



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Mobile Vision Based System For Measuring Object Dimensions & Area From Live Images

¹Prof. Pratik Suresh Deshmukh, ²Mr. Yash Purohit, ³Mr. Ankush Kadukar, ⁴Miss. Vedanti Khorgade,

⁵Mr. Amar Sawade, ⁶Mr. Aditya Nagtode

^{1,2,3,4,5,6} Department of Computer Science & Engineering,

^{1,2,3,4,5,6} PRMIT&R Badnera, Amravati. Amravati,

Abstract - The implementation of the proposed real-time measurement system integrates advanced machine learning, sensor data processing, and an interactive mobile user interface to deliver an efficient and accurate measurement solution. The system is built upon a pre-trained YOLOv5 object detection model, with the flexibility for custom training using datasets such as COCO to improve detection accuracy for domain-specific applications. Careful tuning of training parameters and evaluation using metrics like precision, recall, F1-score, and mean average precision (mAP) ensures reliable model performance.

To enable real-time execution on mobile devices, the model undergoes optimization techniques including quantization, pruning, and hardware acceleration, significantly reducing computational complexity while maintaining accuracy. The integration of smartphone sensors, particularly the accelerometer, allows continuous real-time capture of device orientation data, which is processed and synchronized with the camera feed to support accurate depth estimation.

The system also features a well-designed user interface that overlays detection results and measurement outputs directly onto the live camera view. Real-time visualization of object

dimensions, along with user-controlled inputs such as device height, enhances usability and measurement precision. Overall, the implementation ensures low latency, high responsiveness, and a seamless user experience, making the system robust and suitable for real-world mobile application.

Keywords - Real-time measurement, object detection, YOLOv5, machine learning, mobile application, sensor integration, accelerometer data, depth estimation, model optimization, quantization, computer vision, real-time processing, user interface, bounding box detection, and mean average precision (mAP).

I. INTRODUCTION

The rapid growth of smartphone technology has created new opportunities for developing intelligent applications that simplify everyday tasks. One such area is object measurement, where traditional tools like rulers or measuring tapes are often inconvenient, time-consuming, and limited in dynamic or hard-to-reach environments. With the integration of advanced computer vision and sensor capabilities in modern mobile devices, it is now possible to estimate object dimensions in real time using only a smartphone[1].

This project introduces a real-time object measurement system that combines deep learning-based object detection with sensor-driven depth estimation. The system detects objects through a live camera feed and calculates their physical dimensions by utilizing device orientation data obtained from built-in sensors. This approach eliminates the need for physical contact with the object, making the process faster, more flexible, and user-friendly[2].

To achieve efficient performance on mobile devices, the system employs an optimized object detection model capable of delivering accurate results with low computational overhead. At the same time, continuous sensor data processing ensures that the system adapts to device movement, maintaining consistency in measurement results. The integration of these components allows the application to function smoothly in real-time scenarios[3].

In addition, the system is designed with a simple and intuitive user interface that presents measurement information directly on the camera screen. This enhances usability by allowing users to visualize results instantly without interrupting the viewing experience. Overall, the proposed solution offers a practical and efficient alternative to traditional measurement methods, with potential applications in fields such as construction, education, interior design, and daily use[4].

II. LITERATURE ANALYSIS

The literature on real-time visual understanding and depth estimation demonstrates a clear evolution of models and approaches, each contributing unique capabilities while facing specific limitations. Early work by Redmon et al. [1] introduced YOLO, enabling real-time object detection with a single-stage architecture, though it struggled with small object accuracy.

Subsequent versions, YOLO9000 [2] and YOLOv3 [3], improved class detection and multi-scale predictions, but still faced localization challenges and high computational costs, especially for mobile deployment. YOLOv4 [4] optimized the speed-accuracy trade-off, requiring GPU support for peak performance, while YOLOv5 [5] focused on lightweight deployment for real-time use despite lacking formal academic validation. In depth estimation, monocular vision theories [8, 9] laid the groundwork for understanding projection ambiguities but could not directly estimate depth. Geometric context-based approaches [18] and deep learning methods [19] enabled scene understanding and depth prediction from single images, yet they were limited by scene complexity and computational requirements. Motion-based methods, such as visual odometry [11] and visual-inertial SLAM [12], leveraged sequential frames and sensor fusion to improve accuracy, though they required continuous camera movement and were computationally intensive. Probabilistic sensor fusion techniques [15] further enhanced robustness through multi-sensor integration, while OpenCV-based camera calibration [20] provided practical tools for translating pixel measurements into real-world units, though requiring prior calibration and known parameters. Overall, these works highlight the trade-offs between accuracy, computational efficiency, and mobile applicability, indicating ongoing opportunities to develop lightweight, real-time vision systems suitable for resource-constrained devices.

