Improving the Performance of YOLOv11 in Images with Adverse Weather Conditions

John Butoto
Dept. of Electrical & Computer Engineering
University of North Florida,
Jacksonville, FL, USA
n01623186@unf.edu

***Abstract*—Object detection models such as You Only Look Once version 11 (YOLOv11) achieve high accuracy on standard benchmarks but degrade sharply under adverse weather (fog, rain, sand, snow). This work proposes an end-to-end pipeline that first generates synthetic degraded images, trains an EfficientNetB0 Convolutional Neural Network–Parameter Predictor (CNN-PP) module to predict optimal filter parameters (white balance, gamma, tone curve, contrast, unsharp mask, fog removal), applies those parameters to produce enhanced images, and finally trains YOLOv11 on the enhanced data. Using Common Objects in Context (COCO) and DAWN datasets, the enhanced pipeline did not yield consistent gains compared to the baseline, with only the dataset with images having rainy conditions showing an improvement from 0.55 to 0.61 in Mean Average Precision (mAP). These mixed results are attributed primarily to the specifics of the filter implementations, which will be examined in later sections.**

Keywords—Object detection; YOLOv11; adverse weather; image enhancement; EfficientNet; synthetic data, Mean Average Precision (mAP)

I. INTRODUCTION

Convolutional neural network (CNN) based object detectors (e.g., Faster Region-based Convolutional Neural Network (R-CNN), YOLO series) dominate benchmark datasets such as ImageNet [1], PASCAL Visual Object Classes (PASCAL VOC) [2] and COCO [3] especially in object detection tasks [4], [5] but suffer a dramatic drop in performance when applied to images degraded by fog, low light, rain, sandstorms, or snow. This is because images taken in such weather conditions are obscured, have reduce contrast and noise which leads to missed or misclassified objects thus undermining their real-world reliability. Also due to the domain shift in input images [6], general object detection models trained by high quality images often fail to achieve satisfactory results under adverse weather conditions.

Object detection in adverse weather conditions has been addressed along two directions in the literature: first, through the refinement of image-processing pipelines that preprocess or enhance degraded images [7]–[9] and second, through the evolution of deep learning (DL) architectures tailored for one-stage or two-stage detection frameworks [10]–[14]. Image-processing approaches apply operations such as dehazing, contrast enhancement, or white-balancing to restore latent object features before detection, while DL based methods seek to embed this ability directly within the detection network. These two streams often intersect: restoration modules can be coupled with detection backbones, and detector architectures can be augmented to jointly learn enhancement and classification tasks such as in [11, 15].

Within DL approaches, object detectors are generally classified into one-stage and two-stage paradigms. One-stage detectors (e.g., YOLO, Single-Shot MultiBox Detector (SSD)) perform localization and classification in a single forward pass over all image pixels, emphasizing inference speed, an essential characteristic for real-time autonomous driving. Two-stage detectors (e.g., Faster R-CNN, CoupleNet) first generate region proposals and then refine classification and bounding-box regression in a separate stage, trading off some speed for improved accuracy [15].

Given the stringent latency requirements of on-board vehicle systems, YOLO's single-pass architecture has emerged as a preferred baseline in many studies particularly when coupled with pre or post-processing enhancements. This, together with the availability of preprocessing techniques as suggested by [16, 17] that an image captured under adverse weather can be decomposed into a clean image and its corresponding weather-specific information serve as the basis for conducting this research to see whether the latest YOLO version (YOLOv11) enhanced by a preprocessing pipeline can perform better in object detection for images taken in adverse weather conditions.

Fig. 1 shows an example of object detection in foggy conditions. One can see that if the image can be properly enhanced with respect to the weather condition, more latent information about the original blurred objects and the misidentified objects can be recovered [11] and therefore enable object detection models to better identify items in an image.

Figure 1: Image showing how image enhancement can aid in improving object detection

II. RELATED WORK

Object detection. The current mainstream CNN-based object detection methods can be divided into two categories. One is the two-stage R-CNNs based on the regional proposals, which first generate regions of interest (ROI) from images and then classify them by another neural network [5]. The other is

based on one-stage regression, like the YOLOs [4] and single shot multibox detector (SSD) [18], which predict objects labels and bounding boxes coordinates at once using only one regression network. In this study, the proposed model applied the YOLOv11 architecture because of its one-step, fast and high-precision feature extraction capabilities.

