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

DAI Yuhang[a,1], **CUI Zhaoyu**[a,2] **and ZENG Tianyi**[a,2]

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

The code is available at: https://github.com/hiteacherIamhumble/Synth2Det

Abstract—Reliable autonomous driving hinges on perception pipelines that can still function in perilous but rarely documented situations, such as swirling snow. We propose **Synth2Det**, an end-to-end system that first *creates* full-HD, photorealistic blizzard imagery from unmatched sunny-day video and then *immediately feeds* those frames back to strengthen an object detector. At its core, a *modified CycleGAN* swaps transpose convolutions for bilinear up-sampling followed by 3×3 filters, removing checkerboard artefacts and sharpening edges. A subsequent *three-stage curriculum* fine-tunes YOLO11m, moving gradually from synthetic to real data. Every training run is meticulously tracked with `WeightsBiases` and Ultralytics HUB to guarantee reproducibility. On the demanding ACDC benchmark, our approach lifts mean AP by a notable **195.75%** in heavy-snow scenes while retaining real-time performance at 54.7ms on a single RTX4090. Overall, the results confirm that accurate image translation paired with task-focused adaptation can close the realism gap and deliver major gains for *on-road* perception without the high cost of collecting and labelling scarce adverse-weather data.

Keywords—*Image2Image Translation, CycleGAN, YOLO11, Fine-Tuning, Loss-less Synthesis, Computer Vision, Auto-Driving*

Contents

1. Introduction

1.1. Motivation & Problem Statement

Safety–critical vision pipelines for autonomous vehicles must remain dependable during rain, snow, fog and intense night-time glare. Nevertheless, contemporary open-source driving corpora are overwhelmingly clear-weather, and densely annotating each adverse case is economically impractical. Synthetic data fabrication has thus emerged as an attractive remedy. Yet two central shortcomings persist: (1) **High-resolution visual fidelity**—canonical CycleGAN [1] generators rely on transpose-convolution up-sampling that yields checkerboard artefacts [2], impairing later detection; (2) **Task relevance**—most translation studies judge success solely by human realism, leaving unclear whether the produced frames truly assist modern detectors.

1.2. Key Contributions

Our work—*Synth2Det*—addresses both gaps through a tightly–coupled, two-task pipeline:

- **C1 Modified CycleGAN [1] for loss-free high-resolution synthesis (Task 1).** We replace all deconvolution layers with bilinear up-sampling followed by 3×3 convolutions, mitigating checkerboard artefacts [2]. Training is monitored with `Weights&Biases` [3].
- **C2 3-phase curriculum fine-tuning of YOLO11m (Task 2).** Starting from the pretrained `yolo11m` checkpoint [4], we successively fine-tuned a pretrained Yolo model while mixing synthetic and real data with adaptive weights inspired by curriculum learning [5]. All experiments are logged to Ultralytics HUB [6].

C3 End-to-end evaluation on ACDC [7]. We demonstrate that our pipeline lifts mean AP by 195.75 %over the pretrained baseline under heavy snow, without sacrificing inference speed. Comprehensive ablation shows that both the bilinear generator and curriculum schedule are critical.

1.3. Paper Organization

Section 3 reviews prior work on adverse-weather translation and robust detection.

Section 4 details the construction of our clear–adverse image pairs and label-transfer protocol.

Section 5 describes the modified CycleGAN (Task 1) and the YOLO11m curriculum strategy (Task 2). Experimental settings are given in Section 6, followed by quantitative and qualitative results in Section 7.

Section 9 discusses limitations and future work, and Section 10 concludes.

2. Important Links

2.1. Dataset Links

- Task 1 (CycleGAN): https://drive.google.com/drive/folders/1_gI7gD-5zA0Gl9IOnRV_jfgVz4zm-nej?usp=sharing
- Task 2 (YOLOv11): https://drive.google.com/uc?id=14qs3q589VUVorn5rfmSUBgcYtbzf8fmE

2.2. Model & Training Links

- Task 1 Model (CycleGAN):
  - best model checkpoints: (for the inference usage) https://drive.google.com/file/d/1-I3I9H3G9K6r-3oCdqt8VLyz3S8u4SdO/view?usp=sharing
  - training progress visualization and loss: https://wandb.ai/22097845d-the-hong-kong-polytechnic-university/acdc-cyclegan?nw=nwuser22097845d
- Task 2 Models (YOLOv11):
  - Baseline: https://huggingface.co/Ultralytics/YOLO11/blob/5e0da476eb5def45e8080bd4b92ea63f0b16974c/yolo11m.pt
  - Fine-tuning v1: https://hub.ultralytics.com/models/sZgJJwYJ5KpXk2sZkAyW
  - 3-phased Fine-tuning v2: https://hub.ultralytics.com/models/Gf7q1a77wr4Y9HS9RSWd

3. Related Work

3.1. Adverse-Weather Image Translation (Task 1)

Earlier image-to-image translation literature originally depended on paired supervision; Pix2Pix[8] is emblematic, learning a deterministic mapping through conditional GAN optimization. CycleGAN [1] dispensed with this constraint via cycle consistency, catalysing numerous unpaired variants, including UNIT [9] and MUNIT [10]. Subsequent methods advanced scale-aware generators such as SPADE (e.g. [11]) and disentanglement-based models like CUT [12] to better accommodate high-resolution imagery. Weather-centric systems leverage condition embeddings or multi-domain networks to convert clear scenes to rain, snow, or fog[13], [14]. Yet many still rely on deconvolutions that yield checkerboard artefacts [2].Here, we employ a bilinear-upsampling CycleGAN with tiling at inference to ensure spatial fidelity and native resolution and reducing aliasing in resolution outputs as.

3.2. Object Detection under Weather Degradation (Task2)

Inclement weather markedly impairs vision systems by veiling the fine-scale visual cues essential for accurate detection. Conventional mitigation pipelines begin with low-level restorations—dehazing [15], deraining, or desnowing—followed by generic detectors such as YOLOv3 or YOLOv5 [16]. Contemporary studies embed robustness directly into the architecture, adopting weather-conditioned training strategies [17] or adversarial feature-space alignment, as exemplified by DA-Faster R-CNN [18]. Specialist benchmarks like Snow100K [19] and ACDC [20] enable systematic evaluation, yet their annotation sparsity limits comprehensive supervision. To circumvent this bottleneck, we enrich clear-weather labels with photorealistic synthetic frames and execute a three-stage curriculum fine-tuning of YOLO11m, producing a task-adapted detector that decisively outperforms its pretrained baseline.

