D2SO: Detecting Distant and Small Objects for Vision-based Vehicle Autonomous Systems

Hanzhi Zhang, Harold Lucero, Kewei Sha, Heng Fan, Song Fu, Yunhe Feng
University of North Texas, Denton, TX, USA
{hanzhizhang, haroldlucero}@my.unt.edu, {kewei.sha, heng.fan, song.fu, yunhe.feng}@unt.edu

***Abstract*—Detecting distant and small objects is a critical capability for vision-based vehicle autonomous systems, particularly in safety-critical scenarios such as assisted driving, where constant alertness, early reaction, and safe operation are very important. However, accurately recognizing tiny objects at a distance presents significant challenges due to limited pixel information, preprocessing-induced downscaling, and restricted detector resolution. To address these issues, this paper introduces *D2SO*, a Vision Transformer (ViT)-based framework specifically designed to enhance the detection of distant and small objects for autonomous systems. *D2SO* integrates multiple fine-tuned AI models, derived from the open-source Segment Anything Model (SAM), to detect distant objects such as static structures, humans, and vehicles, thereby improving situational awareness. The system employs visual cues through color mask overlays to efficiently convey essential information, ensuring users remain well-informed about detection outcomes. *D2SO* is explicitly optimized to detect distant entities that occupy fewer than 24×24 pixels on display, a scale often imperceptible to humans. Experimental results demonstrate that *D2SO* significantly outperforms baseline models, including SegFormer, YOLO v11 Segmentation, and U-Net, on a real-world street scene dataset spanning 50 cities, establishing its effectiveness in enhancing autonomous system performance.**

***Index Terms*—Small Object Detection, Distant Object Detection, Image Segmentation, Vehicle Autonomous Systems, Segment Anything Model (SAM), Vision Transformer (ViT)**

I. Introduction

Autonomous systems are rapidly evolving alongside advancements in artificial intelligence and robotics. Safety-critical applications, such as driver-assistance technologies and autonomous vehicles, are transforming transportation by reducing traffic incidents and enhancing mobility efficiency [1]. According to the World Health Organization, road traffic injuries are projected to become the eighth leading cause of death globally [2]. This alarming statistic highlights the pressing need for advanced safety mechanisms in vehicles. Autonomous technologies play a pivotal role in addressing these challenges by minimizing human error and significantly improving vehicle reaction times [3]. However, despite their potential, the widespread adoption and trust in vehicle autonomous systems remain hindered by concerns over their reliability in addressing corner cases and edge scenarios. Continued research and development in this area are crucial for realizing improvements in road safety, with the potential to deliver substantial socio-economic benefits through a reduction in traffic-related fatalities and injuries.

A primary limitation of autonomous systems in traffic lies in their restricted capacity to detect and respond to distant or small objects, particularly in complex or unpredictable settings. For example, Wu et al. [4] reported that advanced driver-assistance systems (ADAS) often struggle to detect and effectively manage objects that can be as small as 32×32 pixels on a display screen. Such objects occupy minimal space on visual interfaces, creating substantial challenges for both human and machine recognition. This limitation ultimately undermines key advantages of autonomous technologies, such as constant alertness and early reaction, and erodes user confidence, resulting in reluctance to rely on these systems [5].

Several factors hinder the detection of small and distant objects in autonomous systems. First, their limited pixel information provides insufficient features for AI models to learn effectively. Second, feature ambiguity arises when crucial visual elements (e.g., edges, textures, shapes) are indistinguishable, making it difficult for models to separate objects from the background. Third, many AI models downscale input images to reduce computational overhead, which can exacerbate the problem by discarding essential information and causing small objects to blend into background noise. Addressing these challenges is very important for improving the reliability and accuracy of vehicle autonomous systems in detecting distant and small objects.

This paper addresses the above issues by proposing *D2SO*, a Vision Transformer (ViT)-based model designed explicitly to improve the detection of small and distant objects in vehicle autonomous systems. The backbone of *D2SO* is the Segment Anything Model (SAM) [6], which we fine-tune to detect objects smaller than 24×24 pixels—sizes that are also challenging for the human eye to discern [7]. To achieve this, we carefully preprocess training and test datasets to include only small and distant objects, then feed them into SAM for fine-tuning. In this process, SAM's accurate pixel-level segmentation capabilities across entire images are leveraged, while model parameters are updated specifically to detect small, distant objects. Consequently, the enhanced models can identify extremely small objects with improved responsiveness in diverse operational contexts. Promptly recognizing small and distant objects is integral to both safety and efficiency in autonomous systems.

