between the camera images and the LiDAR point clouds. Given that the camera operates at a lower frame rate (FPS) than the LiDAR, it was necessary to synchronize the data from both sensors. We accomplished this by matching each camera image to the LiDAR point cloud with the closest timestamp. This synchronization ensures that the data from both sources correspond to the same moment in time, laying the foundation for accurate subsequent processing.

4.4. Data Calibration

After synchronization, the next step was to achieve spatial alignment between the camera and LiDAR data through a process known as data calibration. To achieve accurate spatial alignment, the point cloud data were calibrated using the natural edges present

in the scene utilizing code developed by Yuan et al. [35]. Natural edges refer to the distinct boundaries or transitions in an image, such as the edges of objects, where there is a significant change in color or intensity. These edges on both LiDAR data and the camera data are detected, allowing them to serve as common reference points for alignment. By identifying and matching these natural edges detected by both the LiDAR and the camera, these two datasets were accurately aligned.

For these edges in both the camera images and the LiDAR point clouds, we established common reference points for alignment. Using these reference points, we computed multiple extrinsic matrices that represent the transformation from the LiDAR frame to the camera frame. These matrices encapsulate the rotation and translation necessary to map points from the LiDAR coordinate system to the camera coordinate system.

Given the inherent noise and potential inaccuracies in individual calibration results, we computed the median of these extrinsic matrices. The median matrix provides a robust estimate of the transformation by mitigating the influence of outliers and ensuring a more reliable mapping between the two coordinate systems.

Once the robust transformation matrix was established, we applied it to the 3D LiDAR points, which include the x, y, and z coordinates. This transformation repositions the LiDAR points into the camera's coordinate frame. Subsequently, we projected these transformed 3D points onto the 2D image plane using the camera's intrinsic matrix. The intrinsic matrix accounts for the camera's focal length, optical center, and other internal parameters essential for accurate projection.

Lens distortion, which can cause significant deviations in the projected points, was corrected using distortion coefficients specific to the camera. These coefficients adjust the projected points to account for barrel or pincushion distortion, ensuring the points accurately reflect the true scene's geometry. Given the different fields of view (FOV) of the camera and the LiDAR, not all projected points fall within the camera's image plane. Points that lie outside the image boundaries were discarded as they do not correspond to any valid pixel in the image. This step ensures that only relevant points, which have a corresponding pixel in the image, are retained for further processing.

Through these steps, we effectively synchronized and calibrated the LiDAR and camera data, enabling accurate and meaningful integration of the two data sources for subsequent analysis and applications. Figure 3 shows the point cloud data that has been synchronized and calibrated with the image data.

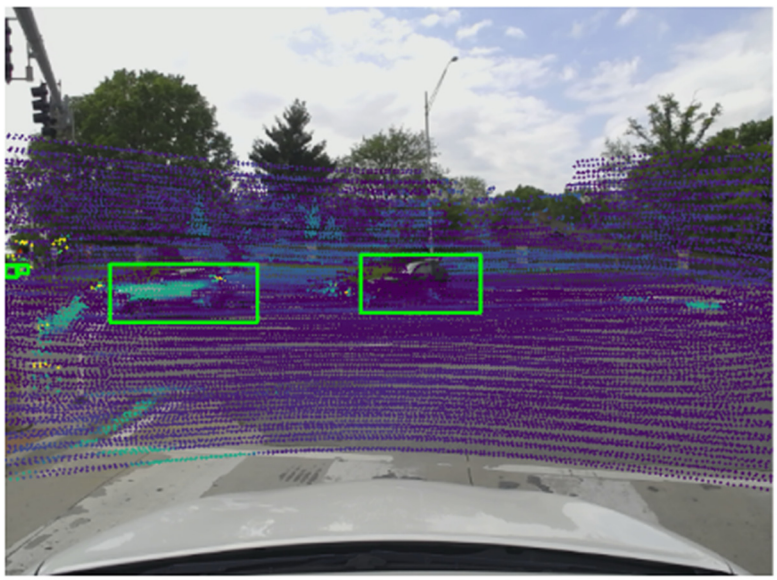

Figure 3. Image data synchronized and calibrated with point cloud data.

4.5. Ground Truth

YOLOPX was utilized to generate ground truth data. YOLOPX is an anchor-free, multi-task learning network specifically designed for panoptic driving perception. It excels in detecting traffic objects, segmenting drivable areas, and identifying lanes. YOLOPX was selected for its real-time object detection capabilities, optimized performance on edge devices, and excellent balance between accuracy and speed compared to other models. YOLOPX's outputs include the bounding box coordinates of the vehicle, lane segmentation, and road segmentation. For distance ground truth creation, the synchronized point cloud data were used. The point cloud distances within each bounding box were obtained, and the 75th percentile of these distances was calculated to determine the car's distance. This distance was then categorized into three distinct ranges: "danger" (0–60 m), "warning" (60–120 m), and "safe" (greater than 120 m). This categorization helps in assessing the safety levels and potential risks posed by oncoming vehicles. The training dataset comprised 6757 images, while the validation dataset included 1691 images (80:20 ratio). For testing, 3825 images were used, all sourced from a video clip that was not part of the training or validation sets, ensuring the model had not previously encountered them.

5. Model Training and Evaluation

5.1. Model Network

The model comprises a shared encoder along with three distinct decoders, illustrated in Figure 4. The encoder initially processes images with a feature size HXWX3. This encoder is composed of both a backbone and a neck, which collectively extract features from the input images. Subsequently, the individual decoders execute specialized tasks, including object detection, distance classification, lane detection, and drivable area segmentation.

5.2. Encoder

The primary function of the encoder is to extract features from images. Our selection of the GELAN (Generalized Efficient Layer Aggregation Network) backbone is grounded in its capability to capture intricate patterns and contextual information within images, qualities that are essential for our decoder heads. Moreover, it addresses issues such as gradient vanishing, thereby improving training stability. Additionally, it strikes a balance between accuracy and computational efficiency, as noted in YOLOv9 [36]. GELAN backbone splits the convolutional layer into two paths and processes each and then merges them back. The dual strategy facilitates efficient gradient flow and feature reuse, which enhances the model's learning efficiency and inference speed by ensuring depth without the computational penalty associated with the increased complexity. These qualities are vital for the TMA alert system, ensuring that the model can effectively capture information, detect vehicles and lanes, and accurately determine distances. The encoder's neck serves the purpose of feature aggregation and refinement, consolidating and refining features from different scales across the network. By combining various levels, it enhances the model's ability to capture both the high-level and low-level features of the image. Various modules have been used in the necks of networks, including SPP (Spatial Pyramid Pooling), FPN (Feature Pyramid Network), and Bi-FPN (Bi-directional Feature Pyramid Network). SPP divides up different feature maps into spatial bins, independently pooling features from these bins, which are then concatenated to form a multi-scale representation (cite). In our network, we have adopted the SPPELAN introduced in YOLOv9, this module incorporates Spatial Pyramid Pooling within the ELAN structure, hence capturing multi-scale contextual information while maintaining computational efficiency.

