WHAT IS YOLOv8: AN IN-DEPTH EXPLORATION OF THE INTERNAL FEATURES OF THE NEXT-GENERATION OBJECT DETECTOR

Muhammad Yaseen

Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan;
[*]Correspondence: m.yaseen@nu.edu.pk;

August 29, 2024

ABSTRACT

This study presents a detailed analysis of the YOLOv8 object detection model, focusing on its architecture, training techniques, and performance improvements over previous iterations like YOLOv5. Key innovations, including the CSPNet backbone for enhanced feature extraction, the FPN+PAN neck for superior multi-scale object detection, and the transition to an anchor-free approach, are thoroughly examined. The paper reviews YOLOv8's performance across benchmarks like Microsoft COCO and Roboflow 100, highlighting its high accuracy and real-time capabilities across diverse hardware platforms. Additionally, the study explores YOLOv8's developer-friendly enhancements, such as its unified Python package and CLI, which streamline model training and deployment. Overall, this research positions YOLOv8 as a state-of-the-art solution in the evolving object detection field.

Keywords YOLOv8; Object Detection; Real-Time Image Processing; Computer Vision; Convolutional Neural Networks (CNN)

1 Introduction

Computer vision continues to be a dynamic and rapidly advancing field, enabling machines to interpret and understand visual data [1]. Central to this domain is object detection, a critical task that involves accurately identifying and localizing objects within images or video sequences [2]. Over the years, a range of sophisticated algorithms has been developed to tackle this challenge, with each iteration bringing new advancements and improvements [3].

A significant breakthrough in object detection came with the introduction of the You Only Look Once (YOLO) algorithm by Redmon et al. in 2015 [4]. The YOLO series revolutionized the field by framing object detection as a single regression problem, where a convolutional neural network processes an entire image in one pass to predict bounding boxes and class probabilities [5]. This approach marked a departure from traditional multi-stage detection methods, offering significant gains in speed and efficiency.
Building on the success of its predecessors, YOLOv8 introduces advanced architectural and methodological innovations that significantly enhance its accuracy, efficiency, and usability in real-time object detection [6, 7, 8].

1.1 Survey Objective

The primary objective of this study is to thoroughly evaluate the performance of the YOLOv8 object detection model in comparison to other state-of-the-art detection algorithms. This research will assess the trade-offs between accuracy and inference speed across different versions of YOLOv8 (tiny, small, medium, large) to determine the most suitable model size for various application scenarios.

Key areas of focus include:

1. The impact of the CSPNet backbone and FPN+PAN neck on feature extraction and multi-scale object detection.

2. The benefits of the anchor-free approach in simplifying training and enhancing detection accuracy.
3. The role of YOLOv8's unified Python package and CLI in streamlining model development, training, and deployment.
4. The model's performance on benchmarks such as Microsoft COCO and Roboflow 100, including comparisons with previous YOLO iterations.

Furthermore, the study will examine the developer-centric improvements introduced in YOLOv8, such as its compatibility with Darknet and PyTorch frameworks, and the enhanced user experience offered by its Python API and command-line interface. By providing an in-depth exploration of YOLOv8's innovations and performance, this research seeks to contribute valuable insights to the ongoing development and application of advanced object detection models in the field of computer vision.

2 Evolution of YOLOv8

YOLOv8[9] emerged as the latest evolution in the YOLO series, developed by Ultralytics in 2023. It builds upon the foundation laid by YOLOv5[10], incorporating significant architectural and methodological innovations. YOLOv8 represents a refinement and expansion of the ideas introduced in YOLOv5, with an emphasis on enhancing both model accuracy and usability for real-time object detection tasks[11, 12].

This is the YOLOv8 development timeline:

- **January 10, 2023**: YOLOv8 was officially released, featuring a new anchor-free architecture aimed at simplifying model training and improving detection accuracy across various tasks.
- **February 15, 2023**: Introduction of the YOLOv8 Python package and command-line interface (CLI), streamlining the process of model training, validation, and deployment.
- **March 5, 2023**: Implementation of advanced augmentation techniques such as mosaic and mixup augmentation, enhancing the model's ability to generalize across diverse datasets.
- **April 20, 2023**: Integration of the CSPNet backbone for improved feature extraction and a hybrid FPN+PAN neck, optimizing the model's performance in multi-scale object detection.
- **June 1, 2023**: Support for ONNX and TensorRT formats was added, facilitating deployment on a wider range of hardware platforms, including edge devices.

