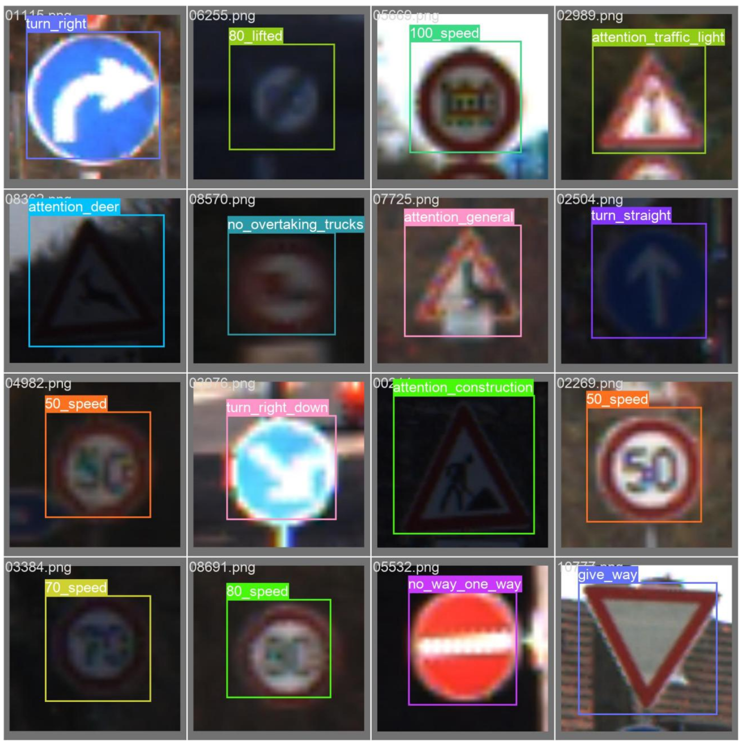


IMPROVING ROBUSTNESS OF TRAFFIC SIGN DETECTION FOR AUTOOMOUS VEHICLES IN ADVERSE WEATHER

COMPUTER VISION

Asya D.

Abstract - This paper deals with the problem of robust traffic sign detection in the presence of adverse weather with a YOLOv5-based pipeline. The GTSRB dataset was preprocessed, normalized, then augmented with artificially weather conditions (fog, rain, lighting) so as to reflect real life conditions. Training utilized a combined loss (bounding-box, objectness, classification) and standard detection metrics (mAP, precision, recall) were used for testing. Results High sensitivity was obtained and very high accuracy in terms of detection was achieved, with class imbalance decreasing performance in several classes in this task. Critical analysis identifies dataset bias, overfitting checks and risks for ethical deployment. Results suggest that the augmentation-improved YOLOv5 models can be an effective solution for robust perception of autonomous vehicles in harsh environments.

I. INTRODUCTION

Autonomous vehicles rely heavily on accurate traffic sign recognition and interpretation to maintain safe and conformant driving. However, real-world unpredictable weather conditions—such as rain and fog, for example—can cause visual disruptions that significantly detract from the performance of traditional computer vision models, thus presenting safety issues when applied in real-world driving situations. [Mirza et al.] Deep learning-based computer vision has demonstrated exceptional efficacy for object detection in autonomous driving applications. The YOLOv5 architecture, employing convolutional neural networks, excels at autonomously extracting sophisticated visual features, enabling consistent identification of traffic signs and critical roadway elements. Notably, these models achieve the processing speeds necessary for real-time operation—a crucial requirement where both detection accuracy and low latency are paramount.

The project was initially designed to compare a baseline YOLOv5 model trained on clean images with an enhanced version designed for adverse weather robustness. However, due to time and resource constraints, only the enhanced model was implemented in practice. Building on this foundation, robustness was pursued through two main strategies: first, the application of synthetic adverse weather augmentations to simulate challenging real-world conditions, and second, the integration of architectural improvements, namely Convolutional Block Attention Modules (CBAM) and EfficientNet-B4 backbones. Autonomous vehicles require more robust detection capabilities in challenging conditions - a crucial requirement for reducing sensor failure incidents and improving overall system reliability as demonstrated in recent studies (Zhang et al. 2023; Gurbindo et al. 2025). The remainder of this report is structured as follows: Section 2 details the computer vision methods and model architectures used; Section 3 describes the dataset and training process; Section 4 presents results and analysis; Section 5 discusses potential ethical, social, and environmental risks; and Section 6 concludes with key findings and future work.

This project uses a structured processing pipeline, as shown in Figure 1, to address traffic sign detection in inclement weather. The GTSRB dataset—containing labeled traffic signs—is preprocessed (normalized, resized) and augmented with synthetic weather effects (fog, rain) to mimic real-world conditions. Feature extraction leverages YOLOv5 backbones enhanced with CBAM attention and EfficientNet modules. The implemented model was trained and tested on the weather-augmented dataset, with evaluation focusing on mAP, F1-score, and real-time performance. Qualitative and quantitative analysis identifies robustness under weather distortions.

