IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Semantic Hierarchy-Guided Adversarial Attack for Autonomous Driving

Gwangbin Kim and SeungJun Kim, *Member, IEEE*

***Abstract*—Autonomous vehicles employ semantic segmentation as a foundational component for perception and scene understanding, upon which driving decisions can be informed. Despite their performance, these deep learning models remain susceptible to subtle input perturbations that can cause severe deviation in model output. To enhance algorithmic robustness by examining such vulnerabilities, researchers have investigated adversarial examples, which are visually imperceptible yet can severely degrade model performance. However, traditional attacks produce arbitrary misclassifications that ignore semantic relationships, making the attack less effective. This letter introduces a semantic hierarchy-guided adversarial attack (SHAA), a white-box adversarial attack against semantic segmentation for autonomous driving. By combining semantic hierarchy and adaptive momentum-based updates across the image, SHAA produces semantically nontrivial yet highly effective perturbations. The SHAA method exposes deeper vulnerabilities with a higher attack success rate in semantic segmentation than existing methods, aiding the design of a more resilient perception system for autonomous vehicles.**

***Index Terms*—Adversarial attack, automated vehicles, autonomous driving, semantic segmentation.**

I. Introduction

MODERN deep learning models have empowered robot vision capabilities, allowing autonomous systems to understand complex environments through visual scene understanding. However, these models, which learn hierarchical representations from image pixels, can be vulnerable to subtle input perturbations [1]. Small changes in input images can lead to critical misclassification or segmentation errors, potentially compromising the safety and reliability of autonomous systems. As perception is the initial step for autonomous vehicles to respond and drive accordingly, threats to robust perception must be addressed. These vulnerabilities can arise from various means, both digital and physical [2], including malicious data poisoning [3], evasion [4], physical attacks on infrastructure [5], or adverse lighting and weather conditions [6].

By studying adversarial scenarios, we can guide the development of more robust algorithms [7], ensuring that autonomous vehicles maintain safe operation despite malicious disturbances or data corruption. Such research exposes architectural and procedural weaknesses in models and prompts the development of defensive strategies such as adversarial training, allowing the models to learn useful features from both clean and adversarial domains [8]. In particular, white-box attacks analyze model vulnerabilities by utilizing complete knowledge of model parameters. These analyses, though representing extreme scenarios, identify critical vulnerabilities that inform defense and highlight potential robustness risks. Exploring such attack scenarios contributes to developing more robust autonomous vehicles that can maintain safe operation despite potential threats.

While traditional adversarial attacks generate imperceptible perturbations for image classification, semantic segmentation presents a more complex challenge due to its holistic scene understanding requirements. Previous works have either emphasized attack success rates and producing disruptive segmentation outputs, or focused on creating semantically relevant adversarial examples, but frequently at the expense of one another [9]. In this work, we integrate semantic hierarchies with white-box methods, guiding nontrivial misclassifications using semantic structure while retaining the benefits of pixel-level perturbations. Our approach leverages class relationships to ensure that misclassifications consider the semantic structure of the scene for a stronger attack.

II. Related Works

Adversarial attacks involve crafting subtle perturbations to input data that cause machine learning models to produce incorrect or unexpected outputs. White-box attacks, in particular, assume access to the model architecture and gradients [10]. This represents a worst-case scenario, revealing fundamental robustness limits of the model and guiding the development of defense methods. Using gradient ascent optimization through the model, white-box methods identify optimal perturbation directions to deceive the model [11].

Adversarial methods for robot vision introduced gradient-based approaches, from single-step FGSM [12] to PGD that refines perturbations over steps, to attack image classification models [13]. However, applying these attacks directly to semantic segmentation poses challenges. Segmentation models output dense label maps and naive pixel-level perturbations yield limited attack success due to their robustness [14].

Meanwhile, various methods have been proposed to enhance attack performance through efficient gradient

Received 8 January 2025; accepted 25 May 2025. Date of publication 18 June 2025; date of current version 25 June 2025. This article was recommended for publication by Associate Editor P. Borja and Editor C. D. Santina upon evaluation of the reviewers' comments. This work was supported in part by GIST-MIT Research Collaboration Grant through the GIST in 2025, in part by Artificial Intelligence Graduate School Program through the IITP (Institute of Information Communications Technology Planning Evaluation) under Grant 2019-0-01842, in part by the National Research Foundation of Korea (NRF) through the MSIT under Grant RS-2024-00343397, and in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) through the Ministry of Land Infrastructure and Transport under Grant RS-2023-00256888. *(Corresponding author: SeungJun Kim.)*

The authors are with the Department of AI Convergence, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea (e-mail: gwangbin@gm.gist.ac.kr; seungjun@gist.ac.kr).

Digital Object Identifier 10.1109/LRA.2025.3580923

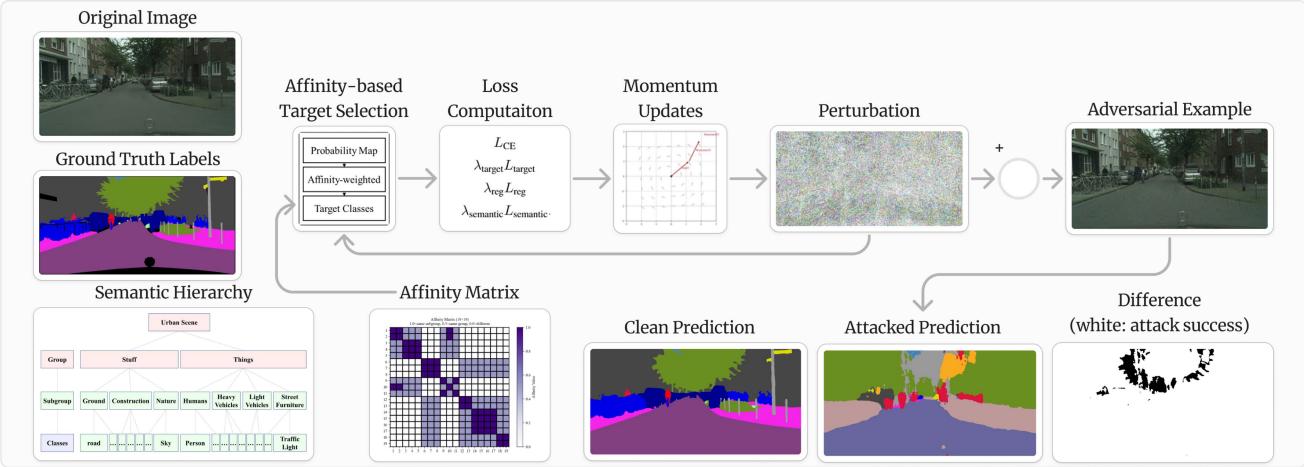


Fig. 1. Overview of the SHAA method. The method generates semantically guided perturbation while maintaining high attack success rates.

exploration and feature manipulation. MI-FGSM [15] incorporates momentum for more stable and transferable perturbations, while TI-FGSM [16] generates adversarial examples robust under spatial transformations. DIM [17] increases input diversity to enhance cross-model success rates, and FIA [18] selectively targets features deemed critical by the model. Though these methods achieve fair success rates in semantic segmentation tasks [11], the attacked results are often semantically trivial, making the attack less effective in their deployment settings.

