ISSN 2327-4018 Volume 14, Number 1, January 2026

Scientific Research Publishing

Open Journal of Modelling and Simulation

ISSN: 2327-4018

9 772327 401006 01

https://www.scirp.org/journal/ojmsi

Journal Editorial Board

ISSN: 2327-4018 (Print), 2327-4026 (Online)
https://www.scirp.org/journal/ojmsi

Open Journal of Modelling and Simulation, 2026, 14(1), 1-43
https://www.scirp.org/journal/ojmsi
ISSN Online: 2327-4026
ISSN Print: 2327-4018

Table of Contents

Volume 14 Number 1 January 2026

Open Journal of Modelling and Simulation (OJMSi)
Journal Information

SUBSCRIPTIONS

The *Open Journal of Modelling and Simulation* (Online at Scientific Research Publishing, https://www.scirp.org/) is published quarterly by Scientific Research Publishing, Inc., USA.

Subscription rates:
Print: $39 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: ojmsi@scirp.org

Open Journal of Modelling and Simulation, 2026, 14(1), 1-20
https://www.scirp.org/journal/ojmsi
ISSN Online: 2327-4026
ISSN Print: 2327-4018

Domain-Robust Marine Plastic Detection Using Vision Models

Saanvi Kataria

John P Stevens High School, Edison, USA
Email: saanvikat@gmail.com

How to cite this paper: Kataria, S. (2026) Domain-Robust Marine Plastic Detection Using Vision Models. *Open Journal of Modelling and Simulation*, **14**, 1-20.
https://doi.org/10.4236/ojmsi.2026.141001

Received: September 26, 2025
Accepted: October 24, 2025
Published: October 27, 2025

Abstract

Marine plastic pollution is a pressing environmental threat, making reliable automation for underwater debris detection essential. However, vision systems trained on one dataset often degrade on new imagery due to domain shift. This study benchmarks models for cross-domain robustness, training convolutional neural networks—CNNs (MobileNetV2, ResNet-18, EfficientNet-B0) and vision transformers (DeiT-Tiny, ViT-B/16) on a labeled underwater dataset and then evaluates them on a balanced cross-domain test set built from plastic-positive images drawn from a different source and negatives from the training domain. Two zero-shot models were assessed, CLIP ViT-L/14 and Google's Gemini 2.0 Flash, that leverage pretraining to classify images without fine-tuning. Results show the lightweight MobileNetV2 delivers the strongest cross-domain performance (F1 ≈ 0.97), surpassing larger models. All fine-tuned models achieved high Precision (~99%), but differ in Recall, indicating varying sensitivity to plastic instances. Zero-shot CLIP is comparatively sensitive (Recall ~80%) yet prone to false positives (Precision ~56%), whereas Gemini exhibits the inverse profile (Precision ~99%, Recall ~81%). Error analysis highlights recurring confusions with coral textures, suspended particulates, and specular glare. Overall, compact CNNs with supervised training can generalize effectively for cross-domain underwater detection, while large pretrained vision-language models provide complementary strengths. Future work should explore hybrid strategies, such as small CNN backbones with foundation-model priors and domain-aware sampling, to combine high Precision with Recall across heterogeneous marine environments and reduce labeling burdens at scale.

Keywords

Vision Models, CNN, Marine Plastic Detection

1. Introduction

Plastic debris in marine environments has become pervasive and harmful, endangering wildlife and ecosystems. More than 8 million tons of plastic enter the oceans annually, resulting in the deaths of over 100,000 marine mammals and turtles, as well as over a million seabirds, each year, with an estimated $13 billion in global economic costs [1]. Locating and removing submerged plastic waste is a challenging and labor-intensive task, motivating the development of autonomous underwater vehicles and computer vision (CV) systems to aid in debris monitoring and cleanup [2]. Deep learning has shown promise in detecting marine debris, significantly improving identification accuracy in recent studies [3]. For instance, Hipolito *et al.* achieved a mean average precision of over 98% in detecting underwater plastic using a YOLOv3 model, although on a limited dataset [3].

However, a persistent challenge is domain shift: models trained on a particular underwater imagery dataset often face degraded performance when applied to images from different locations, cameras, or conditions [4]. Underwater images can vary widely in visibility, lighting color cast, and background (reef vs. open water), making it difficult for a model to generalize beyond its training distribution [4]. Prior works have noted that many marine debris image datasets are highly localized to specific environments, which limits the generality of models [5]. For example, Sánchez-Ferrer *et al.* found that a detector trained in a controlled tank environment performed poorly on real seafloor images, highlighting the need for more diverse training data to encompass various conditions [4]. Van Lieshout *et al.* (2020) and others similarly argue for diversifying water conditions, visibility, and locations in training imagery to improve robustness [5]. Recent surveys confirm that many state-of-the-art models achieve high accuracy on specific datasets (often > 60% - 70% mAP), but their real-world efficacy depends on how well the training data represent the variability of marine settings [6].

Addressing cross-domain robustness is thus crucial for the practical deployment of marine litter detectors in the wild. Researchers have begun exploring domain adaptation and generalization techniques for underwater vision. Chen *et al.* introduced a continual unsupervised domain adaptation method for a custom object detector, aiming to improve detection across different domains [7] progressively. Saoud *et al.* (2023) proposed ADOD, an attention-augmented YOLOv3 that learns domain-invariant features through a residual attention module and an auxiliary domain classifier, resulting in improved detection in unseen underwater conditions [5]. These approaches show that explicit domain generalization strategies can enhance robustness.

On the other hand, the rise of large-scale pre-trained vision models offers an alternate path: models such as OpenAI's CLIP learn from 400 million image–text pairs. It can recognize visual concepts in a zero-shot manner via natural language prompts [8]. Such models have demonstrated surprisingly strong out-of-the-box performance on many tasks without fine-tuning, sometimes rivaling fully supervised models [8]. This study hypothesizes that zero-shot models may effectively detect marine plastics without any task-specific training, due to their broad visual–language knowledge. However, this may come at the cost of Precision or Recall.

This work conducts a comprehensive evaluation of domain-robust marine plastic detection using a range of vision models, comparing conventional supervised CNN/Transformer models against modern zero-shot models. This study includes training MobileNetV2, ResNet-18, EfficientNet-B0 (convolutional networks), and DeiT-Tiny, ViT-B/16 (transformers)—on an underwater plastic vs. no-plastic image dataset. All models are trained under identical settings and then tested on a balanced cross-domain set that simulates a domain shift: positive samples come from a different underwater image source than the training data, mixed with an equal number of negative samples from the training domain. Additionally, two cutting-edge zero-shot models were evaluated, CLIP ViT-L/14 and Google's Gemini 2.0 Flash [9], by using prompt-based inference to classify images without any fine-tuning on the training data. By analyzing their precision, recall, F1-score, and AUC metrics, along with failure cases (e.g., reflections or coral falsely identified as plastic), this study aims to answer: Which models best retain high accuracy under domain shift? Also, what insights do their differences provide for designing robust detectors?

This paper presents a unified benchmark of seven vision models on a cross-domain marine debris detection task, revealing performance trade-offs between model size, architecture, and training strategy. Amongst the results, it is clear that lightweight CNN (MobileNetV2) surprisingly achieved the highest cross-domain F1-score (~0.97), outperforming deeper networks on the balanced test, suggesting that model capacity alone does not guarantee better generalization for this task.

Other observations include that zero-shot models can attain competitive Recall or Precision without seeing any training images. However, each exhibits distinct behavior: CLIP was very sensitive in finding plastics but with many false alarms, whereas Gemini was extremely precise but missed more positives.

Through error analysis, standard failure modes were identified, such as glare (specular highlights underwater) and coral reef structures, which lead to false positives, highlighting the need for targeted improvements. Overall, this serves as a point of guidance on model selection for marine debris detection in unseen conditions and underscores the importance of both training-domain diversity and leveraging pre-trained knowledge for robust environmental monitoring.

2. Methodology

2.1. Datasets

There were two complementary underwater image datasets used to simulate a domain shift between training and testing.

The training dataset consists of an underwater imagery collection (sourced from Kaggle) containing photographs labeled for the presence of plastic waste. This dataset comprises a variety of underwater scenes, including both those with and without plastic, such as images of clear water columns, seabeds with coral or rocks, and occasional marine animals, each annotated as either "plastic" (positive) or "no_plastic" (negative). The total dataset comprised approximately 2150 images, with roughly equal representation of the two classes (about 1050 plastics vs. 1100 non-plastics). This study combined all the provided images from this source and then performed its own train/validation split (details below) [10].

The test dataset was taken from a different source of underwater plastic images (a different dataset also sourced from Kaggle). Notably, this second dataset contains only images positive for plastic debris—for example, divers photographing trash on the seafloor or floating plastic bags in the water. It represents a distinct domain, as the imaging conditions, camera equipment, and locations differ from those in the training set. Many images feature tropical reef settings or different water clarity, introducing a significant domain shift. Since this dataset lacked any explicit negative examples, this study could not use it directly to evaluate binary classification performance. Instead, this study constructed a balanced cross-domain test set by pairing this positive set with an equal number of negative samples from the training domain [11]. Specifically, all available unique positive images from the second source (501 images) were taken and randomly sampled 501 negative images from the training dataset (ensuring these negatives were not used in training or validation, to avoid any information leak). These were then combined and shuffled to form a 1002-image test set (50% plastic, 50% non-plastic).

This balanced design was chosen to prevent bias in Precision, recall, and AUC metrics due to class imbalance, and to stress-test the models' ability to identify plastic in unfamiliar conditions while avoiding over-prediction on familiar-background negatives. The negative samples from the training domain include scenes like coral reefs, rocks, fish, and open water with no artificial objects—some of which are visually complex and could potentially confuse a model that has only seen them in training as negatives. By mixing them with the foreign positive images, the test set evaluates both the sensitivity to novel plastic appearances and the specificity against known non-plastic imagery.

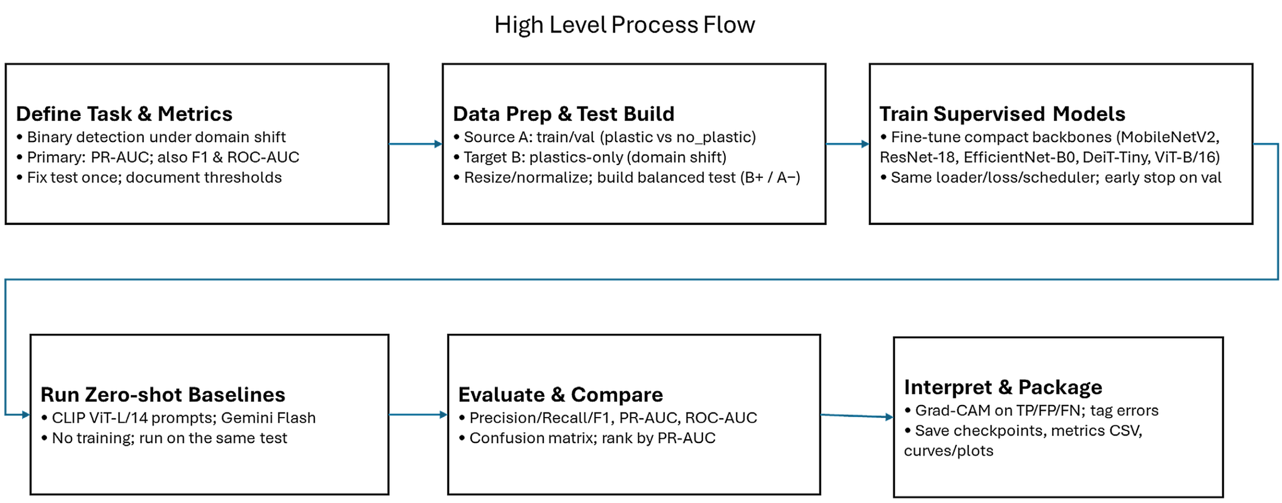


Figure 1. High-level overview.

Figure 1: The pipeline defines the binary detection task and evaluation metrics, prepares balanced source and domain-shifted target datasets, fine-tunes compact CNN and ViT models, runs zero-shot CLIP and Gemini baselines, evaluates performance using PR-AUC, ROC-AUC, and related metrics, and interprets results through Grad-CAM visualization, error tagging, and packaged outputs for reproducibility.

In summary, the training set ("source domain") comprises images from a single distribution of underwater environments. While the test set's plastic images ("target domain") come from a different distribution (*i.e.*, different locations and camera conditions) [10]. The domain discrepancy is evident in qualitative differences—for example, the training images were likely collected in temperate water with relatively muted colors and simpler backgrounds, while the test positives include vibrant tropical scenes with complex coral backgrounds. These differences pose a significant challenge to generalizing the models.

2.2. Preprocessing & Splits

All image files from both datasets were first organized and cleaned. Using the directory structure and file naming, an inference for class labels for Dataset A (training source) was created: images were labeled 1 (plastic) if found in folders named "plastic" or similar, and 0 (no_plastic) if in folders indicating no debris. This ensured the train/val split was stratified (preserving the ~50/50 class ratio). The final counts were approximately 1180 training images and 520 validation images from Dataset A (with roughly 630 plastic versus 550 non-plastic in the training set, and 210 versus 310 in the validation set).

For Dataset B (test source), since all images depict plastic, they were assigned label one, and a single data frame was created for test positives. The sampling procedure was executed to create a balanced test set, involving shuffling and random selection with a fixed random seed for reproducibility. A total of 501 negative images were sampled from the pool of Dataset A's negatives (combining those in training and validation sets to have a larger pool, after confirming none had plastic content). Those selected negatives were removed from any training usage after sampling. They were paired with all 501 positive images from Dataset B to form the test set. Thus, the cross-domain test set consists of 1002 images (501 plastic, 501 no-plastic), with positives all from the novel domain and negatives from the original domain.

All models (MobileNetV2, ResNet-18, EfficientNet-B0, DeiT-Tiny, ViT-B/16) were fine-tuned from ImageNet-pretrained weights with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1\text{e}{-8}$) at initial LR = 3×10^{-4} and weight decay = 1×10^{-4}.

Prior to model training and inference, standard preprocessing to the images was applied. Each image was resized (and center-cropped if necessary) to 224×224 pixels—the typical input size for ImageNet-pretrained models – to be fed into the CNNs or ViTs. The same normalization (ImageNet mean and standard deviation) was used for all models to ensure consistency. Extensive data augmentation was not performed during training, aside from basic flips or mild color jitter, since the focus was not on maximizing in-domain accuracy but on evaluating cross-domain performance; it is essential to not leak target domain characteristics via heavy augmentations inadvertently [12]. That said, mild augmentations were applied to make the model less overfit to exact training images—e.g., random horizontal flips (underwater scenes lack inherent left-right orientation) and slight brightness/contrast shifts to mimic different lighting conditions. Augmentations were applied only to training images. Validation and test images were not augmented; they underwent only deterministic resizing/cropping and normalization.

