# ENHANCING 2D OBJECT DETECTION FOR AUTONOMOUS DRIVING VIA IMAGE PROCESSING AND FINE-TUNING

*Xuanhao Zhu*, Xinhao Xu*, Yuanzhe Hu**

University of California, San Diego
Department of Computer Science and Engineering, Team XXY
La Jolla, CA

## ABSTRACT

This project investigates how image processing algorithms can enhance the performance of machine learning models for 2D object detection under challenging driving conditions. Using a YOLO-based model trained on a clean daytime driving dataset, we evaluate performance under distorted driving environments such as foggy, nighttime, and blurry scenes. Two distinct image processing algorithms are applied to enhance each distorted dataset, and the processed distorted datasets are then tested on the original model to assess whether enhancement improves detection accuracy. Finally, we fine-tune the original model on each distorted dataset and compare the results to those with and without image enhancement. The goal of this work is to analyze whether image processing or fine-tuning offers a more effective strategy for improving model robustness and accuracy in real-world autonomous driving scenarios. Our code and processed datasets are available at https://github.com/N1nomae/ECE-253-XXY.

***Index Terms***— object detection, autonomous driving, YOLO, image processing, fine-tuning

## 1. INTRODUCTION

Object detection is one of the most important tasks in computer vision and plays an essential role in autonomous vehicles. It allows vehicles to recognize and locate objects such as vehicles, pedestrians, and traffic signs, forming the foundation for autonomous driving. Modern object detection algorithms, such as You Only Look Once (YOLO) based models, have demonstrated remarkable performance in real world tasks [1].

However, autonomous driving in the real world often occurs under adverse environmental conditions, such as fog, darkness, or blur, which degrade the visual quality of the images captured by cameras. These distortions can obscure key features of the object, leading to decreased detection accuracy. While retraining or fine-tuning models on distorted dataset can partially address the problem, such solutions require large annotated datasets. Therefore, an alternative strategy is to apply image processing algorithms to enhance image quality before feeding it into the model.

In this project, we investigate how image processing algorithms affect the robustness of YOLO-based 2D object detection under three types of degraded driving conditions. We first train a baseline YOLO model using a clean daytime driving dataset. We then evaluate the model's performance on three customized distorted datasets representing foggy, nighttime, and blurry conditions. For each distorted dataset, we apply two image processing methods designed to improve visibility or restore degraded features. The enhanced images are tested using the original baseline model to assess whether preprocessing improves detection performance. Finally, we fine tune the original model on each distorted dataset and compare the performance against models that rely solely on image enhancement

The objective of this study is to analyze whether image processing or fine tuning provides a more effective strategy for improving model robustness and accuracy in challenging autonomous driving environments. By systematically comparing the effects of enhancement preprocessing and domain-specific fine-tuning, we aim to provide insight into practical methods for enhancing object detection under adverse visual conditions.

## 2. RELATED WORK

### 2.1. Image Defogging for Vehicle Detection

Adverse weather, especially fog, poses severe challenges for perception stacks in autonomous driving by attenuating scene contrast and washing out high-frequency cues that detectors rely on. Early defogging approaches leveraged physics-based priors to estimate transmission and atmospheric light, notably the Dark Channel Prior [2] and Fast Visibility Restoration [3]. While effective for image enhancement, their assumptions often break in the long-range, low-texture road scenes common in driving. With deep learning, single-image defogging shifted toward end-to-end restoration. Representative

[*] Equal contribution.

designs fall into physics-guided, model-based CNNs (e.g., AOD-Net [4]) and multi-scale/attention restoration networks (GridDehazeNet [5], FFA-Net [6], MSBDN [7]) that boost PSNR/SSIM on synthetic hazy benchmarks.

However, naively inserting defogging as a pre-processing step can yield limited or even negative transfer for downstream tasks if restoration artifacts shift the detection domain. This was already observed in fog-centric semantic perception, where defogging brought only marginal gains when strong task training was used [8]. To close the domain gap, recent driving-centric work converges on (i) fog-aware datasets and (ii) task-aligned learning. On the data side, Foggy Cityscapes/Foggy Driving provide scalable synthetic and real fog for urban scenes [8]; ACDC adds pixel-accurate labels with adverse/clear correspondences [9]; and [10] offers multimodal camera–LiDAR/Radar pairs for robust fusion studies. On the learning side, fog simulation plus semi-supervised, depth/transmission-aware domain adaptation boosts detection performance on Foggy Cityscapes and real-fog datasets [11]. Overall, the field is shifting from generic image cleanup to fog-aware, task-coupled training, where restoration is optimized for detection rather than appearance.

### 2.2. Image Enhancement for Nighttime Vehicle Detection

Digital image enhancement has been widely used to mitigate the domain gap faced by nighttime vehicle detectors. Classical, task-agnostic preprocessing like LIME improves visibility but can amplify noise and shift color statistics [12]. Learning-based low-light enhancement (LLIE) further advances quality via deep Retinex decomposition and reference-free curve estimation (Zero-DCE) [13, 14]. On detection benchmarks that include dark scenes, such visibility losses measurably degrade object detectors, motivating enhancement as a front-end[15]. Beyond generic LLIE, task-coupled designs either plug enhancement into the detection stack or build application-specific pipelines; for instance, the AICity fisheye framework shows that an enhancement stage can boost vehicle detection under low illumination and non-linear optics [16, 17].

A second line of work aligns enhancement with the detector objective. End-to-end co-training lets the network learn enhancement policies that maximize detection performance under low light [18]. Vehicle-centric LLIE tailored for intelligent driving (VELIE) further constrains enhancement to be lightweight and exposure-aware for in-vehicle deployment [19]. Complementary to enhancement, domain adaptation and night-to-day/day-to-night translation transfer daytime labels to nighttime, either by synthesizing low-light counterparts or learning illumination-invariant reflectance for detection [20, 21]. Recent taxonomies thus group methods into *detection-by-enhancement* vs. *enhancement-for-detection*, with domain adaptation providing an orthogonal axis to leverage well-lit data when nighttime labels are scarce [21].

### 2.3. Image Deblurring for Vehicle Detection

Image deblurring, particularly blind deblurring where the blur kernel is unknown, remains a challenging ill-posed problem. Classic approaches tackled this via optimization frameworks, often using variational methods based on handcrafted image priors, such as Total Variation (TV) [22], or later framing it as a Maximum A Posteriori (MAP) problem with learned natural image priors [23]. While traditional methods relied on statistical priors, recent advances have been dominated by deep learning. Early deep learning approaches, such as SRN [24], utilized multi-scale and recurrent CNNs to handle spatially varying blur. To enhance perceptual quality, GAN-based models like DeblurGAN-v2 [25] became prominent. Recent trends have focused on architectural efficiency and generalization. Models like NAFNet [26] established strong baselines by simplifying network structures. Concurrently, state-of-the-art performance has been advanced by architectures exploring novel decoder designs, like AdaRevD [27], and by unified ”all-in-one” restoration models such as PromptIR [28], which leverage prompting to handle diverse degradations. However, these methods are predominantly optimized for pixel-level fidelity rather than downstream task performance.

