Utkal University Journal of Computing & Communications,
Vol. 2, Issue. 2 (2024) *e-ISSN:2584-203X*

DimensioNet: A Real-Time System for Object Detection, Dimension Estimation, and Dominant Color Identification Using YOLOv8

Babita Mohalik[1], Jibitesh Mishra[1][0000-0003-2761-470X]

[1]Odisha University of Technology and Research, Odisha, India
jmishra@outr.ac.in

Abstract: This research presents DimensioNet, a real-time object detection and analysis system capable of identifying objects, estimating their physical dimensions, and extracting dominant color information from live video feeds. Developed using the YOLOv8 deep learning framework, the system integrates advanced computer vision techniques to provide accurate, efficient, and interpretable results. Object dimensions are computed using bounding box analysis and a calibrated scaling factor, while dominant color is determined by analyzing the average pixel values of detected objects. The system employs Python libraries such as OpenCV and Matplotlib to enable real-time frame processing and visual annotation. DimensioNet is designed for practical deployment in sectors like logistics, manufacturing, and e-commerce, where rapid and automated object analysis is essential. Evaluation of the system demonstrates high precision in detection, reliable color identification, and consistent dimension estimation. The work lays a foundation for future improvements, including multi-object tracking, integration with IoT devices, and adaptive real-world applications.

Keywords: Real-time Object detection, YOLOv8, Dimension Estimation, Color Identification

1. Introduction

The increasing reliance on intelligent systems for automating industrial and commercial tasks has significantly accelerated advancements in real-time object detection and video analytics. Traditional manual methods for measuring object dimensions and identifying features are not only time-consuming but also prone to human error, especially in fast-paced environments like logistics, e-commerce, and manufacturing. To address these challenges, this research proposes DimensioNet, a real-time system that integrates object detection, dimension estimation, and dominant color analysis into a single pipeline.

With the evolution of deep learning, especially convolutional neural networks (CNNs), object detection has witnessed substantial progress. Among the most notable frameworks is YOLO (You Only Look Once), known for its ability to balance detection speed and accuracy. Leveraging the capabilities of YOLOv8, the latest and most optimized version in the YOLO family, DimensioNet processes live video feeds to identify and analyze objects dynamically.

What distinguishes DimensioNet from conventional detection systems is its focus on three integrated outputs: the identity of the object, its real-world dimensions, and its dominant visual color. These features are extracted in real

✉ jmishra@outr.ac.in

time and overlaid on video frames using Python-based tools such as OpenCV and Matplotlib, offering both precision and interpretability.

This paper presents the design, implementation, and evaluation of DimensioNet, demonstrating its effectiveness in real-time scenarios. The proposed system aims not only to automate visual inspection tasks but also to provide a scalable solution for future integration with edge computing and IoT-enabled infrastructures.

This paper is structured as follows: Section 2 states the problem statement. Section 3 discusses the literature review and background. The proposed methodology is given in Section 4. Section 5 mentions the results and discussion. Finally, Section 6 gives the conclusion of the work.

2. Problem Statements

The idea got conceptualized in response to the limitations of traditional object detection systems that do not provide real-time object description beyond class labels. By incorporating measurement and color analysis into the detection loop, it fills an important gap in intelligent video analytics.

Real-time object detection systems often focus primarily on classification tasks, with limited capabilities for physical measurement or descriptive analysis. In industries such as logistics, e-commerce, and quality inspection, merely identifying an object is not enough dimensions and visual attributes such as color are equally important for automation and categorization. However, implementing a reliable, real-time system that combines all these functionalities remains a complex task.

2.1. Challenges

Various challenges need to be addressed in finding real-time object description as given below.

- **Video input variability**: Lighting changes, object occlusion, and frame quality affect detection consistency.
- **Dimension estimation**: Converting bounding box measurements to real-world units is sensitive to camera calibration and object distance.
- **Color accuracy**: Environmental lighting and shadows can distort color perception, making it hard to extract dominant object colors reliably.
- **System responsiveness**: Maintaining low latency during live frame processing is crucial for usability.
- **Integration and scalability**: The system should be capable of adapting to different devices and real-world deployment environments.

2.2. Objectives

The primary goal of this research is to develop an integrated, real-time visual analyt-ics system capable of not just identifying objects but also describing their physical properties such as dimensions and color. To accomplish this, the following specific objectives have been outlined.