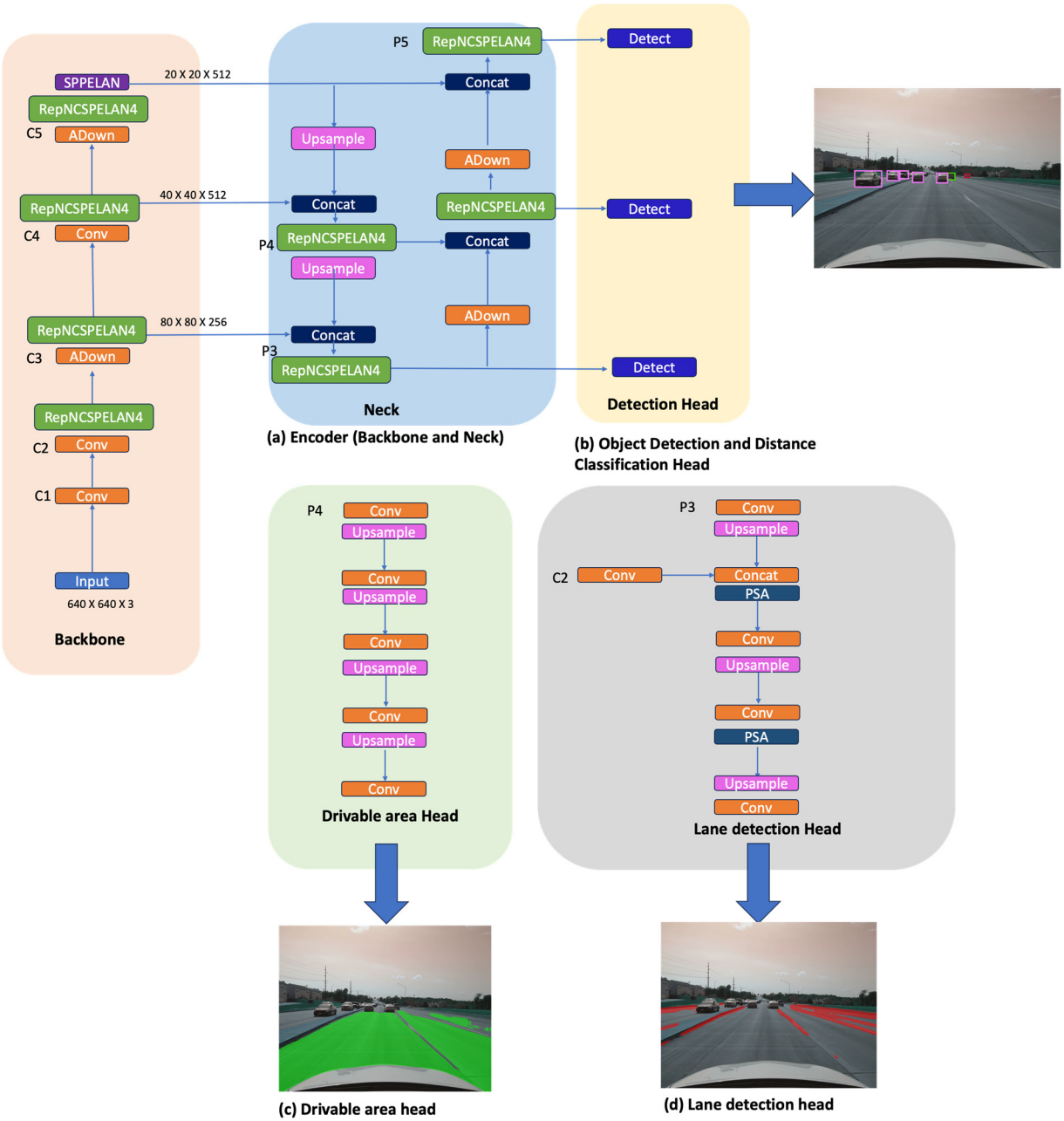

Figure 4. TMA model.

5.3. Detection Head

The detection head is an anchor-based, multi-scale detection head that employs the Path Aggregation Network (PAN) to process a bottom-up transfer of the features. Features from the bottom-up and top-down pathways are concatenated to enhance the detection capabilities. The multi-scale detection heads predict the following: (1) the bounding box coordinates, specifying the position and dimensions (x, y, width, and height) of detected objects; (2) classifying the objects within these bounding boxes; (3) the probability of the predicted class, providing a confidence score for each classification; (4) lastly, they predict the distance category of a detected car. The distance prediction categorizes the car's proximity into three distinct ranges. These are "danger", which corresponds to a distance of 0–60 m; "warning", for a range of 60–120 m; and "safe", for distances greater than 120 m. This categorization helps in assessing the safety levels and potential risks posed by oncoming vehicles to the TMA. If a vehicle falls within a risky distance range,

an alert triggering system is activated to alert the motorist during the risk of the TMA being hit.

5.4. Lane Detection and the Drivable Area Segmentation Decoder

Lane detection and drivable area segmentation differ in their attributes: lane detection segments small linear lines, while drivable area segmentation segments larger areas. Unlike in YOLOP, Hybridnets, and Ehisnet, where features for both decoders (the lane and drivable area) stem from the same feature head, we extract features for these decoders from distinct layers to prevent mutual interference. Drawing inspiration from YOLOPX [12], we utilize lower features of P4 for the drivable areas as shown in Figure 4c, which we then restore to the H × W size through four up-samplings.

The lane detection head (Figure 4d), also influenced by YOLOPX, acquires high-level features C2 and integrates them with an up-sampled P3 that contains multi-scale information. Subsequently, these features are restored to the H × W size using three up-samplings. A Polarized Self-Attention mechanism is employed to separate the direction and magnitude of the features, establishing long-distance contextual dependencies. Overall, our approach optimizes both accuracy and efficiency in lane detection and drivable area segmentation tasks.

5.5. Loss

Our model has multiple heads, therefore, we employ a multi-task loss. The overall loss of the model is given by the following:

$$\mathcal{L} = \alpha_{det}\mathcal{L}_{det} + \alpha_{daseg}\mathcal{L}_{daseg} + \alpha_{llseg}\mathcal{L}_{llseg} \tag{1}$$

where $\mathcal{L}_{det}$ is the detection head loss; $\mathcal{L}_{daseg}$ is the drivable area segmentation loss; $\mathcal{L}_{llseg}$ is the lane detection loss; α_{det}, α_{daseg}, α_{llseg} are tuning parameters to balance the total loss; and α_{det} is 1, α_{daseg} is 0.2, and α_{llseg} is 0.2—these are the same values used in YOLOP.

