

Modular Traffic Sign Detection via Environment-Aware Routing for Adverse Conditions

Xizhi Wang

Abstract

Accurate traffic sign detection is essential for the safety of autonomous vehicles. However, existing object detection models often underperform in adverse environmental conditions due to their reliance on monolithic architectures. In this paper, we propose a modular detection system that incorporates lightweight environmental classification to route input images to specialized object detectors. By separating detection into environment-specific pipelines (e.g., day and night), our architecture enhances accuracy without requiring hardware expansion or retraining of the entire system. We evaluate our system using the Chinese Traffic Sign Detection Benchmark dataset, training a YOLOv8m-cls classifier for environment classification and multiple YOLOv8 models for domain-specific detection. Our results show that the modular system with classifier-based routing outperforms both monolithic models and manual routing baselines under comparable constraints. These findings suggest that modular, environment-aware pipelines offer a scalable, interpretable, and highly effective solution for enhancing traffic sign detection in real-world autonomous systems.

1 Introduction

Traffic sign detection is a crucial task for autonomous vehicles, as it enables the system to interpret road regulations and warnings, thereby ensuring safety (Chaghazardi et al., 2023; Lin et al., 2025). Failure to accurately detect and classify these signs could lead to serious safety hazards such as speeding, failure to yield, and illegal turns. Modern computer vision approaches address this problem by utilizing deep learning-based object detection models to identify the type and location of signs within the vehicle’s field of view. Benefiting from large-scale annotated datasets and high-capacity neural networks, these detectors have achieved strong performance under favorable conditions (Li et al., 2025).

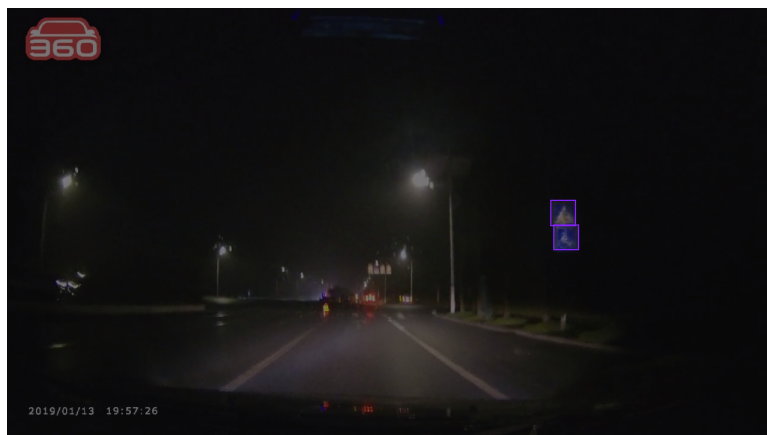


Figure 1: Sample image from the Chinese Traffic Sign Detection Benchmark dataset, showing ground-truth bounding box (Zhang et al., 2022)

A fundamental limitation of current traffic sign detectors is their monolithic design, which overlooks environmental context. Visual conditions can vary dramatically in real-world scenarios. Figure 1 illustrates a nighttime example from the Chinese Traffic Sign Detection Benchmark (CCTSDb). The traffic signs appear dimly lit, distant, and blurred due to low illumination and glare. In such adverse scenes, even state-of-the-art detectors can miss detections or provide incorrect classifications (Li et al., 2025; N. Aldoski & Koren, 2025).

Although some detection systems attempt to improve generalization by training on large datasets with diverse lighting and weather conditions, this approach often introduces domain inconsistencies. For example, the visual patterns in clear daylight and nighttime scenes can differ substantially, confusing generalized models and degrading accuracy in both conditions.

Traffic sign detection under varied real-world conditions demands more than a monolithic model. To address this, we introduce a modular, environment-aware architecture that incorporates a lightweight CNN scene classifier to route each input image into a specialized detection pipeline. This design enables each detector to focus on the visual pattern of its domain, thereby improving accuracy without requiring retraining or hardware expansion.

2 Background

In this section, we will briefly introduce some foundational concepts relevant to traffic sign detection, particularly in the context of deep learning-based object detection under varying environmental conditions.

2.1 CNN Image Classification

In computer vision, image classification refers to the task of assigning one or many class labels based on the image’s prominent contents. Within the context of our modular approach in autonomous vehicles, the most relevant is a single-label classifier based on a Convolutional Neural Network (CNN). In our setup, we utilize YOLOv8m-cls, a standalone classification variant of YOLOv8 for fast and lightweight scene classification. (Ultralytics, 2023).

CNNs are especially effective for image classification as they are designed to recognize hierarchical visual features in an image. A CNN typically consists of four fundamental types of layers:

- **Convolutional Layer:** Applies a set of trainable filters (kernels) across the image to produce feature maps. Each kernel performs a sliding dot-product operation over small regions of the image to capture local features such as edges and patterns.
- **ReLU Layer:** Introduces non-linearity using an activation function after convolution, helping the network learn complex patterns and reduce overfitting.
- **Pooling Layer:** Downsamples the feature maps (commonly using max pooling) to reduce spatial dimensions, lowering computational cost and increasing robustness.
- **Fully Connected Layer:** Flattens the feature maps into a vector and passes it through a softmax function to output a probability distribution for classification.