•*To develop a real-time object detection system using YOLOv8:* Leverage the capabilities of YOLOv8, the latest version in the YOLO (You Only Look Once) series of object detection frameworks, to detect objects in live video streams with high speed and accuracy. The system should be optimized for real-time performance and able to identify multiple object classes dynamically.

•*To implement physical dimension estimation using bounding box geometry and a calibrated scaling factor:* Introduce a method that calculates the real-world width and height of detected objects by translating the pixel di-

mensions of bounding boxes into physical units (such as centimeters). This translation is achieved using a predefined scaling factor, assuming a fixed camera position.

•*To identify and display the dominant color of each detected object:* Apply color analysis techniques to extract the most prominent color from the cropped region of each detected object using mean BGR (Blue, Green, Red) values. This information is used to visually describe objects beyond their class label, making the system more informative and practical for use cases like product identification and quality inspection.

•*To design a unified, annotated visualization pipeline for real-time feed-back:* Integrate object class, estimated dimensions, and dominant color into a single annotated video frame, providing users with a clear and descriptive overlay for each detected object. Use Python libraries like OpenCV and Matplotlib to perform real-time drawing and display.

•*To evaluate the performance of the system based on detection accuracy, dimension estimation reliability, and color consistency:* Conduct experi-mental evaluations under various lighting conditions and camera setups to measure the system's effectiveness in real-world scenarios. Key performance indicators include detection precision, error rate in dimension calculation, and accuracy in color identification.

•*To ensure low-latency, resource-efficient processing suitable for real-time deployment:* Optimize the system for smooth operation on standard computing hardware by maintaining minimal processing delay per frame. The focus is to balance detection accuracy with speed for seamless real-time applications.

•*To design the system for scalability and adaptability to industrial use cases:* Ensure that the system can be deployed in static environments such as factory floors, warehouses, or logistics hubs, where objects frequently pass through fixed camera zones. The modular nature of the pipeline should allow for customization and expansion as per different deployment requirements.

•*To lay the foundation for future enhancements like depth-aware dimension estimation and IoT integration:* Create a flexible architecture that can later be extended to include depth cameras or stereo vision for more accurate scaling, support for multi-object tracking, and connectivity with edge/IoT platforms for automated decision-making and monitoring

3. Literature Review

This section discusses existing work on deep learning based object detection and dimension estimation. This is followed by the research challenges of the studied literature.

3.1. Deep Learning in Real-time Object Detection

Deep learning has revolutionized the field of object detection, allowing for robust identification and tracking of objects in both static images and live video streams. The YOLO (You Only Look Once) architecture is widely recognized for its ability to perform real-time detection by treating the task as a single regression problem [1].

Redmon et al. (2016) introduced the initial YOLO model, which has since undergone multiple improvements. YOLOv4 [2] shows strong balance of speed and precision, outperforming earlier versions. But the redefined anchor boxes for bounding-box can reduce detection accuracy for objects with unusual aspect ratios or sizes. Therefore, the introduction of YOLOv8 made the difference [3]. Padilla et al. analyzes popular object detection evaluation metrics (e.g., AP, IoU thresholds, and mAP) and introduc-es an open-source toolkit for systematic metric computation [4].

Also, use of also YOLOv8 marks much changes realtime object detection though Ren et al. (2016) discussed about a faster R CNN, integrating a Region Proposal Net-work (RPN) with a CNN classifier to a unified architecture [5]. Lin et al. (2017) pro-posed Focal Loss in the RetinaNet framework to address class imbalance in dense detectors [6].

Even Wang et al. (2020) proposed CSPNet that partitioned feature maps across stages to reduce computation while preserving accuracy[12]. Other pa-pers from [8-11] discusses about such object detection techniques, but lacked several problems.

3.2. Object Dimension Estimation Using Bounding Boxes

Object dimension estimation in video analytics often uses bounding boxes generated by detection models. By applying a known scaling factor, these pixel-based dimensions are converted into approximate real-world measurements [1].

Previous work has shown that while bounding box estimation is computationally efficient, it is sensitive to camera placement and calibration [4]. Approaches using depth cameras or stereo vision have been suggested as more dynamic solutions but are less accessible due to cost and complexity.

Color detection, though secondary to shape and size in many detection systems, is increasingly recognized for its role in product classification and quality assurance. Studies describe the use of mean BGR or HSV values to determine the dominant color of a segmented region in an image [5].

Few systems currently offer the integrated functionality that DimensioNet proposes—combining detection, measurement, and descriptive color analysis [6]. Most existing frameworks focus on one or two of these aspects independently.

