

Monocular Object Distance Estimation Using Calibration-Based Perspective Mapping and Detection Integration

Sowndappan S¹, Mythili S², Priya P³, Dr. P. Sachidhanandam⁴, Pavithra U⁵, Aarthi R S⁶

¹Sowndappan S: Student, Dept. of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

²Mythili S: Student, Dept. of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

³Priya P: Assistant Professor, Dept. of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

⁴Dr. P. Sachidhanandam: Head of Department, Department of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

⁵Pavithra U: Student, Dept. of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

⁶Aarthi R S: Student, Dept. of Information Technology, Knowledge Institute of Technology, Tamil Nadu, India

Abstract - Accurate object distance estimation using monocular cameras is challenging due to the absence of explicit depth information and the reliance on specialized hardware such as stereo vision or LiDAR. This paper presents a lightweight calibration-based framework for estimating object distance using a single monocular camera. The proposed method uses camera height and tilt angle to construct a perspective grid that maps image coordinates to real-world ground distance intervals. Detected objects are localized using a detection model, and the bottom pixel of each bounding box is projected onto the calibrated grid to estimate distance. Unlike dense depth estimation methods, the proposed approach performs object-level distance estimation without requiring large training datasets or high computational resources. The method is object-agnostic and can be integrated with different detection models. Experimental results show that the system achieves 100% interval accuracy and a mean absolute error of approximately 0.36 m over a distance range of 5 m to 15 m, excluding the near-field blind region. The system operates at approximately 5 frames per second on CPU-only hardware, demonstrating its suitability for real-time, low-cost monitoring applications.

Key Words: Monocular distance estimation, camera calibration, perspective grid mapping, object detection, geometric modeling, real-time systems, range-based estimation

1. INTRODUCTION

Accurate distance estimation is a fundamental requirement in many computer vision applications, including surveillance, autonomous navigation, robotics, and environmental monitoring. While humans naturally perceive depth through binocular vision, enabling machines to estimate distances using visual input remains a challenging problem, particularly when relying on a single monocular camera. Unlike stereo vision systems, monocular setups do not provide explicit depth cues, making distance estimation an inherently ill-posed problem.

Traditional approaches for distance estimation often rely on specialized hardware such as stereo cameras, LiDAR sensors, or depth cameras. Although these systems can achieve high accuracy, they introduce significant limitations in terms of cost, power consumption, and deployment complexity. In many real-world scenarios—such as large-scale surveillance systems or resource-constrained environments—these requirements make such solutions impractical. As a result, there is increasing interest in developing computationally efficient, monocular vision-based alternatives that can estimate object distance without additional hardware.

Recent advances in deep learning have led to significant progress in monocular depth estimation, where convolutional neural networks are trained to predict dense depth maps from single images. Methods such as MonoDepth2 and DORN have demonstrated impressive performance on benchmark datasets. However, these approaches require large-scale annotated datasets, high computational resources, and often produce dense depth outputs that are unnecessary for applications focused on object-level distance estimation. Moreover, their deployment on edge devices or CPU-only systems remains challenging due to their computational complexity.

To address these challenges, this paper proposes a calibration-based monocular object distance estimation framework that combines geometric modeling with object detection. The proposed method utilizes camera parameters such as height and tilt angle to construct a perspective grid that maps image space to real-world ground distances. Detected objects are localized using a detection model, and their positions are projected onto the calibrated grid to estimate their distance from the camera.

2. RELATED WORK

Monocular distance estimation has been extensively studied in computer vision and can broadly be categorized into three main approaches: deep learning-based depth estimation, geometry-based methods, and object-based distance estimation techniques.

2.1 Deep Learning-Based Monocular Depth Estimation

Recent advancements in deep learning have significantly improved the performance of monocular depth estimation. These methods aim to predict dense depth maps from single images using convolutional neural networks. Notable approaches such as MonoDepth2 utilize self-supervised learning techniques to estimate depth without requiring ground truth annotations, while DORN formulates depth estimation as an ordinal regression problem to improve accuracy. Although these models achieve high performance on benchmark datasets such as KITTI and NYU Depth, they present several limitations including large-scale dataset requirements, significant computational resources, and deployment challenges on resource-constrained devices.

2.2 Geometry-Based Distance Estimation

Geometry-based approaches rely on the principles of camera projection and calibration to estimate distances. Using the pinhole camera model, it is possible to relate image coordinates to real-world measurements when camera parameters such as focal length, height, and tilt angle are known. These methods are computationally efficient and do not require training data, making them suitable for real-time applications. However, many existing approaches are either limited to specific scenarios or lack robustness due to simplified assumptions.