We summarize our contributions as follows:

1) We introduce *D2SO*, a ViT-based architecture fine-tuned from SAM, to improve the detection of small and distant

objects for vehicle autonomous systems.
2) We extensively evaluate the performance of *D2SO*, demonstrating its robustness and reliability in safety-critical scenarios, and show that our approach substantially outperforms baselines such as YOLO v11 Segmentation, U-Net, and SegFormer.
3) We highlight how minimal visual cues can be effectively utilized by AI systems for reliable decision-making in settings with sparse or ambiguous visual information.

The organization of this paper is structured to detail the development and evaluation of *D2SO*. We begin with a review of related work to contextualize our research within the existing literature. The methodology section follows, outlining the research problem formulation and detailing the SAM architecture. We then describe the fine-tuning process and the overall *D2SO* framework. The evaluation section covers the experimental setup, the dataset used, and presents results and analysis, including visual results and comparison with baselines. Finally, we conclude the paper and propose directions for future work to further enhance *D2SO*'s capabilities.

II. Related Work

Detecting small and distant objects in vision-based autonomous systems presents challenges due to limited pixel information, feature ambiguity, and downscaling-induced detail loss. Research explores multiple approaches, including traditional object detection, ViTs, SAM in segmentation, and benchmark datasets for small object detection. Each of these approaches contributes insights into object detection but also presents limitations.

Traditional convolutional neural networks (CNNs) apply to object detection and segmentation. Detection models such as Faster R-CNN [8], YOLO [9], and SSD [10] detect medium-to-large-sized objects. Segmentation models like U-Net [11] and DeepLab [12] classify pixels. These approaches struggle with small and distant objects due to hierarchical feature extraction, which removes small object details through downsampling operations like pooling layers and strided convolutions. Studies [13] show that small object detection accuracy remains lower than that of larger objects due to limited feature representation. CNN-based models fail to distinguish small objects from background clutter, especially in urban environments [14]. Feature pyramid networks (FPNs) and attention-based methods [15] attempt to improve detection, but challenges persist. Since CNN-based methods struggle with small object detection, alternative architectures such as ViTs offer potential improvements.

ViTs model long-range dependencies in object detection. Unlike CNNs, which rely on localized receptive fields, ViTs use self-attention mechanisms to capture global contextual relationships [16]. This allows ViTs to detect distant objects from subtle contextual cues. Detection Transformer (DETR) [17] reformulates object detection as a direct set prediction problem. DETR removes hand-crafted anchor boxes and employs a transformer-based architecture. However, DETR struggles with small objects due to coarse feature representations. Deformable DETR [18] introduces adaptive attention mechanisms to focus on relevant regions, improving small object detection. Research explores hybrid architectures that integrate ViTs with CNN feature extractors to enhance fine-grained feature representation [19]. ViTs require optimization for real-time autonomous systems. Given that ViTs offer global feature modeling, segmentation-specific models such as SAM could further refine small object detection.

SAM, developed by Meta AI [6], advances image segmentation with prompt-based segmentation for object identification. SAM applies to domains such as medical imaging [20] and satellite imagery analysis [21]. Its adaptability to prompt types—points, bounding boxes, and textual descriptions—supports object detection and segmentation. SAM focuses on general segmentation tasks rather than small and distant object detection in autonomous systems. Fine-tuning SAM for specific applications remains an open problem. Studies show that adapting SAM requires domain-specific training datasets and loss functions [22]. We fine-tune SAM on small object datasets and integrate it into a ViT-based detection pipeline to improve performance in autonomous systems. While SAM improves segmentation, evaluating its effectiveness in small object detection requires appropriate benchmarks and performance metrics.

Evaluating small object detection models requires curated datasets and appropriate performance metrics. The COCO dataset [23] includes small object annotations, but object size distribution skews toward larger instances. The DOTA dataset [24], which focuses on aerial imagery, contains a higher proportion of small objects and provides a benchmark for small-scale detection models. Metrics such as mean Average Precision (mAP) assess object detection performance, with adaptations for small object detection. COCO defines an *AP-small* metric for objects smaller than 32×32 pixels [23]. YOLO v11 and SegFormer improve mAP scores for general detection [25], but small object performance remains suboptimal. Our work addresses this gap by fine-tuning SAM for small object segmentation and improving detection accuracy in urban settings.

