ENHANCING OBJECT DETECTION PERFORMANCE ON DEGRADED IMAGES: A COMPARATIVE STUDY OF IMAGE PROCESSING VS. MODEL FINE-TUNING

Xin Zhao, Cheng Qian, Pengfei Zhan

Department of Electrical and Computer Engineering
University of California San Diego
La Jolla, CA 92093, USA

ABSTRACT

Object detection systems can lose substantial accuracy when images are captured in difficult conditions. This paper studies two practical ways to recover performance under three common degradation types: extreme low-light conditions, motion blur, and Gaussian noise. For each degradation type, we compare classic image processing methods against deep learning-based model fine-tuning. Our dataset contains 110 manually annotated real images (50 clean, 30 low-light, 30 motion-blurred) of three everyday objects (banana, apple, orange), plus an additional synthesized Gaussian-noise set (39 images, $\sigma = 30$). We evaluate gamma correction with CLAHE for low-light, unsharp masking with edge enhancement for motion blur, and non-local means denoising with bilateral filtering for Gaussian noise against fine-tuning a pretrained SSD300-VGG16 detector. Traditional enhancement is not always helpful: for extreme low-light, preprocessing reduced mAP from 0.017 to 0.000. Fine-tuning consistently outperformed traditional methods, improving mAP by +0.702, +0.236, and +0.494 for low-light, motion blur, and Gaussian noise, respectively. Overall, the results suggest that adapting the detector is more reliable than attempting to "fix" severely compromised images with standalone preprocessing.

1. INTRODUCTION

Object detectors work well in controlled settings with sharp, well-lit images. In the wild, though, cameras capture plenty of "messy" frames—dim scenes, handheld blur, and sensor noise—where detection accuracy and confidence can drop sharply, leading to missed objects.

This robustness problem matters most on mobile and edge devices, where users take photos in uncontrolled environments and compute budgets are tight. For example, an inventory or assistive app has to cope with different lighting, varying camera stability, and inconsistent framing, not just the clean images seen during training.

1.1. Problem Statement

Our project asks a simple question with practical consequences: when the input is degraded, is it better to improve the image through preprocessing, or to adapt the detector so it can handle low-quality inputs directly? We focus on three degradation types that commonly appear in mobile photography:

1. Extreme Low-Light Degradation: Images captured in very dark conditions (ISO 5280–12500) where most scene information is obscured by darkness, often showing severe underexposure, high noise levels, and loss of color information.

2. Motion Blur Degradation: Images affected by camera shake during exposure at high shutter speeds (1/8000s), producing directional blur that washes out edges and fine details that detectors rely on.

3. Gaussian Noise Degradation: Synthetic additive noise ($\sigma = 30$) that simulates sensor noise and compression artifacts commonly encountered in low-quality imaging scenarios.

For each degradation type, we compare two approaches: traditional image enhancement (gamma correction, sharpening, denoising) versus deep learning-based fine-tuning of a pretrained detection model.

1.2. Key Contributions

Our main contributions are:

(1) Comprehensive Degradation Analysis: We systematically evaluate three distinct degradation types with dedicated processing algorithms for each, showing that different degradations call for fundamentally different treatments.

(2) Negative Result Discovery: We show that traditional enhancement can hurt detection on extreme degradations—a counterintuitive result that challenges the assumption that preprocessing always helps.

(3) Rigorous Evaluation Methodology: By using COCO-recognizable objects, we can report standard metrics (mAP, precision, recall) on 110 manually annotated real images (plus 39 synthesized noisy images), enabling reliable comparisons.

(4) Ablation Studies: We run parameter sensitivity analyses for each traditional algorithm, revealing that more aggressive enhancement often makes performance worse.

(5) Practical Deployment Insights: We compare accuracy gains against computational costs to help guide real-world deployment decisions.

2. RELATED WORKS

2.1. Image Enhancement for Object Detection

Image preprocessing has a long history in computer vision, and many pipelines still use it as a first line of defense against poor image quality. Pizer et al. [1] introduced Adaptive Histogram Equalization (AHE) and its contrast-limited variant (CLAHE), which remain popular for boosting local contrast, especially under uneven illumination. These methods are fast and easy to deploy, but they can also amplify noise in regions where the signal is already weak.

