

Deep Robust Object Detection under High Illumination Conditions Using Modulo Images

Luis Toscano-Palomino, Kebin Contreras, Brayan Monroy and Jorge Bacca
Department of Computer Science, Universidad Industrial de Santander, Colombia
jbaquin@uis.edu.co

Abstract—Drone detection under high-illumination conditions remains a critical challenge due to sensor saturation, which degrades visual information and limits the performance of conventional detection models. A promising alternative to overcome this issue is modulo imaging, an approach based on modulo-ADCs that reset pixel intensities upon reaching a predefined saturation threshold, thus avoiding saturation loss. This work presents a methodology based on fine-tuning a detection model using modulo images, allowing accurate object detection without requiring High Dynamic Range (HDR) image reconstruction. Additionally, an optional reconstruction stage using the Autoregressive High-order Finite Difference (AHFD) algorithm is evaluated to recover high-fidelity HDR content. Experimental results show that the fine-tuned model achieves F1-scores above 96% across different illumination levels, outperforming saturated and raw modulo inputs, and approaching the performance of ideal HDR images. These findings demonstrate that fine-tuning with modulo data enables robust drone detection while reducing inference time, making the reconstruction process optional rather than essential.

Index Terms—Fine-Tuning, Modulo Imaging, YOLOv11, Drone Detection, HDR recovery

I. INTRODUCTION

The exponential advancement of drone technology has enabled its application across various sectors, such as package delivery [1], military reconnaissance [2], aerial photography [3], and environmental monitoring [4]. However, the widespread use has also introduced security challenges due to the invasion of personal and institutional spaces [5], [6], for instance certain reports suggest that drones have been utilized near correction facilities [5], highlighting the urgent need to detect drones through computer vision systems, both in urban environments and open areas [7], which plays a crucial role in the proper development of drone operations [8].

Deep learning has become a key computational approach to detect objects in a scene. By leveraging neural networks trained on large-scale datasets, these methods allow the automatic recognition of complex visual patterns [9], greatly advancing the field of computer vision, particularly object detection using RGB images [10], [11]. A representative example is You Only Look Once (YOLO), a real-time detection model trained to identify and classify objects within images using supervised learning [12]. However, the effectiveness of these computer vision systems depends directly on the quality of the

visual data used to detect and respond to drones [13]. One of the most significant challenges, in acquiring such quality data, is the saturation produced by commercial CCD sensors, where capturing drones flying over bright skies becomes particularly difficult due to their small size and rapid movement [14], resulting in overexposed regions that obscure essential visual information [15]. This degradation severely compromises image quality and, consequently, the performance of visual-based drone detection methods [16].

A promising solution to the saturation issue in imaging is modulo imaging [17], [18], which utilizes modulo-ADC sensors that reset pixel intensity once an intensity threshold is exceeded [19]. This method enables the capture of non-saturated data in a coded format, known as modulo images. While modulo images do not preserve color integrity, they retain sufficient structural and fine detail for detection tasks [20], making them particularly effective in high-illumination environments. For instance, in [21], modulo imaging is combined with deep learning models for vehicle detection. The authors employed the Simultaneous Phase Unwrapping and Denoising Algorithm (SPUD) [22] to recover the High Dynamic Range (HDR) image and reverse the transformation applied by the modulo sensor before applying detection algorithms. However, the use of SPUD introduces significant limitations: it incurs a considerable computational cost during inference and relies on band-limited signal assumptions, which are often not met in real-world scenarios, resulting in poor reconstruction quality and, consequently, suboptimal object detection performance.