Object detection in extreme conditions. Currently, there are mainly three kinds of solutions for object detection in extreme conditions. The most common one is the image augmentation method that applies various transformation functions (e.g., random blur, random rain, random gamma, random brightness, etc.) with a certain probability to simulate extreme conditions. This includes approaches such as the Stylization Data-Driven Neural-Image-Adaptive (SDNIA) proposed by Ghiasi et al. [20]. This approach uses Neural Style Transfer (NST) as an augmentation method to simulate extreme weather conditions by transferring specific "styles" from images with these conditions (fog, rain, low light) to regular training images. This augmentation enables the model to learn valuable information about how objects appear under adverse conditions, helping the model generalize better to real-world situations where extreme weather can severely hinder detection performance.

The second is to perform image enhancement and detection tasks simultaneously through joint or multi-task learning [19]. For instance, Alok, Manisha and G.K. Sharma propose a Low-light Detection Transformer (LDETR) [19] that employs an encoder-decoder transformer architecture to enhance images before detection, adapting to specific lighting and weather-induced degradation. This method uses an attention module to improve the signal-to-noise ratio in low-light or weather-affected images, thereby enhancing visibility for better detection. It effectively addresses common issues in adverse weather and low-light conditions, such as low contrast, inadequate brightness, and excessive noise. However, this method has a loss functions designed based on specific physical/empirical formulas making it difficult to implement. Another good example for this is the AI-YOLO [11] which incorporates an image-adaptive processing module known as the Differentiable Image Processing (DIP) module attached to the YOLOv3 backbone that adjusts the image according to the specific weather condition (fog, rain, snow, or low-light). This module employs a small convolutional neural network (CNN-PP) to predict enhancement parameters based on the input image's content, such as brightness, tone, and weather-related distortions, enabling the model to focus on restoring key latent details from degraded images. However, this kind of adaptive method must know in advance how many transformation functions and which hyperparameters need to be learned. Also, the design of the CNN-PP module needs to be customized and re-trained for specific extreme conditions.

The last one is domain-adaptive [21], which assumes that there is domain transfer between images captured under normal and poor weather conditions, and the weather specific information can be eliminated by learning the domain prior knowledge to make features weather-invariant. Hnewa, Mazin, and Hayder Radha's MultiScale Domain Adaptive YOLO (MS-DAYOLO) [21] is such an approach. They propose a framework that injects a Domain Adaptive Network (DAN) into three distinct feature-extraction scales (F1, F2, F3) of the YOLOv4 backbone. During training, each feature map is routed through a Gradient Reversal Layer (GRL) and a small convolutional domain-classifier head, driving the backbone to generate domain-invariant representations via adversarial learning. The total loss combines the standard detection loss on labelled source images with a domain-classification loss on both source and unlabelled target images $L_t = L_{\text{det}} + \lambda L_{\text{dc}}$ and then the DAN is discarded at inference so there is no run-time penalty. However, this relies on careful tuning of the trade-off hyperparameter (λ) between detection and domain losses, and adversarial training across multiple scales which can introduce instability and increased training overhead.

III. Proposed Approach

Building on these prior efforts, we observe that:

- Data augmentation requires careful selection and tuning of synthetic transformations to cover all weather artifacts.
- Joint enhancement–detection methods introduce additional loss terms and sub-networks that are hard to balance and often need per-condition retraining.
- Domain adaptation adds adversarial objectives and hyperparameter trade-offs, increasing training complexity and instability.

To overcome these limitations, This papers proposes an end-to-end, image-adaptive YOLOv11 framework that automatically predicts and applies optimal enhancement filters for each input via a lightweight CNN-PP module and a differentiable preprocessing pipeline. By learning to restore latent object details before detection and leveraging YOLOv11's fast one-step architecture high-precision object detection can be achieved across a wide range of adverse weather conditions without manual style transfers or unstable adversarial training.

As illustrated in Fig. 2 paired data is first generated by applying the differentiable filters which are Defog, White Balance, Gamma, Tone Curve, Contrast, and Unsharp Mask (USM) to clean COCO images, creating "adverse" inputs with known filter parameters. These 200×200 degraded images are then fed into the EfficientNetB0-based CNN-PP module, which regresses the six filter parameter groups via MSE loss. Once trained, CNN-PP predicts optimal enhancement settings for each real DAWN image; the are applied via the DIP pipeline to produce enhanced images that will be inputs for the YOLOv11 model. Finally, the enhanced images are passed to the YOLOv11 model for detection. The image-adaptive detection framework consists of five main stages, as illustrated in Fig 2.