Therefore, object detection frameworks, including CNN-based and ViT-based models, improve accuracy for medium-to-large objects but remain limited for small and distant ones. SAM contributes prompt-based segmentation capabilities but lacks broad deployment in this context. Benchmark datasets and evaluation metrics further highlight the gap. Our work responds by combining SAM and ViT strategies to improve small and distant object detection for autonomous systems. This integration enhances early hazard recognition and improves decision-making accuracy in safety-critical driving scenarios. By leveraging ViT's global attention and SAM's prompt-aware segmentation, the system can detect objects with minimal visual cues and low pixel occupancy. This capability addresses limitations posed by downscaling and feature ambiguity, enabling more reliable performance in urban scenes with dense and complex environments.

III. Methodology

In this section, we first formulate the research question. Next, we provide an overview of the SAM by detailing its architecture. Finally, we present the design of *D2SO*, which enhances SAM's capability to detect small and distant objects.

A. Problem Formulation

Detecting distant objects is crucial for autonomous systems because early hazard detection can prevent accidents and enhance overall safety. In the context of autonomous vehicles, 'distant' typically refers to objects that are far enough away to allow sufficient time for the vehicle to react, usually several hundred meters depending on the vehicle's speed and stopping distance. In this paper, we consider distant objects as fewer pixels occupation in the visual input system. Accurate detection of these distant objects allows the system to plan and execute safe maneuvers well in advance.

Let the visual input to the autonomous system be represented as an image I with dimensions $H \times W$, where H is the height and W is the width of the image.

1) **Pixel Occupancy of Objects:** Let O_i represent an object in the scene, and $\mathcal{P}(O_i) \subseteq I$ denote the pixels occupied by O_i. The pixel occupancy of O_i can be defined as:

$$|\mathcal{P}(O_i)| = \sum_{(x,y)\in I} \mathbb{1}[(x,y) \in \mathcal{P}(O_i)],$$

where $\mathbb{1}[\cdot]$ is the indicator function.

2) **Distant Object Definition:** An object O_i is classified as distant if:

$$|\mathcal{P}(O_i)| < \tau,$$

where τ is a predefined threshold for pixel occupancy corresponding to distant objects. In this paper, we set $\tau < 24\times24$ (pixel sizes that are challenging for both driver-assistance systems [4] and human eyes to discern [7]), and the height and width of O_i are both less than 24 simultaneously.

3) **Detection Objective:** The objective is to maximize the detection accuracy for distant objects. Define the detection accuracy for an object O_i as $\mathcal{A}(O_i)$, which can be measured by the Intersection over Union (IoU):

$$\mathcal{A}(O_i) = \frac{|\mathcal{P}(O_i) \cap \mathcal{P}^*(O_i)|}{|\mathcal{P}(O_i) \cup \mathcal{P}^*(O_i)|} \tag{1}$$

where $\mathcal{P}^*(O_i)$ is the ground truth pixel occupancy for O_i.

B. Segment Anything Model Architecture

When designing our framework for detecting small and distant objects, we selected SAM as the backbone—a highly flexible, open-source image segmentation model renowned for its prompt-based segmentation capabilities—to meet the stringent accuracy, speed, and responsiveness requirements of real-time object detection in autonomous systems [6]. As shown in Figure 1, the SAM architecture mainly consists of three primary stages: Image Encoding, Prompt Encoding, and Mask Decoding.

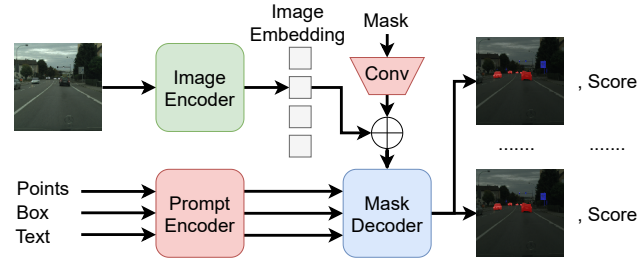

Fig. 1: SAM architecture overview [6].