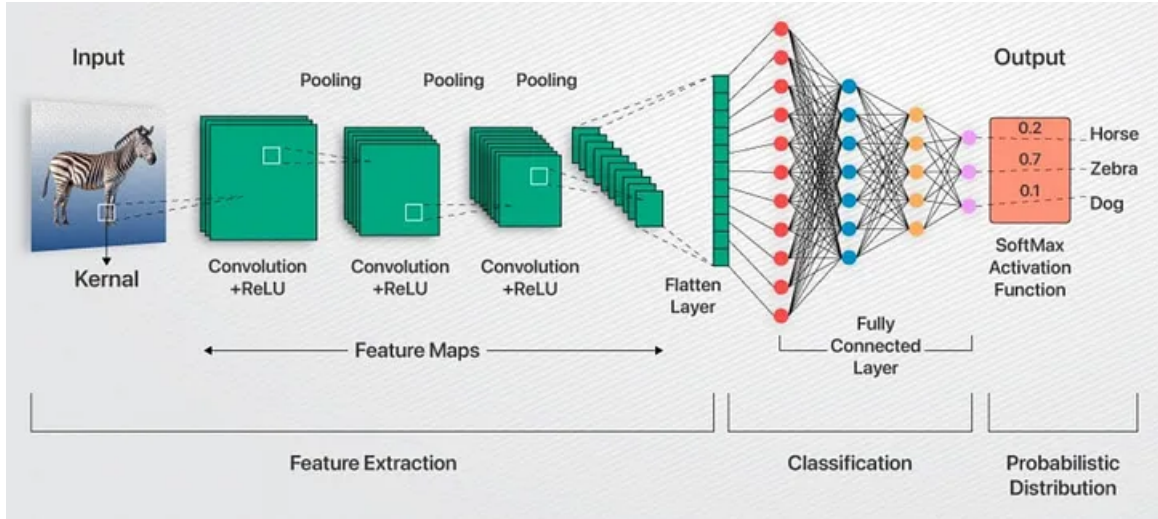


Figure 2: Architecture of a Convolutional Neural Network for Image Classification (Singh, 2023)

2.2 Object Detection

While image classification predicts labels for a given image, object detection expands on this task by localizing objects in the image with bounding boxes. In the context of traffic sign detection, this means not only determining the presence of signs but also drawing bounding boxes around them and classifying their types.

Modern deep learning-based object detection models typically fall into two categories. Two-stage detectors first generate region proposals that potentially contain objects using a Region Proposal Network (RPN), and then refine them in a second stage. These models, such as Faster R-CNN (Ren et al., 2015), tend to achieve higher accuracy, especially in complex scenarios, but require more computational resources. On the other hand, one-stage detectors directly predict object classification and localization across a dense grid over the input image in a single forward pass. One-stage detectors significantly increase computational speed, but often at the cost of reduced detection accuracy.

2.3 YOLO Architecture

YOLOv8 (You Only Look Once, version 8) is a high-performance one-stage detector developed by Ultralytics. Its balance of speed and performance makes it suitable for real-time applications such as autonomous driving. Although it is not the most recent model in the YOLO family, YOLOv8 remains widely used due to its mature software ecosystem, comprehensive documentation, and broad library support (Ultralytics, 2023).

YOLOv8's architecture is divided into three main components. First, the backbone is responsible for extracting feature maps from the input image. YOLOv8 incorporates a custom CSPDarknet CNN architecture, which employs Cross Stage Partial (CSP) connections and C2f modules to improve information flow. The architecture aims to learn both low-level and high-level visual patterns while maintaining a low computational cost.

Then, the neck combines and refines features extracted from different stages of the backbone. YOLOv8 utilizes an enhanced version of the Path Aggregation Network (PANet) and incorporates fusion modules, enabling the model to detect objects at various scales. In traffic sign detection, this enables YOLOv8 models to more accurately detect signs of various sizes, taking into account camera distance and angle.

Lastly, the detection head makes the final predictions. YOLOv8 employs a one-stage, anchor-free design, directly predicting bounding boxes, class probabilities, and object likelihoods from

the feature maps. The direct regression approach enables YOLOv8 models to detect objects more efficiently, particularly in diverse environments and with varying object sizes.