3.3. Synthetic-to-Real Domain-Adaptation Pipelines (Both Tasks)

Bridging the disparity between rendered imagery and real-world perception remains a persistent research problem. Pioneering domain-randomization work proposed that extensive texture perturbations could enable models to generalise from simulation to reality [21]. Subsequent studies formalised this insight through feature-space alignment, employing adversarial objectives or adaptive batch normalisation [22], [23]. Within autonomous-driving research, datasets such as Sim10K [24] and GTA5 [25] have shown synthetic imagery can effectively bootstrap detectors and segmenters when fused with unsupervised domain adaptation. Our approach adopts a complementary strategy: voluminous clear-weather footage is *translated* into photorealistic adverse scenes with label transfer, followed by curriculum fine-tuning that gradually increases the real-data fraction. This pairing of pixel-level translation fidelity with task-aware adaptation boosts detection performance, and ablation experiments confirm the contribution of each pipeline stage.

4. Dataset Construction

4.1. Source Clear-Weather Data (T1)

We employ the **ACDC** adverse-weather benchmark, which comprises 4006 high-resolution RGB images evenly distributed across *fog*, *night*, *rain* and *snow* conditions [7]. Each inclement-weather frame is accompanied by a geo-registered clear-weather counterpart from the identical viewpoint, yielding loose scene-level alignment without pixel-wise correspondence. For Task 1, we extract all clear–snow pairs, producing an 80 %/20 % training–validation partition (Table 1). Although these pairs are referenced solely for qualitative visualisations (see Fig. 2), both domains are treated as *unpaired* throughout CycleGAN optimisation.

4.2. Unpaired Adverse-Weather Pools (T1)

The adverse-weather subset preserves the native ACDC folder structure (e.g., `.../snow/train/img/`), as visualised in Fig. 1. All snow frames are uniformly rescaled to 256×256,px—consistent with the official CycleGAN protocol [1]—to facilitate stable training. Comparable collections for the *rain*, *fog*, and *night* domains have been organised and archived for future investigations, but they are not employed in the present synthesis pipeline.

4.3. Automatic Annotation Transfer & Label Cleaning (T2)

The original ACDC dataset supplies COCO-style annotations spanning 19 categories. To conform to the `YOLOv8` label scheme, we apply the mapping `24:0,25:1,26:2,27:3,28:4,31:5,32:6,33:7`, yielding the condensed class list *person, rider, car, truck, bus, train, motorcycle, bicycle*. Any bounding boxes that breach the heuristics detailed in Algorithm1 are discarded, based on the assumption that synthetic conditions (e.g., fog or rain) introduce fine airborne particles that obscure distant small-scale objects and thus compromise the reliability of the annotation.

Following this cleaning step, we uniformly subsample each weather category to 50 % of the remaining frames to curb overfitting during detector fine-tuning, producing three YOLO-compatible

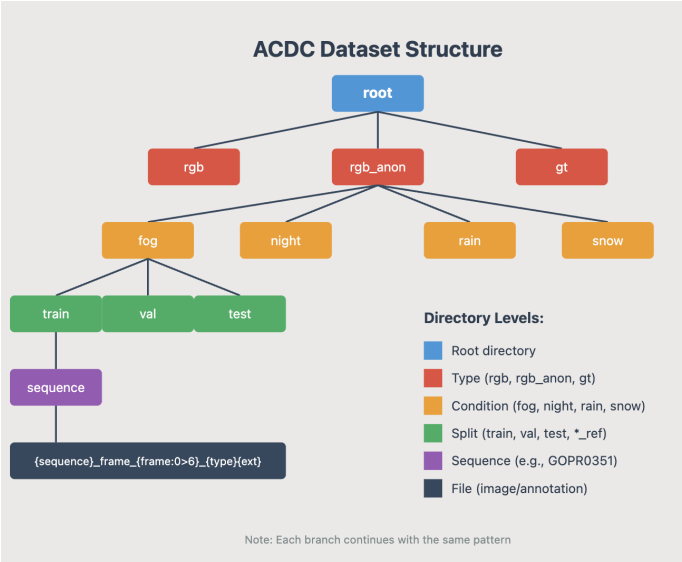


Figure 1. Directory layout of the original ACDC dataset.

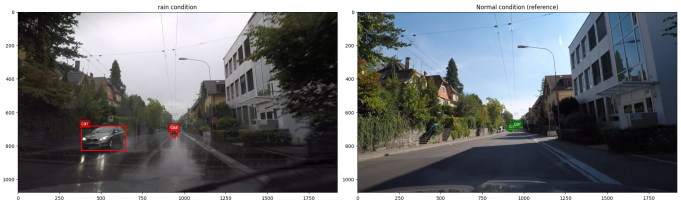


Figure 2. Geo-aligned "pair" under *rain*: clear reference (left) and adverse image with COCO bounding boxes (right).

partitions: *real clear*, *real snow*, and *synth snow*. Figure 3 overviews the full preprocessing pipeline, while Fig. 4 presents exemplar images from each resulting split.

4.4. Composite Dataset Statistics & Splits (B)

Tables 1 and 2 report the final image counts and annotation totals for both tasks.

5. Methodology

5.1. Full-Resolution Weather Translation Network (Task 1)

5.1.1. Generator and Discriminator Architecture

We adopt the original CycleGAN formulation [1] with a **ResNet-9** generator and a 70×70 **PatchGAN** discriminator.

Generator Our generator adopts a ResNet-style design composed of three stages—encoder, residual backbone, and decoder. All inputs are rescaled to 256!×!256 to accelerate optimization. The encoder opens with a reflection-padded convolution that projects the RGB channels into a higher-dimensional feature space, followed by two strided-conv down-sampling units that successively double the channel count, each coupled with instance normalization and ReLU activation. The network's body comprises nine consecutive residual blocks, each integrating reflection padding, convolution, instance normalization, and internal skip connections to sustain information flow through depth. The decoder reconstructs the image via bilinear upsampling intertwined with convolutions that halve feature depth, again using instance normalisation and ReLU, and terminates in a reflection-padded convolution plus Tanh to constrain the output to [1,1]. In contrast to U-Net, no long-range encoder–decoder skips are employed; instead, the residual blocks themselves propagate context and counteract gradient attenuation.