For high-level vision tasks, particularly object detection in autonomous driving, the objective shifts from pixel-level fidelity to maximizing task-specific metrics. Prior work shows that plugging a generic deblurring module in front of a detector can be suboptimal, restoration metrics misalign with detection and sometimes even hurt performance, which motivates task-aware or jointly-optimized pipelines [29, 30]. Along this line, RT-Deblur [31] targets real-time operation and integrates a YOLO-aware perceptual loss to directly optimize downstream detection quality. A complementary direction, crucial for safety, is to learn blur-robust features so that representations remain consistent under motion blur without incurring the latency of an explicit deblurring stage [32, 33]. We follow this latter path to bolster real-time robustness in driving scenarios.

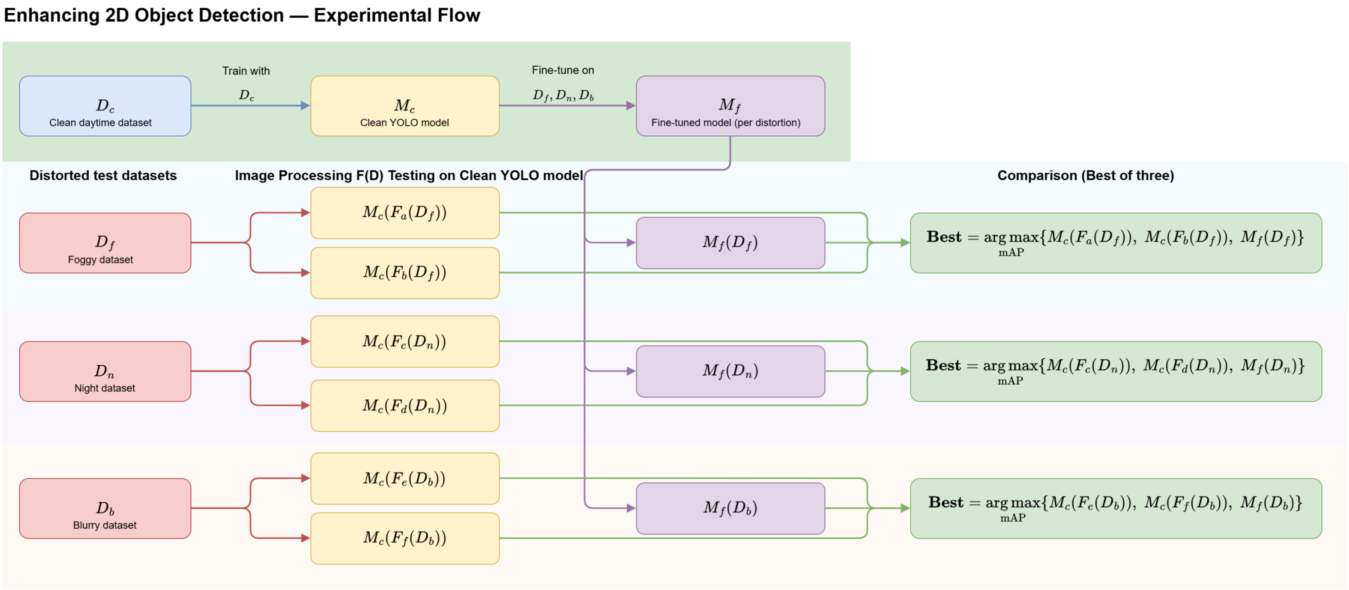
## 3. METHODS

### 3.1. Approaches

#### *3.1.1. Foggy - Dark Channel Prior (DCP)*

For the fog-degraded domain, we will utilize Dark Channel Prior (DCP) [34] as our first enhancement method before performing object detection with YOLO. DCP is a statistical assumption used in dehazing image processing, particularly for outdoor images:

$$I(x) = J(x)t(x) + A(1 - t(x))$$



Enhancing 2D Object Detection — Experimental Flow
$D_c$
Clean daytime dataset
Train with
$D_c$
$M_c$
Clean YOLO model
Fine-tune on
$D_f, D_n, D_b$
$M_f$
Fine-tuned model (per distortion)
Distorted test datasets
Image Processing F(D) Testing on Clean YOLO model
Comparison (Best of three)
$D_f$
Foggy dataset
$M_c(F_a(D_f))$
$M_c(F_b(D_f))$
$M_f(D_f)$
$\textbf{Best} = \arg\max_{\text{mAP}}\{M_c(F_a(D_f)),\ M_c(F_b(D_f)),\ M_f(D_f)\}$
$D_n$
Night dataset
$M_c(F_c(D_n))$
$M_c(F_d(D_n))$
$M_f(D_n)$
$\textbf{Best} = \arg\max_{\text{mAP}}\{M_c(F_c(D_n)),\ M_c(F_d(D_n)),\ M_f(D_n)\}$
$D_b$
Blurry dataset
$M_c(F_e(D_b))$
$M_c(F_f(D_b))$
$M_f(D_b)$
$\textbf{Best} = \arg\max_{\text{mAP}}\{M_c(F_e(D_b)),\ M_c(F_f(D_b)),\ M_f(D_b)\}$

**Fig. 1**. Work flow for enhancing 2D object detection under fog, night, and blur conditions.

where $I(x)$ is the observed foggy image, $J(x)$ is the latent haze-free scene radiance, $A$ represents the global atmospheric light, and $t(x) = e^{-\beta d(x)}$ is the transmission describing the portion of the light, which is determined by scene depth $d(x)$ and scattering coefficient of the atmosphere $\beta$. DCP assumes that in most of the local outdoor patches, at least one RGB channel has very low intensity at some pixels, so that we define:

$$J^{dark}(x) = \min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}} J^c(y)\right) \approx 0$$

where $J^c$ is the color channel of $J(x)$ and $\Omega(x)$ is the local patch centered at x.