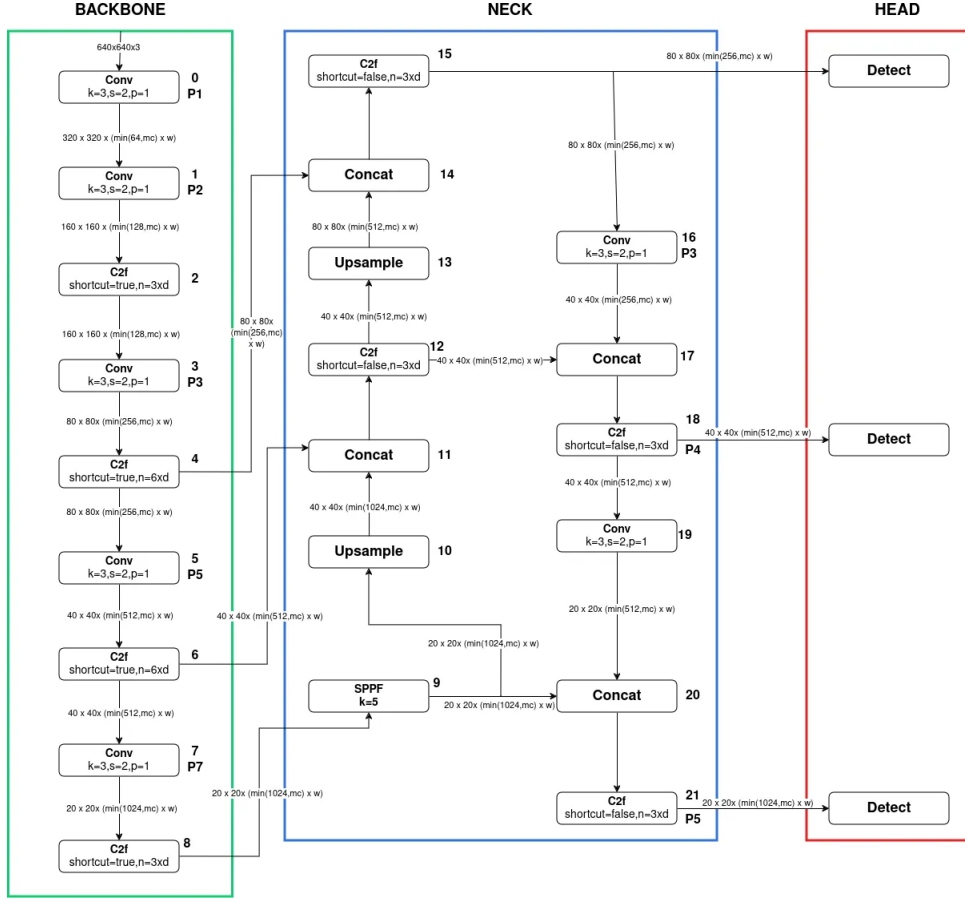


Figure 3: Ultralytics YOLOv8 Model Architecture (Timilsina, 2024)

2.4 Performance Metrics

To evaluate the effectiveness of object detection models, we use several quantitative metrics that assess both classification accuracy (the ability to identify object classes correctly) and localization accuracy (the ability to predict bounding boxes correctly).

Intersection over Union (IoU): IoU measures how well a predicted bounding box matches the ground truth. It is defined as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

- **Area of Overlap:** The area where the predicted and ground-truth boxes intersect.
- **Area of Union:** The total area covered by both the predicted and ground-truth boxes.

A prediction is considered correct (true positive) if the IoU is greater than the set threshold, and incorrect otherwise (false positive).

Precision and Recall: These metrics quantify the accuracy of the model's predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Precision** indicates the proportion of correct predictions among all predicted objects.
- **Recall** indicates the proportion of correctly identified objects among all actual objects.

Average Precision (AP) measures how well a model balances precision and recall for a single class. In object detection, each prediction is accompanied by a confidence score that reflects the model’s certainty in classifying the object. When the model uses a high confidence threshold, it only retains predictions that it is confident about. These predictions are more likely to have a high IoU, resulting in high precision but lower recall, as many actual objects may be missed. When the threshold is lowered, more objects are predicted (higher recall), resulting in more false positives (lower precision).

These trade-offs are visualized through a precision-recall curve, and AP is computed as the area under this curve. Mean Average Precision (mAP) is the average AP across all classes.

- **mAP@0.5**: AP evaluated at IoU threshold of 0.5.
- **mAP@0.5:0.95**: Average AP from IoU threshold of 0.5 to 0.95 (increments of 0.05).

3 Related Works

While deep learning object detection models achieve impressive results on controlled datasets, their performance degrades significantly in real-world scenarios. This is mainly because standard detectors primarily rely on appearance-level features such as brightness and texture, which vary drastically between different environments. As Du et al. (2024) observe, illumination variation introduces domain shifts that cause models trained under well-lit conditions to perform poorly when visibility drops.

Prior research addressing these challenges can be categorized into three main areas: image enhancement-based preprocessing, architecture-level improvements, and modular or adaptive approaches.

3.1 Image Enhancement-Based Preprocessing

Image enhancement is a widely used strategy for mitigating adverse visual conditions by transforming inputs to resemble those captured in ideal, clear-day scenarios. These methods aim to improve object detector performance by increasing visibility and contrast in noisy environments.