In this work, we present an alternative approach that eliminates the need for a recovery stage by fine-tuning a detection model directly on modulo images. This strategy allows the model to adjust its weights for processing modulo images, leading to improved detection robustness under extreme lighting conditions. Additionally, we propose exploring the use of an enhanced reconstruction algorithm, Autoregressive High-Order Finite Difference (AHFD) [23], which offers two key advantages over SPUD: (i) it introduces fewer artifacts around edges, preserving crucial visual details, and (ii) it demonstrates greater stability across varying illumination and noise conditions. While fine-tuning on raw modulo images alone achieves strong detection performance, the application of AHFD becomes particularly valuable in scenarios that require human interpretability (e.g., for visual inspection) or when

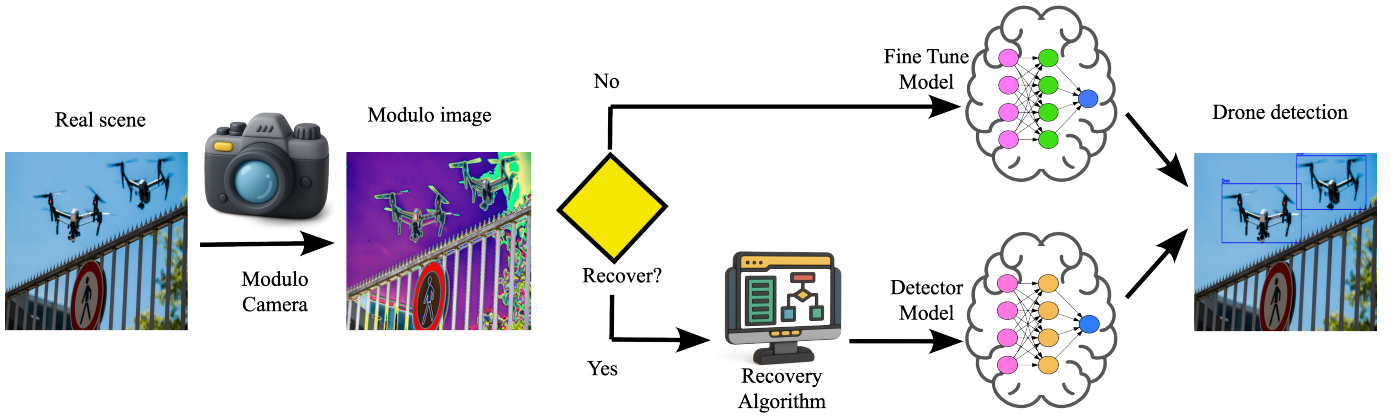


Fig. 1. **Drone detection pipeline using modulo images.** A modulo camera captures the scene and produces a modulo image. Optionally, this image can be processed through a recovery algorithm to estimate HDR content, which is then passed to a standard detector model. Alternatively, the raw modulo image is directly input to a fine-tuned model trained specifically on modulo data, allowing accurate detection without recovery.

additional computer vision tasks (such as segmentation or tracking) need to be performed on the reconstructed data. In these cases, reconstruction serves as an optional enhancement rather than a mandatory step.

II. BACKGROUND

The modulo-sensing process resets the pixel intensity each time the saturation threshold λ is reached (usually $\lambda = 2^8 - 1$) [24]. Assuming a vector signal $\mathbf{x} \in \mathbb{R}^n$ with $n = 3 \times H \times W$, the modulo-sensing model can be mathematically defined as

$$\mathbf{y} = \text{mod}(\mathbf{x}, \lambda), \quad (1)$$

where $\text{mod}(\cdot, \lambda)$ is the modulo operator and $\mathbf{y} \in \mathbb{R}^n$ is the acquired modulo measurements.

Visually, a modulo image may resemble a regular image, but in saturated regions it exhibits desaturation and color distortion due to the wrapping effect. This phenomenon is illustrated in Figure 1. Although modulo images retain structural details, they do not provide direct access to the original signal \mathbf{x} due to the intensity wrapping. Therefore, recovering \mathbf{x} from the measurements \mathbf{y} requires inversion algorithms specifically designed to reverse the modulo operation and reconstruct the original high dynamic range content [22], [23].

III. METHOD

This section presents the proposed methodology, which is structured in three stages: (A) formulating and solving the fine-tuning optimization problem to adapt the weights of the pre-trained model for modulo images, (B) Performing detection task using a YOLOv11 model Fine Tuned with modulo images. Additionally, it is considered an image recovery stage (C) using a HDR recovery algorithm.