In our implementation, the dark channel is first computed by applying a minimum operation across RGB channels followed by a local minimum filter. Atmospheric light is estimated from the brightest pixels in the dark channel while explicitly avoiding overexposed regions to prevent excessive dehazing. The transmission map is then estimated using the normalized dark channel and refined via guided filtering, which significantly reduces block artifacts and preserves object boundaries. Finally, the haze-free image is recovered using a lower-bounded transmission (atmospheric scattering model) to avoid over-amplification in dense fog regions. The algorithm flow can be shown as following:

**Algorithm 1: Dark Channel Prior (DCP) Dehazing.** *Input:* Foggy image $I$ (BGR). *Output:* Dehazed image $J$.

1. Convert $I$ from BGR to RGB and normalize to $[0, 1]$.
2. Compute dark channel:
$$D(x) = \min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}} I^c(y)\right).$$
3. Estimate atmospheric light $A$ from top $0.1\%$ brightest pixels in $D(x)$.
4. Clamp $A \geq 0.7$.
5. Estimate transmission:
$$t(x) = 1 - \omega \cdot D\left(\frac{I(x)}{A}\right), \quad \omega = 0.95.$$
6. Refine transmission using guided filtering:
$$t_{\text{refined}} = \text{GuidedFilter}(I, t).$$
7. Clamp $t_{\text{refined}} \leftarrow \max(t_{\text{refined}}, t_0)$, $t_0 = 0.15$.
8. Recover:
$$J(x) = \frac{I(x) - A}{t_{\text{refined}}(x)} + A.$$
9. Clip $J$ to $[0, 1]$, convert to BGR and uint8.

In our system, DCP is applied as a pre-processing enhancement step before YOLO inference on Foggy Cityscapes images. This method serves as a great classical baseline, enabling us to evaluate whether physically motivated dehazing alone can improve downstream object detection performance without additional neural network training or domain-specific fine-tuning

### *3.1.2. Foggy - Feature Fusion Attention Network*

FFA-Net (Feature Fusion Attention Network), for the fog domain, is used to restore visibility before YOLO object detection [6]. FFA-Net is a deep neural network that uses attention-guided feature fusion to accomplish end-to-end dehazing directly in the image domain. The channel and pixel attention mechanisms are combined in a new feature attention (FA) module. Considering that the haze is unevenly distributed across various image pixels and the weight information contained in various channel features is entirely different. The FA module specifically handles various pixels and features, giving CNN more versatility in processing various kinds of data and enhancing its representational power. This method can model complex non-uniform haze distributions and does not require explicit transmission estimation.

The core architectural idea of FFA-Net is to enhance feature representation through multi-level feature fusion combined with attention mechanisms. The network adopts an encoder–decoder structure with residual blocks, where shallow features preserve fine-grained spatial details and deeper features capture high-level semantic information. Feature Fusion Modules (FFMs) are used to merge encoder and decoder features at multiple scales, ensuring effective information propagation across resolutions.

To adaptively emphasize informative features, FFA-Net integrates both channel attention and spatial attention within a unified attention module. Channel attention selectively reweights feature channels based on their global importance, while spatial attention highlights haze-affected regions at the pixel level. This design allows the network to focus on regions with severe fog while preserving structural details such as object boundaries and textures.

**Algorithm 2: FFA-Net Dehazing with Classical Fallback.** *Input:* Image folder $\mathcal{D}$, pretrained model path `MODEL_PATH`. *Output:* Dehazed images saved to output folder.

```
1. Load the FFA-Net model and set it to evaluation mode.
2. Check whether pretrained weights exist at MODEL_PATH.
3. For each image file $I$ in $\mathcal{D}$:
   (a) Read the image in BGR format.
   (b) If pretrained weights are unavailable (fallback mode):
       i. Apply an enhanced DCP-based dehazing procedure to obtain the restored image $J$.
       ii. Save $J$ to the output folder.
   (c) Else (FFA-Net inference mode):
       i. Convert the image to RGB and normalize it to a tensor in $[-1, 1]$.
       ii. If the image resolution exceeds the maximum allowed size:
           A. Resize the image tensor to satisfy the size constraint.
       iii. Run a forward pass through FFA-Net to obtain the restored output $\hat{I}$.
       iv. Post-process $\hat{I}$ by de-normalizing to $[0, 1]$, clamping values, and resizing back if needed.
       v. Convert the result back to BGR uint8 as $J$, and save $J$ to the output folder.
```

### *3.1.3. Night - Contrast Limited Adaptive Histogram Equalization*

For nighttime enhancement, we use Contrast Limited Adaptive Histogram Equalization (CLAHE), which prevents noise over-amplification while improving contrast locally to improve image visibility. The input image is divided into multiple small regions, and histogram equalization is performed independently within each region to improve local contrast. To guarantee that histogram peaks surpassing this threshold are clipped and redistributed across other gray levels, CLAHE adds a contrast limit (clipLimit), in contrast to conventional Adaptive Histogram Equalization (AHE). This avoids over-enhancement in noisy, dark areas. As a result, CLAHE can effectively brighten nighttime driving scenes while preserving local structure and suppressing noise amplification. CLAHE can also be directly applied to nighttime driving images from BDD Night subsets without requiring any model training.

In our implementation, we further adopt a *brightness-adaptive CLAHE strategy* to account for varying illumination levels across nighttime images. Specifically, we first convert the input image to grayscale and compute its average brightness. Based on this brightness measure, the enhancement strength is adjusted automatically. For very dark images, a higher contrast limit and smaller tile size are used to emphasize local details. For moderately dark images, a slightly increased contrast limit is applied, while for normal or brighter images, the base CLAHE parameters are retained. This adaptive design prevents both under-enhancement in extremely dark scenes and over-enhancement in relatively well-lit regions.

**Algorithm 3: Nighttime Enhancement via Brightness-Adaptive CLAHE.** *Input:* Image folder $\mathcal{D}$, base parameters (`clip_limit`,`tile_grid_size`). *Output:* Enhanced images saved to output folder.

```
1. For each image $I$ in folder $\mathcal{D}$:
   (a) Read $I$ in BGR format. If reading fails, skip this image.
   (b) Convert $I$ to grayscale and compute the average brightness $m$.
   (c) Select adaptive CLAHE parameters based on $m$:
          i. If $m < 50$ (very dark): clip $\leftarrow 1.5 \times$ clip_limit, tile $\leftarrow (6,6)$.
          ii. Else if $m < 100$ (moderately dark): clip $\leftarrow 1.2 \times$ clip_limit, tile $\leftarrow (8,8)$.
          iii. Else (normal/bright): clip $\leftarrow$ clip_limit, tile $\leftarrow$ tile_grid_size.
    (d) Convert $I$ from BGR to LAB and split into $(L, A, B)$ channels.
    (e) Apply CLAHE on the luminance channel:
          $L' \leftarrow \text{CLAHE}\big(L; \texttt{clipLimit} = \texttt{clip}, \texttt{tileGridSize} = \texttt{tile}\big)$
    (f) Merge $(L', A, B)$ to form an enhanced LAB image and convert back to BGR.
```