A standard limitation of many enhancement-based approaches is the separation between the preprocessing and detection pipelines. For instance, Zhang et al. (2023) propose a pix2pixHD GAN to defog images prior to detection. While this improves accuracy under foggy conditions, the preprocessing introduces latency and may cause visual distortions that disrupt features critical for the detection pipeline. Similarly, Wang et al. (2024) introduce LLE-STD, a two-stage framework that combines residual-based enhancement with a specialized detector utilizing attention mechanisms and anchor refinement. Despite increased performance in low-light settings, the architecture still treats enhancement and detection as independent modules.

Recent work moves toward more consolidated architectures. Cui et al. (2023) and Hong et al. (2024) incorporate illumination learning directly into their detection architectures, respectively, in Trash-to-Treasure (T2) and You Only Look Around (YOLA). These approaches treat lighting as meaningful contextual information rather than noise, and jointly optimize decomposition and detection within a unified framework. This reduces the detachment between enhancement and detection, but it still exists as a monolithic framework.

3.2 Architecture and Detector-Level Improvements

Rather than relying on preprocessing, some studies improve robustness by directly modifying the detection architecture to avoid distortions and maintain computational efficiency.

Niu et al. (2024) propose YOLOv8s-DDA, which integrates a Dual-Weighted Residual Dense Block (DWR-DRB) and an enhanced feature fusion neck within the detector. This design improves small object detection while maintaining computational efficiency.

Similarly, Lin et al. (2025) introduce YOLO-LLTS, an end-to-end detection framework specifically optimized for low-light environments. Rather than applying enhancement externally, YOLO-LLTS incorporates lightweight attention modules and internal feature adjustment mechanisms directly into the detection network. This design enables the model to dynamically adapt feature extraction based on input lighting conditions, thereby improving detection reliability without introducing preprocessing.

These architectural improvements illustrate that architectural design mechanisms can effectively adapt to environmental variability. However, these models are trained as generalized models, lacking the specialization required in domain-specific conditions.

3.3 Modular and Multi-Model Approaches

While architectural improvements achieve robustness in a single detection model, some research has already explored modular and adaptive approaches that tailor detection to specific environmental conditions.

Wang et al. (2024) propose a stacking ensemble of YOLOv8 and Mask R-CNN, where each model performs better in specific scenarios (e.g., distinguishing between distant and nearby objects). While effective, the approach lacks environmental context-awareness for model switching, resulting in increased computational cost and reduced specialization.

Liu et al. (2021) introduce a more adaptive approach in Image-Adaptive YOLO (IA-YOLO), which integrates enhancement and detection within a modular framework. The system includes a lightweight CNN that predicts optimal image enhancement parameters for each input, which are then applied through a differentiable image processing (DIP) module before detection. While this pipeline is adaptive to visual input, it still treats enhancement and detection as components of a single monolithic model. It adapts to image-level features such as brightness and fog density but does not account for explicit scene-level context.

To make detection pipelines truly modular and environment-aware, it is essential to introduce a lightweight yet accurate routing mechanism that supports this goal.

3.4 Environmental Classification in Autonomous Vehicles

Recent research highlights the growing importance of real-time environmental classification in autonomous vehicles for safe and adaptive driving. For instance, Introvigne et al. (2024) propose a lightweight convolutional classifier that achieves over 92% accuracy across six environmental categories using traffic image input. It is capable of running at over 20Hz, making it suitable for real-time systems. Such environment classification is applied to adjust camera exposure parameters (Lee et al., 2024) and optimize routing decisions under dynamic road conditions (Julio-Rodríguez et al., 2022). These studies demonstrate that environmental awareness is increasingly utilized to enhance downstream tasks in autonomous vehicles, such as traffic light detection, sensor tuning, and path planning. However, in the context of traffic sign detection, such classifiers remain underutilized.

Our proposed architecture builds upon this line of research by integrating real-time environmental classification directly into the detection pipeline for traffic signs, allowing input images to be routed to environment-specific models and enhancing accuracy under adverse conditions.

4 Proposed Architecture

To address the limitations of conventional object detection models under diverse environmental conditions, we propose a modular, context-aware detection system that performs an environment classification pipeline followed by specialized detection.

4.1 Environment-Aware Routing via Scene Classification

The routing stage of our architecture employs a CNN-based classifier trained on labeled datasets of traffic scenes under different conditions. This component introduces minimal computational overhead but enables powerful modularity. Each detection can be explicitly optimized for the type of image it receives, and routing decisions are interpretable. Notably, scene classifiers can be viewed not only as a prerequisite for routing but also as an indicator of domain overlap. Images that are misclassified (e.g., noisy dawn scenes labeled as night) often share visual characteristics with both domains, allowing for them to still be reasonably handled by the specialized detector.