TABLE I.
LITERATURE WORK

S.No	Author(s)	Approach/Model	Contribution	Limitation
1	Redmon et.al [1]	YOLO	Real-time object detection using single-stage architecture	Lower accuracy for small objects in early versions
2	Redmon & Farhadi [2]	YOLO9000	Improved accuracy and detection of more classes	Still limited in localization precision
3	Redmon & Farhadi [3]	YOLOv3	Multi-scale detection and improved backbone	High computational cost for mobile
4	Bochkovskiy et al. [4]	YOLOv4	Optimized speed and accuracy trade-off	Requires GPU for best performance
5	Jocher et al. [5]	YOLOv5	Lightweight and easy deployment for real-time use	Not originally a formal research paper
6	Szeliski [8], Hartley & Zisserman [9]	Monocular Vision Theory	Explained depth ambiguity and projection models	Cannot estimate depth directly
7	Hoiem et al. [18]	Geometric Context	Scene understanding from single image	Limited accuracy in complex scenes
8	Eigen et al. [19]	Deep Learning Depth Estimation	Deep Learning Depth Estimation	Requires large datasets and high computation

9	Scaramuzza et al. [11]	Visual Odometry	Motion-based depth estimation	Requires continuous camera movement
10	Leutenegger et al. [12]	Visual-Inertial SLAM	Combines vision and IMU for accuracy	Computationally expensive
11	Thrun et al. [15]	Sensor Fusion (Probabilistic Robotics)	Improved accuracy using multiple sensors	Improved accuracy using multiple sensors
12	OpenCV [20]	Camera Calibration	Converts pixel data to real-world measurements	Requires calibration and known parameters

III. ALGORITHM

The proposed system utilizes the YOLO (You Only Look Once) algorithm for real-time object detection due to its high speed and accuracy. YOLO is a deep learning-based object detection algorithm that processes an entire image in a single pass through a convolutional neural network, making it highly efficient for real-time applications.

Unlike traditional object detection methods that apply region proposals or multiple passes over an image, YOLO divides the input image into a grid structure. Each grid cell is responsible for predicting bounding boxes and corresponding class probabilities. These predictions include the coordinates of the bounding boxes, confidence scores, and object class labels. By performing detection in a single forward pass, YOLO significantly reduces processing time and enables real-time performance on mobile devices.

In this system, a pre-trained YOLOv5 model is used to detect objects from the live camera feed. The model identifies objects and generates

bounding boxes around them along with confidence scores. These bounding boxes are then used as the basis for further processing, such as depth estimation and dimension calculation.

YOLOv5 is particularly suitable for this application because of its optimized architecture, lightweight design, and ability to balance speed and accuracy. It supports deployment on resource-constrained devices through optimization techniques such as quantization and hardware acceleration.

Overall, the YOLO algorithm plays a crucial role in enabling fast, accurate, and real-time object detection, forming the foundation of the measurement system.

The proposed system follows a systematic algorithm to perform real-time object detection and measurement by combining image processing, machine learning, and sensor data analysis. The algorithm operates continuously to ensure dynamic updates and accurate results.

Step 1: Start

Initialize the system, activate the camera, and load the pre-trained YOLOv5 model for object detection.

Step 2: Capture Frame

Continuously capture frames from the mobile device camera in real time.

Step 3: Preprocess Image

Resize and normalize the captured frame to match the input requirements of the detection model.

Step 4: Object Detection

Pass the preprocessed image to the YOLOv5 model to detect objects and generate bounding boxes with confidence scores.

Step 5: Acquire Sensor Data

Simultaneously read accelerometer data (x, y, z values) from the device sensors.

Step 6: Process Sensor Data

Apply filtering techniques to reduce noise and compute the device orientation, specifically the pitch angle.

Step 7: Depth Estimation

Use geometric calculations along with device height and pitch angle to estimate the distance between the camera and the detected object.

Step 8: Dimension Calculation

Calculate the real-world dimensions (width, height, and area) of the detected object using the estimated distance and bounding box size.

Step 9: Display Results

Overlay the calculated measurements and bounding boxes onto the live camera feed.

Step 10: Continuous Update

Continuously update detection, sensor data, and measurement results as the device or object moves.

Step 11: End

Terminate the process when the application is closed.

