An Integrated LIME Image Enhancement and YOLOv8 framework for Object Detection in

Low-Light Conditions

Muhammad Ilham Perdana1,b), Putri Nayla Sabri1,c), Diah Risqiwati1,a)

1Department of Informatics, Universitas Muhammadiyah Malang, Malang, Indonesia

a)Corresponding author: [risqiwati@umm.ac.id](mailto:yos.sumantri@upnyk.ac.id)

b)ilhamperdana@umm.ac.id

c) putrinaylasabri@webmail.umm.ac.id

**Abstract.**.Low-light environments pose significant challenges to object detection systems due to poor visibility and degraded image quality. These factors often lead to a significant drop in detection accuracy, limiting the effectiveness of conventional object detection models when applied directly to raw low-light images. To address this issue, this study proposes a framework that integrates the low-light Image Enhancement (LIME) algorithm with the YOLOv8 object detection model. By enhancing visual quality prior to detection, the proposed approach significantly improves performance under challenging illumination conditions. Experiments conducted on the ExDark dataset show that the method delivers strong detection accuracy surpassing several previous low-light object detection approaches. The enhanced images allow the detector to better identify object features that are otherwise obscured in dark conditions. These findings highlight the effectiveness and practicality of incorporating image enhancement techniques like LIME as a preprocessing step to achieve more reliable and precise object detection in adverse lighting conditions.

**Keywords:** Low-Light Image Enhancement, LIME Image Enhancement, YOLOv8, ExDark Dataset

# INTRODUCTION

Object detection in images with low-light conditions faces various unique challenges compared to normal lighting conditions. The lack of reflected light causes dark areas and high noise levels, resulting in a drastic decrease in image quality (1). Images often have poor illumination, low contrast, weak reflectance, and faded or even completely lost object details (2). In extreme situations, color distortion and detail loss further deteriorate object visibility. This condition directly affects the performance of object detection algorithms, including deep learning-based models that generally work optimally under normal illumination, such as Faster R-CNN or SSD, but exhibit significant performance degradation on low-light images (1). Most object detectors are designed for data with sufficient illumination, so few have specific optimizations for low-light environments.

One approach to address this problem is to use image enhancement techniques as a pre-processing step (3,4). Image enhancement aims to improve brightness, increase contrast, reduce noise, and sharpen object edges so that important features can be better identified (2). The advantage of this method lies in its ability to overcome the limitations of expensive hardware by utilizing relatively simple and efficient software solutions. However, overly aggressive enhancement can lead to over-enhancement effects, where bright areas become excessively bright, causing important details to be lost (5,6).

There are two general approaches to image enhancement algorithms, deep learning-based (7) and traditional-based(8). Deep learning-based enhancement methods often require large-scale datasets, extensive training time, and more computational resources, which dan slow down the detection process. In contrast, traditional enhancement techniques are lightweight and better suited for applications that demand high inference speed. Since this research also prioritizes real-time detection capability, low-latency methods are preferred to maintain a high frame rate without sacrificing too much accuracy. For this reason, the present study focuses on implementing traditional image enhancement and integrating it with object detection algorithms to detect objects in low-light conditions.

.

The LIME (9) image enhancement algorithm was selected to perform enhancement on low-light images. LIME is recognized for its ability to improve image quality under low-light conditions through the illumination map estimation. YOLOv8 was chosen as the detection model due to its advantages in speed, accuracy, and implementation flexibility (1). Compared to its previous versions, YOLOv8 adopts an anchor-free detection approach and a lighter architecture, with multi-scale detection capabilities that are highly relevant for low-light conditions. This model has also proven to be competitive in detecting objects of various sizes in real time, making it ideal for testing the impact of image enhancement on object detection performance (1).