2.3. Models

Seven models were evaluated, covering both classical CNN architectures and modern transformer-based and multimodal approaches:

MobileNetV2: A lightweight convolutional neural network known for efficiency on embedded devices. MobileNetV2 utilizes depthwise separable convolutions and an inverted residual structure, allowing it to have significantly fewer parameters (~3.4 M) than standard CNNs while maintaining good accuracy. This was chosen as a baseline for practical deployment considerations (e.g., in underwater drones where computing power is limited). The version pretrained on ImageNet was used as the starting point for training.

ResNet-18: A small ResNet with 18 layers, representing a traditional CNN with residual skip connections. It has about 11.7 M parameters. ResNet-18 is a canonical vision model; although larger ResNets exist, the 18-layer variant was selected to maintain capacity comparable to the others, as prior research in marine debris (e.g., Xue *et al.*, 2021b) often used mid-size ResNets. It serves as a reference for a straightforward convolutional feature extractor.

EfficientNet-B0: A modern CNN that was scaled for an optimal accuracy-efficiency trade-off. EfficientNet-B0 (~5.3 M params) uses a compound scaling strategy of depth, width, and resolution. It often outperforms older CNNs of similar size on ImageNet. This was included to see if its more advanced architecture (swish activations, MBConv blocks) and extensive ImageNet training confer an advantage in generalizing to underwater features.

DeiT-Tiny: A Vision Transformer model of tiny size (~5 M params), introduced by Facebook as a Data-Efficient Image Transformer. It is essentially a ViT model distilled during training to reach good performance despite its limited size. This was used to represent transformer architectures at a small capacity similar to EfficientNet-B0, and to test if transformers handle domain shifts differently than CNNs when both are of modest complexity.

ViT-B/16: A larger Vision Transformer (ViT-Base with patch size 16, ~86 M params) fine-tuned for this task. This model has much higher capacity and a different inductive bias (or lack thereof) compared to CNNs. ViT-B/16 was included to test the hypothesis that a higher-capacity model might learn more general features (potentially improving cross-domain robustness) given the same training data. It was initialized from ImageNet-21k or ImageNet-1k pre-trained weights (PyTorch's ImageNet-1k weights were used) and then fine-tuned.

CLIP ViT-L/14 (Zero-shot): The CLIP model with a ViT-L/14 vision backbone (around 307 M params in the vision encoder). CLIP was not fine-tuned; instead, it was used in zero-shot mode. This involves feeding the image and a set of text prompts to CLIP and letting it predict which prompt is most similar to the image. Specifically, prompts were crafted like "an underwater photo of {category}". For the positive class, descriptions such as "plastic trash" or "plastic debris" were used, and for the negative class, a description like "clean seascape with no human debris". CLIP produces an embedding for the image and for each text prompt. This study labels the image as plastic if the "plastic debris" prompt has a higher similarity to the image than the "no debris" prompt. This approach utilizes CLIP's learned visual semantics to perform binary classification without any task-specific training. CLIP ViT-L/14 was chosen because it is one of the largest publicly available CLIP models and has demonstrated strong zero-shot performance on many tasks.

Gemini 2.0 Flash (Zero-shot): A state-of-the-art multimodal large model from Google's Gemini family. It is an instruction-tuned model capable of analyzing images given a textual query. This was accessed via an API and used in a zero-shot fashion: for each test image, prompt was provided asking the model whether the image contains plastic waste or not. The following was used for prompting:

Binary image classification for underwater camera shots.

Task: Decide if there is visible plastic debris in the image.

Classes:

- "plastic": any visible plastic item or fragment (bags, bottles, wrappers, rope/twine, fishing line/net, foam, labels).
- "no_plastic": no visible plastic (e.g., fish, coral, rocks, sand, plants, bubbles, lighting artifacts, shadows, backscatter).

Decision rules:

- Treat small or translucent plastic as "plastic" if any part is clearly visible.
- Do not count natural materials (algae, seaweed, shells, driftwood) as plastic.
- If uncertain, choose the most likely class but reflect uncertainty in the score.

Output format (must match exactly): return a single JSON object

{"label": "plastic" | "no_plastic", "score": <number between 0 and 1>}

Return only the JSON object—no extra text.

No fine-tuning or additional training was done. The parameter count of Gemini Flash is not publicly disclosed, but it is presumably on the order of billions of parameters, making it by far the largest model in this comparison. Its inclusion helps gauge the performance ceiling of zero-shot AI among the scope.

2.4. Preprocessing & Splits

To assess and compare model performance, metrics that capture both binary classification effectiveness and threshold-independent behavior were utilized:

Precision: The fraction of images predicted as "plastic" that were actually plastic. High Precision means few false positives (*i.e.*, the model is not incorrectly flagging debris when none is present). In the balanced test set, Precision = TP/(TP + FP), where TP is true positives and FP is false positives. With equal positive and negative counts, Precision is especially indicative of how well a model avoids mistaking natural underwater features for trash.

Recall: The fraction of actual plastic images that the model correctly identified (also known as sensitivity). Recall = TP/(TP + FN), with FN being false negatives. High Recall means the model finds most of the plastic instances, even in the new domain. In this study's application, missing plastic (low Recall) would mean failing to detect debris that's actually there—potentially problematic for cleanup efforts.

F1-score: The harmonic mean of Precision and recall, F1 = 2 · (precision × recall)/(Precision + Recall). This metric summarizes the balance between Precision and Recall. It is useful when comparing models that might have a trade-off (one with higher Precision vs. another with higher Recall). An F1 closer to 1 indicates a model that is excelling in both aspects of the test data.

PR-AUC (Precision-Recall Area Under Curve): This metric evaluates the area under the Precision-recall curve, which plots Precision versus Recall as the classification threshold varies. Unlike the single-threshold F1, PR-AUC considers the model's output scores across all possible thresholds. It effectively measures how well the model ranks positive examples higher than negative ones, emphasizing performance on the positive class. Since the test set is balanced, PR-AUC in this case is directly comparable across models (with baseline 0.5 for random guessing). The average Precision (which is equivalent to PR-AUC) was calculated from each model's predicted probability scores for the positive class. This metric is sensitive to both how confident the model is in true positives and how it manages the Precision-recall trade-off across thresholds.

ROC-AUC (Receiver Operating Characteristic Area Under Curve): The ROC curve plots actual positive rate (Recall) against false positive rate at various thresholds. ROC-AUC is the area under this curve, reflecting the model's ability to discriminate between classes overall. An ROC-AUC of 1.0 means the model perfectly separates all positives and negatives by score; 0.5 means performance no better than chance. ROC-AUC was included for completeness, though in highly imbalanced scenarios, PR-AUC is often more informative. In the balanced setup, ROC-AUC and PR-AUC tend to be correlated, but ROC-AUC provides another view of overall separability. ROC-AUC can sometimes be higher than PR-AUC for the same model because it rewards true-negative performance; it indicates how well each model's scoring function separates the two classes, irrespective of a chosen threshold.

Additionally, the accuracy (the overall fraction of correct predictions) was recorded and confusion matrices were examined, but these are less nuanced given the balanced data. For reference, the accuracy is recorded but the above metrics are a better point of reference for insight-driven analysis. For the zero-shot models, since they do not output a calibrated probability by default, a proxy score was derived to compute AUCs. We also acknowledge the limitation for direct comparability to fine-tuned models whose outputs are calibrated probabilities. In CLIP's case, the difference in similarity between the "plastic" prompt and the "no plastic" prompt can serve as a decision score. For Gemini, a high numeric score was assigned to "yes (plastic)" and a low score to "no" answers to simulate a probability (acknowledging this is not as well-defined as for the other models).

3. Results

After training the five supervised models on the source dataset and evaluating all models on the balanced cross-domain test set, a comprehensive set of performance metrics was obtained. **Table 1** summarizes the Precision, recall, F1-score, PR-AUC, and ROC-AUC for each model. **Figures 1-3** visualize key comparisons. Overall. wide spread in recall values was observed, while most models maintained very high Precision on this test. The best performing model in terms of F1 and AUC was MobileNetV2, the smallest CNN—it remarkably achieved both the highest Recall and near-perfect Precision on the cross-domain test. In contrast, the zero-shot models showed divergent behaviors: CLIP attained moderate Recall but with poor Precision (indicating many false detections), whereas Gemini 2.0 Flash was extremely precision-oriented, catching fewer positives. The transformer-based models landed in between, with decent but not top scores. Below is a detailed model of these results.

Table 1. Cross-domain test performance by model.

Model	Precision	Recall	F1-score	PR-AUC	ROC-AUC
MobileNetV2	0.998	0.950	0.972	0.998	0.996
EfficientNet-B0	0.998	0.880	0.935	0.981	0.989
ViT-B/16 (ViT-Base)	0.991	0.850	0.915	0.940	0.982
DeiT-Tiny (ViT-Tiny)	0.995	0.820	0.899	0.922	0.970
ResNet-18	0.987	0.772	0.866	0.833	0.945
CLIP ViT-L/14 (Zero-shot)	0.564	0.804	0.663	0.510	0.810
Gemini 2.0 Flash (Zero-shot)	0.998	0.806	0.892	0.882	0.953

From **Table 1**, it is clear that MobileNetV2 attained 95.0% recall with 99.8% precision, yielding an F1-score of ~0.97. This indicates that MobileNetV2 detected the vast majority of plastics in the new domain while making virtually no mistakes on negatives – an impressive generalization result. EfficientNet-B0, another CNN, also achieved extremely high Precision (99.8%) but with a lower recall of 88.0%. Its F1 (~0.935) was second-best, and it has a high PR-AUC (~0.98), suggesting that most of the time it can distinguish plastics well, but it missed some that MobileNet caught. ViT-B/16, the large Vision Transformer, reached 85.0% recall and

99.1% precision (F1 ~0.915). DeiT-Tiny (the small ViT) had 82.0% recall and 99.5% precision (F1 ~0.899). Both transformers had slightly lower Recall than EfficientNet and much lower than MobileNet, though their Precision was still near perfect. The classic ResNet-18 had the lowest Recall of the fine-tuned models at 77.2%, indicating it failed to detect nearly a quarter of the plastic images; its Precision was 98.7%, lower than others, meaning it had a few more false positives as well. Consequently, ResNet's F1 (0.866) was the lowest among trained models—it struggled the most with the domain shift, comparatively.

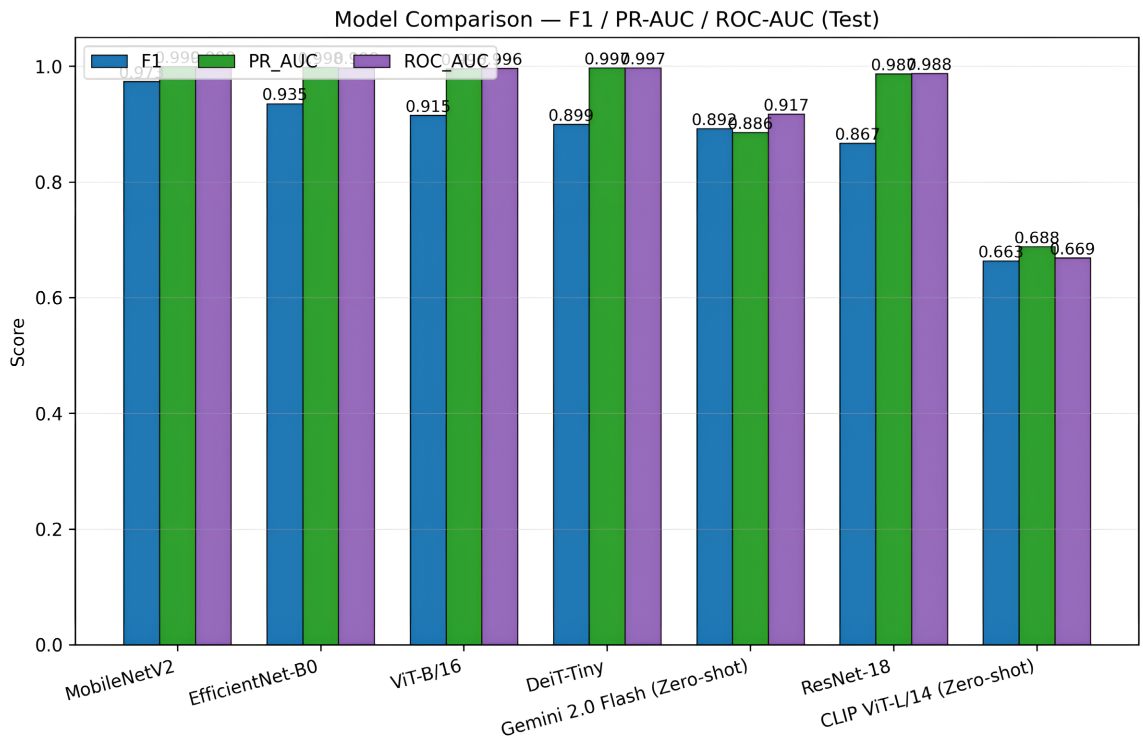


Figure 2. Test set Recall (sensitivity) for each evaluated model on the cross-domain test.

Figure 2: Higher Recall means the model detected a larger fraction of actual plastic images. MobileNetV2 achieved the highest Recall at 95.0%, significantly outperforming other models in finding plastics across the domain shift. Most other fine-tuned models reached 77% - 88% recall. The zero-shot CLIP and Gemini models each recovered ~80% of plastics without any training (comparable to a mid-tier trained model), though they differ significantly in Precision (see Figure 2). ResNet-18 had the lowest Recall (~77%), indicating it missed nearly one-fourth of the plastic debris in the new domain.

Figure 3: Precision is the proportion of predicted "plastic" images that were actually plastic. All the fine-tuned models (MobileNetV2, EfficientNet-B0, ResNet-18, DeiT-Tiny, ViT-B/16) maintained extremely high Precision (above 98%), with MobileNetV2, EfficientNet-B0, and Gemini Flash each at ~99.8% (virtually no false positives). DeiT-Tiny and ViT-B/16 also kept Precision>99%. ResNet-18's Precision (~98.7%) was slightly lower, corresponding to a few false alarms. In

stark contrast, the zero-shot CLIP model's Precision was only ~56%, indicating that nearly half of the images it flagged as plastic were actually clean (many false positives). This highlights CLIP's tendency to over-predict debris in unfamiliar underwater scenes, whereas Gemini's Precision was on par with the fine-tuned models due to its conservative predictions.