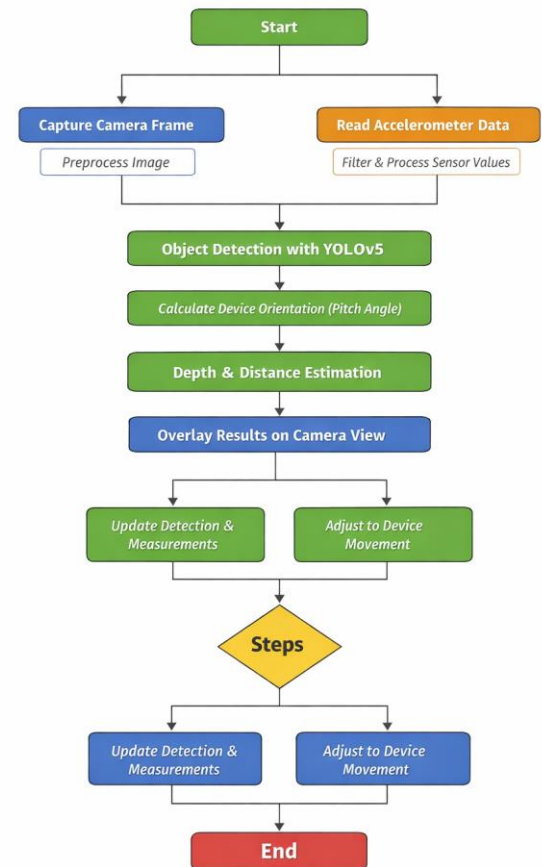


Figure III.2. Algorithm Diagram

IV. WORKING METHODOLOGY

- The working methodology of the proposed system is designed to perform real-time object detection and measurement by integrating computer vision techniques with sensor-based calculations. The system operates in a continuous loop, ensuring that detection, data processing, and measurement updates occur seamlessly as the user interacts with the mobile device.
- The process begins with the activation of the mobile camera, which captures live video frames in real time. Each frame is preprocessed to meet the input requirements of the object detection model. The preprocessed frame is then passed to the YOLOv5 model, which identifies objects within the scene and generates bounding boxes along with confidence scores. These bounding boxes define the region of interest for further measurement calculations.
- At the same time, the system continuously collects data from the device's

accelerometer. This sensor provides real-time values along the x, y, and z axes, which are used to determine the orientation of the device. The pitch angle is calculated from this data, and noise reduction techniques are applied to ensure stable and accurate readings. This orientation information is crucial for estimating the distance between the camera and the detected object.

- Once both object detection and orientation data are available, the system performs depth estimation using geometric relationships. The user inputs the device height through an interactive slider, which serves as a reference parameter for calculating the distance to the object. Based on this estimated distance and the size of the bounding box in the image, the system computes the real-world dimensions of the object, including its width, height, and area.
- The calculated measurements are then displayed on the screen through a transparent overlay on the live camera feed. As the user moves the device or changes the viewing angle, the system continuously updates the detection results, sensor data, and measurement values in real time. This ensures a smooth and responsive user experience without any noticeable delay.
- Overall, the working methodology combines real-time image processing, sensor integration, and efficient computation to deliver accurate and dynamic object measurement. The seamless interaction between these components makes the system practical, reliable, and suitable for real-world applications

Figure IV.1. System Diagram

V. RESULTS AND DISCUSSION

This section presents the execution and test results of the system, showcasing in actual performance and behaviour, during implementation.

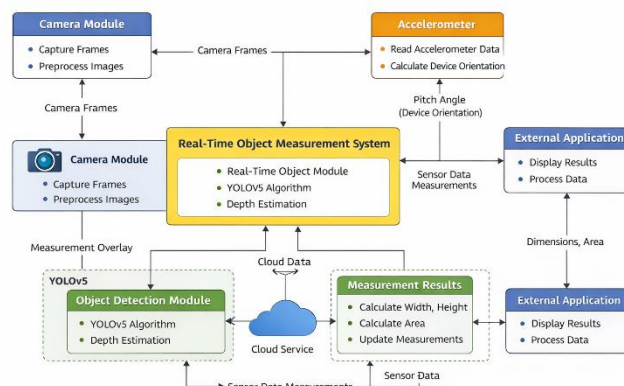
Testcases and Results:

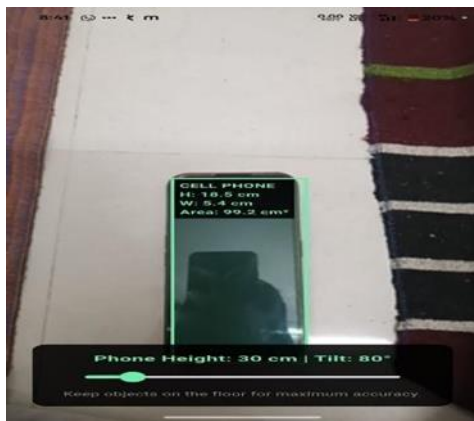


Cellphone

V.1 Cellphone at phone height (15 cm) and tilt: 81

The results demonstrate the effectiveness of the system in estimating real-world object dimensions using a single camera and sensor data. With the phone positioned at a height of 15 cm and a tilt angle of 81 degrees, the system successfully detected a cellphone and calculated its dimensions. The measured height of the object was approximately 15.1 cm, while the width was estimated to be 4.5 cm, resulting in a computed area of 67.8 cm².

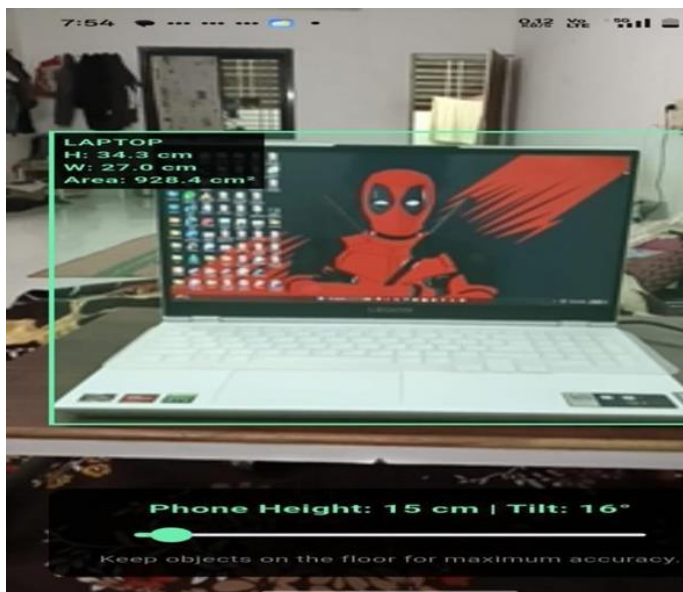




V.2 Cellphone at phone height (30 cm) and tilt : 80

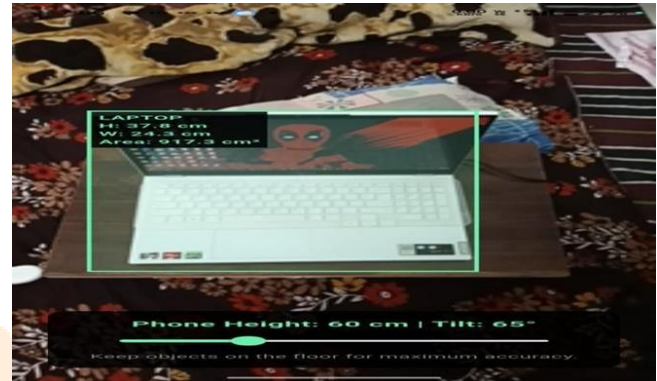
The results further validate the system's performance under a different configuration. With the phone positioned at a height of 30 cm and a tilt angle of 80 degrees, the system successfully detected the object and estimated its dimensions. The measured height was approximately 18.5 cm, and the width was calculated to be 5.4 cm, resulting in an area of 99.2 cm². While these values show a noticeable increase compared to the previous setup, this variation can be attributed to the change in camera height and viewing angle, which influences depth estimation and scaling. Factors such as perspective distortion, sensor inaccuracies, and slight variations in angle measurement may also contribute to the difference. Nevertheless, the system continues to provide consistent and reasonable estimations, demonstrating its ability to adapt to varying conditions maintaining overall reliability.

- Laptop



V.3 Laptop at phone height (15 cm) and tilt: 16

The results demonstrate the system's capability to measure larger objects with reasonable accuracy under different orientation conditions. With the phone positioned at a height of 15 cm and a tilt angle of 16 degrees, the system successfully detected a laptop and estimated its dimensions. The measured height of the object was approximately 34.3 cm, while the width was calculated to be 27 cm, resulting in a computed area of 928.4 cm².