2.3 Object-Based Distance Estimation Approaches

Another category of methods focuses on estimating distance using object-level features such as bounding box size, pixel location, or known object dimensions. While these methods are simple and easy to implement, they often suffer from limited accuracy and poor generalization. Many rely on assumptions about object size or require prior knowledge of object dimensions, which restricts their applicability across different object categories.

2.4 Research Gap

From the above discussion, it is evident that existing approaches exhibit a trade-off between accuracy, computational complexity, and practical applicability. There is a clear need for a lightweight, calibration-based framework that integrates object detection with geometric modeling to provide accurate object-level distance estimation without requiring additional hardware or extensive training data.

2.5 Positioning of the Proposed Work

The proposed method addresses this gap by introducing a calibration-based monocular distance estimation framework that integrates object detection with perspective grid mapping. Unlike deep learning-based depth estimation

methods such as MonoDepth2 and DORN, the proposed approach does not require depth training data and operates with significantly lower computational overhead.

3. PROPOSED METHODOLOGY

3.1 System Overview

The proposed framework estimates the distance of detected objects from a monocular camera using a calibration-based geometric approach. The system integrates object detection with perspective-based distance mapping, enabling efficient object-level distance estimation without requiring additional sensors or dense depth prediction.

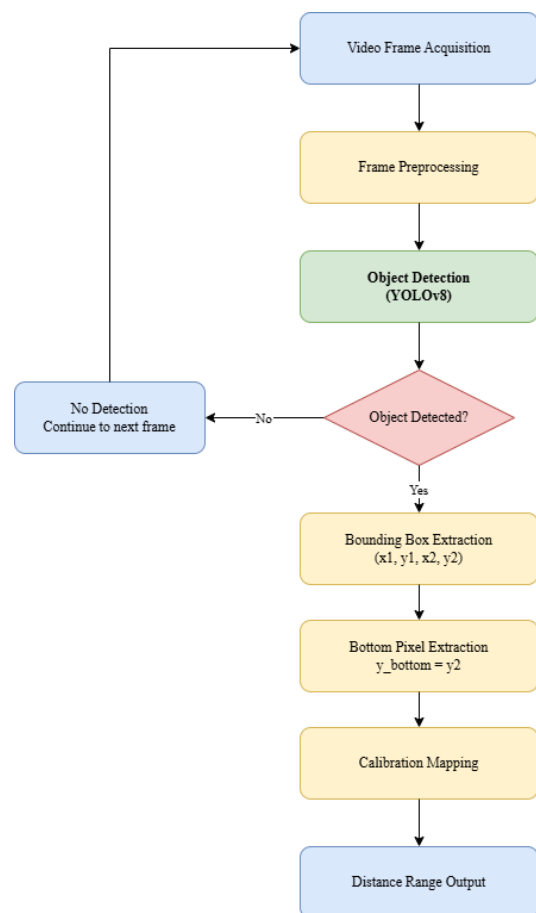


Fig -1: Overall processing pipeline of the proposed system

3.2 Object Detection and Localization

Objects in the scene are localized using a bounding-box-based detection model. For each detected object, the bounding box is defined as $B = (x_{min}, x_{max}, y_{min}, y_{max})$. The reference point used for distance estimation is the bottom-center of the bounding box, defined as $(x_c, y_b) = ((x_{min} + x_{max})/2, y_{max})$, where x_c represents the horizontal center and y_b represents the bottom pixel coordinate. The bottom pixel

is selected because it approximates the point of contact between the object and the ground plane.

3.3 Camera Calibration and Assumptions

The proposed method relies on the following known camera parameters:

- H: Camera height from the ground plane
- θ : Camera tilt angle relative to the horizontal axis
- f: Effective focal length of the camera
- y_c : Vertical center of the image

The method assumes a planar ground surface, fixed camera position and orientation, and that objects are in contact with the ground plane. These assumptions simplify the geometric modeling and are valid for many surveillance and monitoring applications.

3.4 Geometric Distance Estimation

The core of the proposed method is the transformation of image coordinates into real-world ground distances using perspective geometry. The vertical position of the detected object in the image is converted into an angular deviation: $\alpha = \arctan((y_b - y_c) / f)$. Using camera height and tilt angle, the ground distance D is computed as: $D = H / \tan(\theta + \alpha)$.

To improve computational efficiency, the continuous distance space is discretized into predefined intervals using a perspective grid. During runtime, the bottom pixel coordinate y_b is compared against grid boundaries and the corresponding distance interval is assigned, enabling O(1) constant-time distance estimation.