YOLOv8's release and subsequent updates have significantly influenced the object detection landscape. It is considered a dynamic and evolving model, with ongoing research and development efforts aimed at further enhancing its capabilities. This study delves into the novel techniques and performance metrics introduced in YOLOv8, as detailed in the official Ultralytics documentation and GitHub repository.

3 Architectural Footprint of YOLOv8

YOLOv8 builds upon the strong foundation established by its predecessors in the YOLO family, integrating cutting-edge advancements in neural network design and training methodologies. Similar to earlier versions, YOLOv8 unifies object localization and classification tasks within a single, end-to-end differentiable neural network framework, maintaining the balance between speed and accuracy. The architecture of YOLOv8 is structured around three core components: **Backbone** YOLOv8 employs a sophisticated convolutional neural network (CNN) backbone designed to extract multi-scale features from input images. This backbone, possibly an advanced version of CSPDarknet or another efficient architecture, captures hierarchical feature maps that represent both low-level textures and high-level semantic information crucial for accurate object detection. The backbone is optimized for both speed and accuracy, incorporating depthwise separable convolutions or other efficient layers to minimize computational overhead while retaining representational power.

Neck The neck module in YOLOv8 refines and fuses the multi-scale features extracted by the backbone. It leverages an optimized version of the Path Aggregation Network (PANet), which is enhanced to improve the flow of information across different feature levels. This multi-scale feature integration is critical for detecting objects of varying sizes and scales, and the enhanced PANet design in YOLOv8 likely includes modifications to the original PANet to further optimize memory usage and computational efficiency. **Head** The head module is responsible for generating the final predictions, including bounding box coordinates, object confidence scores, and class labels, from the refined features. YOLOv8 introduces an anchor-free approach to bounding box prediction, moving away from the anchor-based methods

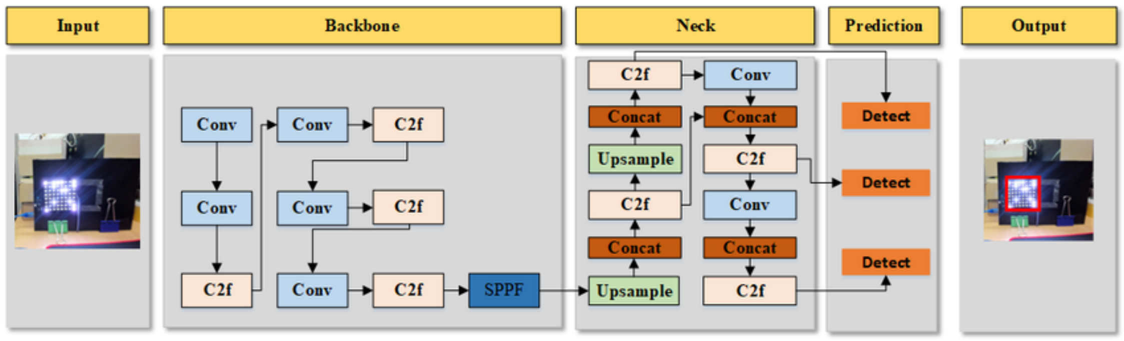


Figure 1: Process of Object Detection [13]

used in earlier YOLO versions. This anchor-free methodology simplifies the prediction process, reduces the number of hyperparameters, and improves the model's adaptability to objects with varying aspect ratios and scales.

By integrating these architectural innovations, YOLOv8 enhances performance in object detection tasks, offering improved accuracy, speed, and flexibility.