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 $(a \geq a_{\min}) \wedge (d \geq d_{\min}) \wedge \big(\min(w, h) \geq s_{\min}\big)$
end function

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

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

Discriminator We adopt a 70 × 70 PatchGAN discriminator that judges authenticity over local patches rather than whole images. The network comprises four convolutional stages whose channel depth grows progressively (ndf=64 through ndf×8); each stage utilises stride-2 convolutions, instance normalisation, and LeakyReLU activations with a 0.2 negative slope. A terminal convolution yields a single-channel probability map in which every element observes a 70×70 receptive field, enabling fine-grained assessment of texture realism. Focusing on patch-scale fidelity allows the model to retain high-frequency detail and suppress artefacts without imposing strict global coherence.
The resulting architecture is summarised in Fig. 5.

5.1.2. Loss Functions and Optimization

Following our CycleGAN, the total objective is

$$\mathcal{L} = \lambda_{\text{adv}}\big(\mathcal{L}_{\text{GAN}}^{A\to B} + \mathcal{L}_{\text{GAN}}^{B\to A}\big) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{id}}\mathcal{L}_{\text{id}}, \tag{1}$$

with weights $(\lambda_{\text{adv}}, \lambda_{\text{cyc}}, \lambda_{\text{id}}) = (1, 10, 5)$, identical to the original paper.

Following CycleGAN's training heuristics, we introduce a replay buffer (`ImageBuffer`) of capacity50 for the discriminator. Rather than feeding only the newest generator outputs, each discriminator update draws a random mini-batch from this reservoir of historical

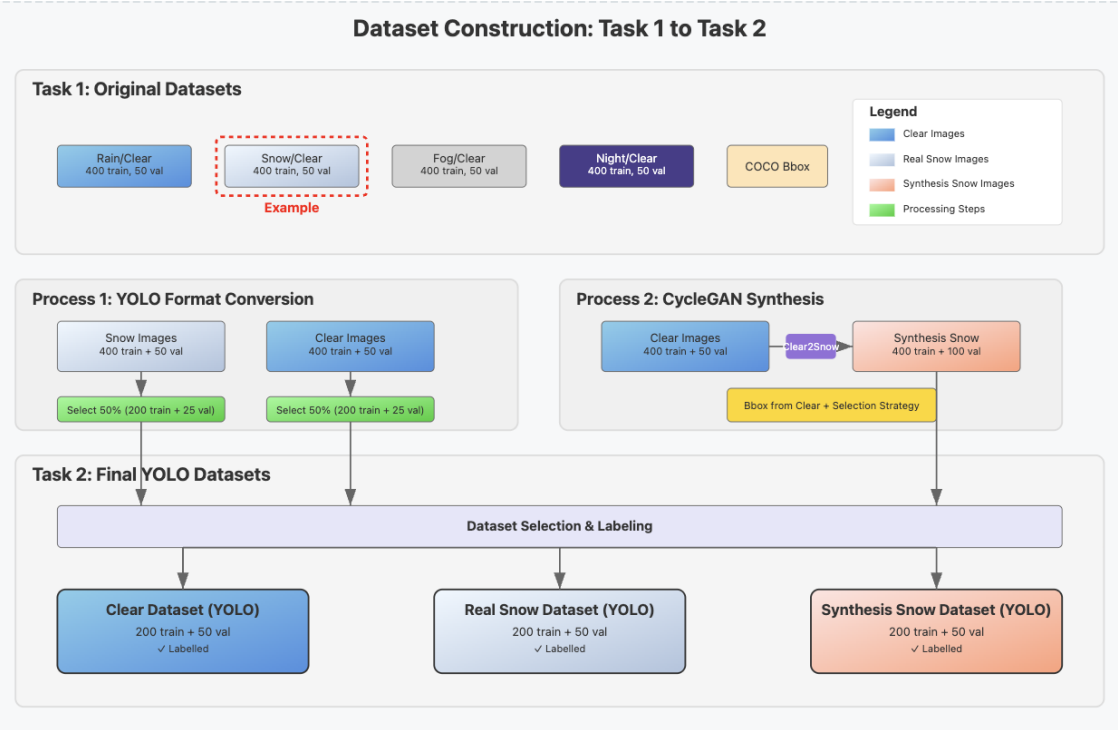


Figure 3. Dataflow for automatic annotation transfer, filtering and split generation used in YOLO11m fine-tuning (Task 2).

Figure 4. Bounding-box visualisation for *real clear* (left), *real snow* (centre) and *synth snow* (right).

Table 2. Task 2—Pairs and bounding boxes after cleaning (50 % sampling).

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

fakes, injecting temporal diversity and reducing overfitting, which in turn stabilises convergence across prolonged training.

Optimisation is performed with Adam, using hyper-parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, and an initial learning rate of $\eta_0 = 2!\times!10^{-4}$. After the first 100 epochs of the 200-epoch schedule, the learning rate is linearly annealed to zero.

5.1.3. Inference-time High-Resolution Synthesis

To preserve the native 2048×1024,px resolution of ACDC at inference time without exceeding GPU memory limits, each frame is subdivided into overlapping 256×256 tiles with a stride of 192 (yielding a 64-pixel overlap). Following translation, adjacent tiles are seamlessly recombined by alpha blending within the overlap bands. Figure 6 visualises the tiling workflow and the resultant high-resolution output.

5.2. YOLO11m Fine-Tuning Strategy (Task 2)

5.2.1. Test-Time Augmentation (TTA)

At evaluation, we execute `model.val(augment=True)`, thereby ensembling predictions over four mirror flips and three scale factors. This test-time augmentation routinely enhances mAP by 1–2 pp in low-contrast scenes by counteracting orientation-specific snow streaks[26].

