



Recent Advances in Science and Engineering

Web page info: <https://rase.yildiz.edu.tr>

DOI: 10.14744/rase.2025.0002



Research Article

Drones and slam: A grassroots approach to GPS-free navigation

Bora İŞILDAK^{1,*}, Mehmet KARABULUT²

¹Department of Physics, Faculty of Arts & Science, Yıldız Technical University, Istanbul, Türkiye

²Uskudar American High School, Istanbul, Türkiye

ARTICLE INFO

Article history

Received: 03 April 2025

Revised: 27 April 2025

Accepted: 06 May 2025

Key words:

Odometry, pose estimation, YOLO, SLAM.

ABSTRACT

In recent years, Unmanned Aerial Vehicles (UAVs) and drones have gained an important place in our lives by providing innovative solutions in a wide range of areas from defense industry, agriculture, logistics, and search and rescue operations. Despite advances in positioning technologies such as the Global Positioning System (GPS), autonomous navigation and reliable location estimation in confined spaces, dense vegetation, or areas with signals remains a major challenge.

This paper presents an innovative approach that combines You Only Look Once (YOLO) and Simultaneous Localization and Mapping (SLAM) algorithms to develop a GPS-independent positioning system. The vehicles beneath the drone are detected in Oriented Bounding Box (OBB) format using YOLO, and the estimated flight height is calculated using the known average wheelbase of the vehicles. This height information is integrated with SLAM algorithms to estimate the drone's position and speed.

The test results show that the method provides accurate estimates with an accuracy of ± 1 meter at heights between 30 m and 50 m and can work effectively in conditions where GPS is inadequate. Furthermore, YOLO's use of OBB improves object detection accuracy and provides a solution to scaling problems in SLAM algorithms. This innovative system offers significant potential for autonomous drone navigation and speed control in environments without GPS access, such as confined spaces.

Cite this article as: İşildak B, Karabulut M. Drones and slam: A grassroots approach to GPS-free navigation. Recent Adv Sci Eng 2025;1:15–21.

INTRODUCTION

In recent years, UAVs have been used in a wide range of applications from defense industry, agriculture, logistics, and search and rescue operations. With the expansion of these applications, the need for accurate and reliable position/speed determination has become even more important. Traditionally, UAVs have relied heavily on GPS data for position determination and navigation. However, in confined environments, tunnels, densely vegetated terrains or areas

with weak GPS signals, GPS-based location estimation may be insufficient. This situation poses a risk to both mission success and flight safety.

This paper aims to develop a visual odometry system for drone position and speed estimation without GPS data. In this camera-based-system, YOLO object detection algorithm and SLAM methods are combined to overcome scaling issues. This system offers an innovative solution that will enable autonomous navigation in environments such as confined spaces, tunnels, densely vegetated areas where

*Corresponding author.

*E-mail address: bora.isildak@yildiz.edu.tr



Published by Yıldız Technical University, İstanbul, Türkiye

This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

GPS signal is insufficient. “Can drones reliably estimate position and speed without GPS data using SLAM algorithms and visual information obtained from objects with known dimensions?”, we will address the following sub-objectives: detecting objects with known physical dimensions (e.g. vehicles) in OBB format with the YOLO algorithm, using the pixel dimensions of the detected objects to estimate the drone’s flight altitude, mitigating the scaling problems in SLAM algorithms with object-based altitude estimation, and testing the developed method in real flight scenarios and different environmental conditions.

Our goal is not only to estimate the drone’s height but also its three-dimensional motion, rotation, and position through SLAM algorithms to ensure accurate tracking in space. In addition, one of the main objectives is to develop a reliable solution for autonomous drone navigation and speed control that eliminates the dependency on GPS. For this purpose, the feasibility of an alternative system to GPS will be demonstrated through tests on camera images and algorithm integrations.

This paper aims to develop a visual odometry system that can estimate position and velocity using only monocular camera images without GPS data. Known object dimensions are a common method to obtain depth information for estimation with monocular cameras. Our approach is based on detecting vehicles on the ground in OBB format using the YOLO algorithm and converting the object dimensions into a drone height estimate. By integrating the calculated height information with SLAM algorithms, both the position and speed of the drone will be estimated more accurately. Thus, the aim is to achieve autonomous navigation and speed control without the need for any external markers and without GPS data. In addition, by testing the system under real flight conditions, it will be possible to compare it with similar applications in the literature and the advantages/shortcomings obtained.