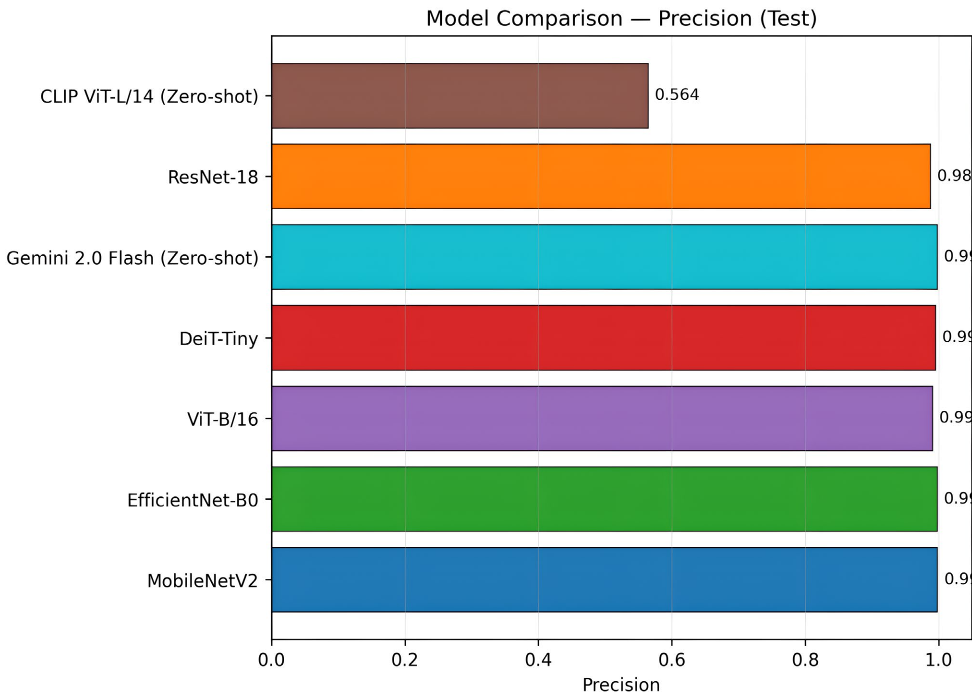


Figure 3. Test set precision for each model.

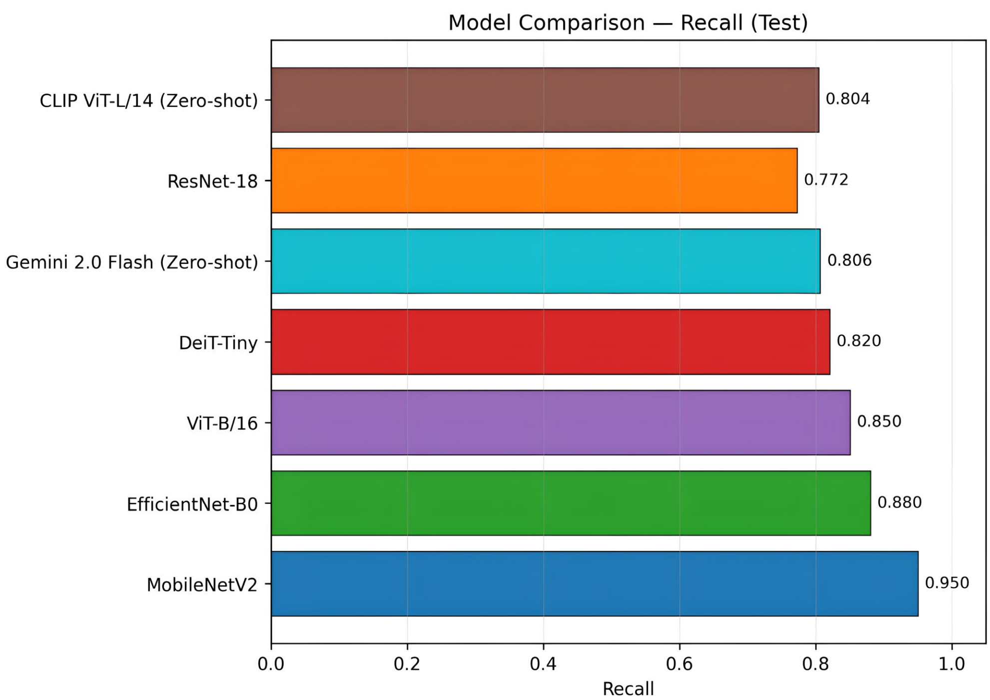


Figure 4. Comparison of F1-score, PR-AUC, and ROC-AUC.

Figure 4: Each group of three bars corresponds to a model (ordered by descending F1). F1-score (blue) balances Precision and recall—MobileNetV2 stands highest (~0.97), followed by EfficientNet (~0.94) and ViT-B/16 (~0.92). PR-AUC (green) reflects the area under the precision-recall curve; MobileNetV2 again leads (~0.998), indicating it achieves near-perfect Precision across recall levels. EfficientNet and ViT-B/16 show very high PR-AUC in the ~0.94 - 0.99 range. ResNet-18 is slightly lower (~0.98), due to its recall shortfall. CLIP's PR-AUC (~0.69) is far below the others, confirming poor Precision-recall performance—its curve drops off steeply once recall increases. Gemini Flash's PR-AUC (~0.88) is much higher than CLIP's, but lower than the fine-tuned models, reflecting that it misses some positives to maintain high Precision. ROC-AUC (purple) is high for all the fine-tuned models (~0.97 - 0.99), showing excellent overall separability; MobileNet and EfficientNet reach ~0.996. CLIP's ROC-AUC (~0.67) is the lowest, indicating relatively weak separation of scores for positives vs. negatives (consistent with its many false positives). Gemini's ROC-AUC (~0.92) is respectable, though well below the top performers, implying its decision criterion, while nearly perfect for negatives, leaves some positives with low confidence scores.

Examining these results, several points stand out. First, MobileNetV2's dominant performance is somewhat unexpected—one might assume a larger model like ViT-B/16 would generalize better. MobileNet's success may be due to a combination of factors: it might have benefited from a strong inductive bias (convolutions focusing on local features relevant to plastic, like edges and textures) [13], and perhaps it did not overfit the source domain as much as larger models could have, thereby remaining flexible enough to capture plastics in the new domain. EfficientNet-B0 also did very well, aligning with its reputation as a well-regularized CNN [14]. ResNet-18's poorer showing suggests that having seen similar training data is not enough; its architecture or optimization might have left it less prepared for domain shift. Possibly, ResNet-18 learned more specific features (or it lacked the capacity to fully learn all plastic variations, causing it to miss some novel appearances). Aside from inductive bias, pre-training likely mattered: because ImageNet is largely object-centric, it tends to bias features toward centered objects and local textures rather than broader scene context.

Figure 5: Confusion matrices share the same layout: accurate labels on the vertical axis, predicted labels on the horizontal. "True negatives" (top-left) are no_plastic correctly predicted; "true positives" (bottom-right) are plastic correctly predicted. From the matrices, the key metrics are derived: MobileNetV2 achieved the highest F1 (~0.97), with both very high Precision (≈0.998) and strong Recall (≈0.95). Gemini (zero-shot) shows a drop in Recall (≈0.81), leading to lower F1 (~0.90), although its Precision remains very high. ResNet-18 lags in Recall the most (≈ 0.77), giving it the lowest F1 among fine-tuned models (~0.87).

The transformers (DeiT-Tiny and ViT-B/16) exhibited a trend of improving Recall with model size—ViT-B/16 (85% recall) > DeiT (82%) – but they still did not reach EfficientNet or MobileNet levels. It is plausible that 12 epochs of

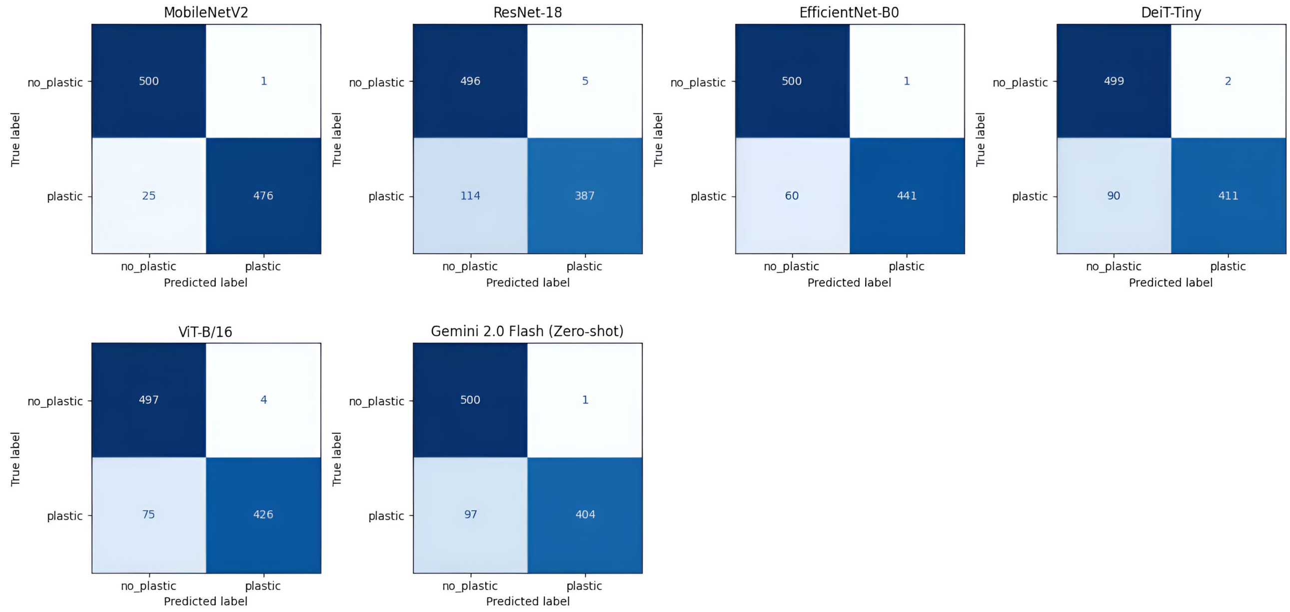


Figure 5. Test-set confusion matrices.

fine-tuning on a relatively small dataset were not sufficient for ViT-B/16 to adapt completely; transformers often need more data or regularization when training data is limited. Nonetheless, both ViTs maintained a very high precision, indicating that when they did predict plastic, it was usually correct. They were more likely to err by omission (false negatives) than by commission [15]. It is important to note that the transformers' PR-AUC, while high (~0.94+), was slightly lower than the CNN counterparts' in this study's test, hinting that their confidence calibration might not be as tight—possibly due to needing more careful fine-tuning or data augmentation.

The zero-shot models' contrasting Precision-recall balance is a crucial finding. CLIP clearly had the knowledge to detect plastic (an 80% recall rate out of the box is impressive), but it was also triggered by many non-plastic cues. On inspecting some false positives of CLIP, it was found that it often marked images with bright reflections or sunbeams underwater as plastic. For instance, one test image of a sunlit seabed with rippling caustic patterns was labeled "plastic" by CLIP—presumably because the irregular bright patch might resemble a piece of plastic or the glint of plastic. Similarly, some images of corals or sponges were misclassified by CLIP, perhaps because their shapes or colors were misinterpreted as debris. CLIP's language-supervised training likely exposed it to many contextual associations (e.g., "coral reefs often mentioned with marine pollution"), leading it to be liberal in declaring pollution. Without fine-tuning, it lacked the context to restrain those predictions in a domain-specific way.

Gemini, conversely, often responded "No plastic" unless the item was very obvious. From a few sample outputs, Gemini correctly identified large, clear pieces of trash (such as a bottle or bag in open water), but it tended to report no plastic if the plastic was small, camouflaged, or ambiguous. This explains its near-zero false positives—it essentially only responded when in confidence. It did, however, result in false negatives for subtle cases (e.g., an image where a small fragment of plastic film was caught on coral might be missed). This behavior is consistent with an LLM-based vision model that aims to avoid hallucinating an object unless it is strongly confident. For practical deployment, one might actually combine such behaviors: for example, using a very high-precision model like Gemini to flag extremely certain cases and a more sensitive one like CLIP or MobileNet as a secondary sweep—but that is beyond this study's current scope.

Error Analysis

To better understand misclassifications, common error patterns attributable to the domain characteristics were reviewed:

Glare and Light Artifacts: Many underwater images have lighting artifacts—caustic wave patterns, glare from the sun, or backscatter (floating particles causing bright spots). Models sometimes misconstrued these as plastic. For example, CLIP frequently treated bright reflections or sunbeams as indicative of plastic litter (likely because shiny highlights might resemble the sheen of plastic). One false positive image was a sun-dappled seabed with wave caustics; CLIP strongly signaled "plastic" where there was none. ResNet-18 had a couple of false positives on images with intense lens flare or reflection, suggesting its features were less robust to these distortions. Overall, glare-induced errors were more of an issue for the zero-shot models, underscoring that robust preprocessing or training on varied lighting conditions can help. Future models might incorporate special data augmentation (or even polarization filters in capture) to avoid confusing glare with plastic highlights.

Reef Structures and Biota: Colorful coral, sponges, or even marine organisms were sometimes mistaken for plastics, especially by less specialized models. CLIP had trouble with corals—in one case, an image of a bright orange sponge was labeled as plastic debris (possibly due to its unnatural-looking color or texture, which CLIP's training might associate with discarded objects). ResNet-18 similarly had a false positive on an image of a white coral that initially looked a bit like a crumpled plastic bag. One way to reduce such errors could be to incorporate more diverse "distractor" examples during training (varied coral and biota images), so the model learns to ignore them. It is critical to note that Gemini Flash did not call corals plastic; likely, its broader understanding and caution prevented that confusion, erring on the safe side.

Small or Camouflaged Plastic: The most common reason a plastic item was missed (false negative) by models was that it was either very small in the image or blended into the environment. For instance, a thin fishing line or a tiny plastic fragment on the sand was sometimes missed. ResNet-18 and DeiT-Tiny, having lower capacity, missed such subtle instances more often. This is inherently more challenging for small objects—essentially, a fine-grained detection task is being treated as classification. Models might benefit from explicit object-detection approaches or saliency mechanisms to specifically handle small debris.

Plastic Lookalikes: Some false negatives occurred because the model seemed to interpret the plastic as a natural part of the scene. For example, an image of a plastic bag partially covered in algae on a rock was missed by EfficientNet-B0—likely it thought it was just part of the algae or rock. This is challenging even for human observers. Increasing training samples of aged marine litter or employing multimodal cues (perhaps combining vision with sonar or spectral imaging in robotics) could help address this corner case beyond purely visual methods.

4. Discussion

The comparative results of this study yield several insights into model performance under domain shift for marine debris detection. Below, key findings and implications are discussed:

Model capacity is not the sole determinant of cross-domain performance. Despite its simplicity, MobileNetV2 achieved the highest Recall and F1 score on the cross-domain test, outperforming larger models such as ResNet-18 and ViT-B/16. This suggests that the inductive biases and training dynamics of MobileNet (or similar efficient CNNs) might better capture the core visual cues of "plastic vs. non-plastic" without overfitting to domain-specific context [16]. In contrast, ResNet-18, with more parameters, surprisingly underperformed—it may have learned more domain-specific features from Dataset A (e.g., focusing on particular background patterns that correlate with "no plastic" in training, which did not generalize to Dataset B).