V.4 Laptop at phone height (60 cm) and tilt: 65

The results further highlight the system's adaptability when measuring objects from a higher position and steeper angle. With the phone placed at a height of 60 cm and a tilt angle of 65 degrees, the system successfully detected the laptop and estimated its dimensions. The measured height was approximately 37.8 cm, while the width was calculated to be 24.3 cm, resulting in an area of 917.3 cm². Despite the increased camera height and change in viewing angle, the system continues to produce consistent and realistic measurements. Slight differences in values compared to other configurations can be attributed to perspective changes, depth estimation sensitivity, and minor sensor inaccuracies. Overall, the results demonstrate that the system maintains reliable performance even under varying heights and tilt angles.

.VI. CONCLUSION

The Mobile Vision-Based System for Real-Time Object Dimension Estimation presents a novel approach for real-time object dimension estimation using a monocular smartphone camera combined with sensor fusion techniques. It effectively addresses the fundamental limitation of monocular vision lack of depth perception by integrating computer vision methods with physical and mathematical modeling. By leveraging both visual data and sensor inputs, the system is able to estimate object dimensions in real time with reasonable accuracy and efficiency.

The key contributions of this work include the development of a lightweight and efficient system capable of real-time dimension estimation on mobile devices, eliminating the dependency on external reference objects such as rulers or markers. It integrates deep learning through the YOLOv5 model with sensor-based depth estimation, and applies trigonometric principles along with the pinhole camera model to convert pixel measurements into real-world units. Additionally, the system is designed with a modular architecture, making it suitable for mobile deployment and future scalability. Overall, this work demonstrates how the combination of fundamental physical concepts and modern machine learning techniques can provide practical and effective solutions to real-world measurement challenges.

The project achieved several important milestones during its development, demonstrating both technical effectiveness and practical applicability. One of the primary achievements is its real-time performance, as the system operates efficiently on mobile devices and provides instant measurement results without noticeable delay. Additionally, the solution maintains hardware independence, functioning effectively on standard smartphones without the need for specialized components such as LiDAR or dedicated depth sensors.

References

- [1]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
- [2]J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263–7271, 2017.
- [3]J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
- [4]A. Bochkovski, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
- [5]G. Jocher et al., "YOLOv5," GitHub Repository, 2020. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [6]TensorFlow Lite, "TensorFlow Lite Documentation," Google Developers, 2023. [Online]. Available: <https://www.tensorflow.org/lite>
- [7]Flutter, "Flutter Documentation," Google Developers, 2023. [Online]. Available: <https://flutter.dev>
- [8]R. Szeliski, Computer Vision: Algorithms and Applications, Springer, London, UK, 2010. Available online: Computer Vision: Algorithms and Applications – Springer
- [9]R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, Cambridge, 2004. Available online: Multiple View Geometry in Computer Vision – Cambridge University Press
- [10]E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, Upper Saddle River, NJ, 1998. Available online: Introductory Techniques for 3-D Computer Vision – Open Library
- [11] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I – The First 30 Years and Fundamentals," IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80–92, 2011.
- [12]S. Leutenegger, P. Furgale, V. Rabaud, et al., "Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization," The International Journal of Robotics Research, vol. 34, no. 3, pp. 314–334, 2015.

- [13]Google, “ARCore Overview,”Google Developers, 2023. [Online]. Available: <https://developers.google.com/ar>
- [14]Apple, “ARKit Overview,” Apple Developer Documentation, 2023. [Online]. Available: <https://developer.apple.com/augmented-reality/>
- [15]S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, MIT Press, Cambridge, MA, 2005. Available online: Probabilistic Robotics – MIT Press
- [16]I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016. Available online: Deep Learning – MIT Press
- [17]Sensors Plus Package, “sensors_plus Flutter Plugin,” pub.dev, 2023. [Online]. Available: https://pub.dev/packages/sensors_plus
- [18]D. Hoiem, A. A. Efros, and M. Hebert, “Geometric Context from a Single Image,” in Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV), Beijing, China, Oct. 2005, pp. 654–661, doi: 10.1109/ICCV.2005.107. Available online: <https://doi.org/10.1109/ICCV.2005.107>
- [19]D. Eigen, C. Puhrsch, and R. Fergus, “Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network,” Advances in Neural Information Processing Systems (NeurIPS), 2014.
- [20]OpenCV, “Camera Calibration and 3D Reconstruction,” OpenCV Documentation, 2023. [Online]. Available: <https://docs.opencv.org>

Author Information

Prof. Pratik Suresh Deshmukh, Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.

Mr. Yash Purohit, Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.

Mr. Ankush Kadukar Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.

Miss. Vedanti Khorgade, Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.

Mr. Amar Sawade, Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.

Mr. Aditya Nagtode, Department of Computer Science and Engineering, PRMIT&R Badnera, Amravati.