Synth2Det: Loss-Free Unpaired Adverse-Weather Synthesis and Object-Detection Enhancement for Autonomous Driving

Zeng Tianyi[1]

[1]*Department of Computing, The Hong Kong Polytechnic University*

This manuscript was compiled on May 10, 2025

Abstract

Autonomous vehicles necessitate resilient perception frameworks that sustain operational reliability in challenging weather conditions, such as snowfall, which pose safety risks yet remain critically underrepresented in current datasets. To address this, we present **Synth2Det**, a framework that *synthesizes* high-resolution adverse-weather imagery from unpaired clear-weather video inputs and *immediately employs* the generated data to refine object detection models, retaining full-HD fidelity throughout. Our methodology introduces two key innovations: (1) A revised CycleGAN architecture substitutes transpose-convolution operations with bilinear upsampling cascaded with 3×3 convolutional layers, systematically eliminating checkerboard artifacts while preserving spatial clarity; and (2) A *three-stage curriculum learning* approach for YOLO11m, where the model undergoes sequential fine-tuning across synthetic-to-real dataset transitions to enhance generalization. Training progress is systematically documented using `WeightsBiases` and Ultralytics HUB, ensuring experimental transparency. When tested on the ACDC benchmark under heavy snowfall, our method attains a **195.75%** increase in mean average precision (mAP) relative to the baseline YOLO model, achieving real-time inference speeds of 54.7ms on a single RTX 4090 GPU. These findings underscore the efficacy of domain-specific image-to-weather translation paired with task-driven adaptation in resolving the realism deficit of autonomous perception systems, delivering measurable improvements in rare-weather robustness—without reliance on labor-intensive data acquisition or labeling pipelines for underrepresented scenarios.

Keywords: *Image2Image Translation, Object Detection, CycleGAN, YOLO, Fine-tuning*

E-mail address: 22098941d@connect.polyu.hk

Contents

1. Introduction

1.1. Motivation

It is critical for perception systems of autonomous vehicles to maintain reliability under adverse weather conditions such as rain, snow, fog, and night-time. However, contemporary public driving datasets predominantly feature clear-weather scenarios, while manually annotating rare weather conditions is prohibitively expensive. Synthetic data generation offers a pragmatic alternative, yet two critical challenges persist: (1) Visual fidelity at full resolution —traditional CycleGAN [5] architectures employ transpose-convolution layers that induce checkerboard artifacts [1], degrading downstream detection performance; and (2) Task-awareness —existing translation methods are often evaluated solely by human realism metrics, leaving unresolved whether synthetic data meaningfully improves modern detectors.

1.2. Key Contributions

Our work addresses the above-mentioned problems by proposing a two-stage pipeline that first focuses on image synthesis and then fine-tunes a pretrained YOLO model [17] to improve its performance on the dataset containing generated images from the previous stage. We achieve loss-free high resolution synthesis of images and improve the pretrained YOLO to better detect the images generated by our modified CycleGAN. This is meaningful for the advancement of autonomous driving systems.

2. Related Work

2.1. Weather-Specific Image2Image Translation

Early advancements in image-to-image translation depended on paired datasets, exemplified by Pix2Pix [2], which employs conditional adversarial learning to establish deterministic input-output mappings. CycleGAN eliminated this dependency by introducing cycle-consistency constraints, enabling unpaired frameworks like UNIT [3] and MUNIT [7]. Recent advancements focus on scale-aware architectures (e.g., SPADE [9]) or disentangled representation learning (e.g., CUT [12]) for high-resolution image processing. Weather-specific adaptations utilize conditional codes or multi-domain generators to convert clear-weather images into adverse weather scenarios such as rain, snow, or fog. However, many implementations still incorporate deconvolution layers, which are known to produce checkerboard artifacts. Our methodology implements a CycleGAN architecture with bilinear upsampling to ensure spatial consistency and artifact-free full-resolution outputs during inference through a tiling-based approach.

2.2. Objection Detection Under Adverse Weather Conditions

Recent advancements in object detection under adverse weather conditions focus on enhancing robustness through domain-specific adaptations. Traditional approaches include pre-processing techniques like dehazing and deraining, followed by off-the-shelf detectors such as YOLOv3/v5. Modern methods integrate weather-aware training directly into network architectures, exemplified by frameworks like DA-Faster R-CNN [6], which employs adversarial feature alignment for domain adaptation. Synthetic data generation has emerged as a cost-effective alternative, with works like Synth2Det leveraging modified CycleGAN architectures to synthesize high-fidelity adverse-weather imagery while preserving resolution. Task-specific frameworks such as D-YOLO and Image-Adaptive YOLO incorporate robustness against haze, snow, and fog through scale-aware designs or curriculum learning strategies. Additionally, domain randomization and synthetic-to-real adaptation pipelines bridge the gap between simulated and real-world data, as seen in studies utilizing ACDC [16] and Snow100K [10] datasets for evaluation. These efforts highlight the shift toward end-to-end solutions that combine synthetic data augmentation, domain adaptation, and architecture innovations to address weather-induced degradation.

3. Dataset Construction

3.1. Prepare Dataset for Image Translation

The foundation of our dataset is the ACDC adverse-weather benchmark, which comprises 4006 high-resolution RGB images evenly distributed across fog, night, rain, and snow scenarios. The dataset structure is illustrated in Figure 1. For each adverse-condition frame, the dataset provides geographically aligned clear-weather counterparts captured at the same location. These "loose pairs" enable qualitative comparisons without requiring strict pixel-level alignment. For image translation, we focus on clear-snow image pairs, resulting in an 4:1 split for training and validation sets. While these pairs aid visual analysis (see Figure 2), all CycleGAN training treats domains as unpaired to maintain methodological rigor.