### 3.1.4. Night - Zero-DCE

Zero-DCE enhances low-light images by using a neural network to learn a brightness mapping curve, and then producing a brightened output based on this curve and the original input image [14]. The core idea lies in two key design advantages: the brightness mapping curve fitting mechanism and a label-free optimization objective. First, Zero-DCE is a low-light enhancement framework that does not require corresponding reference images, and it can effectively avoid overfitting. Second, the method designs a high-order mapping curve where each pixel can assign its own parametric adjustment curve, in order to enable fine-grained luminance correction.

$$LE(I(\mathbf{x}); \alpha) = I(\mathbf{x}) + \alpha I(\mathbf{x})(1 - I(\mathbf{x}))$$

where $\mathbf{x}$ denotes pixel coordinates and $LE(I(\mathbf{x})$ is the enhanced version of given input $I(\mathbf{x})$, $\alpha \in [-1, 1]$ is the trainable curve parameter. Following is the high-order version:

$$LE_n(\mathbf{x}) = LE_{n-1}(\mathbf{x}) + \alpha_n LE_{n-1}(\mathbf{x})\,(1 - LE_{n-1}(\mathbf{x}))$$

Within our pipeline, Zero-DCE enhanced nighttime images will be fed directly into YOLO for inference.

**Algorithm 4: Nighttime Enhancement via Zero-DCE.**
*Input:* Image folder $\mathcal{D}$, pretrained Zero-DCE network $f_\theta$ (DCE-Net), number of curve stages $N$.
*Output:* Enhanced images saved to output folder.

```
1. Initialize Zero-DCE:
    (a) Load pretrained weights into $f_\theta$.
    (b) Set $f_\theta$ to inference mode (e.g., eval()), and select device (CPU/GPU).
2. For each image $I$ in folder $\mathcal{D}$:
    (a) Read $I$ in BGR format. If reading fails, skip this image.
    (b) Convert $I$ from BGR to RGB and normalize pixel values to $[0, 1]$:
          $I_0 \leftarrow \text{toFloat}(\text{RGB}(I))/255.$
    (c) Predict pixel-wise curve parameter maps using Zero-DCE:
          $\{A_1, A_2, \ldots, A_N\} \leftarrow f_\theta(I_0),$
        where each $A_n$ has the same spatial size as $I_0$ (and is applied channel-wise).
    (d) Iteratively apply the light-enhancement curve:
          $LE_0 \leftarrow I_0,$
          $LE_n \leftarrow LE_{n-1} + A_n \odot LE_{n-1} \odot (1 - LE_{n-1}),$
          $n = 1, \ldots, N,$
          $LE_n \leftarrow \text{clip}(LE_n, 0, 1),$
        where $\odot$ denotes element-wise multiplication.
    (e) Post-process and save:
          i. $I' \leftarrow \text{toUint8}(255 \cdot LE_N)$.
          ii. Convert $I'$ from RGB back to BGR.
          iii. Save $I'$ to the output folder with the original filename.
```

### 3.1.5. Blur - Unsharp Masking

Unsharp Mask (USM) is an image sharpening technique that enhances the contrast of edges by comparing the original image with its blurred version. Its core principle is to first blur the image to isolate and emphasize edge structures, and then increase the contrast in these regions to achieve sharpening. The formula is [35]:

$$\hat{I} = \lambda(I - \mathcal{F}_L(I)) + I$$

where $I$ denotes the original image, $\hat{I}$ denotes the enhanced image, $\mathcal{F}_L$ denotes a low-pass filter such as Gaussian filters or box mean filters, and $\lambda$ is the amount coefficient which controls the volume of enhancement achieved at the output.

The main workflow is to generate a blurred version of the original image, then compute the difference between the original and the blurred image. The difference represents the edge information of the image. By comparing and reinforcing the contrast between the edges in the blurred image and original image, the edges become more visually prominent. Finally, this enhanced edge information will be added back to the original image to get a sharpened output.

In our implementation, we apply Unsharp Masking with fixed parameters chosen to effectively enhance motion-blurred images in our dataset. Specifically, we use a Gaussian blur with a fixed standard deviation ($\sigma = 2.0$) and a fixed sharpening amount ($\lambda = 1.5$), which were selected empirically to balance sharpening strength and noise control.

**Algorithm 5: Blur Enhancement via Unsharp Masking.** *Input:* Image folder $\mathcal{D}$, fixed parameters $(\sigma, \lambda)$ *Output:* Enhanced images saved to output folder

```
1. For each image $I$ in folder $\mathcal{D}$:
    (a) Read $I$ in BGR format. If reading fails, skip this image.
    (b) Apply a Gaussian blur to $I$ with standard deviation $\sigma$ to produce the blurred image $I_{\text{blur}}$.
    (c) Compute the detail mask:
            $M = I - I_{\text{blur}}$
    (d) Generate the sharpened output:
            $\hat{I} = I + \lambda \cdot M$
    (e) Save $\hat{I}$ to the output folder.
```

Unsharp Masking with these fixed parameters enhances edge contrast and structural details in motion-blurred images, improving visibility of object boundaries for detection without requiring additional model training.

### *3.1.6. Blur - DeblurGAN*

DeblurGAN [36] is a learning-based blind motion deblurring approach that restores a sharp image from a single blurred input using a conditional GAN. Given a blurry image $I_b$, the generator $G$ predicts a restored image $\hat{I}_s$:

$$\hat{I}_s = G(I_b).$$

In DeblurGAN, the generator is implemented as a ResNet-based encoder–decoder with residual blocks. In our repo, the default generator choice is a 9-block ResNet with two downsampling layers, nine ResNet blocks, and two upsampling layers, followed by a `tanh` output layer. We enable residual learning, so the network predicts a residual $R(I_b)$ and adds it back to the input:

$$\hat{I}_s = \text{clip}\big(I_b + R(I_b),\, -1,\, 1\big),$$

where clipping keeps the output within the normalized range.

During training, DeblurGAN optimizes a combination of an adversarial loss and a perceptual (content) loss. The adversarial component is implemented with WGAN-GP to improve training stability, while the perceptual loss is computed on VGG-19 feature activations to encourage restoration of visually meaningful structures:

$$\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \lambda \mathcal{L}_{\text{perc}}.$$

In our project pipeline, we use a pre-trained DeblurGAN generator.