A. Dataset Overview

The experiment was conducted using two publicly available benchmarks. First, for CNN-PP training, the Microsoft COCO dataset [3] was used. It originally contained over 330,000 images but only 20,151 images were selected; 16 120 for training and 4 031 for validation. These were used to train the CNN-PP module for 100 epochs to regress the six filter parameters. Second, for pipeline evaluation, the DAWN dataset was used. This dataset contains a real-world collection of 5,553 images captured under diverse adverse weather conditions (fog, rain, snow, and sand). All DAWN images were passed through the learned DIP + YOLOv11 pipeline,

and detection performance was assessed against their original bounding-box annotations.

All detection results from the image-adaptive pipeline were directly compared against a baseline YOLOv11 model that received the raw DAWN images without any preprocessing. This comparison quantified the gains attributable solely to the CNN-PP + DIP enhancements.

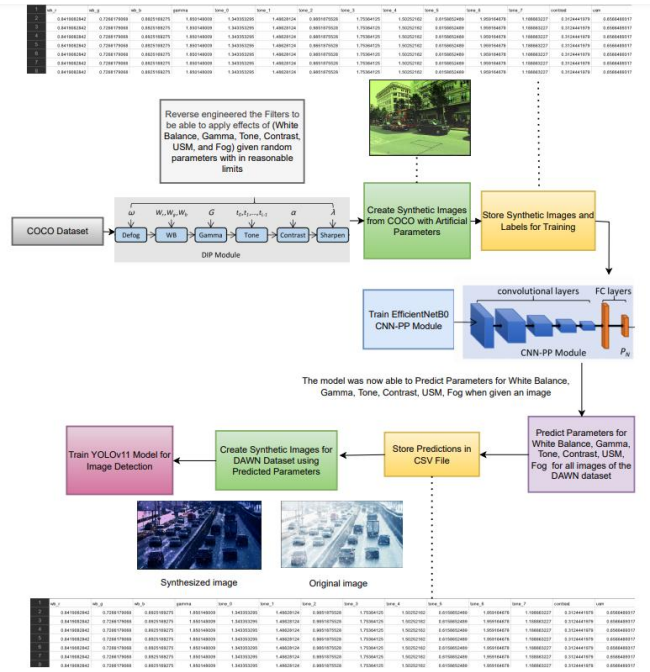

Figure 2: The framework of the Image Adaptive YOLOv11 pipeline

B. Synthetic Data Generation

This step involved sampling clean images from the COCO dataset and applying a suite of randomized, yet controlled, photometric and weather degradations. Specifically, each image is passed through differentiable filter modules—Defog, White Balance (R, G, B channels), Gamma, Tone Curve (8 - point), Contrast, and Unsharp Mask (USM) using parameters drawn uniformly from narrow intervals (e.g., WB $\in$ [0.7, 1.3], Gamma $\in$ [0.5, 3.0], Tone $\in$ [0.5, 2.0], Contrast $\in$ [0.1, 1.0], USM $\in$ [0.1, 1.0], Fog $\in$ [0.3, 1.0]). The resulting synthetic "adverse" images are stored alongside their ground truth filter parameters in a csv file, forming a paired dataset for downstream regression training of the CNN-PP module.

C. CNN-PP Training

The Convolutional Neural Network Parameter Predictor (CNN-PP) was designed as a lightweight, image-driven module to estimate the optimal enhancement parameters needed by our differentiable image processing (DIP) pipeline. By taking a degraded input image, CNN-PP produced six groups of filter coefficients covering defogging, white balance, gamma correction, tone curve adjustments, contrast scaling, and unsharp masking so that each image could be adaptively pre-processed before object detection. To achieve both accuracy and efficiency, CNN-PP combined a frozen, high capacity feature extractor with small, specialized heads that regressed each parameter group in a single forward pass.

The CNN-PP module ingested a 200 × 200 × 3 RGB image and passes it through a frozen EfficientNetB0 backbone. Freezing the EfficientNetB0 weights ensured that the feature extractor retained its pretrained representational power while preventing overfitting on the relatively small synthetic dataset. The convolutional output of shape 7 × 7 × 1280 was then collapsed via global average pooling into a 1280-dimensional feature vector, summarizing spatial activations into channel-wise descriptors. A dropout layer (rate = 0.2) followed to regularize the network by randomly omitting features, and a 512-unit fully connected layer with Rectified Linier Unit (ReLU) activation projected those descriptors into a compact, high-level embedding that served as a common foundation for all parameter predictions.