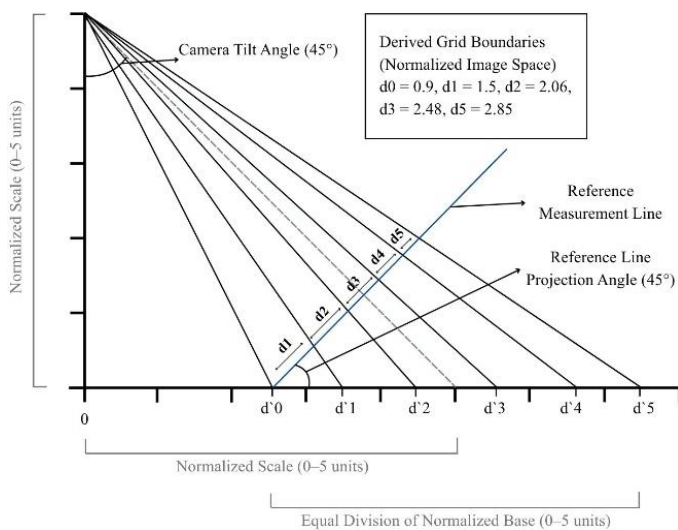


Fig -2: Geometric construction of perspective grid boundaries

Mapping from Uniform Image-Space Divisions to Projected Grid Boundaries

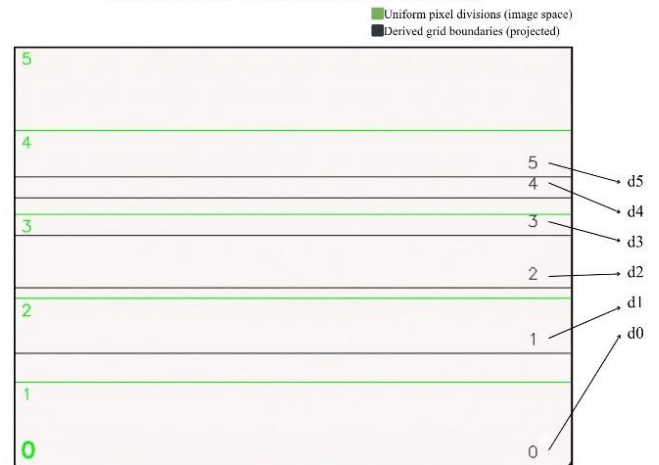


Fig -3: Mapping from uniform image-space divisions to projected grid boundaries

3.5 Calibration Procedure and Lookup Table

The perspective grid boundaries are computed during an offline calibration stage. Camera height H and tilt angle θ are measured, ground distances are selected as reference points, corresponding image positions are computed via geometric projection, and boundary values are stored in a lookup table. This design significantly reduces computational overhead and ensures consistent performance across frames.

3.6 Computational Efficiency

The proposed method achieves O(1) distance estimation complexity per object. It requires no depth model training, introduces minimal runtime overhead beyond object detection, and enables real-time operation on CPU-only hardware.

4. SYSTEM IMPLEMENTATION

4.1 Implementation Environment

The proposed system was implemented using Python with OpenCV for video acquisition and preprocessing, Ultralytics YOLOv8 for object detection, and NumPy for numerical computations. The implementation was designed to operate efficiently on CPU-only hardware.

4.2 Processing Pipeline

For every input frame: (1) the image is passed to the object detection module, (2) detected bounding boxes are extracted, (3) the bottom pixel of each bounding box is identified, (4) the pixel is mapped to the calibrated perspective grid, and (5) the corresponding distance interval is assigned. This pipeline enables continuous monitoring with minimal latency.

4.3 Object Detection Integration

Object detection is performed using a YOLOv8-based model trained on a custom dataset. The distance estimation component remains independent of the detection model and can be integrated with any object detector. This modular design allows flexible deployment across different application domains.

4.4 Real-Time System Integration

The complete system integrates object detection and geometric distance estimation into a unified real-time pipeline. The lightweight nature of the proposed method allows it to operate on CPU-only hardware without GPU acceleration. The system includes an alert mechanism that notifies users when an object is detected along with its estimated distance.

5. EXPERIMENTAL SETUP

5.1 Hardware Configuration

The proposed system was evaluated on a CPU-only setup consisting of an AMD Ryzen 7 5700U processor with 8 GB RAM. No dedicated GPU was used during testing. The system achieved an average inference latency of approximately 190 ms per frame (≈ 5 fps).

