Edge-Assisted Object Perception for Autonomous Vehicles under Challenging Exposure and Blur Conditions

Qiren Wang, Yongtao Yao, Weisong Shi
Department of Computer and Information Science, University of Delaware, Newark, USA
{qirenw, yongtao, weisong}@udel.edu

***Abstract*—With the rapid advancements in machine learning (ML) and computing power, we have witnessed extensive deployment and testing of autonomous vehicles globally. By deploying ML models tailored for different scenarios on edge devices, including connected and autonomous vehicles, they have become increasingly intelligent in handling tasks such as object detection, lane keeping, and pothole detection. However, significant safety concerns persist, particularly concerning object detection in corner cases. We have identified two critical scenarios that urgently require effective handling in camera-based autonomous driving systems: challenging exposure and blur conditions. In this paper, we design and implement a new YOLO-EXB model based on the YOLO structure to enhance the safety of autonomous driving. First, we collect abnormal images for these two scenarios to compensate for the lack of corner case data. Next, we propose a novel backbone architecture integrating Transformer components into the original design, enhancing its feature extraction capabilities and modeling power. Finally, we evaluate our YOLO-EXB model on testing images under those two challenging conditions. Aiming to mitigate the impact of strong light and highly blurred images on autonomous driving perception, the proposed YOLO-EXB model has demonstrated improved detection capabilities and enhanced robustness under challenging exposure and blur conditions compared to both the baseline YOLOv8 and state-of-the-art YOLO11 models. In the conclusion of this paper, we discuss our limitations and highlight our contributions to improving the safety and effectiveness of autonomous driving in corner-case scenarios.**

I. Introduction

Autonomous vehicles are growing fast with the development of intelligent software and automotive hardware systems. Previously, car manufacturers commonly leveraged mechanical parameters such as horsepower and torque to showcase the performance advantages of their different vehicle models. The benefits of this new technology include the fact that software-defined vehicles (SDVs) have become increasingly popular. As its definition, automotive vehicles are equipped with multiple functions and intelligent software such as lane keeping, pothole detection, and collision detection. Software-defined vehicles have garnered substantial attention and interest from industry or researchers from academia. Original equipment manufacturers (OEMs) have shifted part of their business to software-defined vehicles to embrace this new era. Vehicle computing enables vehicles with artificial intelligence by putting advanced deep learning models on the car. Where previously computations were solely performed on central servers, edge devices now possess computational capabilities and accelerate the pace of autonomous vehicles (AVs). Vehicles are not only transportation for people to travel anymore; they are now empowered with computing units. Lu introduced connected vehicles that will perform as computing platforms for diverse edge-enabled services [1]. Unlike Vehicular Networking [2], which serves as a communication enabler for a wide range of transportation-related applications, Vehicle Computing focuses on the computational capabilities of connected vehicles (CVs). It emphasizes that the CV platform is a promising computing resource that can be utilized to analyze data streams from onboard sensors and surrounding connected devices, even when the vehicle is parked or in a charging state. To take advantage of this, autonomous vehicles are now performing as edges, allowing us to do some extra computing tasks.

In the real world, the useful data collected by autonomous vehicles are very limited. When considering driver assistance systems, the primary focus lies in handling everyday, routine driving scenarios. However, when it comes to fully autonomous vehicles, the emphasis shifts towards managing the most challenging and uncommon situations, also known as corner cases or edge cases, that arise at the boundaries of the system's operational domain. Most normal driving situations will not cause any dangers to autonomous vehicles, but they happen when edge cases occur.

(a) Exposure (b) Blur

Fig. 1: Three corner case scenarios under challenging conditions are demonstrated. They were captured with a mounted camera on autonomous vehicles during the daytime on the I-95 turnpike, freeways, and local ways. Vehicles with licenses and registrations were driven. (a) shows the blurred image. (b) shows the scenario with a subtle blurred background. (c) shows the challenging exposure situation.

The alarming statistics from national reports reveal that 44,000 people lose their lives in car accidents annually. The National Highway Traffic Safety Administration (NHTSA) reported an estimated fatality rate of 1.17 deaths per 100 million vehicle miles traveled during the first half of 2024, with 29,135 fatalities recorded in traffic crashes during the first nine months of the year. In the context of advancing autonomous driving technologies, safety remains the paramount concern. Public acceptance of autonomous vehicles hinges on their ability to demonstrate consistent reliability under diverse conditions. However, our observations indicate a notable gap in the object perception capabilities of autonomous vehicles. While standard scenarios are readily available for model training, countless corner cases emerge in real-world applications. Challenging scenarios, particularly those involving adverse weather and lighting conditions, are frequently linked to sensor failures. Such failures can critically impair an autonomous vehicle's ability to detect objects, leading to hazardous outcomes. Consequently, addressing these detection issues is imperative for improving the overall safety and reliability of autonomous driving systems.

Recent investigations conducted by the National Highway Traffic Safety Administration (NHTSA) have highlighted the significant role of low-visibility conditions, such as sun glare, fog, and airborne dust, in autonomous driving accidents. These include several collisions involving Tesla's "Full Self-Driving" system, one of which resulted in a pedestrian fatality in Rimrock, Arizona, in November 2023 [3]. These incidents underscore the challenges posed by variable lighting conditions, which can severely compromise the performance of camera-based perception systems. Autonomous vehicles must contend with the complexities of dynamic lighting and weather scenarios, which remain a significant barrier to reliable hazard detection. Addressing these limitations through enhanced perception models that can operate robustly under such conditions is a critical step toward achieving safer autonomous driving.