**Algorithm 6: Blur Enhancement via DeblurGAN.**
*Input:* Image folder $\mathcal{D}$, pre-trained generator weights $G$
*Output:* Deblurred images saved to output folder

```
1. Load pre-trained DeblurGAN generator $G$ with residual learning enabled.
2. For each image $I$ in folder $\mathcal{D}$:
    (a) Read $I$ and convert to RGB.
    (b) Preprocess (no resizing/cropping):
        i. Convert to tensor: $I \in [0, 1]$.
        ii. Normalize to $[-1, 1]$: $I \leftarrow (I - 0.5)/0.5$.
    (c) Forward pass: $\hat{I} \leftarrow G(I)$.
    (d) Postprocess:
        i. Denormalize to $[0, 255]$: $\hat{I} \leftarrow (\hat{I}+1)/2 \cdot 255$.
        ii. Convert to uint8 image and save outputs.
```

DeblurGAN improves motion-blurred images by learning a data-driven restoration mapping, enhancing edges and recovering textures that are difficult to restore using purely hand-crafted sharpening filters, thereby improving object boundary clarity for downstream detection.

### *3.1.7. Detection Model - YOLOv11 Series*

We adopt the Ultralytics YOLOv11 series as the object detection backbone in our pipeline, and evaluate two model scales, *YOLOv11n* and *YOLOv11s*, to capture the efficiency–accuracy trade-off under domain shift. YOLO models follow a one-stage detection paradigm that performs feature extraction and bounding box regression in a single forward pass, making them well-suited for real-time autonomous driving settings. Given an input image, the detector predicts a set of bounding boxes along with objectness and class probabilities, and applies Non-Maximum Suppression (NMS) to remove redundant predictions.

In our workflow (Fig. 1), YOLO is used as a fixed downstream detector to probe whether image enhancement can improve robustness *without* modifying the detection network. Concretely, for each degraded domain (fog, night, blur), we (1) evaluate the clean-trained YOLO model directly on the raw degraded images, and (2) evaluate it on the enhanced images produced by each restoration method. Importantly, enhancement is applied *offline* and does not change the annotation files, enabling a controlled comparison where only the input image distribution is altered.

To further quantify the benefit of domain adaptation, we also fine-tune YOLOv11n and YOLOv11s on each target degraded dataset using the same label format and evaluation metrics as the baseline. This yields four detector settings in total: YOLOv11n / YOLOv11s (clean-trained) and YOLOv11n (FT) / YOLOv11s (FT). Throughout all experiments, we keep the inference resolution fixed at 640 with letterbox resizing, and report detection performance using mAP50 and $\text{mAP}_{50\text{-}95}$ to measure both coarse detection accuracy and localization quality.

## 4. EVALUATIONS

### 4.1. Experimental Setups

#### *4.1.1. Dataset Collection*

In this project, we constructed a set of degraded datasets to evaluate object detection performance under challenging real-world conditions. While our clean daytime baseline model is trained on the KITTI dataset [37], we additionally created a custom distorted dataset containing foggy, nighttime, and motion-blurred scenes for both fine-tuning and evaluation.

To obtain realistic samples, we collected street-level images and extracted key frames from videos recorded around the UCSD GFH area. All images were resized to match the resolution used in our baseline YOLO training setup. We then performed annotation using Roboflow's SAM 3 segmentation tool [38] to auto-generate object masks. These masks were manually corrected to remove inaccurate regions and add missing objects. After verification, the masks were converted into YOLO-format bounding boxes $(x\text{-}center, y\text{-}center, width, height)$, and each annotated image was visually inspected to ensure alignment and label quality.

For the fog domain, we adopted a depth-guided synthetic fog generation pipeline to produce consistent and physically plausible haze conditions. Using the MiDaS DPT-Hybrid model [39], we first estimated a dense depth map for each daytime image. The predicted depth was normalized and inverted to represent fog accumulation at increasing distances. Then a fog layer with a defined atmospheric light color was blended with the original image using a depth-dependent scattering coefficient, with the fog strength randomly sampled within a preset range. This process allowed us to generate foggy scenes with controlled densities while preserving the geometry of the underlying scene.

For the night domain, we captured nighttime traffic and street environments in several different locations, including residential areas, intersections, and parking zones on the road. These scenes offered varied illumination conditions, such as streetlights, vehicle headlights, and low-light corners. The diversity of environments introduces significant contrast loss and noise, providing a realistic benchmark for evaluating low-light enhancement methods and the effectiveness of fine-tuning under nighttime conditions.

For the blur domain, we applied synthetic motion blur to all daytime images, since our iPhone camera did not naturally produce motion blur even under intentional shaking or panning. By generating motion blur digitally, we were able to control the severity of the blur and ensure consistency between samples. This approach allowed us to create a broad range of motion blur distortions that remain suitable for evaluating deblurring algorithms and domain adapted models.

When combined, these three distorted domains form a diverse and representative dataset for assessing how image enhancement methods and domain specific fine-tuning influence YOLO-based object detection in degraded driving environments. This dataset serves as the foundation for our comparative study of preprocessing versus fine-tuning strategies under foggy, low-light, and blurry conditions.

#### *4.1.2. Settings on YOLO*

**Model variants.** We adopt two YOLO backbones to study the trade-off between efficiency and accuracy: *YOLOv11n* and *YOLOv11s* Ultralytics YOLO. Unless otherwise stated, both models share the same detection head design and are trained/evaluated under identical protocols for fair comparison.

**Input preprocessing.** All datasets are formatted in YOLO annotation style $(x\text{-}center, y\text{-}center, width, height)$. Images are resized to 640 with letterboxing to preserve aspect ratio. For enhanced datasets (DCP/FFA-Net, CLAHE/Zero-DCE, Unsharp Mask/DeblurGAN), we apply enhancement *offline* to generate a new image set, while keeping the original labels unchanged.

**Baseline training on clean daytime data.** We train a baseline detector on the clean daytime dataset (KITTI) using COCO-pretrained. Training is performed for 10 epochs with batch size 64, optimizer SGD, initial learning rate $lr_0 = 5e - 5$, and weight decay $wd = 1e - 4$. We use a learning rate schedule of cosine. Common data augmentation includes `mosaic, hsv jitter, random affine, flip, scale`; augmentation is disabled during validation and testing.

**Evaluation protocol.** We report detection performance using mean Average Precision at IoU 0.5 (mAP50) and averaged across IoU thresholds 0.50:0.95 ($\text{mAP}_{50\text{-}95}$), consistent with standard object detection benchmarks. All results in Table 1 are computed on the same held-out test split for each domain.

**Inference on distorted and enhanced datasets (no retraining).** To isolate the effect of preprocessing, we directly evaluate the clean-trained baseline YOLO on: (1) raw distorted images (fog/night/blur), and (2) their enhanced versions produced by each image processing method. This setting tests whether enhancement alone improves robustness without introducing additional labeled data or domain adaptation.