Alongside efforts to create imperceptible perturbations, recent work has pursued the plausibility of the attacked segmentation maps to evade detection, making attacks harder to identify. Metzen et al. [19] explored perturbations that target specific segmentation as output. Chen et al. [20] demonstrated methods for semantic stealth focused on manipulating specific target objects with heuristics (e.g., vanishing or embedding) while preserving the surroundings. Building on this direction of incorporating semantic considerations, which focused on plausible localized manipulations, SHAA employs an alternative approach to craft perturbations at the scene level. Through direct optimization guided by an explicit semantic structure, SHAA aims to apply controllable, semantic-aware attacks.

In this work, we propose integrating semantic hierarchies with pixel-level perturbations. Our method keeps the effectiveness of pixel-level attacks while considering the scene's semantic structure. This approach creates nontrivial adversarial results, posing deeper challenges for autonomous vehicle perception and defense strategies.

III. Method

Our proposed white-box method, SHAA, uses a semantic hierarchy to produce adversarial examples that degrade segmentation performance while leveraging thematic structure. Rather than causing arbitrary misclassifications, SHAA directs predictions with semantic relationship, leveraging scene structure for nontrivial attack success (Table I). An affinity matrix combined with momentum-based gradient updates and adaptive scaling ensures stable optimization of perturbations and guides attack towards semantically preferred misclassification. Algorithm 1 outlines the procedure, and Fig. 1 illustrates the pipeline of the SHAA method.

TABLE I
Semantic Hierarchy Grouping

Group	Subgroup	Classes
Stuff	Ground	road, sidewalk, terrain
	Construction	building, wall, fence
	Nature	vegetation, sky
Things	Humans	person, rider
	Heavy Vehicles	car, truck, bus, train
	Light Vehicles	motorcycle, bicycle
	Street Furniture	traffic light, traffic sign, pole

A. Notation

- $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$: Input image.
- $\mathbf{y} \in \{0,\dots,K-1\}^{H\times W}$: Ground truth segmentation labels with K classes.
- $f(\mathbf{x})$: Segmentation model outputting logits $\mathbf{z} \in \mathbb{R}^{K\times H\times W}$.
- $\mathbf{p} = \text{softmax}(\mathbf{z}) \in [0,1]^{K\times H\times W}$: Predicted class prob.
- $\delta \in \mathbb{R}^{C\times H\times W}$: Perturbation bounded by $\|\delta\|_\infty \le \epsilon$.
- $\mathbf{g} \in \mathbb{R}^{C\times H\times W}$: Accumulated momentum gradient for update direction.
- α, T, γ, β: Step size, number of iterations, momentum decay, and adaptive scale factor, respectively.
- $\mathbf{g}_{\text{current}}, \mathbf{g}_{\text{prev}}$: Momentum buffers storing current and previous gradients.
- $\lambda_{\text{target}}, \lambda_{\text{reg}}, \lambda_{\text{semantic}}, \lambda_{\text{prev}}$: Weights for targeted misclassification (0.3), regularization (0.1), semantic consistency (0.2), and previous gradient (0.3) terms.
- ℓ_{CE}: Cross-entropy loss function.
- $\text{sign}(\cdot)$: Element-wise sign function.
- $\cos(\theta)$: Cosine similarity between curr. and prev. gradient.
- w_{ik}: the weighted probability of class k for pixel i
- $\mathbf{z}_i$: Logit vector for pixel i.
- $\mathcal{G}$: Set of all groups in the semantic hierarchy.
- $\mathcal{S}_g$: Set of all subgroups within group g.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1: Semantic Hierarchy-Guided Attack.

Input: Model f, image $\mathbf{x}$, labels $\mathbf{y}$, perturbation limit ϵ, step size α, number of steps T
Output: Adversarial image $\mathbf{x}_{\text{adv}}$
Initialize $\delta \leftarrow \mathbf{0}$, $\mathbf{g} \leftarrow \mathbf{0}$, $\mathbf{g}_{\text{prev}} \leftarrow \mathbf{0}$, $\beta \leftarrow 1.2$;
Initialize $L_{\text{best}} \leftarrow -\infty$, $\delta_{\text{best}} \leftarrow$ None;
Compute $\mathbf{A}$ based on the semantic hierarchy;
for $t \leftarrow 1$ **to** T **do**
  $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x} + \delta$; $\mathbf{z} \leftarrow f(\mathbf{x}_{\text{adv}})$; $\mathbf{p} \leftarrow \text{softmax}(\mathbf{z})$;
  Flatten labels and probabilities; construct $\mathbf{A}_{valid}$ *and compute* $\mathbf{w}$
  Select $\mathbf{y}_{target}$ *via* $y_{target,i} = \arg\max_k w_{ik}$
  Compute $L_{\text{CE}}, L_{\text{target}}, L_{\text{reg}}, L_{\text{semantic}}$;
  $\mathcal{L} \leftarrow L_{\text{CE}} + \lambda_{\text{target}} L_{\text{target}} + \lambda_{\text{reg}} L_{\text{reg}} + \lambda_{\text{semantic}} L_{\text{semantic}}$;
  if $\mathcal{L} > L_{best}$ **then**
    $L_{\text{best}} \leftarrow \mathcal{L}$, $\delta_{\text{best}} \leftarrow \delta$
  end
  $\mathbf{g}_{\text{current}} \leftarrow \nabla_\delta \mathcal{L}$;
  $\mathbf{g} \leftarrow \gamma \mathbf{g} + \frac{\mathbf{g}_{\text{current}} + \lambda_{\text{prev}} \mathbf{g}_{\text{prev}}}{\|\mathbf{g}_{\text{current}} + \lambda_{\text{prev}} \mathbf{g}_{\text{prev}}\|_1 + \epsilon}$;
  Compute cosine similarity $\cos(\theta)$ between $\mathbf{g}_{\text{current}}$ and $\mathbf{g}_{\text{prev}}$;
  Update $\beta \leftarrow \min(\beta(1 + 0.5\cos(\theta)), 1.5)$;
  $\alpha_t \leftarrow \alpha(1 - \frac{t}{T})^{0.5}$;
  $\delta \leftarrow \delta + \alpha_t \beta \,\text{sign}(\mathbf{g})$; $\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$;
  $\delta \leftarrow \text{clip}(\mathbf{x} + \delta, \mathbf{x}_{\min}, \mathbf{x}_{\max}) - \mathbf{x}$;
  $\mathbf{g}_{\text{prev}} \leftarrow \mathbf{g}_{\text{current}}$; $\delta.\text{grad} \leftarrow \mathbf{0}$;
end
return $\mathbf{x}_{\text{adv}} = \mathbf{x} + \delta_{\text{best}}$;