In this context, the paper first analyzes the success level and limitations of similar methods through a literature review and then evaluates the effectiveness of YOLO and OBB-based object detection as an effective method to reduce the scale error in SLAM algorithms. Thus, an innovative and high accuracy solution is proposed to enable drones to fly without losing their position when GPS is disabled.

LITERATURE REVIEW

In the literature, it is known that Visual Odometry (VO) and SLAM methods are utilized for position estimation in environments where GPS is disabled or insufficient [1]. VO and SLAM algorithms can simultaneously extract the drone’s motion, orientation and environment map by detecting feature points on consecutive images from the camera [2]. VO can predict drone motion by feature matching between consecutive images. However, problems such as scale error can affect the accuracy of these methods [3]. On the other

hand, in recent years, deep learning-based object detection algorithms (e.g., YOLO) have enabled real-time and high-accuracy object recognition; objects such as vehicles, people, and buildings within the field of view of the drone camera can be detected [4]. The low latency and high accuracy rates of the YOLO algorithm offer unique advantages for real-time applications [5].

Scale problems of visual odometry algorithms are one of the most common challenges, especially in monocular cameras. Detailed investigations on this topic show that combining VO methods with SLAM algorithms provides effective solutions to such problems [6]. However, most existing studies additional markers or pre-mapped information, and a completely “non-landmark-based” system is less frequently. Approaches that scale issues can be mitigated with solutions based on object dimensions have improved the accuracy of SLAM algorithms [7,8].

METHODOLOGY

A. Research Method and Design

In this study, we adopt an experimental approach to estimate drone height from bird’s-eye photos using the deep learning-based YOLO algorithm. The study consists of two main phases:

1. **Data Collection and Preparation:** A data set was created using photographs taken from different heights.
2. **Software Development and Analysis:** Python programming language [9] and YOLO (yolov1x-obb) model were used to estimate the height of vehicles in photographs based on pixel width [10].

B. ORB-SLAM Algorithm

The ORB-SLAM algorithm basically consists of three parts. In the first part, feature points in the images are detected. These points are usually the corners of the objects in the image and are used to detect the change of position. These points are detected using the Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Features (BRIEF) algorithms.

In the second part, these detected feature points are matched with each other in consecutive images. In this way, the position of a point after a position change is determined and this is of great importance in understanding in which direction and how much this change is. This matching process relies on evaluating the Hamming distance between feature points.

In the third part, the matched points are put into the Perspective-n-Point (*PnP*) algorithm to calculate the change in position and thus determine the direction and amount of the position.

The algorithm proceeds through the following steps:

1. **Feature Detection:** Using the FAST and BRIEF algorithms, feature points in the images are detected. These

points are usually the corners of objects.

2. **Feature Matching:** Feature points detected in consecutive images are matched based on their Hamming distance, so that the position change can be performed using these points.
3. **Position Estimation:** Position determination is made by feeding the matched points into the *PnP* algorithm. It is determined how far in which direction.

However, if there is no information about the calibration of the camera or physical measurements within the field of view, this method can only calculate relative positions and movements. If the system is integrated with additional data sources such as GPS, absolute position can be determined.

C. Data Collection Process and Preparation

The photographs used in the study were obtained from various internet sources from different altitudes (30 m, 36 m, 45 m) were compiled as bird's eye view photographs. In the software used includes Python, YOLO, OpenCV and NumPy [11] has been used.

D. Image Processing and Model Processes

The photographs were analyzed by going through the specified steps:

Distortion Removal: The images were undistorted with the OpenCV library using camera parameters (focal length, optical center, distortion coefficients) [12].

Vehicle Detection (YOLO-OBB): Using YOLO's yolo-v11x-obb model, vehicles were labeled in OBB format. The model detected the pixel widths of each vehicle, and the extreme values were filtered out (Fig. 1.).

Width/Length Ratio: Reduced false positives by filtering out detections with very unusual ratios (e.g. various vehicles that do not match the aspect ratio of a passenger car).

Weighted Average with Confidence Level: When more than one vehicle is detected in an image, the pixel width of each vehicle is weighted by the confidence coefficient specified by the YOLO model and averaged.