From this 512-dimensional embedding, six lightweight "heads" branched off, each consisting of a sigmoid output whose value is linearly remapped to the appropriate filter range. The White Balance head used three neurons to predict per-channel gains in [0.7, 1.3], while the Gamma and Contrast heads each used a single neuron to produce values in [0.5, 3.0] and [0.1, 1.0], respectively. The Tone Curve head output eight control-point adjustments in [0.5, 2.0], allowing fine-grained nonlinear brightness remapping. Finally, the Unsharp Mask (USM) and Fog heads each produced one coefficient in [0.1, 1.0] and [0.3, 1.0], enabling adaptive edge sharpening and haze removal. By training all six heads jointly with mean-squared error against known synthetic parameters, the CNN-PP learned to infer optimal enhancement settings directly from degraded images, readying them for the subsequent DIP and YOLOv11 detection stages.

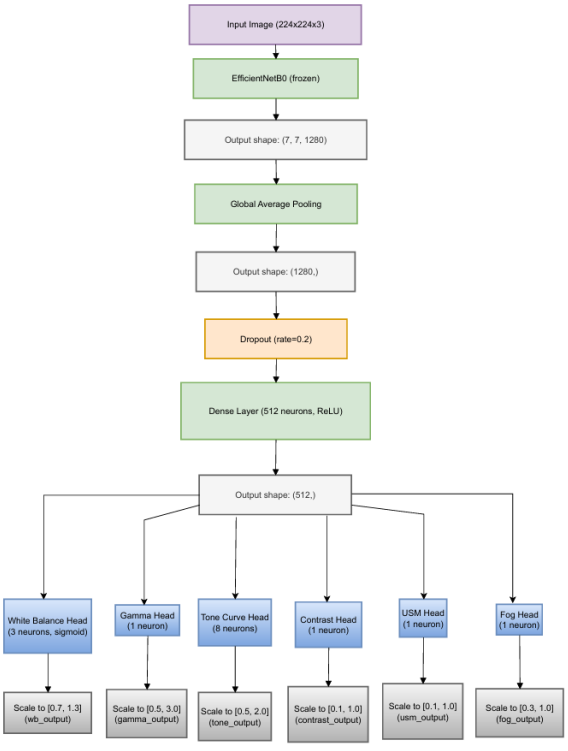

Figure 3: Architecture of the CNN-PP Module

D. Parameter Prediction & CSV Storage

Once trained, the CNN-PP module was run in inference mode on the undegraded DAWN images to discover the set of "optimal" filter parameters that best recover clean appearance. These predicted parameters were exported to a CSV file,

providing a lookup table that mapped each original DAWN image to its ideal enhancement settings.

E. DAWN Data Enhancement

Real-world adverse weather images were then loaded from the DAWN dataset. For each image, corresponding filter parameters from the CSV were retrieved and the six differentiable filters were applied in sequence via the DIP module. This produced enhanced DAWN images that mimiced the optimal restoration learned on synthetic data, aligning it back toward the domain of clear images.

F. YOLOv11 Training

Finally, the enhanced DAWN images, now exhibiting reduced weather-induced degradation, were used to train the YOLOv11 object detector. The original COCO-derived bounding-box annotations were used for supervision. By training on these pre-processed inputs, YOLOv11 learned to detect objects in a better way under adverse weather, without modifying its core detection architecture. The full training and inference code (including data-augmentation scripts and model checkpoints) is available on GitHub [22].

IV. PERFORMANCE EVALUATION

To evaluate the effectiveness of both the enhancement module and object detector, a set of standard performance metrics were used. For the CNN-PP parameter predictor, Mean Absolute Error (MAE) was used to quantify how accurately the model regressed each filter parameter using the COCO dataset. For the YOLOv11 detector, Precision, Recall, and mean Average Precision at Intersection Over Union (IoU) = 0.5 (mAP@0.5) were measured to assess detection quality under various weather conditions. All evaluations on the DAWN dataset used a hold-out test split that balanced image across fog, rain, sand, and snow.