**Table 1**. Performance comparison of YOLOv11n and YOLOv11s under adverse environmental conditions with various image enhancement methods. The table evaluates detection accuracy (mAP50 and mAP$_{50-95}$) across three degradation scenarios: blurred, low-light, and foggy. FT denotes models fine-tuned on the target dataset. We compare the baseline (unprocessed) performance against traditional algorithms (e.g., Unsharp Masking, CLAHE, DCP) and deep learning-based restoration networks (e.g., DeblurGAN, Zero-DCE, FFA-Net).

| **Scenario / Method** | **YOLOv11n** | | **YOLOv11n (FT)** | | **YOLOv11s** | | **YOLOv11s (FT)** | | **Mean** |
|---|---|---|---|---|---|---|---|---|---|
| | mAP50 | mAP$_{50-95}$ | mAP50 | mAP$_{50-95}$ | mAP50 | mAP$_{50-95}$ | mAP50 | mAP$_{50-95}$ | |
| Clean Baseline | 38.5 | 20.6 | 59.2 | 35.3 | 46.8 | 24.3 | 66.0 | 41.8 | 41.6 |
| *Deblurring* | | | | | | | | | |
| None (Blurred Input) | 26.6 | 12.5 | 54.0 | 31.4 | 34.0 | 16.7 | 59.5 | 36.7 | 33.9 |
| Unsharp Masking | 23.9 | 10.5 | 54.6 | 31.2 | 30.8 | 14.2 | 60.7 | 36.7 | 32.8 |
| DeblurGAN | **33.2** | **17.1** | **56.8** | **33.5** | **41.7** | **22.0** | **61.1** | **38.5** | **38.0** |
| *Enhancement for Nighttime* | | | | | | | | | |
| None (Low-Light Input) | **20.4** | **10.6** | **57.5** | **30.7** | **22.3** | **12.0** | **64.7** | **37.3** | **31.9** |
| CLAHE | 17.0 | 8.0 | 54.6 | 30.1 | 21.7 | 11.3 | 60.0 | 33.7 | 29.6 |
| Zero-DCE | 16.3 | 8.2 | 54.7 | 29.1 | 20.9 | 10.8 | 59.4 | 33.9 | 29.2 |
| *Defogging* | | | | | | | | | |
| None (Foggy Input) | 35.4 | 18.9 | **58.3** | 35.4 | 44.0 | 22.1 | 62.2 | 39.8 | 39.5 |
| DCP | **38.2** | 20.3 | 57.3 | **35.5** | **46.3** | **24.6** | 62.3 | 40.0 | **40.6** |
| FFA-Net | 38.1 | **20.4** | 56.3 | 35.1 | 45.3 | 24.0 | **62.9** | **40.7** | 40.4 |

**Fine-tuning on target domains.** For domain adaptation, we fine-tune the clean baseline weights on each distorted dataset (and report as **YOLOv11n (FT) / YOLOv11s (FT)**). Fine-tuning uses a smaller learning rate $lr_{\text{ft}} = 1e-5$, runs for 10 epochs, and follows the same augmentation and evaluation pipeline as baseline training.

### 4.2. Results

Table 1 summarizes detection performance (mAP50 / mAP$_{50\text{-}95}$) of two YOLO variants under three degraded driving conditions (blur, low-light, fog), comparing raw degraded inputs, image-enhanced inputs, and domain-specific fine-tuning (FT).

**Overall impact of domain shift.** Compared to the clean baseline, all degradations reduce detection accuracy, with *low-light* causing the largest drop. For example, YOLOv11s decreases from 46.8/24.3 (clean) to 22.3/12.0 (low-light), while blur and fog reduce performance to 34.0/16.7 and 44.0/22.1, respectively.

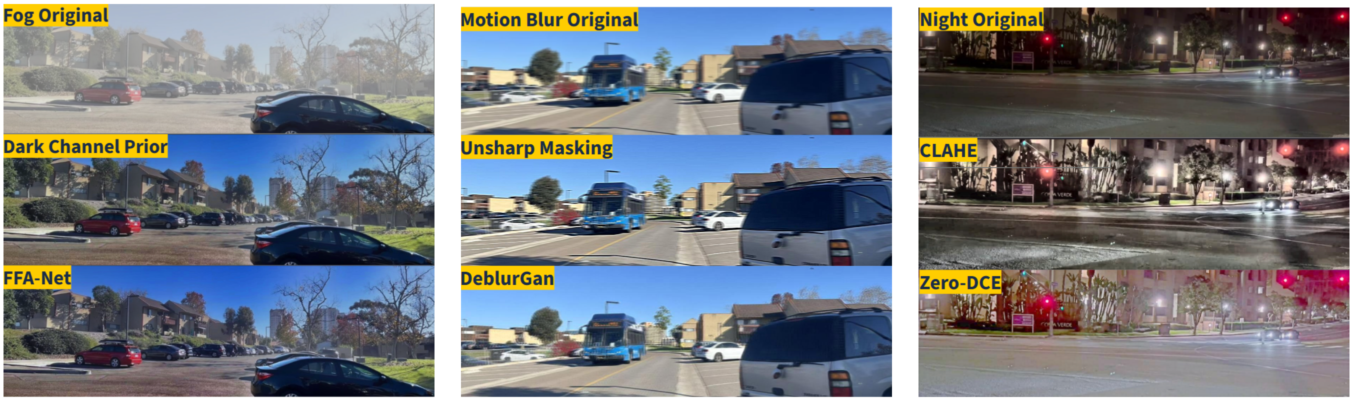
**Effect of image enhancement without retraining.** Enhancement improves detection in *blur* and *fog* but is not consistently beneficial for *low-light*. For motion blur, DeblurGAN provides the strongest gain over the raw blurred input across both model sizes, while Unsharp Masking does not help and often degrades performance. For fog, both DCP and FFA-Net yield modest but consistent improvements over raw foggy inputs. In contrast, under low-light, both CLAHE and Zero-DCE underperform the raw low-light input in our setting, suggesting that brightness/contrast enhancement may introduce distribution shifts or artifacts that do not align with the detector's learned features.

**Effect of fine-tuning.** Fine-tuning on the target domain substantially improves performance across all three degradations, and yields the largest absolute gains under low-light. Fine-tuning also strongly improves blur and fog, indicating that domain adaptation is the most reliable way to recover robustness when labeled target data is available.

**Best-performing methods per scenario.** Among enhancement only approaches, DeblurGAN achieves the best performance in the blur scenario, while DCP and FFA-Net are both competitive for fog (with DCP slightly higher on the mean score in Table 1). For low-light, the best performance is obtained by using the raw input combined with fine-tuning, rather than applying enhancement.

### 4.3. Comparison

Table 1 presents a comprehensive comparison of YOLOv11n and YOLOv11s under three adverse environmental conditions—motion blur, low-light, and fog—using both classical image enhancement methods and learning-based restoration approaches. We evaluate performance using mAP@50 and mAP@50–95, and include both pretrained models and models fine-tuned (FT) on the target dataset.