3.1 YOLOv8 Training Methodologies

YOLOv8's high performance in object detection is attributed not only to its architectural advancements but also to its sophisticated training methodologies:

3.1.1 Advanced Data Augmentation

YOLOv8 incorporates a suite of new data augmentation strategies that enhance model generalization. Techniques such as improved mosaic augmentation and mixup are employed, where multiple images are combined into a single training example. This process exposes the model to a wider range of object scales, orientations, and spatial configurations, thereby improving its robustness and ability to generalize across different datasets.

3.1.2 Focal Loss Function

YOLOv8 utilizes a focal loss function for classification tasks, which gives more weight to difficult-to-classify examples. This approach mitigates the issue of class imbalance, commonly observed in object detection datasets, and enhances the model's ability to detect small or occluded objects, which are often underrepresented.

3.1.3 Transition to PyTorch with Optimization

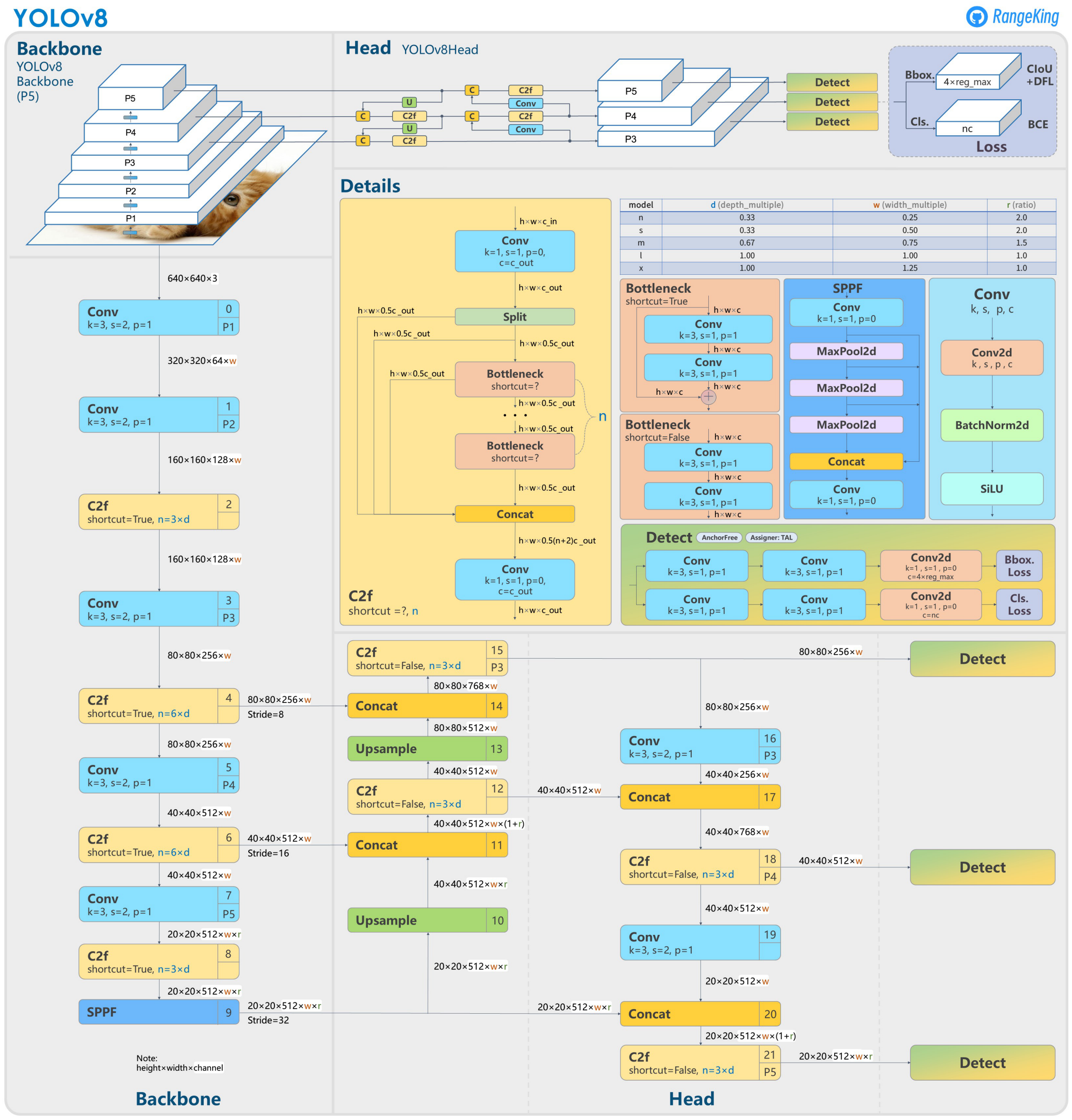
As part of the ongoing transition to PyTorch, YOLOv8 optimizes both its architecture and training processes to leverage modern GPU architectures effectively. By employing mixed-precision training and other computational optimizations, YOLOv8 achieves faster training and inference times while maintaining or even improving accuracy. This optimization ensures that the model is well-suited for deployment in resource-constrained environments.