A. Evaluation Metrics

Assessment of the performance of the CNN-PP module was done using two standard metrics.

a) Mean Squared Error (MSE): This metric measured the average of the squared differences between the true filter parameters and those predicted by the CNN-PP module. By squaring each error, larger deviations were penalized more heavily. In the experiment, MSE served as the training loss for CNN-PP, guiding the model to produce parameter estimates that closely matched the ground truth.

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 \tag{1}$$

b) Mean Absolute Error (MAE): This metric computed the average absolute difference between the true and predicted filter parameters. Unlike MSE, it treats all errors proportionally, providing a clear interpretation of average prediction deviation in the same units as the parameters themselves. The MAE was reported on both training and validation splits to quantify how far, on average, CNN-PP's predictions were from the true values.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2}$$

On the other hand, the assessment of the performance of the YOLOv11 model both for the baseline and pre-processed version was done using three key metrics.

c) Precision: This is the fraction of detected bounding boxes that correctly correspond to actual objects. A high precision means the detector makes few false-positive errors (i.e., it rarely labels background or noise as objects). In our evaluation, precision indicated how reliably YOLOv11 avoided false detections under adverse weather. Precision can be measured using the proportion of true positive predictions out of all instances predicted as positive.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \tag{3}$$

d) Recall: This is the fraction of ground-truth objects that the detector successfully finds. A high recall means the model misses very few real objects (i.e., it seldom overlooks objects in the scene). We used recall to measure YOLOv11's completeness of detection across fog, rain, sand, and snow conditions.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}} \tag{4}$$

e) Mean Average Precision at IoU=0.5 (mAP@0.5): This is a metric that summarizes the detector's overall performance by averaging the precision–recall trade-off across all object classes, using an Intersection-over-Union threshold of 0.5 to determine true positives. It captures both precision and recall in a single number. We compared mAP@0.5 between the baseline YOLOv11 and our CNN-PP + DIP-enhanced pipeline to assess the net gain (or loss) in detection quality under challenging weather.

$$mAP@0.5 = \frac{1}{C}\sum_{c=1}^{C}\mathrm{AP_c^{IoU=0.5}} \tag{5}$$

B. Results

This section presents the quantitative outcomes of the two-stage pipeline. First, we assess the CNN-PP module's ability to regress the six filter parameters on synthetic test data, reporting both mean squared error (MSE) via the training loss curve shown in Fig. 4 and mean absolute error (MAE) for each head in Table I. Next, we evaluate object detection performance on the DAWN dataset by comparing the baseline YOLOv11 model against the CNN-PP + DIP enhanced pipeline. Detection metrics in terms of Precision, Recall, and mean Average Precision at IoU = 0.5 (mAP@0.5) are reported in Table II and Table III for fog, rain, sand, and snow conditions, highlighting the impact of adaptive preprocessing on real- world adverse-weather scenarios.

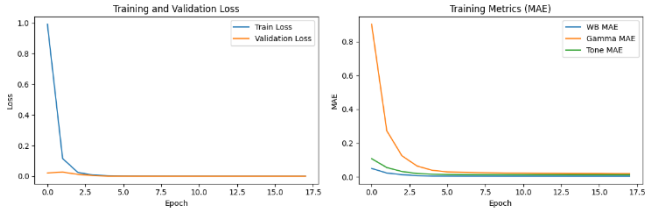

Figure 4: Training and validation loss (left) and per-head MAE (White Balance, Gamma, Tone; right) curves for the CNN-PP module over the training epochs.

TABLE I.
CNN-PP MODEL RESULTS PER FILTER HEAD

Filter	Training MAE	Validation MAE
Overall	0.0348	0.0029
Contrast	0.0078	0.0027
Fog	0.0051	0.0024
Gamma	0.0193	0.00098
Tone	0.0126	0.0024
USM	0.0098	0.0029
WB	0.0045	0.0018

TABLE II.
BASELINE YOLOV11 RESULTS

Weather	Precision	Recall	mAP@0.5
Fog	0.71	0.72	0.75
Rain	0.76	0.57	0.55
Sand	0.73	0.59	0.67
Snow	0.71	0.60	0.63

TABLE III.
CNN-PP + DIP PREPROCESSING YOLOV11 RESULTS

Weather	Precision	Recall	mAP@0.5
Fog	0.82	0.32	0.40
Rain	0.68	0.42	0.61
Sand	0.64	0.36	0.38
Snow	0.65	0.49	0.57