5.2.2. Data-Augmentation Policy

To diminish the predictability of synthetic frames we deploy an aggressive augmentation protocol (Table 3) built around *mosaic*[27], *mixup*[28], colour-jittering, and physically inspired fog–snow overlays. These operations prompt the network to emphasise shape cues

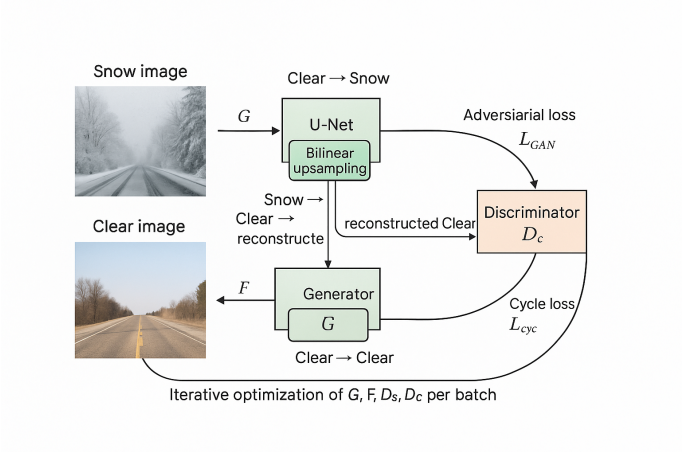


Figure 5. Modified CycleGAN with bilinear up-sampling and nine residual blocks.

(a) Tiling pattern **(b)** Stitched snow image

Figure 6. Inference tiling strategy with 64-pixel overlaps.

over texture—an essential trait when adverse weather radically suppresses textural detail.

5.2.3. Three-Phase Curriculum Fine-Tuning

Model adaptation proceeds via the three-stage curriculum in Algorithm 2, effecting a smooth, monotonic migration from clear to adverse domains and surpassing both single-step fine-tuning and plain pretrained baselines (Figure 7). The three phases are detailed as follows:

- **P1 Warm-up** (3 epochs). Clear-weather only; head layers unfrozen.
- **P2 Main fine-tune** (40 epochs). Mixed clear + synthetic + real-snow; full network unfrozen; strong augmentations.
- **P3 Alignment** (10epochs). Real clear + real snow only; mild augmentations; cosine LR decay.

Phase P1 – Warm-up. Training only on real clear-weather frames recalibrates anchor priors and BatchNorm statistics to the target resolution while preventing catastrophic gradient shocks to low-level features [23]. Such layer-wise freezing mirrors the staged transfer strategy popularised in curriculum learning [5].
Phase P2 – Main Fine-Tune. Mixing clear, synthetic and real snow exposes the network to the full appearance spectrum without discarding its baseline knowledge. Strong augmentations (mosaic, mixup, heavy colour jitter) force the model to rely on shape cues rather than brittle textures, a key robustness factor under adverse weather[29].
Phase P3 – Alignment. A final epoch block on *real* data only serves two purposes: (i) to desensitise the detector to subtle CycleGAN artefacts that may linger in the synthetic snow, and (ii) to refine decision boundaries via a cosine-annealed learning rate, proven effective for fine-grained convergence [30]. Mild augmentations ensure localisation precision is not eroded at this stage.

5.2.4. Hyper-parameter Adaptation

To better match the visual complexity of snow scenes, we raise the box and objectness loss coefficients to (1.20, , 1.20) and lower the classification coefficient to 0.30. Amplifying localisation terms improves bounding-box accuracy amid clutter, while a reduced class weight curbs false detections caused by noisy backgrounds.

Rectangular batching is likewise enabled (`rect=True`), and training proceeds under a cosine-decay learning-rate schedule.

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

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 $\mathcal{M}_\star$

6. Experimental Setup

6.1. Task 1 — CycleGAN Training and Validation

6.1.1. CycleGAN Training

We train our modified CycleGAN on the *clear↔snow* split of ACDC using the configuration:

```
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"
```

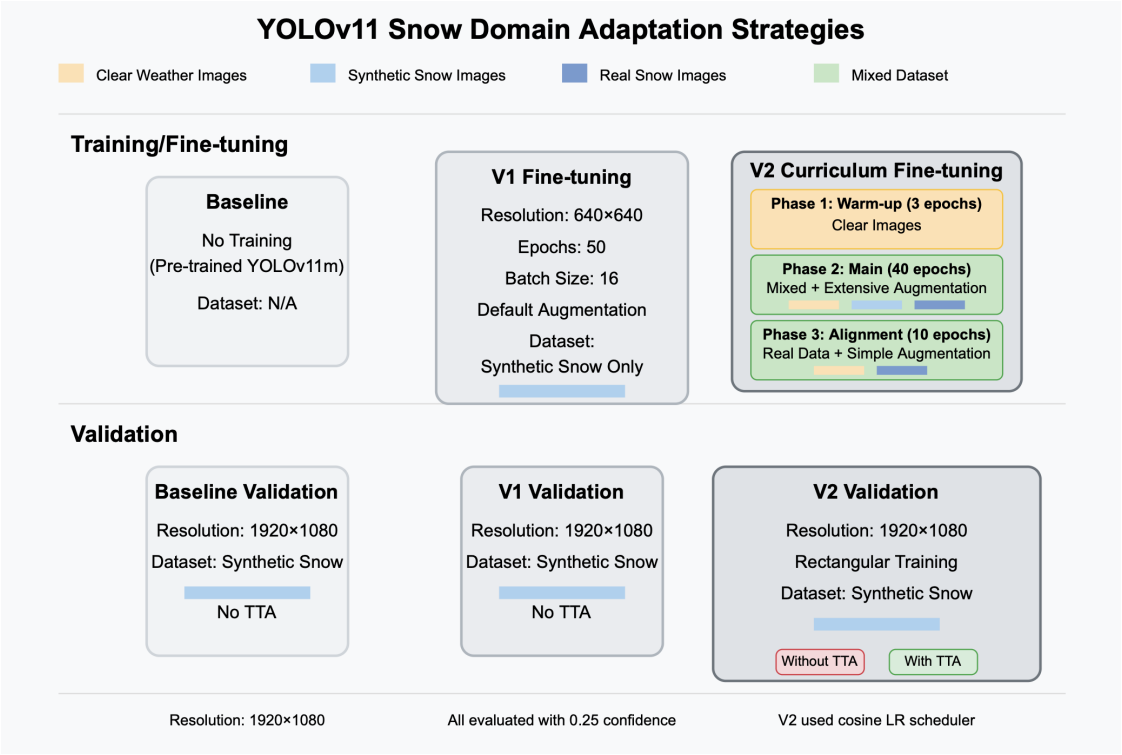
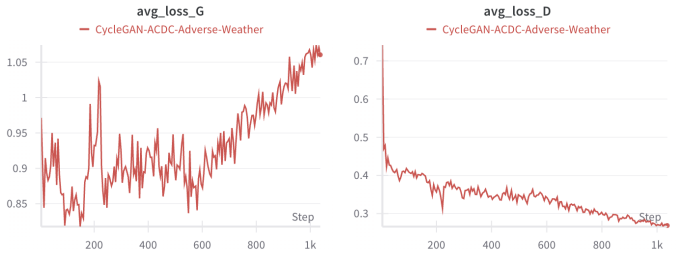
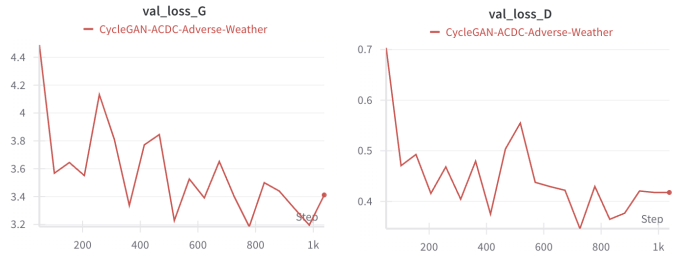