Despite technological advancements, challenges persist in building comprehensive real-time video analysis systems:

- Scaling factor dependency [3]
- Lighting conditions [5]
- Computation efficiency [6]
- Adaptability [4]

4. Proposed Methodology

The workflow diagram of DimensioNet using YOLOv8 and its working procedure is mentioned in this sections. This is justified by the method to improve the concept by height and width of the object.

4.1. Workflow Diagram

The workflow diagram of the procedure adopted by us may be seen in figure 1. The figure starts with initializing the system starting from capturing frames from video stream till detecting the object. It is followed by calculating the dimension or size of the object is calculated as well as the dominant color is found, which is shown in real-time. The video is fed to the designed system and each frame is processed to detect the objects. After the processing of all the frames, the dimensions of the objects are estimated based on the defined formula. The detail algorithm for the purpose is given in section 4.2.

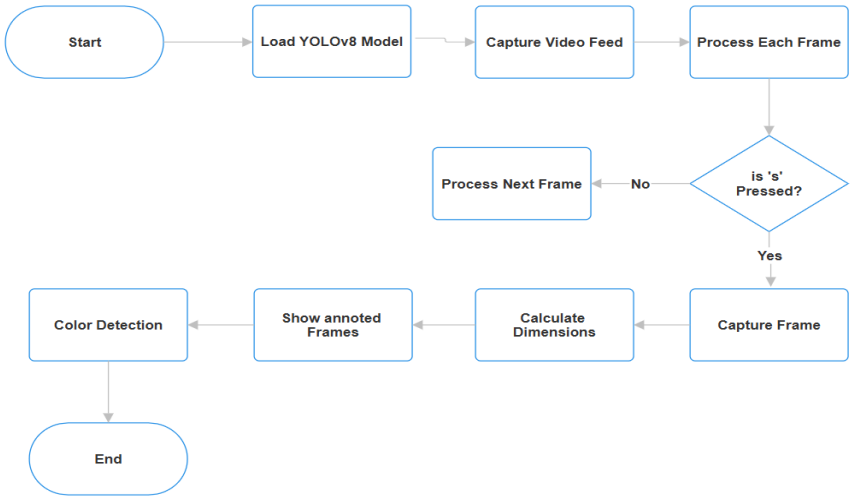

Fig. 1. Workflow of DimensioNet using YOLOv8

4.2. Object Detection Algorithm

The total workflow diagram can be executed using the steps in Algorithm 1. The other steps in alogorithm-1 consist of capturing frames from video stream in order to detect the object. Then the dimension or size of the object is calculated as well as the dominant color is found in order to show them in real-time.

Algorithm 1: DimensioNet – Real-Time Object Detection and Analysis

Input: Live video stream from a camera or video file

Output: Annotated video frames with bounding boxes, object dimensions, and dominant color

Step 1: Initialize System Components

- Load pre-trained YOLOv8 model using Ultralytics library.
- Set video input source (e.g., webcam or video file).
- Define the scaling factor S (experimentally determined in cm/pixel).

Step 2: Capture Frame from Video Stream

- Read the next frame from the video input using OpenCV.
- Resize or preprocess frame if necessary (e.g., for performance).

Step 3: Object Detection

- Apply YOLOv8 model to the frame to detect objects.
- For each detected object:
  - Retrieve bounding box coordinates: (x, y, width, height)
  - Retrieve class label (e.g., "bottle", "box")

Step 4: Dimension Estimation

- For each bounding box:

  Calculate real-world width = width_pixels × S

Calculate real-world height = height_pixels × S

- Format the dimension output (e.g., "Width: 12.5 cm, Height: 18.2 cm")

Step 5: Dominant Color Detection

- Crop the detected object region from the frame using bounding box.
- Compute the mean BGR color values of the cropped region.

Step 6: Visualization and Annotation

- Draw the bounding box around the object on the frame.
- Overlay:
  - Class label (e.g., "Box")
  - Dimensions (in cm)
  - Color label
- Display the annotated frame in a real-time window.

Step 7: Loop

- Repeat Steps 2 to 6 for each incoming frame.
- Allow exit on user command (e.g., pressing 'q').