Bilateral filtering, introduced by Tomasi and Manduchi [2], performs edge-preserving smoothing that reduces noise while keeping important boundaries intact—an attractive property for detectors that depend on edge and texture cues. In practice, its effectiveness depends on careful parameter choices.

More recently, Lore et al. [3] explored deep learning-based enhancement for low-light images and reported gains in downstream detection. Their focus, however, is largely on specific degradation scenarios rather than the mix of degradations we study.

2.2. Deep Learning Approaches to Degraded Images

Chen et al. [4] proposed learning-based enhancement targeted at extreme low-light, training networks to "see in the dark." While effective, such approaches typically require specialized training data and do not directly generalize to other degradations without additional retraining.

For detection, we build on the SSD (Single Shot Multibox Detector) architecture by Liu et al. [5], chosen for its practical speed/accuracy trade-off. Although SSD300-VGG16 has been widely evaluated on high-quality benchmarks such as COCO, its behavior under severely degraded inputs is less well characterized.

2.3. Fine-tuning Strategies

Transfer learning and fine-tuning are widely used for domain adaptation in object detection. Oquab et al. [6] showed that fine-tuning pretrained CNNs with limited domain-specific data can yield meaningful gains by preserving general visual priors while adapting to the target domain.

At the same time, fine-tuning introduces well-known pitfalls: catastrophic forgetting, overfitting on small datasets, and sensitivity to learning rates and layer-freezing strategies. Wang et al. [7] discussed few-shot detection techniques that mitigate some of these issues, but often at the cost of additional training complexity.

2.4. Comparison of Approaches

Direct head-to-head comparisons between image enhancement and model adaptation for degraded inputs are relatively rare. Noh et al. [8] studied detection under adverse weather and found that model adaptation often beats preprocessing, but their degradations are weather-specific rather than the more general image-quality degradations we target.

Our work helps close this gap by comparing both approaches under controlled degradation conditions, complementing the main results with ablation studies and deployment-focused considerations. Instead of narrow domains, we use everyday objects and multiple degradation settings that better resemble real-world mobile photography.

3. METHOD

3.1. Overall Framework

We evaluate two families of solutions for improving detection on degraded images: (i) standalone image processing tailored to each degradation type and (ii) fine-tuning a pretrained detector on degraded data. Figure 1 summarizes the end-to-end pipeline, from data collection and annotation to evaluation.

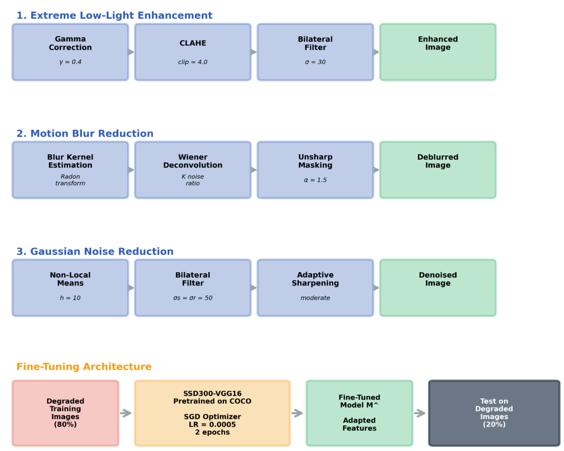

Fig. 1. Experimental workflow showing parallel evaluation of traditional processing versus fine-tuning approaches for three degradation types.

3.2. Dataset Collection and Preparation

3.2.1. Data Collection Protocol

We built a compact but carefully controlled dataset using an iPhone 15 Pro (24mm lens, f/1.8 aperture). We captured three object categories (banana, apple, orange) under clean conditions and under three degraded conditions. These categories appear in the COCO dataset, which allows us to report standard detection metrics. Table 1 summarizes the dataset composition.

Clean Data (50 images): Captured under strong indoor or daylight illumination with stable camera hold and uncluttered backgrounds. Objects occupy 10–40% of the frame and are in clear focus. We photographed each class from multiple viewpoints to encourage diversity.

Low-Light Data (30 images): Captured in extreme darkness requiring ISO 5280. These images are heavily underex-

Table 1. Dataset Specifications and Camera Settings

Condition	Images	Camera Settings
Clean	50	1/60s, f/1.8, ISO64
Low-light	30	1/30s, f/1.8, ISO5280
Motion blur	30	1/8000s, f/1.8, ISO5280
Gaussian noise	39	Synthesized ($\sigma = 30$)
Total (real)	**110**	
Total (incl. synthesized noise)	**149**	

posed (mean brightness below 10/255), with noticeable noise and reduced color fidelity.