The detection head loss is given by the formula below:

$$\mathcal{L}_{det} = \alpha_{cls}\mathcal{L}_{cls} + \alpha_{obj}\mathcal{L}_{obj} + \alpha_{box}\mathcal{L}_{box} + \alpha_{dist}\mathcal{L}_{dist} \tag{2}$$

where $\mathcal{L}_{cls}$ is a classification loss; $\mathcal{L}_{dist}$ is the distance classification loss; $\mathcal{L}_{obj}$ is the object loss; and $\mathcal{L}_{cls}$, $\mathcal{L}_{dist}$, $\mathcal{L}_{obj}$ are all Binary Cross Entropy losses, which is a binary variant of the cross entropy loss—this loss is suitable because each object class is treated independently; $\mathcal{L}_{box}$ is the box loss, in which the Complete Intersection Over Union addresses localization and the size of the predicted bounding box; α_{cls} is tuned to 0.5, α_{dist} is tuned to 1, α_{box} is tuned to 0.05 and α_{obj} is tuned 1.

For the drivable area, cross entropy loss was used to minimize the pixel level classification error.

$$\mathcal{L}_{daseg} = \mathcal{L}_{ce} \tag{3}$$

For the lane detection, we employed a combination of cross entropy loss and IoU loss.

$$\mathcal{L}_{llseg} = \mathcal{L}_{ce} + \mathcal{L}_{iou} \tag{4}$$

$$\mathcal{L}_{iou} = 1 - \frac{TP}{TP + FP + FN} \tag{5}$$

$$\mathcal{L}_{ce} = \sum_{i} y_i \log(p_i) \tag{6}$$

where TP = true positives; FP = false positives; FN = false negatives; y_i is the true label (0 or 1); and p_i is the predicted probability of the class.

5.6. Training

The TMA model was trained on an Nvidia RTX GeForce 3090 (Nvidia, Santa Clara, CA, USA)for 30 epochs, starting from pretrained YOLOPX weights. The initial learning rate was set to 0.001, with a minibatch size of 16. A 3-epoch warm-up period was implemented, and the learning rate was adjusted using cosine annealing. The AdamW optimizer was utilized for training.

5.7. Alarm Triggering System

Figure 5 and Algorithm 1 outline the steps involved in the real-time operation of the alert triggering system. The system receives a video livestream as input, which the model processes through three decoder heads: drivable area, lane detection, and object detection with distance classification. The output includes lane segmentation, drivable area segmentation, vehicle detection, and distance classification, categorizing vehicles into danger zone, warning zone, or safe zone. The livestream is captured by a calibrated camera, which accurately determines vehicle distances. The alert system uses a "follower box" to track vehicles within two critical zones in the same lane as the TMA: the danger zone (0–60 m) and the warning zone (60–120 m). With the known distances, the system calculates vehicle speeds and time to collision. If a vehicle moves too close to the TMA, visual and auditory alarms are triggered to warn drivers, thereby enhancing the safety of road maintenance crews. The following sections provide more details on this process.

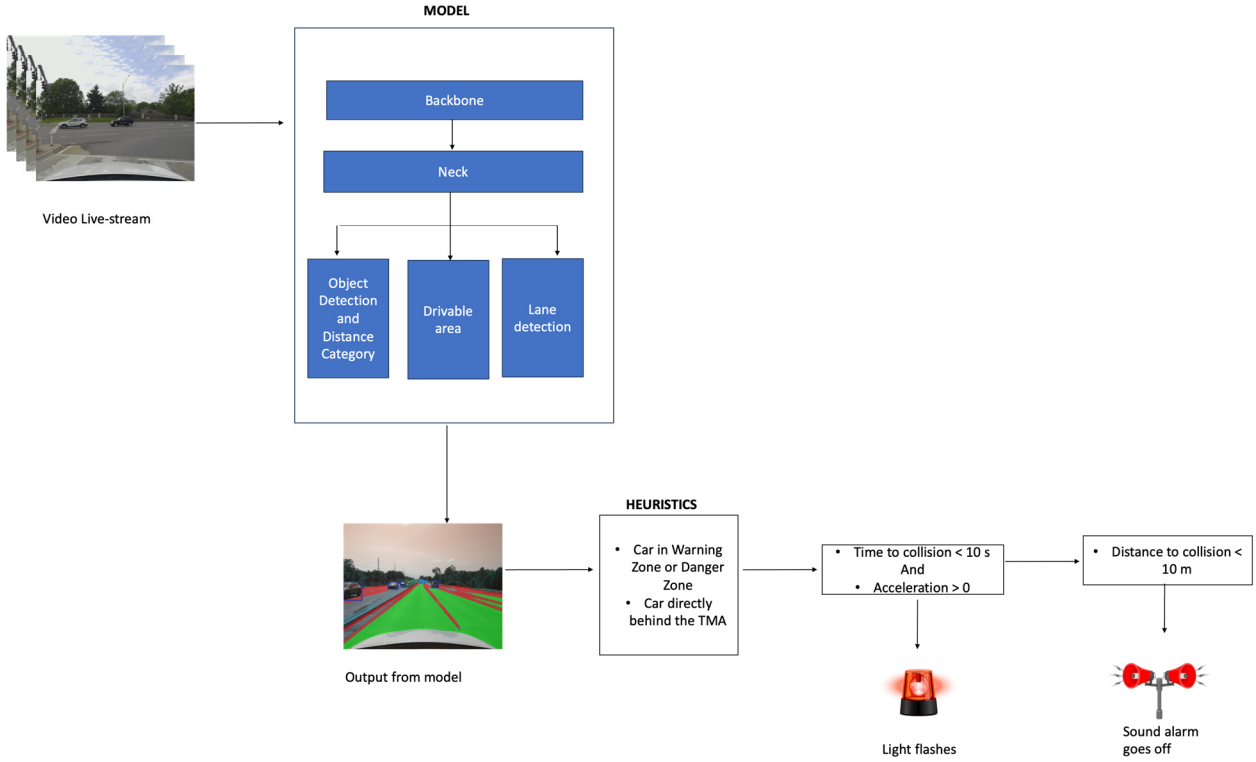

Figure 5. Alarm triggering system.