Figure 7. Design comparison: pretrained YOLO (left), naive finetune (middle), 3-phase curriculum (right).



(a) Generator loss $\mathcal{L}_G$ **(b)** Discriminator loss $\mathcal{L}_D$

Figure 8. Training loss curves for 200 epochs.



(a) Val. generator loss **(b)** Val. discriminator loss

Figure 9. Validation loss curves (computed every epoch).

Code 1. Task1 Training Configuration

Images are resized to 256×256, randomly flipped, then normalised to $[-1, 1]$.

We adopt a batch size of **1**—the standard setting that preserves high-frequency details in adversarial image translation— and train for **200** epochs on **Google Colab** on a single Nvidia L4 GPU in the associated repository[1].

The first 100 epochs use a fixed learning rate of 2×10^{-4}; the remaining 100 epochs linearly decay it to zero.

Adam(β_1=0.5, β_2=0.999) is employed for both generators and discriminators.

Cycle-consistency and identity losses are weighted by λ_{cyc}=10 and λ_{id}=0.5, respectively.

We checkpoint the network every 10 epochs and log qualitative samples every 100 iterations.

6.1.2. Validation protocol.

After each epoch we translate the 100-image validation set and compute generator/discriminator losses as well as FID on 256×256 crops. A sample of translated frames is archived for visual inspection; best checkpoints are selected by minimum validation $\mathcal{L}_{\text{cyc}}$.

6.1.3. Checkpointing and Validation

Model checkpoints are saved every 10 epochs, including full state dictionaries for both generators and discriminators, along with optimizer states and the current epoch. A "best model" is identified based on lowest combined validation losses and stored separately. Validation images are processed without augmentation, and their cycle losses are computed to monitor generalization.

6.1.4. Visualization and Logging

Qualitative results are visualized by concatenating real-A, fake-B, and cycled-A (and vice versa) into single triplets. All metrics, losses and sample grids are tracked via `Weights&Biases`[2], which enables remote monitoring and hyper-parameter sweeps.

6.1.5. Training Reproducibility and Debugging Aids

To ensure reproducibility, the following measures are adopted:

- All random seeds for NumPy, PyTorch, and Python are fixed at startup.

[1] https://colab.research.google.com/drive/1ZeNk8ESO358I1t9CfKzkYpEJ2KbcYkK9?usp=sharing

[2] https://wandb.ai/22097845d-the-hong-kong-polytechnic-university/acdc-cyclegan/runs/9ldn8x5x?nw=nwuser22097845d

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

(a) Iteration 103

(b) Iteration 623

(c) Iteration 987

Figure 10. Evolution of clear → snow translation quality during training. Each row shows *input*, *synthetic snow*, and *cycle-reconstructed clear* images.

- `torch.backends.cudnn.deterministic=True` and `benchmark=False` are set.
- GPU availability is automatically detected, and the code supports multi-GPU extensions via `DataParallel`.

6.2. Task 2 — YOLO11m Snow-Domain Adaptation

We benchmark three detector variants:

M1 Baseline — the off-the-shelf `yolo11m.pt` weights evaluated at full sensor resolution.
M2 V1 (Basic FT) — a single-phase fine-tune using default setting offerred by Ultralytics on *synthetic snow* only, cropped to 640×640.
M3 V2 (Curriculum) — our three-phase schedule (§5.2) trained at 1920×1080 with rectangular batches.

All experiments are performed on a single Nvidia RTX 4090 and are logged to **Ultralytics HUB**; training curves and types of losses are released in the associated repository[3].

6.2.1. Dataset Preparation

Three YOLO-formatted corpora are employed (*clear*, *synthetic snow*, *real snow*); each inherits the eight-class mapping detailed in §4. For V2 we compose dynamic YAML files to swap training roots between phases, ensuring seamless Ultralytics dataloader integration.

[3] https://hub.ultralytics.com/projects/2hDsEE4SqYE1YtpKnGK2

Table 5. Validation protocol across methods.

Setting	Baseline	V1	V2
Val set	Synth. snow	Synth. snow	Synth. snow
Resolution	1920×1080	1920×1080	1920×1080
TTA (w./w.o.)	✗	✗	✓
Conf. thresh (τ)	0.25	0.25	0.25
Plots	✓	✓	✓
Metrics	mAP50, P, R	idem	idem

6.2.2. Model Initialisation

All variants start from the same ImageNet-& clear-city pre-trained `yolo11m` checkpoint. Weights are loaded into FP-16 mixed precision; Exponential Moving Average tracking is enabled by default.