CNNs vs. Transformers—differences in generalization: this study's results show CNNs (MobileNet, EfficientNet) slightly out-generalize ViTs in this scenario. One possible reason is data augmentation and inductive bias. CNNs come pre-equipped to recognize translationally local features; they might latch onto object edges, contours, and textures that signify plastic (e.g., wrinkles of a plastic bag, straight edges of packaging), regardless of the global context. ViTs, on the other hand, attend broadly and might inadvertently pick up on scene context (for example, associating "open water with a certain color tone" with no debris because that was common in training). If the context shifts in the new domain (different water color or background elements), that could throw them off. Without more aggressive regularization, a ViT might also yield over-smooth confidence estimates—being less specific about novel-looking inputs.

Zero-shot vs. fine-tuned—Precision vs. Recall trade-offs: The zero-shot models provided an illuminating contrast. CLIP's high Recall vs. Gemini's high Precision essentially bracket the performance of the fine-tuned models. CLIP, with no task-specific training, could recall ~80% of plastic images, which is on par with a smaller fine-tuned model (DeiT or ResNet). This underscores the power of CLIP's pretraining—it had likely seen many instances of "ocean plastic" in its image-text data [13], giving it a broad (if unrefined) ability to recognize the concept. However, CLIP lacked context specificity: it was too trigger-happy, leading to an untenable precision (~56%). In practical terms, deploying CLIP zero-shot would mean an automated system that raises a lot of false alarms (over half of its alerts would be false).

Importance of training data diversity: The experiments reinforce the notion from related work that training on a narrow domain limits test performance on a different domain [5]. Even though this study's training and test sets were both "underwater", the subtle differences caused significant drops in Recall for some models (ResNet's 77% recall indicates it did not generalize well to the new domain features). If the training data used had included more varied reef scenes, ResNet might have performed better. Domain generalization techniques, such as those in ADOD or DG-YOLO [5] [17], could enhance the performance of models like ResNet or ViT by explicitly learning domain-invariant features.

Model architecture and error patterns: The error analysis suggests that CNNs vs. transformers handle background confusion differently. CNNs seemed better at ignoring irrelevant backgrounds (MobileNet had no coral false positives, and EfficientNet/ResNet had only minor issues). Possibly because CNN filters focus on local patterns—if they see a coral texture that was not associated with plastic in training, they likely output "no". Transformers might consider global similarity—if an image as a whole looks unlike the training images (a new reef type), they might hesitate or misclassify.

Practical considerations—speed and deployment: Although not directly measured in this study, the model sizes and types have implications for real-time use. MobileNetV2 and EfficientNet-B0 are fast on both GPUs and CPUs, making them suitable for on-board use in AUVs or drones with limited computational resources. ResNet-18 is also reasonably fast. ViT-B/16 is much heavier; on a small dataset, it is fine, but inference might be slower and more memory-intensive (transformers can be demanding). The zero-shot models pose other challenges: CLIP ViT-L/14 is large but can be run locally with optimization (it has ~300 M parameters, which is borderline for an embedded system, but possible with pruning or quantization).

Limitations and future work: This study focused on image-level classification (presence/absence of plastic in an image). In practice, object detection (localizing the debris) should be used. Some of the domain robustness issues likely carry over to detection frameworks. The high performance of MobileNetV2 in classification suggests that using MobileNet as a backbone in an object detector (such as SSD or YOLO) could also yield robust cross-domain results, as observed by Liu & Zhou (2023), who improved YOLOv5 with a MobileNet backbone for debris detection [11]. Additionally, image-level positives provide only weak supervision: a "plastic" label may correspond to a small fragment, encouraging models to rely on contextual cues rather than the object itself. This limitation is well-documented in weakly supervised detection/segmentation and is exacerbated for small objects in underwater imagery (e.g., under turbidity/backscatter), reinforcing the need for localized annotations (boxes/masks) in future work [18].

5. Conclusions

This paper presented a comprehensive evaluation of domain robustness in marine plastic detection using a suite of vision models, from compact CNNs to large vision transformers and zero-shot multimodal models. A challenging cross-domain test scenario was created—training on one underwater image dataset and testing on a different one—to simulate a real-world deployment where the operating environment differs from the lab data.

MobileNetV2, a lightweight CNN, emerged as the top performer with 95% recall and 99.8% precision on the cross-domain test. It effectively detected nearly all plastic items in the new domain, while rarely misclassifying a clean scene. This suggests that small, well-structured CNNs can capture the essential features of marine debris in a generalizable way, even outperforming larger models in this context.

Among other supervised models, EfficientNet-B0 and ViT-B/16 also showed strong generalization, albeit with slightly lower Recall (88% and 85%, respectively) but similarly high Precision (~99%). ResNet-18 lagged, missing a substantial number of plastics, indicating that older architectures might struggle more with domain shifts, possibly due to limited capacity or less effective feature extraction for this task. DeiT-Tiny, despite its small size, performed respectably (82% recall, 99.5% precision), demonstrating that transformers can operate at low scales. However, within conducted experiments, they did not surpass the CNN counterparts.

Zero-shot models yielded valuable insights: CLIP ViT-L/14, without any task-specific training, managed to identify ~80% of the plastics, demonstrating the power of language-image pretraining. However, its over-eagerness led to many false positives (precision only ~56%), which would be problematic in practice. In contrast, Gemini 2.0 Flash was almost flawlessly precise (~99.8%), rarely "crying wolf", yet it caught only about 81% of plastics, reflecting a more cautious approach. These represent two different operating points on the Precision-recall spectrum—one could imagine tuning either model to shift along that curve if given some feedback or calibration.

The analysis of errors revealed that glare (lighting artifacts) and certain natural objects, like coral, can confuse models, especially if they have not been exposed to them during training. Fine-tuned models mostly learned to ignore these, but zero-shot CLIP was susceptible. Meanwhile, camouflaged or tiny plastic pieces remain a challenge for all models—a limitation of working with low-resolution images and subtle visual cues.

Model architecture and training regimes have a significant influence on generalization. A convolutional inductive bias helped MobileNet and EfficientNet focus on the right visual cues (edges, textures) that transferred across domains. Transformers, with global attention, performed well but may require additional data or augmentation to fully realize their potential in addressing domain shift problems. The results hint that carefully curated training (including diverse conditions and backgrounds) can narrow the gap further.

From an application viewpoint, if one had to choose a single model from this study to deploy on an underwater drone for plastic detection, MobileNetV2 would be an excellent choice given its top accuracy and efficiency. If the computing environment allowed for a larger model and required even more robustness, one might consider EfficientNet-B0 or ViT-B/16; however, the marginal gains are small in within the conducted scenario. If relying on cloud or offline analysis, incorporating a system like Gemini 2.0 Flash as an assistant could drastically reduce false alarms, essentially confirming detections with near certainty. On the other hand, a combination of models might offer the best of both worlds: for example, a fast CNN to catch most debris, plus a sophisticated validator model to double-check positives.

In conclusion, the ability to reliably detect marine plastic debris across different underwater environments is within reach using today's vision models. The results demonstrate with even modest supervised training, one can build a vision system that is highly accurate in spotting underwater plastics, regardless of the locale—a promising development for efforts to monitor and ultimately reduce marine pollution. The best practices are: choose a model with a good balance of inductive bias and capacity (MobileNetV2 was ideal in this case), train with as diverse data, and consider cross-checking with zero-shot AI for added confidence. Future work can extend this benchmark by exploring unsupervised domain adaptation and moving to object detection frameworks for pinpointing debris in images. Additionally, as new multimodal models (like future versions of Gemini or other large vision-language models) become available, it will be interesting to see if their zero-shot performance further improves, perhaps one day obviating the need for any fine-tuning. For now, this study provides a baseline and insights that can inform the design of robust underwater plastic detectors using contemporary AI models.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Meng, J., Lang, L., Lu, Z., Tang, X. and He, H. (2024) Design and Implementation of Garbage Detection in Water Area Based on YOLOv5 Algorithm. *Proceedings of the 3rd International Conference on Public Management and Big Data Analysis*, *PMBDA* 2023, Nanjing, 15-17 December 2023, 1. https://doi.org/10.4108/eai.15-12-2023.2345400

[2] Walia, J.S., Haridass, K. and Pavithra, L.K. (2025) Deep Learning Innovations for Underwater Waste Detection: An In-Depth Analysis. *IEEE Access*, **13**, 88917-88929. https://doi.org/10.1109/access.2025.3569344

[3] Hipolito, J.C., Sarraga Alon, A., Amorado, R.V., Fernando, M.G.Z. and De Chavez, P.I.C. (2021) Detection of Underwater Marine Plastic Debris Using an Augmented Low Sample Size Dataset for Machine Vision System: A Deep Transfer Learning Approach. 2021 *IEEE 19th Student Conference on Research and Development* (*SCOReD*),

Kota Kinabalu, 23-25 November 2021, 82-86. https://doi.org/10.1109/scored53546.2021.9652703

[4] Sánchez-Ferrer, A., Valero-Mas, J.J., Gallego, A.J. and Calvo-Zaragoza, J. (2023) An Experimental Study on Marine Debris Location and Recognition Using Object Detection. *Pattern Recognition Letters*, **168**, 154-161. https://doi.org/10.1016/j.patrec.2022.12.019

[5] Saoud, L.S., Niu, Z., Sultan, A., Seneviratne, L. and Hussain, I. (2023) ADOD: Adaptive Domain-Aware Object Detection with Residual Attention for Underwater Environments. 2023 21*st International Conference on Advanced Robotics* (*ICAR*), Abu Dhabi, 5-8 December 2023, 3-4. https://doi.org/10.1109/icar58858.2023.10436502

[6] Moorton, Z. (2024) State of the Art Applications of Deep Learning within Tracking and Detecting Marine Debris: A Survey. arXiv: 2403.18067.

[7] Chen, R., Wu, J., Peng, Y., Li, Z. and Shang, H. (2023) Solving Floating Pollution with Deep Learning: A Novel SSD for Floating Objects Based on Continual Unsupervised Domain Adaptation. *Engineering Applications of Artificial Intelligence*, **120**, Article ID: 105857. https://doi.org/10.1016/j.engappai.2023.105857

[8] Radford, A., *et al.* (2021) Learning Transferable Visual Models from Natural Language Supervision. arXiv: 2103.00020.

[9] Shukla, S. and Sonalkar, K. (2025) Can GPT-4o Mini and Gemini 2.0 Flash Predict Fine-Grained Fashion Product Attributes? A Zero-Shot Analysis. arXiv: 2507.09950.

[10] van Lieshout, C., van Oeveren, K., van Emmerik, T. and Postma, E. (2020) Automated River Plastic Monitoring Using Deep Learning and Cameras. *Earth and Space Science*, **7**, e2019EA000960. https://doi.org/10.1029/2019ea000960

[11] Liu, J. and Zhou, Y. (2023) Marine Debris Detection Model Based on the Improved YOLOv5. 2023 3*rd International Conference on Neural Networks*, *Information and Communication Engineering* (*NNICE*), Guangzhou, 24-26 February 2023, 725-728. https://doi.org/10.1109/nnice58320.2023.10105682

[12] Huangfu, Z., Li, S. and Yan, L. (2024) Ghost-YOLO V8: An Attention-Guided Enhanced Small Target Detection Algorithm for Floating Litter on Water Surfaces. *Computers*, *Materials & Continua*, **80**, 3713-3731. https://doi.org/10.32604/cmc.2024.054188

[13] Sasaki, K., Sekine, T. and Emery, W. (2024) Enhancing the Detection of Coastal Marine Debris in Very High-Resolution Satellite Imagery via Unsupervised Domain Adaptation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **17**, 6014-6028. https://doi.org/10.1109/jstars.2024.3364165

[14] Kako, S., Kataoka, T., Matsuoka, D., Takahashi, Y., Hidaka, M., Aliani, S., *et al.* (2026) Remote Sensing and Image Analysis of Macro-Plastic Litter: A Review. *Marine Pollution Bulletin*, **222**, Article ID: 118630. https://doi.org/10.1016/j.marpolbul.2025.118630

[15] Kim, J.W., *et al.* (2021) Learning Transferable Visual Models from Natural Language Supervision. arXiv: 2103.00020.

[16] Xue, B., Huang, B., Chen, G., Li, H. and Wei, W. (2021) Deep-Sea Debris Identification Using Deep Convolutional Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **14**, 8909-8921. https://doi.org/10.1109/jstars.2021.3107853

[17] Zhao, F., Huang, B., Wang, J., Shao, X., Wu, Q., Xi, D., *et al.* (2025) Seafloor Debris Detection Using Underwater Images and Deep Learning-Driven Image Restoration: A Case Study from Koh Tao, Thailand. *Marine Pollution Bulletin*, **214**, Article ID:

117710. https://doi.org/10.1016/j.marpolbul.2025.117710

[18] Shao, F., Chen, L., Shao, J., Ji, W., Xiao, S., Ye, L., *et al.* (2022) Deep Learning for Weakly-Supervised Object Detection and Localization: A Survey. *Neurocomputing*, **496**, 192-207. https://doi.org/10.1016/j.neucom.2022.01.095

Open Journal of Modelling and Simulation, 2026, 14(1), 21-43
https://www.scirp.org/journal/ojmsi
ISSN Online: 2327-4026
ISSN Print: 2327-4018

A Simulation Framework for Exploring the Impacts of Vehicle Platoons on Mixed Traffic under Connected and Autonomous Environment

Dian Yuan, Ardeshir Faghri, Erfan Ranjbar

Department of Civil, Construction and Environmental Engineering, University of Delaware, Newark, USA
Email: diany@udel.edu, faghri@udel.edu, eranjbar@udel.edu

How to cite this paper: Yuan, D., Faghri, A. and Ranjbar, E. (2026) A Simulation Framework for Exploring the Impacts of Vehicle Platoons on Mixed Traffic under Connected and Autonomous Environment. *Open Journal of Modelling and Simulation*, **14**, 21-43.
https://doi.org/10.4236/ojmsi.2026.141002

Received: October 31, 2025
Accepted: December 14, 2025
Published: December 17, 2025

Abstract

Vehicle platooning, first studied as an application of Intelligent Transportation Systems (ITS), is increasingly gaining attention in recent years as autonomous driving and connected vehicle technologies advance. When platooned, vehicles communicate within the platoon and operate with coordination to maintain a relatively steady state status with each other and with the outside. The major goal of this study is to build a conceptual simulation framework to help with exploring the impacts of connected and autonomous vehicle platoons on the existing traffic. The first part of this work effort is reviewing autonomous and connected vehicle technologies for depicting the functional structure of a platooning-ready connected and autonomous vehicle (CAV) platform. Then models and simulation tools are reviewed to break down the simulation framework into two levels—vehicle level and traffic level. The vehicle-level model provides in-depth modeling of CAVs and platooning modules. The traffic-level simulator provides the simulation of the existing traffic with the built CAV platoons. The simulation framework has been developed by integration and usage of GIS, MATLAB/Simulink, SUMO, and OMNeT++. GIS tools are used to gather the necessary traffic data. MATLAB/Simulink serves as the platform for vehicle-level modeling and simulation. SUMO and OMNeT++ are used to build traffic and communication simulations, respectively. The completed model was used to conduct two case studies based on a section of the US Interstate Highway to explore the impacts of CAV platoons on existing traffic. The results indicate that, with the existing traffic pattern and infrastructure design, traffic can be improved after the introduction of CAV platoons, even after taking into consideration the rate of traffic growth. Moreover, deploying dedicated lanes and separating CAV platoon traffic from the non-platooning traffic can benefit the traffic using such output as the travel speed/time and delay measures. However, using such new traffic patterns and infrastructure designs is not recommended for a low percentage of CAV platoon traffic.