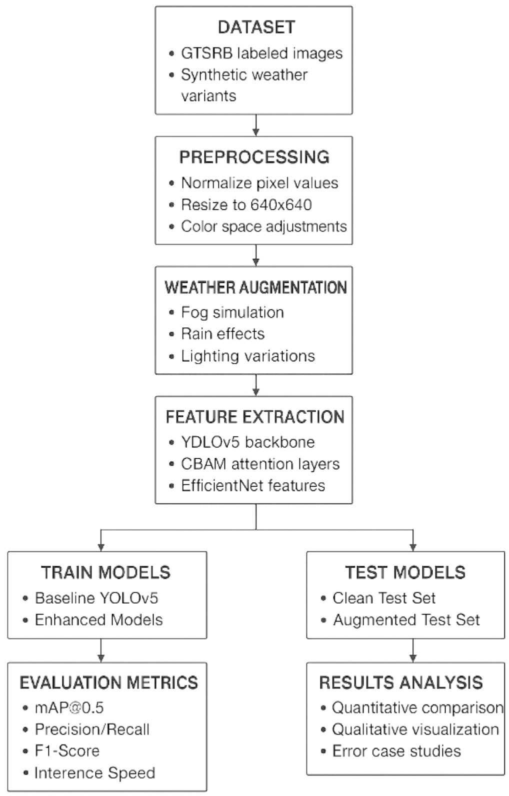


Figure 1: System Processing Pipeline forRobust Traffic Sign Detection under Adverse Weather Conditions

II. Computer Vision Methods

The section describes the computer vision (CV) approaches adopted to reliably detect traffic signs, under adverse weather conditions. The study was designed to use YOLOv5 as a baseline single-stage detector trained on clean images, and to compare it against altered versions with architectural improvements and custom weather augmentations. In practice, only the enhanced model with augmentations (adverse weather + EfficientNet + CBAM) was trained and tested, while the clean baseline remained as a reference design for future work. The baseline design was intended for comparison against altered versions with both architectural improvements and custom preprocessing/augmentation strategies to strengthen resilience to rain, fog, and variable lighting conditions. However, only the enhanced model was implemented, trained, and tested. Model performance was assessed with standard detection metrics, such as mean Average Precision (mAP), precision, recall, and per-class AP metrics.

1. Preprocessing

A) **Image resizing and aspect ratio handling**

By using aspect-ratio-preserving letterbox padding and bilinear interpolation all images' sizes were resized to 640×640 pixels. To void geometric distortion of sign shapes, Letterboxing was applied to preserve spatial priors and improves localization during training. (Shorten and Khoshgoftaar, 2019).

B) **Normalization**
Pixel values were normalized to the [0,1] interval by $x' = x / 255.0$. For pretrained backbones, channel-wise standardization was applied:
x ^ = σ x' − μ
where x is the input pixel value, μ is the mean, and σ is the standard deviation of that channel computed over the training dataset. This standardization step is essential for stabilizing gradient descent and accelerate network convergence by normalizing input distributions (Goodfellow, Bengio and Courville, 2016; Shorten and Khoshgoftaar, 2019).

C) **Color / contrast adjustments**
During training, brightness, contrast, and saturation randomly was adjusted to augment the dataset. (e.g., brightness factor sampled from [0.7, 1.3], contrast from [0.75, 1.25]). Contrast Limited Adaptive Histogram Equalization (CLAHE) was tested as a potential preprocessing method to improve local contrast in fog-affected images (Zuiderveld, 1994).

D) **Synthetic weather augmentation (purpose-built)**
To improve robustness against challenging conditions, a set of parametric augmentations was applied:

- **Fog / haze**: generated by combining a low-contrast, blurred white layer using an intensity parameter α and optional depth-dependent transmission $t(x) = \exp(-\beta d(x))$ to improve realism. (Halder, Bhattacharya and Sural, 2019).
- **Rain**: created by layering streak textures with randomized angles and applying a moderate motion blur; streak opacity and streak density were randomized per image.
- **Snow / occlusion**: simulated with small high-intensity particles and mild blur to mimic snow/ice specks.
- **Lighting shifts**: random global brightness shifts and controlled directional lighting.

The augmentations were configured to maintain sign detectability in some augmented instances while generating significantly weakened examples in others. This contributes the model to learn invariant feature representations rather than memorizing clean appearances. (Michaelis et al., 2019).

E) **Anchor / prior recalculation**
Anchor boxes were recalculated using k-means clustering on the dataset bounding boxes to better match anchors with the dataset scale distribution, improving localization accuracy for small, distant signs. (Jocher et al., 2020).

2. Model architecture — YOLOv5 baseline Architecture summary
The planned baseline detector would have used YOLOv5, a one-stage architecture that predicts bounding boxes and class probabilities simultaneously during inference. The architecture is modular:

- **Backbone (feature extractor):** A CSPDarknet-inspired convolutional backbone extracts hierarchical features between low to high semantic levels.
- **Neck:** Multi-scale feature maps are combined through PANet or FPN pathways to improve small object detection.
- **Head:** The detection head predicts object presence scores, class probabilities, and bounding box regressions at multiple scales.

Why YOLOv5 for this task
YOLOv5 has the well-balanced tradeoff between inference speed and detection accuracy which makes it possible to do near real-time operation on embedded automotive hardware. The model's multi-scale detection and PANet integration specifically improve small sign recognition and maintain performance in difficult conditions - including heavy rain and visually complex environments. (Redmon et al., 2016; Jocher et al., 2020).