Motion Blur Data (30 images): Captured with intentional camera shake at 1/8000s shutter speed combined with high ISO. The resulting directional blur degrades edges and fine details.

Gaussian Noise Data (39 images): Synthesized by adding Gaussian noise ($\sigma = 30$) to a subset of clean images. We reuse the original PASCAL VOC annotations by copying the corresponding XML files.

3.2.2. Annotation and Ground Truth

Because our objects are COCO-recognizable, we can evaluate class-specific detection performance with standard metrics (rather than generic objectness scores).

Annotation Tool: We used Roboflow to manually annotate bounding boxes for all objects in each image. Annotations were exported in PASCAL VOC format (XML files) containing bounding box coordinates and class labels.

Ground Truth and Labels: All real images are manually annotated in PASCAL VOC XML format. For the synthesized noisy set, we copy the XML annotation from the source clean image so that bounding boxes and labels remain identical across clean/noisy pairs.

Evaluation Metrics:

(1) mAP (mean Average Precision): Primary metric, computed at IoU threshold 0.5

(2) Precision: Proportion of correct detections among all predictions

(3) Recall: Proportion of ground truth objects successfully detected

(4) Per-class AP: Individual performance for banana, apple, and orange

We set the detection confidence threshold to 0.3 (lower than the typical 0.5) to accommodate degraded images while still filtering obvious false positives.

3.3. Detection Model Architecture

We use SSD300-VGG16 pretrained on the COCO dataset as our base detector:

(1) Backbone: VGG16 convolutional layers for feature extraction

(2) Detection Head: Multi-scale feature maps (38×38, 19×19, 10×10, 5×5, 3×3, 1×1)

(3) Input Size: 300×300 pixels

(4) Output: Bounding boxes, class labels, confidence scores

(5) Classes: COCO class 52 (banana), 53 (apple), 55 (orange)

This pretrained model provides a consistent baseline for measuring how each degradation affects performance and how much each mitigation strategy recovers.

3.4. Traditional Image Processing Algorithms

3.4.1. Algorithm I: Extreme Low-Light Enhancement

For severe underexposure, we use a three-stage pipeline:

Stage 1: Gamma Correction

Nonlinear brightness adjustment that preferentially lifts dark regions:

$$I_{out} = 255 \times \left(\frac{I_{in}}{255}\right)^{\gamma} \tag{1}$$

where $\gamma = 0.4$ provides aggressive brightening. We tested four gamma values: $\{0.2, 0.3, 0.4, 0.5\}$ in ablation studies.

Stage 2: CLAHE

Applied in LAB color space to enhance local contrast:

(1) Convert RGB to LAB color space

(2) Apply CLAHE to L channel (clip limit tested: $\{2.0, 4.0, 6.0, 8.0\}$, tile size = 8×8)

(3) Merge channels and convert back to RGB

extbfNote: Our final low-light pipeline applies only Stage 1 (gamma correction) and Stage 2 (CLAHE), matching the implementation used for all reported results.

3.4.2. Algorithm II: Motion Blur Reduction

For motion blur, we use sharpening-based enhancement:

Stage 1: Unsharp Masking

Boost edges to counteract blur:

$$I_{sharp} = \alpha \cdot I + (1 - \alpha) \cdot G_\sigma(I) \tag{2}$$

where $\alpha \in \{1.2, 1.5, 1.8, 2.0\}$ is the sharpening weight and G_σ is Gaussian blur with $\sigma = 2.0$.

Stage 2: Edge Enhancement

Apply edge enhancement kernel:

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3}$$

Blend with the sharpened image using weight $\beta \in \{0.1, 0.3, 0.5, 0.7\}$.

3.4.3. Algorithm III: Gaussian Noise Reduction

For additive noise, we apply denoising techniques:

Non-Local Means Denoising

Apply NLM denoising with parameters:

(1) Filter strength $h \in \{5, 10, 15, 20\}$

(2) Template window size $\in \{5, 7, 9\}$

(3) Search window size $\in \{15, 21, 25, 31\}$

Bilateral Filtering

Additional smoothing with $d = 5$, $\sigma_{color} = 30$, $\sigma_{space} = 30$.