B. Semantic Hierarchy and Affinity Matrix

We group classes into hierarchical structures (see Table I), forming an affinity matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ such that:

$$A_{ij} = \begin{cases} 1 & \text{if } i, j \text{ are in the same subgroup,} \\ 0.5 & \text{if } i, j \text{ are in different subgroups,} \\ & \text{but in the same group,} \\ 0 & \text{otherwise.} \end{cases}$$

We constructed the hierarchy based on the "things" and "stuff" classification [21], grouping semantically related labels to form subgroups. For example, we placed sky with vegetation in the Nature subgroup, and building with fence in the Construction subgroup. Traffic signs and poles were assigned to the same Street Furniture subgroup given their spatial association. This hierarchical structure helps leverage semantic relationships when generating adversarial attacks.

C. Semantic-Aware Target Selection

For each pixel, we apply class-specific weights based on semantic hierarchy to guide target misclassification loss. Let $\mathbf{y}_{\text{flat}}$ be the flattened ground truth labels of size $N = H \times W$, and $\mathbf{m}$ be a mask indicating valid pixels. We consider:

$$\mathbf{p}_{\text{flat}} \in \mathbb{R}^{N \times K}, \quad \mathbf{A}_{\text{valid}} \in \mathbb{R}^{N \times K},$$

where $\mathbf{p}_{\text{flat}}$ is obtained by reshaping $\mathbf{p}$ and selecting only valid pixels, and $\mathbf{A}_{\text{valid}}$ is derived by indexing $\mathbf{A}$ with the true classes at those pixels:

$$\mathbf{A}_{\text{valid}} = \{A_{\mathbf{y}_{\text{flat}}[i],j} \mid i \in \mathbf{m}, j \in [0, K-1]\}.$$

To steer predictions, we define weighted probabilities:

$$\mathbf{w} = \mathbf{p}_{\text{flat}} \odot (1 - \mathbf{A}_{\text{valid}}),$$

where $\odot$ denotes element-wise multiplication. The target class for pixel i is:

$$y_{\text{target},i} = \arg\max_k w_{ik}.$$

D. Loss Functions and Optimization

The loss function combines multiple objectives to maximize attack success:

$$\mathcal{L} = L_{\text{CE}} + \lambda_{\text{target}} L_{\text{target}} + \lambda_{\text{reg}} L_{\text{reg}} + \lambda_{\text{semantic}} L_{\text{semantic}}.$$

SHAA maximizes a combined objective loss to achieve effective misclassification. The loss terms are defined as follows:

1) **Cross-Entropy Loss**: Encourages deviation from correct class, where $\mathbf{z}_i$ is the logit vector for pixel i.

$$L_{\text{CE}} = \frac{1}{N} \sum_{i \in \mathbf{m}} \ell_{\text{CE}}(\mathbf{z}_i, y_i).$$

2) **Targeted Misclassification Loss**: Encourages the model to misclassify each pixel i into the chosen class $y_{\text{target},i}$:

$$L_{\text{target}} = -\frac{1}{N} \sum_{i \in \mathbf{m}} \ell_{\text{CE}}(\mathbf{z}_i, y_{\text{target},i}).$$

3) **L2 Regularization**: Penalizes large perturbations to maintain imperceptibility:

$$L_{\text{reg}} = -\|\delta\|_2^2.$$

4) **Semantic Consistency Loss**: For each group $g \in \mathcal{G}$ and each subgroup $s \in \mathcal{S}_g$, let $\mathbf{S}_s$ be a binary mask indicating pixels belonging to that subgroup in the ground truth.

$$L_{\text{semantic}} = \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} \text{MSE}\left(\sum_{k \in s} \mathbf{p}_k, \mathbf{S}_s\right),$$

where $\mathbf{p}_k$ is the predicted probability map for class k.

E. Gradient Update and Stability Mechanisms

1) **Momentum Accumulation**: We use L1 norm in the gradient accumulation for stable and balanced updates:

$$\mathbf{g} \leftarrow \gamma \mathbf{g} + \frac{\mathbf{g}_{\text{current}} + \lambda_{\text{prev}} \mathbf{g}_{\text{prev}}}{\|\mathbf{g}_{\text{current}} + \lambda_{\text{prev}} \mathbf{g}_{\text{prev}}\|_1 + \epsilon}$$

This norm choice was informed by the MI-FGSM [15]

2) **Adaptive Scaling**: We adjust the step size based on gradient alignment between consecutive iterations:

$$\beta \leftarrow \min(\beta(1 + 0.5\cos(\theta)), 1.5)$$

where

$$\cos(\theta) = \frac{\langle \mathbf{g}_{\text{current}}, \mathbf{g}_{\text{prev}} \rangle}{\|\mathbf{g}_{\text{current}}\|_2 \, \|\mathbf{g}_{\text{prev}}\|_2}$$

with the inner product $\langle \cdot, \cdot \rangle$ computed over the flattened gradient tensors.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

3) **Decay Schedule**: We gradually decrease the step size using a square root decay:

$$\alpha_t = \alpha \left(1 - \frac{t}{T}\right)^{0.5}$$

4) **Best Loss Tracking**: We keep track of the perturbation with the highest loss to yield the most effective AE.

IV. Experimental Settings

A. Dataset

We used the **Cityscapes** [22], an urban driving dataset with pixel-level annotations, using its 500 validation sets. For overall performance evaluation, we also used the **NightCity** [23], a night driving dataset, using 1399 validation images, to include nighttime scenarios. Both datasets share the 19-class annotation format plus an ignored class. For Cityscapes only, the images and labels were resized to 1024×512 beforehand with bilinear and nearest-neighbor interpolation, respectively, to match the models' training and deployment setup.