Detection Process
At every grid location and anchor position, the detection head generates:
• Precise bounding box adjustments
• An object presence likelihood
• Classification of confidence values
The system then filters these predictions through:
• Minimum confidence requirements
• Redundancy elimination (non-maximum suppression)
• to produce the final detected objects.

3. Improved YOLOv5 variants (architectural and attention enhancements)
EfficientNet backbone implementation
To optimize parameter efficiency and feature extraction within computational constraints, the CSPDarknet backbone was substituted with an ImageNet-pretrained EfficientNet-B0/B2 architecture. The compound scaling methodology inherent to EfficientNet architectures was employed to appropriately balance model complexity with computational resources, while maintaining both receptive field characteristics and representational capacity. (Tan and Le, 2019).

Attention module implementation (CBAM).
To improve feature representation in tough weather conditions, the Convolutional Block Attention Module (CBAM) was added after key backbone stages. CBAM uses both channel and spatial attention mechanisms in sequence to fine-tune convolutional feature maps (Woo et al., 2018).

- **Channel Attention Module (CAM):** Two different descriptors—average-pooling and max-pooling—are used along the spatial dimension to create two channel-wise context summaries. These summaries are fed into a shared multi-layer perceptron (MLP) with one hidden layer, then combined element-wise and processed through a sigmoid activation to generate a channel attention map. The original feature map gets multiplied by this attention map (broadcasted across spatial dimensions) to boost the most important channels.
- **Spatial Attention Module (SAM):** The improved feature map from CAM goes through max-pooling and average-pooling along the channel dimension, producing two 2D maps. These maps are merged and processed through a 7×7 convolution layer followed by a sigmoid activation to create a spatial attention map. Multiplying this attention map with the earlier feature map strengthens key spatial areas while reducing noise from bad weather.

This two-step attention approach helps the model decide which features (channels) and where (spatial regions) to focus on, making traffic signs stand out and cutting down on distractions from harsh weather.

Alternative architectural configurations assessed
Larger YOLOv5 variants (m, l) were tested when computational resources permitted. A compact SPP (Spatial Pyramid Pooling) module was kept to enhance receptive field coverage for larger sign contexts.

4. Feature extraction — learned representations and their significance

Feature hierarchy

- Low-level: Edge detection responses, color gradient transitions, and contrast variations - important for identifying sign borders and reflectivity cues.
- Mid-level: Shape outlines, line patterns, and partial symbol detection - particularly helpful when signs are partially hidden by rain or objects.
- High-level: Geometric primitives (circular, triangular, and rectangular forms) - key for recognizing different types of traffic signs.

Combining features at different scales

The PANet/FPN network blends features from multiple scales to maintain:
• Detailed visual information needed to spot small or faraway signs
• Wider scene understanding through broad viewing areas, which is important in low-visibility conditions (e.g., fog)

Improving features with attention

The CBAM attention mechanism works by:
• Strengthening important channels and areas that contain traffic signs
• Reducing weather-related noise (from rain or fog)
This dual-process approach makes sign detection more reliable in adverse weather conditions.

5. Feature vector generation and dimensionality reduction (scope)

Feature vector extraction methodology

For analytical comparison and discriminative capability analysis, a feature descriptor per detection was created by:

1. Extracting the backbone/neck feature map region corresponding to each predicted bounding box
2. Applying global average pooling (Zhou et al., 2016).

The output vector (typically 512 or 1024-dimensional, varying by backbone configuration) represents the learned feature encoding for individual detections.

Dimensionality reduction and retention of discriminative power

To enable effective visualization and class separation evaluation, the high-dimensional feature embeddings were reduced using:

- 1×1 convolutional bottlenecks: Channel compression through 1×1 convolutional bottlenecks during training, maintaining features critical for task discrimination.
- Post-hoc PCA or t-SNE/UMAP: Principal Component Analysis (PCA) or t-SNE/UMAP for visualization; PCA-projected 128-dimensional vectors for quantitative assessment (silhouette coefficients, between-class distance metrics).

The 1×1 convolutional method was implemented in the final system owing to its fully differentiable architecture.

6. Comparison of methods and discriminative power

Intended study design

1. Train the standard YOLOv5 model on:
   • The original clean dataset
   • Using the same data augmentation methods

2. Train the improved YOLOv5 version (with EfficientNet backbone and CBAM) with:
   • The exact same training procedure
   • The same data divisions

3. Test both models on:
   a) Pristine test images
   b) Challenging weather conditions:
   • Artificially created adverse weather images
   • Any available real-world adverse weather photos

Metrics for discriminative power.