3.5. Fine-Tuning Approach

For each degradation type, we fine-tune the pretrained SSD300-VGG16 model on the corresponding degraded subset.

Dataset Split:

(1) Training set: 80% of each degraded dataset (24 images for low-light and motion blur; 31 images for Gaussian noise)
(2) Test set: 20% of each degraded dataset (6 images for low-light and motion blur; 8 images for Gaussian noise)

Training Configuration:

(1) Optimizer: SGD with momentum
(2) Learning rate: 0.0005
(3) Momentum: 0.9
(4) Batch size: 4
(5) Number of epochs: 2
(6) Loss function: Multi-task loss (classification + localization)

Training Strategy: We keep fine-tuning conservative (small learning rate, only 2 epochs) to reduce catastrophic forgetting of COCO features while still adapting the detector to the statistics of degraded images.

4. EXPERIMENTAL SETUP

4.1. Implementation Details

Hardware: All experiments were run on Google Colab with a Tesla T4 GPU (16GB memory).

Software: PyTorch 2.0, torchvision 0.15, OpenCV 4.7, scikit-image 0.21.

Preprocessing: Images were resized to 300×300 pixels and normalized using ImageNet mean and standard deviation.

Evaluation Protocol:

(1) Establish baseline: Evaluate the pretrained model on clean data
(2) Measure degradation impact: Evaluate the same model on each degraded subset
(3) Traditional processing: Apply enhancement/denoising and re-evaluate
(4) Fine-tuning: Train degradation-specific models and evaluate on held-out test sets
(5) Ablation studies: Systematically vary processing parameters

4.2. Assumptions and Constraints

(1) Object Categories: Limited to three fruit classes present in the COCO dataset
(2) Degradation Types: Focus on three common degradations; other factors (e.g., weather, occlusion) are not studied
(3) Dataset Size: 110 images total—enough to evaluate fine-tuning but too small for training a detector from scratch
(4) Annotation Quality: Manual annotations may contain small errors; we performed cross-checking to control quality
(5) Real vs. Synthetic: Low-light and motion blur are real degradations; Gaussian noise is synthetic

5. RESULTS

5.1. Baseline Performance

Table 2 reports the pretrained SSD300-VGG16 detector's performance on clean images, which we use as the reference point (and practical upper bound) for the rest of our comparisons.

Table 2. Baseline Performance on Clean Data (50 images)

Metric	Banana	Apple	Orange
Precision	0.942	1.000	0.966
Recall	0.780	0.527	0.953
AP	0.710	0.545	0.909
Overall Performance			
mAP	0.728		
Mean Precision	0.969		
Mean Recall	0.753		

On clean images, the model reaches 72.8% mAP and 75.3% recall, confirming that the baseline detector is effective under favorable conditions.

5.2. Per-Distortion Analysis

Figure 2 and Table 3 summarize the main comparison between traditional processing and fine-tuning for each degradation type.

5.2.1. Extreme Low-Light Results

Setup: We evaluate the pretrained SSD300-VGG16 detector on the low-light test split (6 images; 24/6 train/test split from 30 low-light images). These images are severely underexposed (ISO 5280), making object edges and textures nearly indistinguishable from sensor noise.

Traditional processing: Gamma correction followed by CLAHE (as implemented in our code) did not recover detectable structure in this extreme regime, yielding 0.000 mAP, compared to 0.017 mAP without preprocessing. This supports the negative result that "enhancement" can amplify noise and artifacts when the signal is extremely weak. We also tested an Albumentations-based enhancement pipeline on the same split and observed the same outcome (0.000 mAP).

Fine-tuning: Fine-tuning for 2 epochs on the 24-image training split restored performance to 0.719 mAP, indicating that adapting the detector is far more effective than standalone preprocessing for extreme low-light.