The objective of this study is to investigate the effectiveness of combining the LIME image enhancement algorithm with the YOLOv8 object detection algorithm for object detection in low-light environments. The LIME algorithm is applied to improve the visibility and quality of low-light images through illumination map estimation, while YOLOv8 serves as the detection backbone due to its accuracy and real-time performance. By integrating these two approaches, the study aims to evaluate whether pre-processing images with LIME can enhance the detection accuracy of YOLOv8 in challenging low-light scenarios.

# LITERATURE REVIEW

## Image Enhancement in Digital Image Processing

Image enhancement is an important process in digital image processing (3,4), especially to address issues of image quality caused by low illumination, noise, or poor contrast. Traditional image enhancement methods are widely used due to their simplicity, speed, and computational efficiency, making them suitable to be applied as a pre-processing step in recognition or object detection systems (5).

Some commonly used classical methods include:

### Histogram Equalization (HE)

This is a contrast enhancement technique that works by flattening the pixel intensity distribution of the image so that its histogram becomes more uniform. Thus, HE can improve the visibility of details in areas with low illumination or low contrast (1). However, HE also has limitations, such as a tendency to introduce noise or over-enhancement effects (10), especially in parts of the image that are already sufficiently bright, which can cause loss of important details (11). Mathematically, HE maps an input pixel intensity ​ to an output intensity ​ ​ using the cumulative distribution function (CDF) of the image histogram as follows Eq. 1

1

where *​*) ​ is the probability of intensity level ​ occurring in the image.

### Contrast Limited Adaptive Histogram Equalization (CLAHE)

As an improvement over HE, CLAHE performs contrast enhancement locally on sub-regions of the image and limits noise amplification by controlling the maximum allowed contrast value (6,10,12,13). This makes CLAHE more effective in enhancing details in images with uneven illumination and reduces artifacts such as halo effects commonly seen in traditional HE algorithm (1,6). CLAHE is also known to be computationally lightweight (13), thus it is widely applied in various fields of image processing, including medical and surveillance. Mathematically, CLAHE modifies the histogram of each local region by clipping it at a predefined limit to prevent over-amplification of contrast. The clipped histogram for intensity level can be expressed as Eq 2.

2

Where denotes the original histogram frequency for intensity level . After clipping, the excess pixel counts are redistributed equally among all intensity levels, ensuring that the overall histogram remains balanced while avoiding excessive contrast in bright regions. This process helps maintain local detail without introducing significant noise.

### Gamma Correction (GC)

This technique performs a non-linear transformation on pixel intensity values based on a gamma parameter (14). Mathematically, the output pixel intensity ​ is computed from the input intensity ​ using the power-law equation:

3

where is a scaling constant (often set to 1), ​ and ​ are normalized intensity values in the range [0,1], and controls the degree of brightness adjustment. Values of < 1 will brighten the image, while > 1 will darken it. Gamma correction is very useful for adjusting brightness and contrast, especially to brighten dark areas without making bright areas excessively bright (overexposed) (12). However, the choice of value must be done carefully as an incorrect value can cause color distortion or loss of details in certain areas (10).

These classical methods are often chosen due to their ability to quickly improve image quality using software, without the need for expensive specialized hardware. Nevertheless, the use of traditional methods must be adjusted according to the characteristics of the image and the application purpose, because overly aggressive enhancement can produce artifacts or loss of important details that impact the performance of object detection systems. In addition to these classical techniques, this study also explores the LIME (Low-Light Image Enhancement via illumination Map Estimation) (9). Unlike HE, CLAHE, or GC, the LIME algorithm estimates the illumination map of an image and adjust brightness adaptively based on local illumination levels. This approach enables more natural enhancement and better preservation of details in both dark and bright regions. The implementation and integration of LIME with YOLOv8 will be discussed in the following section.

## The Evolution of YOLO

You Only Look Once (YOLO) is one of the most popular object detection architectures that has continuously evolved since it was first introduced by Redmon et al. in 2016 (15). The main advantage of YOLO is its ability to perform real-time object detection with competitive accuracy by combining detection and classification in a single efficient network (6). Over time, the YOLO architecture has undergone multiple improvements to enhance accuracy, speed, and the ability to detect objects at various scales, including in challenging lighting conditions. The following **Table 1** summarizes the key features and differences among YOLO versions. The data presented in this table are derived from the studies in (15,16).