After obtaining the dataset, we do data preprocessing work for all snow images by unifying their sizes to 256x256px resolution for better suitability to CycleGAN. In our work we only consider snow images to simplify the tranining process. Other adverse conditions can be taken into our pipeline by exactly the same strategy. The total number of images used in CycleGAN training specific to clear-snow domain is shown in Table 1.

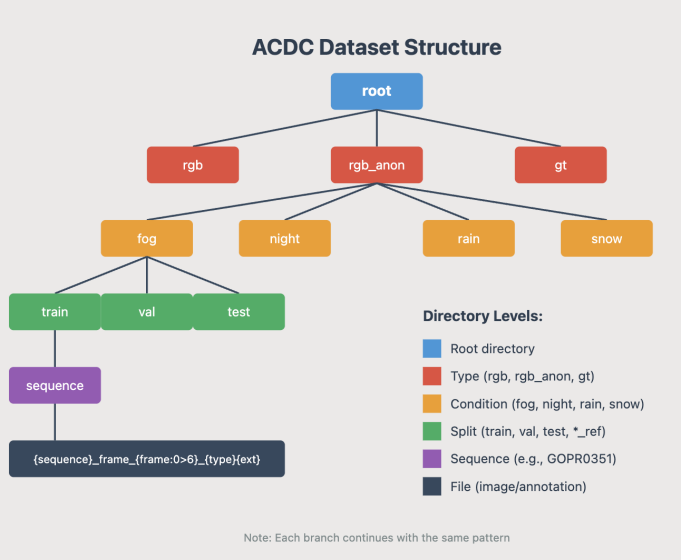

Figure 1. The ACDC dataset's directory layout

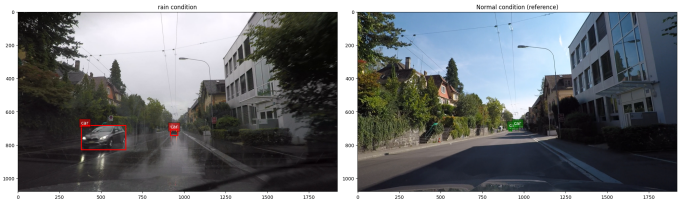

Figure 2. An example pair of aligned clear-rain image

3.2. Prepare Dataset for YOLO detection: Annotation Transfer and Label Cleaning

The ACDC dataset's COCO annotations encompass 19 object categories, which we converted to the YOLO framework using a mapping dictionary: 24:0, 25:1, 26:2, 27:3, 28:4, 31:5, 32:6, 33:7. This mapping collapses to eight target classes—person, rider, car, truck, bus, train, motorcycle, and bicycle. Bounding boxes violating geometric thresholds defined in our algorithm (refer to Algorithm 1) are discarded to address detection challenges posed by synthetic environments. Specifically, adverse weather simulations (e.g., fog or rain) may introduce atmospheric particles that obscure distant objects, mimicking real-world visibility degradation.

By this filtering strategy, each dataset partition is reduced to 50% of its retained samples to minimize overfitting during fine-tuning. This results in three YOLO-formatted subsets: real clear-weather, real snow scenes, and synthetically generated snow images. The final statistics after cleaning are shown in Table 2. The workflow of the overall dataset construction is shown in Figure 3.

4. Methodology

4.1. Task1: Image2Image Translation Network

We employ a ResNet-based generator and a PatchGAN discriminator.

4.1.1. Generator Architecture

We employ a ResNet-based encoder-transformer-decoder structure specifically optimized for adverse weather translation tasks. The symmetrical design begins with an input processing block utilizing 7x7 convolutions with reflection padding to mitigate border artifacts, followed by instance normalization with learnable affine parameters to preserve instance-specific stylistic features. The encoder phase incorporates two downsampling blocks that progressively halve spatial dimensions while doubling channel depth through strided convolutions (256×256→64×64), capturing hierarchical weather patterns.

Table 1. Task 1—Image counts used for CycleGAN training (clear–snow example).

Domain	Train	Val
Clear (good)	400	100
Snow (adverse)	398	102

Algorithm 1 Automatic Annotation Transfer & Label Cleaning

Require: COCO-annotated dataset $\mathcal{D}$ containing domains {clear, snow};
synthetic-snow generator G_{cyc};
class–mapping dictionary $\mathcal{M}$;
thresholds $\theta = (a_{\min} = 100,\ d_{\min} = 15,\ s_{\min} = 10)$;
sampling ratio $p = 0.5$
Ensure: Three YOLO-formatted datasets $\mathcal{D}^{\text{clear}}$, $\mathcal{D}^{\text{snow}}$, $\mathcal{D}^{\text{syn}}$

for all domain $c \in \{\text{clear}, \text{snow}\}$ **do**
  $\mathcal{I}_c \leftarrow$ random sample of size $p\,|\mathcal{I}_c^{\text{all}}|$ {**half-image selection**}
  for all image $I \in \mathcal{I}_c$ **do**
    $\mathcal{A} \leftarrow$ COCO annotations of I
    for all bounding box $(b, \ell) \in \mathcal{A}$ **do**
      if $\ell \in \text{keys}(\mathcal{M})$ **then** {**class transfer**}
        $\ell' \leftarrow \mathcal{M}[\ell]$
        if filter_pass(b, θ) **then** {**bbox filtering**}
          write $(\ell', \text{normalize}(b))$ to YOLO label file
        end if
      end if
    end for
  end for
end for
$\mathcal{I}_{\text{syn}} \leftarrow G_{\text{cyc}}(\mathcal{I}_{\text{clear}})$ {**generate synthetic-snow images**}
copy YOLO label files from $\mathcal{I}_{\text{clear}}$ to $\mathcal{I}_{\text{syn}}$
split each domain into `images/`{train,val} and `labels/`{train,val}
return $\mathcal{D}^{\text{clear}}$, $\mathcal{D}^{\text{snow}}$, $\mathcal{D}^{\text{syn}}$
function FILTER_PASS(b, θ) {geometric checks}
  $(w, h) \leftarrow$ width&height(b); $a \leftarrow w \times h$; $d \leftarrow \sqrt{w^2 + h^2}$
  return $(a \geq a_{\min}) \wedge (d \geq d_{\min}) \wedge (\min(w, h) \geq s_{\min})$
end function=0