Our motivation is that an edge-based model could handle object perceptual tasks for autonomous driving under challenging exposure and blur conditions. Meanwhile, an edge-assisted object detection model will enable autonomous driving to detect objects in real-time while operating on the road. In this study, we introduce a novel transformer-integrated YOLO model to enhance object perception under challenging exposure and blur. At the same time, we deploy our YOLO-EXB on real edge devices, which are specific for autonomous vehicles such as the NVIDIA Orin AGX series, to validate feasibility and reliability. The main contributions of this paper are as follows:

The main innovations of our work are as follows:

- To address the lack of specific perceptual corner case data in the public dataset for autonomous driving, we create and collect our own dataset called AViX Dataset under challenging exposure and blur conditions in the real environment without any synthetic methods. After that, each image is labeled manually for detection purposes.
- We propose and implement a novel YOLO-EXB model to handle object detection under challenging exposure and blur conditions. We integrated a Transformer-based structure into the original model to enhance feature extraction.
- We conduct experiments and deploy our YOLO-EXB model on edge devices to validate its performance. The model's performance is thoroughly evaluated to demonstrate its feasibility and reliability. Meanwhile, model profiling is conducted to assess the hardware utilization when running on the NVIDIA Orin AGX platform.

The rest of this paper is organized as follows: Sec. II will review related work. Sec. III will describe our proposed YOLO-EXB model. The experimental results are shown in Sec. IV. Finally, we will present our discussion and conclusion in Sec. V.

II. Related Work

This section will introduce the related work from four aspects regarding simulators for autonomous driving, autonomous driving datasets, image fusion methods, and object detection techniques.

A. Real-world and Synthetic Datasets for Autonomous Driving

Numerous datasets have been developed to address the data shortage in autonomous driving research. Geiger *et al.* [4] introduced the KITTI Vision Benchmark Suite, a pioneering dataset for tasks like stereo vision, optical flow, and 3D object detection. Yu *et al.* [5] presented BDD100K, a diverse driving video dataset designed for multitask learning across various conditions. The Virtual KITTI2 dataset [6] extends the original Virtual KITTI with synthetic data to simulate diverse weather conditions and viewpoints for robust algorithm evaluation. NuScenes [7] offers a multimodal dataset with extensive sensor coverage, supporting advanced detection and tracking research.

Other significant contributions include the FordAV Dataset [8], focusing on seasonal and urban variations; the Waymo Open Dataset [9], one of the largest multimodal datasets with diverse urban and suburban scenes; and the Zenseact Open Dataset (ZOD) [10], which emphasizes long-range perception and multi-task learning using high-resolution data. These datasets collectively enhance autonomous driving research by providing diverse, high-quality benchmarks for robust perception algorithm development.

Additionally, some datasets under adverse weather were collected. Researchers evaluated the detection tasks on it to get more comprehensive feedback for autonomous driving. Kenk and Hassaballah [11] introduced the DAWN dataset, comprising 1,000 real-world images collected under adverse weather conditions, such as fog, rain, snow, and sandstorms. The dataset provides annotations for vehicle detection in diverse traffic scenarios, offering a benchmark for autonomous driving and visual surveillance tasks. The authors emphasize the limitations of current detection methods, which often rely on synthetic datasets and struggle to balance accuracy and real-time performance in challenging weather conditions.

DAWN addresses these gaps by enabling robust evaluation of detection systems in real-world adverse environments. In the same year, the Canadian Adverse Driving Conditions (CADC) dataset [12] focuses on autonomous driving in challenging winter weather conditions. Collected using the Autonomoose platform, it includes multi-modal data from cameras, LiDAR, and GNSS/INS sensors under various snowfall scenarios. CADC addresses the gaps in existing datasets by providing annotated data specifically for adverse weather, facilitating research in 3D object detection and tracking for autonomous vehicles. This dataset highlights the unique challenges of perception tasks in snow-covered environments. WEDGE, introduced by Marathe *et al.* [13], leverages vision-language generative models to create a synthetic dataset for autonomous driving under extreme weather conditions. The dataset facilitates research in weather classification and object detection while addressing the Sim2Real gap in perception systems. Their work highlights the potential of synthetic data to improve the robustness of autonomous driving models in challenging weather scenarios.

B. Classic Object Detection Models

Classical object detection methods are generally categorized into one-stage and two-stage approaches. One-stage detectors, such as the SSD and YOLO families, prioritize speed and simplicity, while two-stage detectors, such as the R-CNN family, focus on accuracy and robust feature extraction.

The Single Shot MultiBox Detector (SSD), introduced by Liu *et al.* [14], is an efficient one-stage method that uses a single neural network to predict object bounding boxes and class probabilities. It employs default boxes at multiple scales and aspect ratios, achieving high accuracy and speed suitable for real-time applications. DSSD [15] enhances SSD by adding deconvolution layers to provide better context, particularly for small object detection, further improving detection performance.

The YOLO (You Only Look Once) series, first proposed by Redmon *et al.* [16], redefines object detection as a regression problem, enabling rapid prediction of bounding boxes and class probabilities in a single pass. Over the years, the YOLO family has evolved significantly, with models like YOLOv3 [17] introducing multi-scale predictions to improve detection accuracy across various object sizes. Recent iterations, such as YOLOv8, YOLOv9, YOLOv10, and YOLO11 [18]–[21], represent the latest advancements in the series, maintaining state-of-the-art (SOTA) performance in object detection tasks.

Two-stage detectors, such as the R-CNN family, began with the introduction of R-CNN by Girshick *et al.* [22], which combined region proposals with CNN-based feature extraction for classification and bounding box regression. Fast R-CNN [23] improved computational efficiency by processing the entire image once and mapping region proposals onto the feature map. Faster R-CNN [24] further streamlined the process by introducing a region proposal network (RPN) for end-to-end training. Sparse R-CNN [25] challenges traditional dense detection methods by using a small set of learned object proposals, eliminating the need for non-maximum suppression and dense anchor boxes. This approach simplifies the detection pipeline while achieving competitive performance, paving the way for efficient sparse object detection frameworks.