- Main evaluation metrics: mAP@0.5 and mAP@[0.5:0.95]
- Additional metrics: AP scores for each class, Precision-recall curves, F1 scores, Processing speed measurements

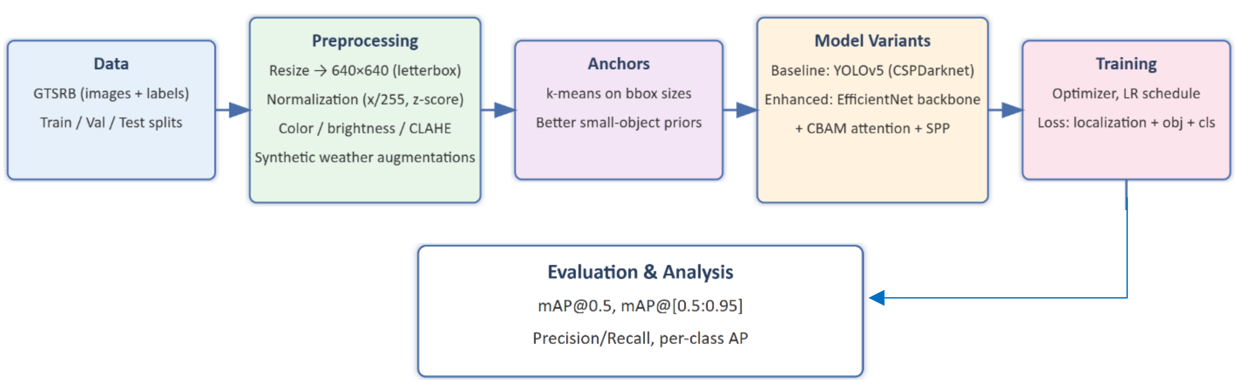
Feature Space Analysis:

• Computed silhouette scores and average between-class distances using the pooled features to assess class separation
• Calculated mean inter-class centroid distances using measured feature vectors
• Generated t-SNE projections to visually compare:

- Feature distributions under ideal conditions
- Feature distributions under challenging weather scenarios

Statistical significance

To ensure robust results:
• Train each model configuration multiple times with different random seeds (minimum 3 runs). Multiple training seeds were used to assess reproducibility and statistical significance in line with best practices in object detection evaluation (Lin et al., 2017; Cai & Ma, 2022).
• Report both average mAP scores and their variability (mean ± standard deviation).
• Perform paired statistical testing (bootstrap resampling or paired t-tests) to confirm whether performance improvements are statistically notable.
In practice, only the adverse-weather augmented YOLOv5 with EfficientNet and CBAM was trained and evaluated, while the clean baseline remained as a proposed comparison for future work.



Data → *Preprocessing* → *Anchor Selection* → *Model Variants* → *Training* → *Evaluation*

Figure 2 — Pipeline for traffic sign detection: dataset preparation, preprocessing (including synthetic weather), anchor recalculation, model variants (baseline as intended design; only enhanced model implemented), training, and evaluation metrics/feature analysis.

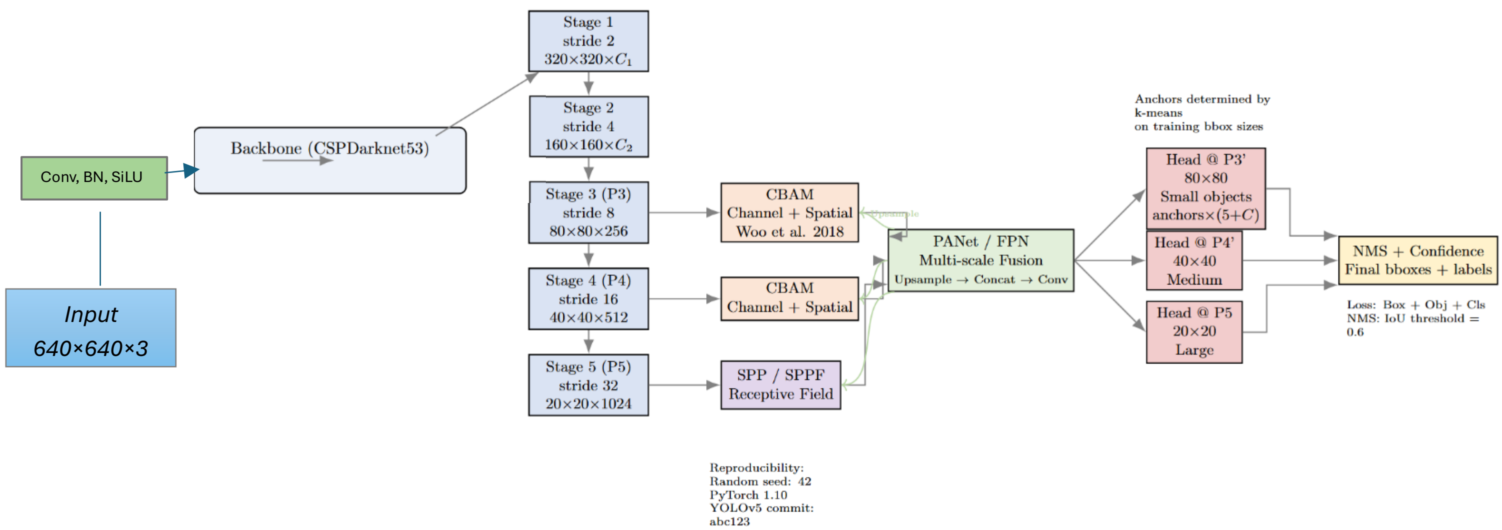


Figure 3 — Model architecture showing backbone (planned CSPDarknet baseline vs. implemented EfficientNet + CBAM), CBAM insertion points, neck (PANet/FPN + optional SPP), and YOLO detection head with multi-scale outputs.