4.2 Scalable Modular Design and Selective Inference Efficiency

A key advantage of the proposed architecture is its modularity, which enables seamless scalability without requiring the retraining of existing components or an increase in computational cost. New classifiers and domain-specific detectors can be integrated into the pipeline with minimal disruption. Each added module operates independently and is only invoked when it is relevant; therefore, modular expansion does not increase computational cost at time of inference while preserving existing performance. The design also allows components to be swapped, extended, or fine-tuned in isolation. For instance, if a new type of visual noise becomes prevalent in a specific region or season, a corresponding detector can be optimized without impacting other detectors.

Additionally, the routing mechanism can evolve into a multi-stage system, where an initial classifier identifies broad environmental categories, and subsequent classifiers refine this classification into more specific subtypes (e.g., thick fog vs. light fog) (see Figure 4). The model can also adapt smoothly to additional hardware expansions. For example, additional detectors could be loaded after the classification stage and can also incorporate additional inputs (e.g., thermal imaging, radar) into the classification mechanism to enhance accuracy.

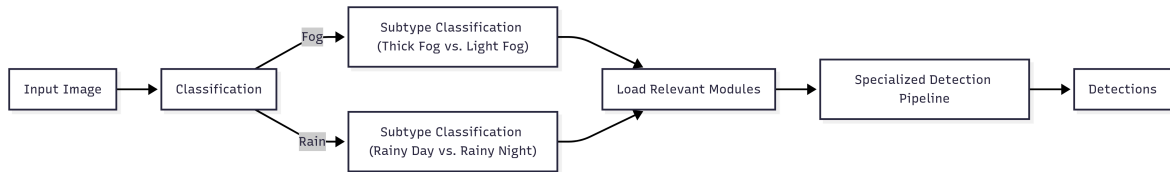


Figure 4: Example of a Multi-Stage Environmental Routing Pipeline for Specialized Traffic Sign Detection

Many advanced architectural design mechanisms and preprocessing strategies are discussed in Section 3. Related Works can be easily incorporated within this modular framework. For example, the illumination-aware feature adjustment modules introduced in YOLO-LLTS (Lin et al., 2025) could be integrated into the nighttime detection pipeline to improve performance under low-light conditions, without modifying the detectors used for other scenes. This illustrates how specialized techniques can be selectively applied to relevant domains within the modular system.

5 Evaluation

To assess the effectiveness of our proposed architecture, we experimented with comparing different detection strategies under varying routing methods.

5.1 Datasets Used

For our experiment, we used two datasets: CCTSDB for object detection and BDD100K for scene classification.

The Chinese Traffic Sign Detection Benchmark (CCTSDB) introduced by Zhang et al. (2022) consists of over 16,000 images with three broad classes (mandatory, prohibitory, and warning). CCTSDB was created to evaluate traffic sign detection models in increasingly challenging conditions; images were intentionally replaced with those from more difficult environments, such as those with environmental noise, blur, and shadows. Additionally, CCTSDB includes a subset with traffic signs categorized by specific weather and lighting conditions, making it well-suited for experimentation under variable environments.

The Berkeley DeepDrive 100K (BDD100K), released by Yu et al. (2020), contains 100,000 images across diverse geographic regions and weather conditions. Each image includes annotations for attributes such as weather, time of day, and road type.

5.2 Justification of Experimental Design

Our experimental design reflects a balance between methodological rigor and the practical limitations of available datasets. Key considerations include:

- **Limited Availability of Large Environment-Labeled Detection Datasets:** No traffic sign detection benchmarks provide large volumes of images across labeled environmental conditions. As a result, we structured our experiment around CCTSDB’s night and day subsets, which were the largest subsets available (500 nighttime images).
- **Binary Environmental Classification:** We restricted scene classification to two broad categories (day and night) based on their clear visual distinction and data availability. While real-world conditions are more diverse, expanding to more categories would require larger datasets with environment annotations, which are currently lacking.
- **Equal Data for All Models:** To ensure fair comparison, each detection model was trained on a dataset of equal size (500 images). Additionally, we trained a monolithic model on a 1,000-image set to examine the effect of scaling and establish an upper-bound reference.

5.3 Experimental Procedure

We first trained a lightweight YOLOv8m-cls scene classifier based on a balanced subset of the BDD100K dataset, consisting of 14,218 day images and 14,218 night images. We then trained several YOLOv8s detection models using subsets of the CCTSDB dataset, with configurations designed to evaluate both monolithic and modular pipelines:

- **Monolithic Model (1,000 images):** A YOLOv8s object detector trained on a combined dataset of 500 day and 500 night images (including validation and test sets). This model serves as a high-capacity, general-purpose baseline.
- **Monolithic Model (500 images):** A YOLOv8s model trained on 250 day and 250 night images, randomly sampled from the same image pool. This setup provides a baseline under the same data budget as the modular configuration.