5.8. Determining Vehicles behind the TMA ("Follower Box")

To trigger the alarm in our alert system, the heuristics code monitors vehicles within the danger zone (0–60 m) and warning zone (60–120 m) that follow behind the TMA in the same lane, disregarding vehicles in other lanes. This is accomplished using a "follower box" to identify the relevant lane. The process of drawing the follower box is semi-automated. The Python code calculates the distances between continuous lane markings and selects those with the largest distance between them as these are usually behind the TMA when the camera is positioned at its rear; however, this method may occasionally be inaccurate,

requiring manual adjustment by dragging the follower box to the correct lane markings behind the TMA. The box is then aligned within the lane where the TMA is moving. This setup allows the alarm system to focus on cars in the same lane as the TMA, as illustrated in Figure 6.

Algorithm 1. Alarm Triggering System

```
let D ← []
let S ← []
let A ← []
for frame_i in livestream:
    detected_vehicles, distance_category, drive_area, lane_segment ← TMA.Model(frame_i)
    if distance_category = "Safe Zone":
        None
    elif distance_category = "Danger Zone" or "Warning Zone":
        l_d = Calculate distance between two consecutive lane_segmentation
        selected_lane = max(l_d)
        followerBox ← Draw a follower box in-between selected lane
        for each corner (bottom-right, top-right, bottom-left, top-left) of followerBox:
            xsp, ysp = Calculate intersection with nearest lane line from the selected lane
        lane_line_equation ← Derive lane line equation for left and right lanes
        if detected_vehicle lies within lane_line_equation:
            vehicle_following_TMA ← True
        else:
            vehicle_following_TMA ← False
        if vehicle_following_TMA = true:
            distance_from_TMA ← calibrated_camera(detected_vehicles)
            D ← append(distance_from_TMA)
            speed = distance_from_TMA / time
            S ← append(speed)
            acceleration = speed / time
            A ← append(acceleration
            if len(D) = 100:
                time_to_collison = D[100] / S[100]
                if A [100] > 0 and time_to_collision < 8.5:
                    light_alarm = True
                else:
                    light_alarm = None
            if D [100] < 10:
                sound_alarm = True
            else:
                sound_alarm = False
            let D ← [ ], A ← [ ], S ← [ ]
        else:
            None
return light_alarm, sound_alarm
```

To identify vehicles moving in the same lane as the TMA, the code first determines the lane lines surrounding the TMA's lane using the follower box. The process begins by identifying the bottom-right corner of the box and calculating the horizontal intersection with the nearest lane line, recording this point. A similar calculation is performed from the top-right corner to obtain a second point on the right lane line. This procedure is repeated for the bottom-left and top-left corners of the box to identify points on the left lane line. Using these points, the system derives the equations for the right and left lane lines. These equations are useful when the detected lane lines do not extend far enough to the car. By

extending the lane lines using the derived equations, the system can determine if the car's bottom center lies between these lines, as shown in the image above.

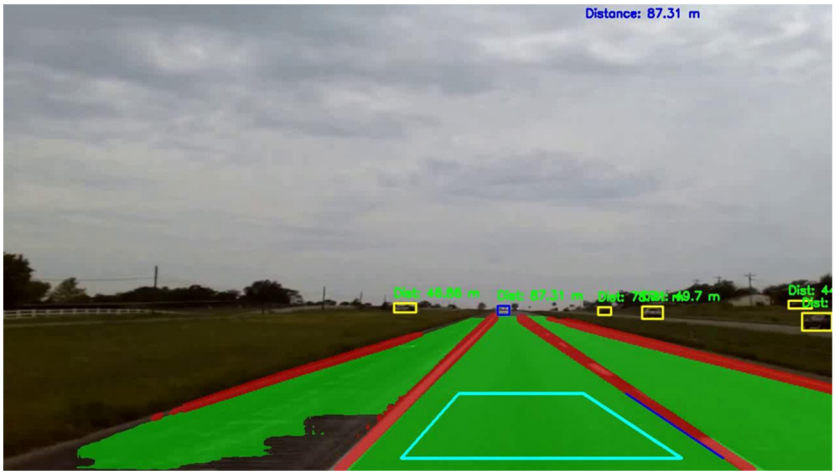

Figure 6. Follower box in cyan.

To confirm if a car is within the lane lines, the system checks if its bottom center falls between the right and left lane lines. It calculates the horizontal intersection of the car's bottom center with the right lane line and records this position. The same calculation is performed for the left lane line. If the car's bottom center lies between these two positions, and the car is within the danger zone and warning zone, it is marked as following the TMA. This scenario is depicted in Figure 7.

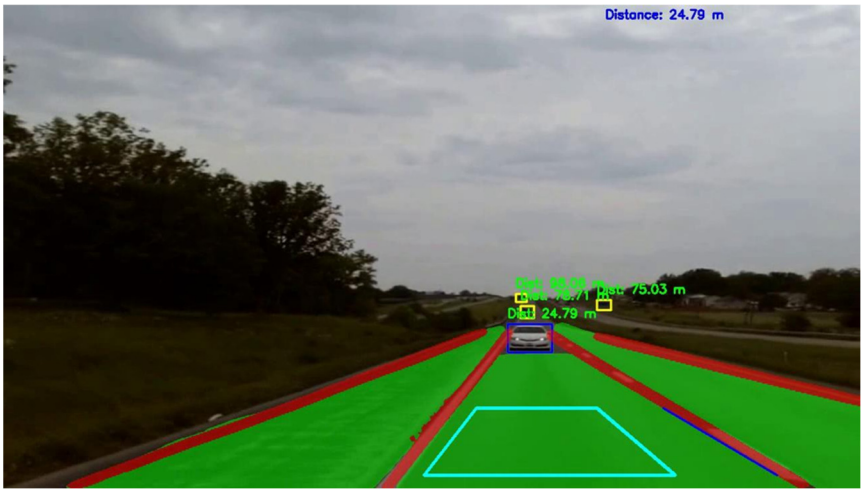

Figure 7. Car within danger zone following the TMA.

5.9. Camera Calibration and LiDAR Correction

Once a vehicle is detected, it is tracked, and the distance from the TMA is calculated using a calibrated camera. This distance is used to determine parameters such as speed, acceleration, and time to collision. The camera is calibrated using chessboard images captured from various angles, processed with OpenCV's camera calibration code to generate the camera calibration matrix. However, since this calibration does not yield accurate distances, LiDAR is employed to obtain true distance values. These true distances are then used to correct the camera calibration results.

5.10. Speed, Acceleration, and Time to Collision

Distances are recorded and maintained in a window of the past 100 values, with the current distance being the average of these values. Speed is calculated by taking the

difference between the first and last distances in this window and dividing by the time difference. Similarly, a window of the past 100 speed values is maintained to calculate acceleration by taking the difference between the first and last speed values and dividing by the time difference. Time to collision is determined by dividing the distance by the speed at each point.

5.11. Alarm Triggering

The system features two alarms: a light alarm and a sound alarm. The light alarm alerts drivers approaching the TMA with acceleration, triggered when the time to collision is less than 8.5 s and acceleration is greater than zero. If the driver does not heed the light alarm and comes dangerously close to the TMA, the sound alarm is triggered when the vehicle is within 10 m of the TMA to alert the driver.