Core transformation occurs via nine residual blocks featuring dual 3×3 convolutional layers with skip connections, enabling deep feature learning while maintaining gradient flow through identity mappings.

A critical innovation lies in the decoder's bilinear upsampling approach, which reduces checkerboard artifacts common in transposed convolutions by combining smooth interpolation with learnable 3×3 convolutions for feature refinement. This architecture implements instance normalization throughout to decouple style statistics from weather characteristics, particularly effective for domain shifts between clear and adverse conditions. Weather-specific adaptations include a =0.5 identity loss to preserve structural integrity during snow/rain transformations and multi-scale processing that handles both local precipitation patterns and global fog distribution through intermediate 64×64 feature maps.

Compared to the original CycleGAN, our implementation demonstrates enhanced artifact suppression through systematic reflection padding and replaces transposed convolutions with upsampling-convolution pairs. This architecture balances computational efficiency with translation quality through its symmetrical design, making it particularly suitable for autonomous vehicle applications requiring reliable weather adaptation while preserving critical structural details.

4.1.2. Discriminator Architecture

We employ a PatchGAN architecture for the discriminator that analyzes local image regions rather than the entire image, using a series

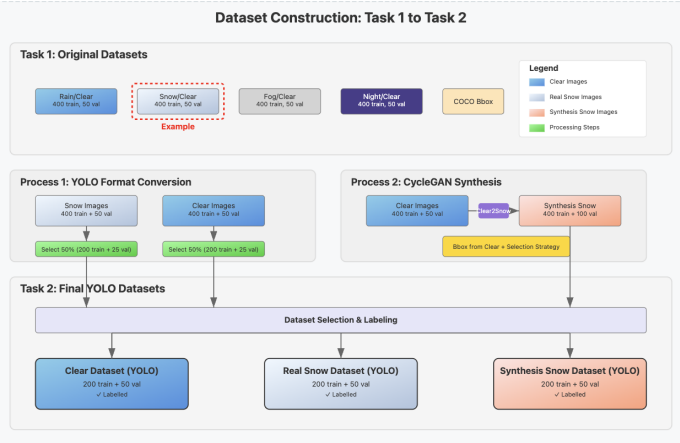

Figure 3. Workflow for the dataset construction

Table 2. Task 2—Pairs and bounding boxes after cleaning.

Condition	Split	Pairs	Cond. BBox	Ref. BBox
Fog	train	200	1146	2227
	val	50	340	544
Night	train	200	1448	2726
	val	53	276	539
Rain	train	200	1025	2125
	val	50	347	491
Snow	train	200	1421	2553
	val	50	428	637

of convolutional layers (64→128→256→512 filters) with instance normalization and LeakyReLU activations. Our design processes the image at multiple scales through strided convolutions, reducing spatial dimensions while increasing channel depth to capture hierarchical features.

Unlike traditional discriminators, it omits fully-connected layers, preserving spatial information through a final convolutional output layer that produces a matrix of patch-wise authenticity predictions. The architecture focuses on texture and style consistency by evaluating overlapping 70x70 pixel regions (receptive field size), providing detailed feedback to generators while maintaining computational efficiency.

Instance normalization stabilizes training by normalizing feature statistics within each image, and LeakyReLU prevents gradient stagnation in early layers. This localized approach proves particularly effective for weather translation tasks where coherent texture synthesis across the image plane is critical.

Figure 4 illustrates a simple workflow of our generator and discriminator.

4.1.3. Loss Functions & Optimization

We deploy a multi-component loss function system and Adam optimization to enable stable unpaired image translation. The core adversarial loss uses Mean Squared Error (MSE) with label smoothing through a modified GANLoss class, training discriminators to distinguish real/fake images while generators create convincing translations. A critical cycle consistency loss enforces bidirectional reconstruction consistency between domains (A→B→A and B→A→B). An identity loss preserves structural elements when inputs already belong to the target domain.The discriminators use standard GAN loss.

For optimization strategies, we utilize separate Adam optimizers for generators and discriminators with linear learning rate decay over 200 epochs. This balanced approach coordinates adversarial training with geometric preservation, enabling the model to learn