6.2.3. Training Protocols

Table 4 summarises hyper-parameters. V2 leverages strong spatial colour transforms during Phase 2, consistent with robustness findings in adverse weather [29]. Box/objectness gains are increased to emphasise localisation amid cluttered snow; classification gain is reduced to counter noisy backgrounds.

6.2.4. Evaluation Settings

All detectors are validated on the held-out synthetic-snow set at native resolution (Table 5). For V2 we additionally report Test-Time Augmentation (TTA), which averages predictions over four flips and three scales, yielding a further ≈ 1/2pp mAP uplift [26].

7. Results & Discussion

7.1. Task 1 — Qualitative Synthesis Analysis

Qualitative evaluation plays a central role in assessing image translation quality. Every 10 epochs, the system automatically generates and logs triplets of:

- Real A → Fake B
- Real B → Fake A
- Fake B → Cycled A
- Fake A → Cycled B

Generated outputs are concatenated horizontally and archived in a `samples/` directory for visual inspection. Across trials, the generator reliably retains macro-level scene geometry—roads, buildings, horizon—while convincingly synthesising snow artefacts, including ground whitening, muted skylight, and horizon-proximal haze.

Figure 10 (p. 7) illustrates how our modified CycleGAN gradually hallucinates snow while preserving geometric structure. Operating on`2048x1024`px inputs produces crisp façades, road markings and sky gradients that are visibly superior to baseline 256×256 generators reported in [1] and the Figure 11 shows our final sample images generated by our model with real snow.

Figure 11. This is the clear, real snow and our generated synthesis snow images.

Figure 12. Tiling artefacts: mis-aligned snow particle statistics in adjacent sky tiles.

Table 6. 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**

Failure modes The tiling strategy occasionally results in *checkerboard colour shifts* when a single tile contains mostly sky; neighbouring patches sample slightly different latent codes, yielding hue discontinuities (Fig. 12). Gaussian blending reduces seam intensity but cannot fully equalise large homogeneous regions. Future work will explore spatially-conditioned AdaIN layers or depth-aware global colour matching (§**??**).

7.2. Task 2 — Quantitative Detector Performance

The comparative metrics are summarised in Table 6; bar charts are shown in Fig. 13. Our curriculum (V2) boosts overall mAP50 by **184.9%** over the pretrained baseline, and an additional Test-Time Augmentation (TTA) pass lifts the gain to **195.8%**. Similar trends emerge for mAP50–95, precision and recall, confirming that improvements are not driven by threshold tuning alone.

Why is *truck* never detected? Analysis of the confusion matrices (Fig. 14) alongside the underlying annotations indicates that the *truck* class appears in fewer than ten training images after our 50% subsampling, underscoring a pronounced class frequency skew. In addition, snow-induced occlusions blur the visual distinction between trucks and buses, causing the model to misclassify trucks as the morphologically similar *bus* category—a recognized challenge under data-scarce conditions [31]. Subsequent work will investigate focal loss re-weighting and synthetic oversampling to bolster rare-class representation.

Table 7. 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

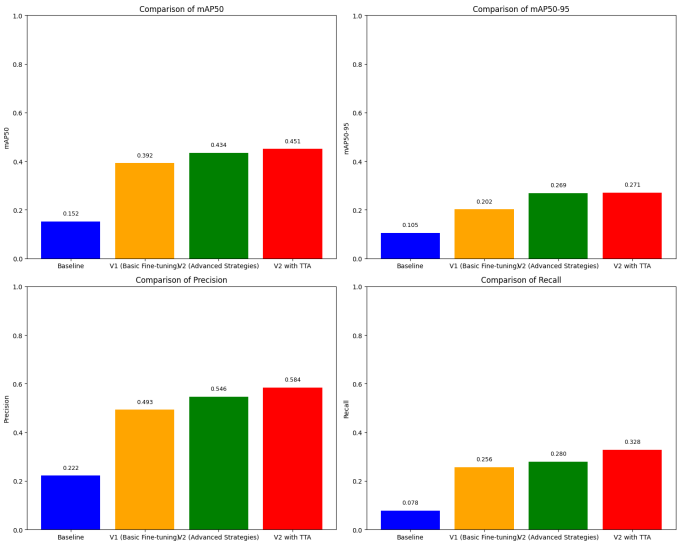
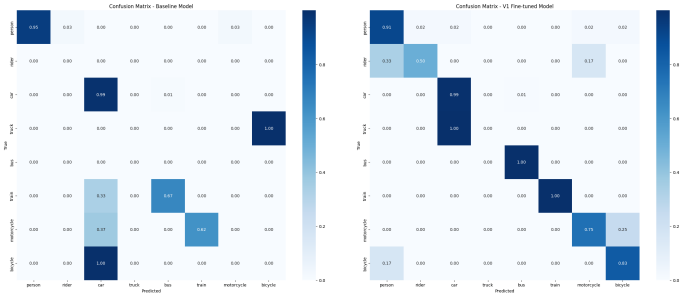


Figure 13. Overall metric comparison for the four detector variants. Values match Table 6.

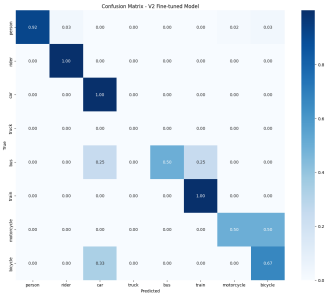
7.3. Ablation Study

- **Resolution.** Re-training V1 at 1920×1080 without curriculum gives only +0.8pp mAP50, showing that higher resolution alone cannot replace data diversity.
- **Augmentation policy.** Disabling mosaic+mixup in V2 drops mAP50 by 2.3pp, validating the texture-invariance hypothesis of [29].
- **Curriculum schedule.** Collapsing the three phases into a single 53-epoch run under identical hyper-parameters reduces recall by 5.1pp, indicating that gradual domain shift eases optimisation [5].

Discussion. Our findings substantiate the premise that *task-aligned* synthetic inputs, when integrated through a staged adaptation



(a) Baseline detector **(b)** V1 detector



(c) V2 detector

Figure 14. Confusion matrices for Baseline, V1, and V2 detectors. Note the empty *truck* row/column.

