Demo Abstract: Edge-Cloud Switched Image Segmentation for Autonomous Vehicles

Siyuan Zhou
siyuan.zhou@ntu.edu.sg
Nanyang Technological University
Singapore

Duc Van Le
levanduc@ieee.org
Nanyang Technological University
Singapore

Rui Tan
tanrui@ntu.edu.sg
Nanyang Technological University
Singapore

ABSTRACT

Existing autonomous vehicles have not utilized the cloud computing for execution of their deep learning-based driving tasks due to the long vehicle-to-cloud communication latency. The increasing data transmission speed of the commercial mobile networks sheds light upon the feasibility of using the cloud computing for autonomous driving. In this demo, we introduce the design and implementation of ECSeg, an edge-cloud switched image segmentation system that dynamically selects between the edge and cloud to execute deep learning-based semantic segmentation models. This enables real-time understanding of a vehicle's visual scenes while adapting to dynamic wireless conditions and changing environments.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**.

KEYWORDS

Image segmentation, cloud-assisted system, autonomous vehicles

ACM Reference Format:
Siyuan Zhou, Duc Van Le, and Rui Tan. 2025. Demo Abstract: Edge-Cloud Switched Image Segmentation for Autonomous Vehicles. In *The 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys '25), May 6–9, 2025, Irvine, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3715014.3724378

1 INTRODUCTION

Autonomous vehicles (AVs) have substantial potential to mitigate traffic congestion, enhance road safety, and curtail carbon emissions. Deep learning (DL) has been increasingly employed for various driving tasks of the AVs. For example, the DL models [2] can be used for the vehicles to understand their visual driving scenes correctly, facilitating the safe driving navigation and accurate collision avoidance. To avoid the long latency of data transmission, the commercial AV platforms (e.g., Apollo [1]) are often equipped with the resource-limited edge computing devices to directly execute the DL-based autonomous driving tasks on the vehicles. Meanwhile, the execution of deep models often requires high demand on computing resources. Thus, current AV design strategies adopt customized lightweight, on-board deep models [5] which can be executed by the edge devices in real time to achieve autonomous driving. This design choice compromises the accuracy of the deep models. A possible approach to address this problem is to increase the computing capabilities of the edge devices which allow implementation of complex deep models with high accuracy. However, the powerful computing devices are energy-intensive, which reduces the vehicle's battery system lifetime. It is also challenging for the vehicle's heat dissipation system to handle the tremendous amount of heat dissipated by the energy-intensive computing devices [4].

Compared with the edge devices, the cloud servers can provide the AVs with sufficient computing capabilities without power limitation and heat dissipation issues. In this demo, we present the design of an edge-cloud switched image segmentation system, called *ECSeg* which aims to provide pixel-level understanding of the vehicle's visual driving scenes in real time. Specifically, ECSeg switches between different deep models to obtain the segmentation result of each image frame captured by the AV's camera before a certain deadline (e.g., the time when the next image frame is captured). To achieve the goal, ECSeg has the following two processing options. First, *edge processing option* executes a lightweight convolutional neural network (CNN) model locally on the AV's edge computing device to obtain the segmentation results of the image frames. Second, *cloud processing option* compresses the raw images and transmits them to a cloud server via the mobile network. Then, the cloud server executes an advanced CNN model to process the images and sends the results back to the vehicle.

However, due to the cloud data transmission latency, the vehicle may not always obtain the segmentation result of a transmitted image frame before the image's processing deadline. To mitigate the long cloud latency issue, the cloud processing option allows the vehicle to only spend a certain time period to wait for the cloud result of the current image frame (i.e., current driving scene). We define the latest frame among the previous frames whose cloud segmentation results have already arrived at the vehicle as the source frame. If the vehicle does not receive the cloud result of the current frame after the waiting period, it will use the segmentation result of the source frame to interpolate the result of the current frame. Specifically, we develop an optical flow-based approach [3] to propagate the segmentation result of the source frame to the current frame.

2 SYSTEM OVERVIEW

Fig. 1 overviews the design of ECSeg which has two options: edge and cloud processing options for image segmentation as follows.

■ **Edge processing option:** This option executes a segmentation model locally to obtain the segmentation results of the captured image frames on the AV's edge platform. Given the limited computing resources of the edge platform, ECSeg employs a lightweight

SenSys '25, May 6–9, 2025, Irvine, CA, USA

ACM ISBN 979-8-4007-1479-5/25/05.
https://doi.org/10.1145/3715014.3724378

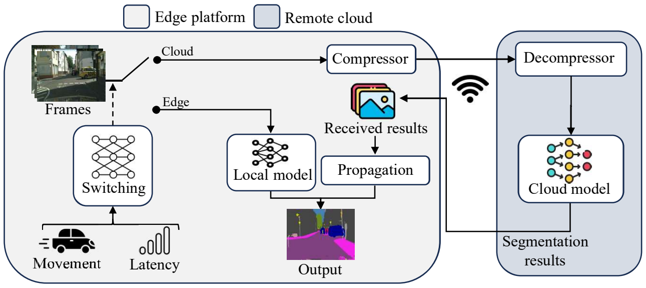

Figure 1: Design overview of ECSeg.