B. Network Models

We evaluated the attack on three segmentation models. Each model has different architecture and use different backbone. All evaluations were based on Cityscape checkpoints and subsequently fine-tuned on the NightCity dataset:

- **DeepLabV3+** [24]: Encoder-decoder with atrous separable convolution and a ResNet-50 backbone (`deeplabv3+-r50-d8`).
- **OCRNet** [25]: Object-Contextual Representations using an HRNet-48 backbone (`ocrnet-hr48`).
- **SegFormer** [26]: SegFormer with a MiT-B5 backbone (`mit-b5-8x1`).

C. Attack Methods and Parameters

We compare SHAA against PGD [13] and state-of-the-art white-box attacks including MI-FGSM [15], TI-FGSM [16], DIM [17], and FIA [18]. All attacks were run in 20 steps.

Common Attack Parameters:

- **Maximum Perturbation** (ϵ): 8.0in pixel values.
- **Step Size** (α): 2.0in pixel values.

Parameter-Specific Overview:

- **Momentum Decay Factor** ($\gamma = 0.9$): Employed by MI-FGSM, DIM, TI-FGSM, and SHAA.
- **Diversity Probability** ($p = 0.7$): DIM.
- **Kernel Size (15) and Gaussian Kernel Sigma** ($\sigma = 3.0$): TI-FGSM.
- PGD and FIA have no other extra parameters.

D. Evaluation Metrics

We evaluate attacks using pixel accuracy (PA), mean pixel accuracy (mPA), mean intersection over union (mIoU), mean Dice, peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM), following the standard definitions in Elmezain et al. [27]. We also evaluate attack success rate (ASR) using the following definition: Let N be the total pixel numbers, K the number of classes, y_i the ground truth label of pixel i, and $\hat{y}_i$ the predicted label. Denote $\hat{y}_i^{\text{clean}}$ and $\hat{y}_i^{\text{adv}}$ as clean and adversarial predictions:

$$\text{ASR} = \frac{\sum_{i=1}^{N} \delta\left(y_i = \hat{y}_i^{\text{clean}}\right) \delta\left(\hat{y}_i^{\text{adv}} \neq y_i\right)}{\sum_{i=1}^{N} \delta\left(y_i = \hat{y}_i^{\text{clean}}\right)}.$$

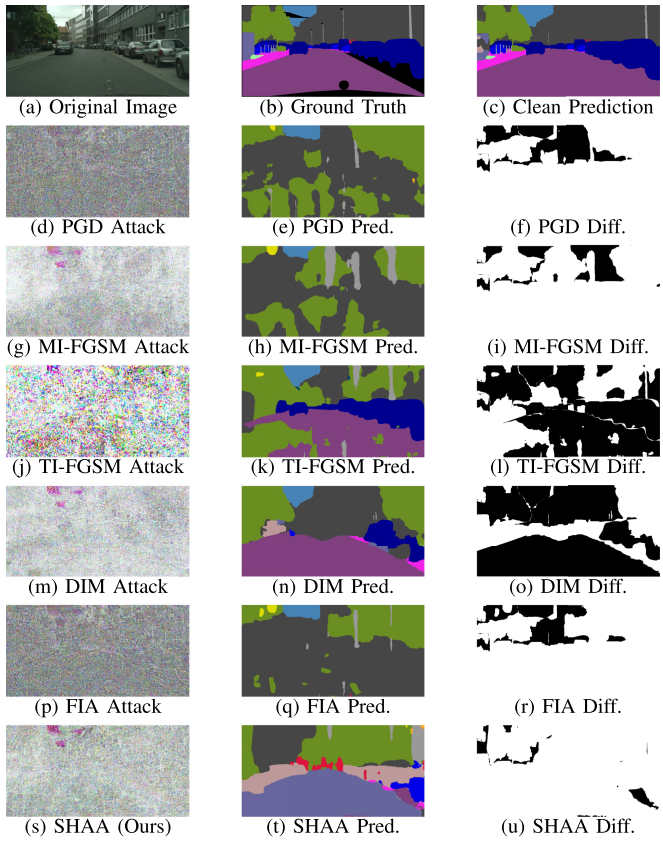


Fig. 2. Comparison of attack method results. Each row shows adversarial attack (left), attacked segmentation (middle), and difference map (right) where **white regions mean attack success** (altered from clean predictions).

E. Experimental Environments

The experiments were conducted with PyTorch 2.0.0 and TorchVision 0.15.0 on a NVIDIA RTX 3090.

V. Experimental Results

The experiments were five-folded; 1) attack performance of SHAA compared to other methods, 2) performance with varying perturbation budget, 3) attack transferability to target models, 4) performance under defense methods, and 5) ablation study on the effect of SHAA's algorithm components.

A. Overall Performance

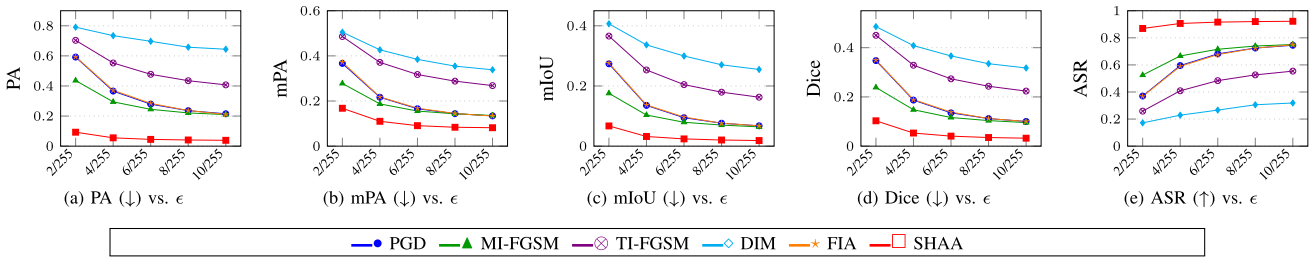
As shown in Fig. 2, SHAA generates adversarial examples that cause attacked segmentation (Fig. 2(u)) that is semantically nontrivial (Fig. 2(t)). Unlike other methods that produce arbitrary, scattered pixel-wise changes, SHAA guides the attack considering the semantic relationship. While all attacks alter model predictions, SHAA leads to more structured misclassification (Fig. 2(t)). This semantic attack can disrupts autonomous vehicle