(a) Foggy scenes

(b) Motion-blurred scenes

(c) Nighttime scenes

**Fig. 2**. Qualitative comparisons across different scenarios. (a) Foggy scenes: Input, DCP, and FFA-Net. (b) Motion-blurred scenes: Input, Unsharp Masking, and DeblurGAN. (c) Nighttime scenes: Input, CLAHE, and Zero-DCE. All comparisons show results arranged from top to bottom within each sub-image.

### *4.3.1. Baseline Performance*

Across all scenarios, detection performance drops significantly when using degraded inputs compared to the clean baseline, highlighting the sensitivity of object detectors to adverse visual conditions. Fine-tuning consistently improves performance across all settings, particularly for YOLOv11s, indicating that larger models benefit more from domain adaptation. Nevertheless, even with fine-tuning, performance under adverse conditions remains below the clean baseline, motivating the use of image enhancement and restoration techniques.

### *4.3.2. Motion Blur*

Under motion blur, the unprocessed blurred input leads to a substantial performance degradation. Classical Unsharp Masking provides only marginal or inconsistent improvements and in some cases further degrades detection accuracy, suggesting that simple edge enhancement is insufficient for severe motion blur. In contrast, DeblurGAN achieves the strongest performance among deblurring methods, yielding notable gains in both mAP@50 and mAP@50–95 across all model variants. This indicates that learning-based deblurring is more effective at restoring object-relevant structures than traditional sharpening methods, especially for fine-grained localization metrics.

### *4.3.3. Nighttime Enhancement*

In the low-light setting, the unprocessed nighttime input (“None”) achieves the best detection performance across all evaluated detector configurations. Both CLAHE and Zero-DCE consistently reduce mAP@50 and mAP@50–95, indicating that these enhancement methods do not improve—and may even harm—detection under our real nighttime data distribution. CLAHE may amplify noise and local artifacts in dark regions, while Zero-DCE can change the intensity distribution and color/contrast in ways that distort object appearance. As a result, the enhanced images become less aligned with the detector’s learned feature statistics, leading to decreased detection accuracy.

### *4.3.4. Foggy Conditions*

In foggy scenes, both DCP and FFA-Net improve detection accuracy compared to the unprocessed foggy input. DCP achieves consistent gains across all models, particularly in mAP@50–95, indicating improved localization accuracy after physically grounded dehazing. FFA-Net attains comparable or slightly higher performance in mAP@50 for some configurations, especially for YOLOv11s with fine-tuning, but its gains are less consistent across models. This suggests that while learning-based dehazing can effectively restore visibility, its benefits depend on model capacity and training conditions. Overall, DCP provides a strong and robust baseline, while FFA-Net offers competitive performance with slightly higher variance.

### *4.3.5. Summary of Strengths and Weaknesses*

In summary, classical enhancement methods such as CLAHE and DCP offer stable, training-free improvements with low computational overhead, making them reliable baselines for adverse conditions. Learning-based restoration methods, including DeblurGAN and FFA-Net, generally achieve higher peak performance but exhibit greater variability and depend more strongly on model size and fine-tuning. These results highlight a trade-off between robustness and performance gain: classical methods are simple and consistent, while deep restoration models can provide larger improvements when appropriately matched with detector capacity and training strategy.

### 4.4. Discussion

Our experimental results highlight that different types of environmental degradation affect object detection performance in fundamentally different ways, and that the effectiveness of image enhancement strongly depends on the underlying degradation characteristics.

First, for motion blur, learning-based restoration clearly shows its advantage. DeblurGAN consistently outperforms both the raw blurred input and classical unsharp masking across all model configurations. This indicates that motion blur removes fine-grained structural information that simple edge enhancement cannot recover, while deep deblurring models are able to reconstruct more detection-relevant features. In this scenario, visual restoration and detection objectives are well aligned.

In contrast, nighttime enhancement exhibits an opposite and somewhat counterintuitive behavior. Both CLAHE and Zero-DCE lead to worse detection performance compared to using the raw low-light input. This suggests that nighttime degradation in our dataset is not purely a low-illumination problem. Since images are captured manually at night, additional motion blur from camera shake and strong headlight over-exposure are common. Enhancement methods that focus on brightness or contrast may amplify noise, distort textures, or saturate already over-exposed regions, resulting in inputs that look visually clearer but are less compatible with the detector's learned feature representations.

For foggy scenes, both DCP and FFA-Net achieve very similar detection performance. Although dehazing significantly improves visual quality, the corresponding gains in mAP are relatively modest. This implies that while fog reduces contrast globally, many object cues remain detectable even in the degraded input.

Across all degradation scenarios, fine-tuning consistently outperforms pre-processing alone. Fine-tuned models achieve higher mAP than any enhanced input evaluated using the baseline detector. Fine-tuning allows the detector to learn distortion-specific features and decision boundaries, rather than relying on visually enhanced inputs that may introduce distribution shifts.

Overall, our results suggest that image enhancement is not universally beneficial for object detection. While restoration can be effective for certain degradations such as motion blur, it may hurt performance when the enhancement alters color and texture statistics in ways that conflict with the detector's learned representation. These findings emphasize the importance of aligning enhancement objectives with downstream tasks, and suggest that model adaptation is often a more reliable strategy than preprocessing alone.

## 5. FUTURE WORK

While our current work has constructed degraded datasets for foggy, nighttime, and motion blur conditions and evaluated YOLO-based object detection under these distortions, there remain several directions worth exploring in future research.

First, we plan to expand dataset and distortion domain coverage. The current dataset is limited in scale and environmental diversity. Collecting larger and more diverse real-world data, as well as incorporating additional types of distortion (e.g., rain, snow, glare), would improve the robustness and applicability of our evaluation. A larger dataset will also provide more reliable benchmarks for comparing detection and enhancement strategies.

Second, we intend to explore multi domain fine-tuning instead of training separate models for each distortion domain. By jointly fine-tuning across multiple domains, we aim to learn a shared representation that generalizes better under mixed or unseen conditions. This may also allow us to combine different image enhancement techniques with domain adaptation to achieve hybrid performance gains.

Lastly, to support real-world deployment, we plan to investigate real-time performance and efficiency. This includes evaluating latency and computational cost of both enhancement and detection models on edge or embedded platforms typical of autonomous systems.

By pursuing these extensions, we hope to further advance the understanding of how object detection systems perform under challenging visual degradations and develop more robust perception pipelines for real-world driving scenarios.