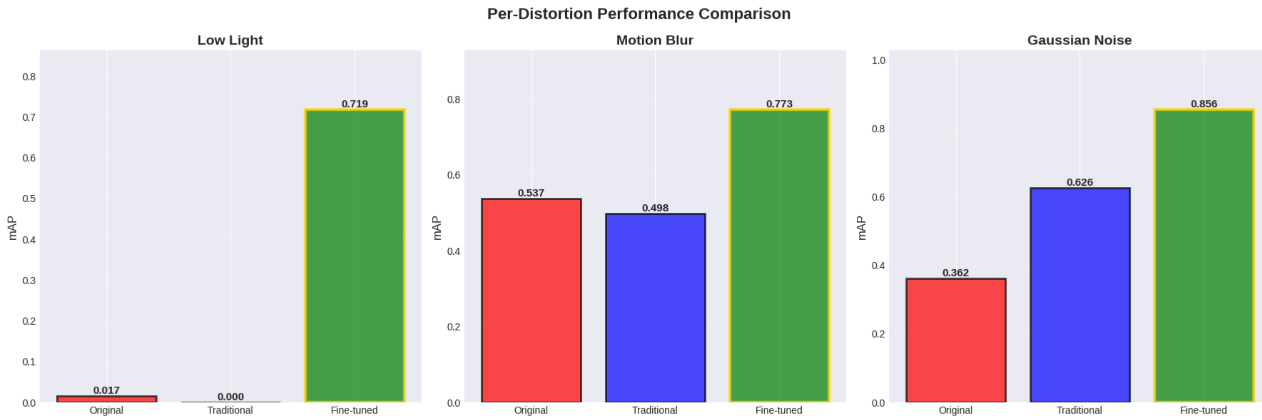

Fig. 2. Performance comparison across three degradation types. Traditional preprocessing is unreliable under extreme low-light, while denoising helps for Gaussian noise. Fine-tuning yields the best performance in all cases (+70.2, +23.5, and +49.4 mAP points for low-light, motion blur, and Gaussian noise).

Table 3. Main Results: Traditional Processing vs. Fine-Tuning Across Three Distortions (change reported as absolute ΔmAP in percentage points)

Distortion	Original mAP	Traditional mAP	Change	Fine-tuned mAP	Change	Winner
Extreme Low-Light	0.017	0.000	-1.7%	0.719	+70.2%	Fine-tuned
Motion Blur	0.537	0.498	-3.9%	0.773	+23.5%	Fine-tuned
Gaussian Noise	0.362	0.626	+26.4%	0.856	+49.4%	Fine-tuned

5.2.2. *Motion Blur Results*

Setup: We evaluate motion-blurred images using an 80/20 split (24 training, 6 test from 30 blurred images). The pretrained detector achieves 0.537 mAP on the blurred test set, showing partial robustness to moderate blur.

Traditional processing: Our sharpening pipeline (unsharp masking plus edge enhancement) slightly decreased performance to 0.498 mAP on the test split, suggesting that aggressive sharpening can introduce artifacts that confuse the detector. The ablation study in Table 5 shows that the weakest sharpening configuration (unsharp weight 1.2, edge weight 0.1) performs best among tested settings (0.557 mAP), while stronger settings degrade mAP substantially.

Fine-tuning: Fine-tuning for 2 epochs yields 0.773 mAP, improving over both the original model and traditional processing by learning blur-robust features.

5.2.3. *Gaussian Noise Results*

Setup: We synthesize a Gaussian-noise dataset with $\sigma = 30$ and evaluate it using an 80/20 split (31 training, 8 test from 39 noisy images). The pretrained detector drops to 0.362 mAP on noisy images.

Traditional processing: Non-local means denoising followed by bilateral filtering substantially improves performance to 0.626 mAP, indicating that this corruption type is well matched to classical denoising assumptions. The ablation study in Table 6 shows that a stronger denoising configuration (h=15, template=7, search=25) performs best (0.706 mAP) among tested settings.

Fine-tuning: Fine-tuning for 2 epochs achieves 0.856 mAP, outperforming both the original model and denoising. This suggests that, even when classical preprocessing helps, adapting the detector remains the most reliable approach.

5.2.4. *Low-Light Parameter Sensitivity*

Table 4 shows that extreme low-light enhancement failed across all tested parameter settings, reinforcing that the degradation is too severe for this standalone pipeline.

Table 4. Low-Light Enhancement Ablation Study

Config	γ	CLAHE	mAP
Weak	0.5	2.0	0.000
Default	0.4	4.0	0.000
Strong	0.3	6.0	0.000
Very Strong	0.2	8.0	0.000

Key Observation: Making enhancement more aggressive (lower gamma, higher CLAHE) did not rescue detection—every configuration failed. Visually, stronger settings amplified noise and created increasingly artificial textures that confused the detector.