6. Results and Discussion

6.1. Overall Model Performance

The TMA model demonstrates exceptional performance across various tasks crucial for enhancing work zone safety. The model achieved a recall of 90.5%, a mean Average Precision (mAP) of 0.792 for vehicle detection, a mean Intersection over Union (mIOU) of 0.948 for road segmentation, an accuracy of 81.5% for lane segmentation, an IOU of 0.711 for lane segmentation, and an accuracy of 83.8% for distance classification, as shown in Table 1. These results underscore the model's robust capability in detecting vehicles, accurately determining their distances, and providing real-time alerts.

Table 1. TMA model's performance.

Objection detection		Lane detection		Driveable area	Distance classification
Recall	mAP50	Pixel accuracy	IOU	mIOU	
0.905	0.792	0.815	0.711	0.948	81.5%

Figure 8 showcases the output of the TMA model, highlighting its ability to detect vehicles, classify distances, and segment lanes and roads. In the images, the danger zone is represented in pink, the warning zone in cyan, and the safe zone in yellow, clearly demonstrating the model's proficiency in distance assessment.

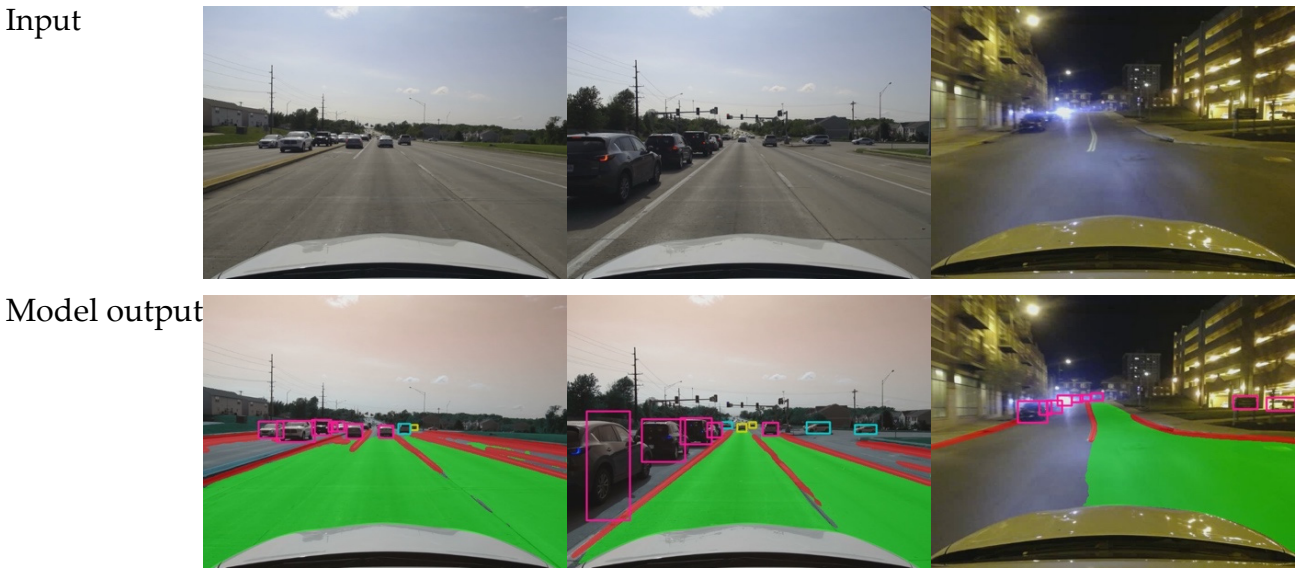

Figure 8. *Cont.*

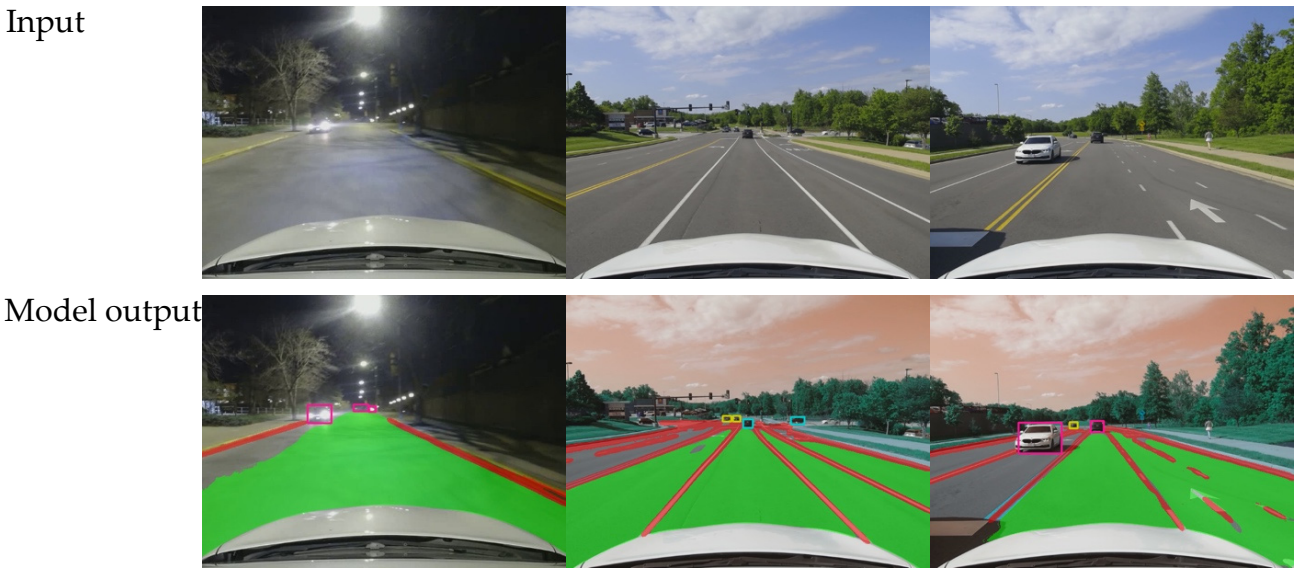

Figure 8. TMA model's output.

6.2. Distance Detection Accuracy

The model's accuracy in detecting the distance of oncoming vehicles was evaluated across three distance ranges and the accuracy as shown in Table 2 are as follows:

- 0–60 m: the model achieved an accuracy of 0.94;
- 60–120 m: the accuracy dropped to 0.58;
- Over 120 m: the model had an accuracy of 0.73.

Table 2. Accuracy across different classes.