The image encoder E_{img} processes the image I to produce an image embedding $E \in \mathbb{R}^{N\times d}$, where N is the number of patches (or spatial tokens), and d is the embedding dimension:

$$E = E_{\text{img}}(I)$$

The image encoder E_{img} is typically a Vision Transformer (ViT), which splits the image into patches and captures long-range dependencies.

The prompt encoder E_{prompt} processes user-provided prompts to guide segmentation. Prompts can include: 1) Points: $P \in \mathbb{R}^{K\times2}$, where K is the number of points (e.g., coordinates (x, y)); 2) Boxes: $B = [x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}]$, specifying regions of interest; 3) Text: natural language instructions or descriptions. The prompt encoder E_{prompt} maps prompts into a latent space:

$$Q = E_{\text{prompt}}(\text{prompts}),$$

where $Q \in \mathbb{R}^{M\times d}$, and M is the number of prompt tokens. The encoded prompts Q are designed to interact with the image embedding E during the decoding stage. Note that masks can also be viewed as a type of prompt in SAM. Different from points, boxes, and text, they are embedded using convolutions and summed element-wise with the image embedding (see the Conv component in Figure 1).

The mask decoder D fuses the image embeddings E and the prompt embeddings Q to generate segmentation masks through cross-attention:

$$Z = \text{CrossAttention}(E, Q),$$

where $Z \in \mathbb{R}^{N\times d}$ is the fused feature representation. Then, a convolutional layer processes the fused representation Z to generate the segmentation mask M:

$$M = \sigma(\text{Conv}(Z))$$

where $M \in \mathbb{R}^{H\times W}$ and σ is the sigmoid activation function, which produces mask probabilities. For each mask, the decoder computes a confidence score S, indicating the quality of the predicted mask. Note that mask decoder D can output multiple segmentation masks M along with scores S, i.e., $(M_i, S_i)\}_{i=1}^{n}$ where n is the number of masks generated.

C. D2SO Framework

Fine-tuning the SAM model is necessary to enhance its detection capabilities for small and distant objects, which are often critical in autonomous systems for early hazard detection. The original SAM model, while proficient in pixel-level segmentation, was not specifically optimized for identifying tiny, remote objects to prevent accidents and ensure safety. In *D2SO* framework, we define an object O_i is classified as distant if $|\mathcal{P}(O_i)| < \tau$, where τ is a predefined threshold for pixel occupancy corresponding to distant objects. For *D2SO*, we set $\tau = 24 \times 24$, ensuring that both the height and width of O_i are less than 24 pixels simultaneously.

To fine-tune SAM for detecting small and distant objects, we first prepare a specialized training dataset that focuses specifically on these types of objects. As shown in Figure 2, we then freeze the points and text prompts, allowing only the bounding box prompts and the ground truth masks M_g to be input into the mask decoder D. In this process, the image encoder E_{img} generates feature embeddings, while the prompt encoder E_{prompt} encodes the bounding box prompts. These embeddings are then fused within the mask decoder D to accurately predict the segmentation mask M_p.

1) Training Dataset Preparation: This data processing workflow prepares labeled images and ground truth masks for further use by filtering, resizing, mapping, and organizing the data. The original images, which are large in size, are first processed to filter out objects with regions smaller than 9216 pixels, based on their pixel count in the instance masks, and simplifying them by truncating their values. These images are then resized so that their largest dimension is reduced to 512 pixels using the nearest-neighbor method, preserving ID fidelity and aspect ratios. As a result of this resizing, the previously filtered objects now appear even smaller, often shrinking to dimensions less than 24×24 pixels in the resized images. Finally, the resized images are center-cropped to a fixed size of 256×256 pixels for uniformity. Specific instance IDs are filtered or remapped according to predefined criteria, and bounding boxes are extracted from the ground truth masks to highlight relevant regions. These bounding boxes are further refined by applying random perturbations, expanding or shrinking the boundaries by a small amount up to 20 pixels, to improve robustness.

To adapt the data for use with the SAM, the processed images and bounding box prompts are formatted to meet SAM's compatibility requirements. This involves encoding the images and bounding boxes into a structured format where the images are resized to a fixed input size, and the bounding box coordinates are normalized relative to the image dimensions. For example, if an original bounding box coordinate is $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ and the image dimensions are $[H, W]$, the normalized coordinates become $\left[\frac{x_{\min}}{W}, \frac{y_{\min}}{H}, \frac{x_{\max}}{W}, \frac{y_{\max}}{H}\right]$. This normalization ensures the bounding box scales correctly across varying image resolutions. These inputs are converted into tensors to ensure compatibility with SAM's neural network architecture. By aligning the bounding box prompts and images with SAM's input specifications, the workflow guarantees the data can be seamlessly integrated into the model.