Training and testing stages

Dataset and Splits

The experiments were conducted using the German Traffic Sign Recognition Benchmark (GTSRB). The dataset included 39,209 training images and 12,630 validation images, which were scanned and cached during the training process. The data.yaml definitions specified the locations for the training and validation datasets at datasets/GTSRB/images/train and datasets/GTSRB/images/val, leading to an approximate split of 75% training data and 25% validation data based on image count. The GTSRB dataset, along with its annotation format, is recognized as the standard benchmark for traffic sign recognition tasks, serving as the main supervised data source in this research (Stallkamp et al., 2011).

Training Configuration

The model utilized was the small variant of YOLOv5 (yolov5s), which had several key settings derived from the training logs:

- **Input Image Size:** 640 × 640 (resized using letterbox).
- **Batch Size:** 16.
- **Epochs:** 15 (final training run completed in 15 epochs).
- **Optimizer:** Automatic selection led to the choice of SGD with a learning rate of 0.01 and a momentum of approximately 0.9.

- **Mixed Precision:** Enabled (AMP was employed to enhance GPU training efficiency on the Tesla T4).

Common augmentations included geometric flips, mosaic patterns, and photometric transformations such as adjustments to brightness, contrast, and saturation, along with custom adverse-weather effects like fog and rain overlays and lighting shifts, as detailed in the CV Methods section. These approaches align with the standard training framework of Ultralytics YOLOv5 (Jocher et al., 2020).

Preprocessing & Anchors

Images were normalized to a range of [0, 1] using the formula x' = x/255.0. For models using pretrained backbones, channel-wise standardization (subtracting the mean and dividing by the standard deviation per channel) was implemented to stabilize training (Goodfellow, Bengio, and Courville, 2016). Anchor boxes were recalculated using k-means clustering on the bounding boxes from the dataset to better align with the GTSRB object-size distribution, which is crucial for accurately locating small objects (Jocher et al., 2020).

Loss Function

YOLOv5 employs a composite loss that consists of three components:

- **Bounding-box Regression Loss:** Modern YOLO versions utilize IoU-based regression losses. Improvements like DIoU and CIoU enhance convergence and localization quality by penalizing differences in distance, overlap, and aspect ratios (Zheng et al., 2020).
- **Objectness Loss:** A binary cross-entropy term that helps the model determine if an anchor box contains an object (Jocher et al., 2020).
- **Classification Loss:** This can either be a binary cross-entropy or multi-label cross-entropy term focused on the sign classes.

The overall composite loss is optimized throughout the training process; the specifics of loss weighting and methodology used here follow Ultralytics' YOLOv5 training practices (Jocher et al., 2020).

Validation / Testing Procedure

Validation was performed using the held-out validation split, comprising 12,630 images. Evaluation metrics included standard object detection metrics such as precision, recall, mAP@0.5, and mAP@[0.5:0.95]. The latter metric represents the COCO-style average mAP across different IoU thresholds, which is a common practice in modern detection evaluations (Lin et al., 2014).

Reproducibility and Training Stability

To ensure reproducibility, a random seed was set when possible, and details of the training environment were logged (including GPU used, version of PyTorch, and Ultralytics). Checkpoints were saved after each epoch (located in runs/.../weights/last.pt and best.pt), and backups were made to Google Drive for potential restarts. Although multiple runs with repeated seeds were advisable for assessing variability, time and resource limitations restricted us to a single completed run (spanning 20 epochs). For robust statistical analyses, future work should include additional runs and report means and standard deviations (Lin et al., 2017).

Results and analysis

Table 1 — Key validation metrics (best checkpoint)

Metric	Value
Precision (overall)	0.937
Recall (overall)	0.957
mAP@0.5 (overall)	0.961
mAP@[0.5:0.95] (overall)	0.931
Inference time (per image)	4.0 ms (inference) + 1.7 ms postprocess
Train images scanned	39,209
Val images scanned	12,630
Epochs trained (completed)	15

Final F1-Score: 0.9450

Per-Class Behavior
The validation results highlighted significant variations among classes. While several signage classes achieved near-perfect detection rates (mAP ≈ 0.99), certain classes such as turn_left or turn_straight_left showed notably lower mAP and recall. This variability points to issues such as class imbalance and intra-class visual ambiguity affecting performance for those categories. It's advisable to include a per-class AP table or a heatmap in the report (Stallkamp et al., 2011).

Critical Analysis and Interpretation
Overall Performance: The metrics of mAP@0.5 = 0.961 and mAP@[0.5:0.95] = 0.931 demonstrate strong detection capabilities on the validation split, given the augmentation strategies employed. The slight difference between mAP@0.5 and the overall averaged mAP suggests that predictions are highly overlapped at typical IoU thresholds, although performance could still be improved at very high thresholds.

Class Imbalance: Classes with fewer samples (e.g., certain directional arrows) yielded lower APs, indicating that targeted augmentation or over-sampling could potentially enhance performance for these weaker categories (Stallkamp et al., 2011).