	Zone	Distance	Accuracy
1	Danger zone	0–60 m	0.94
2	Warning zone	60–120 m	0.58
3	Safe zone	>120 m	0.73

These results indicate that the model is highly accurate at shorter distances but faces challenges as the distance increases.

6.3. Comparison with Existing Models

The performance of the TMA model was benchmarked against several existing models, including YOLOP, YOLOPv2, Hybridnets, and Ehisnet, focusing on key tasks such as object detection, lane detection, drivable area segmentation, and key metrics like recall, mean Average Precision at 50% Intersection over Union (mAP50), pixel accuracy, Intersection over Union (IOU), and mean Intersection over Union (mIOU). Figure 9 illustrates that the TMA model excels across these tasks, particularly in detecting vehicles, classifying distances, and segmenting lanes and roads, consistently maintaining high accuracy across all parameters.

Our choice of the GELAN backbone is a key factor in the model's performance. The GELAN backbone efficiently captures intricate patterns and contextual information within images—critical for the success of the multi-task learning employed in the TMA model. This network architecture solves issues such as gradient vanishing, improving training stability, and allows for better feature extraction, particularly useful for tasks like lane detection and road segmentation, where capturing fine details is essential. Unlike simpler backbones used in the other models, GELAN utilizes a dual-path strategy, splitting convolutional layers into two paths and merging them, which facilitates better gradient flow and feature

reuse. This strategy enhances learning efficiency and improves inference speed without the computational penalties typically associated with increased model complexity. Such an approach ensures that the TMA model achieves a better balance between accuracy and computational efficiency, surpassing models like YOLOP and Hybridnets in performance while maintaining speed, which is critical for real-time applications

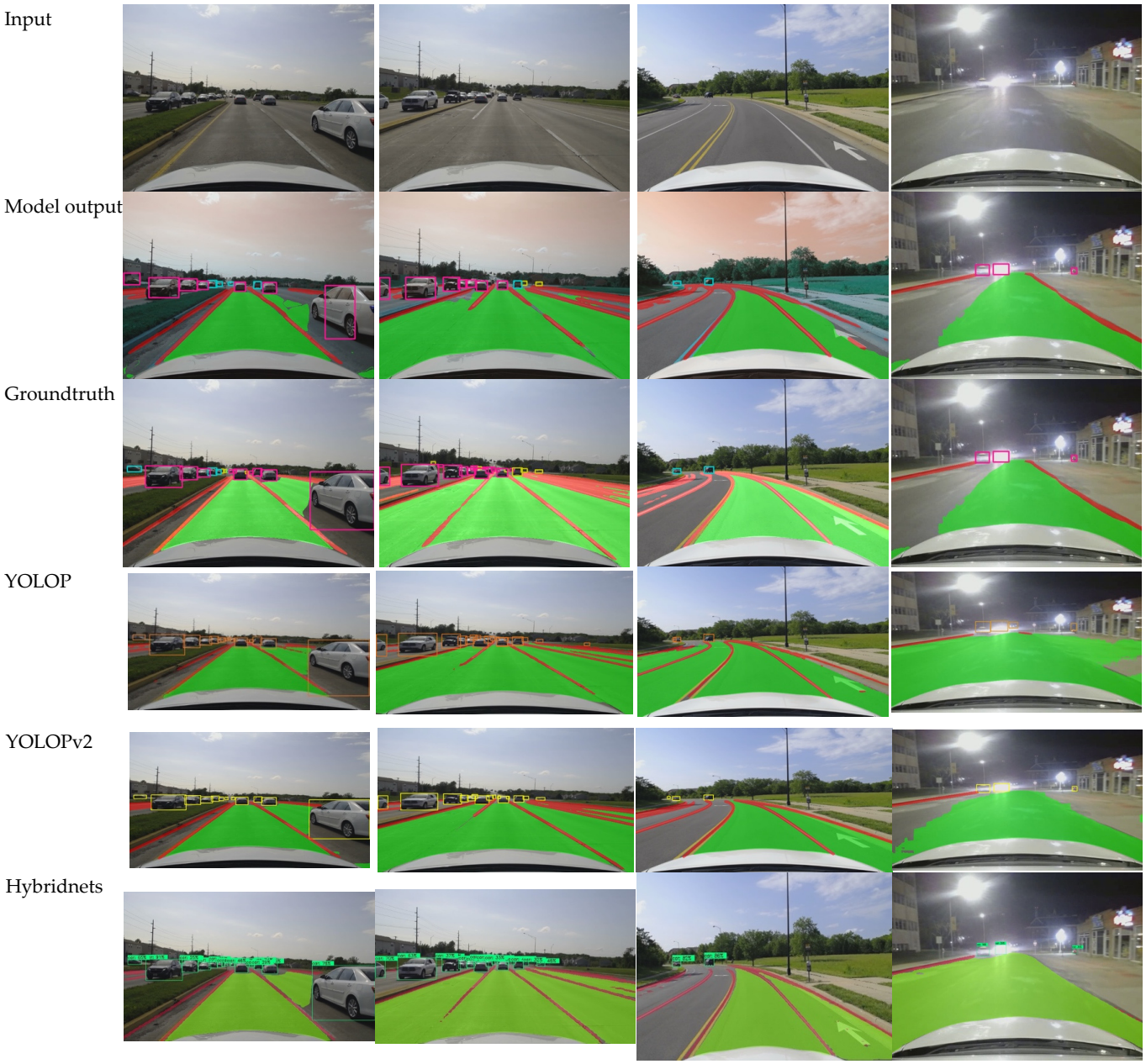

Figure 9. TMA model's comparison with other models.

6.4. Object Detection

Table 3 highlights that the TMA model achieves a recall of 0.905 and an mAP50 of 0.792, which is competitive with existing models. While Ehisnet outperforms in recall with 0.923, the TMA model maintains a strong balance between recall and mAP50, showcasing its reliability in object detection tasks. Notably, incorporating distance classification into the model does not significantly impact its performance, indicating the model's robustness and efficiency. This can be attributed to the GELAN backbone's ability to capture more detailed features, making it better suited for distinguishing vehicles in work zones.

Table 3. Performance comparison on object detection.

Object Detection		
	Recall	**mAP50**
TMA model	0.905	0.792
YOLOP	0.915	0.791
Hybridnets	0.845	0.688
Ehisnet	0.923	0.811

6.5. Lane Detection

Table 4 shows the TMA model's superiority in lane detection, with a pixel accuracy of 0.815 and an IOU of 0.711. It outperforms YOLOP, Hybridnets, and Ehisnet, demonstrating its capability to accurately segment lanes. The TMA model uses its enhanced feature extraction and segmentation capabilities to achieve higher accuracy and mIOU in challenging conditions, such as work zones with inconsistent lane boundaries.

Table 4. Performance comparison on lane detection.