Keywords

Car Platooning, Connected and Autonomous Vehicles (CAVs), Traffic Simulation

1. Introduction

In the past decade, autonomous-vehicle (AV) technologies have become a central focus in transportation research and practice. Their potential safety benefits largely by reducing human error are widely recognized. In parallel, advances in wireless communications have made connecting vehicles in reliable, low-latency networks increasingly feasible. Rather than treating automation and connectivity as separate paths, recent work integrates them to improve overall traffic performance. Platooning connected and automated vehicles (CAVs) is one such integration, with reported benefits that include improved safety, reduced energy use and emissions, mitigation of human error, and increased roadway capacity. Early Intelligent Transportation Systems (ITS) programs explored automated platoons in the 1990s; with today's sensing, computation, and communications, CAV platooning now shows substantial real-world potential.

The goal of an Automated Driving System (ADS) is to assume the driving task. Before introducing ADS technologies, it helps to outline what human drivers do. After planning a trip and securing occupants, drivers must 1) perceive and interpret the vehicle state and the surrounding environment, 2) make decisions that ensure safe, comfortable, and efficient progress to the destination under uncertain traffic conditions, and 3) execute control by steering, braking, and accelerating. These three functions map directly to a commonly adopted AV architecture: a) perception, b) planning, and c) control.

Enabling connected vehicles requires multiple technologies. Inter-vehicle communication (IVC) was identified early in ITS research as a means for vehicles to "see" farther than onboard sensors alone. As the Internet of Things (IoT) has matured, connected-vehicle capabilities have advanced markedly. Three core modalities are vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and broader vehicle-to-everything (V2X). V2V enables direct data exchange among vehicles, extending environmental awareness. V2I connects vehicles to roadside and intersection devices; for example, signals can broadcast phase timing so approaching CAVs adjust speed profiles for safer, smoother passage, while signal controllers can optimize phases using approaching vehicle states. V2X encompasses all other relevant links, including vehicle-to-pedestrian (V2P), vehicle-to-grid (V2G) for managing electric-vehicle charging and storage at scale and vehicle-to-cloud (V2C), which provides access to richer data and additional computational resources.

Given these enabling technologies, it is increasingly practical to assess how CAV platoons may affect traffic operations and infrastructure. Planners, designers, engineers, and policymakers need tools to anticipate impacts under varied conditions and design choices. Although several software packages simulate traffic with AVs and connectivity, few offer explicit, flexible modeling of CAV platoons or the ability to modify platooning logic and CAV-related technologies. A simulation framework with high adaptivity is therefore essential.

The novelty of this work lies in the unique integration of GIS, MATLAB/Simulink, SUMO, and OMNeT++ into a unified multi-resolution simulation framework, combining vehicle-level platoon dynamics with network-level traffic behavior, and validating the system using real-world I-95 operational data.

1.1. Purpose and Problem Statement

This article addresses the lack of a reliable, flexible simulation tool for evaluating the impacts of CAV platoons on existing traffic systems. The framework must adapt to evolving technologies and contexts. To capture both vehicle behavior and network effects, it employs two simulation levels:

1) a lower-level module that models individual CAVs and platoons from a vehicle-centric perspective, and

2) a higher-level traffic module that examines network-level implications under varying demand, control, and infrastructure designs.

1.2. Research Questions

1) What is the functional architecture of a connected and autonomous/automated vehicle, and which technologies are essential for platoon-capable CAVs?

2) How should vehicles be formed into platoons, how do platoons operate effectively in connected/automated environments, and what is an appropriate functional architecture for CAV platooning?

3) How are platoon-capable CAVs and CAV platoons modeled at the microscopic (vehicle) level, and how should mixed traffic with platoons be simulated?

4) Once platooned CAVs are introduced, how should their network-level impacts be evaluated?

Solving transportation problems requires attention to temporal and spatial dimensions. Roadway plans and designs often span a decade or more, and the rapid evolution of CAV technologies necessitates anticipating future operating conditions. Governments and standards bodies worldwide are actively shaping policies and technical guidance for real-world deployment. Within this context, vehicle platooning, a form of cooperative driving, has drawn significant interest. Developing a tool to explore its impacts is both timely and important for surface-transportation planning, design, and operations.

1.3. Objectives

1) Survey CAV technologies and applications, and formulate a functional architecture for both standalone CAVs and CAV platooning.

2) Develop vehicular models for (a) human-driven vehicles, (b) non-connected automated vehicles, and (c) connected and automated vehicles with platoon functions.

3) Model platoon-capable CAV operations in a microscopic environment.

4) Build traffic-level simulation scenarios with varying market shares of platoon-capable CAVs and differing roadway designs.

5) Integrate the vehicle-level models, traffic simulations, and test scenarios to evaluate CAV-platoon impacts under diverse conditions; verify the framework with sample tests.

6) Validate the framework through case studies on real infrastructures and traffic conditions, assessing outcomes across different CAV-platoon shares.

1.4. Background

This section has three parts. First, it presents the functional architecture of connected and automated vehicles (CAVs) adopted in this study. Next, it reviews implemented and emerging technologies for connected driving, automated driving, and vehicle platooning. Finally, it defines and explains the components that make up the architecture.

Functional Architecture of a Connected and Automated Driving System

Drawing on current implementations and ongoing developments, Figure 1 summarizes the functional architecture used in this study. A viable CAV must operate safely with limited human intervention, moving occupants and goods from origin to destination safely, correctly and reliably. To achieve these goals, the architecture is organized into three functional layers:

- **Perception**

Acquires and interprets information about the vehicle and its surroundings.

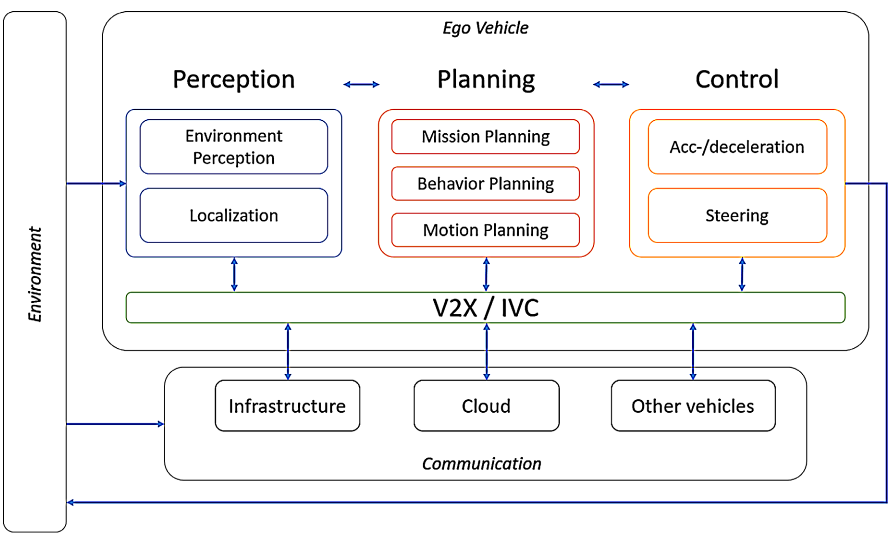


Figure 1. Functional Architecture of a CAV.

Core capabilities include object/event detection and tracking, localization and mapping, and state estimation. Sensor fusion and connectivity (e.g., V2X) extend situational awareness beyond line-of-sight.

- **Planning**

Determines what the vehicle should do to complete the trip safely, comfortably, and efficiently. This spans route planning, behavior/decision planning (e.g., yielding, lane selection, platoon join/leave), and motion planning that generates feasible trajectories under traffic rules, policies, and constraints.

- **Control**

Executes the planned actions accurately by generating steering, throttle, and braking commands. This includes longitudinal and lateral controllers, trajectory tracking, and fault-tolerant mechanisms to maintain performance under disturbances or component degradations.

The remainder of this section reviews the enabling technologies for connected driving, automated driving, and platooning, and then details how these components interact within the architecture.

Implemented technologies include widely available driver-assistance features on the market. The Society of Automotive Engineers (SAE) defines vehicles that activate a single automated function as Level 1 driving automation. Table 1 shows

Table 1. Driver assistant technologies.

Category	Technology	Brief Description
Collision Warning	Forward Collision Warning	Warning driver of a potential crash with the vehicle ahead
	Lane Departure Warning	Warning driver while the vehicle is approaching or crossing lane markers
	Rear Cross Traffic Warning	Warning the driver of potential crash while driving backward
	Blind Spot Warning	Warning the driver of objects in driver's blind spot
Collision Intervention	Automatic Emergency Braking	Appling brakes or enhancing braking when a forward collision is imminent
	Pedestrian Automatic Emergency Braking	Breaking once a pedestrian(s) is detected ahead and a collision is imminent
	Rear Automatic Braking	Breaking when a potential crash is imminent while driving backward
	Blind Spot Intervention	Braking or steering (either or both) when a vehicle is detected when ego vehicle is changing lanes
Driving Control Assistance	Adaptive Cruise Control	Adjusting speed automatically to maintain a safe distance from the vehicle ahead
	Lane Centering Assistance	Steering continually to keep the vehicle driving at the middle of its lane
	Lane Keeping Assistance	Steering automatically to keep the vehicle form departing the lane
Other Systems	Automatic High Beams	Activating and deactivating high beams depending on the distance from oncoming vehicle or the vehicle ahead
	Backup Camera	Providing a clear view behind the vehicle
	Automatic Crash Notification	Reporting automatically to an emergency responder when a crash occurred

the levels of driving automation per the SAE J3016 standard. An integrated system with multiple driver-assistance functions operating simultaneously is considered Level 2.

The National Highway Traffic Safety Administration (NHTSA) groups driver-assistance technologies into four functional categories: collision warning, collision intervention, driving-control assistance, and other systems. Table 2 lists commonly implemented technologies within these categories. Features in Table 2 are often combined to deliver broader assistance.

Table 2. Summarizes the setting items of the calibrated model.

Setting Items	Values	Other Information
Platoon Size (number of Veh.)	2, 5, 8	-
ACC Time Headway (sec)	1, 1.5, 2	-
CACC Constant Spacing Distance (m)	2, 5, 12, 20	-
Velocity of slow-moving vehicle (mph)	55, 60, 65, 70	-
Communication Delay (ms)	1 to 10	PDF: $f_{hn}(x;1,5)$
Perception Delay (ms)	5 to 20	PDF: $f_{hn}(x;5,5)$

For example, Forward Collision Warning (FCW) can be integrated with Automatic Emergency Braking (AEB) to reduce the risk of rear-end collisions or mitigate severity when a crash is unavoidable. When a potential forward collision is detected, FCW alerts the driver to take corrective action; if the driver does not respond in time, AEB applies the brakes automatically. Another example is the pairing of Rear Cross-Traffic Warning (RCTW) with Rear Automatic Braking (RAB), which work together to help avoid collisions with overlooked objects while reversing—mirroring the FCW+AEB interaction. During steady cruising, combinations of Adaptive Cruise Control and Lane-Keeping Assistance/Lane-Centering Assistance can improve comfort for both driver and passengers.

2. Methodology

In this section, modeling and simulation algorithms in both vehicle and traffic scopes are reviewed. As introduced in previous sections, the main goal of this study is to develop a simulation framework to explore the impacts of CAV platoons on traffic. The whole simulation framework and modeling will be computational. The entire simulation has been divided into two parts: microscopic simulation and macroscopic simulation. Different from general terminologies used to categorize traffic simulation software with different scopes, the meaning of these two terms will be explained in the following sections.

The first section of this section is a review of studies that focus on CAV platooning algorithms and technologies. Besides algorithms and technologies, the tools and approaches adopted in those studies for modeling and simulating are presented. The next section explains the vehicle's kinematic and dynamic models, communication topology, and control algorithms related to autonomous and connected techniques. In the third section, traffic simulation tools/platforms are reviewed by going through official documents and related publications.

2.1. Research Review

In 2011, Schünemann developed a runtime infrastructure for coupling communication and traffic simulators and conducting sophisticated V2X simulations. The infrastructure is called V2X Simulation Runtime Infrastructure abbreviated as VSimRTI [1]. The infrastructure was designed to manage a federation of simulators. A Federate consists of a simulator, a federate ambassador, and a VSimRTI ambassador. The simulator could be coupled with VSimRTI, which includes SUMO, VISSIM, JiST/SWANS, OMNeT++, etc. VSimRTI has been upgraded to Eclipse MOSAIC.

Segata *et al.* [2] developed and verified a simulation tool for automated platooning in mixed highway scenarios with the combination of a communication simulation platform and a traffic simulation platform, OMNeT++ and SUMO, in 2012. The proposed platooning protocol in the study was CACC-based. The results showed that with 30% penetration of platoon-enabled vehicles, over 80% of platoons have a size of less than three. The recorded largest size of a platoon during the simulation was 8. In 2014, Segata *et al.* published the works done based on this simulation tool and named the completed tool PLEXE.

In 2013, Zhao and Sun developed a simulation framework that explores the vehicle platooning and car-following behavior under the CV environment. A six-vehicle CACC platoon was simulated on a microscopic traffic simulation platform VISSIM. Examined platoon maneuvers include forming, adjusting, splitting, dismissing, and joining. They concluded that the lane capacity is positively correlated with the market penetration of CACC-based platoons. On the contrary, platoon size impacts the capacity subtly. The minimum desired headway for ACC vehicles was set to 1.4 seconds as well as 0.5 seconds for CACC vehicles [3].

Jia *et al.* presented a disturbance-adaptive (DA) design for VANET-based vehicle platoons. The design aimed at improving the platoon's stability of communication and operation. Within a DA platoon, four roles were played by vehicles. The first is the Leader. Like the role in a common platoon system, a Leader makes decisions on platoon-level operations (formatting, splitting, merging, broadcasting the existence of the platoon, etc.). The second role is the Tail. As the name indicates, the last vehicle in a platoon plays this role, and oversees communication with the leader of next platoon. The third role is called Relay. Relay vehicles perform data-forwarding in a multi-hop VANET environment. The last role is Member, who simply follows the plan decided by the Leader. Based on such theory, authors developed a driving strategy for Leaders of DA-platoon. The driving strategy adopts a sliding mode controller and determines the desired inter-platoon gap by gathering information from the Tail of the preceding platoon. The principles guaranteed the gap to be small enough for not breaking the connectivity with adjacent platoons and large enough for avoiding collisions. The principles also helped with determining the platoon size. Then, simulations were conducted for algorithm validation [4].