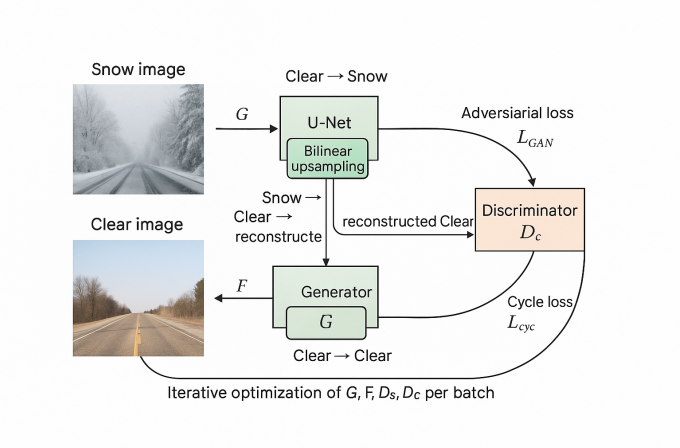

Figure 4. The workflow of generator and discrimintor

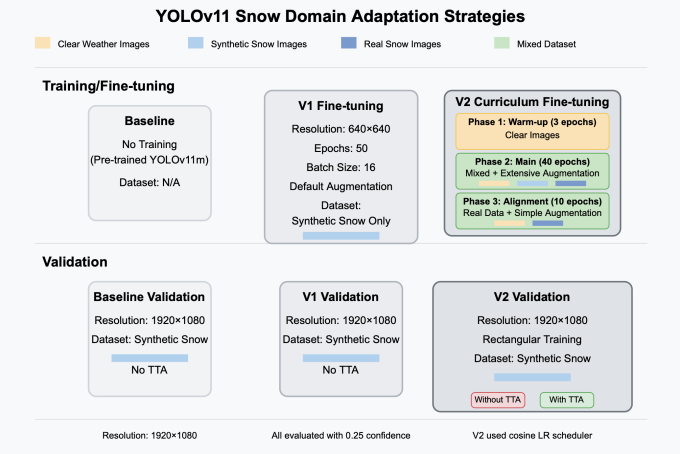

Figure 5. Method Component and Comparison

Table 3. Augmentation hyper-parameters.

Operator	Probability	Range/Details
MixUp	0.25	—
Mosaic	0.50	—
HSV Jitter	1.00	$h = 0.15, s = 0.7, v = 0.4$
Random Rotation	—	$\pm 10°$
Random Translation	—	± 0.2
Random Scaling	—	± 0.5
Horizontal Flip	0.50	—
Perspective Transform	—	0.0005

domain-invariant features while maintaining content integrity.

4.2. Task2: Fine-tuning for YOLO Detection

We design and implement several techniques to fine-tune the pra-trained YOLO model for snow domain adaptation, focusing on preserving spatial details and gradual domain transition.

4.2.1. Test-Time Augmentation & Augmentation Strategy

We implement a strategy to average predictions over 4 flips and 3 multi-scales during validation. The metric mAP can be improved slightly by adopting this method in low-contrast scenarios [15].

To reduce overfitting to synthetic patterns, we implement a robust augmentation strategy (Table 3) incorporating mosaic blending [11], mixup interpolation, color jitter perturbations, and physics-based fog/snow overlays. These methods prioritize structural consistency over texture memorization, addressing the challenge of texture degradation prevalent in adverse weather scenarios. By emphasizing geometric invariance, the model learns to leverage shape-based features—critical for reliable detection when environmental conditions distort surface-level visual cues.

4.2.2. Curriculum Learning-Based Three-Phase Fine-tuning

We devise a three-phase fine-tuning approach (refer to Algorithm 2) that employs a curriculum learning strategy to systematically adapt YOLOv11 to snowy environments while preserving detection capabilities in clear conditions.

Phase 1 (Warm-up) initializes training on original high-resolution (1920×1080) clear-weather images for 3 epochs with a reduced learning rate (0.0001), allowing the model to retain critical spatial relationships and object details at native resolution before introducing domain-shifted data.

Phase 2 (Main Fine-tuning) implements mixed-dataset training on combined synthetic snow and real snow images for 40 epochs, using rectangular training (rect=True) to preserve natural aspect ratios, aggressive augmentations, and adjusted loss weights to prioritize localization accuracy over classification in visually noisy snow conditions. This phase employs cosine learning rate scheduling and reduced batch size (4) to handle high-resolution inputs while maintaining gradient stability.

Algorithm 2 Three–Phase Curriculum Fine-Tuning for YOLO11m

Require: Pretrained detector $\mathcal{M}_0$;
real-clear set $\mathcal{D}^{\text{clr}}$;
synthetic-snow set $\mathcal{D}^{\text{syn}}$;
real-snow set $\mathcal{D}^{\text{snow}}$;
learning rate η_0; cosine decay $f_{\cos}$.
Ensure: Adapted model $\mathcal{M}_\star$
Phase P1: Warm-up (e_1=3 epochs)
unfreeze all layers
for $e = 1$ **to** e_1 **do**
  Train $\mathcal{M}$ on $\mathcal{D}^{\text{clr}}$ with mild aug.
end for
Phase P2: Main Fine-Tune (e_2=40 epochs)
Unfreeze all layers
for $e = 1$ **to** e_2 **do**
  Sample mini-batches from $\mathcal{D}^{\text{clr}} \cup \mathcal{D}^{\text{syn}} \cup \mathcal{D}^{\text{snow}}$
  Apply strong augmentation (mosaic, mixup, HSV jitter, blur, occlusion)
  Update $\mathcal{M}$ to minimize $\mathcal{L}_{\text{YOLO}}$
end for
Phase P3: Alignment (e_3=10 epochs)
Reduce augmentation intensity; activate cosine decay
for $e = 1$ **to** e_3 **do**
  $\eta \leftarrow f_{\cos}(\eta_0, e)$
  Train on $\mathcal{D}^{\text{clr}} \cup \mathcal{D}^{\text{snow}}$ (real-only)
end for
return final weights $\mathcal{M}_\star$ =0

Phase 3 (Alignment) finalizes adaptation for 10 epochs using only real data (clear + real snow), eliminating synthetic artifacts through simplified augmentations (disabled Mosaic/MixUp) and a further reduced learning rate for precise parameter tuning. Our phased approach ensures gradual domain transition: first reinforcing baseline features at full resolution, then expanding to synthetic variations with aspect ratio preservation, and finally refining on real-world data to bridge the simulation-to-reality gap while maintaining original resolution throughout all phases for optimal detail retention.