4.3. Justifications

To estimate the physical size of detected objects, the DimensioNet system utilizes the bounding box coordinates returned by the YOLOv8 detection model. These co-ordinates are expressed in pixel units and must be converted to real-world units (e.g., centimeters). This is accomplished using a predefined scaling factor, which serves as a static calibration value determined during initial setup.

$$\text{Width(cm)} = \frac{Width(pixels)}{Scaling\ Factor} \quad \text{Height(cm)} = \frac{Height(pixels)}{Scaling\ Factor} \tag{1}$$

Where:

- Width (pixels) = x2 − x1, the horizontal distance between the left and right edges of the bounding box.
- Height (pixels) = y2 − y1, the vertical distance between the top and bottom edges of the bounding box.
- Scaling Factor = A fixed numerical value (e.g., 20) that converts pixel units into real-world measurements.

Fixed Scaling Factor

A fixed scaling factor provides a static method to map pixel measurements to real-world units without needing to know the object's exact distance.This assumes that

- The camera position is fixed.
- Objects appear within a consistent physical range.

This method improves computational speed and maintains reasonable accuracy.

5. Results and Discussions

The increasing reliance on intelligent systems for automating industrial and commercial DimensioNet system utilizes the bounding box coordinates by the YOLOv8 detection model to estimate the physical size of badmiton racket as 28.82 cm x 10.85 cm in figure 2. It gives the colour value as RGB (95,135,115).

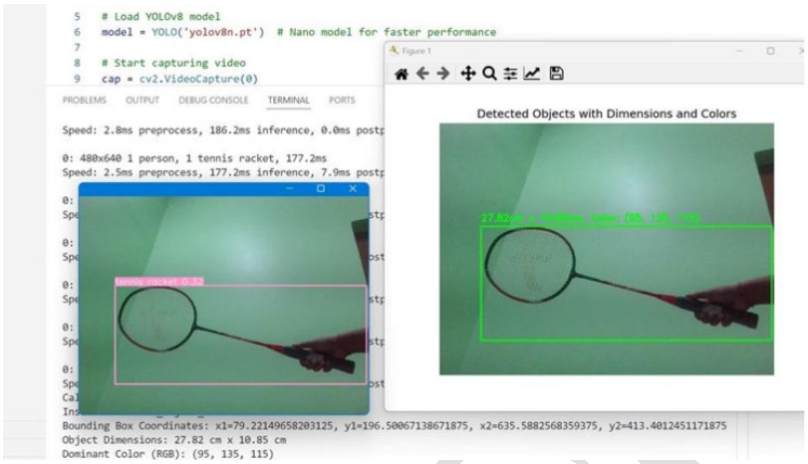

Fig. 2. Output Diagrams of DimensioNet

Further, in another example, it detects a flower vase with flower to estimate the physical size of flower vase as 12.77 cm x 23.41 cm in figure 3. It gives the colour value as well.

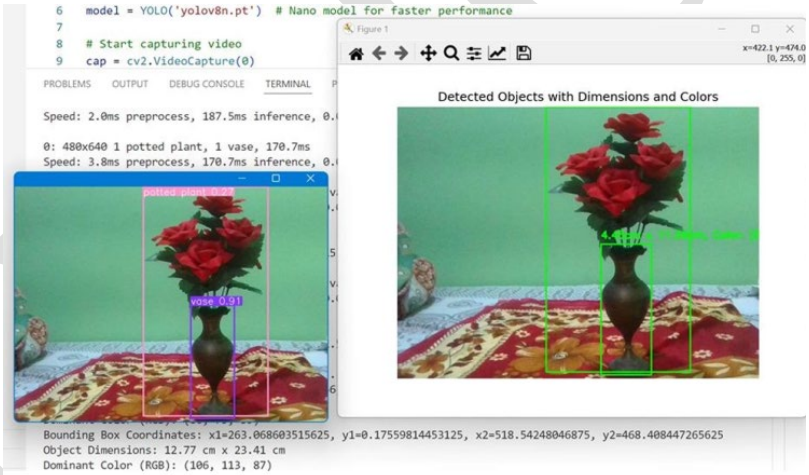

Fig. 3. Output Diagrams of DimensioNet

The accuracy for finding the size of the object is 99% and therefore, the formula adopted by us as far as the size of the image is concerned is a novel technique of using it for fixed distance objects. It can be very well used in industrial automation.

6. Conclusion

This research successfully introduces DimensioNet, a real-time system that unifies object detection, physical dimension estimation, and dominant color analysis using the YOLOv8 deep learning framework. By integrating these components into a single pipeline, the system addresses a significant gap in current video analytics applications delivering not only object classification but also spatial and visual descriptors necessary for industrial automation.