5.2 Camera Configuration

A monocular camera with a resolution of 480×640 pixels was used. The camera was mounted at a height of 10 m above the ground and oriented at a downward tilt angle of approximately 45° . Under this configuration, the observable ground region extends from approximately 5 m to 15 m, with a near-field blind zone (0–5 m) due to field-of-view constraints. The camera configuration used for calibration is illustrated in Fig. 4.

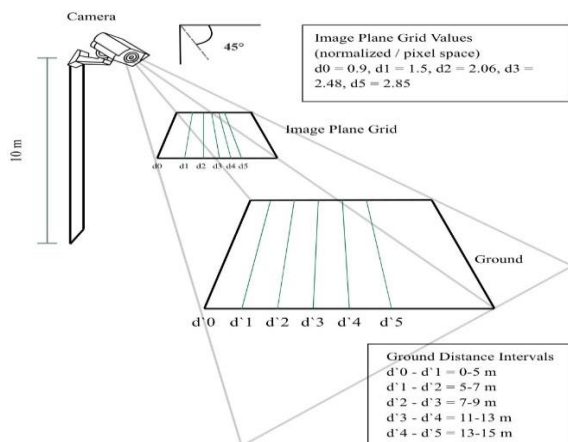


Fig -4. Camera setup showing height and tilt angle used for calibration and distance estimation.

5.3 Distance Segmentation

Using the proposed calibration framework, the observable region was discretized into the following predefined distance intervals:

- 5–7 m
- 7–9 m
- 9–11 m
- 11–13 m
- 13–15 m

5.4 Evaluation Protocol

To evaluate performance, 50 test samples were generated across the valid operating range (5–15 m). Evaluation metrics include: Interval Accuracy (whether the predicted interval contains the ground truth) and Mean Absolute Error ($MAE = (1/n) \sum |D_{actual} - D_{predicted}|$), where $D_{predicted}$ is the midpoint of the assigned interval.

6. RESULTS AND EVALUATION

6.1 Object Detection Performance

The object detection module provides spatial localization of objects within the scene. Detected bounding boxes are used to extract the bottom pixel coordinate required for geometric mapping. The proposed framework is detector-agnostic and can be integrated with any object detection model.



Fig -5. Sample system output showing detected object and estimated distance range overlaid on the camera frame.

6.2 Distance Estimation Results

The proposed method was evaluated across distances ranging from 5 m to 15 m, excluding the near-field blind region. A total of 50 samples were tested across all defined intervals. Table 1 presents a representative subset of the results.

Table -1. Sample Distance Estimation Results

Actual Distance (m)	Predicted Range (m)	Midpoint (m)	Error (m)
5.4	5-7	6	0.6
6.5	5-7	6	0.5
5.5	5-7	6	0.5
6.0	5-7	6	0.0
7.5	7-9	8	0.5
8.5	7-9	8	0.5
9.5	9-11	10	0.5
10.0	9-11	10	0.0
11.5	11-13	12	0.5
14.0	13-15	14	0.0

The results show that all 50 samples were assigned to the correct distance interval. Future work will extend evaluation to benchmark datasets such as KITTI and NYU Depth.

6.3 Evaluation Metrics

Interval Accuracy: Accuracy = Correct Predictions / Total Samples = 50/50 = 100%

Mean Absolute Error (MAE): MAE ≈ 0.36 m (over 50 samples)

6.4 Comparison with Existing Methods

Table -2. Comparison with Existing Distance Estimation Methods

Method	Hardware	Output Type	Runtime	Characteristics
Stereo Vision	Stereo camera	Dense depth	Real-time	Accurate, hardware dependent

LiDAR	LiDAR sensor	Precise depth	Real-time	Very high cost
MonoDepth2	Monocular	Dense depth	GPU	High computational cost
Heuristic methods	Monocular	Object distance	Real-time	Limited robustness
Proposed	Monocular	Range-based distance	~5 fps CPU	Lightweight, calibration-based

6.5 Real-Time Performance

The system operates at approximately 5 fps (190 ms per frame) on CPU-only hardware. The distance estimation module introduces negligible computational overhead due to its constant-time lookup-based implementation. The majority of computational cost is attributed to the object detection stage.

6.6 Limitations

- Near-field blind region (0-5 m) due to camera configuration
- Discretization error caused by fixed interval width
- Assumption of planar ground surface
- Performance dependency on object detection quality

7. DISCUSSION

The experimental results demonstrate that the proposed calibration-based monocular framework provides reliable range-based object distance estimation using only a single camera and minimal geometric parameters. The system achieves 100% interval accuracy across 50 test samples, indicating that the perspective grid mapping consistently assigns objects to the correct distance ranges.