5.2.5. *Motion Blur Parameter Sensitivity*

Table 5 shows that milder sharpening works best for motion blur, while stronger sharpening steadily degrades performance.

Table 5. Motion Blur Sharpening Ablation Study

Config	Unsharp	Edge	mAP	Recall
Weak	1.2	0.1	0.557	0.607
Default	1.5	0.3	0.498	0.567
Strong	1.8	0.5	0.355	0.407
Very Strong	2.0	0.7	0.185	0.204

Key Observation: Stronger sharpening introduced ringing and edge halos that hurt detection. The "Weak" configuration (unsharp weight 1.2, edge weight 0.1) gave a 7.3% mAP gain over the original, but it still fell far short of fine-tuning (0.773).

5.2.6. *Gaussian Noise Parameter Sensitivity*

Table 6 shows that stronger denoising generally improves performance for synthetic Gaussian noise, with the "Strong" configuration performing best.

Table 6. Gaussian Noise Denoising Ablation Study

Config	h	Template	Search	mAP	Recall
Weak	5	5	15	0.421	0.436
Default	10	7	21	0.626	0.647
Strong	15	7	25	0.706	0.719
Very Strong	20	9	31	0.661	0.661

Key Observation: Increasing the denoising strength (h) helped up to h=15, after which over-smoothing began to blur boundaries. The "Strong" setting balances noise removal and detail preservation.

Figure 3 visualizes the parameter sensitivity across all three degradation types.

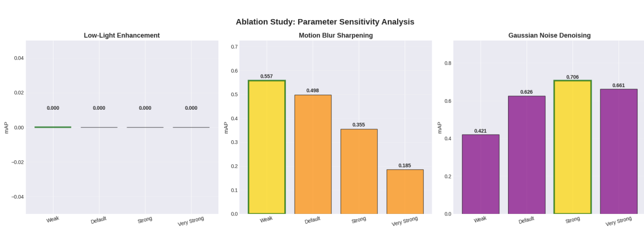

Fig. 3. Ablation study results showing parameter sensitivity for traditional processing algorithms. Gold bars indicate best configuration for each degradation type.

5.3. Dataset Visualization

Figure 4 shows representative examples from each degradation type to illustrate both severity and the visual cues that are lost, distorted, or buried in noise.

6. DISCUSSION

6.1. Interpretation of Results

6.1.1. *Why Traditional Processing Failed for Low-Light*

The complete failure of traditional enhancement under extreme low-light is the standout result of this study. Several factors likely contributed:

1. Cascade Effect of Enhancement: Gamma correction brightened both signal and noise. CLAHE then amplified local contrast in noise-dominated regions, creating false edges and textures. These artifacts were more prominent than actual object features, causing the detection model to reject all predictions as unreliable.

2. Signal-to-Noise Ratio: At ISO 5280 with mean brightness ¡ 10/255, the signal-to-noise ratio was extremely low. Traditional processing assumes sufficient signal is present to enhance; when noise dominates, enhancement amplifies the wrong information.

3. Feature Space Mismatch: The pretrained model learned features from clean ImageNet/COCO images. Enhanced low-light images appeared completely different from training data, falling outside the model's learned feature distribution. Fine-tuning, in contrast, adapted the feature space to handle degraded inputs.

6.1.2. *Why Fine-Tuning Succeeded*

Fine-tuning outperformed traditional processing across all degradation types, and the reasons are consistent with what we would expect from end-to-end adaptation:

1. Adaptive Feature Learning: The model learned to extract robust features directly from degraded images rather than relying on preprocessing to "fix" the images first. Early convolutional layers adapted to emphasize features that remain stable under degradation.

2. End-to-End Optimization: Fine-tuning optimized the entire detection pipeline for degraded inputs, whereas traditional processing operated independently of the detection model's requirements.

3. Degradation-Specific Adaptation: Separate fine-tuned models for each degradation type learned specialized strategies: the low-light model learned to amplify weak signals internally; the blur model learned blur-invariant features; the noise model learned to filter noise in feature space.