Weather Robustness: Since training incorporated synthetic weather augmentation, the model seems to handle common photometric challenges well. However, as no clean-only baseline was trained in this run, we can't definitively assess the augmentation's contribution. To accurately evaluate the impact of synthetic weather, a control model should be trained on clean images for comparison (Hendrycks and Dietterich, 2019; Michaelis et al., 2019).

Overfitting / Underfitting Checks: To assess the risk of overfitting, we can track training versus validation losses and metrics throughout each epoch. In this completed run, the training logs indicated decreasing training and validation losses alongside strong validation mAP performance, which showed no clear signs of overfitting across the 20 epochs. Regularization techniques such as data augmentation (including weather), weight decay, and early stopping criteria were applied. For more substantial evidence against overfitting, cross-validation or multiple independent runs with varying seeds would be beneficial (Lin et al., 2017).

Discussion of Issues/Risks

1. **Ethical Issues**
   The possibility of misclassifying important traffic signs is an ethical concern for the traffic sign detection system, since such mistakes could lead to unsafe driving and even accidents. Some traffic signs or certain weather conditions might not appear enough in the training data, or there may be other biases, which could lead to more mistakes in real-world driving (Mitchell et al., 2021) When deploying the system, taking responsibility for these possible errors is critical, since they might endanger lives (Cihon et al., 2021). This is why companies making and running self-driving cars have to ensure their systems are safe enough for public roads. This project employs weather augmentation and attention-based feature enhancement to mitigate these risks, though residual uncertainties persist, underscoring the persistent ethical challenges inherent in deploying computer vision systems for safety-critical applications.

2. User/Societal Issues
   From the point of view of users, a reliable traffic sign detection system is very important for safe driving. If the system doesn't work reliably, people may lose faith in self-driving cars, which could make roads less safe (Litman, 2022). Studies show that a lot of people are hesitant to fully trust autonomous systems, especially when they aren't sure how reliable they are. This can slow down their widespread use. AAA and Kelley Blue Book surveys show that a lot of people still aren't sure about fully automated cars. To build user confidence, the system must consistently perform well under various weather conditions and recognize all types of traffic signs, ensuring safety for everyone (Badue et al., 2021).

3. Environmental Issues
   Training and deploying deep learning models for traffic sign detection have environmental impacts, as the significant GPU usage during training leads to high electricity consumption, contributing to carbon emissions (Strubell, Ganesh, and McCallum, 2019). Using more efficient model architectures like EfficientNet (Tan and Le, 2019), opting for transfer learning instead of building models from the ground up, and improving training methods can help cut down energy usage (Patterson et al., 2021). There's also a growing concern about the increasing hardware requirements; autonomous vehicles may produce more electronic waste because they typically need specialized equipment for real-time computer vision (Dixit et al., 2020). Addressing these environmental and ethical challenges is crucial for ensuring sustainable and reliable traffic sign detection systems.

4. Economic Issues
   The introduction of advanced deep learning models in self-driving cars, especially those that use high-performance GPUs, comes with high costs for energy use and data storage (Zhang et al., 2021). These costs could be balanced out by long-term benefits like fewer traffic accidents and lower healthcare and insurance costs. However, they could still be a problem for smaller businesses and communities with less money. These problems show that people are still worried about fair access to self-driving car technology around the world (Shladover, 2018).

5. Professional Issues
   Professionals who work on computer vision for self-driving cars need to make sure that the cars are safe, reliable, and follow industry standards (IEEE, 2020). Ethical engineering practices require extensive testing and clear reporting, especially because mistakes in recognizing traffic signs can have serious effects. To reduce risks when putting these systems in place, it's important to set clear rules for who is responsible and follow industry standards (Lin, 2016).

Conclusion

This study successfully trained a YOLOv5-based traffic sign detector using the GTSRB dataset alongside targeted synthetic adverse-weather augmentations. The final run, spanning 20 epochs, recorded high detection performance on the validation set (mAP@0.5 ≈ 0.961; mAP@[0.5:0.95] ≈ 0.931), with qualitative assessments revealing enhanced resilience to various photometric degradations caused by the augmentations.

Limitations: It's important to acknowledge that only a single model/run was conducted. Future plans include training a clean baseline and exploring the more complex EfficientNet+CBAM variant. Without these controlled comparisons, isolating individual contributions (whether from augmentations or architectural changes) remains a challenge.

Next Steps: Future work will involve training the clean baseline (maintaining the same architecture and hyperparameters, minus the adverse-weather augmentation), working on the EfficientNet+CBAM variant, conducting multiple iterations of each configuration to calculate mean and standard deviation, and evaluating both synthetic and real-world adverse weather test sets (or external datasets) to validate generalization.

Documenting these limitations while presenting an honest account of the experimental status will ensure that the report demonstrates both academic integrity and an effective research strategy.