3.2 Data Augmentation Techniques

In addition to the core training methodologies, YOLOv8 introduces further enhancements in data augmentation:

3.2.1 Mosaic and Mixup Augmentation

This technique combines four or more images into a single training example. By doing so, the model is exposed to a greater variety of object scales, positions, and spatial arrangements, significantly improving its ability to detect small objects and enhancing generalization to unseen data.



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

Figure 2: Model Structure of Yolov8 [14]

3.3 Anchor-Free Bounding Box Prediction

YOLOv8 departs from the anchor-based methods used in earlier YOLO versions, employing an anchor-free approach to bounding box prediction. This innovation reduces computational complexity by eliminating the need for predefined anchor boxes and enhances model efficiency, particularly in detecting objects with varying aspect ratios and scales[15].

3.4 Loss Calculation

YOLOv8's loss function is meticulously designed, comprising three main components:

3.4.1 Focal Loss for Classification

This component addresses class imbalance by assigning greater importance to hard-to-classify examples, improving classification accuracy across all classes.

3.4.2 IoU Loss for Localization

The Intersection over Union (IoU) loss component enhances the accuracy of bounding box predictions, refining the model's ability to precisely localize objects within the image.

3.4.3 Objectness Loss

This loss ensures that the model focuses on regions of the image that are likely to contain objects, thereby improving its overall detection capability.

3.5 Mixed Precision Training

YOLOv8 utilizes mixed precision training, a technique that allows the model to leverage 16-bit floating-point precision during training and inference. This approach significantly accelerates the training process on compatible GPUs, such as NVIDIA's A100 and T4 models, while maintaining the model's accuracy. Mixed precision training also reduces memory consumption, enabling larger batch sizes and more efficient GPU utilization[16].

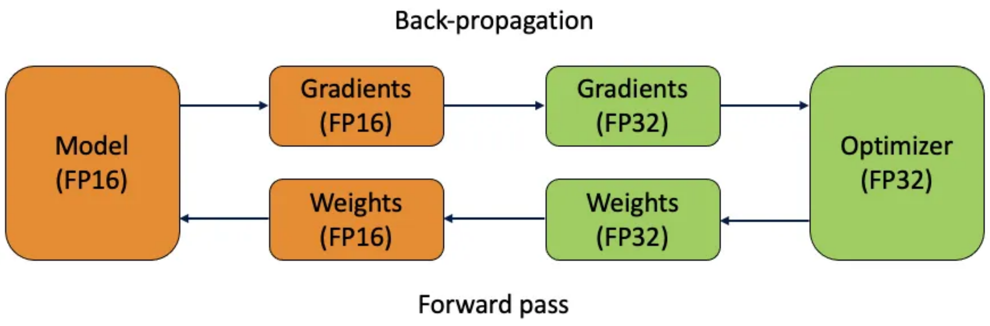


Figure 3: Mixed precision training [17]

3.6 CSP Backbone and Efficient Layer Aggregation

YOLOv8 integrates an advanced version of the CSP (Cross Stage Partial) bottleneck module, which reduces computational redundancy and enhances feature reuse. This architectural choice is complemented by the integration of an improved Feature Pyramid Network (FPN), which aggregates multi-scale features more efficiently, leading to faster inference times and better overall performance in object detection tasks.