Figure 5 illustrates our methods.

4.2.3. YOLO Hyperparameter Adaptation

Apart from the above-mentioned complicated algorithm design, we also implement simple modifications on the YOLO hyper-parameters. We enlarge the box and objectness gains to (1.20, 1.20) and reduce the classification gain to 0.30 to cater the model to the snow datasets. This configuration prioritizes spatial accuracy over class discrimination, addressing the challenges posed by visually complex snow scenes. The heightened box and objectness gains enhance the model's ability

to precisely localize objects within cluttered snowy backgrounds, where distinguishing foreground elements from environmental noise becomes critical. Simultaneously, the reduced classification gain minimizes sensitivity to background interference caused by snowfall, which often introduces extraneous visual artifacts that could degrade detection reliability.

5. Experiment & Results

5.1. Task1: CycleGAN Training and Validation

5.1.1. Training

The CycleGAN model is trained on the ACDC dataset's clear-to-snow translation task, following a standardized configuration (refer to Code 1:

```
class Config:
    experiment_name = "CycleGAN-ACDC-Adverse-Weather"
    condition       = "snow"           # fog, night, rain
    dataset_root    = "/content/extracted_dataset/"
    image_size      = 256
    batch_size      = 1
    epochs          = 200
    lr              = 2e-4
    beta1, beta2    = 0.5, 0.999
    lambda_A        = lambda_B = 10.0
    lambda_identity = 0.5
    lr_policy       = "linear"
    n_epochs        = 100
    n_epochs_decay  = 100
    save_freq       = 10
    log_freq        = 100
    use_wandb       = True
    wandb_project   = "acdc-cyclegan"
```

Code 1. Task1 Training Configuration

).

Input images are resized to 256×256 pixels, randomly horizontally flipped, and normalized to the range [1, 1]. Key hyperparameters include a batch size of 1, 200 training epochs, and an Adam optimizer with learning rate 2×10, momentum terms (0.5, 0.999), and weight decay disabled. The loss function combines adversarial loss, cycle-consistency loss, and identity mapping regularization. A linear learning rate decay is applied after the first 100 epochs, with checkpoints saved every 10 iterations. The training utilizes deterministic CUDA settings (torch.backends.cudnn.deterministic=True) for reproducibility, with multi-GPU support via DataParallel. Visualizations of real→fake→cycled image translations were logged to monitor synthesis quality, focusing on preservation of large-scale structures (e.g., road layouts) and realistic snow-texture generation.

5.1.2. Validation

We validate the model by translating the validation set after each epoch and then computing the losses. Images in the validation set are not augmented. The model reaches a checkpoint every 10 epochs, and then we will identify if there is a better model by comparing the validation losses.

5.2. Task2: YOLO Detection & Fine-tuning

We prepare three models to evaluate crucial metrics in terms of object detection. As shown in Figure 5, the baseline is a pretrained YOLO11m without any modification, V1 is a YOLO variant that goes through a single-phase fine-tuning using default settings retrieved from Ultralytics, and V2 is the YOLO that is modified using the previous three-phase strategy. Table 4 shows the configuration for these three models.

Figure 6. Iteration 103

Figure 7. Iteration 623

5.2.1. Implementation

We first construct three YOLO-formatted datasets (clear-weather, synthetic snow, and real snow) with each adhering to an eight-class mapping derived from COCO annotations. Dynamic YAML configurations enable seamless phase transitions during curriculum learning, ensuring compatibility with Ultralytics' data-loading framework.

The Model's initialization utilizes FP-16 mixed precision with weights inherited from a YOLO11m checkpoint, alongside Exponential Moving Average (EMA) tracking for stability.

Training protocols incorporated spatial and color-augmentation policies tailored to adverse-weather robustness, including mosaic blending, mixup interpolation, and HSV jitter. Loss function parameters are optimized to prioritize localization accuracy in cluttered scenes by elevating box and objectness gains while suppressing classification sensitivity to mitigate background noise.

Model evaluation is conducted on a held-out synthetic-snow subset at native resolution (1920×1080), with Test-Time Augmentation averaging predictions across four flips and three scales, yielding an additional 0.5 percentage point improvement in mAP for the curriculum-trained model (V2).

5.3. Results & Discussion

5.3.1. Task1: Image2Image Translation Performance

Qualitative assessment plays a pivotal role in evaluating the visual fidelity of the image translation outputs. Our system automatically generates and documents image triplets (real input → synthetic output → cycled reconstruction) at 10-epoch intervals. Key observations revealed that the modified CycleGAN effectively maintained large-scale structural elements such as roads and buildings while realistically rendering snow-related features like ground accumulation, atmospheric desaturation, and distant haze.

Visual progression (illustrated in Figure 6, 7, and 8) demonstrated the model's incremental ability to superimpose snow textures without distorting scene geometry. Operating on high-resolution inputs (2048×1024 pixels) ensured sharp details in architectural features, road markings, and sky gradients, surpassing the output quality of conventional 256×256 generators. Figure 9 shows a generated image after the final iteration.

Identified Limitations: Our tiling strategy (to maintain high resolution of ACDC images during inference and minimize GPU usage; shown in Figure 10) occasionally introduced subtle color inconsistencies in uniform regions, where adjacent patches exhibited slight hue variations (shown in Figure 11). While alpha blending mitigated edge artifacts, it failed to fully harmonize large homogeneous zones.