**TABLE 1.** Comparison of YOLO Versions

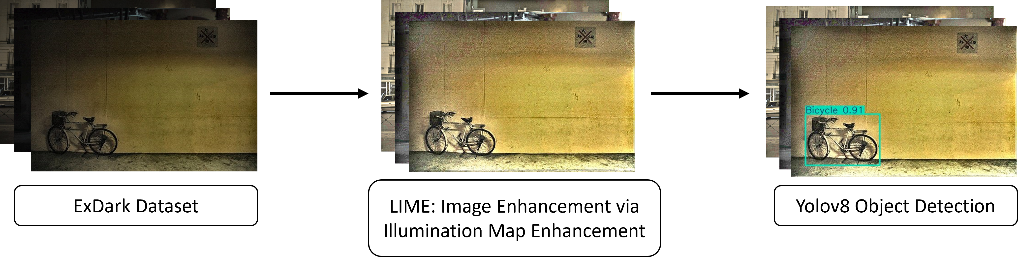
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **YOLO Version** | **Release Year** | **Backbone** | **Key Features** | **Strengths** | **Limitations** |
| YOLOv1(17) | 2016 | Custom CNN (24 conv layers + 2 FC), Darknet | Unified detection framework, grid-based prediction, NMS, Leaky ReLU activations | Real-time detection (45 FPS), fast, simple | Large localization errors, low recall, poor small/overlapping object detection |
| YOLOv2(15) | 2017 | Darknet-19 | Anchor boxes, batch normalization, high-res classifier, direct location prediction, multi-scale training | Better accuracy & recall, outperforms SSD and Faster R-CNN | Better accuracy & recall, outperforms SSD and Faster R-CNN |
| YOLOv3(16) | 2018 | Darknet-53 | Feature Pyramid Network (FPN), multi-scale prediction (3 scales), residual blocks | Improved small object detection, strong, twice as fast as ResNet-152 | Moderate difficulty detecting medium/large objects |
| YOLOv4(18) | 2020 | CSPDarknet-53 | Spatial Pyramid Pooling (SPP), Path Aggregation Network (PANet), Bag of Freebies/Specials | Balanced speed & accuracy, 65 FPS, strong real-world performance | Computationally intensive Mish activation, sometimes lower mAP on benchmarks |
| YOLOv5(19) | 2020 | Modified CSP v7 with C2f modules | PyTorch implementation, scalable models (n,s,m,l,x), automated anchor learning, advanced augmentations | Fast & accurate, practical deployment, open source, 200 FPS | Still uses anchor boxes, licensing controversy |
| YOLOv6(20) | 2022 | EfficientRep (RepVGG-based) | Anchor-free approach, decoupled head, task alignment learning, new loss functions (VFL, SIoU, DFL) | Industry focused, faster & more accurate, up to 802 FPS | Newer version, ongoing improvements |
| YOLOv7(21) | 2022 | E-ELAN (Extended Efficient Layer Aggregation Network) | Planned re-parameterized convolution (RepConvN), model scaling, batch norm integration | High accuracy (56.8% mAP), efficient training & inference | Not optimized for CPU high computation for large models |
| YOLOv8(22) | 2023 | Modified CSPDarknet53 + Transformer layers (C2f bottleneck) | Transformer layers, dynamic/anchor-free boxes, scalable (n,s,m,l,x), semantic segmentation (YOLOv8-Seg) | Very fast (280 FPS on NVIDIA A100), excellent multi-scale detection, suitable for low-light | Official paper pending, some ambiguity in anchor vs anchor-free approach |

From the comparison above, it is evident that each successive YOLO version introduces significant architectural and functional enhancements to better balance detection accuracy and computational efficiency (15). Among these, YOLOv8 stands out as the latest iteration incorporating cutting-edge techniques such as a modified CSPDarknet53 backbone, transformer layers, and anchor-free detection mechanisms that collectively boost both speed and accuracy (1). Its flexibility in model sizes and additional segmentation capabilities further broaden its applicability (15).