C. Discussing model performance

The training curves in Fig. 4 and the head-wise MAE values in Table I demonstrate that the CNN-PP module learned to predict filter parameters with high accuracy on synthetic data. As shown in Fig. 4 he steep drop in both training and validation losses over the first few epochs, followed by a gentle plateau, suggests rapid convergence without significant overfitting. The overall training MAE of 0.0348 fell to only 0.0029 on the validation split, and each individual head achieved MAE lower than 0.01 (e.g., Fog at 0.0051/0.0024, White Balance at 0.0045/0.0018, and Contrast at 0.0078/0.0027) indicating accurate, stable regression of enhancement parameters.

Turning to object detection, the baseline YOLOv11 model trained directly on raw DAWN images produced solid results under certain conditions as seen in Table II. In foggy scenes, it achieved 0.71 precision, 0.72 recall, and 0.75 mAP@0.5 which was the strongest performance while rain posed the greatest challenge (0.76/0.57/0.55). Sandstorms and snow delivered intermediate scores (mAPs of 0.67 and 0.63, respectively), reflecting the varied difficulty of each weather type.

When we incorporated the CNN-PP + DIP preprocessing pipeline, detection performance shifted in mixed ways. Fog precision rose to 0.82 an 11% absolute gain but recall collapsed to just 0.32, dragging fog mAP down to 0.40. In rain, mAP improved modestly to 0.61 despite small drops in both precision and recall, suggesting a better precision–recall balance overall. Sand and snow both saw lower mAPs (0.38 and 0.57), indicating that the current filter suite may not generalize well to in high-contrast scenarios.

Comparing baseline and pre-processed results side by side reveals that while preprocessing can sharpen object boundaries (boosting precision), it often suppresses subtle features needed for complete detection and therefore hurting recall measurement. The rain subset was the sole case where mAP rose measurably, implying that moderate noise and contrast distortions are best addressed by the synthetic-trained filters. In more extreme conditions dense fog, heavy snow, or sandstorms the DIP adjustments either over-corrected or under-corrected, leading to missed detections.

Overall, despite excellent regression performance on synthetic data, the CNN-PP + DIP preprocessing did not deliver consistent improvements on real-world DAWN images. The domain gap between synthetic and actual weather artifacts likely limited parameter generalization, and some filters inadvertently removed critical object cues. These findings highlight the need for domain-aware fine-tuning of the CNN-PP module and possibly the introduction of specialized filters tailored to each adverse condition.

V. CONCLUSION AND FUTURE WORKS

In this work, we introduced an end-to-end image-adaptive pipeline that first learns to predict optimal enhancement parameters via a CNN-PP module trained on synthetically degraded COCO images and then applies those parameters through a differentiable preprocessing (DIP) stage before feeding into a standard YOLOv11 detector. The CNN-PP achieved very low validation MSE ($< 3.5 \times 10^{-5}$) and MAE (< 0.003) across all six filter heads, confirming its ability to regress fog, white-balance, gamma, tone-curve, contrast, and unsharp-mask parameters with high fidelity on synthetic data. When evaluated on the DAWN dataset, the preprocessing pipeline boosted precision most notably in foggy conditions (from 0.71 to 0.82) and yielded modest mAP gains in rain (from 0.55 to 0.61), but it also led to pronounced recall degradation under heavy occlusion (e.g., fog) and failed to improve detection in sandstorm and snowy scenes.

These mixed outcomes underscore the challenges of transferring enhancement parameters learned on synthetic examples to real-world weather artifacts. The CNN-PP filters, while accurately calibrated on controlled degradations, sometimes over or under-corrected actual images suppressing subtle object cues and thereby causing missed detections. Furthermore, the one-size-fits-all filter ranges were insufficient to capture the full variability of particulate noise,

low-light scattering, and texture distortions present in DAWN. This domain gap points to the necessity of more adaptation strategies, such as mixed synthetic and real training, adversarial domain alignment, or specialized heads tailored to individual weather types.

Despite these limitations, this framework provides a clear demonstration of how image-adaptive preprocessing can be integrated with a modern one-stage detector without altering its core architecture.