C. Transformer-based Model for Perception Tasks

Detection with Transformer (DETR) [26] revolutionizes object detection by treating it as a direct set prediction problem, utilizing a transformer-based encoder-decoder architecture. This approach eliminates the need for traditional components like non-maximum suppression (NMS) and anchor generation, simplifying the detection pipeline. By leveraging bipartite matching and global image context, DETR achieves competitive performance with state-of-the-art methods on COCO [27] while also being capable of extending to tasks like panoptic segmentation. This end-to-end model not only simplifies training and deployment but also enhances detection accuracy through its innovative use of transformers for object relations and global context reasoning.

ViT [28], designed for large-scale image recognition tasks, introduced a transformer-based structure that has inspired numerous subsequent models in computer vision. Building on the success of ViT, MobileViT [29], developed by Apple, adapts this architecture for mobile devices. It is a lightweight model that integrates the local feature detection strengths of CNNs with the global information processing capabilities of Vision Transformers, achieving a balance between efficiency and performance.

Swin Transformer [30] further advanced transformer-based vision models by introducing a hierarchical structure that employs shifted windows for self-attention. This innovative design enables efficient multi-scale feature learning, making Swin Transformer suitable for both image recognition and dense prediction tasks. Swin Transformer V2 [31] improved upon its predecessor with enhanced training techniques and larger model capacities, achieving state-of-the-art results across various benchmarks and further cementing its versatility.

Similarly, the Pyramid Vision Transformer (PVT) [32] addressed the challenge of applying transformers to dense prediction tasks, such as object detection and semantic segmentation. By introducing a pyramid structure, PVT processes feature maps at multiple scales, combining the hierarchical efficiency of CNNs with the global modeling power of transformers. PVTv2 [33] refined this design by incorporating linear complexity attention mechanisms and optimized feature fusion techniques. These improvements reduced computational overhead while maintaining high performance, establishing PVTv2 as a robust and efficient backbone for a wide range of vision tasks.

D. Transfer Learning for Autonomous Driving

Transfer learning plays a critical role in autonomous driving, focusing on addressing domain gaps under varying conditions through model adaptation and enhancing model generalization across different environments. A systematic approach

leveraging simulated accident scenarios has been developed to address the scarcity of real-world data for edge cases in autonomous driving [34]. By parameterizing common accident scenarios based on NHTSA pre-crash descriptions, this method combines simulated and real-world data through transfer learning, leading to improved model generalization and collision avoidance. The work demonstrates the value of simulation data in enhancing real-world driving models, particularly for rare and critical driving scenarios. A transfer learning method for autonomous driving [35] leverages spatio-temporal features to improve cross-domain generalization. This method combines spatial information from CNNs with temporal dynamics from LSTMs, enabling robust adaptation across domains. By incorporating salient data augmentation and a two-phase training process, it demonstrates significant improvements in unseen environments, effectively addressing domain shifts between simulated and real-world driving scenarios.

E. Risk Mitigation under Adverse Scenairos

Autonomous safety has been a topic of discussion among academic researchers and industry experts for several years. Various challenges, such as heavy rain and dense fog, pose significant obstacles to autonomous vehicles, particularly in object detection. Researchers are actively working to address these critical issues and advance the underlying technologies through different aspects of techniques. Volk [36] proposed a method to enhance CNN robustness in autonomous driving by augmenting datasets with synthetic rain effects, including falling rain and raindrops on the windshield. The approach applied these effects to the KITTI dataset and demonstrated improved model performance. Compared to traditional augmentation techniques like Gaussian noise and Salt-and-Pepper noise, this method achieves better results when validated on a real rain dataset. Additionally, the optimized models are compared with the robust RRC model, showcasing the effectiveness of the proposed method. Sakaridis *et al.* [37] addressed the challenge of semantic foggy scene understanding (SFSU) by generating synthetic fog on real clear-weather images using a scalable fog simulation pipeline. They introduced the Foggy Cityscapes dataset with synthetic fog applied to the Cityscapes dataset and a real-world Foggy Driving dataset for evaluation. Their study combines supervised and semi-supervised learning approaches, demonstrating that synthetic fog data and domain adaptation techniques significantly enhance performance in foggy conditions. Additionally, they evaluated the impact of image dehazing on SFSU and provided insights into human perception of foggy scenes. Li [38] proposes a domain adaptive object detection framework for autonomous driving under foggy weather, addressing the performance degradation caused by domain gaps between clear and foggy conditions. The method combines image-level and object-level adaptations to reduce discrepancies in image style and object appearance, leveraging labeled clear-weather data and unlabeled foggy-weather data. It introduces an Adversarial Gradient Reversal Layer (AdvGRL) to perform hard example mining and incorporates an auxiliary domain generated through data augmentation to enforce domain-level metric regularization. Experimental results on Cityscapes and Foggy Cityscapes demonstrate superior performance compared to baseline and existing methods.