YOLOv8’s superior performance in real-time detection, especially under low-light and degraded image quality conditions (23), makes it the most suitable choice for this study. Its ability to effectively handle noise, overexposure, and multi-scale object detection ensures robust and efficient operation, aligning perfectly with the objectives of achieving reliable and responsive object detection in challenging environments.

# METHOD

This research proposes a pipeline that integrates Low-light Image Enhancement via Illumination Map Estimation (LIME) (9) with the YOLOv8 object detection framework to improve detection accuracy under poor illumination conditions. LIME (9) was chosen as the enhancement technique primarily due to its computational efficiency and minimal hardware requirements compared to deep learning-based enhancement models, which often demand extensive GPU resources. The proposed method aims to strike a balance between enhancement quality and processing speed, making it suitable for practical low-power or real-time applications. The complete method of our proposed algorithm is illustrated in **Fig. 1**.



**FIGURE 1. Illustration of our method.**

## Dataset Preparation

The experiments in this study use the ExDark (Exclusively Dark) dataset (24), a publicly available benchmark specifically curated for low-light image in various object. The ExDark dataset contains 7,363 images distributed across twelve object categories, such as Bottle, Table, People, Bicycle, Boat, Bus, Car, Cat, Chair, Cup, Dog, Motorbike. These images were captured under diverse low-light conditions such as dim indoor lighting, nighttime urban environments, and natural low-light outdoor scenes. In this study, we split the dataset into three parts, training, validation, and testing, containing 3,000, 1800, and 2563 images respectively based on the annotation labels provided in the ExDark dataset (24).

## LIME Image Enhancement Method

LIME (9) enhances low-light images by estimating and refining an illumination map for each pixel. The method begins by estimating the initial illumination map for every pixel as:

1

where in Eq. 1 is the intensity of the pixel in channel , this equation captures the brightest component per pixel, serve as an estimate of the local lighting conditions.

The initial illumination map often contains noise, which can degrade the quality of enhanced image. To solve this, LIME (9) applies structure-aware smoothing by minimizing a weighted total variation cost function. Weights specifically designed to maintain strong edges while reducing small-scale texture patterns. This process ensures to make object boundaries remain sharp after enhancement and to reduce noise.

After we get our refined illumination , the enhanced image is reconstructed using:

2

where is a constant added to avoid division by zero. Eq. 2 aims to brightens darker regions in proportion to their estimated illumination. Finally, Gamma Correction (GC) is applied with the value of is 0.7. Some of examples of the enhancement result can be seen in **Fig 2**.

**FIGURE 2. Example of Enhanced Image using LIME Algorithm. (a) Raw Images (b) Enhanced Images.**

## YOLOv8 Object Detection

For this study, we use YOLO architecture as a baseline of our method for object detection task. YOLO is one of CNN architecture that already well known for its efficiency and robustness for object detection. It employs a single-stage detection approach, predicting bounding boxes and class probabilities directly from the extracted image features without a separate region proposal step. The medium (YOLOv8m) version was selected to achieve a balance between accuracy and computational efficiency.

During training, images were resized to 640 × 640 pixels while preserving their aspect ratio through padding. The network was trained for 100 epochs with a batch size of 16 using AdamW optimizer with a momentum of 0.9 and an initial learning rate of 0.000625. The loss function combined classification loss (cross-entropy), localization loss (CIoU), and confidence loss to guide model optimization. Data augmentation techniques such as mosaic augmentation and random horizontal flipping were employed to increase generalization capability.