- **Modular Model (Classifier Routing):** This architecture uses the YOLOv8m-cls scene classifier to classify test images as either day or night and then routes each image to:
 - **Day Model:** Trained on 500 daytime images (same as the monolithic dataset).
 - **Night Model:** Trained on 500 nighttime images.
- **Modular Model (Manual Routing):** Identical to the classifier-based modular setup in terms of detection models, but uses ground-truth environmental labels for routing. This configuration assumes perfect classification and serves as an upper-bound reference for modular routing.

All models were trained using consistent training settings to ensure comparability.

5.4 Evaluation Results

We evaluated all trained models on a shared test set of 75 day and 75 night images from CCTSDB which were not used during the training of any models.

Table 1: Detection performance under different model configurations. Classifier routing has classification accuracy of 95.33%.

Configuration	mAP@0.5	mAP@0.5:0.95	Precision	Recall
Monolithic (1,000 images)	0.9010	0.7845	0.8376	0.5134
Monolithic (500 images)	0.8077	0.7201	0.7731	0.4529
Modular (Manual Routing)	0.9077	0.7596	0.8467	0.5357
Modular (Classifier Routing)	0.9084	0.7606	0.8477	0.5365

5.5 Discussion

The results of Table 1 suggest several noteworthy observations regarding our modular detection strategy.

The Modular (Classifier Routing) configuration achieved the highest performance across all key metrics, including mAP@0.5 (0.9084) and mAP@0.5:0.95 (0.7606). This routing strategy slightly outperformed the Monolithic Model (1,000) and significantly outperformed the Monolithic Model (500) in both localization accuracy and precision. This indicates that even coarse environmental separation (binary day/night) enables specialized detectors to better adapt to their respective domains, especially in conditions with apparent visual differences.

The scene classifier achieved a 95.33% accuracy on the test set, meaning that a small fraction of images were misrouted to the incorrect detector. Nonetheless, the Modular (Classifier Routing) configuration slightly outperformed the Modular (Manual Routing) setup across all metrics. This outcome reinforces our hypothesis that in our modular detection framework, routing errors do not necessarily degrade performance and may even enhance it. The classifier is routing images based on visual similarity, and misrouted images may resemble the target domain more closely, resulting in improved detection performance.

The Monolithic Model (1,000) benefited from more training data, but both modular setups still outperformed it. This highlights the limitations of scaling training data and emphasizes the effectiveness of modularity. It also reinforces that specialized detectors trained on focused domains outperform general-purpose detectors.

Both modular setups reported higher precision (> 0.907) than either Monolithic model, indicating fewer false positives. This is especially desirable in real-world autonomous driving applications, where over-detection can lead to serious safety risks.

5.6 Limitations and Future Work

While our results demonstrate the potential of the modular detection system, several limitations remain to be addressed in future studies.

5.6.1 Limited Environmental Classes

In our experiments, we used a simple binary scene classifier (day vs. night) and evaluated it on a small curated subset of the CCTSDB dataset. Such an approach does not encompass the full range of real-world environmental classes or the diverse visual noises that occur in the real world. To overcome these limitations, future work should collect larger object-detection datasets that collect large samples from a broad range of environments. With such data, our routing framework could be extended to a multi-class or hierarchical structure (e.g., rainy nights), enabling each specialized detector to extract patterns from its respective domain more effectively. However, adding environmental categories will reduce training data for each model and increase the complexity of classification. Future work should investigate these tradeoffs and propose a balanced solution that can still effectively mitigate domain shifts.

5.6.2 Real-Time Deployment

Minimizing computational cost is crucial for implementing our system in a real-time autonomous system. Although our modular design incurs only a single lightweight classification step followed by a specialized detector, we have yet to quantify the hardware cost of the system. Future work should focus on optimizing each component, investigating model compression techniques, and exploring dynamic loading strategies to minimize resource usage when environmental conditions are easily interpretable. Although not the intention of our pipeline, future work could explore incorporating simple hardware sensors (e.g., luminance sensors) to assist with classifying the environment and detecting signs.

5.6.3 Ensemble-Based Evaluation

While we observed that misrouting errors do not always degrade detection performance, this insight remains theoretical. Future work can still implement an ensemble-based evaluation framework in which multiple specialized detection models operate in parallel (see Figure 5). We can then directly