Algorithm 1: Training Loop Using Adam Optimizer

```
Input: dataset containing batches (I, M_g)
Output: Optimized model parameters
for each epoch do
    for each batch (I, M_g) in dataset do
        M_p ← ModelForward(I, M_g)
        L ← DiceLoss(M_g, M_p, ε_l)
        Backpropagate(L)
        UpdateParameters(Adam)
    end
end
```

2) Fine Tuning SAM: To achieve high IoU $\mathcal{A}(O_i)$ accuracy, we adopt the dice loss function, which is particularly effective for segmentation tasks where the balance between different classes, to fine tune SAM. SAM's image encoder E_{img} and prompt encoder E_{prompt} generate feature embeddings based on the input image and box prompts. These embeddings are fused in the mask decoder D to predict the segmentation mask M_p. The predicted mask M_p is compared against the ground truth mask M_g using the Dice Loss. The loss is computed as:

$$\text{DiceLoss}(M_g, M_p, \epsilon_l) = 1 - \frac{2\sum_i M_{p,i} M_{g,i} + \epsilon_l}{\sum_i M_{p,i} + \sum_i M_{g,i} + \epsilon_l}$$

where ϵ_l is a small constant added to prevent division by zero, $M_{p,i}$ is the predicted value for pixel i in the predicted segmentation mask M_p, and $M_{g,i}$ is the ground truth value for pixel i in the segmentation mask M_g. The numerator represents twice the intersection between the predicted and ground truth masks. The denominator represents the total number of pixels in both masks.

The Dice Loss is backpropagated to update the weights of the mask decoder D. Minimizing this loss ensures that the predicted mask M_p closely aligns with the ground truth M_g, enhancing the model's segmentation accuracy. We utilize the Adam optimizer to refine the performance of our model for its ability to adapt learning rates for individual parameters and its robustness in handling sparse gradients on noisy problems. We set the learning rate at 1×10^{-5} to balance the convergence speed and stability of training. Importantly, we do not apply weight decay in this setup, as we focus on optimizing performance without regularizing the parameter scale. This decision is based on our preliminary experiments, which indicated that excluding weight decay promotes better fine-tuning of the decoder's parameters in our specific application context. The configuration parameters are set as follows:

$$\text{Adam}(\eta, \beta_1, \beta_2, \epsilon_a)$$

where η represents the learning rate, and β_1 and β_2 are the exponential decay rates for the moment estimates. ϵ_a is a small constant added to the denominator for numerical stability. The iterative training process for each epoch using the Adam optimizer is illustrated as Algorithm 1.

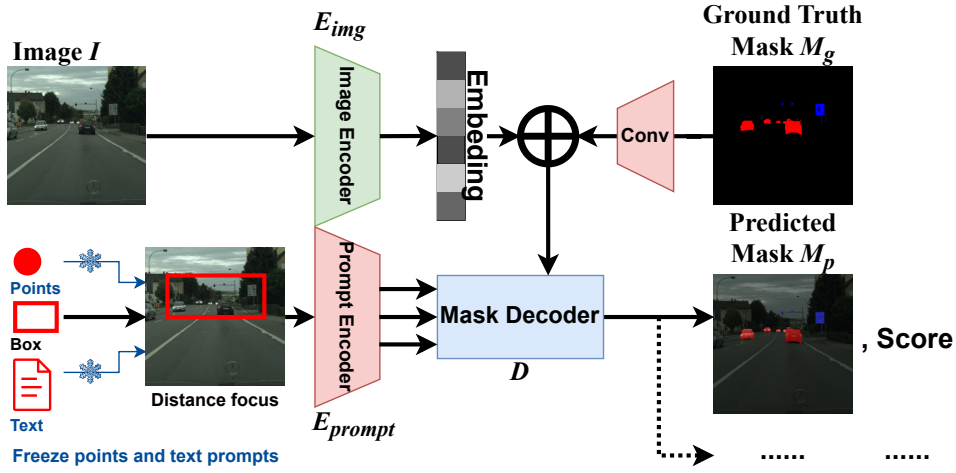

Fig. 2: Fine-tuning the SAM model to enhance its ability to detect small and distant objects. We freeze the point and text prompts while utilizing box prompts and ground truth masks to train an effective mask decoder for accurate object detection.