In 2014, a distributed framework was developed for coordinating HDVs for platooning. The core of this framework is a virtual controller system that coordinates HDVs at each vertex (e.g., each major intersection) for platooning. The objective is to maximize the earnings (energy savings minus costs). A case study was conducted based on the German Autobahn network. 0 to 7000 HDVs were initialized in the network. The results showed that as the number of HDV increases more fuel saving can be achieved [5].

Artery, a simulation framework based on OMNeT++, was introduced in 2015 as an extension of Veins. Recently, an extension called Artery-C to adapt C-V2X communication approaches was released. An overview of Artery-C was also published. The pros of Artery-C include 5G-selected features, open-source licensing, and capabilities for modeling facilities [6].

In 2015, Santini *et al.* presented a consensus-based approach for vehicle platooning under IVC environments. The consensus algorithm was implemented to achieve equal inter-vehicle gaps within a platoon. Validation simulations were conducted. The results showed that the approach performed better than the classic CACC approach by improving the stability and convergence of the platoons. Later, in another publication, Santini *et al.* presented their work on validating the consensus-based platooning approach when vehicle communication topology changes due to platoon maneuvers. Simulations were developed and run upon PLEXE. The results proved the reliability of the approach, even while a platoon with heterogeneous vehicles is maneuvering. Tested maneuvers include join-at-tail, leave-at-tail, join-at-middle, and leave-at-middle. Since the communication topology is a typical leader and predecessor following topology, vehicles in a platoon communicate with the platoon leader and the two adjacent vehicles (one ahead and the one after) [2].

In 2015, Li *et al.* reviewed relevant studies and presented a four-component framework of the vehicle platoon system. The four components of the framework are 1) Node dynamics, 2) information flow topology, 3) distributed controller, and 4) formation geometry. Node dynamics describes the behavior of every platooned vehicle and all others involved. Information flow topology means how vehicles exchange information with each other. Feedback controllers handle feedback control with neighboring information. Formation geometry indicates the desired inter-vehicle gap within a platoon [7].

Deng published a simulation framework for modeling and analyzing heavy-duty vehicle (HDV) platoons. The simulation framework was built upon the commercial simulation platform, VISSIM. Within the framework, a fuel consumption model was embedded to estimate the influences of HDV platoon on fuel saving. Input data for fuel consumption estimation is obtained by recording the state of each vehicle. The truck's state data includes motion state (lane, longitudinal position, speed, acc-/deceleration), platooning information (in-platoon position, inter-vehicle distance), and temporal data (time instance). Three cases were studied to figure out the influences of HDV platoon and platoon formation maneuvers. The results showed that, with HDV platoons, even if the aggregate highway velocity dropped, the traffic flow rate increased on a two-lane highway section. Otherwise, under medium/high traffic scenarios, the interest in reducing platoon formation time conflicted with the arrival time of the HDVs.

Ribeiro *et al.* proposed a platooning management protocol. In the study, the protocol covers the procedures of platoon creating, merging, and dissolving operations as well as vehicle joining and leaving maneuvers. Simulation and testing were conducted on a combination of SUMO, V2X Simulation Runtime Infrastructure (VSimRTI), and ns-3. Two-truck platoons were introduced into the simulation network. The results showed that, with the protocol,

a) vehicle joining the platoon from rear has been completed in 69 seconds(s) on average, or in 114.2 seconds when joining from the adjacent lane;

b) on average, a vehicle leaving the platoon took 23.3 s to change to the adjacent lane, or 1 s if it's the last vehicle in the platoon and leaves by increasing gap to its preceding vehicle;

c) adjusting the gap took 12.6 s for a joining maneuver or 17.4 s for a leaving maneuver on average;

d) merging two platoons was similar to the joining maneuver, but with 62.7 s adjusting on average;

e) dissolving maneuver took 3.8 s on average.

Additionally, message latency was limited to 100 milliseconds (ms), which meets the required maximum delay. The capacity of a lane was proven to be better with platoons.

In Bang and Ahn's 2017 publication, a platooning strategy for CAVs based on the spring-mass-damper system is described. With the system, the longitudinal platoon control of CAVs by controlling the spring constant and damping coefficient is presented. The maximum acceleration/deceleration, mass, and length were used to determine the controlling parameters. Moreover, different relations between the spring constant and traffic flow were considered when developing simulation scenarios. The results proved that, with the critical damping coefficient, the maximum efficiency of completing platoon formation could be achieved when the maximum relation between flow and spring constant was selected.

In 2017, Jain *et al.* published their work on developing a prediction-based framework for vehicle platooning. An MPC-based control algorithm, which solely relied on V2V communication, was developed and implemented for vehicle platoon control. Both simulations and experiments were conducted to test the algorithm. Simulations were performed on the Dominion framework developed by German Aerospace Center. A two-vehicle platoon with both 5G-V2X and 802.11p communications was used for the experiments. Both simulation and experiment results showed the excellent potential of controlling a vehicle platoon with pure V2V communication. The authors also concluded that 5G performed better than 802.11p in providing V2V/V2I communication because of its larger data rate and communication range.

In 2017, Liu *et al.* published their research on the platoon system engineering process that considers safety and cybersecurity issues. The engineering process has four steps—1) defining the safety goal, 2) defining the attack model, 3) deriving the security goal, and 4) deriving functional security requirements. Due to the tight coupling among vehicles in a platoon, the impact of a cyber-attack against the platoon system could lead to decreasing stability, platoon dissolution, and even collision. Functional security requirements of a platoon system included but were not limited to detecting false messages, ensuring the timeliness of messages and responses, and keeping messages intact from attackers. The general approaches to developing a platooning system with the proper capability to maintain safety and security include optimizing the gap between vehicles with security consideration, enabling cyber-attack detection, and deploying fail-safe mechanisms to eliminate harm when attacks are encountered. In the paper, a proactive platooning approach was presented. The approach calculated the optimal acceleration difference threshold based on the desired acceleration under CACC and pure ACC situations. Moreover, the desired gap was determined based on the desired acceleration. Simulations with PLEXE and MATLAB validated the algorithm [8].

In 2018, Ramezani *et al.* developed a simulation model for exploring the influences of CACC-based truck platooning operations on traffic. Aimsun Next MicroSDK was used for developing the simulation platform. The aspects of trucks, including desired acceleration speed, were computed explicitly and implemented in the simulation. A case study was conducted based on a 15-mile urban section of the I-710 Northbound an Interstate highway in Southern California. The results showed that truck CACC platoon could increase the speeds of cars by reducing congestion when penetration rates reach a high level (over 80%). However, in on-ramp areas, since truck platoons used the rightmost rule, the merging traffic had to wait longer when penetration rates were low. Generally, when penetration rates of CACC trucks reached 100%, the benefits included easing congestion propagation and increasing the average speed of traffic at uncongested areas. Otherwise, similar benefits were not found with CACC car platoons [9].

In a publication by Ibrahim *et al.*, a co-simulation framework developed by the authors for vehicle platooning was presented. The framework consists of ns-3, SUMO, and MATLAB. Ns-3 simulated the packet broadcast of vehicles. SUMO simulated the traffic. Control algorithms were developed in MATLAB and replaced the algorithms that were given in SUMO. MATLAB also performed as the interface between SUMO and ns-3. The developed algorithms focused on longitudinal acceleration control. Model predictive control and state-feedback control algorithms were implemented for the upper- and lower-layer control. The upperlayer controller determined acceleration or deceleration regarding the gap to the preceding vehicle. The lower-layer controller worked to eliminate errors in acceleration/deceleration. Tests were conducted for framework validation. Simulations based on realistic highway scenarios were conducted as well. The results showed that both the severity and frequency of platoon speed fluctuations increased when packet losses grew.

Vieira *et al.* developed a realistic simulation framework for vehicular platooning based on the Robotic Operating System (ROS) framework and published a paper in 2019. ROS is a popular framework for designing robotics applications. An integration of Gazebo (ROS robotic simulator) and OMNeT++ was presented. ROS publish/subscribe mechanisms played a critical role in data delivery and simulator synchronization. In a later Vieira *et al.* publication, after more work had been done, this realistic simulation framework was named COPADRIVe.

In 2019, Gerrits *et al.* developed a study exploring the influences of opportunistic truck platooning matchmaking algorithms. One is First-Viable Match (FVM), and the other is Best-Match (BM). The FVM takes waiting time into account as the cost. Once a match can lead to a positive earning (subtracting savings over costs), the match is selected. The BM selects the match with the highest earnings within a searching area. Properties of the truck (hourly wage, urgency, brand, destination, matching locations) were added as factors into the simulation model. The results showed that BM performed better than FVM on saving. The wage savings are significant when platooning trucks.

Sethuraman *et al.* developed a simulation to evaluate the impacts of bus platoons on traffic. The simulation scenario was developed based on a 16-kilometer (KM) section of an urban roadway in Singapore with two major signalized intersections. The simulation was run on the VISSIM platform. Both the quality of services of the bus and the performance of the traffic were assessed. Numerical analysis showed that, generally, the simulation results showed that bus platooning increased the operational speed of buses and other cars and, as a result, the overall delay was reduced for both types of vehicles. Moreover, output data indicated similarity with platooning trucks since platooning buses reduced the aerodynamic drag and fuel consumption. Correlation between the number of buses in the platoon showed a positive coefficient with energy savings.

In 2019, Hoef *et al.* published their predictive framework for dynamic HDV platoon coordination. The presented framework aimed to coordinate the in-route formation of platoons. The core of the framework is a platoon coordinator. A layered control system architecture for coordinated platooning was also presented. The layers are the service layer, strategic layer, tactical layer, and operational layer. On the first layer, transport tasks are managed. On the second layer, strategies, such as coordinating platoons, are performed. For the remaining two layers, platoon management systems and vehicle control systems are deployed.

In 2019, Hyun *et al.* published a paper that overviews a statistical verification framework for a platooning system of systems (SoS). The framework, called

StarPlateS, consists of three modules, scenario generation module, simulation module, and verification module. The first module performed platoon configuration generation and scenario generation. The simulation module was handed over to a SUMO/OMNeT++ integrated extension, Vehicular Network Open Simulator (VENTOS). The verification module checked the achievement rates of goals with the statistical model checking (SMC) algorithms. The two checked properties of the SoS were the throughput within a specific time horizon and the rejection rate of operation.

Since 2006, Sommer *et al.* have worked on a model library for OMNeT++. It's named Veins. A 2019 overview summarized the developments of the library. Now, Veins supports simulations not only relating to the IEEE 802.11p family but also LTE and Visible Light Communication (VLC). Veins does not manage road vehicle simulations. However, by bidirectional coupling with SUMO through the Traffic Control Interface (TraCI), users may customize vehicular mobility models on demand. Models on the communication layer are created by Veins in the OMNeT++ simulator to represent vehicles. Then, by establishing mapping with the mobility models in SUMO, the cyber-physical system of a connected-vehicle environment is simulated.

Quadri *et al.* published work on a MEC-based vehicle platoon control framework for vehicle platooning in 2020. MEC is the abbreviation of multi-access edge communication. Being different from some distributed controllers, a MEC-based controller offers a centralized approach to platoon control. Simulations relying on SUMO and Python-based applications were conducted for two types of scenarios—sinusoidal and real-trace vehicle movement patterns. According to the results, with a MEC-based controller, inter-vehicular spacing can be shortened to 5 meters or less. However, since the round-trip time (RTT) of cloud computing could hardly be achieved below 150 ms on average, deploying the controller onto the cloud was not suitable for such centralized control.

In 2021, Hidayatullah and Juang published their study on the centralized and distributed control framework under homogeneous and heterogeneous platoons. In the paper, they used PreScan and MATLAB/Simulink to establish simulations to investigate the string stability of both centralized and distributed control frameworks comprehensively. Features that include vehicle dynamics, sensing, and V2X communication are added to the simulations. The performance index integral square error (ISE) was used to evaluate two frameworks. The results showed that, with 0.05 latency and 30% packet loss probability, a distributed framework achieved slightly less ISE of mean spacing error than a centralized framework.

In 2021, Miekautsch *et al.* published their study on a situation-dependent communication topology for platooning heterogeneous vehicles. In that publication, platoon systems were studied via the four-component framework. A heterogeneous vehicle platoon indicates that a platoon was formatted by vehicles with various configurations (power-train time constant, max acc-/deceleration, max speed). Two situations were simulated, emergency braking and count-in. For the emergency braking situations, the author presented a flexible reversed Leader-predecessor-follower (LPF) communication topology. With such topology, the vehicle's configuration and the order within its platoon were considered. By deploying this topology, a platooned vehicle with a higher maximum deceleration rate can brake harder compared to where the traditional LPF topology was deployed, and the vehicle was limited to use the lowest deceleration rate of all vehicles within its platoon.

In 2021, a publication by Xu *et al.* revealed an open-source simulation tool for cooperative driving automation (CDA). It's called OpenCDA. The key features of OpenCDA were summarized as IFMBC, integration, full-stack platform, modularity, benchmark, connectivity, and cooperation. OpenCDA selected CARLA and SUMO as the simulator for traffic. The hierarchical architecture of the tool was demonstrated. The top layer was the "PlatoonManager" class. Then is the "VehicleManager" class. The bottom layer includes "PerceptionManager", "LocalizationManager", "BehaviorManager", "ControlManager", and "V2XManager". The internal operational sequence of the OpenCDA modules is illustrated in the OpenCDA logic flow shown in Figure 2.