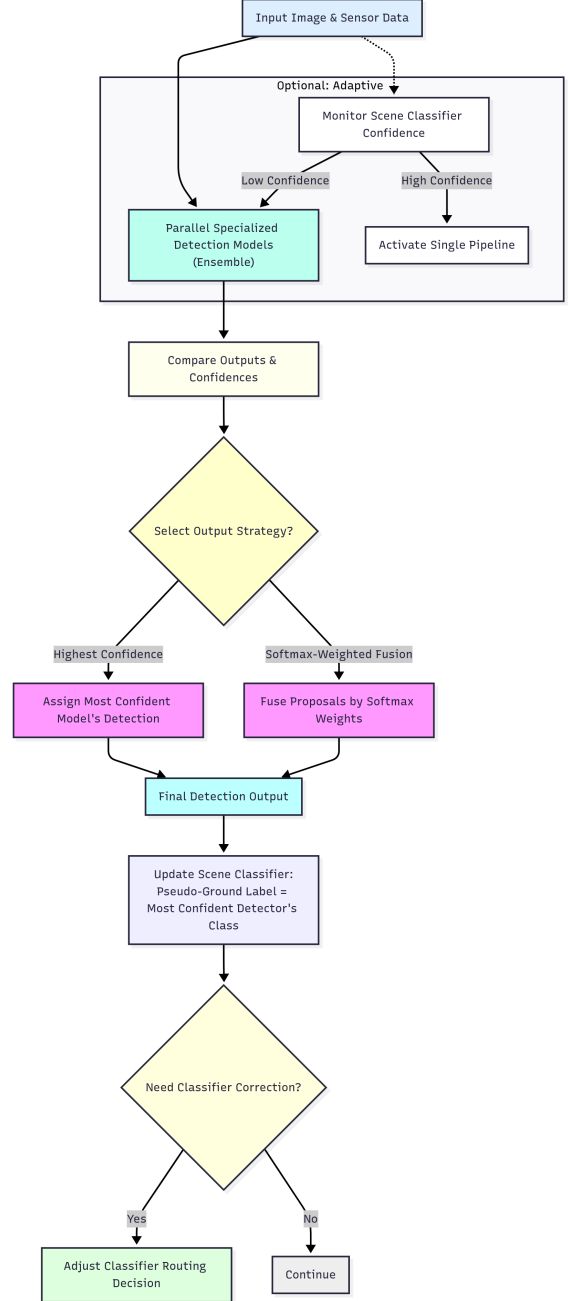


Figure 5: Feedback Loop Architecture for Ensemble-Based Evaluation in the Environment-Aware Routing Pipeline

compare their outputs and confidence to quantify the impact of routing decisions under various environmental conditions.

Rather than relying on a single scene classifier to choose a pipeline, the system would select the most reliable detections by comparing confidence measures across all candidate models. We would select the result of the detector that outputs the highest confidence, or combine proposals through softmax-weighted fusion.

A key benefit of ensemble evaluation is the creation of a self-learning feedback loop for the scene classifier. We can treat the environment class of the most confident detector as the pseudo-ground label, and correct the routing decision of the classifier if necessary.

However, running multiple detectors increases computational cost, particularly on a real-time system. Therefore, future experiments should benchmark the latency of a whole ensemble system and compare it against selective activation strategies. Alternatively, we could trigger the ensemble system only in more complex scenarios where the classifier is less confident.

6 Conclusion

In this paper, we addressed the problem of degraded traffic sign detection performance in monolithic models under adverse environmental conditions due to domain shifts. We proposed a modular, environment-aware traffic sign detection system that utilizes a lightweight CNN scene classifier to route input images to specialized YOLOv8 detectors for both daytime and nighttime conditions. Through evaluation on the Chinese Traffic Sign Detection Benchmark, our classifier-based routing configuration outperformed other monolithic models and manual routing setups. These results demonstrate that environment-based routing enhances both traffic sign categorization accuracy and localization precision without incurring significant inference or data costs. The modular design also allows for the seamless integration of new environmental categories without requiring retraining of existing detectors, offering a scalable solution for real-world autonomous vehicles. Future work will focus on extending the routing framework to multi-class environmental categories, optimizing real-time deployment, and exploring ensemble-based evaluation to enhance detection robustness further.

References

- Alsiyeu, U., & Duisebekov, Z. (2024). *Enhancing Traffic Sign Recognition with Tailored Data Augmentation: Addressing Class Imbalance and Instance Scarcity*. doi: 10.48550/arXiv.2406.03576
- Chaghazardi, Z., Fallah, S., & Tamaddoni-Nezhad, A. (2023, September). Explainable and Trustworthy Traffic Sign Detection for Safe Autonomous Driving: An Inductive Logic Programming Approach. *Electronic Proceedings in Theoretical Computer Science*, 385, 201–212. Retrieved from <http://dx.doi.org/10.4204/EPTCS.385.21> (Publisher: Open Publishing Association) doi: 10.4204/eptcs.385.21
- Cui, X., Ma, L., Ma, T., Liu, J., Fan, X., & Liu, R. (2023). *Trash to Treasure: Low-Light Object Detection via Decomposition-and-Aggregation*. Retrieved from <https://arxiv.org/abs/2309.03548> (_eprint: 2309.03548)
- Du, Z., Shi, M., & Deng, J. (2024). *Boosting Object Detection with Zero-Shot Day-Night Domain Adaptation*. Retrieved from <https://arxiv.org/abs/2312.01220> (_eprint: 2312.01220)
- He, Y., Guo, M., Zhang, Y., Xia, J., Geng, X., Zou, T., & Ding, R. (2024). *NTS-YOLO: A Nocturnal Traffic Sign Detection Method Based on Improved YOLOv5*. doi: 10.21203/rs.3.rs-4421906/v1