3) Multi-Class Detection in D2SO: The *D2SO* framework performs multi-class segmentation on images, identifying objects across the three most common distinct categories: static structures, humans, and vehicles. The process begins with data loading and preprocessing, followed by model inference. A unique aspect of *D2SO* is the integration of box prompts during preprocessing, which enhances the detection accuracy of distant objects by emphasizing their features.

D2SO utilizes three independent instances of SAM models, denoted as $\text{SAM}_{\text{structures}}, \text{SAM}_{\text{humans}}, \text{SAM}_{\text{vehicles}}$, each fine-tuned for detecting specific object classes:

$$\mathcal{M}_c = \{\text{SAM}_{\text{structures}}, \text{SAM}_{\text{humans}}, \text{SAM}_{\text{vehicles}}\},$$

where $c \in \{\text{structures, humans, vehicles}\}$ represents the class label. Each model is initialized with pretrained weights, W_c, and parameters loaded from pre-defined paths. To ensure computational efficiency, specific layers in the SAM architecture, such as the image encoder E_{img} and prompt encoder E_{prompt}, are frozen during inference to reduce unnecessary computation and memory usage.

The inference phase processes each batch of input images, where each image I_i is preprocessed with box prompts to enhance object saliency. For each class c, the corresponding SAM model SAM_c predicts the segmentation mask M_p^c as:

$$M_p^c = \text{SAM}_c(I_i, P_c),$$

where P_c represents the box prompts associated with class c. The predicted mask M_c is further refined using predefined pixel count thresholds τ, ensuring that only valid detections are retained:

$$\mathcal{P}(O_i) = |\text{Mask}_c|, \quad \text{Valid if } \mathcal{P}(O_i) > \tau.$$

For visual feedback, the predicted masks are overlaid onto the original input image I_i, resulting in a color-coded visualization where red represents vehicles, green denotes humans, and blue indicates static structures. This approach allows for an intuitive and immediate understanding of the segmentation results, facilitating easier interpretation and analysis of the model's performance in diverse scenarios.

IV. Evaluation

A. Experiment Setup

We conduct evaluations on a server equipped with a 12-core CPU and dual NVIDIA A5500 GPUs, each with 24GB of memory. The models are evaluated using PyTorch version 2.0.1. For optimization, we configure the Adam optimizer with a learning rate η of 1×10^{-5}, and exponential decay rates β_1 and β_2 set at 0.9 and 0.999, respectively. Additionally, ϵ_a, a small constant added for numerical stability, is set at 1×10^{-8}. This hardware and software setup provides the necessary computational power and resources to effectively train and evaluate *D2SO*, ensuring it can process and analyze complex urban scene data efficiently.

B. Dataset

The dataset used in our experiments is the Cityscapes [26] dataset, renowned for its rich annotations and urban scene focus. It includes stereo video sequences captured across 50 cities in Germany and neighboring countries during spring, summer, and fall, deliberately excluding adverse weather conditions. Cityscapes provides 5,000 finely annotated images and 20,000 coarsely annotated images, enabling the fine-tuning of SAM with a mix of high-quality and large-scale weakly-labeled data. Captured with an automotive-grade stereo camera, the images offer high resolution and dynamic range, essential for detecting distant objects critical to *D2SO* 's operation. The dataset's pixel-level precision and instance-level annotations for dynamic objects like humans and vehicles are vital for *D2SO*, enhancing its ability to identify and respond to distant objects, thereby improving situational awareness in autonomous systems.

Fig. 3: This collage displays six samples, each consisting of three panels: input images, ground truth images, and predicted images. In these samples, *D2SO* specifically targets the detection of distant objects, defined as those smaller than 24×24 pixels in size. The predicted outputs are color-coded to enhance clarity: vehicles are marked with red, static structures with green, and humans with blue. This visual representation emphasizes the model's capability to accurately identify and differentiate between various object classes at a distance.

C. Baselines

To demonstrate the effectiveness of *D2SO*, we select SegFormer, YOLO v11 Segmentation, and U-Net as baselines in our experiments.