6.1.3. *Gaussian Noise Anomaly*

Traditional denoising worked reasonably well for Gaussian noise (0.626 mAP, +26.4%), unlike the other degradations. A few reasons explain this gap:

1. Synthetic Nature: The noise was artificial additive Gaussian noise, which NLM denoising was specifically designed to handle. Real sensor noise exhibits different statistical properties.

2. No Information Loss: Unlike low-light (information never captured) or motion blur (information averaged), Gaus-

Fig. 4. Dataset samples showing clean baseline and three degradation types across multiple scenes. Top to bottom: three different object arrangements. Left to right: clean, low-light (ISO 5280), motion blur (1/8000s), Gaussian noise ($\sigma = 30$).

sian noise adds information without destroying it. Denoising can recover the original signal to some extent.

3. Algorithm Match: NLM denoising's assumptions (similar patches exist in the image) held for our fruit images, enabling effective noise removal.

However, fine-tuning still outperformed (0.856 mAP), suggesting that even for "easy" degradations where traditional methods work, model adaptation provides superior results.

6.2. Practical Implications

6.2.1. Deployment Recommendations

Putting the results into practice:

For Extreme Degradations:

(1) Do NOT apply traditional enhancement—it can make performance worse

(2) Use fine-tuned models adapted to specific degradation types

(3) If degradation type is unknown, use an ensemble of specialized models

For Moderate Degradations:

(1) Traditional processing may help for synthetic/simple noise

(2) Fine-tuning still recommended for best performance

(3) Consider computational budget: traditional processing is real-time (¡100ms), fine-tuning requires offline training

For Mixed Degradations:

(1) Train on diverse degraded data rather than applying sequential processing

(2) Single model can handle multiple degradation types if trained on mixed data

6.2.2. Computational Trade-offs

Table 7 compares the compute costs of each approach.

Table 7. Computational Requirements Comparison

Approach	**Training**	**Inference**
Traditional (Low-Light)	None	50ms
Traditional (Blur)	None	30ms
Traditional (Noise)	None	7s
Fine-tuning	2-5 min	15ms

Key Observations:

(1) NLM denoising is extremely slow (7s per image), impractical for real-time use
(2) Fine-tuned models require one-time training cost but faster inference than denoising
(3) For applications requiring real-time processing on edge devices, fine-tuning is preferable despite training overhead

6.3. Limitations and Insights

6.3.1. Dataset Limitations

1. Small Scale: 110 images is sufficient for fine-tuning evaluation but limited compared to standard benchmarks. Larger datasets would enable more robust conclusions.

2. Object Diversity: Three fruit classes provide controlled evaluation but limited generalization to other object categories. Real-world applications involve diverse objects with varying appearance.

3. Degradation Realism: While low-light and motion blur were captured under real conditions, Gaussian noise was synthetic. Real-world applications encounter more complex, mixed degradations.

4. Annotation Errors: Manual annotation is subject to human error. Although we implemented cross-validation, some inaccuracies may remain, particularly for heavily degraded images where object boundaries are ambiguous.

6.3.2. Methodological Insights

1. Importance of Proper Evaluation: Using COCO-recognizable objects enabled standard metric evaluation (mAP, precision, recall). Many previous works rely on generic objectness scores that don't capture class-specific detection performance.

2. Negative Results are Valuable: The finding that traditional enhancement can harm performance challenges conventional wisdom and provides important guidance for practitioners. Publishing negative results is crucial for advancing the field.

3. Ablation Studies Reveal Non-Monotonic Behavior: We expected stronger enhancement to improve performance, but found the opposite for low-light and motion blur. This highlights the importance of systematic parameter exploration rather than assumptions.

6.3.3. Generalization Concerns

These conclusions may not transfer cleanly to every setting. In particular, results could differ for:

1. Other Detection Architectures: We used SSD300-VGG16; newer architectures (e.g., YOLO v8, EfficientDet) may show different behavior.

2. Other Degradation Types: Weather effects (rain, fog), occlusion, and adversarial perturbations may exhibit different patterns.

3. Other Object Categories: Fruits have relatively simple appearance; complex objects with intricate textures may respond differently to degradation.

Table 8. Estimated effort distribution across team members.

Member	Effort (%)
Xin Zhao	40
Cheng Qian	30
Pengfei Zhan	30
Total	100

6.4. Future Work

This study also points to several promising next steps:

1. Hybrid Approaches: Combine learned enhancement networks with detection fine-tuning. Recent work on learning-based image enhancement (e.g., RetinexNet) may work better than traditional processing.