III. Methodology

A. Data Collection and Preparation

In our study, we created our own dataset called AViX Dataset. AViX stands for Autonomous Vehicles Interference by X, where X can represent any extreme and challenging conditions or environments. In our current work, X specifically refers to data containing Challenging Exposure and Blur. We employed a Ford Lincoln equipped with multiple sensors as our autonomous driving test platform. Fig. 3 shows the configuration of our autonomous vehicle. The platform runs on ROS2 Humble based on Ubuntu 22.04 LTS and is equipped with various advanced sensors, including 7 Basler ace acA1920-40gc industrial cameras, 2 Velodyne VLP16 LiDARs, 1 Hesai Pandar64 LiDAR, and 1 Novatel OEM7 GPS/GNSS receiver. The Basler ace acA1920-40gc industrial cameras utilize Sony IMX249 CMOS sensors, featuring a resolution of 2.3 megapixels (1920x1200) and are capable of capturing up to 42 frames per second. The camera sensor size is 1/1.2 inches, and it communicates with the computer through a Gigabit Ethernet (GigE) interface. The Velodyne VLP16 LiDAR provides reliable point cloud data through 16 channels, while the Hesai Pandar64 LiDAR features 64 channels with a detection range of up to 200 meters. The Novatel OEM7 GPS/GNSS receiver delivers a centimeter-level high-precision positioning service. In this research, we focus on visual perception technology, therefore only utilizing the camera sensors from our test platform to collect high-quality image data. To avoid excessive data redundancy, we set the camera acquisition frequency to 1 frame per second. The data collection routes covered various driving scenarios, gathering extensive raw image data under different lighting and weather conditions to support subsequent research and development of autonomous driving vision algorithms. For our AViX Dataset, data collection was conducted from mid-2024 to late 2024, primarily covering residential areas, local streets, and some highway sections. Data was collected across different days and at various times throughout each day. Dataset examples are shown in Fig. 2 In the current dataset, we have annotated cars and trucks without further distinguishing between specific vehicle types. For instance, sedans, SUVs, and minivans are all labeled as cars. However, pickup trucks, vans, large trucks, cargo vehicles, recreational vehicles, and container trucks are categorized as trucks. Fig. 4. illustrates the specific annotation distribution in our training and testing datasets. From the 10,000 images collected, we carefully selected 800 images to avoid scene repetition. Each of these 800 images contains either exposure or blur conditions, with moderate to severe levels of exposure and blur effects. Following standard guidelines, we split the 800 images into a training set of 640 images and a test set of 160 images, maintaining an 8:2 ratio. During the dataset division, we maintained the same proportion of exposure

Fig. 2: This figure shows the examples of our AViX Dataset. The AViX dataset includes challenging exposure and blur images. The dataset is collected on the highway, local ways, etc.

and blurred images in both training and test sets to ensure consistency in data distribution. Additionally, all objects in our dataset are annotated following the YOLO label format, where each object is described by five values: class index and four normalized bounding box coordinates. The first value represents the object class (0 for car and 1 for truck in our case), followed by the center coordinates (x, y) and dimensions (width, height) of the bounding box. All coordinate values are normalized to [0,1] by dividing by the image dimensions, where (0,0) represents the top-left corner of the image and (1,1) represents the bottom-right corner. This standardized annotation format ensures compatibility with YOLO-based object detection frameworks while maintaining consistent label representation across different image resolutions.

Fig. 3: These two images showcase our autonomous vehicles, with all data manually collected by two autonomous vehicles from our lab. The image on the left shows the full view of our autonomous car, while the right image provides a close-up view of the cameras mounted on top, used for data collection in this experiment.

B. Proposed Model Framework

To ensure the reliability of our results, we first conducted a thorough analysis and selection of the baseline model. We focused on improving and comparing models from the YOLO series within the one-stage detector family. Among these, YOLOv8 is currently officially acknowledged by Ultralytics as their proprietary model, with YOLO11 being the latest

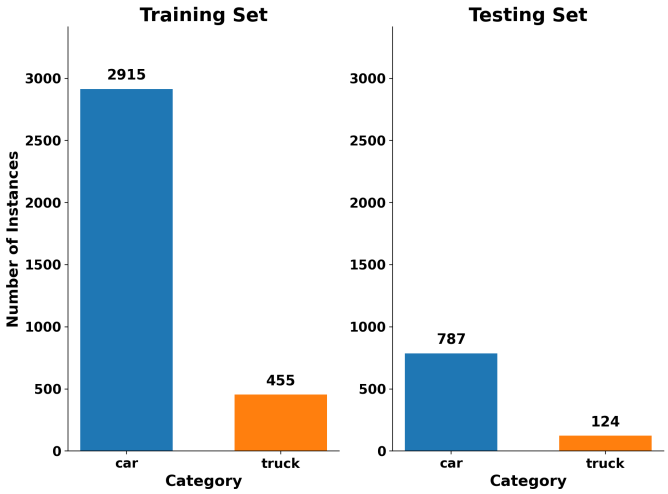

Fig. 4: This figure shows the counts of each category (car and truck).

iteration. Each YOLO series includes models of varying sizes: n, s, m, l, and x, in ascending order of complexity. In this study, we consistently used the smallest 'n' variant from each YOLO version for comparison. During our preliminary observation phase, we fine-tuned YOLOv8, YOLOv9, YOLOv10, and YOLO11 on our training dataset. YOLOv8 demonstrated superior balanced performance in both model accuracy and speed on our AViX dataset, leading to its selection as our baseline model.

In this research, we enhanced the backbone of YOLOv8 by introducing an improved version of the PVTv2 attention mechanism into its feature extraction network. Specifically, we replaced the original C2f module with our newly designed C2f-PVTv2 module, which maintains the branch structure characteristics of CSP (Cross Stage Partial Network) while