# RESULT & DISCUSSION

This section describes the scenario of our experiment. For our experiment, a PC with GPU GTX 1060 6GB VRAM, Intel i7 processor, and 8 GB RAM was utilized for the computations. We developed our model using Pytorch framework

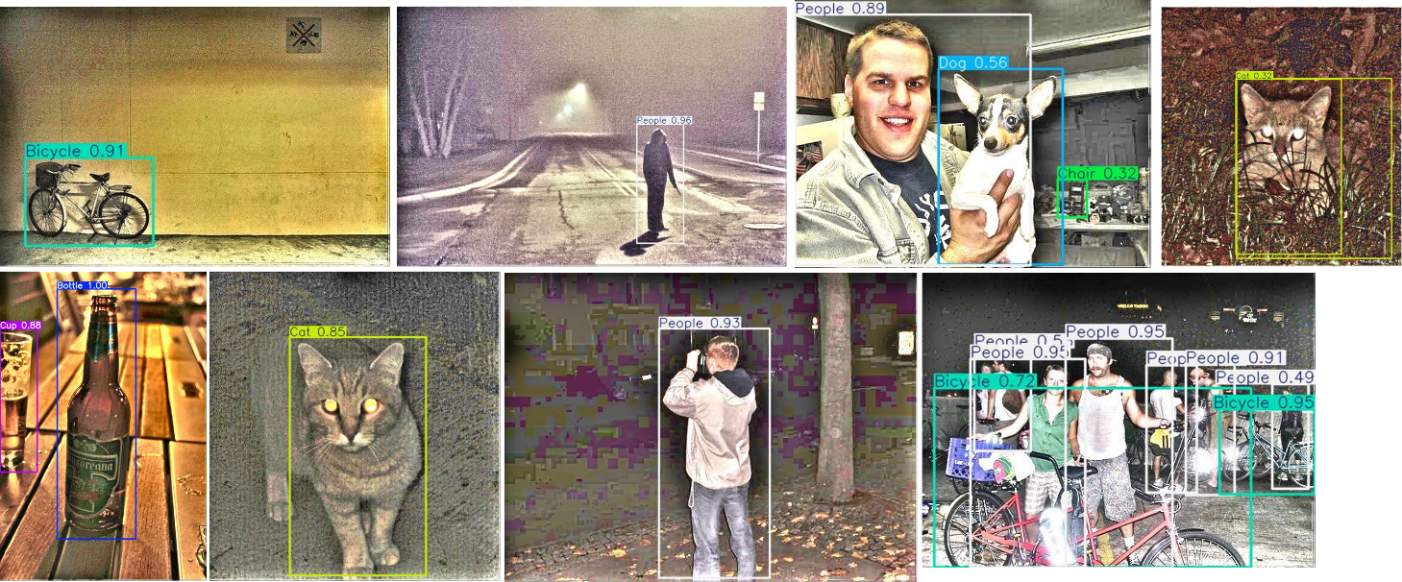
## Training Result and Discussion

Our model was trained for 100 epochs using pretrained weights from the COCO dataset. This approach accelerated convergence and improved detection accuracy. Our model was evaluated on the test dataset to assess its detection performance in low-light conditions. The evaluation employed standard object detection metrics, including precision, recall, mean Average Precision (mAP). The detailed results are presented in **TABLE 2.**

Overall, the model achieved an average precision of 0.733, recall of 0.612, mAP0.5 of 0.691, and mAP0.5:0.95 of 0.450 across all classes. Among individual classes, the highest mAP0.5 was obtained for Bus (0.889), followed by Car (0.781) and People (0.772). Conversely, Table had the lowest performance, with mAP0.5 of 0.469 and mAP0.5:0.95 of 0.298, likely due to its lower feature visibility in low-light conditions. Several examples of detection results can be seen in **Fig 3**.