Lane Detection		
	Pixel Accuracy	**IOU**
TMA model	0.815	0.711
YOLOP	0.575	0.558
Hybridnets	0.77	0.532
Ehisnet	0.652	0.634

6.6. Drivable Area Segmentation

Table 5 highlights the TMA model's excellent performance in drivable-area detection, with an mIOU of 0.948, surpassing YOLOP, Hybridnets, and Ehisnet.

Table 5. Performance comparison on drivable area segmentation.

Driveable Area Segmentation	
	mIOU
TMA model	0.948
YOLOP	0.91
Hybridnets	0.931
Ehisnet	0.926

6.7. Inference Speed

Table 6 highlights a comparison of the size and speed (FPS) of the TMA model against existing models. The TMA model strikes an optimal balance with a model size of 22.27 M and a processing speed of 58.2 FPS on an Nvidia RTX GeForce 4080, outperforming other models in speed while maintaining a competitive model size. For example, YOLOPx, while larger at 32.9 M, achieves a slower processing speed of 50.7 FPS, and Hybridnets, though smaller in size at 12.8 M, processes at only 32 FPS.

Table 6. Inference speed.

Model's Speed		
	Model Size	**Speed**
Yolopx	32.9 M	50.7 FPS
TMA model	22.27	58.2 FPS
Ehsinet	12.81	46.5 FPS
Hybridnets	12.8 M	32 FPS

However, deploying the TMA model on edge devices such as the Nvidia Jetson reveals more about its real-world performance under resource-constrained conditions. On the Nvidia Jetson, the inference speed drops to 12 FPS, which, although slower than high-end GPUs, remains efficient for work zone safety applications. Given the Jetson's limited computational power compared to desktop GPUs, achieving 12 FPS demonstrates the model's robustness and adaptability to lower-resource environments while still providing adequate performance for safety-critical tasks.

The comprehensive evaluation and comparison demonstrate that the TMA model excels in key performance metrics, including object detection, lane detection, and drivable area detection, while also achieving a high processing speed and efficient model size. These advancements highlight the TMA model's potential to significantly enhance real-time work zone safety applications.

6.8. Limitations

Real-world deployment of the proposed AI-enabled vision system could face several challenges that impact its performance, especially when applied to diverse environments beyond the controlled conditions of the original training dataset. Since the model was trained primarily on U.S. roads, it may struggle to accurately detect and classify vehicles or road features in other countries with different traffic rules, road designs, or vehicle types. For example, regions with very hilly or mountainous terrains, sharp curves, or narrow roads may present difficulties, as these conditions were not fully represented in the dataset. Differences in vehicle types and sizes, such as motorcycles, tuk-tuks, or large trucks that are more common in other regions, could pose further detection and classification challenges, as the model may be biased toward vehicles common in the U.S. To ensure reliability and scalability across diverse geographies, future research should focus on training the model on more comprehensive and diverse datasets, including a variety of road types and vehicle types from different countries. Incorporating driving data from various regions would improve the robustness of the system. Another important measure would be deploying adaptive algorithms that allow the system to dynamically learn and adjust to new environments in real-time.

6.9. Broader Impact and Application

The AI-enabled vision system using multi-task learning (MTL) developed in this study offers significant potential for improving road safety and autonomous driving beyond its application to Truck-Mounted Attenuators (TMAs). By efficiently detecting vehicles, classifying distances into danger zones, and performing lane and road segmentation, this system can enhance Advanced Driver Assistance Systems (ADAS), providing real-time alerts for collision avoidance and lane-keeping in passenger vehicles. Secondly, the AI-enabled vision system could be applied to smart crosswalks where it would detect oncoming vehicles and estimate their speed and distance. The system could automatically trigger warning lights or alerts for both drivers and pedestrians when a potential collision is detected. This could be particularly useful in school zones, busy city intersections, or areas with heavy foot traffic. Similar technology could be applied to bicycle lanes, where the system would detect vehicles encroaching into bike lanes or moving dangerously close

to cyclists. It could trigger visual and audio alerts for both cyclists and drivers to avoid potential accidents, contributing to safer shared roads. The system could be integrated into V2X communication platforms, allowing vehicles to share real-time safety information with nearby vehicles and infrastructure. For instance, when a vehicle equipped with this system detects an imminent collision or road hazard, it could broadcast alerts to surrounding vehicles or traffic management systems, improving overall situational awareness on the road and thus improving overall road safety and traffic management.

7. Conclusions

This study presents the development and evaluation of an AI-enabled vision system designed to enhance work zone safety by automatically triggering alerts for drivers on a collision course with Truck-Mounted Attenuators (TMAs). The methodology encompassed three key components: data collection and processing, model training and evaluation, and an alert triggering system.

The data collection process utilized synchronized video and point cloud data from a high-definition webcam and Livox LiDAR sensor mounted on a vehicle, ensuring comprehensive coverage of various driving environments. The model, based on the Generalized Efficient Layer Aggregation Network (GELAN) backbone, was trained to perform multiple tasks including object detection, lane segmentation, road segmentation, and distance classification. The training was conducted using an Nvidia RTX GeForce 3090—starting from pretrained YOLOPX weights—and utilized advanced techniques such as cosine annealing and the AdamW optimizer.

The TMA model demonstrated outstanding performance across all tasks, achieving a recall of 90.5%, an mAP of 0.792 for vehicle detection, an mIOU of 0.948 for road segmentation, an accuracy of 81.5% for lane segmentation, and an accuracy of 83.8% for distance classification. The model maintained high accuracy in distance detection, particularly at shorter ranges, and was able to provide real-time alerts effectively.

Comparative analysis with existing models such as YOLOP, Hybridnets, and Ehisnet highlighted the TMA model's superior performance in key metrics, including object detection, lane detection, and drivable area segmentation, while also achieving faster processing speeds and maintaining an efficient model size. The incorporation of a distance classification module did not significantly affect the overall model performance, showcasing the model's robustness and efficiency.

The comprehensive evaluation demonstrates that the TMA model is a significant advancement over existing models, offering enhanced accuracy, reliability, and real-time processing capabilities. These enhancements highlight the TMA alert system's potential to greatly enhance work zone safety by providing timely and precise alerts to prevent collisions and reduce associated costs, such as human injuries, fatalities, vehicle damage, and traffic disruptions. This study contributes to the field by presenting a novel application of multi-task learning (MTL) techniques for TMA automatic audible alerts, setting a new benchmark for work zone safety technologies.

8. Future Directions

To build on the current work, future research could focus on enhancing the model's robustness across diverse environmental conditions, such as including a variety of road types and vehicle types from different countries. Integrating additional data sources, such as radar or thermal imaging, could further enhance the system's reliability and effectiveness. Moreover, exploring the adaptability and scalability of this model for different types of roadwork scenarios or its integration into existing traffic management infrastructure could open new avenues for further development in this field. By addressing these areas, the TMA model could evolve into a more versatile and comprehensive solution, continuing to set higher standards in work zone safety technology.