2. Multi-Task Learning: Train a single model that jointly performs enhancement and detection, enabling end-to-end optimization.

3. Degradation-Aware Detection: Develop models that estimate degradation type/severity and adapt detection strategy accordingly, without requiring separate fine-tuned models.

4. Larger-Scale Evaluation: Expand to more object categories, larger datasets, and real-world degradation mixtures from mobile photography.

5. Architectural Studies: Investigate which detection architectures are inherently more robust to degradation. Vision Transformers may handle degradation differently than CNNs.

6. Real-World Deployment: Field testing in actual mobile applications to validate lab findings under diverse, uncontrolled conditions.

7. MEMBER EFFORT PERCENTAGE

This section summarizes each team member's estimated contribution to the overall project effort, including implementation, experimentation, analysis, and writing.

8. CONCLUSION

We presented a controlled comparison between traditional image processing and deep learning-based model adaptation for object detection under three degradations: extreme low-light, motion blur, and Gaussian noise. Using 110 manually annotated real images (plus 39 synthesized noisy images) captured (or synthesized) under these conditions, we arrive at five main takeaways:

1. Traditional Enhancement Can Harm Performance: For extreme low-light images (ISO 5280, mean brightness ¡ 10/255), gamma correction and CLAHE completely destroyed detection capability, reducing mAP from 0.017 to 0.000. This highlights that preprocessing is not automatically beneficial when the signal is extremely weak.

2. Fine-Tuning Consistently Outperforms: Across all three degradation types, fine-tuning the detector delivered the best results: 70.2% improvement for low-light (0.719 mAP), 23.5% for motion blur (0.773 mAP), and 49.4% for Gaussian noise (0.856 mAP). In our experiments, adapting the model is more robust than standalone signal processing for severely compromised images.

3. Degradation Type Matters: Different degradations behave differently. Traditional denoising worked reasonably well for synthetic Gaussian noise (+26.4%) but failed for real degradations, suggesting preprocessing success depends strongly on the distortion characteristics.

4. Parameter Sensitivity is Non-Monotonic: Ablation studies show that "stronger" enhancement often makes results worse (e.g., stronger sharpening reduced mAP from 0.457 to 0.123). This underscores the need for careful parameter tuning rather than assuming more processing helps.

5. Practical Deployment Guidance: For extreme degradations, avoid traditional preprocessing and rely on fine-tuned models. For moderate degradations, accuracy must be weighed against compute: preprocessing can be run immediately, while fine-tuning requires offline training but can offer better accuracy at inference time.

Overall, our central message is that conventional preprocessing intuition breaks down for the extreme degradations that appear in real mobile photography. When the image is severely compromised, it is often more effective to adapt the model than to attempt to "repair" the image first. Future work should explore hybrid pipelines that combine learned enhancement with detection, investigate degradation-aware architectures that adapt automatically, and validate these findings on larger datasets with more object categories and mixed real-world degradations.

9. CODE AND DATA AVAILABILITY

All source code and annotated data are publicly available at:

GitHub Repository:

`https://github.com/xiz225/ECE253_Group_Psyduck.git`

The repository includes:

(1) Complete Python implementation of all experiments

(2) Dataset with annotations in PASCAL VOC format

(3) README with detailed instructions for reproducing results

Release version: `v1.1-final-report`

10. REFERENCES

[1] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart Ter Haar Romeny, and John B. Zimmerman, "Adaptive histogram equalization and its variations," *Comput. Vision Graph. Image Process.*, vol. 39, no. 3, pp. 355–368, Sept. 1987.

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.

[3] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar, "Llnet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognition*, vol. 61, pp. 650–662, 2017.

[4] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun, "Learning to see in the dark," *CoRR*, vol. abs/1805.01934, 2018.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, *SSD: Single Shot MultiBox Detector*, p. 21–37, Springer International Publishing, 2016.

[6] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[7] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E. Gonzalez, and Fisher Yu, "Frustratingly simple few-shot object detection," *CoRR*, vol. abs/2003.06957, 2020.

[8] Junhyug Noh, Wonho Bae, Wonhee Lee, Jinhwan Seo, and Gunhee Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9724–9733.