Focal Calibration and Height Calculation: Based on images of known height, the relationship between the height of the drone H (in meters), image pixel size w_{pix} (in pixels) of the detected vehicle is obtained using the following relation.

$$H = f \frac{w_{actual}}{w_{pix}}$$

In this expression, f is the focal length of the camera in pixels. As the actual object size (w_{actual}), the average wheelbase of a passenger car of 1.85 m is considered.

With this method, a **height estimation** based only on the focal width of the camera and the pixel dimensions of the object is realized by detecting vehicles in bird's eye view photographs.

RESULTS

Height Estimation Success: Height estimates are accurate to within ± 1 meter, while the uncertainty above these values is 10% at 30 m. At altitudes of 50 m and above, the margin of error increased slightly due to the reduction in pixel size (Fig. 1).

Impact of using OBB: a. The OBB feature of the YOLO model allowed the correct corner coordinates to be deter-

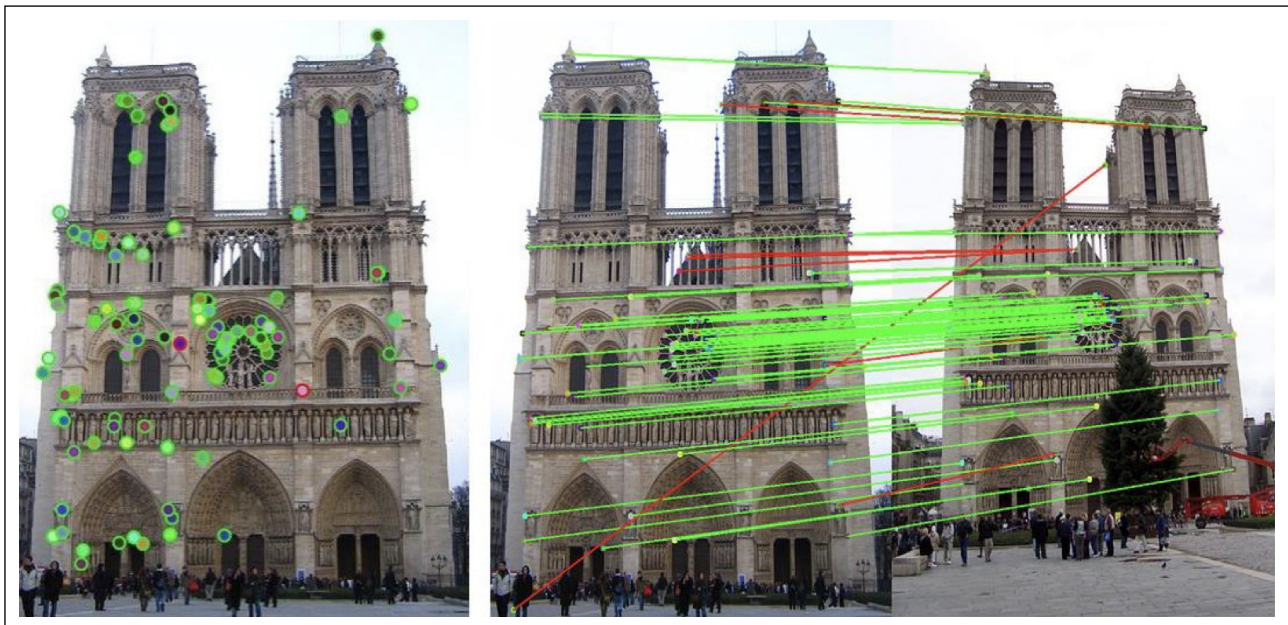


Figure 1. Feature detection and matching stages of the ORB-SLAM algorithm on Notre-Dame Cathedral. On the left, the feature points detected with the FAST algorithm, in the center the matching of these points in consecutive frames, and on the right the connecting lines showing the directions and distances between the matched points.

mined even when the vehicles were rotated, resulting in more stable results in pixel width calculation, b. Filtering by width/length ratio and confidence threshold reduced false positive detections (Figs. 2,3).

ORB-SLAM Integration: a. The ORB-SLAM algorithm's position estimation based on feature matching in consecutive frames provided position/motion information from a different perspective. The quality of the feature matching is effective in determining the drone's direction and amount of motion (rotation, translation), b. This method, which can track position without depending on GPS or reference marks, can be considered as an additional source of information that will partially reduce the uncertainty in height.