Author Contributions: The authors confirm contribution to the paper as follows: Study Conception and Design: Y.A.-G. and C.S.; Data Collection: L.Z. and N.J.O.; Analysis and Interpretation of Results: N.J.O., L.Z., C.S., and Y.A.-G.; Draft Manuscript Preparation: N.J.O., C.S., and Y.A.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Fund under grant number 2045786.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request from authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Work-Zone-Crash-Facts.pdf. Available online: https://www.workzonebarriers.com/media/docs/print/work-zone-crash-facts.pdf (accessed on 29 May 2024).
2. Work Zones—Injury Facts. Work Zone Injury Facts. Available online: https://injuryfacts.nsc.org/motor-vehicle/motor-vehicle-safety-issues/work-zones/ (accessed on 27 August 2024).
3. Sixty-Four Percent of Firms Working on Highway Upgrades Experienced Cars Crashing into Their Work Zone during the Past Year, New Data Finds. Associated General Contractors of America. Available online: https://www.agc.org/news/2022/05/25/sixty-four-percent-firms-working-highway-upgrades-experienced-cars-crashing-their-work-zone-during (accessed on 27 August 2024).
4. Work Zone Awareness | Missouri Department of Transportation. Available online: https://www.modot.org/work-zone-awareness (accessed on 29 May 2024).
5. FHWA Work Zone— Temporary Traffic Control Devices Final Rule 23 CFR 630 Subpart K Questions and Answers (Updated 29 February 2008). Available online: https://ops.fhwa.dot.gov/wz/resources/temptraf_qa.htm (accessed on 29 May 2024).
6. Ullman, G.L.; Iragavarapu, V. Analysis of Expected Crash Reduction Benefits and Costs of Truck-Mounted Attenuator Use in Work Zones. *Transp. Res. Rec.* **2014**, *2458*, 74–77. [CrossRef]
7. Cottrell, B.H., Jr.; Virginia Transportation Research Council. Investigation of Truck Mounted Attenuator (TMA) Crashes in Work Zones in Virginia. FHWA/VTRC 16-R7; October 2015. Available online: https://rosap.ntl.bts.gov/view/dot/29718 (accessed on 28 January 2024).
8. TMA Crashes and Associated Employee Injuries-1h | Missouri Department of Transportation. Available online: https://www.modot.org/tma-crashes-and-associated-employee-injuries-1h (accessed on 27 August 2024).
9. Wu, D.; Liao, M.W.; Zhang, W.T.; Wang, X.G.; Bai, X.; Cheng, W.Q.; Liu, W.Y. YOLOP: You Only Look Once for Panoptic Driving Perception. *Mach. Intell. Res.* **2022**, *19*, 550–562. [CrossRef]
10. Han, C.; Zhao, Q.; Zhang, S.; Chen, Y.; Zhang, Z.; Yuan, J. YOLOPv2: Better, Faster, Stronger for Panoptic Driving Perception. *arXiv* **2022**, arXiv:2208.11434. Available online: http://arxiv.org/abs/2208.11434 (accessed on 28 January 2024).
11. Vu, D.; Ngo, B.; Phan, H. HybridNets: End-to-End Perception Network. *arXiv* **2022**, arXiv:2203.09035.
12. Zhan, J.; Luo, Y.; Guo, C.; Wu, Y.; Meng, J.; Liu, J. YOLOPX: Anchor-free multi-task learning network for panoptic driving perception. *Pattern Recognit.* **2024**, *148*, 110152. [CrossRef]
13. Yao, J.; Li, Y.; Liu, C.; Tang, R. Ehsinet: Efficient High-Order Spatial Interaction Multi-task Network for Adaptive Autonomous Driving Perception. *Neural Process. Lett.* **2023**, *55*, 11353–11370. [CrossRef]
14. Zhang, Y.; Yang, Q. An overview of multi-task learning. *Natl. Sci. Rev.* **2018**, *5*, 30–43. [CrossRef]
15. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
16. Thung, K.-H.; Wee, C.-Y. A brief review on multi-task learning. *Multimed. Tools Appl.* **2018**, *77*, 29705–29725. [CrossRef]
17. Crawshaw, M. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv* **2020**, arXiv:2009.09796.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**. [CrossRef]
20. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**. Available online: https://www.semanticscholar.org/paper/YOLOv3:-An-Incremental-Improvement-Redmon-Fa-hadi/ebc96892b9bcbf007be9a1d7844e4b09fde9d961 (accessed on 7 June 2024).
21. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October 2019; pp. 6568–6577. [CrossRef]
22. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872.
23. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv* **2015**, arXiv:1411.4038.

24. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. *arXiv* **2016**. [CrossRef]
25. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**. Available online: https://www.semanticscholar.org/paper/Rethinking-Atrous-Convolution-for-Semantic-Image-Chen-Papandreou/ee4a012a4b12d11d7ab8c0e79c61e807927a163c (accessed on 8 June 2024).
26. Ouyang, Y. Strong-Structural Convolution Neural Network for Semantic Segmentation. *Pattern Recognit. Image Anal.* **2019**, *29*, 716–729. [CrossRef]
27. Zhou, X.; Zhang, W.; Chen, Z.; Diao, S.; Zhang, T. Efficient Neural Network Training via Forward and Backward Propagation Sparsification. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2021; pp. 15216–15229.
28. Wang, Y.; Peng, Y.; Li, W.; Alexandropoulos, G.C.; Yu, J.; Ge, D.; Xiang, W. DDU-Net: Dual-Decoder-U-Net for Road Extraction Using High-Resolution Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [CrossRef]
29. Hough, V.; Paul, C. Method and Means for Recognizing Complex Patterns. U.S. Patent 3069654, 18 December 1962. Available online: https://www.osti.gov/biblio/4746348 (accessed on 9 June 2024).
30. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 286–291. [CrossRef]
31. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, No. 1.
32. Lee, S.; Kim, J.; Shin Yoon, J.; Shin, S.; Bailo, O.; Kim, N.; Lee, T.H.; Hong, H.S.; Han, S.H.; So Kweon, I. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1965–1973. [CrossRef]
33. Zheng, T.; Fang, H.; Zhang, Y.; Tang, W.; Yang, Z.; Liu, H.; Cai, D. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. *arXiv* **2021**, arXiv:2008.13719. [CrossRef]
34. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. *arXiv* **2019**, arXiv:1908.00821.
35. Yuan, C.; Liu, X.; Hong, X.; Zhang, F. Pixel-Level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7517–7524. [CrossRef]
36. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.