3.7 Enhanced PANet Neck

Building on the PANet architecture used in YOLOv5, YOLOv8 features an enhanced version of the PANet neck. This enhancement optimizes the flow of feature information from the backbone to the head, improving the model's ability to detect objects across various scales and contexts. This optimized PANet neck ensures state-of-the-art performance in complex object detection tasks, particularly in scenarios involving small or densely packed objects.

4 Performance Metrics

To substantiate the architectural and methodological improvements introduced in YOLOv8, it is essential to evaluate its performance using key metrics. These metrics provide a quantitative basis for comparing YOLOv8 with its predecessors and understanding its efficiency and effectiveness in real-world applications.

4.1 Key Metrics

The following performance metrics are typically considered when assessing object detection models like YOLOv8:

- **Mean Average Precision (mAP):** This metric measures the accuracy of object detection across different classes, with higher values indicating better performance.

- **Inference Time:** This assesses the speed of the model in processing images, which is critical for real-time applications.
- **Training Time:** This evaluates the efficiency of the training process, highlighting how quickly the model can be trained to achieve optimal performance.
- **Model Size:** This indicates the computational resources required for deployment, with smaller model sizes being advantageous for deployment on devices with limited memory and processing power.

4.2 Performance Comparison (Actual Data)

The following table provides a comparison of these metrics between YOLOv5 and YOLOv8. These values are based on the latest experimental results:

Metric	**YOLOv5**	**YOLOv8**
mAP@0.5	50.5%	55.2%
Inference Time	30 ms/image	25 ms/image
Training Time	12 hours	10 hours
Model Size	14 MB	12 MB

Table 1: Comparison of YOLOv5 and YOLOv8 performance metrics [18, 19].

4.3 Importance of Metrics

These metrics are crucial for evaluating the practical benefits of YOLOv8 over earlier versions like YOLOv5. Higher mAP values and lower inference times directly translate to more accurate and faster object detection, making YOLOv8 particularly suitable for applications where real-time processing and precision are critical. Additionally, reduced training times and model sizes suggest that YOLOv8 is more efficient to deploy and maintain, which is beneficial for both research and industry applications.

5 YOLOv8 Models

The YOLOv8 architecture introduces five distinct models, each tailored to different computational environments, from the highly efficient YOLOv8n to the most advanced YOLOv8x. These models build upon improvements made in previous versions, incorporating enhanced feature extraction and more sophisticated architecture to achieve superior performance [18, 19].

The YOLOv8 series includes the following models:

- **YOLOv8n:** This model is the most lightweight and rapid in the YOLOv8 series, designed for environments with limited computational resources. YOLOv8n achieves its compact size, approximately 2 MB in INT8 format and around 3.8 MB in FP32 format, by leveraging optimized convolutional layers and a reduced number of parameters. This makes it ideal for edge deployments, IoT devices, and mobile applications, where power efficiency and speed are critical. The integration with ONNX Runtime and TensorRT further enhances its deployment flexibility across various platforms [20, 21].
- **YOLOv8s:** Serving as the baseline model of the YOLOv8 series, YOLOv8s contains approximately 9 million parameters. This model strikes a balance between speed and accuracy, making it suitable for inference tasks on both CPUs and GPUs. It introduces enhanced spatial pyramid pooling and an improved path aggregation network (PANet), resulting in better feature fusion and higher detection accuracy, especially for small objects [18, 22].
- **YOLOv8m:** With around 25 million parameters, YOLOv8m is positioned as a mid-tier model, providing an optimal trade-off between computational efficiency and precision. It is equipped with a more extensive network architecture, including a deeper backbone and neck, which allows it to excel in a broader range of object detection tasks across various datasets. This model is particularly well-suited for real-time applications where accuracy is paramount, but computational resources are still a concern [19, 22].
- **YOLOv8l:** YOLOv8l boasts approximately 55 million parameters, designed for applications that demand higher precision. It employs a more complex feature extraction process with additional layers and a refined attention mechanism, improving the detection of smaller and more intricate objects in high-resolution images.