2.2. Simulation Framework

The simulation framework developed in this research integrates four key platforms including GIS, MATLAB/Simulink, SUMO, and OMNeT++, into a co-simulation environment to model vehicle-level dynamics, network-level traffic behavior, and real-time V2X communications of Connected and Autonomous Vehicle (CAV) platoons. This integrated structure allows for the detailed exploration of both microscopic and macroscopic effects of platooning, bridging the gap between control-theory-based vehicle modeling and transportation-system-level

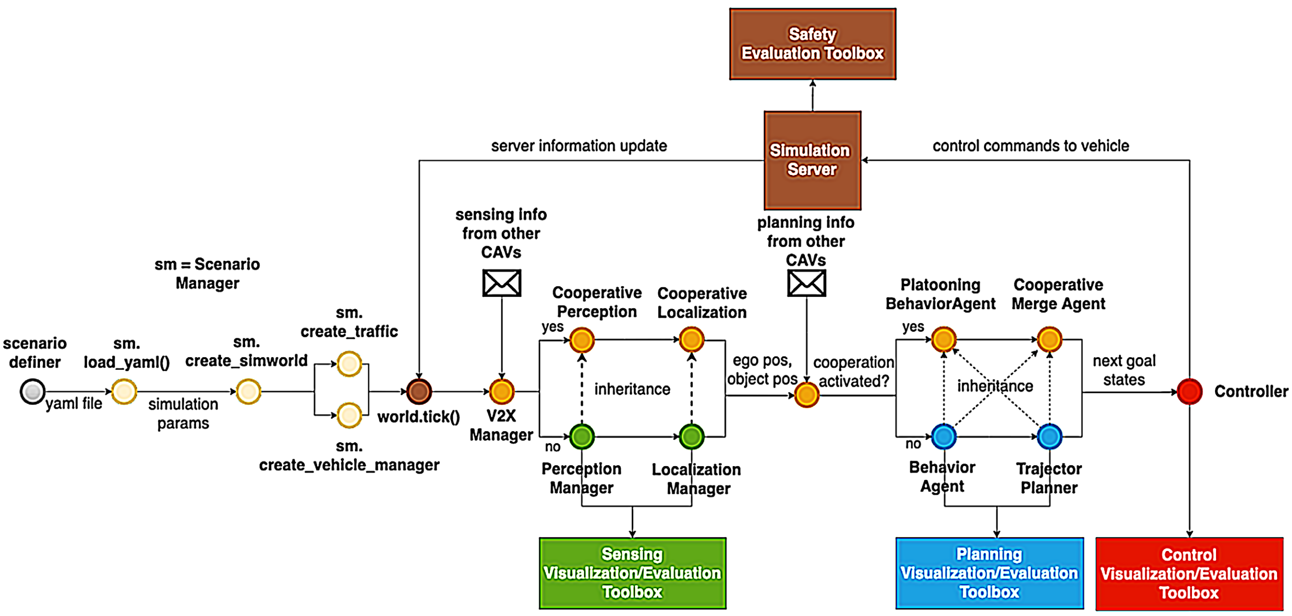


Figure 2. OpenCDA logic flow.

traffic evaluation.

This suite of platforms was selected because each provides distinct and complementary modeling capabilities. GIS ensures spatial accuracy and realistic roadway geometry; MATLAB/Simulink enables detailed modeling of vehicle dynamics and cooperative control algorithms; SUMO provides scalable microscopic traffic simulation; and OMNeT++ supports realistic modeling of V2X communication performance. Together, these tools offer an adaptable and comprehensive environment for studying CAV platoon behavior.

2.2.1. Architectural Overview

The framework is built upon the Veins-PLEXE architecture, a widely used simulation interface for coupling traffic and communication simulators. In this structure, each vehicle in the SUMO traffic simulator is mirrored by a corresponding communication node (module) in OMNeT++, following the APP-MAC-PHY hierarchy compliant with IEEE 802.11p and IEEE 1609.4 standards. These layers collectively simulate the Wireless Access in Vehicular Environments (WAVE) and ETSI ITS-G5 communication stacks, enabling the emulation of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) messaging within platoons. The interaction among SUMO, OMNeT ++, and the TraCl interface in our integrated simulation setup is illustrated in Figure 3.

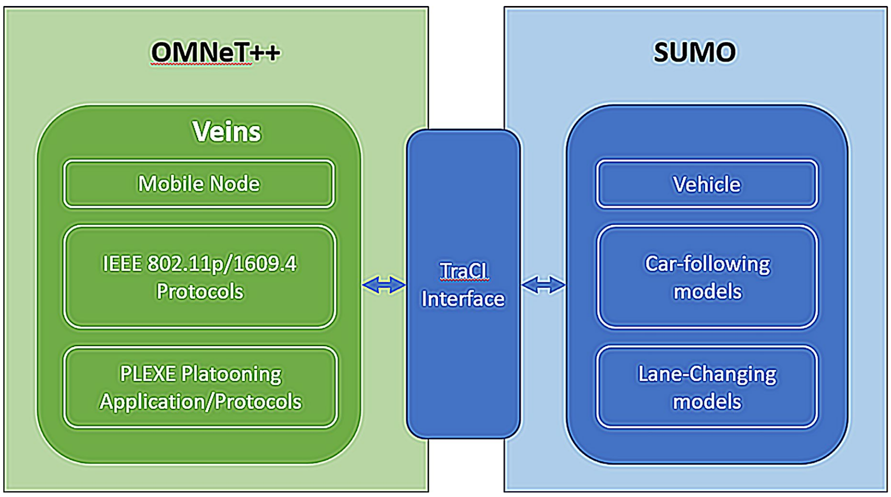


Figure 3. Architecture of Integrated Simulation based on PLEXE.

The integration is managed through the Traffic Control Interface (TraCI), a TCP-based client-server protocol that synchronizes state data between SUMO and other platforms such as MATLAB or OMNeT++. This ensures that changes in vehicle states (e.g., speed, acceleration, position) in SUMO are instantaneously reflected in OMNeT++ communication nodes and MATLAB-based controllers, forming a cyber-physical feedback loop.

2.2.2. Vehicle-Level Modeling

At the microscopic scale, MATLAB/Simulink is employed to simulate vehicle dynamics, platoon control, and autonomous driving algorithms. Three distinct control models were implemented to represent different vehicle types:

- Cruise Control (CC) for Human-Driven Vehicles (HDVs),
- Adaptive Cruise Control (ACC) for partially autonomous vehicles, and
- Cooperative Adaptive Cruise Control (CACC) for connected platoon-enabled CAVs.

Each control model is designed around the perception-planning-control architecture of an Autonomous Driving System (ADS). The longitudinal controller determines acceleration using spacing errors, relative speeds, and communication inputs, while lateral control manages lane-keeping and merging behaviors.

Communication latency and perception delay are also introduced in the Simulink environment, typically set between 1 - 10 ms for V2V/V2X communication and 5 - 20 ms for perception delay, modeled as Gaussian distributions with parameters μ and σ calibrated through field studies.

The platoon management logic includes state transitions such as "join platoon", "leave platoon", and "split/merge platoon". For each vehicle, the control system dynamically switches between ACC and CACC modes depending on whether communication data is available from preceding vehicles. This approach supports realistic simulation of disturbances such as temporary communication losses, and the resulting adaptive control responses.

2.2.3. Traffic-Level Simulation

The macroscopic traffic simulation is executed in SUMO (Simulation of Urban Mobility), chosen for its scalability, open-source flexibility, and lightweight computational requirements. SUMO allows multi-lane, multi-vehicle microscopic traffic modeling while accommodating heterogeneous vehicle types. Using GIS-based roadway geometry, the northbound I-95 corridor from the Delaware House Travel Plaza to the Christiana Interchange was reconstructed with high positional accuracy. Real-world parameters such as lane configurations, speed limits, and traffic demand profiles were directly imported from DelDOT datasets and spatial layers.

In this simulation layer, PLEXE extends SUMO to represent CAV platoon behaviors including:

- Platoon formation and dissolution,
- Inter-vehicle communication topology updates (Leader-Predecessor-Follower),
- Speed synchronization,
- Cooperative lane changing and merging, and
- String stability assessment.

The platoon's operational logic is event-driven, controlled by thresholds in relative distance, speed, and communication delay, ensuring that vehicles can join or leave platoons dynamically based on traffic context and communication reliability.

2.2.4. Communication Network Simulation

OMNeT++ operates as the dedicated communication simulator. Each vehicle in SUMO is instantiated as a mobile communication node in OMNeT++, connected through the Veins framework. These nodes exchange Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) using the IEEE 802.11p protocol. The Leader–Predecessor–Follower topology is used, where the platoon leader transmits acceleration and speed data downstream, while each follower relays acknowledgment messages upstream.

Key communication metrics such as latency, packet delivery ratio (PDR), and network throughput, are continuously logged to assess communication quality. In the developed framework, average delays remained within 5 - 8 ms, and PDR exceeded 97% under nominal load, confirming stable connectivity for cooperative control. Additionally, multi-hop relaying was modeled for extended platoons, enabling stable information flow beyond direct line-of-sight.

2.2.5. Integration and Synchronization

The co-simulation achieves synchronization through three primary interfaces:

1) SUMO-OMNeT++ (Veins/PLEXE): Exchanges vehicle states and message events in real time through TraCI.

2) MATLAB-SUMO: Transfers dynamic control outputs (e.g., acceleration commands) and retrieves aggregated performance indicators such as travel time, flow, and density.

3) GIS–SUMO: Provides geometric and spatial accuracy for roadway networks, demand assignment, and coordinate referencing.

The simulation timestep was fixed at 0.1 seconds to ensure smooth coupling across all platforms. Synchronization accuracy was validated by comparing time-stamped vehicle trajectories, confirming sub-millisecond temporal deviations between SUMO and OMNeT++ layers.

2.2.6. Framework Capabilities

The resulting simulation environment enables:

- Detailed evaluation of string stability and inter-vehicle spacing performance.
- Network-wide assessment of mobility and emission impacts under varying platoon penetration levels.
- Testing of dedicated lane policies, mixed-traffic interactions, and communication delay effects.
- And analysis of platoon coordination failures due to network disruptions.

Overall, the proposed framework functions as a comprehensive multi-resolution platform, capable of translating low-level control dynamics into observable large-scale traffic phenomena, thus offering a powerful tool for assessing the real-world implications of CAV platooning on major highway systems such as I-95. The complete workflow of the integrated simulation framework used in this study is elaborately shown in **Figure 4**.

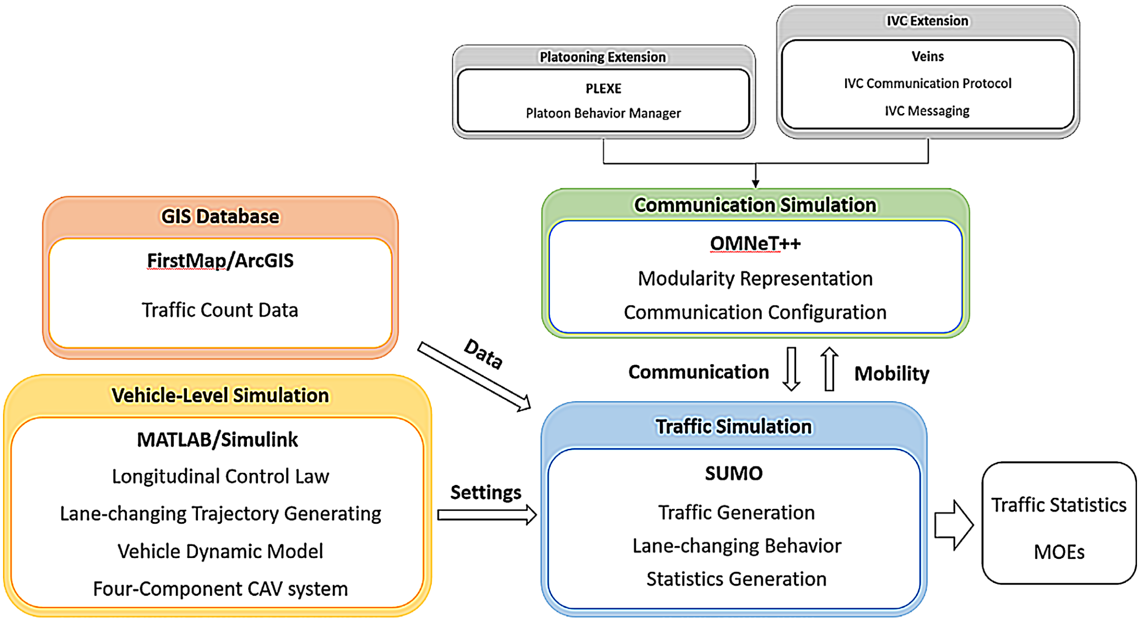


Figure 4. Logic flow of the instance of the proposed simulation framework in this study.

2.3. Testing and Validation

The testing and validation of the simulation framework were carried out in two main stages: calibration of vehicle dynamics and control parameters, followed by validation against real-world data collected from the I-95 corridor in Delaware.

2.3.1. Calibration Process

Calibration focused on ensuring that the longitudinal and lateral control laws reflected realistic driving dynamics. Three categories of vehicles were represented:

- Human-Driven Vehicles (HDVs) using CC control,
- Autonomous Vehicles (AVs) using ACC control, and
- Platoon-Enabled CAVs (P-CAVs) using CACC control.

Key parameters such as platoon size (2, 5, 8 vehicles), ACC time headway (1 - 2 s), and CACC constant spacing (2 - 20 m) were systematically tested. During calibration runs, MATLAB/Simulink simulated low-speed and lane-changing scenarios to assess string stability and speed synchronization between vehicles. Two measures of effectiveness (MOEs) were employed:

1. Integral of Gap Error, quantifying spacing deviations over time.
2. Speed Synchronization Duration, measuring how long a following vehicle takes to match its leader's speed after perturbations.
3. Results indicated that smaller headways (5 m) improved stability without compromising safety, reducing speed synchronization time by up to 35% compared to larger gaps. The effect of communication or perception delays was limited, with less than 15% degradation in MOE performance, confirming the robustness of the model. Control parameters were finalized as: CACC damping ratio = 1, bandwidth = 0.2 Hz, and leader-weight factor = 0.7, ensuring optimal response across different platoon compositions.

2.3.2. Validation on I-95 Corridor

The validated scenario was modeled on the northbound I-95/Delaware Turnpike, extending from the Delaware House Travel Plaza to the Christiana Interchange. This 6-mile section was selected for its representative geometric design and heavy commuter traffic. Using field data obtained through GIS-based datasets, DelDOT sensors, and manual counts, baseline traffic volumes, densities, and average speeds were extracted. These data served as the benchmark for comparison with simulated results.

Simulation outputs, including speed profiles, flow rates, and density distributions, were statistically compared to field observations using two sample t-tests. The resulting p-value ≈ 0.07, evaluated using a significance level of $\alpha = 0.05$, indicated no statistically significant difference between the two datasets, confirming the simulation's ability to reproduce real-world conditions. Further verification confirmed that platoon maneuvers such as join-at-tail, leave-at-middle, and split/merge operations remained stable under high traffic demand, even with message delays of up to 10 ms.

In addition to behavioral validation, network-level communication reliability was evaluated using OMNeT++ metrics such as packet delivery ratio, latency, and inter-platoon connectivity duration. The average communication delay remained within 5 ms, ensuring that control commands were executed within acceptable real-time bounds. The consistency between observed and simulated macroscopic variables including average travel time and flow-density relationships, confirmed the high fidelity of the integrated model.

Through this calibration and validation process, the framework proved capable of accurately reproducing both vehicle-level dynamics and corridor-level traffic performance. It therefore provides a reliable analytical environment for exploring future deployment strategies of CAV platoons, including dedicated-lane designs, control algorithm optimization, and mixed-traffic interactions under realistic conditions on I-95.

3. Results

Two case studies were conducted using the validated framework to evaluate the potential effects of CAV platoon traffic on roadway performance.