**TABLE 2**. Per-class Detection Metrics of YOLOv8 with LIME Enhancement on ExDark Test Dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | mAP50 | mAP50-90 |
| All Class | 0.733 | 0.612 | 0.691 | 0.45 |
| Bootle | 0.717 | 0.656 | 0.708 | 0.421 |
| Table | 0.598 | 0.397 | 0.469 | 0.298 |
| People | 0.787 | 0.686 | 0.772 | 0.482 |
| Bicycle | 0.819 | 0.691 | 0.797 | 0.546 |
| Boat | 0.565 | 0.74 | 0.708 | 0.398 |
| Bus | 0.897 | 0.787 | 0.889 | 0.706 |
| Car | 0.885 | 0.608 | 0.781 | 0.534 |
| Cat | 0.652 | 0.573 | 0.606 | 0.372 |
| Chair | 0.724 | 0.564 | 0.625 | 0.4 |
| Cup | 0.728 | 0.413 | 0.596 | 0.396 |
| Dog | 0.753 | 0.682 | 0.741 | 0.475 |
| Motorbike | 0.678 | 0.547 | 0.604 | 0.377 |



### Figure 3. Sample Detection Results on The Test Dataset

## Comparing with Others Model

This section presents a comparative analysis of experimental results obtained from others model. The performance comparison is summarized in **Table 3**.

**Table 3**. Comparison Perfomance with Others Model

|  |  |  |
| --- | --- | --- |
| Model | mAP50 | mAP50-90 |
| Sivasubramanian et al. (25) | - | 0.288 |
| Shovo et al. (26) | 0.5513 | 0.3328 |
| LIME-YOLOv8 (Ours) | **0.691** | **0.45** |

The table presents a comparative performance between the proposed LIME-YOLOv8 model and two existing approaches, Sivasubramanian et al. (23) and Shovo et al. (24). Performance is evaluated using two standard metrics: mean Average Precision at IoU threshold 0.50 (mAP50) and mean Average Precision over IoU thresholds from 0.50 to 0.90 (mAP50-90). Sivasubramanian et al. (23) reported an mAP50-90 value of 0.288, while the mAP50 metric was not provided. Shovo et al. (24) achieved an mAP50 of 0.5513 and an mAP50-90 of 0.3328. The proposed LIME-YOLOv8 model outperformed both models, achieving the highest scores across all reported metrics, with an mAP50 of 0.691 and an mAP50-90 of 0.45. These results demonstrate the superior detection accuracy and robustness of LIME-YOLOv8, particularly at higher IoU thresholds, indicating its effectiveness in precise object localization.

# Conclusion

This study presents the development of a model designed to detect object under low-light environment. Our approach applies LIME as image enhancement and YOLOv8 as object detection algorithm. This approach demonstrate significant improvements in detection performance under challenging illumination conditions. Experimental results on the ExDark dataset shows that our method achieves an mAP50 0.691 and mAP50-90 0.45. Furthermore, we compared our results with several previous studies on low-light object detection. The findings indicate that our approach achieved higher performance across evaluation metrics. This implies that our approach is optimal and capable of accurately and precisely detecting objects under challenging low-light illumination conditions.

As suggestion for future work, deep learning-based image enhancement should be further examined, as it is well known for its effectiveness in improving image quality. However, attention should also be given to its complexity to ensure it does not impose excessive computational resources. Moreover, research should be directed toward designing an end-to-end object detection architecture specificially for low-light object detection.

# REFERENCES

1. Bahi A, Zrigui S, Zrigui M, Nicolas H. Object detection in low light environments. Procedia Comput Sci [Internet]. 2024;246(C):3381–9. Available from: https://doi.org/10.1016/j.procs.2024.09.219

2. Chen W, Shah T. Exploring Low-light Object Detection Techniques. 2021;1–5. Available from: http://arxiv.org/abs/2107.14382

3. Zhang F, Shao Y, Sun Y, Zhu K, Gao C, Sang N. Unsupervised Low-Light Image Enhancement via Histogram Equalization Prior. 2021;14(8):1–13. Available from: http://arxiv.org/abs/2112.01766

4. A review of advancements in low-light image enhancement using deep learning Fangxue Liu. 2020;1–34.

5. Wang W, Yuan X, Chen Z, Wu X, Gao Z. Weak-Light Image Enhancement Method Based on Adaptive Local Gamma Transform and Color Compensation. J Sens. 2021;2021.