## 6. CONCLUSION

This paper evaluates the robustness of YOLO-based 2D object detection under three adverse driving conditions—fog, low-light, and motion blur—by comparing test-time image enhancement with domain-specific fine-tuning. Using YOLOv11n and YOLOv11s trained on clean daytime data, we test performance on degraded datasets and their enhanced versions, and further fine-tune the detectors on each target domain to quantify the benefit of adaptation.

Overall, fine-tuning provides the most consistent and largest performance improvements across all degradations when labeled target data is available. Enhancement-only preprocessing offers a lightweight alternative with selective benefits: deblurring (DeblurGAN) and dehazing (DCP/FFA-Net) improve detection under blur and fog, while low-light enhancement (CLAHE, Zero-DCE) does not yield gains in our setting. These results suggest a practical guideline for deployment: use enhancement when retraining is infeasible, and prioritize fine-tuning when annotations and compute budgets permit.

## 7. TEAM EFFORT

For this project, the workload was evenly distributed among three members of the group. Each of us contributed equally to all major aspects of the project, including dataset collection and preprocessing, implementation of image enhancement algorithms, model training and evaluation, analysis of experimental results, and writing of the final report.

The percentage breakdown of effort for each member is as follows:

1. Xuanhao Zhu: 33.3%
2. Xinhao Xu: 33.3%
3. Yuanzhe Hu: 33.3%

Each member was responsible for significant project components and the tasks were collaboratively coordinated to ensure balanced participation and shared understanding of all stages of the work.

## 8. PROJECT REPOSITORY

The codebase and supplementary materials for this project are publicly available on GitHub. The repository includes all used dataset, relevant scripts, data preprocessing code, model training and evaluation files, as well as documentation for reproducing the results presented in this report:

**GitHub Repository:**
https://github.com/N1nomae/ECE-253-XXY

## 9. REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[2] Kaiming He, Jian Sun, and Xiaoou Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[3] Jean-Philippe Tarel and Nicolas Hautiere, "Fast visibility restoration from a single color or gray level image," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2201–2208.

[4] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng, "Aod-net: All-in-one dehazing network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4770–4778.

[5] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7314–7323.

[6] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia, "Ffa-net: Feature fusion attention network for single image dehazing," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 11908–11915.

[7] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang, "Multi-scale boosted dehazing network with dense feature fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2157–2167.

[8] Christos Sakaridis, Dengxin Dai, and Luc Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018.

[9] Christos Sakaridis, Dengxin Dai, and Luc Van Gool, "Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10765–10775.

[10] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11682–11692.

[11] Xin Yang, Michael Bi Mi, Yuan Yuan, Xin Wang, and Robby T Tan, "Object detection in foggy scenes by embedding depth and reconstruction into domain adaptation," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 1093–1108.

[12] Xiaojie Guo, Yu Li, and Haibin Ling, "Lime: Low-light image enhancement via illumination map estimation," *IEEE Transactions on image processing*, vol. 26, no. 2, pp. 982–993, 2016.

[13] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu, "Deep retinex decomposition for low-light enhancement," *arXiv preprint arXiv:1808.04560*, 2018.

[14] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong, "Zero-reference deep curve estimation for low-light image enhancement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1780–1789.

[15] Yuen Peng Loh and Chee Seng Chan, "Getting to know low-light images with the exclusively dark dataset," *Computer vision and image understanding*, vol. 178, pp. 30–42, 2019.

[16] Shijie Hao, Zhonghao Wang, and Fuming Sun, "Ledet: a single-shot real-time object detector based on low-light image enhancement," *The Computer Journal*, vol. 64, no. 7, pp. 1028–1038, 2021.

[17] Dai Quoc Tran, Armstrong Aboah, Yuntae Jeon, Maged Shoman, Minsoo Park, and Seunghee Park, "Low-light image enhancement framework for improved object detection in fisheye lens datasets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7056–7065.

[18] Haifeng Guo, Tong Lu, and Yirui Wu, "Dynamic low-light image enhancement for object detection via end-to-end training," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5611–5618.

[19] Linwei Ye, Dong Wang, Dongyi Yang, Zhiyuan Ma, and Quan Zhang, "Velie: a vehicle-based efficient low-light image enhancement method for intelligent vehicles," *Sensors*, vol. 24, no. 4, pp. 1345, 2024.

[20] Che-Tsung Lin, Sheng-Wei Huang, Yen-Yi Wu, and Shang-Hong Lai, "Gan-based day-to-night image style transfer for nighttime vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 951–963, 2020.

[21] Zhipeng Du, Miaojing Shi, and Jiankang Deng, "Boosting object detection with zero-shot day-night domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12666–12676.

[22] Leonid I Rudin, Stanley Osher, and Emad Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[23] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman, "Removing camera shake from a single photograph," in *Acm Siggraph 2006 Papers*, pp. 787–794. 2006.

[24] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8174–8182.

[25] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8878–8887.

[26] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun, "Simple baselines for image restoration," in *European conference on computer vision*. Springer, 2022, pp. 17–33.

[27] Xintian Mao, Qingli Li, and Yan Wang, "Adarevd: Adaptive patch exiting reversible decoder pushes the limit of image deblurring," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 25681–25690.

[28] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan, "Promptir: Prompting for all-in-one image restoration," *Advances in Neural Information Processing Systems*, vol. 36, pp. 71275–71293, 2023.

[29] Shangquan Sun, Wenqi Ren, Tao Wang, and Xiaochun Cao, "Rethinking image restoration for object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4461–4474, 2022.

[30] Yongzhen Wang, 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*. Wiley Online Library, 2022, vol. 41, pp. 465–476.

[31] Hanzhao Wang, Chunhua Hu, Weijie Qian, and Qian Wang, "Rt-deblur: Real-time image deblurring for object detection," *The Visual Computer*, vol. 40, no. 4, pp. 2873–2887, 2024.

[32] Thekke Madam Nimisha, Akash Kumar Singh, and Ambasamudram N Rajagopalan, "Blur-invariant deep learning for blind-deblurring," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4752–4760.

[33] Rong Zou, Marc Pollefeys, and Denys Rozumnyi, "Retrieval robust to object motion blur," in *European Conference on Computer Vision*. Springer, 2024, pp. 251–268.

[34] Kaiming He, Jian Sun, and Xiaoou Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[35] Zenglin Shi, Yunlu Chen, Efstratios Gavves, Pascal Mettes, and Cees G.M. Snoek, "Unsharp mask guided

filtering," *IEEE Transactions on Image Processing*, vol. 30, pp. 478–491, 2021.

[36] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," 2018.

[37] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[38] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer, "Sam 3: Segment anything with concepts," 2025.

[39] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun, "Vision transformers for dense prediction," *CoRR*, vol. abs/2103.13413, 2021.