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II
PERFORMANCE METRICS OF VARIOUS MODELS UNDER DIFFERENT ATTACK METHODS

Dataset	Model	Attack	PA(↓)	mPA(↓)	mIoU(↓)	Dice(↓)	ASR(↑)	PSNR(↑)	SSIM(↑)	Latency(s)
Cityscape	DeepLabV3+	Clean	0.9533	0.7318	0.6586	0.7362	-	-	-	-
		PGD	0.0530	**0.0430**	0.0068	0.0112	0.9027	**33.6900**	**0.8273**	**2.9707**
		MI-FGSM	0.0640	0.0468	0.0076	0.0124	0.8920	31.3515	0.7456	2.9739
		TI-FGSM	0.0615	0.0490	0.0077	0.0123	0.8941	31.3076	0.7994	2.9846
		DIM	0.2064	0.1016	0.0425	0.0619	0.7530	31.3805	0.7497	2.9745
		FIA	0.0565	0.0442	0.0072	0.0118	0.8992	**33.6895**	**0.8273**	3.1204
		SHAA (Ours)	**0.0323**	0.0536	**0.0043**	**0.0079**	**0.9233**	33.1606	0.8043	3.1323
	OCRNet	Clean	0.9530	0.7271	0.6548	0.7308	-	-	-	-
		PGD	0.0710	0.0579	0.0162	0.0265	0.8850	33.5482	0.8269	**3.1646**
		MI-FGSM	0.0680	0.0564	0.0150	0.0245	0.8880	31.2506	0.7456	3.1659
		TI-FGSM	0.0877	0.0665	0.0209	0.0328	0.8680	31.2446	0.8070	3.1746
		DIM	0.1818	0.1266	0.0640	0.0864	0.7762	31.3722	0.7525	3.1683
		FIA	0.0691	0.0581	0.0163	0.0267	0.8868	33.5497	0.8269	3.3232
		SHAA (Ours)	**0.0394**	**0.0494**	**0.0090**	**0.0153**	**0.9176**	**33.9058**	**0.8306**	3.3391
	SegFormer	Clean	0.9572	0.7605	0.6835	0.7593	-	-	-	-
		PGD	0.2370	0.1454	0.0769	0.1128	0.7233	**33.5667**	**0.8259**	**3.2088**
		MI-FGSM	0.2194	0.1427	0.0694	0.1029	0.7408	31.3243	0.7469	3.2115
		TI-FGSM	0.4338	0.2888	0.1794	0.2435	0.5275	31.3580	0.8146	3.2191
		DIM	0.6624	0.3555	0.2720	0.3368	0.3006	31.3755	0.7508	3.2097
		FIA	0.2395	0.1467	0.0783	0.1146	0.7207	33.5662	**0.8259**	3.3692
		SHAA (Ours)	**0.0411**	**0.0846**	**0.0204**	**0.0349**	**0.9198**	31.9267	0.7687	3.3882
NightCity	DeepLabV3+	Clean	0.8303	0.4875	0.4114	0.4856	-	-	-	-
		PGD	0.0951	0.0709	0.0202	0.0308	0.7434	**33.7455**	0.7985	**2.9578**
		MI-FGSM	0.0945	0.0722	0.0196	0.0301	0.7437	31.4802	0.7138	2.9752
		TI-FGSM	0.1077	0.0807	0.0247	0.0365	0.7308	31.4624	0.7615	2.9844
		DIM	0.0687	**0.0619**	0.0177	0.0269	0.7710	31.5267	0.7171	2.9758
		FIA	0.0961	0.0720	0.0206	0.0314	0.7422	33.7452	**0.7986**	3.1363
		SHAA (Ours)	**0.0616**	0.0627	**0.0105**	**0.0176**	**0.7781**	31.9513	0.7313	3.1208
	OCRNet	Clean	0.8023	0.4557	0.3772	0.4497	-	-	-	-
		PGD	0.0835	0.0734	0.0137	0.0230	0.7297	33.8440	**0.8002**	**3.5977**
		MI-FGSM	0.0829	0.0736	0.0129	0.0216	0.7301	31.3657	0.7077	3.6031
		TI-FGSM	0.0805	0.0743	0.0133	0.0222	0.7322	31.4032	0.7584	3.6060
		DIM	0.1228	0.0754	0.0193	0.0294	0.6959	31.4989	0.7152	3.6017
		FIA	0.0828	0.0732	0.0138	0.0231	0.7298	**33.8448**	**0.8002**	3.7777
		SHAA (Ours)	**0.0610**	**0.0671**	**0.0119**	**0.0192**	**0.7528**	32.1633	0.7424	3.7577
	SegFormer	Clean	0.8123	0.5129	0.4150	0.4937	-	-	-	-
		PGD	0.0586	0.0634	0.0174	0.0275	0.7568	**33.6493**	**0.7963**	**3.2215**
		MI-FGSM	0.0571	0.0643	0.0165	0.0260	0.7580	31.3909	0.7111	3.2235
		TI-FGSM	0.0937	0.1005	0.0387	0.0575	0.7228	31.3790	0.7677	3.2312
		DIM	0.1821	0.1277	0.0628	0.0892	0.6367	31.4720	0.7145	3.2235
		FIA	0.0577	0.0637	0.0173	0.0274	0.7576	33.6489	**0.7963**	3.4022
		SHAA (Ours)	**0.0111**	**0.0435**	**0.0046**	**0.0080**	**0.8050**	32.1933	0.7449	3.3952

(**Bold** indicates best among attacks per column)



(a) PA (↓) vs. ϵ (b) mPA (↓) vs. ϵ (c) mIoU (↓) vs. ϵ (d) Dice (↓) vs. ϵ (e) ASR (↑) vs. ϵ

Fig. 3. Comparison of attacks at varying perturbation budgets. While all attacks show higher ASR at larger ϵ, SHAA had robust ASR with smaller ϵ.

functionality (e.g., misclassifying roads as walls) while considering a semantic scene structure. Difference maps (Fig. 2(u) and Fig. 2(f), (i), (l), (o), (r)) illustrate SHAA's strong attack coverage.

Quantitatively, SHAA consistently showed stronger attack success in all metrics on both datasets Table II. While it was not the best in PSNR/SSIM, the differences were modest. Although SHAA required higher latency, the overhead was modest relative to its improved attack performance.

On **Cityscape** dataset, SHAA showed particularly strong performance against SegFormer. While other attacks exhibited degraded performance on SegFormer, SHAA maintained its effectiveness, achieving the lowest mIoU (0.0204) compared to the next best attack MI-FGSM (0.0694, 3.40× higher). This suggests SHAA's semantic-guided approach performs especially well against transformer architectures.