Future work could begin by fine-tuning the CNN-PP module on a small set of real-world adverse-weather images, blending synthetic and actual data so that the predicted filter parameters better reflect true fog, rain, snow, or sandstorm artifacts. In parallel, the network architecture could be extended with weather-specific prediction heads that learn distinct parameter ranges for each condition, rather than relying on a single, one-size-fits-all model. Finally, the implementation of the DIP filters themselves could be refined moving beyond fixed, hand-designed functions toward lightweight, learnable filter blocks (e.g., small CNN-based or parameterized convolutional modules) that can adaptively shape their operations based on the input image's content.

REFERENCES

[1] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[2] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88 (2010): 303-338.

[3] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740-755. Springer International Publishing, 2014.

[4] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[5] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[6] Sindagi, Vishwanath A., Poojan Oza, Rajeev Yasarla, and Vishal M. Patel. "Prior-based domain adaptive object detection for hazy and rainy conditions." In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 763-780. Springer International Publishing, 2020.

[7] Wang, Yongzhen, Xuefeng Yan, Kaiwen Zhang, Lina Gong, Haoran Xie, Fu Lee Wang, and Mingqiang Wei. "Togethernet: Bridging image restoration and object detection together via dynamic enhancement learning." In *Computer graphics forum*, vol. 41, no. 7, pp. 465-476. 2022.

[8] Li, Wei. "Vehicle detection in foggy weather based on an enhanced YOLO method." In *Journal of Physics: Conference Series*, vol. 2284, no. 1, p. 012015. IOP Publishing, 2022.

[9] Hassaballah, Mahmoud, Mourad A. Kenk, Khan Muhammad, and Shervin Minaee. "Vehicle detection and tracking in adverse weather using a deep learning framework." *IEEE transactions on intelligent transportation systems* 22, no. 7 (2020): 4230-4242.

[10] Wang, Rui, He Zhao, Zhengwei Xu, Yaming Ding, Guowei Li, Yuxin Zhang, and Hua Li. "Real-time vehicle target detection in inclement weather conditions based on YOLOv4." *Frontiers in Neurorobotics* 17 (2023): 1058723.

[11] Liu, Wenyu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. "Image-adaptive YOLO for object detection in adverse weather conditions." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 2, pp. 1792-1800. 2022.

[12] Zhang, Hongyi, Rabia Sehab, Sheherazade Azouigui, and Moussa Boukhnifer. "Application and comparison of deep learning methods to detect night-time road surface conditions for autonomous vehicles." *Electronics* 11, no. 5 (2022): 786.

[13] Farid, Annam, Farhan Hussain, Khurram Khan, Mohsin Shahzad, Uzair Khan, and Zahid Mahmood. "A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments." *Applied sciences* 13, no. 5 (2023): 3059.

[14] Xu, Lingzhi, Wei Yan, and Jiashu Ji. "The research of a novel WOG-YOLO algorithm for autonomous driving object detection." *Scientific reports* 13, no. 1 (2023): 3699.

[15] Özcan, İbrahim, Yusuf Altun, and Cevahir Parlak. "Improving YOLO detection performance of autonomous vehicles in adverse weather conditions using metaheuristic algorithms." *Applied Sciences* 14, no. 13 (2024): 5841.

[16] Narasimhan, Srinivasa G., and Shree K. Nayar. "Vision and the atmosphere." *International journal of computer vision* 48 (2002): 233-254.

[17] You, Shaodi, Robby T. Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. "Adherent raindrop modeling, detection and removal in video." *IEEE transactions on pattern analysis and machine intelligence* 38, no. 9 (2015): 1721-1733.

[18] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21-37. Springer International Publishing, 2016.

[19] Tiwari, Alok Kumar, Manisha Pattanaik, and G. K. Sharma. "Low-light DEtection TRansformer (LDETR): object detection in low-light and adverse weather conditions." *Multimedia Tools and Applications* 83, no. 36 (2024): 84231-84248.

[20] Ding, Yuexiong, and Xiaowei Luo. "SDNIA-YOLO: A Robust Object Detection Model for Extreme Weather Conditions." *arXiv preprint arXiv:2406.12395* (2024).

[21] Hnewa, Mazin, and Hayder Radha. "Multiscale domain adaptive yolo for cross-domain object detection." In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3323-3327. IEEE, 2021.

[22] J. R. Butoto, "Improving the Performance of YOLOv11 in Images with Adverse Weather Conditions," GitHub repository, Apr. 2025. [Online]. Available: https://github.com/JRButoto/Improving-the-Performance-of-YOLOv11-in-Images-with-Adverse-Weather-Conditions.git.