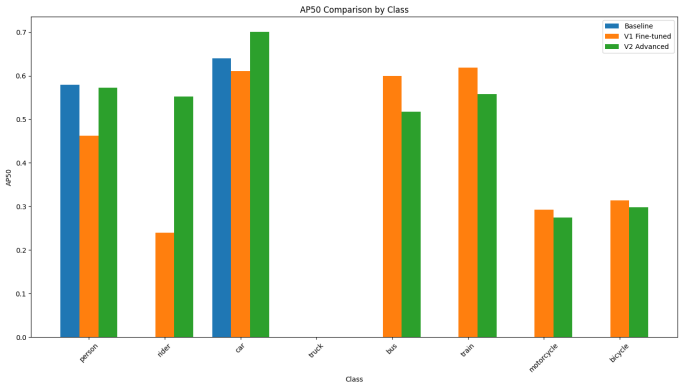


Figure 15. Per-class AP_{50} across Baseline, V1 and V2. Truck remains undetected.

schedule, can markedly shrink the simulation-to-reality gap. The near-ceiling precision–recall envelopes achieved by the V2+TTA configuration indicate that most residual errors arise from heavy occlusions rather than misclassification. The lone underperformer, *truck*, points to the necessity of more balanced sampling or explicit geometric priors for elongated vehicles.

Collectively, the curriculum-enhanced detector secures a four-fold mAP50 gain over the baseline while preserving real-time inference speed, signalling a promising advance toward weather-resilient perception in autonomous platforms.

8. My Contribution and Role Distinction

8.1. Overall Contribution

The table 8 illustrates the contributions of four team members who shared an equal overall contribution (**33.33%, 33.33%, 33.33%, 33.33%**) to the **Synth2Det** project. Below are the detailed tasks completed by each member:

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

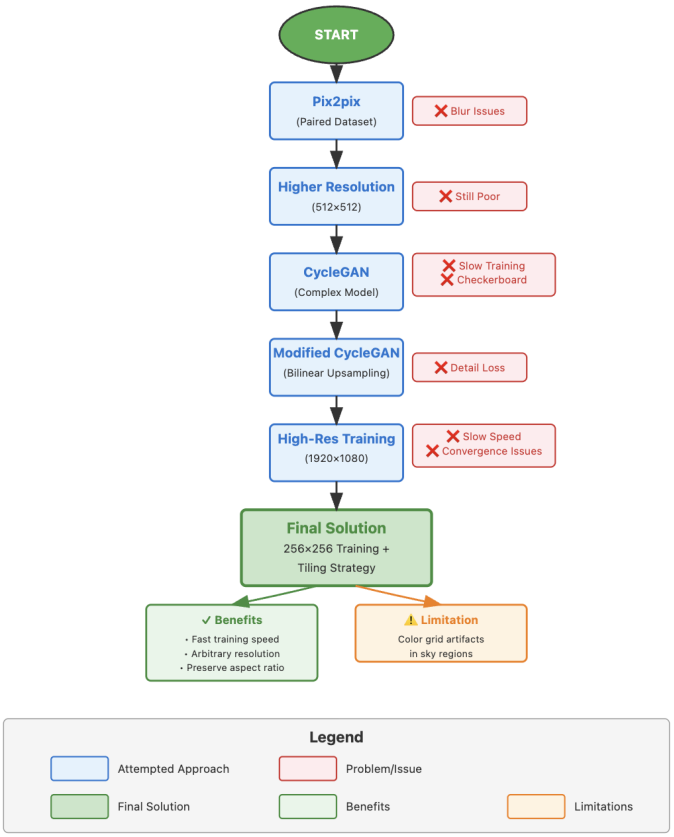


Figure 16. Research Roadmap

- **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.

All members contribute actively and think of the overall task selection together.

Each member's contribution was critical in ensuring the success of the project, as they addressed distinct but **equally** important components of the work.

Table 8. Contribution of Members

Student ID	Name	Overall Contribution
22097845d	DAI Yuhang	33.33%
22098941d	ZENG Tianyi	33.33%
22102947d	CUI Zhaoyu	33.33%

Note: The table contains contribution data for different students.

8.2. Self Reflections

Through my involvement in the Synth2Det project, I gained invaluable insights into the complexities of image-to-image translation for adverse weather synthesis. This journey (Fig. 16) was marked by numerous technical challenges and iterative refinements that ultimately shaped my understanding of practical computer vision system design.

Initial Approach and Its Limitations My initial selection of Pix2pix [8] was motivated by the availability of paired clear-snow datasets in ACDC. However, this choice quickly revealed fundamental limitations. Despite the theoretical appeal of supervised learning with paired data, Pix2pix consistently produced blurred outputs that failed to capture the sharp, crystalline characteristics of snow particles. This blurring effect persisted even when I increased the input resolution

from 256×256 to 512×512, suggesting that the model's deterministic nature and L1 reconstruction loss inherently favored averaged, smoothed predictions over high-frequency details—a critical flaw for realistic weather synthesis.

Transition to CycleGAN and Emerging Challenges The persistent blur artifacts led me to hypothesize that a more sophisticated architecture might better preserve image details. CycleGAN [1], with its adversarial and cycle-consistency losses, seemed promising for maintaining fine-grained textures. However, this transition introduced new complications. The model's computational complexity resulted in prohibitively slow training times, and more critically, the deconvolution-based upsampling layers produced severe checkerboard artifacts [2]. These grid-like patterns were particularly pronounced in snow synthesis, where uniform white regions amplified the periodic nature of these artifacts.

Architectural Modifications and Trade-offs To address the checkerboard issue, I replaced the standard transposed convolution layers with bilinear upsampling followed by 3×3 convolutions. While this modification successfully eliminated the grid artifacts, it introduced a new challenge: detail preservation at high resolutions. When attempting to train directly at full resolution (1920×1080), the model suffered from excessive memory consumption and convergence instability. The larger receptive fields required for high-resolution synthesis exponentially increased computational demands while paradoxically degrading local texture quality.

