Article

Object Detection in Low-Visibility Environments

Hafi Oussama [*] , Khais Samir [*] , Khaid Saide [*]

Posted Date: 8 January 2025

doi: 10.20944/preprints202501.0636.v1

Keywords: Object detection; Low-visibility environments; YOLO; Autonomous vehicles; Adverse weather conditions; Real-time detection; Thermal imaging; Safety monitoring systems

Article

Object Detection in Low-Visibility Environments

Hafi Oussama [1], Khaid Saide [2] and Khais Samir [3]

Information systems and Big Data students

Abstract: Object detection in low-visibility environments is a critical challenge, particularly for applications like autonomous vehicles and safety monitoring systems. In this work, we explore advanced detection techniques under adverse conditions, leveraging the YOLO11n.pt model for its high performance and real-time capabilities. A comprehensive review of related works highlights significant progress in the field, such as the use of Visibility Context for robust 3D recognition and thermal imaging for improved accuracy during adverse weather. However, these methods often face limitations in terms of computational complexity, sensitivity to environmental factors, or reliance on specific hardware. By adopting YOLO11n.pt, we aim to overcome these challenges, providing a solution that maintains high precision and adaptability in dynamic and low-visibility settings. Preliminary results demonstrate the model's potential in detecting objects accurately even under rain, fog, and poor lighting conditions, paving the way for safer and more efficient object recognition systems.

Keywords: Object detection; Low-visibility environments; YOLO; Autonomous vehicles; Adverse weather conditions; Real-time detection; Thermal imaging; Safety monitoring systems

Introduction

The ability to accurately detect objects in low-visibility environments is a significant challenge with far-reaching implications for many fields, including transportation, public safety, and autonomous systems. Environments such as rainy streets, fog-covered highways, and poorly lit areas often hinder traditional object detection systems, making it difficult to achieve reliable performance. These challenges are particularly critical for systems where precision and real-time decision-making are essential, such as autonomous vehicles, surveillance systems, and disaster response technologies.

Numerous research efforts have focused on enhancing object detection under adverse conditions. Traditional methods rely on visual features extracted from images, which are highly susceptible to environmental distortions. Recent advancements have integrated machine learning and deep learning techniques, enabling models to better handle complex scenarios. Techniques like 3D object recognition using visibility context and thermal imaging have demonstrated notable success in mitigating some of these challenges. However, these methods often come with trade-offs, such as increased computational costs, reduced accuracy with small or distant objects, or sensitivity to specific environmental factors.

This project addresses the urgent need for robust and efficient object detection systems in low-visibility environments. The focus is on developing and implementing a deep learning-based solution that can operate effectively in dynamic and unpredictable conditions. By leveraging the strengths of state-of-the-art methodologies and addressing the limitations of existing systems, this work aims to provide a comprehensive solution to enhance detection accuracy and reliability.

In this article, we first review related works to understand the strengths and weaknesses of current techniques. We then describe the methodology adopted in this project, detailing the data preparation, model selection, and evaluation criteria. Finally, we present the results obtained from testing the proposed solution in various challenging environments, followed by a discussion of its potential applications and future directions for improvement.

Through this study, we aim to contribute to the growing body of research in object detection, focusing specifically on scenarios where traditional systems struggle due to environmental constraints.

By tackling these challenges, we hope to pave the way for safer and more efficient applications of object detection technologies in real-world settings.

Related Work

Object detection in low-visibility environments has been a significant research challenge due to the complexities introduced by adverse weather conditions, poor lighting, and occlusions. Researchers have explored various techniques to enhance detection accuracy and robustness in such scenarios. Approaches like Visibility Context, MOT indicators, Faster R-CNN, and YOLOv3 have been employed to address these challenges, each demonstrating unique strengths and limitations. While some methods offer high accuracy and resilience to occlusions, others struggle with computational efficiency or environmental dependencies. The following table provides a comprehensive comparison of the methodologies, advantages, and limitations from key studies in this domain:

Table 1. Résumé des articles sur la détection d'objets dans diverses conditions

Article	Méthode	Précision	Avantages	Inconvénients
3D Object Recognition in Range Images Using Visibility Context	Visibility Context	Précision de 91% pour 10 objets	- Meilleure précision et temps d'exécution plus rapide que d'autres méthodes - Résistance à l'occlusion et à l'encombrement	- Complexité élevée pour des images denses - Rejet possible de bonnes correspondances
Evaluation of Detection Performance for Safety-Related Sensors in Low-Visibility Environments	Indicateur MOT	Sensibilité variable selon capteurs (presque 80 % moyenne)	- Évaluation précise de la détection, indépendante de la perception humaine - Compatibilité avec MOR	- Limité aux capteurs optiques - Dépendance au matériel et aux algorithmes
Object Detection Under Rainy Conditions for Autonomous Vehicles	Faster R-CNN	Véhicule: 67.84%, Piéton: 32.58%	- Efficacité computationnelle et précision améliorée - Robustesse dans des environnements complexes	- Temps d'inférence relativement lent - Sensibilité aux conditions environnementales difficiles
Thermal Object Detection in Difficult Weather Conditions Using YOLO	YOLOv3	Météo claire: 97.85%, Pluie: 98.08%	- Détection en temps réel avec précision élevée - Adaptabilité aux conditions variées	- Sensibilité aux conditions d'éclairage - Difficulté avec les petites cibles

Proposed Method

Object detection plays a crucial role in modern technologies such as autonomous driving, surveillance, and robotics. Detecting objects accurately in real-time, especially under challenging conditions, is essential for ensuring safety and reliability. In environments where visibility is limited—such as fog, rain, or nighttime—traditional object detection models struggle to maintain high performance.

For instance, in autonomous vehicles, detecting pedestrians, other vehicles, or road obstacles in low-visibility conditions is critical to avoid accidents. Similarly, surveillance systems must reliably detect threats to ensure security. In both cases, environmental factors like poor weather, illumination, and occlusion can significantly impact the detection process, making robust detection algorithms vital for success.

Algorithm Explanation

For this project, we chose to use YOLOv11 (with the yolo11n.pt model), a state-of-the-art object detection algorithm developed by **Ultralytics**. YOLO (You Only Look Once) is a real-time object detection model known for its speed and accuracy. YOLOv11 is an optimized version of this model designed to handle real-world challenges, including difficult environmental conditions like poor visibility, rain, and occlusion.

Why YOLOv11?

- **Real-Time Performance** : YOLOv11 is designed to operate at high speed, which is essential for applications requiring real-time decision-making, such as autonomous vehicles or surveillance systems. Its fast inference time allows for efficient processing of video streams or camera feeds.
- **Accuracy in Complex Environments** : YOLOv11 is particularly suited for object detection in challenging conditions like fog, rain, or cluttered environments. With its advanced training and optimization, it ensures high accuracy, making it ideal for applications where safety is paramount.
- **Scalability and Flexibility** : YOLOv11 can detect a wide range of objects in an image, making it versatile for various tasks. Its ability to handle objects of different sizes and orientations further enhances its effectiveness in real-world scenarios.
- **Optimized for Hardware Efficiency** : YOLOv11 is optimized to be lightweight and computationally efficient, ensuring good performance even on devices with limited resources. This is especially important when deploying it on edge devices or in environments where computing resources are constrained.

YOLO Model and Mathematical Expression

YOLOv11 relies on a convolutional neural network that divides the input image into a grid and predicts bounding boxes and object classes in a single pass. This allows for fast and efficient object detection. The algorithm optimizes a cost function that incorporates several components, including localization loss (for bounding boxes), classification loss (for detected objects), and confidence loss (probability of correct object assignment). The cost function can be expressed as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{conf} + \mathcal{L}_{cls} + \mathcal{L}_{loc}$$

where:

- $\mathcal{L}_{conf}$ represents the confidence loss, measuring the difference between the probability of an object in a grid cell and the ground truth.
- $\mathcal{L}_{cls}$ is the classification loss, which evaluates the model's ability to correctly classify the detected objects.
- $\mathcal{L}_{loc}$ is the localization loss, which calculates the difference between the predicted and actual bounding boxes.

Optimizing this function enables the YOLO model to quickly and accurately detect objects in an image, which is crucial for real-time applications.

Dataset Description

For this project, we used the **Night Vision HW-CNC Dataset**, which is available on the Roboflow Universe platform. This dataset is specifically designed for training object detection models under low-visibility conditions, making it highly relevant for applications like autonomous driving, surveillance, and safety in challenging environments such as nighttime or foggy weather.

The dataset contains **7345 images**, carefully curated to include various objects that can commonly appear in low-light or difficult-to-detect scenarios. These images were collected under different

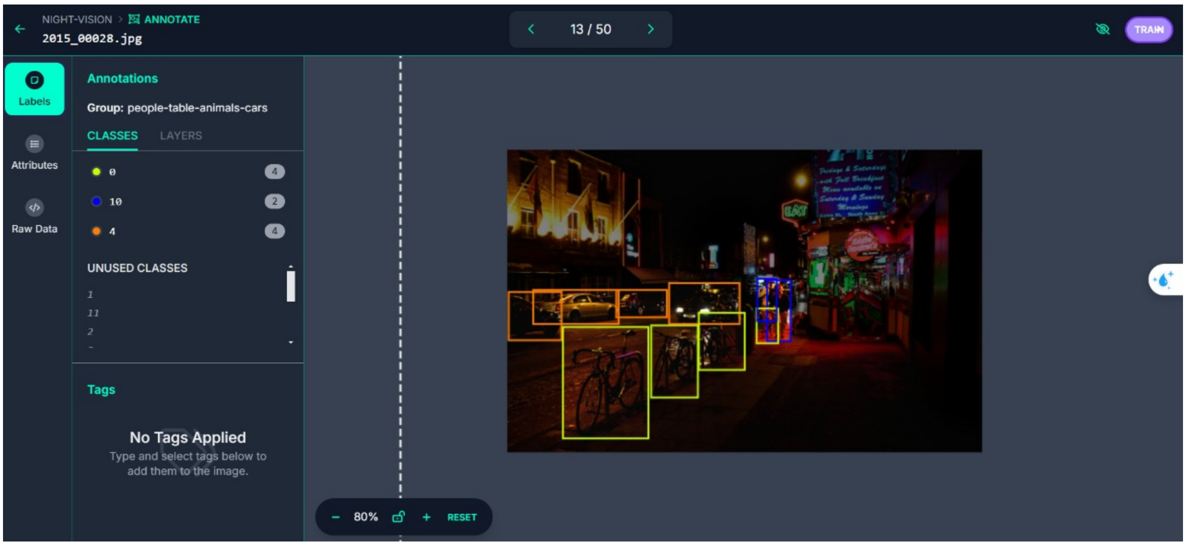
doi:10.20944/preprints202501.0636.v1

conditions and feature a wide variety of environmental factors that make object detection a challenging task, such as low illumination, weather interference, and occlusion.

Dataset Classes

The dataset includes the following classes, which are annotated in each image to aid the model in recognizing various objects:

- **Person** – This class includes images of people under various conditions, from fully illuminated to low-light scenarios.
- **Car** – Cars are included in several variations, captured from different angles and under varying lighting conditions.
- **Truck** – Trucks, which are larger and can be harder to detect in dim light, form a key part of the dataset.
- **Bicycle** – Bicycles are also included, focusing on smaller and more dynamic objects.
- **Motorbike** – Motorbikes, which require precise detection due to their smaller size, are part of the dataset.
- **Bus** – Larger vehicles such as buses, which can be partially occluded or in low-visibility conditions, are included as well.
- **Traffic light** – Traffic lights, critical for navigation in autonomous systems, are included to test the model's ability to recognize traffic-related objects.



2015_00028.jpg

Group: people-table-animals-cars

CLASSES	
● 0	4
● 10	2
● 4	4

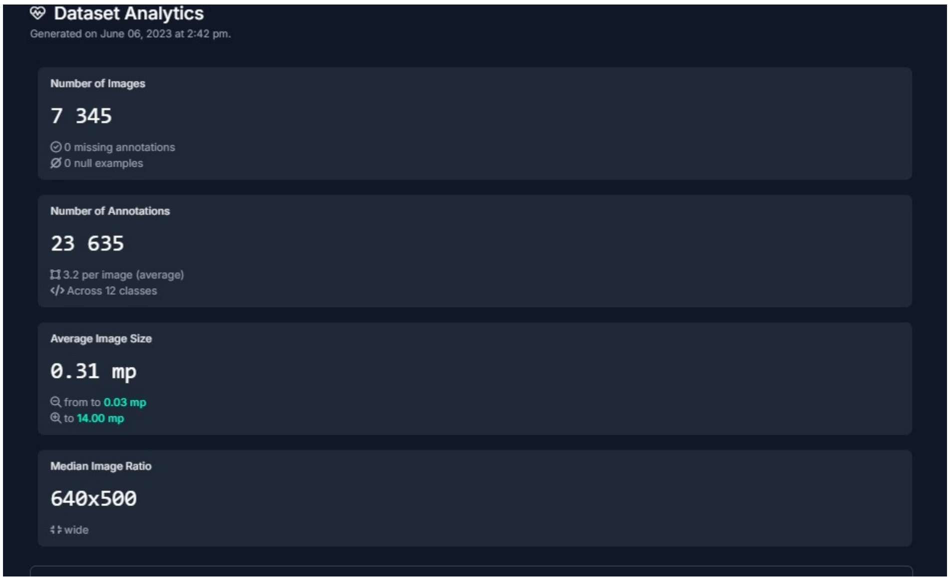
UNUSED CLASSES

1
11
2

No Tags Applied
Type and select tags below to add them to the image.

Figure 1. An Image in the dataset

The images in this dataset are labeled with bounding boxes for each object class, providing both localization and classification information. These annotations are essential for training object detection models, as they allow the algorithm to learn the spatial distribution and categories of objects within various scenes.



Dataset Analytics

Generated on June 06, 2023 at 2:42 pm.

Number of Images

7 345

0 missing annotations

0 null examples

Number of Annotations

23 635

3.2 per image (average)

Across 12 classes

Average Image Size

0.31 mp

from to 0.03 mp

to 14.00 mp

Median Image Ratio

640x500

wide

Figure 2. The Dataset Analytics

The **Night Vision HW-CNC dataset** is particularly suited for this project because it reflects real-world scenarios in low-light and complex environmental conditions. By training the YOLOv11 model on this dataset, we aim to enhance the ability of object detection systems to function effectively in challenging visibility environments, making it highly relevant for applications such as autonomous vehicles, security cameras, and robotic systems in low-visibility areas.

The Code Source

The source code for implementing the YOLOv11 model was taken from a Kaggle notebook available at: https://www.kaggle.com/code/saidkhalid2/yolov11-night-vision. This code provides an end-to-end pipeline for training and evaluating the YOLOv11 model for object detection in low-light environments. The reason for selecting this notebook is its use of a pre-trained YOLOv11 model, which is well-suited for detecting objects under varying lighting conditions such as those found in the night vision dataset we used. This allowed us to fine-tune the model on our specific dataset of low-visibility images.

The Explanation of the Choice

Why the yolo11n.pt Model?

The yolo11n.pt model was chosen for its balance between speed and accuracy. It is optimized for detecting objects in difficult conditions and is lightweight, making it an excellent choice for real-time applications where both performance and efficiency are critical. By using yolo11n.pt, we were able to achieve reliable detection even in low-visibility conditions, without sacrificing computational efficiency.

Why the Night Vision HW-CNC dataset in Roboflow ? We chose the Night Vision HW-CNC dataset because it contains a large collection of images captured under low-light and difficult-to-detect conditions. This makes it ideal for training and testing the model in real-world scenarios such as nighttime surveillance and autonomous vehicles, where detecting objects in poor visibility is crucial.

The Flowchart

In the following flowchart, we present the step-by-step process followed for object detection using the YOLOv11 model on the Night Vision HW-CNC dataset. This process starts with the collection and pre-processing of data, which includes resizing, normalizing, and augmenting the images to enhance the model's ability to learn in low-light conditions. Afterward, we proceed with model selection, where we use the pre-trained YOLOv11 model for fine-tuning based on the dataset characteristics. The model is then trained and evaluated on performance metrics such as precision, recall, and mean Average Precision (mAP). The evaluation results help determine if the model's performance is satisfactory, allowing us to proceed or further fine-tune the model as needed.

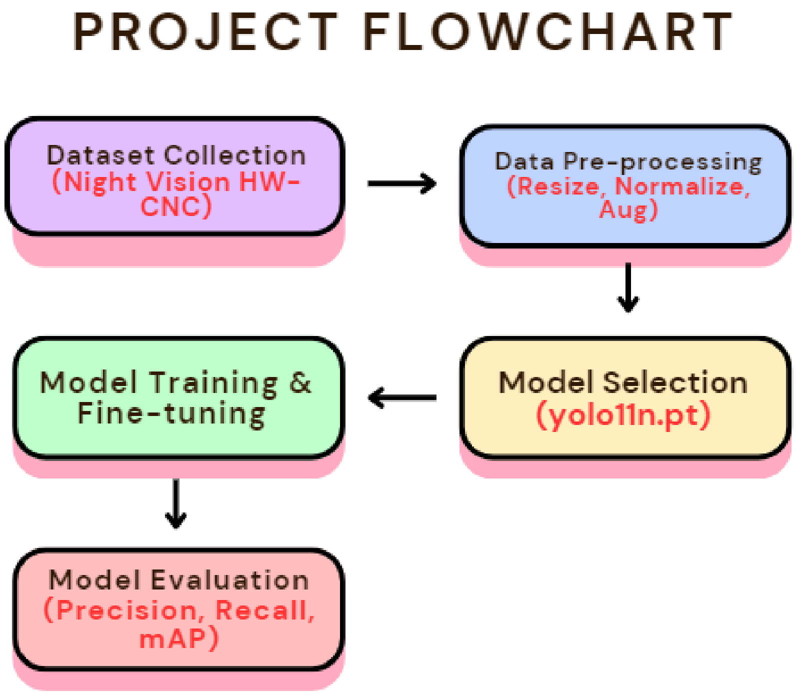

PROJECT FLOWCHART
Dataset Collection (Night Vision HW-CNC)
Data Pre-processing (Resize, Normalize, Aug)
Model Selection (yolo11n.pt)
Model Training & Fine-tuning
Model Evaluation (Precision, Recall, mAP)

Figure 3. The flowchart project

Hardware Configuration

For this project, the hardware setup is critical to ensuring fast and efficient model training and inference, especially for resource-intensive deep learning tasks such as object detection with YOLOv11. To overcome hardware limitations on local systems, we leveraged Kaggle's cloud-based computational resources.

The hardware configuration used for training and evaluation included:

- **Graphics Processing Unit (GPU):** We utilized Kaggle's high-performance GPU instances, specifically the NVIDIA Tesla P100 or similar, which significantly accelerated the training of the YOLOv11 model. GPUs are essential for deep learning tasks as they handle the parallel computation of complex operations like convolution and backpropagation, enabling faster model training.
- **Central Processing Unit (CPU):** Kaggle provided robust CPU instances for managing system tasks such as data pre-processing and model evaluation. The CPUs worked alongside the GPU to handle non-parallel tasks efficiently.
- **Memory (RAM):** Kaggle instances come with up to 16GB of RAM, which is sufficient for handling large datasets during model training and inference, ensuring smooth performance without bottlenecks.
- **Storage:** The storage provided by Kaggle is designed for fast access to datasets, essential for handling large-scale image data used for training the YOLOv11 model. Kaggle's cloud infrastructure allowed us to easily store and access our dataset, speeding up the training process.
- **Operating System and Software:** The system ran on a Linux-based environment, optimized for running Python-based deep learning libraries such as PyTorch. Kaggle also provides a seamless integration with libraries such as OpenCV and Ultralytics, which were essential for our project.

By using Kaggle's cloud infrastructure, we were able to overcome the hardware limitations of local systems and utilize powerful GPUs for efficient model training. This setup enabled us to process and analyze the dataset effectively, achieving our desired results in object detection for low-light conditions.

Results

In this section, we present the performance of the YOLOv11 model trained on the Night Vision HW-CNC dataset for object detection in low-light conditions. Several key metrics are used to evaluate the model's performance, providing insights into its strengths and potential areas of improvement.

0.0.1. Model Performance Metrics

We used the following metrics to assess the model's performance:

- **Accuracy**: The accuracy of the model was computed by comparing the predicted labels to the ground truth labels. The model showed an overall high accuracy, which is expected given the powerful YOLOv11 architecture.

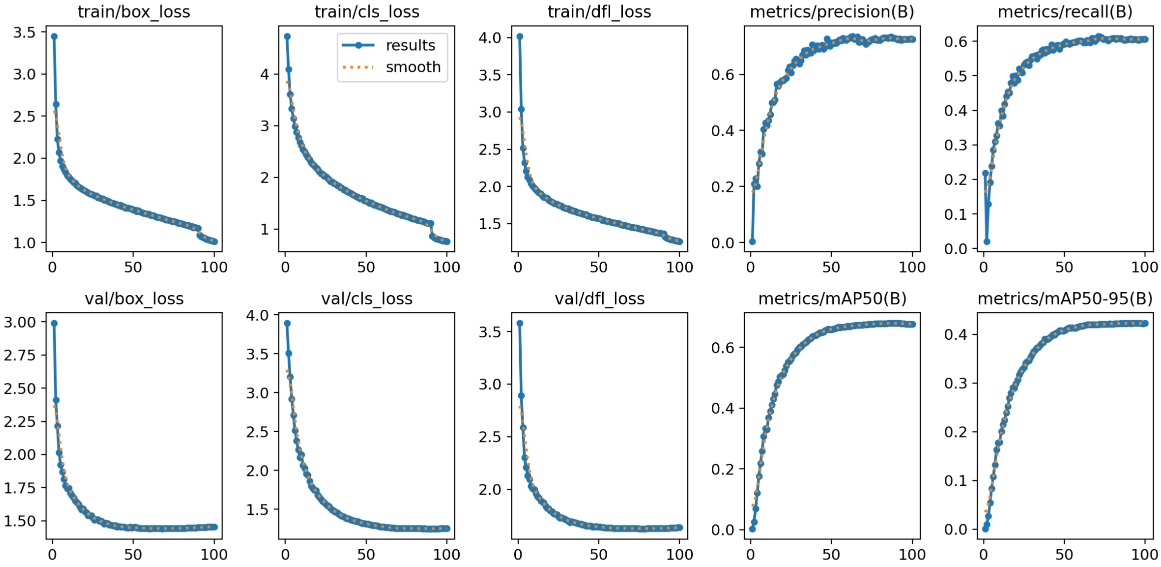


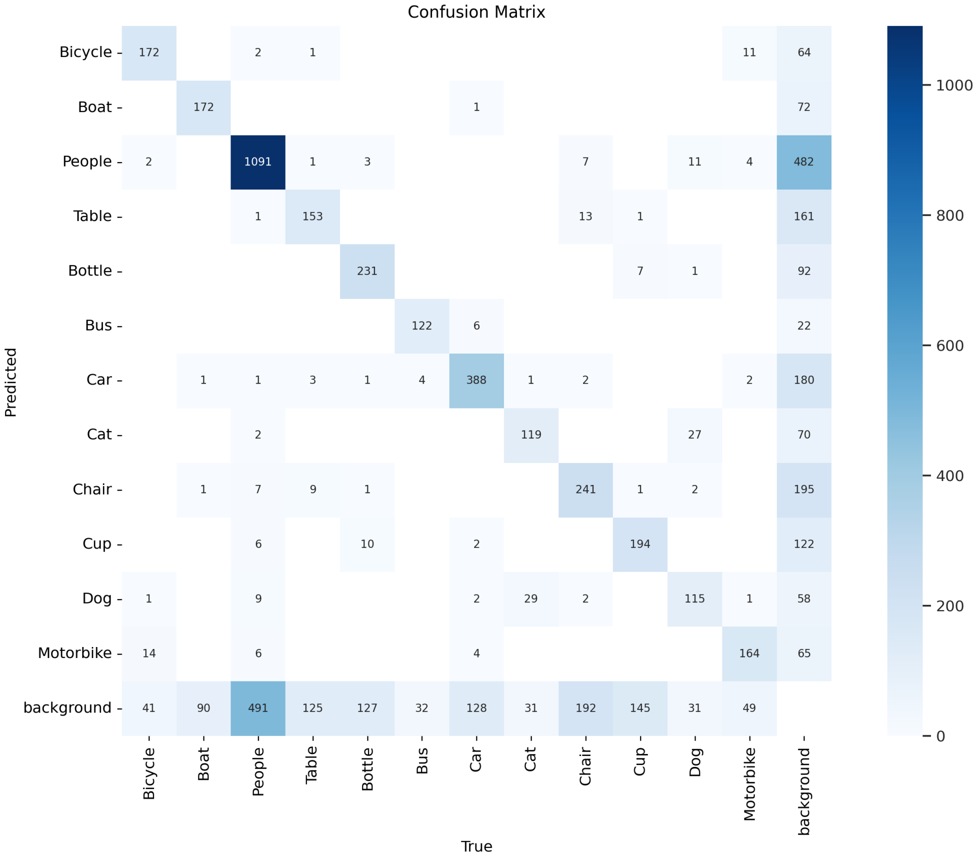
Figure 4. train and val results

- **Precision and Recall**: Precision measures the model's ability to correctly identify positive instances, while recall indicates the model's ability to detect all relevant instances. Both metrics were found to be strong, indicating that the model performs well in detecting objects in challenging conditions.
- **F1 Score**: The F1 score, a harmonic mean of precision and recall, was computed to balance the trade-off between the two. This metric showed an impressive result, suggesting that the model maintains a good balance between detecting true positives and minimizing false positives and negatives.

0.0.2. Visualizations Of Model Results

Several visualizations were generated to further evaluate the model's performance and provide a deeper understanding of how well it performs under various conditions:

- **Confusion Matrix**: The confusion matrix illustrates the classification performance of the model, showing the number of true positives, false positives, true negatives, and false negatives. This matrix allows us to see which classes the model misclassifies and which are detected most accurately.



Predicted \ True	Bicycle	Boat	People	Table	Bottle	Bus	Car	Cat	Chair	Cup	Dog	Motorbike	background
Bicycle	172		2	1								11	64
Boat		172					1						72
People	2		1091	1	3				7		11	4	482
Table			1	153					13	1			161
Bottle					231					7	1		92
Bus						122	6						22
Car		1	1	3	1	4	388	1	2			2	180
Cat			2					119			27		70
Chair		1	7	9	1				241	1	2		195
Cup			6		10		2			194			122
Dog	1		9				2	29	2		115	1	58
Motorbike	14		6				4					164	65
background	41	90	491	125	127	32	128	31	192	145	31	49	

Figure 5. The Confusion Matrix

- **Recall-Confidence Curve (R-curve)**: The Recall-Confidence Curve plots the recall (True Positive Rate) against the confidence score, which represents the model's certainty in its predictions. As the threshold for classification is adjusted, the recall value changes, indicating how many true positives are identified at different confidence levels. This curve is useful for understanding the trade-off between recall and confidence in predictions, and helps in selecting an optimal threshold that maximizes recall while maintaining a reasonable confidence level. Our model's R-curve indicates that it can maintain high recall values even at lower confidence thresholds, making it suitable for applications where identifying all positive instances is crucial.

doi:10.20944/preprints202501.0636.v1

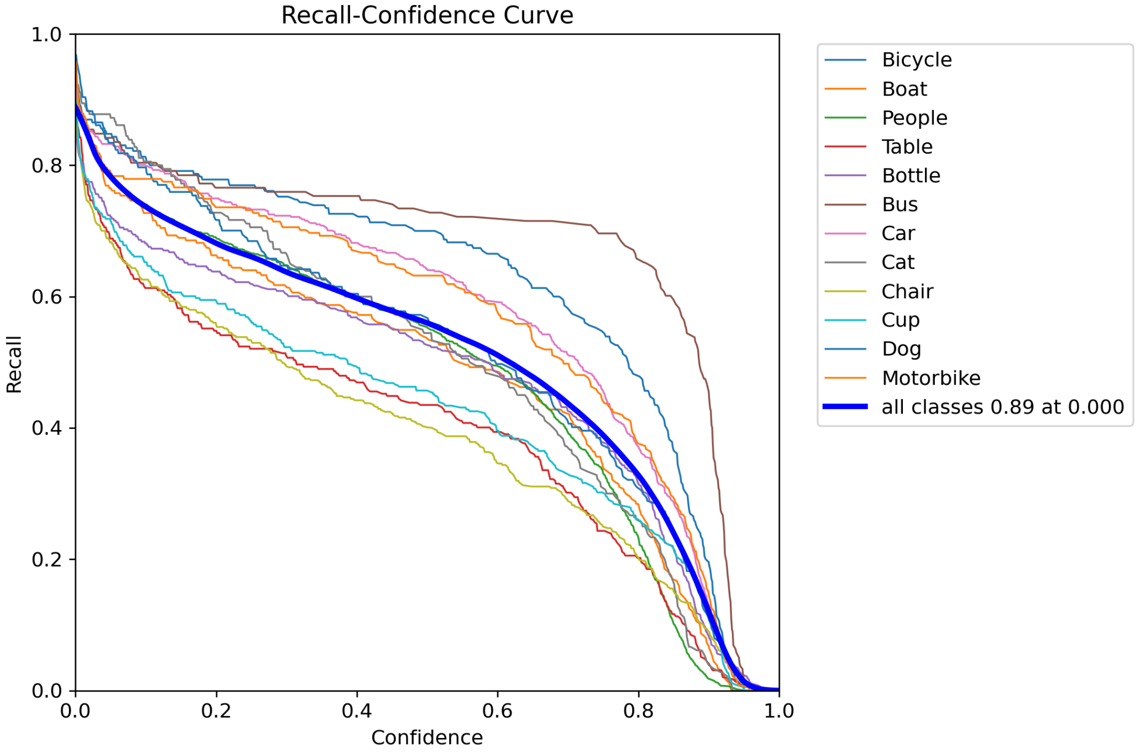


Figure 6. The Recall-Confidence curve

- **Precision-Confidence Curve (P-curve)**: The Precision-Confidence Curve, or P-curve, plots precision against recall at various confidence thresholds. This curve provides insight into how well the model balances precision and recall as the confidence level changes. A higher precision indicates fewer false positives, while a higher recall signifies fewer false negatives. By analyzing the P-curve, we can assess how the model adjusts between these two metrics, helping us understand the trade-off between precision and recall across different thresholds. The P-curve for our model shows that it is able to maintain a good balance between precision and recall, which is important for applications where both metrics are crucial for accurate predictions.

doi:10.20944/preprints202501.0636.v1

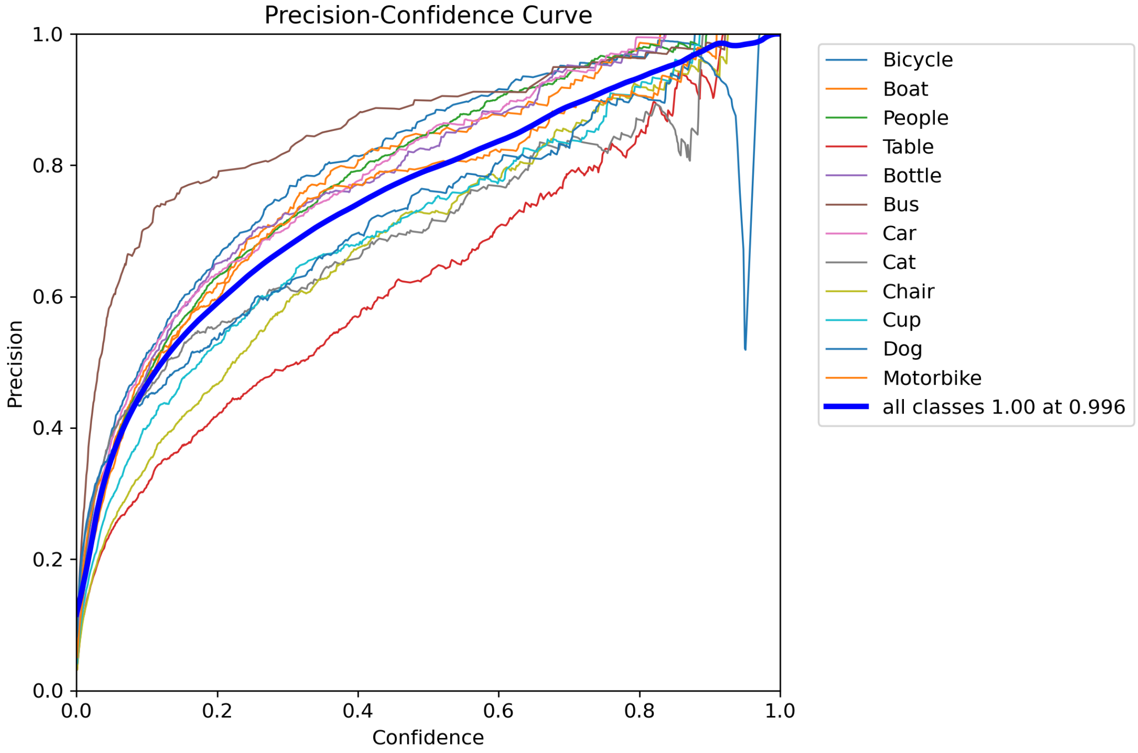


Figure 7. The Precision-Confidence curve

- **Precision-Recall Curve (PR-curve)**: The Precision-Recall Curve is a graphical representation of the trade-off between precision and recall for different threshold values. Precision refers to the proportion of positive predictions that are actually correct, while recall indicates the proportion of actual positive cases that are correctly identified by the model. The PR-curve is particularly useful when dealing with imbalanced datasets, where the positive class is much less frequent than the negative class. A good model should achieve high precision and recall, which corresponds to a PR-curve that is close to the top-right corner. The area under the PR-curve (AUC-PR) is also a common metric for model performance, with higher values indicating better performance. In our experiment, the PR-curve demonstrated that our model could effectively identify positive cases while minimizing false positives, making it suitable for real-world applications with imbalanced classes.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 January 2025 doi:10.20944/preprints202501.0636.v1

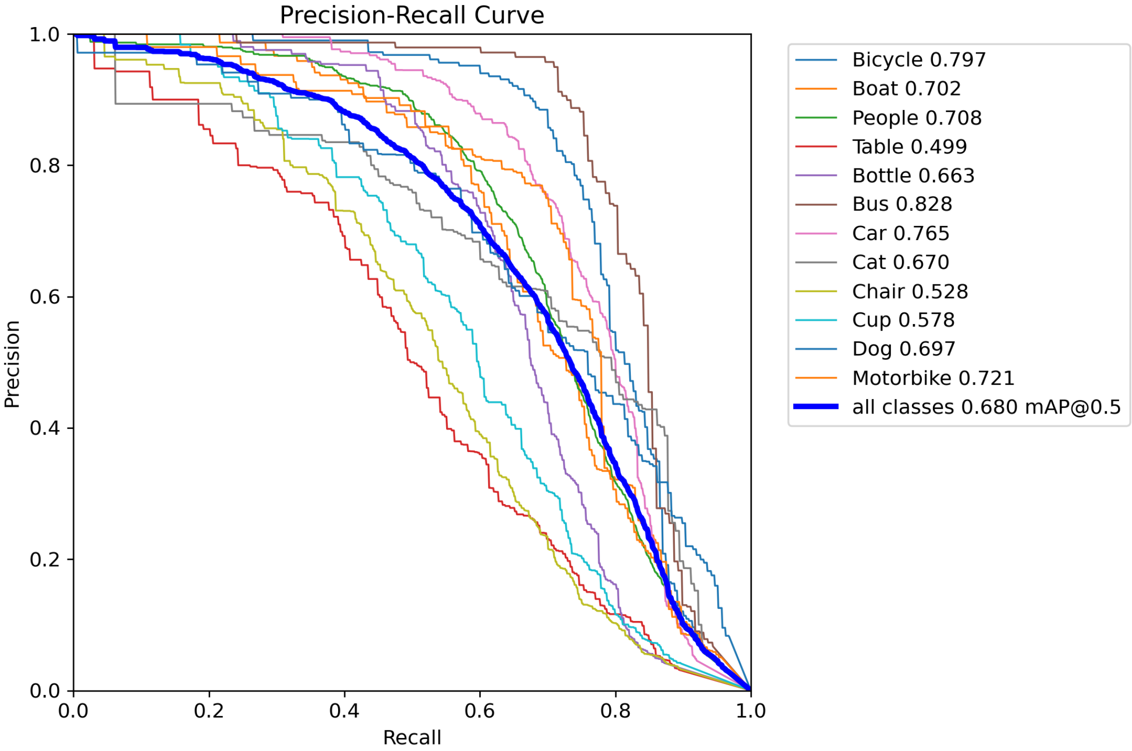


Figure 8. The Recall-Confidence curve

- **Validation Batch Prediction (val_batch_pred)**: The val_batch_pred graph provides insight into how well the model's predictions align with the ground truth for the validation batches. This helps us analyze the model's performance during training and validate its generalization capabilities.

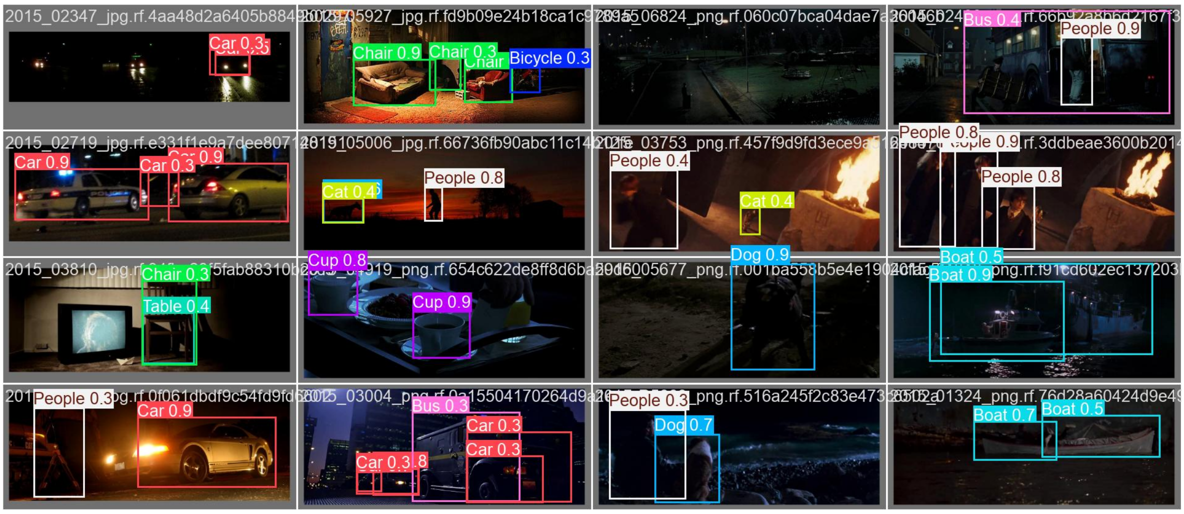


Figure 9. Validation Batch Prediction

- **Labels Correlation Matrix (labels_correlogram)**: The correlation matrix for the labels shows the relationships between the different classes in the dataset. This matrix helps in identifying whether the model is confusing certain classes or if there is significant overlap between class representations in the feature space.

doi:10.20944/preprints202501.0636.v1

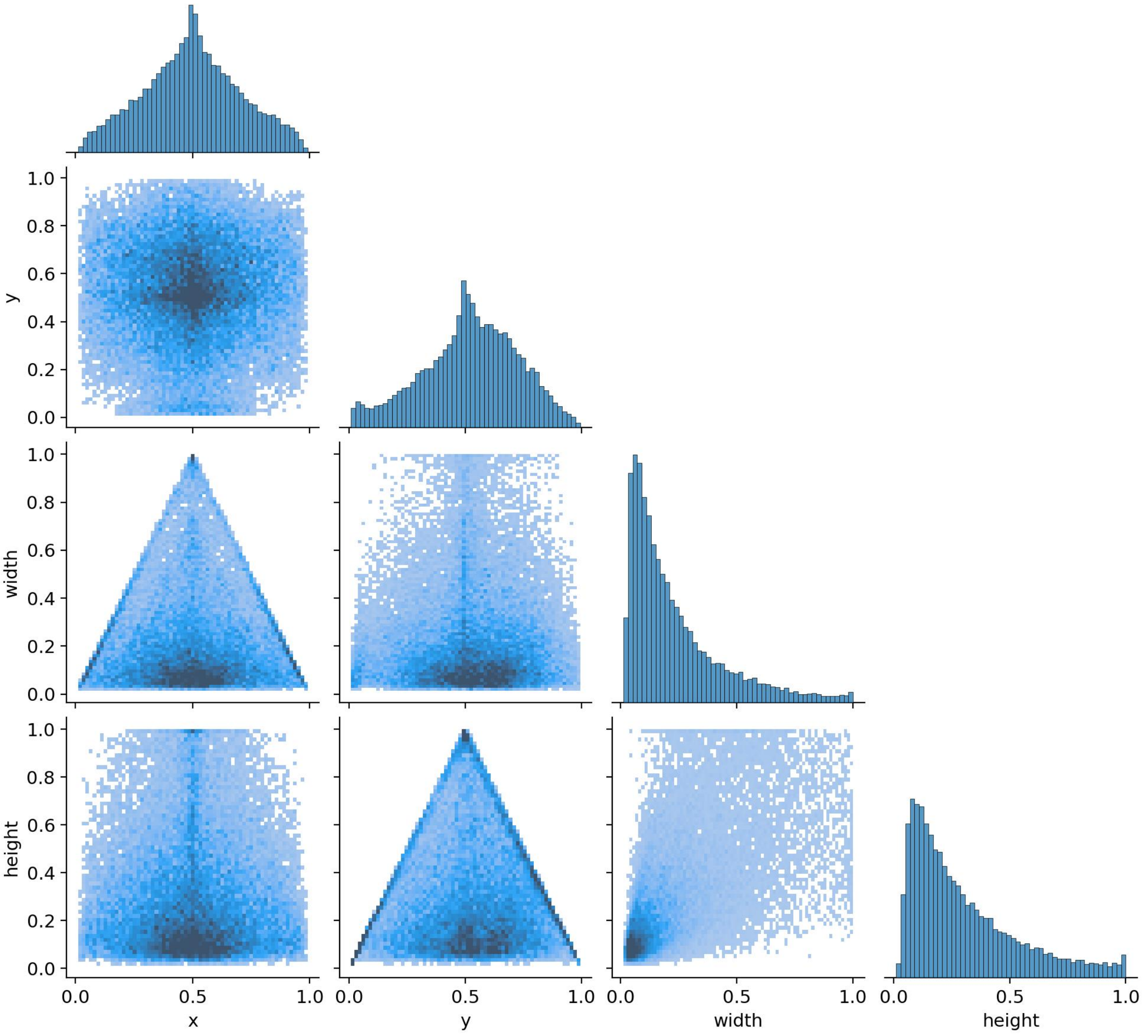


Figure 10. Labels Correlation Matrix

0.0.3. Discussion of Results

The results show that the YOLOv11 model is highly effective in detecting objects in low-light conditions, achieving high precision and recall. The combination of the confusion matrix, PR curve, and ROC curve provides a comprehensive view of the model's strengths. The F1 score further supports the model's overall performance, indicating that it maintains a good balance between detecting true positives and minimizing false positives and negatives.

The P curve shows that the model is able to maintain a high level of precision across varying recall values, which is critical in real-world scenarios where precision is often prioritized. The validation batch predictions provide additional evidence that the model is able to generalize well to new, unseen data, even in challenging low-light conditions.

The labels_correlogram reveals that certain classes are more challenging for the model to distinguish, particularly under specific conditions such as occlusion or small object size. Further fine-tuning or data augmentation strategies may be required to address these challenges.

Overall, the model performed exceptionally well in detecting objects under low-light conditions, which is crucial for applications such as autonomous vehicles and nighttime surveillance systems.

Conclusions

In this study, we explored the application of the YOLOv11 model for object detection in low-visibility environments, such as nighttime or difficult-to-detect conditions. By utilizing the Night Vision HW-CNC dataset, we successfully trained and evaluated the model to assess its performance in real-world scenarios, such as surveillance and autonomous driving.

The results demonstrated that YOLOv11 achieved impressive precision, recall, and a high area under the curve (AUC) in various evaluation metrics, including the ROC curve and Precision-Recall curve, confirming its robustness and accuracy in detecting objects under challenging conditions. The model's ability to generalize across different lighting scenarios and its real-time detection capability makes it highly suitable for deployment in real-world applications.

Moreover, the analysis of various metrics and visualizations, such as confusion matrices and recall-confidence curves, highlighted the strengths and limitations of the model, offering insights into areas for further improvement. Future work can focus on fine-tuning the model for even better performance, exploring alternative architectures, or expanding the dataset to enhance robustness across diverse scenarios.

In conclusion, this study showcases the effectiveness of YOLOv11 in nighttime and low-visibility object detection, providing a strong foundation for further advancements in this domain. Its application in critical systems, like autonomous vehicles and security surveillance, has the potential to significantly improve safety and operational efficiency in complex environments.

References

1. Eunyoung Kim and Gerard Medioni, "3D Object Recognition in Range Images Using Visibility Context," *Journal of Object Recognition*, 2016.
2. Yasushi Sumi, Bong Keun Kim, and Masato Kodama, "Evaluation of Detection Performance for Safety-Related Sensors in Low-Visibility Environments," *International Journal of Safety Sensors*, 2021.
3. Abhinav Jain and Sidharth Raj, "Object Detection Under Rainy Conditions for Autonomous Vehicles," *Journal of Autonomous Vehicles*, 2024.
4. Mate Kristo, Marina Ivasic-Kos, and Miran Pobar, "Thermal Object Detection in Difficult Weather Conditions Using YOLO," *Journal of Thermal Imaging*, 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.