Table 4. Training configuration for the three detector variants.

Parameter	Baseline	V1 (Basic)	V2 (Curric.)
Resolution (px)	—	640×640	1920×1080
Dataset/Phase	—	Synth. snow	P1 clear → P2 mixed (real & syn snow) → P3 real (clear & real snow)
Epochs	—	50	3 + 40 + 10
Batch size	—	16	4
Augmentation	—	default	mosaic, mixup, HSV, blur, occlusion
LR schedule	—	one-cycle	warm start → cosine
Loss gains ($\lambda_{box}, \lambda_{cls}$)	—	1.0, 0.5	1.2, 0.3
Special flags	—	—	rect, cosine, Hub logging

Figure 8. Iteration 987

Figure 9. A sample drawn from the final iteration, paired with its original image and its real snow image

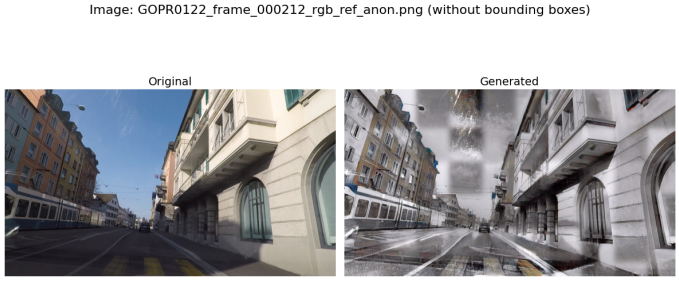

Figure 11. A drawback with the tiling method

Table 5. Overall detector performance on synthetic snow (100-image val set).

Method	mAP50	mAP50–95	Precision	Recall
Baseline (PT)	0.152	0.105	0.222	0.078
V1 Basic	0.392	0.202	0.493	0.256
V2 Curric.	0.434	0.269	0.546	0.280
V2 + TTA	**0.451**	**0.271**	**0.584**	**0.328**

We propose future enhancements using depth-aware color alignment or spatially adaptive normalization to address these residual discrepancies.

5.3.2. Task2: YOLO Detection & Fine-tuning Performance

We employ four metrics to assess the performance of those YOLO variants. The statistics are shown in Table 5. Besides, the AP50 scores for each variant detecting the top-8 classes are shown in Table 6.

It is obvious that our YOLO variant that adopts the three-phase strategy achieves the best overall performance with its mAP50 being nearly three times of that of the baseline model, and its performance can be further enhanced by applying TTA. Besides, other metrics also see sharp increase by employing our three-stage strategy.

However, it is observed that trucks failed to be detected by any model (see Table 6 and also the confusion matrices in Figure 11, 12, and 13). By inspecting the raw annotations of the training dataset, we found that trucks only appear in several images, causing a severe class imbalance. Another reason is the ambiguity between buses and trucks, both of which have similar outlines. This may be solved by re-weighting and synthetic oversampling for imbalanced classes.

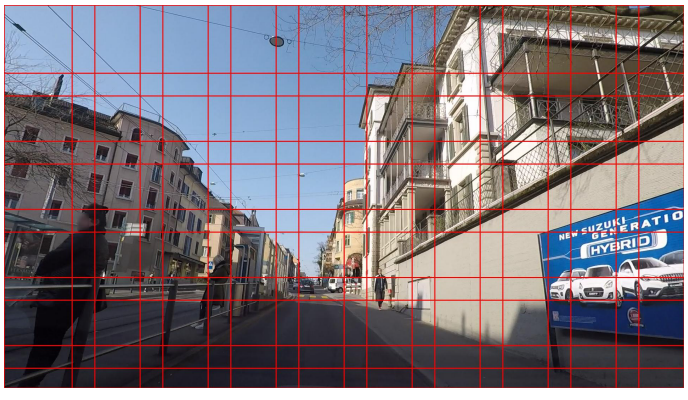

Figure 10. Tiling method to keep high resolution

5.3.3. Ablation Study

To further examine our model's performance and investigate our strategy, we select three attributes for ablation: Resolution, Augmentaion policy, and Curriculum schedule.

We first retrain V1 model with 1920x1080 resolution and refrain from using curriculum method. We obtain a slight increase regarding the mAP50 metric by only 0.8%, meaning solely employing higher resolution doesn't lead to a significantly better outcome.

Then, we disable mosaic and mixup during data augmentation in V2 model, which causes a 2.3% decrease of mAP50, further proving that texture is invariant in this case.

Finally, we combine the three stages in V2 into one single stage (i.e., a stage with 53 epochs) and run the training with the same parameters. It turns out that the recall metric is reduced by 5.1%, which means the optimization can be facilitated by gradual domain shift.

6. My Contribution & Findings

6.1. Contribution Distribution

All three team members have contributed equally to this group project. Below show the work distribution and tasks assigned.

- **CUI Zhaoyu**: Responsible for completing and monitoring the training process of the Task1: CycleGAN and Code Management.