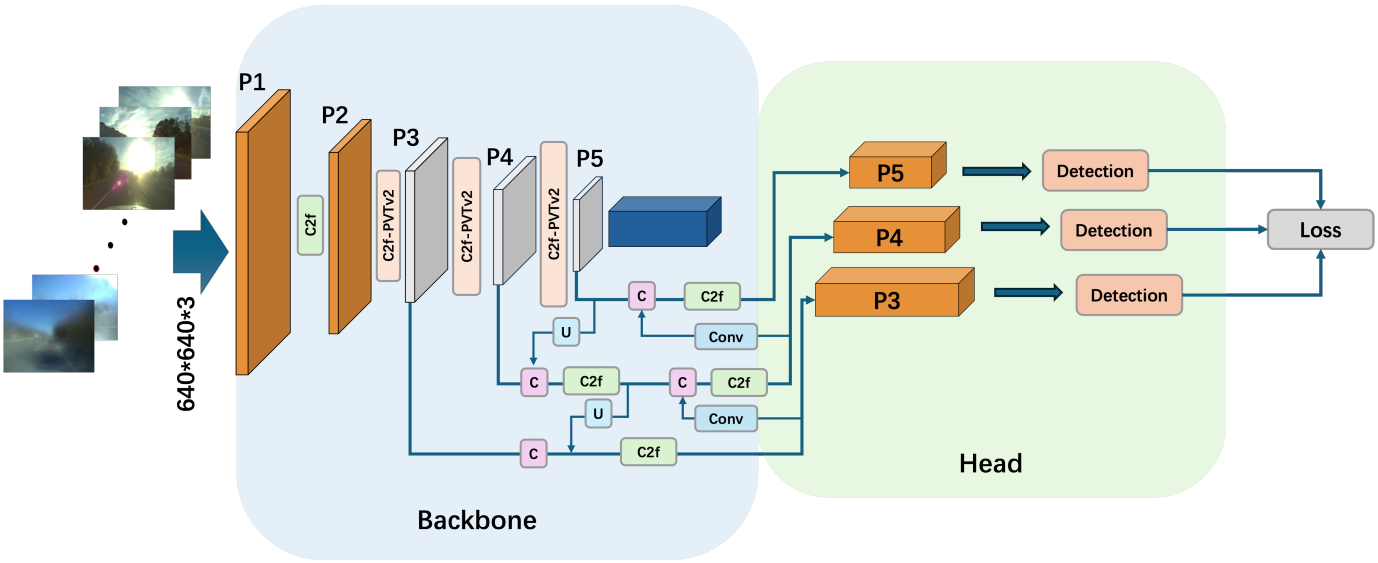

Fig. 5: This figure illustrates the structure of our YOLO-EXB model. We split the model into the backbone part and the head part for more detailed reference. Our proposed transformer-based module is shown as C2f-PVTv2. Here, "C" represents the concatenation operation, "U" denotes upsampling, and "P" refers to "Pyramid".

incorporating PVTv2's spatial reduction attention mechanism. Within the backbone, we deployed two C2f-PVTv2 modules at three different feature scales, with channel dimensions of 256, 512, and 1024 respectively. Each C2f-PVTv2 module utilizes 8 attention heads and implements a spatial reduction ratio (sr-ratio) of 4. The Fig. 5 gives the overview of our proposed model. In terms of implementation details, we employed a compression coefficient e=0.5 to regulate the hidden channel dimensions, effectively halving the intermediate feature channels relative to the output channels in each C2f-PVTv2 module. For regularization, we incorporated dropout mechanisms in both attention and projection layers, alongside DropPath in the feature transmission pathway, to enhance model robustness. Additionally, the depth-wise separable convolution integrated within the MLP module employs a 3×3 convolution kernel with stride 1 and padding 1, maintaining feature map dimensions while effectively capturing local features. This PVTv2-based improvement significantly enhances the model's feature extraction capabilities while maintaining computational efficiency, particularly demonstrating superior adaptability when processing objects of varying scales in our AViX dataset.

Fig. 6: On the left is our workstation equipped with four NVIDIA GeForce RTX 2080Ti graphics cards, which provide computing resources during the training stage. The bottom-right image shows the physical appearance of the GPU cards, while the top-right image features a NVIDIA Jetson Orin AGX edge board with 64GB of memory.

C. Hardware Setup

In our work, we trained our model only on one NVIDIA GPU graphic card (GeForce RTX 2080 Ti) with a workstation. We assume an autonomous driving (AV) car was equipped with computing capability. Considering autonomous vehicles as a computing platform. Our workstation as shown in Fig. 6, has four NVIDIA GeForce RTX 2080Ti graphics cards. Each of them has 12 GB of memory and far enough for object detection models. The NVIDIA GeForce RTX 2080Ti is a reliable hardware device widely adopted by researchers because of its robust performance. This workstation also uses an Intel i9-9940X CPU with 3.30GHz and 64GB of main memory. For our inference purposes, we adopt the NVIDIA Jetson Orin AGX board. An NVIDIA Jetson Orin AGX board is equipped with 64GB memory, Ampere GPU, and Arm Cortex CPU. This kind of board is the latest edge device, especially for autonomous driving settings. Within this testbed, we put our model on it to do performance evaluation and provide solid evidence for real deployment on autonomous vehicles.

IV. EXPERIMENTS

A. Evaluation Metric

To make our experiments more comprehensive and quantitative, we will evaluate two aspects of our model which are model performance and hardware performance. Model performance includes precision(P), recall(R), f1-score, and mean average precision(AP). Hardware performance includes average memory usage, average CPU usage, and average GPU usage. For hardware evaluation, we monitor the usage and then calculate the average usage within the running time.

a) Intersection-over-Union: First, a common and well-established object detection metric is Intersection-over-Union (IoU) [39]. IoU is a commonly used metric for measuring object localization accuracy. It quantifies the overlap between the predicted bounding box and the ground truth bounding box. IoU serves as a straightforward and effective measure for any task that involves producing a predicted bounding box in the output. It can be represented as the following formula:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

A be the area of the predicted bounding box, B be the area of the ground truth bounding box.

b) F1-Score: To get the F1-score for a model, we need to calculate precision and recall first. The F1-score is the harmonic mean of precision and recall, providing a balance between the two. The precision quantifies the proportion of true positive predictions among all positive predictions. It reflects the ability of a model to accurately identify negative samples, while recall indicates the model's proficiency in recognizing positive samples. The calculation formula is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