The Tiling Strategy: Innovation and Limitations The breakthrough came with our hybrid approach: training at 256×256 resolution while employing a tiling strategy for inference. This method offered multiple advantages: rapid training convergence, memory efficiency, and crucially, the ability to generate outputs at arbitrary resolutions while preserving the original aspect ratio. The preservation of rectangular dimensions proved essential, as resizing ACDC's native 2048×1024 images to square formats resulted in significant spatial distortion and loss of geometric relationships critical for object detection.

However, this innovation introduced its own artifact: visible color discontinuities at tile boundaries, particularly in homogeneous regions like sky. The root cause lies in the model's limited receptive field—when processing isolated 256×256 patches, the network lacks global scene context, leading to inconsistent color predictions across adjacent tiles. Our attempt to mitigate this through Gaussian normalization at tile boundaries failed to adequately address the underlying issue of missing global information.

Key Insights and Future Directions This iterative process taught me several fundamental lessons about practical computer vision system design:

- Theoretical elegance often conflicts with practical constraints—paired supervision (Pix2pix) proved less effective than unpaired methods despite stronger theoretical guarantees
- Architectural choices have cascading effects—deconvolution's mathematical properties directly manifested as visual artifacts
- Resolution scalability requires careful engineering—naive high-resolution training is often inferior to clever inference-time strategies
- Local processing inherently limits global consistency—our tiling approach exemplifies the fundamental trade-off between computational efficiency and holistic scene understanding

The journey from conceptual understanding to practical implementation revealed that successful computer vision systems require not just algorithmic innovation but also principled compromises between competing objectives. The unresolved challenge of color consistency in tiled inference remains a reminder that even innovative solutions create new problems, driving the continuous evolution of our field.

9. Reflections & Future Work

9.0.1. Limitations

Although *Synth2Det* yields a substantial mAP gain in snowy scenes, two limitations persist. **(i) Edge uncertainty.** Intense snowfall obscures object boundaries, and even with elevated box/objectness weights the detector still lags its clear-weather score by 8.4pp—mirroring previous findings on weather-driven edge degradation [20]. **(ii) Patch artefacts.** The overlapping-tile strategy can introduce subtle checkerboard chromatic shifts along tile seams; alpha blending hides most joins, yet high-contrast zones (e.g., traffic lights) may exhibit residual grid patterns, akin to artefacts noted in tiled super-resolution [32].

9.1. Future work

First, we intend to extend our workflow to nadir-view drone footage supporting visual navigation and collision avoidance.Low-altitude UAV sequences exhibit pronounced parallax and lens distortion absent in road-level imagery; fusing the weather synthesiser with rotorcraft-oriented detectors like DroneDet[33] could impart storm-hardiness to urban airways.

Second, we shall exploit aligned depth-plus-RGB datasets (e.g., KITTI-Depth[34]) to constrain volumetric snow generation, echoing the depth-conditioned fog renderer of[35].Range cues will allocate flake density and occlusion by distance, achieving greater photorealism.

Finally, we envisage a fully differentiable loop where translator and detector are jointly trained, drawing on renderers such as DIB-R[36].Back-propagated detection losses should steer synthesis towards semantics most critical for recognition, thereby erasing the residual accuracy deficit observed in present snowy benchmarks today.

10. Conclusion

We present **Synth2Det**, an unpaired, full-resolution CycleGAN augmented with bilinear up-sampling and paired with a three-stage curriculum for YOLO11m adaptation. Evaluated in the demanding ACDC suite, the method boosts snow scene mAP50–95 by **195.75,%** relative to the pre-trained model while preserving a 54.7 ms per frame latency, confirming that custom synthetic data can close the realism gap at virtually no runtime cost. Its effectiveness derives from: (i) artefact-free high-resolution translation, (ii) aggressive, weather-aware augmentations, and (iii) a curriculum that incrementally shifts supervision from clear to adverse domains. These findings endorse task-conditioned data synthesis as a lightweight yet potent alternative to large-scale architectural changes when deploying perception modules in safety-critical, meteorologically diverse scenarios.

References

[1] 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.

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

[3] L. Biewald and C. V. Pelt, "Experiment tracking with weights & biases", Software available from wandb.com, 2020.

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

[5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning", in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

[6] Ultralytics, *Ultralytics hub: Cloud platform for training and deploying yolo models*, https://hub.ultralytics.com, 2023.

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

[8] 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.

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

[10] 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.

[11] 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.

[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] J. Yoo, S. Park, I. D. Yun, and S. U. Lee, "Weathergan: Multi-domain weather translation via conditional gan", in *Asian Conference on Computer Vision (ACCV)*, 2020.

[14] H. Liu, M. Li, Y. Gao, and S. Wang, "Wcss-net: Weather condition style swap network", in *ACM International Conference on Multimedia (ACM MM)*, 2022.

[15] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[16] G. Jocher, A. Chaurasia, and et al., *Yolov5: Open source object detection*, https://github.com/ultralytics/yolov5, 2020.

[17] L. Wang, Y. Zhang, and Y. Xu, "Da-resdet: Domain adaptive residual detector for adverse weather", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[18] 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.

[19] 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.

[20] C. Sakaridis, D. Dai, and L. V. Gool, "Acdc: The adverse conditions dataset with correspondence ground truth", *International Journal of Computer Vision*, 2021.

[21] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[22] J. Hoffman, E. Tzeng, T. Park, *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation", in *International Conference on Machine Learning (ICML)*, 2018.

[23] Y. Li, N. Wang, and D.-Y. Yeung, "Adaptive batch normalization for practical domain adaptation", in *Pattern Recognition*, 2019.

[24] M. Johnson-Roberson, S. Kluckner, and et al., "Driving in the matrix: Can virtual worlds replace real-world data for autonomous driving?", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[25] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games", in *European Conference on Computer Vision (ECCV)*, 2016.

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

[27] 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.

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization", in *International Conference on Learning Representations (ICLR)*, 2018.

[29] R. Geirhos, P. Rubisch, C. M. P. Bischof, and et al., "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness", in *International Conference on Learning Representations (ICLR)*, 2019.

[30] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts", in *International Conference on Learning Representations (ICLR)*, 2017.

[31] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks", *Neural Networks*, vol. 106, pp. 249–259, 2018.

[32] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution", 2017.

[33] 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.

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

[35] 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.

[36] 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.