DimensioNet demonstrates high efficiency and precision in detecting various objects, accurately estimating their

dimensions using a calibrated fixed scaling factor, and extracting dominant color features based on average pixel values. The use of widely available tools such as Python, OpenCV, and Matplotlib makes the system both accessible and adaptable for real-world deployment. While the fixed scaling approach introduces some limitations regarding distance variability, it ensures fast and repeatable results under controlled conditions—an essential factor in logistics, manufacturing, and retail settings.

The results confirm that DimensioNet is a viable and scalable solution for environments that demand quick, descriptive object analysis without sacrificing performance. Moreover, this work lays a strong foundation for future enhancements, including depth-aware scaling, adaptive lighting compensation, and integration with IoT and edge computing systems. Ultimately, DimensioNet represents a meaningful step toward smarter, context-aware computer vision solutions in real-time applications.

6.1. Research Motivation and DimensioNet's Contribution

DimensioNet was conceptualized in response to the limitations of traditional object detection systems that do not provide real-time object description beyond class labels. By incorporating measurement and color analysis into the detection loop, it fills an important gap in intelligent video analytics [6].

6.2. Problem Statement and Objectives

Real-time object detection systems often focus primarily on classification tasks, with limited capabilities for physical measurement or descriptive analysis. In industries such as logistics, e-commerce, and quality inspection, merely identifying an object is not enough. The dimensions and visual attributes such as color are equally important for automation and categorization. However, implementing a reliable, real-time system that combines all these functionalities remains a complex task.

Several challenges exist in developing such a system:

- Video input variability: Lighting changes, object occlusion, and frame quality affect detection consistency.
- Dimension estimation: Converting bounding box measurements to real-world units is sensitive to camera calibration and object distance.
- Color accuracy: Environmental lighting and shadows can distort color perception, making it hard to extract dominant object colors reliably.
- System responsiveness: Maintaining low latency during live frame processing is crucial for usability.
- Integration and scalability: The system should be capable of adapting to different devices and real-world deployment environments.

To address these challenges, this research proposes DimensioNet, a unified system that detects objects in real time, estimates their physical dimensions using a fixed scaling factor, and identifies dominant colors through pixel analysis—all visualized within a live video stream.

6.3. Limitations of the Proposed Work

While DimensioNet demonstrates promising results in integrating object detection, dimension estimation, and dominant color analysis in real time, several constraints limit its generalization and scalability in dynamic environments. These limitations are discussed as follows:

Fixed Distance and Scaling Factor Dependence

The system currently uses a fixed scaling factor to convert pixel dimensions into real-world measurements. This

approach assumes that all objects are located within a constant distance from the camera. As a result, measure-ment accuracy decreases when objects appear closer or farther than the calibrated distance. Future improvements should incorporate adaptive scaling using either depth estimation, stereo vision, or LiDAR-based range sensing to enhance robustness.

Lighting Dependency in Color Detection

The accuracy of dominant color extraction heavily depends on ambient lighting and reflections. Under low or uneven illumination, the mean BGR calculation can misrepresent the actual object color. Techniques such as white balance normalization, histogram equalization, or illumination-invariant color models (like LAB or HSV-based normalization) can mitigate this issue in future versions.

Limited Object Categories and Dataset Scope

The system's evaluation is currently limited to a small number of object types captured under controlled conditions. Broader dataset coverage and class diversity are required to improve generalization across various indus-tries. Incorporating domain-specific fine-tuning or transfer learning can fur-ther enhance detection accuracy and contextual adaptability.

Computational Efficiency Constraints

While YOLOv8 offers a strong balance between accuracy and inference speed, real-time operation on edge or embedded devices remains challenging without GPU acceleration. Future optimization can leverage lightweight backbones such as YOLO-NAS or quantization techniques to achieve faster deployment on resource-limited platforms.

Potential Improvements

Integrating multi-object tracking (MOT), depth-aware scaling, and adaptive illumination correction would make the system more versatile. Additionally, coupling DimensioNet with IoT-enabled infrastructure can allow automated analytics and decision-making in smart industrial setups.

References

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788.
2. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
3. Jocher, G. (2023). YOLOv8 Docs & Tutorials. Ultralytics. https://docs.ultralytics.com
4. Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. B. (2021). A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. Electronics, 10(3), 279.
5. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems (NeurIPS), 28.
6. Lin, T. Y., Goyal, P., Girshick, R., He, K., &Dollár, P. (2017). Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision, 2980–2988.
7. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision (ECCV), 740–755.
8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

9. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.
10. Russakovsky, O., Deng, J., Su, H., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211–252.
11. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2921–2929.
12. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.