On **NightCity**, the clean mIoU was lower due to challenging lighting conditions. Even so, SHAA consistently outperformed the other attacks, indicating that its semantic-guided approach maintains robust performance under adverse scenarios. Meanwhile, PSNR and SSIM results suggest that SHAA's stronger perturbations are achieved while not compromising image quality, even on more demanding datasets.

B. *Performance With Perturbation Budget*

Fig. 3 presents the performance of attack methods under varying perturbation budgets (ϵ from 2 to 10) against the SegFormer model on the Cityscape dataset. While all attack metrics improved with larger perturbation budgets, SHAA maintained consistently lower segmentation metrics across all perturbation budgets compared to other methods.

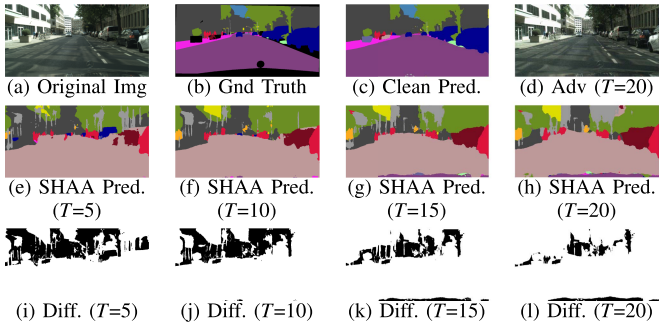


Fig. 4. Comparison of SHAA outputs across varying iteration counts T.

In particular, at $\epsilon = 2$, SHAA achieves an ASR of 0.8691, while the next best method (MI-FGSM) reaches 0.524 (approximately 66% higher) and reduces the mIoU to 0.067 compared to MI-FGSM's 0.1756 (2.61× higher). This demonstrates that SHAA can achieve effective attacks with small perturbations, requiring less distortion than other methods and making it stealthier in budget aspects.

C. Effect of Step Numbers on Attack Dynamics

Fig. 4 illustrates how the segmentation output evolves with increasing attack iterations. As the number of steps grows, SHAA progressively produces more potent adversarial perturbations with larger alternation from the clean prediction.

D. Transferability of Attack Methods

To evaluate the transferability of SHAA and other attack methods, we generated adversarial examples using SegFormer and tested them on DeepLabV3+, OCRNet, and Mask r-cnn [28] (`r50-fpn-cityscapes`). We included Mask r-cnn to assess transferability beyond semantic segmentation models, as it performs instance segmentation with a different architecture and approach to pixel classification.

SHAA showed performance degradation on the transferred models in all metrics (Fig. 5). For DeepLabV3+, SHAA showed the lowest mIoU (0.327, compared to the clean mIoU of 0.659). SHAA transferred to OCRNet remained effective (e.g., mIoU of 0.333 vs. clean 0.655). For Mask r-cnn, we adapted the evaluation by converting instance masks to semantic labels and measured mIoU on instance-level categories (i.e., "things"). SHAA reduced the adjusted mIoU to 0.151 (vs. clean 0.674), suggesting its performance against different segmentation paradigms. The results indicate that SHAA can exploit vulnerabilities across multiple architectures beyond the source model where it is generated.

E. Performance Under Defense Methods

1) Input-Level Defenses: We compared SHAA with other attacks under input-level defenses using the SegFormer model. We applied BitDepth reduction (3 bits) [29], JPEG compression (quality=50) [30], Median filter of kernel 3 [29], and pixel deflection with wavelet denoising (PD+WD) [31]. We included these four input-level defenses, BitDepth (quantization), JPEG (lossy compression), Median filter (spatial smoothing), and Pixel Deflection (randomized transformation), as they differ in how they affect the input and gradients.

TABLE III
ATTACK PERFORMANCE ON SEGFORMER UNDER DIFFERENT DEFENSES

Defense	Attack	PA(↓)	mIoU(↓)	ASR(↑)	PSNR(↑)	SSIM(↑)
BitDepth	Clean	0.9174	0.5739	-	27.8831	0.8059
	PGD	0.7156	0.3111	0.2467	**26.8712**	0.6814
	MI-FGSM	0.6229	0.2487	0.3386	26.2995	0.6485
	TI-FGSM	0.7689	0.3565	0.1944	26.3078	**0.6825**
	FIA	0.7145	0.3113	0.2478	26.8709	0.6814
	DIM	0.7733	0.3565	0.1904	26.3134	0.6495
	SHAA	**0.5191**	**0.2198**	0.4442	26.4584	0.6567
JPEG	Clean	0.9379	0.6025	-	37.1225	0.9591
	PGD	0.8594	0.4461	0.1034	**34.7514**	**0.9074**
	MI-FGSM	0.8195	0.3954	0.1430	33.1936	0.8559
	TI-FGSM	**0.5751**	**0.2453**	**0.3869**	30.9181	0.8160
	FIA	0.8603	0.4491	0.1026	34.7504	0.9074
	DIM	0.8252	0.4040	0.1392	32.7837	0.8395
	SHAA	0.8116	0.4108	0.1532	33.4850	0.8643
Median	Clean	0.9544	0.6664	-	38.8898	0.9820
	PGD	0.4987	0.2148	0.4626	34.7491	0.9018
	MI-FGSM	0.4723	0.1906	0.4889	32.3011	0.8099
	TI-FGSM	0.4395	0.1780	0.5219	30.8504	0.8106
	FIA	0.4993	0.2131	0.4619	34.7494	0.9018
	DIM	0.7408	0.3299	0.2230	32.1335	0.8054
	SHAA	**0.2611**	**0.1399**	**0.7016**	**32.9419**	**0.8373**
PD+WD	Clean	0.9345	0.5818	-	34.4664	0.9398
	PGD	0.8851	0.4767	0.0790	33.5549	0.9119
	MI-FGSM	0.8688	0.4521	0.0950	33.0017	0.8897
	TI-FGSM	**0.5088**	**0.2105**	**0.4534**	29.9857	0.7894
	FIA	0.8849	0.4762	0.0792	33.5551	0.9119
	DIM	0.8687	0.4532	0.0967	32.6678	0.8758
	SHAA	0.8634	0.4648	0.1025	**33.0849**	**0.8948**

(**Bold** indicates best per column among attacks).