Code and Real-Time: The codes developed in Python, OpenCV and YOLO were able to process close to 10-15 frames per second (FPS) on a given hardware configuration.

In this study, drones were used to detect images of passenger cars in images acquired at three different heights (30 m, 36 m and 45 m). The median of the widths of the detected cars at each height was used to create the graph in Figure 4 and the data points were fitted to the curve $y=1.85 \cdot (f/x)$. The aim here is both to evaluate the compatibility of the obtained camera height values (blue) with the actual height values (black) and to make a validation by re-obtaining the approximate focal length of the camera, 1416 (px), based on the apparent width of the vehicles.

As can be seen in Figure 5, the results obtained with the proposed method are very consistent with the actual height values. Moreover, the focal length value obtained from the curve is 1416 pixels, which is the same as the actual value of focal length 1416. This result demonstrates the feasibility of the main objective of the study, which is to estimate the drone height without the use of GPS.

DISCUSSION

In this paper, a method combining a YOLO-based object detection model (yolov11x-obb) and camera calibration steps is developed to estimate drone height without GPS data. By detecting vehicles in bird's eye view photographs taken at different altitudes, the width information in pixels is obtained, and then the estimated height of the drone is calculated using the known vehicle wheelbase (1.85 m) and camera focal length. The results show that the method is applicable in various environments and at different height ranges (approximately 10 m - 50 m).

The advantages of the method we have developed. It shows the potential to estimate drone height with only monocular camera data in confined spaces or in environments where GPS signal is insufficient by reducing the dependency on GPS. It also provides easy applicability with existing camera and simple calibration processes, without requiring additional sensors, such as LiDAR (Laser Imaging Detection and Ranging) and barometer. YOLO's use of the OBB format accurately measures vehicles in different orientati-

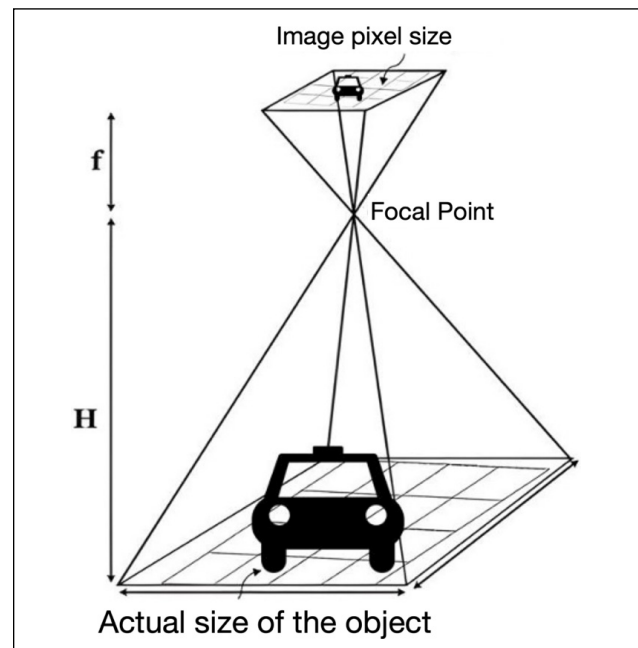


Figure 2. Schematic diagram illustrating the relationship between an object's actual size, its projected pixel size in the drone camera image, and the drone's flight height. The flight height of the drone (H) can be calculated based on the camera's focal length (f), the real-world dimensions of the detected object (e.g., the average wheelbase of a passenger car), and the object's observed pixel width (w_{pix}). Accurate focal length calibration and precise object detection are critical to minimizing scale errors and ensuring the robustness of the height estimation process.

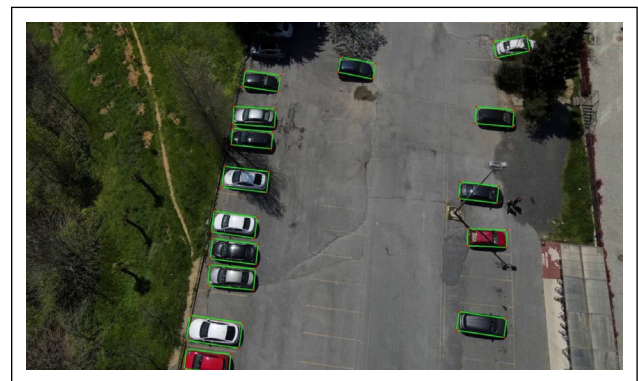


Figure 3. An example drone camera image showing multiple passenger cars detected using the YOLO-OBb model and enclosed in directional bounding boxes. Each vehicle is accurately identified, and its bounding box is rotated to align with the orientation of the vehicle on the ground, rather than being constrained to the horizontal and vertical axes. This alignment allows for more precise estimation of the true dimensions of the vehicles, which is critical for reliable altitude calculation.