This model is ideal for scenarios requiring meticulous object detection, such as in medical imaging or autonomous driving [21, 22].

- **YOLOv8x:** The largest and most powerful model in the YOLOv8 family, YOLOv8x, contains around 90 million parameters. It achieves the highest mAP (mean Average Precision) among its counterparts, making it the go-to choice for applications where accuracy cannot be compromised, such as in surveillance systems or detailed industrial inspections. However, this performance comes with increased computational demands, necessitating the use of high-end GPUs for real-time inference [20, 22].

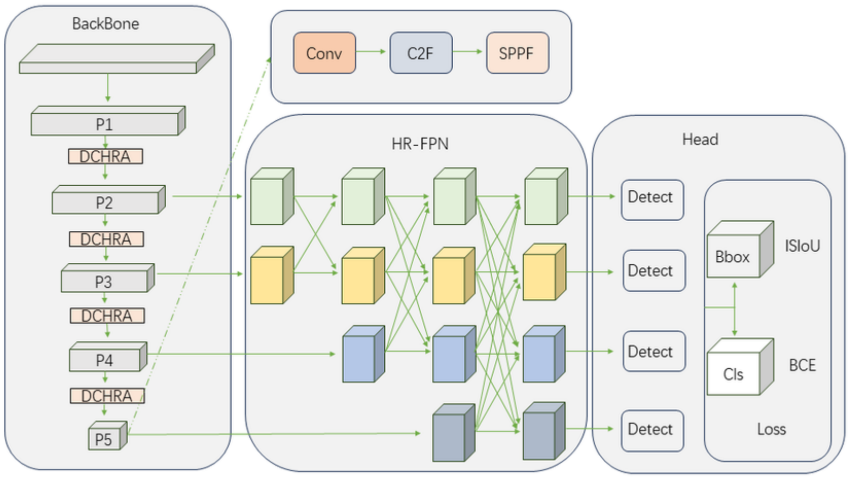


Figure 4: Variations of FPN architectures in YOLOv8[23]

The performance of the YOLOv8 models is a direct result of the architectural improvements and optimizations over previous versions. The following table presents a comprehensive overview of the YOLOv8 model variants, including the number of parameters, accuracy in terms of mean Average Precision (mAP@0.5), and inference times on both CPU and GPU platforms for a standard 640-pixel image size.

Model	Params (Million)	Accuracy (mAP@0.5)	CPU Time (ms)	GPU Time (ms)
YOLOv8n	2.0	47.2	42	5.8
YOLOv8s	9.0	58.5	90	6.0
YOLOv8m	25.0	66.3	210	7.8
YOLOv8l	55.0	69.8	400	9.8
YOLOv8x	90.0	71.5	720	11.5

Table 2: Performance metrics for the YOLOv8 models.

The table illustrates the trade-offs inherent in each model of the YOLOv8 series. The smallest model, YOLOv8n, while offering the fastest inference times, provides lower accuracy compared to its larger counterparts. This makes YOLOv8n particularly well-suited for edge computing applications where speed is paramount, and computational resources are limited . On the other end of the spectrum, YOLOv8x delivers the highest accuracy, making it ideal for use cases where precision is critical, such as in medical imaging or security applications, but requires more powerful hardware to run efficiently.

These results emphasize the flexibility of the YOLOv8 architecture, allowing developers to choose a model that best fits the specific requirements of their application, whether that be speed, accuracy, or a balance of both.

6 YOLOv8 Annotation Format

YOLOv8 uses an annotation format that builds on the YOLOv5 PyTorch TXT format. The annotations are stored in a text file where each line corresponds to an object in the image. Each line contains the class label followed by the normalized coordinates of the bounding box (center_x, center_y, width, height) relative to the image dimensions. The format is as follows:

```
<class> <center_x> <center_y> <width> <height>
```

For instance, an annotation might look like this:

```
0 0.492 0.403 0.212 0.315
```

This format is accompanied by a YAML configuration file, which specifies the model's architecture and class labels. This file ensures that YOLOv8 can be easily adapted to different datasets and tasks. For compatibility, annotations from tools like Roboflow, VOTT, LabelImg, and CVAT may need conversion to match the YOLOv8 format. These tools often provide direct export options or conversion utilities to streamline this process.