As shown in Table III, all attacks show reduced attack success in the defenses of the input level, but the severity of degradation varies by defense and attack methods. SHAA maintains performance under BitDepth and Median filtering, likely due to its semantic hierarchy and momentum guiding more robust perturbations that survive coarse quantization and smoothing. In contrast, TI-FGSM stands out against Gaussian blur and JPEG compression. Its translation-invariant approach may generate perturbations that persist through blurring and compression artifacts. Overall, while defenses weaken adversarial attacks, certain attack strategies adapt better to specific defense-induced distortions.

2) Model-Level Defenses: To test attacks against model-level defenses, we performed adversarial training (AT) with the attack methods compared. Following Losch et al. [32], we used a mixed training batch (half clean, half adversarial). We fine-tuned SegFormer for 10 epochs with each attack serving as the perturbation generator, then evaluated each model against all attacks to measure cross-attack robustness. To provide a baseline robustness evaluation, AT examples were generated with 10 steps, with evaluation attacks kept at 20 steps as in previous experiments.

Fig. 6 shows the comparative performance of the defenses. TI-FGSM-based AT demonstrates strong resistance against its matching attack. DIM-based AT was an effective defense in general, maintaining reasonable performance across most attacks. SHAA attack consistently decreased the mIoU against all defenses, especially against TI-FGSM-based AT (0.014 mIoU). While baseline AT provides robustness against attacks to some extent, attacks still degrade segmentation performance, requiring nuanced defense strategies.

F. Ablation Study on Algorithm Components

We tested four ablation conditions to evaluate the influence of semantic terms and adaptive momentum within SHAA. Table IV summarizes the ablation results on SegFormer. SHAA fully retains the semantic terms (targeted misclassification and semantic consistency) and adaptive step factors, yielding the lowest PA,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

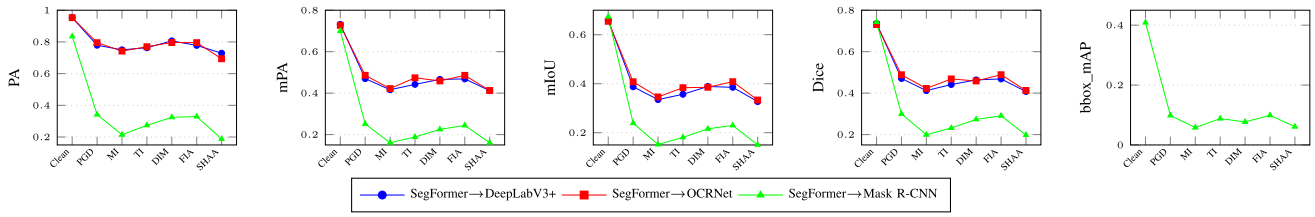


Fig. 5. Performance comparison of attack methods from Segformer against transfer models. bbox_mAP indicates the bounding box mean average precision.

TABLE IV
PERFORMANCE METRICS OF SEGFORMER UNDER DIFFERENT ABLATION CONDITIONS

Model	Attack Method	PA(↓)	mPA(↓)	mIoU(↓)	Dice(↓)	ASR(↑)	PSNR(↑)	SSIM(↑)	Latency(s)
SegFormer	SHAA Full	**0.0409**	**0.0852**	**0.0204**	**0.0351**	**0.9201**	31.9286	0.7688	3.4291
	Ablation Semantic	0.3164	0.1744	0.0992	0.1413	0.6440	**32.6013**	**0.7944**	3.2908
	Ablation Adaptive Step	0.0449	0.0937	0.0241	0.0406	0.9163	31.6434	0.7578	3.4272
	Ablation Both (MI-FGSM)	0.2196	0.1430	0.0689	0.1025	0.7406	31.3231	0.7469	**3.2674**

(**Bold** indicates the best result per column)

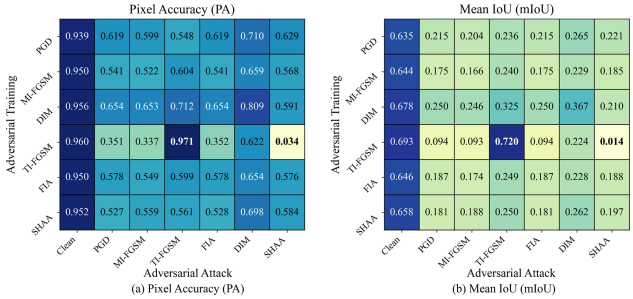


Fig. 6. Performance of adversarial attacks against adversarial training.

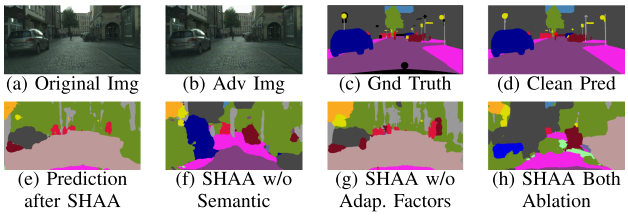


Fig. 7. Visual comparisons of ablation study for algorithm components.

mPA, mIoU, and Dice with the highest ASR. Ablation Semantic disables semantic terms, often leaving large unattacked areas (e.g., road in (f) of Fig. 7) but achieving slightly higher PSNR and SSIM. Ablation Adaptive Step removes the dynamic step factors of the momentum (dynamic α and β) yet keeps semantic grouping, leading to a marginal ASR drop. Ablation Both removes both semantic terms and adaptive step factors (which now becomes MI-FGSM), resulting in moderate effectiveness. Overall, combining the semantic terms with the adaptive step produces a semantically nontrivial attack result and achieves the strongest attack success.

VI. DISCUSSION

A. Effectiveness and Generalization Capabilities of SHAA

The proposed SHAA method shows that incorporating semantic structures and adaptive momentum-based optimization enhances attack performance in semantic segmentation. Unlike arbitrary pixel-level methods, it directs misclassifications with semantic hierarchy, producing nontrivial, yet effective adversarial examples. Our results demonstrate stronger attack success and preserved image quality, especially on the tested transformer model. However, its high latency compared to naive pixel-level methods should be noted.

SHAA demonstrates its potential as a white-box method, revealing segmentation model's vulnerabilities in worst-case scenarios. The SHAA's robust adversarial attack against input and model level defenses, particularly quantization or smoothing-based strategies, or TI-FGSM-based adversarial training, inform the need for more robust defense strategies.

SHAA also demonstrates transferability, deceiving non-source models as targets. Perturbations generated on a source model reduced mIoU (roughly by half) across various target architectures, including cross-task transfer to an instance segmentation model (Mask R-CNN). While requiring white-box access to the source model for perturbation generation, this transferability suggests potential black-box applicability, although performance is lower compared to direct attacks.