Figure 4. Example images showing passenger cars being detected and enclosed in directional bounding boxes using the YOLO-OBB model. The figure illustrates how the algorithm not only identifies the vehicles but also captures their orientation by aligning the bounding boxes with the direction of the vehicles. This capability significantly improves the robustness of detection compared to traditional axis-aligned bounding boxes, especially in cases where the vehicles are rotated at various angles relative to the drone's field of view. The use of oriented bounding boxes ensures more precise width measurements, which is critical for accurate flight height estimation.

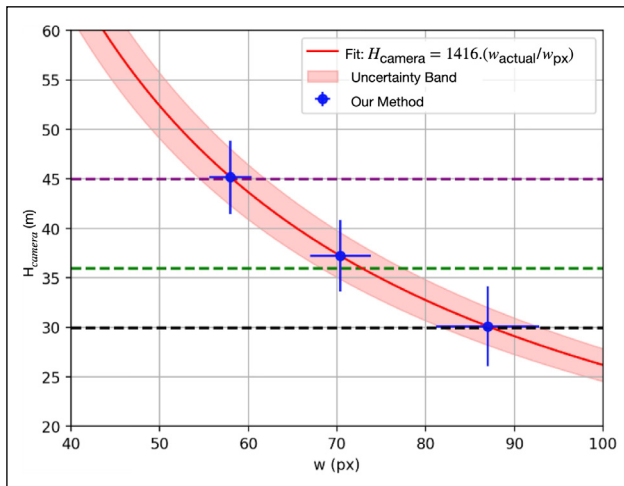


Figure 5. Graph illustrating the relationship between the drone's flight height and the visible width of detected vehicles in the camera image. The curve represents the inverse proportionality between the apparent pixel width of the vehicles and the drone's altitude, based on the calibrated focal length and known average vehicle dimensions. The actual flight heights of 30 meters, 36 meters, and 45 meters are indicated with horizontal dashed lines for reference. The strong agreement between the measured and actual heights validates the effectiveness of the proposed method for estimating altitude from monocular images without GPS input.

ons and reduces false detections, and OBB support is very important in achieving this.

There are also several limitations of our study. At altitudes above 50 m, the vehicle size becomes very small in pixels, which reduces the estimation accuracy. Lighting and weather conditions such as dense fog, heavy rain or low illumination conditions can reduce the object detection performance, that can lead to incorrect height estimations.

In our study, when the possibility of combining with ORB-SLAM is evaluated, it can be suggested to integrate VO and SLAM algorithms with this project since they can track the drone motion in 3D space even if they do not estimate the height directly. Reducing the "depth" and "scale" errors with additional image information (e.g. stereo camera or additional sensor) can increase the reliability of the method.

Improvements that can be made in the study can be categorized under three headings: a. More calibration points: Using different known elevation values can make the system more stable, b. Expanding the data set: The generalizability of the model can be increased with additional photos including day and night, different weather conditions and different vehicle types, c. Real-time application: With GPU optimization and faster CNN models, real-time (online) high altitude prediction on a drone can be achieved.

The photographs used in this study reflect height values at certain ranges. Collecting data over a wider range (5 m to 200 m) and in different environmental conditions (evening hours, cloudy weather, etc.) would contribute to a more comprehensive validation of the pixel width - actual height relationship and thus increase the diversity of data for different heights and environmental conditions.

YOLO-based object detection models are more successful in daylight and high contrast images. It is important to train the model for night and low illumination conditions. To improve the performance of the model in adverse conditions such as low light, fog or rain, additional data sets should be created and different sensor technologies such as infrared cameras or heat cameras should be investigated.

In our study, only one altitude value (25 m) was used as a reference to calibrate the focal length. Using calibration points at different altitudes (10 m, 25 m, 40 m, etc.) can further reduce scale errors and improve accuracy at different altitudes. Using more calibration points would be valuable in this respect.

The ORB-SLAM algorithm can extract the drone's position and motion information through feature matching and *PnP* calculations. A SLAM system that is fully integrated with the altitude estimation module can provide a fully autonomous navigation solution that accurately provides not only altitude, but also horizontal position and rotation information. This can be achieved with full integration with ORB-SLAM.

Our results show the importance of real-time (online) application and hardware optimization. Optimization techniques such as *model compression*, *quantization* or *TensorRT* should be investigated for real-time execution of code developed on Python/OpenCV on embedded platforms (e.g. NVIDIA Jetson, Raspberry Pi). When high speed in real-time (≥ 20 FPS) is achieved, it will be possible for drones to provide instant altitude feedback during flight.

In this work, the focus is on estimating the height of vehicles based on their pixel width. Buildings, road signs or objects of different sizes can be used in a similar way. Determining the dimensions of different objects can help to provide a more robust basis for estimation by providing many reference points. This can be achieved by diversifying the dataset and using objects other than vehicles.

The camera-based approach we used in our study is particularly suitable for high altitude and adverse may be limited in weather conditions. By fusing data with additional sensors such as barometer, IMU (Inertial Measurement Unit), LiDAR or radar, the accuracy of drone height and position information can be significantly improved. This can be achieved with the integration of embedded sensors.

CONCLUSION

Testing the method developed in our study in autonomous drone projects in industrial environments (e.g. cargo delivery, mapping, agricultural spraying) or in different academic studies may provide the opportunity to test the validity of the method in field conditions and to benefit from larger data sets. Achieving this with industrial and academic cooperation will make a great contribution to our lives. As future work, further improvements could focus on expanding the range of detectable objects, optimizing the system for real-time applications on embedded hardware, and integrating additional sensor data such as barometric pressure or LiDAR measurements to enhance accuracy under diverse environmental conditions.

AUTHORSHIP CONTRIBUTIONS

Concept: Mehmet Karabulut Design: Mehmet Karabulut, Supervision: Bora Isildak, Data: Mehmet Karabulut, Analysis: Mehmet Karabulut, Literature Search: Mehmet Karabulut, Writing: Bora Isildak, Mehmet Karabulut.

DATA AVAILABILITY STATEMENT

The published publication includes all graphics and data collected or developed during the study.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

- [1] Mur-Artal R, Tardós JD. ORB-SLAM: A versatile and monocular SLAM system. *IEEE Trans Robot* 2015;31:1163. [\[CrossRef\]](#)
- [2] Hartley R, Zisserman A. *Multiple View Geometry in Computer*. 2nd ed. Cambridge: Cambridge Univ Press; 2003. [\[CrossRef\]](#)
- [3] Scaramuzza D, Fraundorfer F. Visual odometry [tutorial survey]. *IEEE Robot Autom Mag* 2011;18:80–92. [\[CrossRef\]](#)
- [4] Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed accuracy of object detection. *arXiv*. <https://doi.org/10.48550/arXiv.2004.10934>.
- [5] Redmon J, Divvala S, Girshick R, Farhadi A. You

- only look once: Unified, real-time object detection. Proc IEEE Conf Comput Vis Pattern Recognit CVPR 2016:779–788. [\[CrossRef\]](#)
- [6] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. IEEE ACM Int Symp Mixed Augment Real 2007:225–234. [\[CrossRef\]](#)
- [7] Tateno K, Tombari F, Laina I, Navab N. CNN-SLAM: Dense monocular SLAM with learned depth prediction. Proc IEEE Conf Comput Vis Pattern Recognit CVPR 2017:6565–6574. [\[CrossRef\]](#)
- [8] Liu C, Yuen J, Torralba A. SIFT flow: Dense correspondence scenes and its applications. IEEE Trans Pattern Anal Mach Intell 2010;33:978–994. [\[CrossRef\]](#)
- [9] Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace 2009. [\[CrossRef\]](#)
- [10] Ultralytics. YOLO documentation. Available at: <https://docs.ultralytics.com> Accessed May 14, 2025. [\[CrossRef\]](#)
- [11] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming NumPy. Nature 2020;585:357–362. [\[CrossRef\]](#)
- [12] OpenCV Documentation. Camera calibration and 3D reconstruction. Available at: https://docs.opencv.org/4.x/d9/d0c/group_calib3d.html Accessed May 14, 2025.