- Hong, M., Cheng, S., Huang, H., Fan, H., & Liu, S. (2024). *You Only Look Around: Learning Illumination Invariant Feature for Low-light Object Detection*. Retrieved from <https://arxiv.org/abs/2410.18398> (_eprint: 2410.18398)
- Introvigne, M., Ramazzina, A., Walz, S., Scheuble, D., & Bijelic, M. (2024). *Real-Time Environment Condition Classification for Autonomous Vehicles*. Retrieved from <https://arxiv.org/abs/2405.19305> (_eprint: 2405.19305)
- Julio-Rodríguez, J., Rojas-Ruiz, C., Santana, A., Bustamante-Bello, R., & Ramirez-Mendoza, R. (2022, May). Environment Classification Using Machine Learning Methods for Eco-Driving Strategies in Intelligent Vehicles. *Applied Sciences*, 12, 5578. doi: 10.3390/app12115578
- Karbouj, B., Topalian Rivas, G. A., & Krüger, J. (2024, May). Comparative Performance Evaluation of One-Stage and Two-Stage Object Detectors for Screw Head Detection and Classification in Disassembly Processes. In *Procedia CIRP* (Vol. 122). doi: 10.1016/j.procir.2024.01.077
- Lee, K., Shin, U., & Lee, B.-U. (2024). *Learning to Control Camera Exposure via Reinforcement Learning*. Retrieved from <https://arxiv.org/abs/2404.01636> (_eprint: 2404.01636)
- Li, J., Deng, Q., Gao, W., Yang, B., Jia, L., Zhou, J., & Pu, H. (2025, July). DSF-YOLO for robust multiscale traffic sign detection under adverse weather conditions. *Scientific Reports*, 15. doi: 10.1038/s41598-025-02877-0
- Lin, Z., Wu, Y., Ma, Y., Chen, J., Zhang, R., Wu, J., . . . Lin, L. (2025). *YOLO-LLTS: Real-Time Low-Light Traffic Sign Detection via Prior-Guided Enhancement and Multi-Branch Feature Interaction*. Retrieved from <https://arxiv.org/abs/2503.13883> (_eprint: 2503.13883)
- Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., & Zhang, L. (2022). *Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions*. Retrieved from <https://arxiv.org/abs/2112.08088> (_eprint: 2112.08088)
- N. Aldoski, Z., & Koren, C. (2025, June). Evaluating the impact of simulated weather conditions on traffic sign detection for autonomous vehicles using YOLOv8..
- Niu, M., Chen, Y., Li, J., Qiu, X., & Cai, W. (2024, September). YOLOv8s-DDA: An Improved Small Traffic Sign Detection Algorithm Based on YOLOv8s. *Electronics*, 13, 3764. doi: 10.3390/electronics13183764
- Ren, S., He, K., Girshick, R., & Sun, J. (2015, June). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39. doi: 10.1109/TPAMI.2016.2577031
- Singh, A. (2023, February). *Deep Dive into the World of CNNs*. Retrieved from <https://ai.plainenglish.io/deep-dive-into-the-world-of-cnns-8cf22cd84e7>
- Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32, 323–332. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0893608012000457> doi: <https://doi.org/10.1016/j.neunet.2012.02.016>
- Tabernik, D., & Skočaj, D. (2020). Deep Learning for Large-Scale Traffic-Sign Detection and Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1427–1440. doi: 10.1109/TITS.2019.2913588
- Timilsina, A. (2024, March). *YOLOv8 Architecture Explained!* Retrieved from <https://abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5>
- Ultralytics. (2023, November). *Explore Ultralytics YOLOv8* (Tech. Rep.). Retrieved from <https://docs.ultralytics.com/models/yolov8>
- Wang, T., Qu, H., Liu, C., Zheng, T., & Lyu, Z. (2024, October). LLE-STD: Traffic Sign Detection Method Based on Low-Light Image Enhancement and Small Target Detection. *Mathematics*, 12, 3125. doi: 10.3390/math12193125
- Wang, Y., & Wang, Q. (2024). Robust Stacking Ensemble Model for Traffic Sign Detection and Recognition. *IEEE Access*, 12, 178941–178950. doi: 10.1109/ACCESS.2024.3504827
- Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., . . . Darrell, T. (2020). *BDD100K: A*

- Diverse Driving Dataset for Heterogeneous Multitask Learning*. Retrieved from <https://arxiv.org/abs/1805.04687> (_eprint: 1805.04687)
- Zhang, J. (2023). Traffic sign defogging based on conditional adversarial neural network pix2pixHD. In *2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE)* (pp. 61–65). doi: 10.1109/NNICE58320.2023.10105664
- Zhang, J., Zou, X., Kuang, L., Wang, J., Sherratt, R., & Yu, X. (2022, May). CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark. *Human-centric Computing and Information Sciences*, 12, 23. doi: 10.22967/HCIS.2022.12.023
- Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-Sign Detection and Classification in the Wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.