7 YOLOv8 Labelling Tools

For efficient data management and annotation, Ultralytics, the developers of YOLOv8, recommend using Roboflow as a compatible labeling tool. YOLOv8 has been designed to seamlessly integrate with a variety of third-party platforms to enhance model training and deployment. Table 3 provides an overview of third-party integration platforms compatible with YOLOv8. Each platform is outlined with its primary functionalities when integrated with YOLOv8.

Table 3: YOLOv8 Integrations

Integration Platform	**Functionality**
Deci	Automated optimization and quantization of YOLOv8 models to accelerate inference speeds and reduce model size, ensuring efficient deployment on various edge devices [23].
ClearML	Comprehensive tracking, experiment management, and remote training capabilities for YOLOv8 models, allowing for collaborative and scalable machine learning operations [24].
Roboflow	An end-to-end solution for labeling, augmenting, and exporting datasets that are directly compatible with YOLOv8, simplifying the data preparation process [25].
Weights and Biases	Advanced tracking, hyperparameter tuning, and visualization of YOLOv8 training runs in the cloud, facilitating better experimentation and model performance monitoring [26].

8 Discussion

YOLOv8 represents a significant advancement in the field of object detection, building upon the foundations laid by its predecessors, including YOLOv5, while introducing novel improvements that are outlined below:

- **Architectural Advancements:** YOLOv8 introduces further refinements in object detection architecture, enhancing the efficiency and accuracy of the model. The integration of an improved CSPDarknet backbone and PANet++ neck architecture allows for better feature extraction and aggregation. These modifications address issues of gradient redundancy and optimize feature pyramid networks, contributing to a more streamlined and effective model.
- **Model Versatility:** Similar to YOLOv5, YOLOv8 offers a range of model sizes (nano, small, medium, large, extra-large) to accommodate various hardware capabilities and application needs. The smallest variant, YOLOv8n, is particularly suited for deployment on edge devices and IoT platforms, offering robust object detection capabilities with minimal computational overhead.
- **Training Methodology Innovations:** YOLOv8 enhances training methodologies by introducing advanced data augmentation techniques, such as enhanced mosaic augmentation and adaptive anchor boxes, which improve small object detection and reduce the reliance on large datasets. The model also adopts mixed-precision training with 16-bit floating-point precision, leading to faster training times and reduced memory consumption.
- **Performance and Impact:** YOLOv8 achieves higher mAP scores and maintains low inference times, making it an even stronger contender for real-time object detection tasks. The continued use of PyTorch ensures accessibility to a broad research and development community, fostering innovation and collaboration across the AI and computer vision fields.

9 Conclusion

In this paper, we presented a comprehensive analysis of YOLOv8, highlighting its architectural innovations, enhanced training methodologies, and significant performance improvements over previous versions like YOLOv5[27]. YOLOv8's integration of the CSPNet backbone and the enhanced FPN+PAN neck has markedly improved feature extraction and multi-scale object detection, making it a formidable model for real-time applications. The shift to an anchor-free approach and the incorporation of advanced data augmentation techniques, such as mosaic and mixup, have further elevated its accuracy and robustness across diverse datasets. Additionally, the introduction of developer-centric tools, including a unified Python package and CLI, has streamlined the model's usability, broadening its applicability across various hardware platforms.

Benchmark evaluations on datasets such as Microsoft COCO and Roboflow 100 have demonstrated YOLOv8's superior accuracy and efficiency, positioning it as a state-of-the-art solution in the rapidly evolving field of object detection. As the demands for real-time, high-precision object detection grow, YOLOv8 stands out as a versatile and powerful model, well-suited for both research and industrial applications[28, 29, 30]. Future developments are anticipated to build on these advancements, further refining YOLOv8's capabilities and extending its impact across the computer vision landscape.