In classification tasks, a True Positive (TP) occurs when actual positive samples are correctly identified as positive, while a False Negative (FN) occurs when actual positive samples are incorrectly identified as negative. A False Positive (FP) happens when actual negative samples are mistakenly identified as positive, and a True Negative (TN) occurs when actual negative samples are correctly identified as negative. The recall value indicates the proportion of actual positive samples among all positive samples in the prediction results. Specifically, it measures the proportion of actual positive samples within the predicted positive samples in the entire dataset. The calculation formula is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

After that, the F1 score is calculated as follows:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

The F1 score integrates both aspects, offering a comprehensive measure of the robustness of a model.

c) Mean Average Precision: Mean Average Precision (mAP) is a widely used metric to evaluate object detection models, representing the average of the Average Precision (AP) values across all categories. AP is computed as the mean of the maximum precision values at various recall levels, typically evaluated separately for each category. For each category, the precision values at 11 recall levels from 0 to 1, with increments of 0.1, are calculated. These 11 points contribute to the AP calculation by averaging these precision values. The formula for Average Precision (AP) is expressed as:

$$AP = \frac{1}{11} \sum_{i=0}^{10} P_{\text{smooth}}(i) \tag{5}$$

During the evaluation of model performance, the IoU threshold is set at 0.5 according to the Pascal VOC 2008 challenge [40]. When multiple detections are made for a single object, the one with the highest confidence is considered a true positive, and the others are treated as false positives. The AP is derived by averaging precision values at 11 equidistant points on the smoothed precision-recall curve.

To compute mAP, the AP values for all categories are averaged as follows:

$$mAP = \frac{1}{N} \sum_{c=1}^{N} AP_c \tag{6}$$

where N is the total number of categories, and AP_c represents the Average Precision for category c. This metric provides a comprehensive measure of the overall performance of the detection model across multiple object categories.

B. Ablation Study

We first conducted an ablation study explicitly comparing our proposed YOLO-EXB model which integrated the C2f-PVTv2 module against YOLOv8n as our baseline on the AViX dataset. Table I summarizes the performance comparison between YOLO-EXB and YOLOv8n under challenging exposure and blur conditions. At a strict confidence threshold of 0.5, YOLO-EXB achieves a notable improvement in $\text{mAP}_{50:95}$, rising by 1.5% from 45.4% (YOLOv8n) to 46.9%. At a lower confidence threshold of 0.001, YOLO-EXB further demonstrates enhanced robustness, increasing $\text{mAP}_{50:95}$ by 1.7% compared to YOLOv8n. These results confirm the effectiveness of integrating Transformer-based components within the YOLO backbone.

TABLE I: Ablation Study: YOLO-EXB vs. YOLOv8n (All Classes)

Model	Conf. Thr.	P (%)	R (%)	F1 (%)	mAP_{50} (%)	$\text{mAP}_{50:95}$ (%)
YOLOv8n(Baseline)	0.5	89.5	45.4	60.2	68.1	45.4
YOLO-EXB	0.5	89.5	45.2	60.1	**68.8**	**46.9**
YOLOv8n(Baseline)	0.001	77.9	55.0	64.5	64.6	37.0
YOLO-EXB	0.001	69.9	60.4	64.8	**66.5**	**38.7**

ALL Classes							
Model	**#Param. (M)**	**FLOPs (G)**	**Precision (P)**	**Recall (R)**	**F1-Score**	**mAP$_{50}$ (%)**	**mAP$_{50:95}$ (%)**
Confidence Threshold = 0.5							
YOLO-EXB (Ours)	2.7	7.2	89.5	45.2	60.1	**68.8**	**46.9**
YOLOv8n	3.2	8.7	89.5	45.4	60.2	68.1	45.4
YOLOv9t	2.0	7.7	90.2	42.4	57.7	66.7	45.2
YOLOv10n	2.3	6.7	91.2	35.9	51.5	64.4	44.7
YOLO11n	2.6	6.5	84.3	45.9	62.2	68.3	45.9
Confidence Threshold = 0.001 (default)							
YOLO-EXB (Ours)	2.7	7.2	69.9	60.4	64.8	**66.5**	**38.7**
YOLOv8n	3.2	8.7	77.9	55.0	64.5	64.6	37.0
YOLOv9t	2.0	7.7	74.7	55.7	55.7	62.8	36.5
YOLOv10n	2.3	6.7	62.3	54.6	58.2	61.6	36.3
YOLO11n	2.6	6.5	68.9	62.1	65.3	66.3	38.4

TABLE II: Performance comparison of YOLO models, under different confidence thresholds for all classes.

Class Car							
Model	**#Param. (M)**	**FLOPs (G)**	**Precision (P)**	**Recall (R)**	**F1-Score**	**mAP$_{50}$ (%)**	**mAP$_{50:95}$ (%)**
Confidence Threshold = 0.5							
YOLO-EXB (Ours)	2.7	7.2	96.1	63.0	75.8	80.3	56.2
YOLOv8n	3.2	8.7	97.6	62.5	76.3	80.5	56.9
YOLOv9t	2.0	7.7	95.8	58.2	72.5	77.8	55.3
YOLOv10n	2.3	6.7	93.9	53.2	68.0	74.5	52.8
YOLO11n	2.6	6.5	92.5	67.9	78.4	81.8	57.0
Confidence Threshold = 0.001 (default)							
YOLO-EXB (Ours)	2.7	7.2	85.2	78.8	82.0	86.4	51.4
YOLOv8n	3.2	8.7	90.7	74.0	81.5	85.8	51.4
YOLOv9t	2.0	7.7	85.0	73.4	78.8	81.8	49.4
YOLOv10n	2.3	6.7	74.3	77.0	75.7	81.4	49.2
YOLO11n	2.6	6.5	81.4	79.8	80.6	85.4	51.6

TABLE III: Performance comparison of YOLO models for the class "car" under different confidence thresholds.