- **SegFormer** [27]: SegFormer is a semantic segmentation framework that integrates a hierarchically structured Transformer encoder with a lightweight multilayer perceptron decoder. This architecture avoids the need for positional encoding and produces multiscale features that adapt well to varying resolutions. SegFormer is efficient and accurate, making it suitable for segmenting semantically labeled objects in diverse scenarios. It has demonstrated strong performance in benchmarks such as ADE20K and Cityscapes, with significant gains in mean Intersection over Union.
- **YOLO v11 Segmentation** [25]: YOLO v11 Segmentation is an object detection and segmentation model designed for efficiency and accuracy in complex environments. It incorporates the C2PSA (Cross-Stage Partial with Self-Attention) module to capture contextual information across multiple layers, enhancing the detection of small and hidden objects. Additionally, the C3k2 block, an optimized CSP bottleneck with two small convolutions, enables YOLO v11 to maintain high accuracy while improving computational efficiency and speed.
- **U-Net** [11]: U-Net is a convolutional network for image segmentation with a 23-layer architecture, comprising a contraction path for feature extraction and an expansive path for resolution restoration. This model balances global context extraction and local detail refinement, making it effective for segmentation tasks, even with limited data.

D. Qualitative Comparison to Baseline Methods

The qualitative performance of *D2SO* was evaluated across urban scenes in Cityscapes under varying environmental conditions. Six sample results are illustrated in Figure 3, where each scene is processed and presented in three panels: input images, ground truth annotations, and predictions from the models (*D2SO*, YOLO v11 Segmentation, U-Net, and SegFormer). The ground truth annotations identify object classes using shades of gray, while predictions are color-coded: red for vehicles, green for humans, and blue for static structures. The visual comparison highlights *D2SO* 's ability to detect small objects smaller than 24×24 pixels in urban settings. The overlap between *D2SO*'s predicted masks and the ground truth confirms alignment with annotated regions. Other models miss several instances or misclassify object boundaries. *D2SO* marks targets with clear shape and location. The method separates small objects from the background without relying on large pixel regions. The figure shows detection results that remain stable across different positions, densities, and types of objects. This behavior supports detection tasks in scenes where space is limited and occlusions are frequent.

E. Quantitative Comparison to Baseline Methods

The quantitative performance of *D2SO*, SegFormer, YOLO v11 Segmentation, and U-Net was compared using metrics such as accuracy, precision, recall, and F1-score. To calculate accuracy for a single object class in object detection, we first evaluate the Intersection Over Union (IoU) for each predicted bounding box against its corresponding ground truth box. IoU is calculated using Equation 1. If $\mathcal{A}(O_i) \geq 0.5$, the prediction is considered correct. The total number of correct predictions for the class is then divided by the total number of samples (ground truth instances) for that class to compute accuracy. Additionally, if all three classes in an image are predicted correctly, the image is considered all correct. The overall accuracy is then calculated as the ratio of totally correct images to the total number of images in the dataset. Table I summarizes these results for vehicles, humans, static structures, and overall performance.

- **Vehicle Detection:** *D2SO* achieves the highest performance, with an accuracy of 97.41%, precision of 97.41%, recall of 100%, and F1-score of 98.69%. In comparison, all baselines (SegFormer, YOLO v11 Segmentation, and U-Net) show significantly lower metrics, with U-Net being the closest competitor.
- **Human Detection:** For detecting humans, *D2SO* outperforms all baselines with an accuracy of 80.91%, precision of 90.56%, recall of 85.08%, and F1-score of 87.73%. These results highlight *D2SO* 's superior ability to distinguish humans from other objects, essential for pedestrian detection in autonomous driving systems.
- **Static Structure Detection:** In detecting static structures, *D2SO* achieves an accuracy of 82.52%, precision of 97.77%, recall of 81.72%, and F1-score of 89.00%. It outperforms SegFormer, YOLO v11 Segmentation, and U-Net, demonstrating its effectiveness in identifying buildings, road signs, and trees that provide critical environmental context.
- **Overall Performance:** *D2SO* 's overall metrics are as follows: accuracy of 72.49%, precision of 86.13%, recall of 70.95%, and F1-score of 77.86%. These values consistently outperform the baselines, reinforcing its capability for robust segmentation in complex urban environments.