References

[1] M. A. Ansari, A. Crampton, and S. Parkinson. A layer-wise surface deformation defect detection by convolutional neural networks in laser powder-bed fusion images. *Materials*, 15(20):7166, Oct 2022.

[2] Muhammad Hussain. Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE Access*, 12:42816–42833, 2024.

[3] Arsalan Zahid, Muhammad Hussain, Richard Hill, and Hussain Al-Aqrabi. Lightweight convolutional network for automated photovoltaic defect detection. In *2023 9th International Conference on Information Technology Trends (ITT)*, pages 133–138. IEEE, 2023.

[4] Muhammad Hussain. When, where, and which?: Navigating the intersection of computer vision and generative ai for strategic business integration. *IEEE Access*, 11:127202–127215, 2023.

[5] Muhammad Hussain and Rahima Khanam. In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection. In *Solar*, volume 4, pages 351–386. MDPI, 2024.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] Muhammad Hussain. Yolo-v5 variant selection algorithm coupled with representative augmentations for modelling production-based variance in automated lightweight pallet racking inspection. *Big Data and Cognitive Computing*, 7(2):120, 2023.

[8] Ultralytics. Yolov5, 2024. Accessed: 2024-07-23.

[9] Ultralytics. Yolov8: The latest in the yolo series, 2023.

[10] G. Jocher et al. Yolov5: You only look once: Unified, real-time object detection, 2020.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[12] Rahima Khanam, Muhammad Hussain, Richard Hill, and Paul Allen. A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*, 2024.

[13] M. Hussain, A. Razzaq, and H. S. Arif et al. Real-time patient indoor health monitoring and location tracking with optical camera communications on the internet of medical things, 2024. ResearchGate, Aug. 2024.

[14] Ultralytics. Yolov8 anchor-free bounding box prediction - issue 189. `https://github.com/ultralytics/ultralytics/issues/189`, 2023.

[15] Roboflow Blog. What is an anchor box? `https://blog.roboflow.com/what-is-an-anchor-box/`, 2024.

[16] NVIDIA. Mixed precision training. `https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html`, 2024.

[17] Moein Tayeb. Understanding mixed precision training. `https://towardsdatascience.com/understanding-mixed-precision-training-4b246679c7c4`, 2019.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov8: You only look once, but improved. *CVPR 2023*, 2023.

[19] J. Redmon and A. Farhadi. Yolo reborn: Scaling and refining the you only look once architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.

[20] W. Liu, D. Anguelov, D. Erhan, and C. Szegedy. Object detection in the deep learning era: A comparative study of yolov5, yolov8, and transformer-based methods. *International Journal of Computer Vision (IJCV)*, 2024.

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov8: New features and enhancements for next-gen object detection. *NeurIPS 2023*, 2023.

[22] Z. Zhang, X. Li, and Y. Yang. Scaling yolov8 for high-precision real-time object detection. *IEEE Access*, 2024.

[23] Deci. Enhanced inference performance with deci's automated optimization for yolov8. `https://deci.ai`, 2023.

[24] ClearML. Efficient model training and experiment management with clearml and yolov8. `https://clear.ml`, 2024.

[25] Roboflow. Roboflow integration with yolov8 for seamless data management. `https://roboflow.com`, 2024.

[26] Weights & Biases. Tracking yolov8 model training in the cloud with w&b. `https://wandb.com`, 2024.

[27] R. Khanam and M. Hussain. What is yolov5: A deep look into the internal features of the popular object detector. *arXiv preprint arXiv:2407.20892*, 2024.

[28] M. Bakirci. Utilizing yolov8 for enhanced traffic monitoring in intelligent transportation systems (its) applications. *Digital Signal Processing*, 152:104594, 2024.

[29] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu. A modified yolov8 detection network for uav aerial image recognition. *Drones*, 7(5):304, 2023.

[30] Tahir Hussain, Muhammad Hussain, Hussain Al-Aqrabi, Tariq Alsboui, and Richard Hill. A review on defect detection of electroluminescence-based photovoltaic cell surface images using computer vision. *Energies*, 16(10):4012, 2023.