Class Truck							
Model	**#Param. (M)**	**FLOPs (G)**	**Precision (P)**	**Recall (R)**	**F1-Score**	**mAP$_{50}$ (%)**	**mAP$_{50:95}$ (%)**
Confidence Threshold = 0.5							
YOLO-EXB (Ours)	2.7	7.2	82.9	27.4	41.1	57.3	37.7
YOLOv8n	3.2	8.7	81.4	28.2	41.6	55.6	33.4
YOLOv9t	2.0	7.7	84.6	26.6	40.4	55.6	35.1
YOLOv10n	2.3	6.7	88.5	18.5	30.7	54.3	36.7
YOLO11n	2.6	6.5	76.0	30.6	43.4	54.9	34.8
Confidence Threshold = 0.001 (default)							
YOLO-EXB (Ours)	2.7	7.2	54.5	41.9	47.6	46.6	26.1
YOLOv8n	3.2	8.7	65.1	36.1	46.4	43.4	22.5
YOLOv9t	2.0	7.7	64.4	37.9	47.5	43.9	23.5
YOLOv10n	2.3	6.7	50.2	32.3	39.4	41.8	23.4
YOLO11n	2.6	6.5	56.5	44.4	50.1	47.3	25.2

TABLE IV: Performance comparison of YOLO models for the class "truck" under different confidence thresholds.

C. Comprehensive Performance Evaluation

Having established the improvements of YOLO-EXB through our ablation study, we conducted an extensive evaluation to comprehensively benchmark its performance against other state-of-the-art YOLO variants, namely YOLO11n, YOLOv10n, and YOLOv9t, using the AViX dataset.

At a confidence threshold of 0.5, YOLO-EXB distinctly excels by achieving the highest mAP$_{50:95}$ of 46.9%, which notably surpasses YOLO11n (45.9%), YOLOv9t (45.2%), and YOLOv10n (44.7%). This superior performance highlights the effectiveness of our model in accurately detecting objects across diverse challenging conditions. Specifically, for the car detection class, YOLO-EXB demonstrates remarkable precision (96.1%) and a competitive recall (63.0%), leading to a strong F1 score of 75.8%. Similarly, for truck detection, a category characterized by greater variability and complexity, YOLO-EXB attains the highest mAP$_{50}$ of 57.3% and mAP$_{50:95}$ of 37.7%, confirming its robustness and enhanced adaptability.

Evaluating model performance at the default lower confidence threshold of 0.001 further underscores YOLO-EXB's

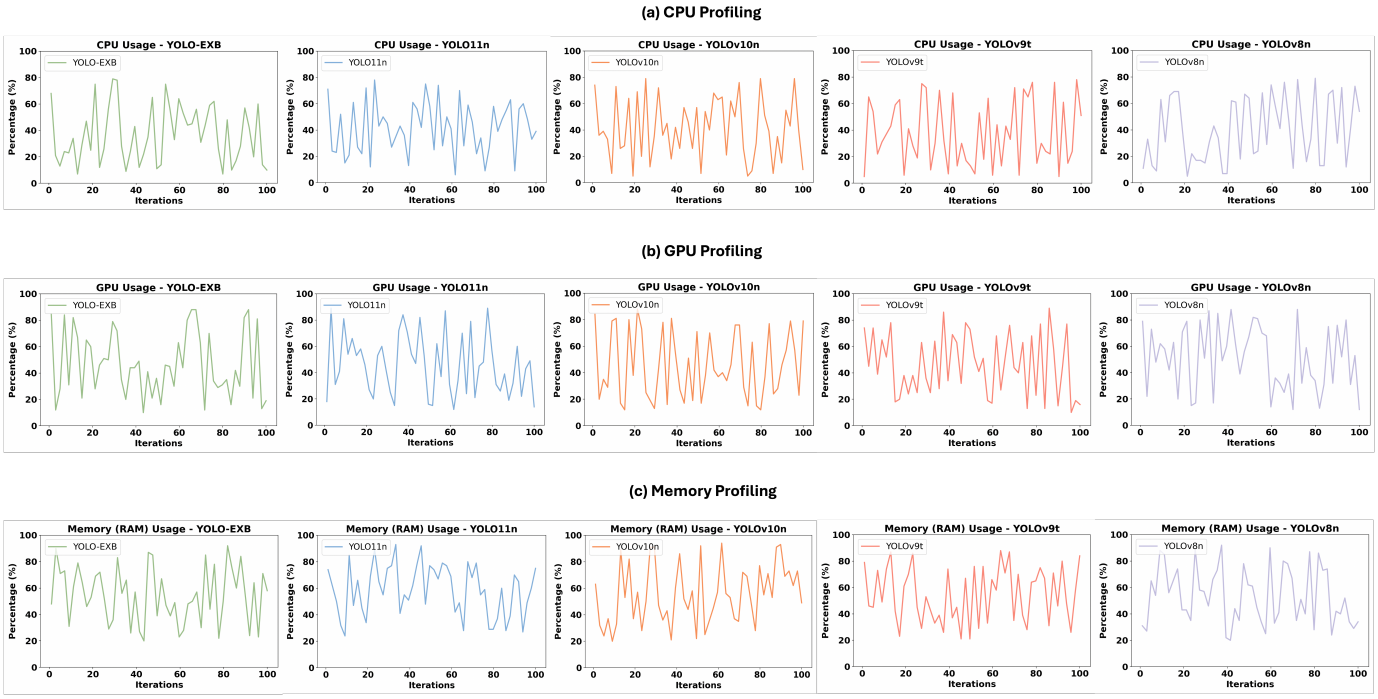

Fig. 7: This figure shows the (a)CPU Profiling (b)GPU Profiling (c)Memory Profiling information among YOLO-EXB(Ours), YOLO11n, YOLOv10n, YOLOv9t, YOLOv8n

advantage. It maintains an optimal balance between precision (69.9%) and recall (60.4%), producing a solid F1 score of 64.8%. While YOLO11n occasionally achieves higher recall rates, YOLO-EXB consistently delivers superior mAP metrics, showcasing its robustness and reliability in handling adverse perceptual scenarios.