The observed MAE of approximately 0.36 m reflects the discretization of the distance space into fixed intervals. Since each interval spans 2 m, the midpoint approximation introduces inherent estimation error even when the predicted interval is correct. Reducing the interval width would proportionally decrease the MAE, at the cost of increased calibration complexity.

A key strength of the proposed method is its computational efficiency, operating via constant-time lookup with negligible overhead beyond object detection, running at 5 fps on CPU-only hardware. The distance estimation framework is also model-independent, relying solely on geometric

relationships without dependency on object-specific features.

However, several limitations must be acknowledged: the assumption of a planar ground surface, the near-field blind region (0–5 m), fixed distance interval discretization, controlled evaluation conditions, and partial dependency on detection quality. Despite these limitations, the proposed method offers a strong balance between accuracy, efficiency, and deployability.

8. CONCLUSION

This paper presented a calibration-based monocular framework for object distance estimation using a single camera. The proposed method leverages camera height and tilt angle to construct a perspective grid that maps image coordinates to real-world ground distance intervals. By associating the bottom pixel of detected object bounding boxes with this calibrated grid, the system enables efficient object-level distance estimation without requiring additional sensors or dense depth prediction.

Experimental results demonstrated an MAE of 0.36 m over distances from 5 m to 15 m with 100% interval accuracy, operating at approximately 5 fps on CPU-only hardware. Unlike deep learning-based methods such as MonoDepth2, the proposed approach does not require large-scale depth datasets or GPU acceleration, offering a computationally efficient and scalable alternative.

Future work will focus on extending the framework to handle non-planar environments, improving robustness under varying lighting and occlusion conditions, and developing adaptive calibration techniques. Additionally, integrating more advanced detection models and evaluating across diverse real-world scenarios will further enhance applicability.

REFERENCES

- [1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1–3, pp. 7–42, Apr. 2002.
- [2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [3] J. Levinson et al., "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Veh. Symp.*, Baden-Baden, Germany, Jun. 2011, pp. 163–168.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
- [5] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 270–279.
- [6] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1851–1858.
- [7] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. M. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, Jul. 2022.
- [8] Y. Kuznetsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6647–6655.
- [9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
- [10] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (ToF) cameras: A survey," *IEEE Sensors J.*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011.
- [11] D. Eigen, C. Puhersch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 27, Montreal, QC, Canada, Dec. 2014.
- [12] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.
- [13] H. Liang, X. Zhang, Y. Wang, and J. Li, "Self-supervised object distance estimation using a monocular camera," *Sensors*, vol. 22, no. 8, p. 2936, Apr. 2022.
- [14] S. Lee, J. Kim, H. Yoon, J. Shin, and H. Kim, "Vehicle distance estimation from a monocular camera for advanced driver assistance systems," *Symmetry*, vol. 14, no. 12, p. 2657, Dec. 2022.
- [15] M. Rezaei and R. Klette, "Computer vision for driver assistance: Simultaneous traffic and driver monitoring," *Springer Tracts Adv. Robot.*, vol. 145, 2017.
- [16] S. Gasparini, P. Sturm, and J. P. Barreto, "Plane-based calibration of central catadioptric cameras," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[17] Y. Yu and K. Hasegawa, "Relative distance estimation using monocular camera," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 14, no. 6, pp. 714–721, 2010.

[18] M. A. Raza, A. Khattak, and H. Ali, "Framework for estimating distance and dimension of pedestrians in real-time using monovision," *Neurocomputing*, vol. 275, pp. 2572–2585, Jan. 2018.

[19] M. Habibi, M. M. Nikkhah, and A. Aghdam, "Distance estimation between moving objects using monocular vision," in *AIP Conf. Proc.*, vol. 2591, no. 1, p. 080019, 2023.

[20] B. U. Toreyin, Y. Dedeoglu, U. Gudukbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognit. Lett.*, vol. 27, no. 1, pp. 49–58, Jan. 2006.

[21] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.

[22] G. Xu, Y. Zhang, Q. Zhang, G. Lin, and J. Wang, "Deep domain adaptation based video smoke detection using synthetic smoke images," *Fire Saf. J.*, vol. 93, pp. 53–59, Oct. 2017.

[23] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan.2023.[Online].Available:<https://github.com/ultralytics/ultralytics>

[24] F. Amzajerian et al., "Lidar systems for precision navigation," *NASA Tech. Rep.*, 2011.