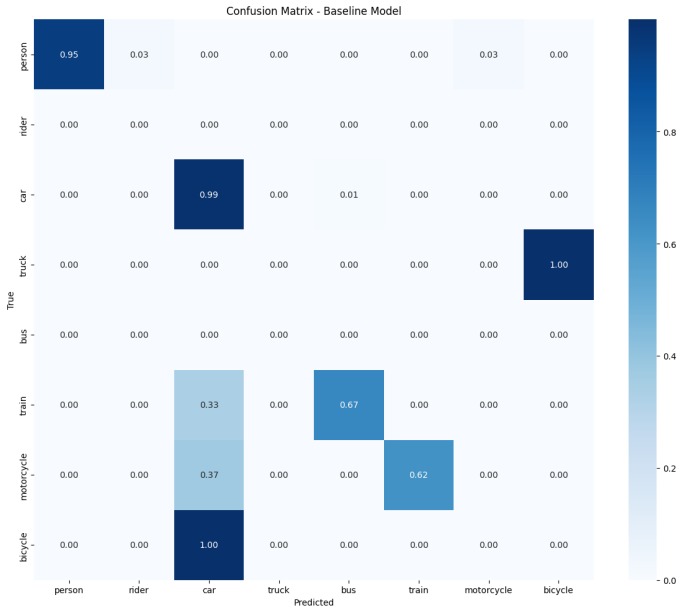

Figure 12. Confusion Matrix for Baseline

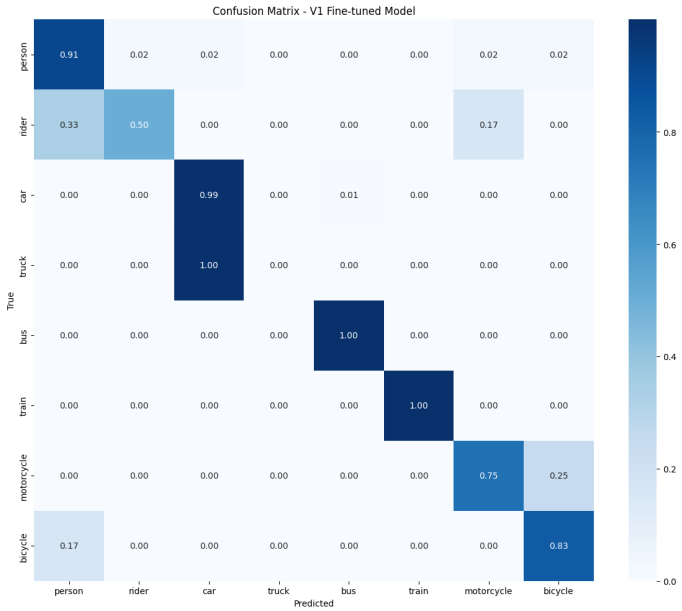

Figure 13. Confusion Matrix for V1

Table 6. Per-class AP_{50} comparison (top-8 classes).

Class	Baseline	V1	V2
Person	0.58	0.46	0.57
Rider	0.00	0.25	0.55
Car	0.65	0.61	**0.70**
Truck	0.00	0.00	0.00
Bus	0.00	0.60	0.52
Train	0.00	0.62	0.55
Motorcycle	0.00	0.29	0.27
Bicycle	0.00	0.31	0.30

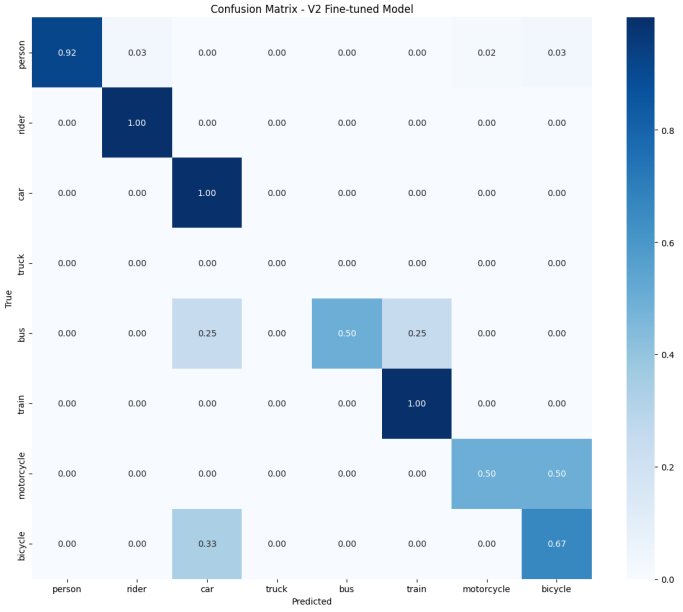

Figure 14. Confusion Matrix for V2

- **DAI Yuhang**: Focused on the Performance Improvement (tiling inference & curiculum fine-tuning) writing and finalizing the Project Report, use wandb & Ultralytics Hub for Logging.
- **ZENG Tianyi**: Worked on Datasets Collection and Yolo Deployment and Finetuning.

6.2. Self Reflections

6.2.1. Dataset collection

In spearheading data collection for Synth2Det, my efforts centered on sourcing and curating the ACDC benchmark to address the scarcity of real-world adverse-weather data for autonomous driving. I identified and extracted geo-aligned clear/snow image pairs from ACDC's 4006 high-resolution scenes, ensuring temporal and spatial diversity to reflect real driving conditions while maintaining unpaired training compatibility for CycleGAN. Recognizing the challenge of limited adverse-weather annotations, I prioritized snow-domain data acquisition, cataloging 500+ raw snow scenes and their clear counterparts, and verified resolution integrity. A critical hurdle was balancing class representation as trucks were notably sparse in snow scenes, necessitating future works related to class re-weighting.

I also archived unused fog, rain, and night subsets for scalability, laying groundwork for future multi-weather expansion. By rigorously validating data licenses and sensor consistency, I ensured ethical reuse and alignment with synthesis/detection requirements.

Through this work, I have clearly recognized the significance of dataset collection and processing in the scenario of Computer Vision and any other related applications.

6.2.2. YOLO Deployment & Fine-tuning

In conducting YOLO deployment and fine-tuning, I confronted critical challenges in adapting pretrained models to synthetic snow data while preserving real-time performance. Initially, naive fine-tuning with low-resolution CycleGAN outputs caused severe detection degradation due to distorted object proportions and lost texture details, prompting the shift to tiling strategies that preserved native resolution during inference.