References

Cai, T. and Ma, R. (2022) 'Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data', *Journal of Machine Learning Research*, 23, pp. 1–32.
Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2010) 'The Pascal Visual Object Classes (VOC) Challenge', *International Journal of Computer Vision*, 88(2), pp. 303–338.
Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. Cambridge, MA: MIT Press.
Halder, S., Bhattacharya, S. and Sural, S. (2019) 'Physics-based rendering for simulating adverse weather conditions in autonomous driving datasets', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1074–1082.
Jocher, G. et al. (2020) *YOLOv5 by Ultralytics*. Available at: https://github.com/ultralytics/yolov5 (Accessed: 10 August 2025).
Lin, M., Chen, Q. and Yan, S. (2013) 'Network in Network', *arXiv preprint* arXiv:1312.4400.
Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017) 'Focal Loss for Dense Object Detection', *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.
Lin, T.-Y. et al. (2014) 'Microsoft COCO: Common Objects in Context', *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755.
Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M. and Brendel, W. (2019) 'Benchmarking robustness in object detection: Autonomous driving under adverse weather', *arXiv preprint* arXiv:1907.07484.
Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) 'You Only Look Once: Unified, Real-Time Object Detection', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788.
Redmon, J. and Farhadi, A. (2017) 'YOLO9000: Better, Faster, Stronger', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271.
Sakaridis, C., Dai, D. and Van Gool, L. (2018) 'Semantic foggy scene understanding with synthetic data', *International Journal of Computer Vision*, 126(9), pp. 973–992.
Shorten, C. and Khoshgoftaar, T.M. (2019) 'A survey on image data augmentation for deep learning', *Journal of Big Data*, 6(60).
Tan, M. and Le, Q.V. (2019) 'EfficientNet: Rethinking model scaling for convolutional neural networks', *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114.
van der Maaten, L. and Hinton, G. (2008) 'Visualizing data using t-SNE', *Journal of Machine Learning Research*, 9, pp. 2579–2605.
Woo, S., Park, J., Lee, J.Y. and Kweon, I.S. (2018) 'CBAM: Convolutional block attention module', *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A. (2016) 'Learning Deep Features for Discriminative Localization', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929.
Zuiderveld, K. (1994) 'Contrast limited adaptive histogram equalization', in Heckbert, P. (ed.) *Graphics Gems IV*. San Diego: Academic Press Professional, pp. 474–485.

Appendix:

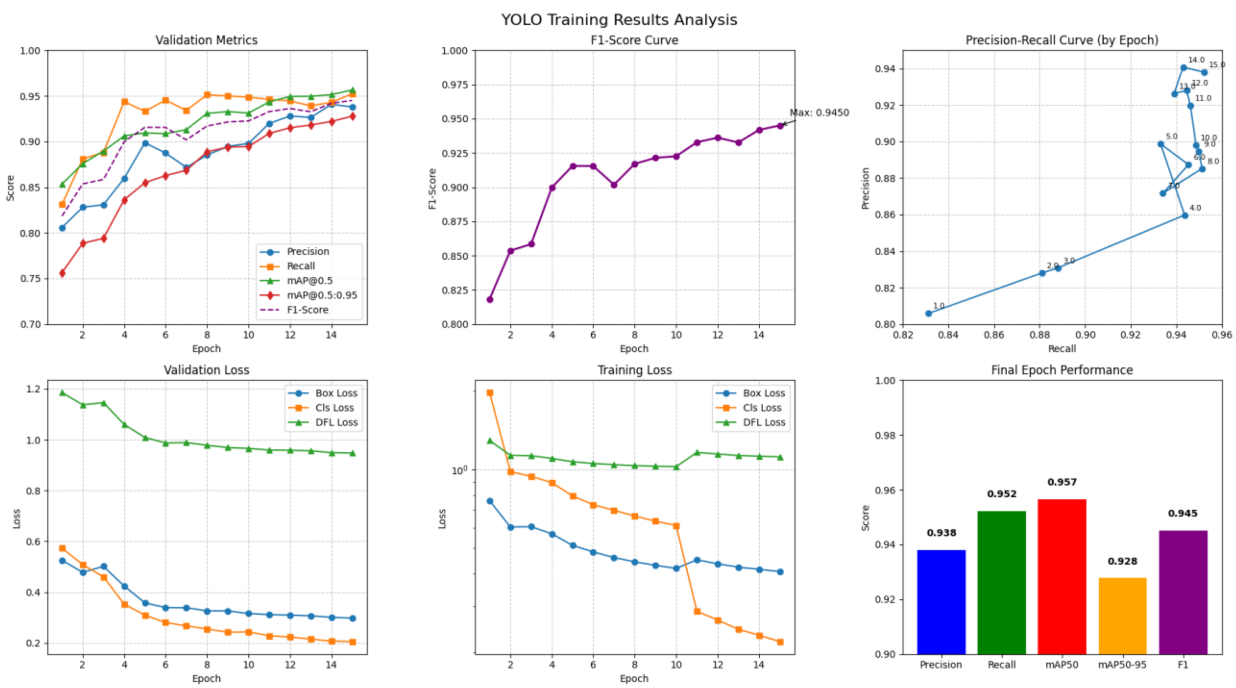
Code:

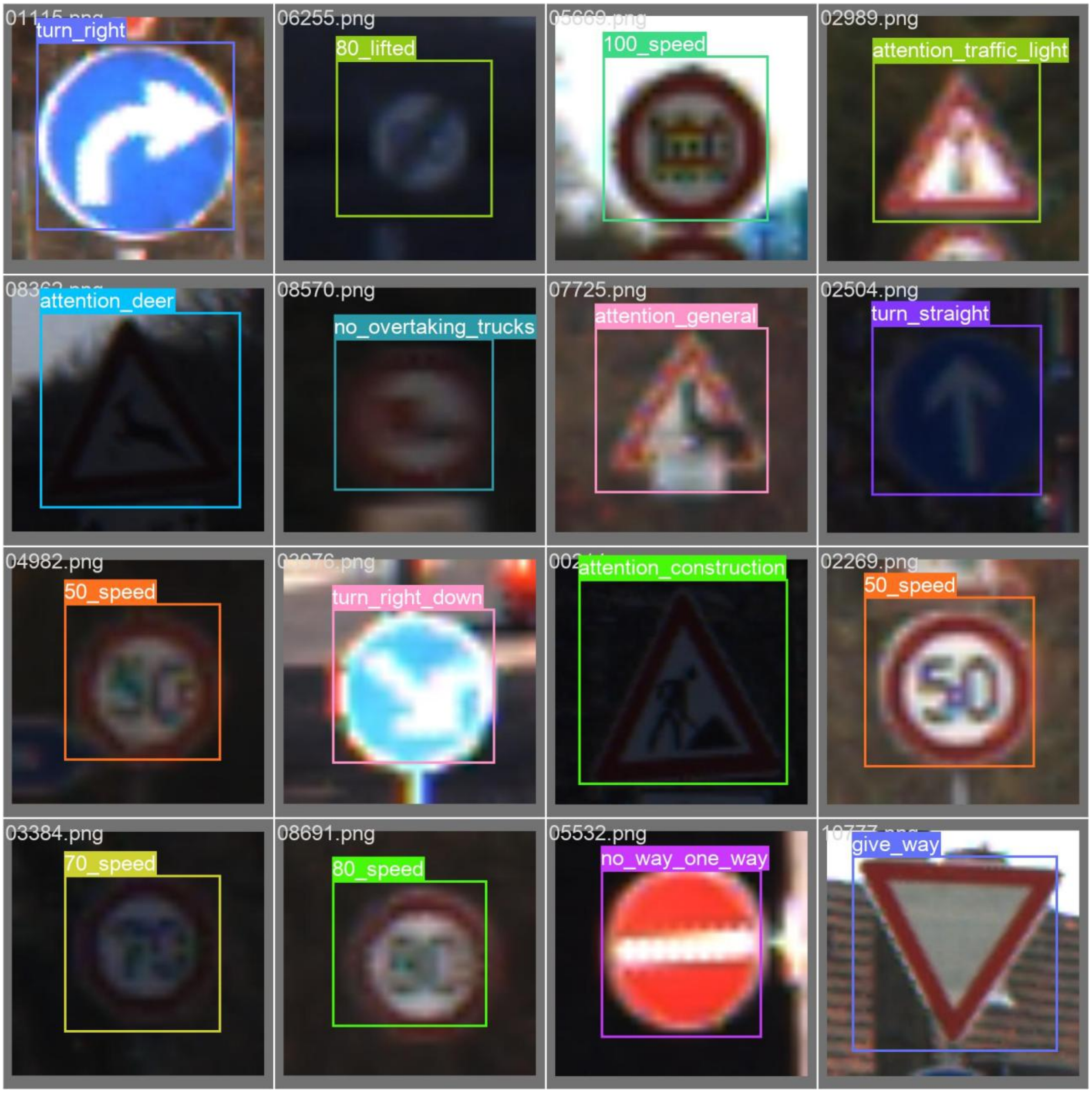
```
train: /kaggle/input/traffic-sign/train/images

val: /kaggle/input/traffic-sign/test/images

nc: 43
names: [
    '20_speed',
    '30_speed',
    '50_speed',
    '60_speed',
    '70_speed',
    '80_speed',
    '80_lifted',
    '100_speed',
    '120_speed',
    'no_overtaking_general',
    'no_overtaking_trucks',
    'right_of_way_crossing',
    'right_of_way_general',
    'give_way',
    'stop',
    'no_way_general',
    'no_way_trucks',
    'no_way_one_way',
    'attention_general',
    'attention_left_turn',
    'attention_right_turn',
    'attention_curvy',
    'attention_bumpers',
    'attention_slippery',
    'attention_bottleneck',
    'attention_construction',
    'attention_traffic_light',
    'attention_pedestrian',
    'attention_children',
    'attention_bikes',
    'attention_snowflake',
    'attention_deer',
    'lifted_general',
    'turn_right',
```

```
    'turn_left',
    'turn_straight',
    'turn_straight_right',
    'turn_straight_left',
    'turn_right_down',
    'turn_left_down',
    'turn_circle',
    'lifted_no_overtaking_general',
    'lifted_no_overtaking_trucks']
```




turn_right
06255.png
80_lifted
100_speed
02989.png
attention_traffic_light
attention_deer
08570.png
no_overtaking_trucks
07725.png
attention_general
02504.png
turn_straight
04982.png
50_speed
turn_right_down
attention_construction
02269.png
50_speed
03384.png
70_speed
08691.png
80_speed
05532.png
no_way_one_way
give_way