A. Fine-tuning optimization problem

The optimization process, known as fine-tuning, can be mathematically expressed as:

$$\theta^* \in \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\text{YOLO}_{\theta}(\mathbf{y}_i), \mathbf{c}_i). \quad (2)$$

In this optimization problem, θ^* represents the optimal parameters of the YOLOv11 model and \mathcal{L} the cost function composed of three loss functions, box loss, classification loss and distributed focal loss [25], which quantifies the difference between the model predictions $\text{YOLO}_{\theta}(\mathbf{y}_i)$ and the true labels \mathbf{c}_i . The goal is to obtain the optimal model parameters θ that allow the YOLOv11 detector to perform detection task directly on modulo images. Instead of training the model from scratch, it leverages a YOLOv11 network pre-trained on RGB images. This is because modulo images preserve the spatial structure and object boundaries of the original scenes, allowing the model to reuse the low-level features learned during pretraining. Fine-tuning adapts the model to the visual properties of modulo data, enabling effective interpretation without the need for recovering the original HDR image.

B. Detection task

After optimizing the model weights, the detection task is performed employing the YOLOv11 n configuration, which, is represented as follows:

$$\mathbf{c} = \text{YOLO}_{\theta^*}(\mathbf{y}), \quad (3)$$

where θ^* represents the optimized weights and $\mathbf{c} = \{\mathbf{c}^{(i)}\}_{i=1}^N$ represents the set of N bounding boxes predicted by the YOLOv11 detection model for the N classes instances where \mathbf{y} denoting the raw modulo image. Each bounding box $\mathbf{c}^{(i)}$ is defined by the coordinates $\mathbf{c}^{(i)} = [j_1, k_1, j_2, k_2, z]$, where (j_1, k_1) and (j_2, k_2) are the top-left and bottom-right coordinates of the box, respectively, and z is the confidence score associated with the detection.

Although fine-tuning allows the model to operate directly on modulo images, there may still be scenarios where recovering a more interpretable image is beneficial, for example, to support visual inspection or ensure compatibility with models not adapted to modulo data.

C. Recovery with AHFD

As a complementary step, the AHFD algorithm [23] is explored to optionally recover an HDR image from the modulo representation. Given a modulo image, this method estimates the original HDR image by solving the optimization problem:

$$\tilde{x} \in \arg \min_x \|\Delta^N x - \text{mod}(\Delta^N y, \lambda)\|_2^2 + R(x), \quad (4)$$

where Δ^N represents the high-order finite differences, x the original signal, y the modulo measurements, and $R(x)$ a regularization term. This optimization is solved in two steps:

1) *Step 1: Autoregressive phase unwrapping*: In the first stage, the algorithm estimates how many times each pixel exceeded the sensor dynamic range, rather than directly reconstructing the original image. To do so, the image is first vectorized following a zig-zag path along columns and rows, ensuring that neighboring pixels remain adjacent in the vector representation. This facilitates the detection of local variations in intensity. The algorithm then analyzes these variations using high-order finite differences and solves an iterative process based on frequency-domain projections via the 1D Discrete Cosine Transform (DCT).

2) *Step 2: Stripe Artifact Removal*: As a result of the first step, stripe-like artifacts may appear in the reconstructed image, especially when the band-limited signal assumption is not fully met [26]. To mitigate this, a sparse correction map is computed using a spatial filtering process. This step employs the 2D DCT to efficiently remove structured artifacts, improving the visual quality while preserving fine details.

IV. SIMULATIONS AND RESULTS

This section evaluates the performance of the proposed Fine-Tuned YOLOv11 model trained with modulo images, in comparison to standard detection models and recovery-based approaches. The main objective is to analyze object detection performance under varying intensity levels. For this, the methodology simulates the acquisition of modulo images, saturated images, and recovered HDR images. Recovery is performed using SPUD [22], as the state-of-the-art, and AHFD [23], as an enhanced reconstruction algorithm. In contrast, the proposed approach bypasses the recovery stage by directly fine-tuning YOLOv11 on modulo inputs, aiming to improve detection under high dynamic range scenarios, without a recovery stage. Specifically, to simulate different intensity conditions, each HDR image x is normalized to the range $[0, 1]$ and then scaled by an intensity factor α .

A. Experimental Setup

Dataset: Images are obtained from the public Drone Dataset (UAV), which contains 1359 RGB images, available on Kaggle [27], and then standardized through a resizing procedure in which the longest side of each image is scaled to 1024 pixels while preserving the original aspect ratio. As the dataset provides RGB images that have already undergone demosaicing, the potential effects of Bayer pattern and its interaction with modulo sensing are not considered in this

work. Following this, the methodology generates two types of image variants: modulo images and saturated images. To obtain the modulo images, it first applies an intensity factor to the resized images, and subsequently performs a modulo operation, mathematically defined as:

$$y = \text{mod}(x_r \cdot \alpha, \lambda), \quad (5)$$

where x_r is the resized image, y is the modulo image, $\lambda = 2^8 - 1$, and $\alpha \in \{1.5, 2, 3\}$ is the intensity factor that adjusts the image intensity to simulate different light exposure levels.

In addition, the methodology generates the saturated variants by applying the same intensity factor to the resized images, followed by a clipping operation that limits pixel intensity values to the sensor's maximum capacity λ . Formally, the resulting saturated image is given by:

$$x_s = \text{minimum}(x_r \cdot \alpha, \lambda), \quad (6)$$

where x_s is the saturated image. The performance of different imaging methodologies is evaluated across saturation scenarios, ranging from low to extreme. During the fine-tuning stage, raw modulo images were used, avoiding the image reconstruction step.

Network Setup: For drone detection, the methodology employs YOLOv11 using its n configuration, a lightweight variant optimized for real-time applications. YOLOv11 is a convolutional neural network (CNN) with a backbone for feature extraction, a neck for multi-scale feature aggregation, and a detection head for predicting object bounding boxes and class probabilities. Given an input image, the model outputs bounding boxes for detected drones, each with a confidence score and predicted class label. The model's process follows the formulation given in Equation 3.

Metrics: To assess the effectiveness of the proposed detection method, three metrics are employed: precision, recall, and F1-score [28]. Precision evaluates how many of the instances predicted as positive are actually correct, reflecting the model's ability to avoid false alarms.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (7)$$

where TP (True Positives) refers to the number of correctly detected positive instances, and FP (False Positives) represents the number of instances that were incorrectly classified as positive by the model.

Next, recall quantifies the model's capacity to detect all relevant objects, measuring how many of the actual positives were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (8)$$

where FN (False Negatives) denotes the number of actual positive instances that were not detected by the model.

Finally, the methodology uses the F1-score to obtain a single performance indicator that balances both precision and recall. This metric is particularly useful when dealing with

TABLE I
PERFORMANCE COMPARISON FOR YOLOv11 UNDER DIFFERENT INPUT TYPES AND PROCESSING METHODS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, WHILE THE SECOND-BEST RESULTS ARE UNDERLINED. RESULTS ARE REPORTED FOR DIFFERENT ILLUMINATION LEVELS $\alpha \in 1.5, 2, 3$.

Method	F1-Score (α)			Recall (α)			Precision (α)			Time(ms)
	1.5	2	3	1.5	2	3	1.5	2	3	
Saturated	87.1 \pm 18.9	79.9 \pm 26.4	73.3 \pm 29.7	91.3 \pm 16.8	87.8 \pm 22.7	84.8 \pm 26.4	85.9 \pm 20.7	78.5 \pm 28.2	70.7 \pm 32.4	24.495 \pm 3.2453
Modulo	80.7 \pm 22.2	81.1 \pm 23.0	75.3 \pm 26.7	<u>92.5 \pm 13.3</u>	91.0 \pm 17.5	89.0 \pm 21.3	77.0 \pm 25.4	77.5 \pm 26.1	71.9 \pm 28.8	25.677 \pm 3.4491
Recovery SPUD	90.1 \pm 12.5	88.2 \pm 16.1	88.1 \pm 15.5	92.4 \pm 13.3	91.2 \pm 15.3	91.8 \pm 14.3	89.1 \pm 13.1	86.9 \pm 18.1	87.4 \pm 16.6	81.170 \pm 0.5312
FineTune Modulo 25% (Ours)	83.7 \pm 19.4	80.5 \pm 21.7	80.7 \pm 23.2	84.9 \pm 19.9	83.7 \pm 21.9	84.0 \pm 24.3	85.1 \pm 21.3	80.3 \pm 24.0	80.4 \pm 25.1	25.547 \pm 3.3524
FineTune Modulo 50% (Ours)	88.2 \pm 13.0	85.1 \pm 16.8	85.9 \pm 16.9	90.6 \pm 13.2	90.8 \pm 14.2	89.8 \pm 15.2	87.6 \pm 15.5	83.1 \pm 19.9	84.7 \pm 19.8	26.700 \pm 3.3996
FineTune Modulo 100% (Ours)	91.7 \pm 7.21	89.0 \pm 13.1	89.9 \pm 13.1	92.4 \pm 5.48	93.9 \pm 11.4	92.7 \pm 13.4	90.2 \pm 10.0	86.3 \pm 15.8	88.3 \pm 14.5	22.043 \pm 0.4429
Recovery AHFD (Ours)	91.2 \pm 11.1	88.6 \pm 16.1	90.0 \pm 11.6	94.5 \pm 7.47	92.1 \pm 14.2	93.3 \pm 10.9	89.7 \pm 14.1	87.2 \pm 18.5	88.1 \pm 14.4	81.573 \pm 0.6701
Ideal HDR	92.2 \pm 8.99			94.7 \pm 6.80			91.2 \pm 10.9			22.022 \pm 0.4292





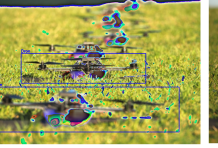
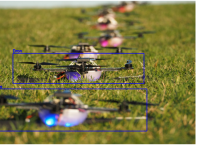

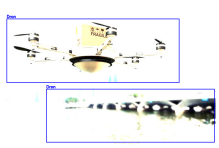




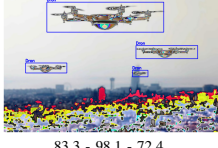
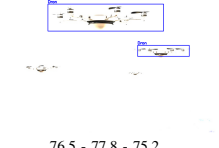


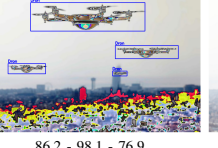

α	Modulo	Saturated	SPUD	AHFD (Ours)	FineTune (Ours)	Ideal HDR
1.5	 0 - 0 - 0	 85.2 - 79.4 - 91.8	 45.2 - 30.3 - 92.8	 85.5 - 77.5 - 95.2	 86.4 - 84.2 - 88.7	 86.4 - 78.9 - 95.5
2	 64.0 - 100 - 47.2	 67.4 - 99.5 - 51.0	 92.9 - 99.5 - 87.12	 91.4 - 100 - 84.3	 93.0 - 100 - 86.9	 94.4 - 100 - 88.7
3	 83.3 - 98.1 - 72.4	 76.5 - 77.8 - 75.2	 83.2 - 98.8 - 71.9	 81.3 - 96.6 - 70.2	 86.2 - 98.1 - 76.9	 86.4 - 97.8 - 77.4

Fig. 2. Visual and metric results for each method under different intensity levels ($\alpha = 1.5, 2, 3$). Detection images are shown along with their corresponding F1-score, Recall, and Precision metrics.

imbalanced datasets or when it is crucial to maintain a trade-off between false positives and false negatives. It is computed as the harmonic mean of precision and recall:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (9)$$

All simulations were conducted using the Google Colab environment, which provides access to an Intel(R) Xeon(R) CPU @ 2.00GHz, 12.7 GB of RAM, and an NVIDIA Tesla T4 GPU with 15 GB of VRAM. The GPU was used during the training stage to accelerate the fine-tuning process, whereas the CPU was employed during the validation stage to measure the model's performance, ensuring efficient execution.

B. Performance Analysis of YOLOv11 Fine-Tuned with Modulo Images for Drone Detection

As evidenced in Table I, the proposed fine-tuning strategy using modulo images demonstrates a strong capability to enhance detection performance under saturated lighting conditions. The configurations labeled as "FineTune Modulo X%" correspond to training the model using X% of the available modulo image training data, while the test set (20% of the full dataset) stays unchanged and separate from all training scenarios. When the YOLOv11 model is fine-tuned using

100% of the modulo dataset, the results show remarkable improvements across all evaluation metrics, achieving an F1-score of 91.7% for $\alpha = 1.5$, a precision of up to 90.2% for $\alpha = 1.5$, and a recall of 93.9% for $\alpha = 2$. These results are not only superior to those obtained with saturated or raw modulo images but also closely rival the performance of recovery-based approaches, such as AHFD, which delivers the best metrics overall. However, it is important to note the strong dependency of the fine-tuning approach on the quantity of training data available. When the dataset is reduced to 50% or 25%, the F1-score drops to 88.2% and 83.7% respectively, highlighting a critical limitation: while fine-tuning avoids reconstruction and achieves fast inference time (22.043 ms), it demands a sufficiently big and well-annotated dataset to be effective. This trend is further supported by the visual comparisons in Figure 2. At low intensity levels $\alpha = 1.5$, the raw modulo input fails to produce any detections, while the fine-tuned model correctly identifies drones with precision and spatial consistency. At $\alpha = 2$, detection with saturated inputs and the SPUD method degrades noticeably, either due to overexposure or failed recovery. In contrast, both AHFD and FineTune (Ours) maintain robust detection outputs. Finally, under extreme intensity $\alpha = 3$, recovery-based methods show

visual artifacts or detection errors, whereas the fine-tuned model continues to offer accurate bounding boxes, visually aligning with the high scores reported. These visual results reinforce the effectiveness of the proposed approach and confirm its resilience in high dynamic range scenarios. The proposed fine-tuning strategy exhibits competitive performance using approximately 50% of the processing time, offering an efficient and competitive alternative to computationally intensive reconstruction pipelines, in which approximately double the total processing time, presenting a critical drawback for real-time applications where fast inference is essential.

V. CONCLUSION

This work demonstrates that fine-tuning a detection model with modulo images significantly enhances drone detection in high-illumination environments, overcoming the limitations of traditional sensors affected by saturation. The experimental results reveal F1-scores exceeding 89% across various lighting conditions, validating the effectiveness of the fine-tuning approach. Unlike state-of-the-art methods that rely on computationally intensive HDR reconstruction, our approach eliminates the need for image recovery, offering faster and more efficient processing while maintaining high detection accuracy. In contrast, recovery algorithms require up to 22.043 ms for drone detection. Additionally, we introduce an optional HDR recovery step using the AHFD algorithm, which further improves image quality in scenarios where human visual interpretation is necessary. This approach provides a compelling alternative for real-time drone detection in challenging lighting conditions, with potential for deployment in practical applications where computational efficiency is critical.

VI. ACKNOWLEDGMENT

The authors acknowledge the Vicerrectoría de Investigación y Extension of Universidad Industrial de Santander for supporting this work with the project 3968.

REFERENCES

- [1] A. W. Sudbury and E. B. Hutchinson, "A cost analysis of amazon prime air (drone delivery)," *Journal for Economic Educators*, vol. 16, no. 1, pp. 1–12, 2016.
- [2] J. H. Kim and S. K. Park, "A study on the advancement of intelligent military drones: Focusing on reconnaissance operations," *Journal of Advanced Military Studies*, vol. 8, no. 1, pp. 55–78, 2025.
- [3] M. F. M. Rahman, D. D. Prasetyo, and R. R. S. Putra, "Analysis of aerial photography with drone type fixed wing in kotabaru, lampung," *International Journal of Scientific & Technology Research*, vol. 8, no. 5, pp. 1–5, 2019.
- [4] M. F. Mwaura, "The use of drones in environmental monitoring and conservation," *Research Inventory Journal of Biological and Applied Sciences*, vol. 10, no. 3, pp. 69–75, 2024.
- [5] G. Gonzalez, B. A. Jackson, and S. McKay, "Countering the emerging drone threat to correctional security," RAND Corporation, Tech. Rep., 2024.
- [6] R. A. Clothier, D. A. Greer, D. G. Greer, and A. M. Mehta, "Risk perception and the public acceptance of drones," *Risk Analysis*, vol. 35, no. 6, pp. 1167–1183, 2015.
- [7] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-uav detection and tracking," *arXiv preprint arXiv:2205.10851*, 2022.
- [8] B. K. S. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, and H. P. H. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," *arXiv preprint arXiv:2103.13933*, 2021.
- [9] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.
- [10] J. Bacca, "Projection-based correction for enhancing deep inverse networks," *arXiv preprint arXiv:2505.15777*, 2025.
- [11] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.
- [13] F. Mahdavi and R. Rajabi, "Drone detection using convolutional neural networks," *arXiv preprint arXiv:2107.01435*, 2021.
- [14] Z. Liu, P. An, Y. Yang, S. Qiu, Q. Liu, and X. Xu, "Vision-based drone detection in complex environments: A survey," *Drones*, vol. 8, no. 11, 2024.
- [15] H. Photonics, "Ccd saturation and blooming," 2024, accessed: 2025-04-07. [Online]. Available: <https://hamamatsu.magnet.fsu.edu/articles/ccdsatandblooming.html>
- [16] C. Hu, B. B. Sapkota, J. A. Thomasson, and M. V. Bagavathiannan, "Influence of image quality and light consistency on the performance of convolutional neural networks for weed mapping," *Remote Sensing*, vol. 13, no. 11, 2021.
- [17] H. Zhao, B. Shi, C. Fernandez-Cull, S.-K. Yeung, and R. Raskar, "Unbounded high dynamic range photography using a modulo camera," 04 2015.
- [18] J. Bacca, B. Monroy, and H. Arguello, "Deep plug-and-play algorithm for unsaturated imaging," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 2460–2464.
- [19] S. Dutta, A. Chugh, R. Tamburo, A. Rowe, and S. G. Narasimhan, "Performance characterization of reactive visual systems," in *2015 IEEE International Conference on Computational Photography (ICCP)*, 2015, pp. 1–9.
- [20] S. Zheng, B. Shi, and A. C. Kot, "Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 20815–20825.
- [21] K. Contreras, B. Monroy, and J. Bacca, "High dynamic range modulo imaging for robust object detection in autonomous driving," *arXiv preprint arXiv:2504.11472*, 2025.
- [22] J. Pineda, J. Bacca, J. Meza, L. A. Romero, H. Arguello, and A. G. Marrugo, "Spud: simultaneous phase unwrapping and denoising algorithm for phase imaging," *Appl. Opt.*, vol. 59, no. 13, pp. D81–D88, May 2020.
- [23] B. Monroy, K. Contreras, and J. Bacca, "Autoregressive high-order finite difference modulo imaging: High-dynamic range for computer vision applications," *arXiv preprint arXiv:2504.04228*, 2025.
- [24] A. Bhandari, F. Krahmer, and R. Raskar, "On unlimited sampling and reconstruction," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3827–3839, 2021. [Online]. Available: <http://dx.doi.org/10.1109/TSP.2020.3041955>
- [25] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024, accessed: 2025-04-07. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [26] A. Bhandari, F. Krahmer, and R. Raskar, "On unlimited sampling and reconstruction," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3827–3839, 2021.
- [27] M. Özel, "Drone dataset (uav)," <https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav>, 2022, accessed: 2025-04-08.
- [28] A. Ng and K. Katanforoosh, "Cs230: Deep learning - section 8: Advanced evaluation metrics," 2023, accessed: 2025-04-08. [Online]. Available: <https://cs230.stanford.edu/section/8/>