CNN-based image segmentation model as the local model such that the image segmentation result of each image can be always obtained before its deadline.

■ **Cloud processing option:** This option follows a streaming mode to continuously transmit the image frames from the vehicle to the cloud server via a mobile network. To reduce the communication overheads, we implement a JPEG approach to compress each image before transmitting it to the cloud server. Upon receiving the image data, the cloud server employs a JPEG decompressor to reconstruct the original image. Then, the cloud server executes the cloud model to process the reconstructed image. To achieve high image segmentation accuracy, an advanced CNN-based model with large size is implemented as the cloud model in the cloud server. Finally, the cloud image segmentation result is sent back to the vehicle.

Due to the long vehicle-to-cloud communication latency, the cloud segmentation result of an image frame may not arrive at the vehicle before the deadline. Thus, we also develop a propagation approach which uses the received cloud result of a previous frame as input to obtain the segmentation result for the current frame.

■ **Switching:** The edge processing option can always provide the image segmentation result within the deadline. However, it suffers from low segmentation accuracy due to the use of the lightweight model. In contrast, the cloud processing option can execute an advanced model to achieve high accuracy, but has high latency uncertainty due to the dynamic vehicle-to-cloud communication latency. Due to the vehicle movement and poor wireless channel condition, the communication latency can be long, which causes the cloud segmentation results to become stale, decreasing the segmentation accuracy. To maximize the segmentation accuracy, at the edge platform, we implement a DRL-based controller which aims to dynamically switch between the edge and cloud processing options in response to changes of the communication latency and driving scene.

3 DELAY-MITIGATED MIOU

To assess the accuracy of the cloud segmentation results obtained via propagation, we introduce a segmentation accuracy metric, called delay-mitigated mean intersection over union (mIoU). Fig. 2 shows an example of delay-mitigated mIoU, where the i^{th} image is transmitted to the cloud and its cloud result arrives at the vehicle within the interval between the j^{th} and $(j+1)^{th}$ images. Due to the vehicle's movement, the scene captured in the j^{th} image may shift from the i^{th} image, causing the received cloud result to mismatch with the j^{th} image's ground truth. We adopt the propagation method to mitigate the mismatch. As a result, the delay-mitigated mIoU for the j^{th} image is the mIoU between the propagated i^{th} cloud result and the ground truth of the j^{th} image. When the pixel content of the i^{th} image differs significantly from that of the j^{th} image, the delay-mitigated mIoU may decrease.

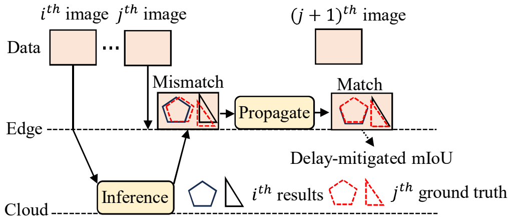

Figure 2: Delay-mitigated mIoU.

4 RESULTS AND DEMONSTRATION

Evaluation results. We utilize a laptop equipped with an NVIDIA GeForce RTX 3080 Ti GPU as the edge platform mounted in the vehicle. The cloud server is prototyped using a tower server with an RTX 8000 GPU and an Intel Xeon Gold 6246 CPU located in a university server room. ECSeg achieves 0.612 delay-mitigate mIoU on the street scene images under the dynamic mobile network latency and driving environments.

Demonstration. To demonstrate ECSeg's effectiveness, we set up a real-time visualization system on the edge device using pre-collected image and latency data. The data is stored locally and displayed on a visual interface. Specifically, to simulate cloud processing, we use pre-collected cloud results and corresponding cloud latency traces. Similarly, edge processing is simulated using pre-collected edge results. The DRL agent dynamically selects between edge and cloud processing based on changing conditions. The switching decisions and corresponding outcomes are visually presented to illustrate ECSeg's adaptive capability.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-006) and Singapore Ministry of Education under its AcRF Tier-1 grant RT14/22.

REFERENCES

[1] 2024. Apollo. https://www.apollo.auto
[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2017), 834–848.
[3] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast optical flow using dense inverse search. In *ECCV*.
[4] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. 2020. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet Things J.* 8, 8 (2020), 6469–6486.
[5] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. 2018. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*.