Model	Accuracy	Precision	Recall	F1-Score
		SegFormer		
Vehicle	47.74	47.29	96.97	63.47
Human	51.27	48.37	58.62	53.00
Structures	63.44	73.39	76.06	74.69
		YOLO_v11_Seg		
Vehicle	66.88	68.67	82.55	74.97
Human	63.54	63.49	74.34	68.49
Structures	35.77	35.76	71.30	47.63
All	41.52	60.30	41.11	48.88
		U-Net		
Vehicle	89.25	91.41	87.25	89.28
Human	80.05	71.71	74.76	73.20
Structures	48.57	70.41	46.28	55.85
All	76.27	82.75	70.75	76.28
		D2SO		
Vehicle	**97.41**	**97.41**	**100**	**98.69**
Human	**80.91**	**90.56**	**85.08**	**87.73**
Structures	**82.52**	**97.77**	**81.72**	**89.00**
All	**72.49**	**86.13**	**70.95**	**77.86**

TABLE I: Accuracy, precision, recall, and F1-score for all objects, vehicles, humans, and static structures, comparing the performance of SegFormer, YOLO v11 Segmentation, U-Net, and *D2SO*.

The quantitative and qualitative experimental results show that *D2SO* outperforms baseline methods in accuracy, precision, recall, and F1-score across all object classes. The model detects small and distant objects with consistency. It handles scale variation, background clutter, and class imbalance. The system processes input and produces output without delay. The method fits deployment settings that need fast and stable results. The design supports adaptation to new environments and tasks without changes to core components.

V. Conclusion

This paper introduced *D2SO*, a Vision Transformer-based framework fine-tuned from the Segment Anything Model (SAM) to detect small and distant objects for vision-based autonomous systems. By focusing on objects smaller than 24×24 pixels, *D2SO* addresses key challenges in early hazard detection and significantly outperforms baseline models, including SegFormer, YOLO v11 Segmentation and U-Net, in accuracy, precision, recall, and F1-score. Its robust performance highlights its potential to improve autonomous system reliability and safety in urban environments. Future work will focus on extending *D2SO* to handle more diverse environmental conditions and optimizing it for resource-constrained platforms, broadening its applicability.

VI. Acknowledgments

This work was supported in part by National Science Foundation (NSF) grants NSF 2505686, NSF CNS-2231519, and DUE-2225229.

References

[1] D. Garikapati and S. S. Shetiya, "Autonomous vehicles: Evolution of artificial intelligence and the current industry landscape," *Big Data and Cognitive Computing*, vol. 8, no. 4, p. 42, 2024.

[2] World Health Organization, "Global status report on road safety 2023." https://www.who.int/publications/i/item/WHO-GSR-2023, 2023. Accessed: 2023-06-27.

[3] *International Journal of Science and Engineering Applications*, p. 1–9, May 2023.

[4] B.-X. Wu, V. M. Shivanna, H.-H. Hung, and J.-I. Guo, "Concentratenet: Multi-scale object detection model for advanced driving assistance system using real-time distant region locating technique," *Sensors*, vol. 22, no. 19, p. 7371, 2022.

[5] M. Hou, G. Ho, and D. Dunwoody, "Impacts: A trust model for human-autonomy teaming," *Human-Intelligent Systems Integration*, vol. 3, no. 2, pp. 79–97, 2021.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

[7] S. Li, J. Hu, X. Chai, and Y. Peng, "Image recognition with a limited number of pixels for visual prostheses design," *Artificial organs*, vol. 36, no. 3, pp. 266–274, 2012.

[8] 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.

[9] 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*, pp. 779–788, 2016.

[10] 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*, pp. 21–37, Springer, 2016.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015*, pp. 234–241, Springer, 2015.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[13] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310–7311, 2017.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[15] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] 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*, pp. 213–229, Springer, 2020.

[18] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[19] 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*, pp. 10012–10022, 2021.

[20] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on mri," *Journal of magnetic resonance imaging*, vol. 49, no. 4, pp. 939–954, 2019.

[21] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE geoscience and remote sensing magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[22] C. Zhang, F. D. Puspitasari, S. Zheng, C. Li, Y. Qiao, T. Kang, X. Shan, C. Zhang, C. Qin, F. Rameau, *et al.*, "A survey on segment anything model (sam): Vision foundation model meets prompt engineering," *arXiv preprint arXiv:2306.06211*, 2023.

[23] 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*, pp. 740–755, Springer, 2014.

[24] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

[25] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024.

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12077–12090, 2021.