B. Sensitivity to Semantic Hierarchy Design

The choice of $A_{ij} = 0.5$ for classes within the same group but different subgroups serves to modulate the attack's preference between targeting highly similar classes (e.g., within the same subgroup) and semantically distant classes (e.g., in different groups). While this letter employs a specific hierarchy with $A_{ij} = 0.5$ as a baseline, SHAA's explicit use of the semantic hierarchy means that how different hierarchy structures and affinity parameters influence attack performance and scene structure warrants further investigation.

While the attack was designed for semantic segmentation in autonomous driving contexts, its reliance on a predefined semantic hierarchy may pose challenges for scenarios with larger label sets. Defining a meaningful hierarchy becomes increasingly complex as the number of classes grows, potentially limiting direct application. In scenarios where constructing a domain-specific hierarchy is difficult, leveraging transferability by generating attacks on a related source model could be considered as an alternative approach.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

C. Challenges and Potential of SHAA in Real-World Scenarios

As SHAA involves optimization processes, translating it from digital to physical world presents challenges. Like other iterative attacks, the iterative nature requires direct model access for gradient computation, crafting barriers for real-world deployment, attacking unknown models in real-time.

Nevertheless, SHAA's transferability suggests possibilities for physical implementation. Digitally optimized perturbations from a source model could be translated into physical objects such as adversarial billboards or textures [2], [5]. The effectiveness of such physical manifestations would be influenced by environmental factors, including lighting, viewpoints, distances, and deformations of the perturbation-carrying objects [5]. Incorporating techniques like expectation over transformation would help account for these variations, leading to robust adversarial effects under mild changes [2]. Future work could explore these physical implementations to assess the real-world impact of SHAA.

VII. Conclusion

This work introduces SHAA, an adversarial attack that uses scene-level semantics to disrupt semantic segmentation models. By leveraging semantic hierarchy, SHAA surpasses existing pixel-centric perturbations in attack success. Its transferability across different architectures suggests its potential as a white-box method beyond source models. Although defenses remain partially effective, SHAA demonstrates moderate to high attack success against the tested input-level and model-level defenses. Future research could investigate shared and distinct vulnerabilities across various models while exploring alternative groupings and parameters for semantic hierarchy, as well as the attack's feasibility in physical settings. Understanding these aspects can advance adversarial strategies and defensive mechanisms by revealing potential reliability gaps in autonomous vehicle perception systems.

References

[1] H. -J Yoon, H. Jafarnejadsani, and P. Voulgaris, "Learning when to use adaptive adversarial image perturbations against autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 8, no. 7, pp. 4179–4186, Jul. 2023.

[2] G. Rossolini, F. Nesti, G. D'Amico, S. Nair, A. Biondi, and G. Buttazzo, "On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18328–18342, Dec. 2024.

[3] A. D. M. Ibrahum, M. Hussain, and J. -E. Hong, "Deep learning adversarial attacks and defenses in autonomous vehicles: A systematic literature review from a safety perspective," *Artif. Intell. Rev.*, vol. 58, no. 1, pp. 1–53, 2025.

[4] K. Yamanaka, R. Matsumoto, K. Takahashi, and T. Fujii, "Adversarial patch attacks on monocular depth estimation networks," *IEEE Access*, vol. 8, pp. 179094–179104, 2020.

[5] N. Patel, P. Krishnamurthy, S. Garg, and F. Khorrami, "Overriding autonomous driving systems using adaptive adversarial billboards," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11386–11396, Aug. 2022.

[6] L. Chietal, "Adversarial attacks on autonomous driving systems in the physical world: A survey," *IEEE Trans. Intell. Veh.*, early access, Oct. 21, 2024, doi: 10.1109/TIV.2024.3484152.

[7] L. Wang, W. Cho, and K. -J. Yoon, "Deceiving image-to-image translation networks for autonomous driving with adversarial perturbations," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1421–1428, Apr. 2020.

[8] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 819–828.

[9] A. Bar,etal, "The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: Enhancing extensive environment sensing," *IEEE Signal Process. Mag.*, vol. 38, no. 1, pp. 42–52, Jan. 2021.

[10] B. Badjie, J. Cecílio, and A. Casimiro, "Adversarial attacks and countermeasures on image classification-based deep learning models in autonomous driving systems: A systematic review," *ACM Comput. Surveys*, vol. 57, no. 1, pp. 1–52, 2024.

[11] Y. Lu, H. Ren, W. Chai, S. Velipasalar, and Y. Li, "Time-aware and task-transferable adversarial attack for perception of autonomous vehicles," *Pattern Recognit. Lett.*, vol. 178, pp. 145–152, 2024.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[14] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 888–897.

[15] Y. Dong,etal, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[16] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4312–4321.

[17] C. Xie,etal, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2730–2739.

[18] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, "Feature importance-aware transferable adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7639–7648.

[19] J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2755–2764.

[20] Z. Chen, C. Wang, and D. Crandall, "Semantically stealthy adversarial attacks against segmentation models," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2022, pp. 4080–4089.

[21] X. Liang, H. Zhou, and E. Xing, "Dynamic-structured semantic propagation network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 752–761.

[22] M. Cordts,etal, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

[23] X. Tan, K. Xu, Y. Cao, Y. Zhang, L. Ma, and R. W. H. Lau, "Night-time scene parsing with a large real dataset," *IEEE Trans. Image Process.*, vol. 30, pp. 9085–9098, 2021.

[24] L. -C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[25] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, 2020, pp. 173–190.

[26] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12077–12090.

[27] M. Elmezain, L. S. Saoud, A. Sultan, M. Heshmat, L. Seneviratne, and I. Hussain, "Advancing underwater vision: A survey of deep learning models for underwater object recognition and tracking," *IEEE Access*, vol. 13, pp. 17830–17867, 2025.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[29] Y. Cheng, B. Zhou, Y. Chen, Y. -C. Chen, X. Ji, and W. Xu, "Evaluating compressive sensing on the security of computer vision systems," *ACM Trans. Sensor Netw.*, vol. 20, no. 3, pp. 1–24, 2024.

[30] N. Das,etal, "SHIELD: Fast, practical defense and vaccination for deep learning using JPEG compression," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 196–204.

[31] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8571–8580.

[32] M. Losch, M. Omran, D. Stutz, M. Fritz, and B. Schiele, "On adversarial training without perturbing all examples," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 1–21.