6. Chen RC, Dewi C, Zhuang YC, Chen JK. Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models. IEEE Access. 2023;11(September):92926–42.

7. Tian Z, Qu P, Li J, Sun Y, Li G, Liang Z, et al. A Survey of Deep Learning-Based Low-Light Image Enhancement. Vol. 23, Sensors. Multidisciplinary Digital Publishing Institute (MDPI); 2023.

8. Liu W, Zhao P, Song X, Zhang B. Frontiers in Computing and Intelligent Systems A Survey of Low-light Image Enhancement.

9. Guo X, Li Y, Ling H. LIME: Low-Light Image Enhancement via Illumination Map Estimation. IEEE Transactions on Image Processing. 2017 Feb;26(2):982–93.

10. Li C, Tang S, Yan J, Zhou T. Low-light image enhancement via pair of complementary gamma functions by fusion. IEEE Access. 2020;8:169887–96.

11. Alavi M, Kargari M. A new contrast enhancement method for Color dark and low-light images. 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2022. 2022;

12. Retha A, Khayeat H. Object Detection and Recognition in Low Light Image Using Deep Learning Techniques. 2025;4617:305–12.

13. Tianyang Z, Lei Y. Low illumination image enhancement based on improved CLAHE algorithm. J Phys Conf Ser. 2024;2891(11).

14. Rahman T, Khandakar A, Qiblawey Y, Tahir A, Kiranyaz S, Abul Kashem S Bin, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Comput Biol Med [Internet]. 2021;132(March):104319. Available from: https://doi.org/10.1016/j.compbiomed.2021.104319

15. Hussain M. YOLOv1 to v8: Unveiling Each Variant-A Comprehensive Review of YOLO. IEEE Access. 2024;12(March):42816–33.

16. Hussain M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines. 2023;11(7).

17. Terven J, Córdova-Esparza DM, Romero-González JA. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach Learn Knowl Extr. 2023;5(4):1680–716.

18. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020; Available from: http://arxiv.org/abs/2004.10934

19. Khanam R, Hussain M. What is YOLOv5: A deep look into the internal features of the popular object detector. 2024;3–10. Available from: http://arxiv.org/abs/2407.20892

20. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. 2022; Available from: http://arxiv.org/abs/2209.02976

21. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2023;2023-June:7464–75.

22. Yaseen M. What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. 2024;8:1–10. Available from: http://arxiv.org/abs/2409.07813

23. Nuwara Y, Trinh QH. Improving Object Detection Performance on Low-Quality Images Using Histogram Matching and Model Stacking [Internet]. Vol. 15444 LNCS, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) . Springer Nature Switzerland; 2025. 146–155 p. Available from: http://dx.doi.org/10.1007/978-3-031-80507-3\_15

24. Loh YP, Chan CS. Getting to know low-light images with the Exclusively Dark dataset. Computer Vision and Image Understanding. 2019 Jan 1;178:30–42.

25. Sivasubramanian A, Prakash JA, Kumar KSD, Prashanth VR, Sowmya V, Variyar VVS. Object Detection Under Low-Lighting Conditions Using Deep Learning Architectures: A Comparative Study. In: Lecture Notes in Electrical Engineering. Springer Science and Business Media Deutschland GmbH; 2023. p. 269–76.

26. Shovo SUA, Abir MGR, Kabir MM, Mridha MF. Advancing low-light object detection with you only look once models: An empirical study and performance evaluation. Cognitive Computation and Systems. 2024 Dec 1;6(4):119–34.