Moreover, the comprehensive computational profiling depicted in Fig. 7 and Table V illustrates YOLO-EXB's efficient hardware resource utilization. Our model achieves balanced CPU (42.46%), GPU (49.28%), and memory (57.76%) usage, indicating that YOLO-EXB is optimally designed not only for accuracy but also for efficient deployment in practical autonomous driving environments. This holistic performance underscores YOLO-EXB's suitability for real-world applications, demonstrating its significant advantages over contemporary YOLO-based detection models.

TABLE V: Model Average Usage

Model	CPU (%)	GPU (%)	Memory (%)
YOLO-EXB (Ours)	42.46	49.28	57.76
YOLOv8n	38.52	48.32	55.08
YOLOv9t	40.90	51.08	57.32
YOLOv10n	40.48	47.34	57.50
YOLO11n	40.00	57.78	52.86

To evaluate computational efficiency, we conducted detailed profiling of CPU, GPU, and memory utilization across YOLO-EXB, YOLO11n, YOLOv10n, YOLOv8n, and YOLOv9t under identical conditions (Table V, Fig. 7). YOLO-EXB demonstrates balanced resource usage (CPU: 42.46%, GPU: 49.28%, memory: 57.76%), highlighting its effective workload distribution and adaptability for practical autonomous driving applications. By comparison, the baseline YOLOv8n shows slightly lower CPU demands (38.52%) but similar GPU and memory usage. YOLO11n exhibits the highest GPU utilization (57.78%) but less uniform overall resource usage. The profiling confirms YOLO-EXB's superior efficiency and balanced hardware performance, making it particularly suitable for real-world deployments requiring reliable and comprehensive hardware utilization.

V. Conclusion and Discussion

In this work, we proposed a novel fused backbone structure incorporating a transformer-based architecture to address detection challenges under conditions of extreme exposure and blur. Additionally, we created the AViX dataset, specifically designed to tackle the challenges of exposure and blur, providing a valuable resource for addressing corner cases and challenging scenarios that are difficult to capture in real-world settings. Our proposed YOLO-EXB model demonstrates improved accuracy and robustness in such conditions. However, latency was not extensively analyzed in this study. This integrated module enhances the safety of autonomous vehicles by improving detection performance in extreme situations while maintaining low computational complexity. Moreover, we evaluated both software and hardware performance to ensure efficient deployment. As a next step, our proposed solution will further validate its effectiveness in physically autonomous vehicles. In the end, to promote continuous attention and advancement in this field, we will open-source our code and related data for researchers and industry professionals.

REFERENCES

[1] S. Lu and W. Shi, "Vehicle computing: Vision and challenges," *Journal of Information and Intelligence*, vol. 1, no. 1, pp. 23–35, 2023.

[2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE communications surveys & tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[3] Associated Press, "Us to probe tesla's 'full self-driving' system after pedestrian killed in low visibility conditions," *AP News*, 2024, retrieved from https://apnews.com/article/f2121166d60d85bd173a734c91049e73.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[5] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[6] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[8] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1367–1376, 2020.

[9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[10] M. Alibeigi, W. Ljungbergh, A. Tonderski, G. Hess, A. Lilja, C. Lindström, D. Motorniuk, J. Fu, J. Widahl, and C. Petersson, "Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 178–20 188.

[11] M. A. Kenk and M. Hassaballah, "Dawn: vehicle detection in adverse weather nature dataset," *arXiv preprint arXiv:2008.05402*, 2020.

[12] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 681–690, 2021.

[13] A. Marathe, D. Ramanan, R. Walambe, and K. Kotecha, "Wedge: A multi-weather autonomous driving dataset built from generative vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 3318–3327.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. 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*. Springer, 2016, pp. 21–37.

[15] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[18] G. Jocher, "Yolov8: Github repository," https://github.com/ultralytics/ultralytics, 2023.

[19] C.-Y. Wang and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," 2024.

[20] L. L. e. a. Ao Wang, Hui Chen, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[21] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[25] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 454–14 463.

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[28] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[29] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[31] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[32] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[33] ——, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.

[34] S. Akhauri, L. Y. Zheng, and M. C. Lin, "Enhanced transfer learning for autonomous driving with systematic accident simulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5986–5993.

[35] S. Akhauri, L. Zheng, T. Goldstein, and M. Lin, "Improving generalization of transfer learning across domains using spatio-temporal features in autonomous driving," *arXiv preprint arXiv:2103.08116*, 2021.

[36] G. Volk, S. Müller, A. Von Bernuth, D. Hospach, and O. Bringmann, "Towards robust cnn-based object detection through augmentation with synthetic rain variations," in *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 285–292.

[37] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, pp. 973–992, 2018.

[38] J. Li, R. Xu, J. Ma, Q. Zou, J. Ma, and H. Yu, "Domain adaptive object detection for autonomous driving under foggy weather," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 612–622.

[39] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[40] D. Hoiem, S. K. Divvala, and J. H. Hays, "Pascal voc 2008 challenge," *World Literature Today*, vol. 24, no. 1, pp. 1–4, 2009.