3.1. Case Study I—Baseline I-95 Operations

This scenario simulated existing infrastructure without dedicated lanes, under traffic volumes projected for 2026, 2031, and 2036. The analysis revealed that as CAV platoon penetration increased, both average travel speed and throughput improved. Specifically, with 10% CAV platoon penetration, average speeds increased marginally by 0.1% - 0.2%; at 30%, improvements reached 0.9% - 1.6%; and at 50%, gains rose to approximately 3% - 4%. Moreover, the duration of sub-desired speed travel, an indicator of congestion, decreased by about 20% when platoon shares ranged between 30% and 50%. These findings highlight that even moderate platoon integration can enhance mobility under high-volume traffic.

3.2. Case Study II—Dedicated Platoon Lanes

The second scenario investigated both added and converted dedicated lane configurations for platoon-exclusive use on the same I-95 segment. Design volumes representing 2036 were employed to reflect future congestion levels. When dedicated lanes were introduced, network efficiency improved further at medium to high platoon penetration rates (≥20%). The simulations demonstrated that converting one existing lane for platoon use initially caused slight slowdowns for low penetration levels (<15%), due to underutilization. However, when penetration exceeded 25% - 30%, total corridor throughput increased, and average travel times decreased significantly. Scenarios with added lanes (rather than converted ones) yielded the best overall system performance.

Environmental metrics also showed benefits: fuel consumption and CO_2 emissions declined as platoon penetration grew, attributed to smoother accelerations, reduced headways, and fewer stop-and-go events. At 50% penetration, CO_2 emissions decreased by roughly 6% - 8% relative to the baseline, indicating both traffic and environmental advantages from coordinated platoon deployment.

A detailed analysis of the emission profiles was performed for selected vehicles under different penetration rates of CAV platoons. The emission curves showed that CAV platoons generally produced smoother acceleration and deceleration patterns, resulting in reduced CO_2 emissions among regular vehicles as platoon penetration increased. However, CAV platoon vehicles themselves exhibited higher instantaneous emissions at higher speed limits due to more frequent acceleration adjustments required to maintain string stability. The emission trends over time indicated that when the penetration rate reached 30% - 50%, the overall system-level CO_2 emissions decreased by approximately 6% - 8% compared to the baseline scenario.

The simulation results showed that when speed limits increased for the CAV platoon dedicated lane(s), CAV platoon flow could achieve better travel time than the regular lanes. Moreover, converting a lane to dedicated with low percentages (settings A and B) of CAV platoon traffic increased the regular lanes' burden (lowered the capacity). But as the percentage increased, the average travel speed of all types of traffic increased. When adding a dedicated lane instead of converting one, settings F, G, and H showed the benefits for both types of traffic. However, when the percentage went higher, such a pattern solely improved the regular traffic, but situations of CAV platoon traffic remained similar for both adding and converting. As a result, improvement for both types of traffic have been achieved. As the percentage of CAV traffic increased, the improvement increased gradually.

4. Conclusions

This study developed and validated a multi-platform simulation framework linking vehicle-level platoon dynamics with corridor-level traffic performance.

This study has several limitations. Communication conditions were assumed to be ideal, without cybersecurity threats or extreme packet-loss scenarios. Human-driven vehicle behavior in mixed traffic was represented with simplified models, and external factors such as adverse weather or sensor noise were not explicitly simulated. These assumptions should be considered when interpreting the results and assessing their general applicability.

The vehicle-level model is instantiated on the MATLAB/Simulink platform. The vehicle's longitudinal and lateral dynamic and control theories were designed and implemented to calibrate the vehicle model for the network-level simulations. By feeding the calibration results (critical control parameters) to the traffic-level simulations, the users could interpret the simulations more precisely. Calibrations were conducted to tune the selected longitudinal control laws, CC, ACC, and CACC. The three laws are used to represent the three components of traffic, human-driven vehicles, AVs, and CAVs. Then, the results of calibrations were injected into the integrated CAV simulators to explore the impacts of CAV platoons on traffic under different scenarios. PLEXE framework was selected to perform traffic simulations. PLEXE is a simulation framework developed upon the integration of SUMO and OMNeT++. The former is one of the most popular graphical microscopic traffic simulators, and the latter is also a well-known communication network simulator in the inter-vehicle communication field. One of the main reasons to use these two platforms, MATLAB/Simulink and PLEXE/SUMO/OMNeT++, is that they allow users to customize simulation models extensively and in-depth.

For vehicle-level simulation, a review of vehicle autonomy and connected driving technologies was conducted. It summarized the technological structure of CAV systems. In this study, the traditional functional structure of the autonomous driving system is selected to build the vehicle model. The three main components of a traditional autonomous driving system are perception, planning and control. Communication systems perform in an inter-component manner and provide information to leverage the performance of all three components of the ADS. Derived from the CAV system and platooning management system, the basic functional structure of a CAV platooning system is completed.

Following the review of the CAV and platooning technologies, modeling and simulation algorithms were reviewed. Vehicle models included the longitudinal dynamic model, lateral dynamic model, three degrees-of-freedom vehicle body model, and string stability model. Longitudinal control laws of lane-changing trajectory methods were also investigated. Then, communication issues such as communication topology and communication network simulator were studied and used in the developed system.

The traffic-level simulation was conducted by utilizing SUMO, the OMNeT++ simulator, and a CAV platooning simulation framework, PLEXE. In SUMO, a selected real-world infrastructure, a section of the US Interstate highway was modeled as a roadway network based on its geometric information gained through the GIS platform. Injection of the traffic without connection to other vehicles was managed by SUMO. From the communication perspective, OMNeT++ simulates the communication actions among vehicles simultaneously. During the simulation, each connected vehicle received a representation in OMNeT++ as a communication unit. The communication protocols, message types, and lower-layer applications were preset in OMNeT++ based upon the IEEE 1609.4p family. PLEXE performed as the platooning management system. This framework managed platooning maneuvers and the platooned vehicle's behaviors.

Two case studies were conducted to explore the potential influences of CAV platoons on the traffic. The studies focused first on how different percentages of CAV platoon traffic affected the traffic based on existing infrastructure in a realistic manner. The other topic addressed how different platoon-oriented infrastructure designs and traffic patterns impacted traffic. A section of a US Interstate highway was selected as the infrastructure to establish the scenarios in. Before studying CAV platoon traffic's impacts, validation was conducted to examine if the simulation models could represent the real-world traffic. The validation was completed by comparing the traffic data (e.g., density, travel time, and volume) from the fieldwork and simulations. The settings of the simulation were based on calibration outputs and included vehicle specifications, vehicle control laws, and corresponding parameters.

The results of the case studies showed that with CAV platoon traffic, the overall performance of the traffic improved. When no changes were deployed to the traffic pattern or infrastructure design, improvements were found in measures such as the travel times under different levels of traffic density. However, traffic with low percentages of CAV platoon or with low overall traffic volume made the improvement of introducing CAV platoon traffic subtle. After deploying a new traffic pattern and infrastructure design (dedicated separate lane(s) for CAV platoon traffic), the simulation results showed that such patterns and designs have the potential to help with reducing the overall travel time. However, when the percentages of CAV platoon traffic were low, the regular traffic flow could deteriorate. Emission issues were measured in the second case study. The results showed that separating CAV platoon traffic from regular traffic could lessen the emissions on the regular traffic side when CAV traffic reaches a relatively high percentage. But overall, due to implementing higher speed limits on the CAV platoon's dedicated lanes, the emissions of CAVs increased significantly. Otherwise, the sensitivity to speed and spacing error required CAVs to adjust speed more intensively, which eventually led to higher overall emissions. The findings also suggest that dedicated platoon lanes become operationally beneficial once platoon penetration reaches moderate to high levels. At low penetration rates, lane conversion may reduce regular-lane performance due to underutilization. This provides a practical threshold guideline that transportation agencies may consider when planning CAV-platoon corridor deployments.

A conceptual simulation framework for exploring CAV platoons was the main objective of this study. The framework takes both vehicle-level models and traffic-level simulations into account. From a single-vehicle perspective, the technologies that leverage vehicle platooning were investigated. A component-based functional structure of the CAV system is presented in this study. The four components are perception, planning, control, and communication. Such a structure has been studied and widely implemented in the development of autonomous driving systems and vehicle communication protocol, and application designs. To test the theory, a CAV model was developed. The model integrated vehicle dynamic models, car longitudinal control laws, controller actuation delay, perception delay, lane-changing trajectory planning, and lateral control. Then, the model was utilized to calibrate the CAV's longitudinal control algorithms for the traffic-level simulation model.

Based on the reviewed literatures, conducting the traffic level simulation of CAV platoons has been found to be a multi-platform integration task. Such integration of traffic simulators and communication network simulators helps the users to focus on single or multiple objective topics instead of managing every piece of the simulation task. The most challenging part was to integrate those simulators properly. In this study, a SUMO-OMNeT++-based extension/framework was studied. Based upon this, and by modifying the framework and applications, an instance of the CAV platoon simulator has been developed.

By utilizing the developed simulator, two case studies were conducted to investigate two topics related to CAV platooning traffic. The first topic was how CAV platoon traffic could affect existing traffic patterns and infrastructure when considering traffic growth and truck traffic components. The second topic was whether implementing CAV platoon dedicated lanes could improve the traffic flow. The results of the first case study showed that when traffic increased as expected, the CAV platoon could slightly improve the average travel speed (also reflects the reduction of average travel time) when its percentage reached a certain level. Then, the next case study proved that dedicated lanes could improve the performance of traffic with some percentages of CAV traffic. Either adding lanes or converting lanes could benefit the overall traffic. However, deploying dedicated lanes with low percentages of CAV platoon traffic was not suggested since it could increase the burden of the regular lanes in the form of capacity reduction. There is no evidence to assert that separating CAV platoon traffic from the regular traffic with dedicated lanes could mitigate the CO_2 and NOx emissions when powertrain techniques do not vary between CAVs and human driven vehicles.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Schünemann, B. (2011) V2X Simulation Runtime Infrastructure Vsimrti: An Assessment Tool to Design Smart Traffic Management Systems. *Computer Networks*, **55**, 3189-3198. https://doi.org/10.1016/j.comnet.2011.05.005

[2] Santini, S., Salvi, A., Valente, A.S., Pescape, A., Segata, M. and Cigno, R.L. (2019) Platooning Maneuvers in Vehicular Networks: A Distributed and Consensus-Based Approach. *IEEE Transactions on Intelligent Vehicles*, **4**, 59-72. https://doi.org/10.1109/tiv.2018.2886677

[3] Zhao, L. and Sun, J. (2013) Simulation Framework for Vehicle Platooning and Car-Following Behaviors under Connected-Vehicle Environment. *Procedia—Social and Behavioral Sciences*, **96**, 914-924. https://doi.org/10.1016/j.sbspro.2013.08.105

[4] Jia, D., Lu, K. and Wang, J. (2014) A Disturbance-Adaptive Design for VANET-Enabled Vehicle Platoon. *IEEE Transactions on Vehicular Technology*, **63**, 527-539. https://doi.org/10.1109/tvt.2013.2280721

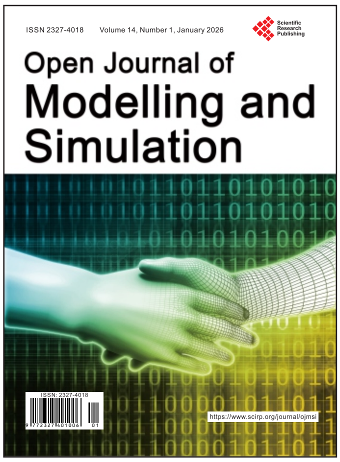
[5] Vieira, B., Severino, R., Koubaa, A. and Tovar, E. (2019) Towards a Realistic Simulation Framework for Vehicular Platooning Applications. 2019 *IEEE* 22*nd International Symposium on Real-Time Distributed Computing* (*ISORC*), Valencia, 7-9 May 2019, 93-94. https://doi.org/10.1109/isorc.2019.00028

[6] Hegde, A. and Festag, A. (2020). Artery-C—An OMNeT++ Based Discrete Event Simulation Framework for Cellular V2X. *Proceedings of the* 23*rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Alicante, 16-20 November 2020, 47-51. https://doi.org/10.1145/3416010.3423240

[7] Krasniqi, X. and Hajrizi, E. (2016) Use of IoT Technology to Drive the Automotive Industry from Connected to Full Autonomous Vehicles. *IFAC-PapersOnLine*, **49**, 269-274. https://doi.org/10.1016/j.ifacol.2016.11.078

[8] Liu, C., Chau, K.T., Wu, D. and Gao, S. (2013) Opportunities and Challenges of Vehicle-to-Home, Vehicle-to-Vehicle, and Vehicle-to-Grid Technologies. *Proceedings of the IEEE*, **101**, 2409-2427. https://doi.org/10.1109/jproc.2013.2271951

[9] Talebpour, A. and Mahmassani, H.S. (2016) Influence of Connected and Autonomous Vehicles on Traffic Flow Stability and Throughput. *Transportation Research Part C*: *Emerging Technologies*, **71**, 143-163. https://doi.org/10.1016/j.trc.2016.07.007



Call for Papers

Open Journal of Modelling and Simulation

ISSN: 2327-4018 (Print) ISSN: 2327-4026 (Online)
https://www.scirp.org/journal/ojmsi

Open Journal of Modelling and Simulation (OJMSi) is an open access journal published quarterly. The goal of this journal is to provide a platform for scientists and academicians all over the world to promote, share, and discuss various new issues and developments in all aspects of modelling and simulation.

Editorial Board

Subject Coverage

All manuscripts must be prepared in English, and are subject to a rigorous peer-review process. Accepted papers will immediately appear online followed by printed hard copy. The journal publishes original papers including but not limited to the following fields:

- Aerospace Technology Simulation
- Aircraft Design Simulations
- Applications of Social and Economic Models
- Chemical Process Simulation
- Computational Intelligence
- Computer Science
- Constitutive Modelling of Materials
- Control Theory
- Dynamic Response of Mechanical Systems
- Dynamical Systems
- Finite Element/Computational Fluid Dynamics Simulation
- Many Agents Complex Systems
- Mathematical Biology
- Military Technology Simulation
- Nonlinear Processes
- Numerical Analysis
- Numerical Modelling
- Rigid/Flexible Body Mechanisms Simulations
- Seismic Simulation of Civil Structures
- Simulation Modeling Theories and Methodologies
- Visual Simulation

We are also interested in: 1) Short reports—2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data; 2) Short communications—2-5 page papers; 3) Technical notes—2-5 page papers; 4) Letters to the editor (the number of pages is not restricted); 5) Reviews or book reviews—comments and critiques (the number of pages is not restricted); 6) Advertisement—1-2 page papers; 7) News letters—1-5 page papers.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

https://www.scirp.org/journal/ojmsi E-mail: ojmsi@scirp.org