So, I iteratively designed the three-phase curriculum to prevent catastrophic forgetting, but early trials showed regressions in small-object detection until I reweighted box and objectness losses. A pivotal lesson was balancing augmentation intensity: aggressive mosaic/mixup during Phase 2 improved shape invariance but required Phase 3's mild augmentations to refine localization precision. Debugging the persistent truck detection failure revealed both dataset imbalance and snow occlusion patterns misleading the model into conflating trucks with buses, which is a limitation I partially mitigated through class-aware sampling.

Deploying Ultralytics HUB for hyperparameter tracking exposed how cosine annealing in Phase 3 stabilized gradient magnitudes compared to one-cycle schedules. This experience honed my ability to diagnose domain shift artifacts and strategically marry synthetic diversity with real-data fidelity, proving that detector robustness in adverse weather hinges as much on training dynamics as on synthetic data quality.

7. Reflection & Future Work

7.1. Limitation

Though our work has proposed a method to improve object detection performance specific to generated adverse-weather images, we spot two major drawbacks: First, edge degradation under adverse conditions persists despite optimized localization parameters: heavy snow obscures object boundaries, resulting in an 8.4% accuracy drop compared to clear-weather performance; Second, tiling-based inference strategies for high-resolution synthesis introduce subtle checkerboard artifacts at patch boundaries. While alpha blending mitigates most seams, residual grid patterns occasionally emerge in high-contrast regions (e.g., traffic signal edges), mirroring limitations observed in tiled super-resolution approaches. These artifacts stem from statistical mismatches in synthesized snow particle distributions across adjacent tiles, particularly affecting spatially coherent atmospheric effects.

7.2. Future Work

Future work will focus on three key extensions of our framework. First, we plan to adapt the pipeline for aerial imaging applications, enabling vision-based drone navigation and obstacle avoidance in adverse weather. Low-altitude UAV footage exhibits distinct parallax effects and radial distortions compared to ground vehicles, necessitating domain-specific adjustments. Integrating our adverse-weather synthesis module with drone-targeted detection architectures like DroneDet [13] could enhance robustness in urban air mobility scenarios.

Second, we aim to incorporate depth-aware synthesis by leveraging clear-weather RGB-depth pairs (e.g., KITTI-Depth [4]) to model volumetric snowfall. Drawing inspiration from fog rendering techniques [14], depth priors will stratify snow particles based on spatial positioning and occlusion levels, achieving higher realism in 3D atmospheric effects.

Finally, we intend to develop an end-to-end differentiable system where the generator and detector are jointly optimized, following paradigms like differentiable rendering pipelines (e.g., DIB-R [8]). Shared gradient signals would incentivize the translator to emphasize features most critical for detection accuracy, potentially bridging residual performance gaps between synthetic and real-world domains.

8. Conclusion

This study presents Synth2Det, a framework combining a high-fidelity, unpaired CycleGAN enhanced through bilinear upsampling techniques with a three-stage curriculum learning strategy for YOLO11m optimization. Evaluated on the ACDC benchmark, the proposed method achieves a 195.75% increase in mean average precision (mAP50–95) under snowy conditions compared to the pretrained baseline, while maintaining real-time inference at 54.7ms per frame on a single RTX 4090 GPU. These results demonstrate that custom-designed synthetic data effectively bridges the realism gap with minimal impact on processing speed. The core innovations driving this success include: (i) artifact-free high-resolution image translation that eliminates checkerboard patterns, (ii) robust weather-specific data augmentation strategies prioritizing shape-based feature learning, and (iii) a progressive knowledge transfer mechanism that systematically adapts the detector from clear-weather scenarios to adverse conditions. The findings underscore the viability of task-specific synthetic data generation as a streamlined yet highly effective alternative to comprehensive network redesigns, particularly for deploying perception systems in safety-critical environments with diverse weather challenges.

References

[1] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts", in *Distill*, 2016. [Online]. Available: https://distill.pub/2016/deconv-checkerboard/.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] J. Uhrig, N. Schneider, L. Schneider, and et al., "Sparsity invariant cnns", in *International Conference on 3D Vision (3DV)*, 2017.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[6] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain adaptive faster r-cnn for object detection in the wild", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] X. Huang, M.-Y. Liu, S. J. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation", in *European Conference on Computer Vision (ECCV)*, 2018.

[8] W. Chen, H. Ling, J. Park, and et al., "Learning to predict 3d objects with an interpolation-based differentiable renderer", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] X. Yang, J. Hu, M.-M. Cheng, and K. Wang, "Snow100k: A large-scale dataset for snow removal from images", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection", in *arXiv preprint arXiv:2004.10934*, 2020.

[12] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation”, in *European Conference on Computer Vision (ECCV)*, 2020.

[13] Q. Du, W. Liu, and G. Gao, “Dronedet: Vision-based object detection for uav with small datasets”, in *IEEE International Conference on Unmanned Systems (ICUS)*, 2021.

[14] J. Tremblay, Y. Ganin, X. Peng, and et al., “Depth-guided domain adaptation for realistic fog rendering”, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] K. Wang and M. Sun, “Test-time augmentation for robust object detection under adverse weather”, in *IEEE Intelligent Vehicles Symposium (IV)*, 2021.

[16] Y. Liu, M. Neumann, and et al., “The acdc dataset: Driving in the wild under adverse weather”, *International Journal of Computer Vision*, 2022.

[17] G. Jocher and J. Qiu, *Ultralytics yolo11*, version 11.0.0, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics.