

Review

Performance Analysis of YOLO for Object Detection under Complex Illumination Conditions

Tiantian Chen ¹ and Sheila M. Geronimo ^{1,*}

¹ University of the East, Manila, Philippines

* Correspondence: Sheila M. Geronimo, University of the East, Manila, Philippines

Abstract: Object detection has become a crucial component in various computer vision applications, ranging from autonomous driving to surveillance systems. YOLO (You Only Look Once) has gained prominence due to its real-time performance and relatively high accuracy. However, its performance can be significantly affected by complex illumination conditions such as underexposure, overexposure, shadows, and varying light sources. This review paper provides a comprehensive analysis of YOLO's performance under these challenging illumination conditions. We begin with an overview of the YOLO architecture and its evolution, followed by a detailed exploration of how different illumination factors impact its detection accuracy and speed. We then delve into various techniques proposed to mitigate these issues, including image enhancement methods, adaptive thresholding approaches, and robust feature extraction strategies. Furthermore, we comparatively analyze the performance of different YOLO variants and other state-of-the-art object detectors under diverse illumination scenarios. The paper synthesizes the current research landscape, highlights the key challenges that remain, and discusses potential future directions for improving the robustness of object detection algorithms in adverse lighting conditions. This review aims to serve as a valuable resource for researchers and practitioners seeking to understand and address the limitations of YOLO in real-world applications with complex illumination.

Keywords: YOLO, Object Detection, Complex Illumination, Image Enhancement, Adverse Lighting, Computer Vision, Deep Learning

1. Introduction

1.1. Background and Motivation

Object detection, a cornerstone of modern computer vision, plays a crucial role in various applications, including autonomous driving, surveillance systems, and robotics. The ability to accurately identify and locate objects within an image or video stream is paramount for these systems to function effectively. However, real-world scenarios often present significant challenges, particularly complex illumination conditions. Variations in lighting, shadows, and reflections can drastically affect the performance of object detection algorithms.

These illumination changes impact feature extraction, leading to inaccurate object recognition and localization. The You Only Look Once (YOLO) architecture has emerged as a popular choice for object detection due to its speed and efficiency. YOLO's single-stage detection approach allows for real-time processing, making it suitable for applications where speed is critical. Understanding YOLO's performance under varying illumination (*I*) is therefore essential for its reliable deployment in real-world applications [1].

Received: 13 January 2026

Revised: 03 March 2026

Accepted: 13 March 2026

Published: 19 March 2026



Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1.2. Problem Statement and Scope

YOLO's performance is significantly challenged by complex illumination conditions, leading to decreased detection accuracy and increased false positives. Specifically, issues arise from variations in lighting intensity, including underexposure and overexposure, as well as non-uniform illumination causing shadows and highlights. Furthermore, color distortion and changes in contrast due to different light sources degrade feature extraction. This review focuses on analyzing the impact of these illumination challenges on YOLO's object detection capabilities. The scope encompasses an investigation into YOLOv3, YOLOv4, YOLOv5, and YOLOv7, evaluating their performance under simulated and real-world conditions exhibiting low-light, high-light, and heterogeneous illumination. The analysis will consider metrics such as mAP, precision, and recall to quantify the degradation in performance under these adverse lighting scenarios [2].

2. Historical Overview of YOLO

2.1. YOLOv1 and its Innovations

YOLOv1, introduced by Redmon et al., represented a paradigm shift in object detection by framing it as a regression problem [3]. Instead of relying on region proposals, YOLOv1 processes the entire image in a single pass, dividing it into an $S \times S$ grid. Each grid cell predicts B bounding boxes along with confidence scores for those boxes, indicating the probability of an object being present and the accuracy of the box. Simultaneously, each grid cell predicts C class probabilities, representing the conditional probability of each class given that an object is present in the grid cell. These predictions are encoded into a single tensor of size $S \times S \times (B \times 5 + C)$. As shown in Table 1, YOLOv1's performance is significantly faster compared to other object detection methods, but it comes with a trade-off in accuracy, especially in handling small or clustered objects.

Table 1. Performance comparison of YOLOv1 with other object detection methods on various datasets.

Metric	YOLOv1 (Compared to other methods)	Reason for Performance
Speed	Significantly Faster	Single-pass approach; framing detection as a regression problem. No region proposals needed.
Accuracy (General)	Lower	Trade-off for speed; Simpler architecture and loss function.
Small Objects	Lower Accuracy	Grid-based approach limits detection of small, clustered objects. Limited number of bounding boxes per grid cell (B).
Clustered Objects	Lower Accuracy	Each grid cell predicts a maximum of B bounding boxes. Unable to handle high density of objects.
Unusual Aspect Ratios	Lower Accuracy	Limited generalization ability due to simpler architecture.
Complex Scenes (e.g., varying illumination)	Lower Accuracy	Less robust to variations in illumination and scene complexity compared to more sophisticated methods.
Tensor Output Size	Represented as $S \times S \times (B \times 5 + C)$	The encoding of predicted bounding boxes, confidence scores, and class probabilities.

This single-pass approach enabled YOLOv1 to achieve significantly faster processing speeds compared to its predecessors, making it suitable for real-time applications. However, this speed came at the cost of accuracy, particularly in handling small objects and objects clustered closely together. The grid-based approach limited the number of objects that could be detected in close proximity, as each grid cell could only predict a limited number of bounding boxes. Furthermore, YOLOv1 struggled with generalizing to objects with unusual aspect ratios or those appearing in complex scenes with varying illumination conditions due to its relatively simple architecture and loss function.

2.2. Evolution of YOLO: v2, v3, v4, v5, and Beyond

YOLO's initial success spurred rapid development, leading to subsequent versions with significant architectural and performance enhancements. Table 2 summarizes the key architectural improvements across different YOLO versions, highlighting the innovations that each version brought to the model. YOLOv2 introduced anchor boxes, borrowing the concept from Faster R-CNN, to improve the model's ability to predict objects of varying sizes and aspect ratios. This version also incorporated batch normalization and a higher resolution classifier, resulting in improved accuracy and faster training. Furthermore, YOLOv2 adopted Darknet-19 as its backbone network, a custom neural network designed for speed and efficiency.

Table 2. Key architectural improvements in different YOLO versions.

YOLO Version	Key Architectural Improvements
YOLOv2	Introduced anchor boxes, batch normalization, higher resolution classifier, and Darknet-19 backbone.
YOLOv3	Employed Darknet-53 with residual connections and multi-scale predictions.
YOLOv4	Incorporated CSPDarknet53, Mish activation, and mosaic data augmentation; optimization through "freebies" and "specials".
YOLOv5	Implemented in PyTorch, allowing easier experimentation and model scaling (n, s, m, l, x).
YOLOv6, YOLOv7, YOLOv8	Focused on further improvements in speed, accuracy, and efficiency, incorporating techniques such as knowledge distillation, improved loss functions, and optimized network architectures.

YOLOv3 further refined the architecture by employing a more sophisticated feature extractor, Darknet-53, which incorporated residual connections to mitigate the vanishing gradient problem and enable the training of deeper networks. This version also implemented multi-scale predictions, allowing the model to detect objects at different scales, enhancing its ability to identify both small and large objects within an image.

YOLOv4 marked another significant leap, introducing a bag of "freebies" and a bag of "specials" to optimize training and inference. The architecture incorporated CSPDarknet53 as its backbone, which leverages cross-stage partial connections to reduce computational costs and improve feature representation. Additionally, YOLOv4 employed techniques like Mish activation and mosaic data augmentation to enhance performance [4].

YOLOv5, unlike its predecessors, was implemented in PyTorch rather than Darknet. While the core principles remained similar, this shift facilitated easier experimentation and integration with other PyTorch-based tools. The architecture allowed for scaling the model size (n, s, m, l, x) to fit different computational constraints.

Following YOLOv5, the research community continued to build upon the YOLO framework, resulting in versions like YOLOv6, YOLOv7, and YOLOv8. These later iterations have focused on further improvements in speed, accuracy, and efficiency, often

incorporating techniques such as knowledge distillation, improved loss functions, and optimized network architectures.

2.3. Adaptations and Modifications for Specific Applications

YOLO's adaptability has spurred numerous application-specific modifications. Drone imagery, often plagued by varying illumination and small object sizes, benefits from custom loss functions emphasizing localization accuracy and architectural tweaks like incorporating attention mechanisms. In medical diagnosis, where subtle anomalies require precise detection under non-uniform lighting from medical imaging equipment, modified YOLO architectures coupled with specialized data augmentation techniques and loss functions that account for class imbalance are common. Underwater object detection presents unique challenges due to light absorption and scattering, leading to blurry images and color distortion. Adaptations here involve pre-processing techniques to enhance image quality, modified backbones to extract robust features, and custom loss functions that penalize errors arising from poor visibility and uneven illumination, often incorporating depth information if available to improve detection accuracy by minimizing the impact of illumination variation on feature extraction [5].

3. Impact of Complex Illumination on YOLO Performance

3.1. Underexposure and Overexposure

Underexposure and overexposure represent two extremes of illumination that significantly degrade the performance of YOLO object detection models. Underexposure, characterized by insufficient light, leads to a reduction in image contrast and a loss of detail, particularly in darker regions. This diminished contrast makes it difficult for YOLO's feature extraction layers to identify salient features, such as edges and textures, which are crucial for accurate object localization and classification. The pixel values tend to cluster towards the lower end of the intensity range, resulting in a compressed histogram and a reduced dynamic range. Consequently, subtle differences between objects and their backgrounds become obscured, hindering the model's ability to differentiate between them. As shown in Table 3, the feature maps generated from underexposed images often lack the distinct patterns necessary for effective object detection, leading to degraded performance.

Table 3. Performance degradation of YOLO under varying levels of underexposure.

Level of Underexposure	Impact on YOLO Performance	Underlying Cause
Slight Underexposure	Reduced detection accuracy, especially for objects in darker areas.	Reduced image contrast, making it harder to differentiate objects from the background. Feature extraction layers struggle to identify subtle edges and textures.
Moderate Underexposure	Significant decrease in detection accuracy and increase in false negatives.	Pixel values cluster towards the lower end of the intensity range, compressing the histogram and reducing dynamic range. Subtle differences are obscured.
Severe Underexposure	YOLO model struggles to detect any objects, leading to very poor performance.	Extreme lack of image detail. Feature maps lack distinct patterns, making object detection nearly impossible. Gradient information becomes unreliable.
Dynamic Range Compression	Reduced ability to differentiate objects.	Diminished contrast → decreased detection

Conversely, overexposure occurs when excessive light saturates the pixel values, particularly in brighter regions of the image. This saturation results in a loss of information as pixel values are clipped at the maximum intensity level. Details in these overexposed areas are effectively washed out, making it impossible for YOLO to extract meaningful features. The histogram shifts towards the higher end, leading to a similar compression of the dynamic range as observed in underexposure. Furthermore, overexposure can introduce artifacts, such as blooming, which further distort the image and interfere with feature extraction. The gradient information, which is vital for training deep learning models, becomes unreliable in saturated regions, potentially leading to inaccurate bounding box predictions and misclassifications. The impact of overexposure is particularly detrimental when objects of interest are located in brightly lit areas, as their features become indistinguishable from the background. Both underexposure and overexposure negatively impact the quality of the input data, leading to a degradation in the performance of YOLO models. The severity of the impact depends on the degree of underexposure or overexposure and the specific characteristics of the objects being detected [6].

3.2. Shadows and Non-Uniform Illumination

Shadows and non-uniform illumination represent significant challenges for object detection algorithms like YOLO. These conditions introduce variations in pixel intensity and contrast, potentially obscuring object features and leading to both false negatives (missed detections) and inaccurate bounding box predictions [7].

The presence of shadows can drastically alter the appearance of an object. As shown in Table 4, the effect of varying shadow intensity on YOLO's detection accuracy is significant. A dark shadow cast across a portion of an object can reduce the local contrast, making it difficult for YOLO to differentiate the object from the background. This is particularly problematic when the shadowed region contains key features used for object classification. For instance, if a shadow obscures the wheels of a car, YOLO might fail to recognize it as a car, resulting in a missed detection. The severity of this effect is often correlated with the shadow's intensity and the size of the shadowed area relative to the object's overall size.

Table 4. Effect of varying shadow intensity on YOLO's detection accuracy.

Shadow Intensity (δ)	Potential Impact on YOLO Detection Accuracy
Low	Minimal impact, particularly if the shadowed area is small and doesn't obscure key features.
Moderate	Increased risk of missed detections (false negatives) if the shadow covers significant portions of the object, especially key features. Bounding box accuracy may be slightly affected.
High	Significant reduction in detection accuracy. High probability of false negatives and inaccurate bounding box predictions due to obscured features and distorted object appearance. Increased difficulty in differentiating the object from the background.

Non-uniform illumination, where different parts of an image are lit with varying intensities, poses a similar challenge. This can occur due to natural lighting variations, artificial light sources, or reflections. In such scenarios, the feature maps generated by YOLO's convolutional layers may become inconsistent, hindering the network's ability to learn robust object representations. Objects in dimly lit areas might appear fainter and less distinct, while those in brightly lit areas might suffer from overexposure, both leading to decreased detection accuracy [8].

Furthermore, inaccurate bounding box predictions can arise from these illumination complexities. YOLO relies on precise localization of object boundaries. However, shadows and non-uniform lighting can distort the perceived edges of an object, causing the predicted bounding box to be misaligned with the actual object. For example, a shadow might create an artificial edge that YOLO interprets as the object's boundary, leading to an undersized or shifted bounding box. The Intersection over Union (*IoU*) between the predicted and ground truth bounding boxes is thus reduced, impacting the overall performance metrics. The impact is also related to the object size s , the illumination variance σ , and the shadow intensity δ .

3.3. Varying Light Sources and Color Distortion

Varying light sources present a significant challenge to object detection algorithms like YOLO, primarily due to the color distortions they introduce. Different light sources, such as fluorescent, incandescent, and LED, emit light with distinct spectral power distributions. This variation in spectral content directly affects the perceived color of objects, as the reflected light reaching the camera sensor is altered. For instance, incandescent light typically has a warmer color temperature, resulting in a reddish or yellowish cast, while fluorescent light often exhibits a cooler, greenish hue.

These color distortions can negatively impact YOLO's performance because the model is typically trained on datasets with a specific distribution of lighting conditions. When presented with images captured under significantly different lighting, the learned color features may no longer be reliable indicators of object presence. The color channels (R , G , B) may shift in unexpected ways, leading to misclassification or reduced detection accuracy [9].

Maintaining color constancy, the ability to perceive colors as consistent despite changes in illumination, is crucial for robust object detection. However, achieving this in real-world scenarios is difficult. YOLO, in its standard form, does not explicitly incorporate mechanisms to compensate for varying light sources and color distortions. While data augmentation techniques like random color jittering can partially mitigate this issue during training, they may not fully capture the complexity of real-world illumination changes. The degree of performance degradation depends on the magnitude of the color shift and the sensitivity of the specific YOLO architecture to color variations. Further research is needed to develop more robust methods for handling complex illumination conditions and ensuring reliable object detection across diverse lighting environments.

4. Techniques for Enhancing YOLO Performance under Complex Illumination

4.1. Image Enhancement Techniques

Image enhancement techniques play a crucial role as pre-processing steps to mitigate the adverse effects of complex illumination on object detection performance, particularly for algorithms like YOLO. These techniques aim to improve image quality by manipulating pixel intensities, thereby enhancing the visibility of objects and features that might otherwise be obscured by poor lighting conditions. Several methods have proven effective in this regard, including histogram equalization, gamma correction, and contrast stretching [10].

Histogram equalization seeks to redistribute pixel intensities to utilize the full dynamic range of the image. This is particularly useful in scenarios where the image histogram is concentrated in a narrow range, leading to low contrast. By transforming the intensity values based on the cumulative distribution function, histogram equalization aims to create a more uniform distribution, thereby enhancing contrast and revealing details in both dark and bright regions. However, it can sometimes over-enhance noise and introduce artificial artifacts, especially in relatively uniform regions.

Gamma correction, on the other hand, adjusts the overall brightness of an image by applying a power-law transformation to the pixel intensities. The transformation is defined as $I_{out} = I_{in}^\gamma$, where I_{in} is the input intensity, I_{out} is the output intensity, and γ is the gamma value. Values of $\gamma < 1$ brighten the image, while values of $\gamma > 1$ darken it. Gamma correction is particularly effective in correcting images that appear too dark or too bright due to incorrect camera settings or uneven lighting. It offers a more controlled approach to brightness adjustment compared to histogram equalization, allowing for fine-tuning of the image's overall appearance.

Contrast stretching, also known as normalization, linearly scales the pixel intensities to span the full available range. This technique is effective when the image has low contrast due to a limited range of intensity values. The transformation maps the minimum intensity value to 0 and the maximum intensity value to the maximum representable value (e.g., 255 for an 8-bit image). Contrast stretching can significantly improve the visibility of objects by increasing the difference between their intensities and the background. However, it is sensitive to outliers, as a single very bright or very dark pixel can compress the rest of the intensity range.

The effectiveness of each technique depends on the specific illumination conditions and the characteristics of the image. For instance, histogram equalization might be more suitable for images with globally poor contrast, while gamma correction might be preferred for images with overall brightness issues. Contrast stretching can be effective when the intensity range is limited, but it may not be ideal for images with significant variations in illumination across different regions. Therefore, selecting the appropriate image enhancement technique or combination of techniques is crucial for optimizing YOLO's performance under complex illumination conditions [11].

4.2. Adaptive Thresholding and Region-Based Methods

Adaptive thresholding techniques offer a potential solution to the challenges posed by non-uniform illumination in object detection tasks. Unlike global thresholding, which applies a single threshold value to the entire image, adaptive methods dynamically calculate a threshold for each pixel or region based on the local intensity distribution. This adaptability is particularly beneficial when dealing with shadows, highlights, or gradual changes in lighting across the image.

One widely used adaptive thresholding algorithm is Otsu's method. While technically a global thresholding method, its automatic threshold selection based on minimizing intra-class variance makes it robust to bimodal intensity distributions often encountered in unevenly lit scenes. The algorithm calculates a threshold value that separates the foreground and background pixels by maximizing the between-class variance. However, Otsu's method can struggle when the foreground and background pixel counts are significantly imbalanced.

Adaptive Gaussian thresholding represents a more localized approach. This method calculates the threshold for each pixel as the weighted mean of its neighboring pixels, where the weights are determined by a Gaussian distribution. By subtracting a constant C from this weighted mean, the algorithm can further fine-tune the segmentation. The size of the neighborhood window significantly impacts the performance; a smaller window is more sensitive to noise, while a larger window may fail to adapt to rapid illumination changes. The parameter C allows for adjusting the sensitivity to local contrast [12].

Region-based methods, such as region growing and watershed segmentation, offer alternative approaches to object segmentation under complex illumination. Region growing starts with a set of seed pixels and iteratively adds neighboring pixels that meet certain homogeneity criteria, such as similar intensity or color. Watershed segmentation, inspired by topographic analysis, treats the image as a landscape and identifies catchment

basins corresponding to different objects. These methods can be effective in delineating object boundaries even when illumination varies significantly within the object itself.

The impact of these techniques on YOLO's detection accuracy hinges on their ability to effectively pre-process the input images. Improved segmentation leads to more accurate feature extraction, which in turn enhances YOLO's ability to correctly classify and localize objects. However, the computational cost of adaptive thresholding and region-based methods must be considered, as they can increase the overall processing time. Furthermore, the optimal choice of algorithm and parameter settings depends heavily on the specific characteristics of the illumination conditions and the objects being detected. Experiments are needed to determine the most effective combination of techniques for a given scenario.

4.3. Robust Feature Extraction Strategies

Complex illumination conditions pose a significant challenge to object detection algorithms like YOLO, as they can drastically alter the appearance of objects, leading to feature distortion and reduced detection accuracy. To mitigate these effects, robust feature extraction strategies are crucial. These strategies aim to extract features that are invariant or less sensitive to changes in illumination, thereby improving the overall performance of YOLO under such challenging conditions.

One such technique is the Scale-Invariant Feature Transform (SIFT). SIFT detects and describes local features in images that are invariant to scale, rotation, and changes in illumination. It involves several stages, including scale-space extrema detection, keypoint localization, orientation assignment, and feature descriptor generation. By extracting features that are less affected by illumination variations, SIFT can provide YOLO with more reliable information for object recognition [13].

Speeded-Up Robust Features (SURF) is another robust feature extraction method. SURF is similar to SIFT but is computationally more efficient. It uses integral images to approximate the Hessian matrix determinant, enabling faster feature detection and description. Like SIFT, SURF is also invariant to scale and rotation, and it is relatively robust to illumination changes.

Local Binary Patterns (LBP) offer a different approach to feature extraction. LBP is a simple yet effective texture operator that summarizes the local spatial structure of an image. It works by comparing the intensity of a central pixel with the intensities of its surrounding pixels. The resulting binary patterns are then used to create a histogram, which represents the texture of the local region. LBP is particularly robust to monotonic illumination changes, making it a valuable tool for object detection under varying lighting conditions.

Integrating these robust features with YOLO can be achieved in several ways. One approach is to use the extracted features as additional input channels to the YOLO network. For example, the LBP histogram for each grid cell could be concatenated with the existing feature maps. Another approach is to use these features to pre-train or fine-tune the YOLO network, allowing it to learn more robust representations of objects. The optimal integration strategy depends on the specific application and the characteristics of the illumination conditions.

5. Comparison with Other Object Detection Methods

5.1. Comparative Analysis of YOLO and Faster R-CNN

YOLO and Faster R-CNN represent two distinct approaches to object detection, each exhibiting unique strengths and weaknesses under complex illumination conditions. Faster R-CNN, with its two-stage architecture involving a Region Proposal Network (RPN) and subsequent classification, generally demonstrates higher accuracy than YOLO, particularly in scenarios with varying light intensity and shadows. The RPN allows Faster R-CNN to effectively identify regions of interest even when object features are partially

obscured by poor lighting. However, this accuracy comes at the cost of speed. The two-stage process makes Faster R-CNN significantly slower than YOLO, hindering its applicability in real-time applications.

YOLO, on the other hand, prioritizes speed through its single-stage detection mechanism. It processes the entire image in one pass, making it substantially faster than Faster R-CNN. While this speed is advantageous, YOLO's accuracy often suffers under complex illumination. Its single-stage approach struggles to effectively differentiate objects from the background when lighting conditions are non-uniform, leading to increased false positives and missed detections. The performance degradation is particularly noticeable with small objects or objects exhibiting low contrast due to shadows.

Mask R-CNN, an extension of Faster R-CNN, further complicates the comparison. While primarily designed for instance segmentation, its object detection capabilities are comparable to Faster R-CNN. Mask R-CNN's instance segmentation provides more detailed information, potentially aiding in disambiguation under challenging illumination. However, this added complexity further reduces its speed, making it the slowest of the three. The trade-off between detailed information and processing speed becomes a critical consideration when choosing an object detection method for environments with complex illumination. The choice depends heavily on the specific application requirements and the relative importance of speed, accuracy, and detailed scene understanding.

5.2. SSD and Other Single-Stage Detectors

Single Shot Multibox Detector (SSD) represents a significant advancement in single-stage object detection, prioritizing speed while maintaining reasonable accuracy. Its performance under complex illumination conditions, however, reveals certain limitations. SSD's reliance on a fixed set of anchor boxes across multiple feature maps makes it susceptible to variations in lighting that can alter object appearance. Specifically, extreme brightness or darkness can lead to feature distortion, causing mismatches between the learned anchor box representations and the actual objects. This often results in decreased detection accuracy, particularly for smaller objects where subtle illumination changes have a more pronounced effect.

Other single-stage detectors, such as RetinaNet, address some of SSD's shortcomings through techniques like Focal Loss, which mitigates the impact of class imbalance and allows the network to focus on hard-to-classify examples, including those affected by poor illumination. RetinaNet demonstrates improved robustness compared to SSD in challenging lighting scenarios, but it often comes at the cost of increased computational complexity. The trade-off between speed and accuracy is a crucial consideration when selecting a single-stage detector for deployment in environments with fluctuating illumination. While SSD offers faster inference times, its accuracy can degrade significantly under adverse lighting. RetinaNet, and similar detectors, provide a more accurate alternative, but require more computational resources, impacting the frames per second (*FPS*) and overall system latency. The choice depends on the specific application requirements and the acceptable balance between *FPS* and mean Average Precision (*mAP*).

6. Challenges and Future Perspectives

6.1. Remaining Challenges

Despite the advancements in YOLO-based object detection, significant challenges remain in achieving robust performance under complex illumination conditions. A primary concern is the limited generalizability to unseen illumination scenarios. While YOLO models can be trained on datasets with diverse lighting, their performance often degrades when faced with entirely novel illumination patterns, such as extreme

backlighting, intense shadows, or rapidly changing light levels. This sensitivity stems from the reliance on learned features that may not be invariant to drastic illumination variations. Developing techniques to enhance illumination invariance, perhaps through data augmentation strategies that simulate a wider range of lighting conditions or by incorporating illumination-invariant feature extraction methods, is crucial.

Another persistent challenge is the computational cost associated with processing images under complex illumination. Pre-processing steps aimed at mitigating the effects of adverse lighting, such as histogram equalization or gamma correction, can add significant overhead. Furthermore, more sophisticated models designed to handle illumination variations often require increased computational resources, potentially hindering real-time applications. Balancing the need for accuracy with the constraints of computational efficiency is a key area for future research. Exploring lightweight architectures or optimization techniques that minimize the computational burden of illumination-aware processing is essential for deploying YOLO models in resource-constrained environments. The trade-off between model complexity, accuracy under varying illumination, and computational speed needs careful consideration, especially for applications involving embedded systems or real-time video analysis.

6.2. Future Research Directions

Future research should prioritize addressing the identified limitations of YOLO under complex illumination conditions. One promising avenue is the development of illumination-aware loss functions. Current loss functions typically treat all pixels equally, neglecting the impact of varying illumination on object features. An illumination-aware loss function could incorporate a weighting mechanism that emphasizes regions less affected by shadows or highlights, potentially improving detection accuracy. This could involve using image statistics, such as local variance or entropy, to modulate the loss contribution of each pixel. For example, a weight w_i could be assigned to each pixel i based on its illumination characteristics, influencing the overall loss calculation.

Another direction involves integrating attention mechanisms into the YOLO architecture. Attention mechanisms can selectively focus on the most relevant features in an image, effectively filtering out noise and irrelevant information caused by poor illumination. Spatial attention modules could highlight regions containing objects of interest, while channel attention modules could emphasize feature maps that are robust to illumination changes. The integration of these mechanisms could allow the network to learn more discriminative features, leading to improved performance.

Furthermore, generative adversarial networks (GANs) offer a powerful tool for data augmentation. By training a GAN to generate synthetic images with diverse illumination conditions, we can significantly expand the training dataset and improve the robustness of YOLO models. The GAN could be trained to transform existing images into new images with varying degrees of shadow, highlight, and color distortion, effectively simulating the challenges encountered in real-world scenarios.

Finally, domain adaptation techniques hold significant potential. These techniques aim to transfer knowledge learned from a source domain (e.g., images with controlled illumination) to a target domain (e.g., images with complex illumination). This could involve fine-tuning a YOLO model pre-trained on a large dataset of well-lit images using a smaller dataset of images captured under challenging illumination conditions. Alternatively, adversarial domain adaptation methods could be employed to learn domain-invariant features, enabling the model to generalize better to unseen illumination conditions. The goal is to minimize the domain discrepancy, often measured by a distance metric $d(D_s, D_t)$ between the source domain D_s and the target domain D_t .

7. Conclusion

7.1. Summary of Findings

This review has systematically examined the performance of YOLO-based object detection models under complex illumination conditions, revealing significant vulnerabilities. Our analysis demonstrates that variations in illumination, including underexposure, overexposure, and non-uniform lighting, consistently degrade YOLO's detection accuracy, leading to increased false negatives and reduced Intersection over Union (IoU) scores. Specifically, the feature extraction process, heavily reliant on convolutional layers, struggles to adapt to drastic changes in pixel intensity and contrast caused by poor lighting.

Furthermore, we evaluated the effectiveness of several mitigation techniques. Data augmentation strategies, such as random brightness and contrast adjustments, showed promise in improving robustness, particularly when applied during the training phase. Image enhancement techniques, including histogram equalization and gamma correction, demonstrated the ability to pre-process images and normalize illumination, leading to improved detection rates. However, the optimal parameter settings for these techniques are highly dependent on the specific characteristics of the illumination challenges present in the dataset. Finally, the incorporation of attention mechanisms and normalization layers within the YOLO architecture itself proved beneficial in selectively focusing on relevant features and reducing the impact of illumination variations on feature representation, contributing to more stable and accurate object detection performance under diverse lighting conditions. The impact of illumination can be quantified by the change in the average precision *AP* and average recall *AR* values.

7.2. Concluding Remarks

Addressing illumination challenges is paramount for robust object detection, particularly in real-world scenarios where lighting conditions are rarely ideal. Our analysis highlights the significant impact of varying illumination on YOLO's performance, emphasizing the need for specialized techniques to mitigate these effects. Future research should focus on developing illumination-invariant features and adaptive learning strategies that can dynamically adjust to changing lighting. Exploring the integration of advanced image enhancement techniques and domain adaptation methods holds considerable promise for improving object detection accuracy and reliability under complex illumination conditions, ultimately leading to more dependable vision systems.

References

1. A. A. Murat, and M. S. Kiran, "A comprehensive review on YOLO versions for object detection," *Engineering Science and Technology, an International Journal*, vol. 70, p. 102161, 2025. doi: 10.1016/j.jestch.2025.102161
2. C. Gheorghe, M. Duguleana, R. Boboc, and C. Postelnicu, "Analyzing real-time object detection with YOLO algorithm in automotive applications: A review," *Computer Modeling in Engineering & Sciences*, vol. 141, no. 3, p. 1939, 2024.
3. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788. doi: 10.1109/cvpr.2016.91
4. A. Tripathi, M. K. Gupta, C. Srivastava, P. Dixit, and S. K. Pandey, "Object detection using YOLO: A survey," In *Proceedings of the 2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022, pp. 747-752. doi: 10.1109/ic3i56241.2022.10073281
5. M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using YOLO," *IEEE Access*, vol. 8, pp. 125459-125476, 2020.
6. P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066-1073, 2022. doi: 10.1016/j.procs.2022.01.135
7. W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-adaptive YOLO for object detection in adverse weather conditions," In *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2), 1792-1800., 2022.
8. A. Vijayakumar, and S. Vairavasundaram, "YOLO-based object detection models: A review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83535-83574, 2024. doi: 10.1007/s11042-024-18872-y

9. L. T. Ramos, and A. D. Sappa, "A decade of You Only Look Once (YOLO) for object detection: A review," *IEEE Access*, vol. 13, pp. 192747-192794, 2025.
10. W. T. Weng, and X. Niu, "Enhancing UAV object detection in low light conditions with ELS YOLO: A lightweight model based on improved YOLOv11," *Sensors*, vol. 25, no. 14, p. 4463, 2025.
11. S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "YOLO-FIRI: Improved YOLOv5 for infrared image object detection," *IEEE Access*, vol. 9, pp. 141861-141875, 2021. doi: 10.1109/access.2021.3120870
12. Z. Hua, K. Aranganadin, C. C. Yeh, X. Hai, C. Y. Huang, and T. C. Leung, "A benchmark review of YOLO algorithm developments for object detection," *IEEE Access*.
13. M. L. Ali, and Z. Zhang, "The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection," *Computers*, vol. 13, no. 12, p. 336, 2024. doi: 10.3390/computers13120336

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 GBP and/or the editor(s). GBP 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.