

# Urban Landscape Mapping: Estimating Building Coordinates Using Smartphone Camera and Geospatial AI Techniques

UrbanX Technologies, Inc.

## Abstract

Accurate localization of buildings is an important task in urban planning and development, impacting areas such as emergency response, navigation, and augmented reality. This research presents several advancements in urban building mapping using moving smartphone camera images, crucial for maintaining precise geographic information systems. We created a comprehensive dataset, comprising more than 7,000 instances of diverse building types, including low-rise residence, apartment, high-rise residence for the detection and classification of urban buildings. Utilizing YOLOv8, a highly accurate building detection model was developed, achieving an mAP of 76%. This work introduces an innovative integration of AI algorithms with geospatial data and photogrammetry techniques to accurately localize buildings from smartphone camera images. The precision of our building localization method is evaluated, with an error margin of up to 5.22 meters. Moreover, by identifying and analyzing the sources of errors in the localization process, this study provides insights into potential areas for improvement, setting the stage for future enhancements in urban mapping technologies. As well as providing support for urban planning and emergency services, we believe that this research has implications for delivery services, disaster management, and preservation efforts, indicating a significant impact on both societal and technological domains.

**Keywords:** building detection, building id, urban monitoring, YOLOv8

## 1 Introduction

Accurately building localization is an important aspect of urban planning and development. It holds importance in societal and technological domains, including emergency response, urban planning, navigation systems and augmented reality applications. Given the growth of areas, efficiently mapping and updating building coordinates is crucial, for maintaining precise geographic information systems (GIS) that are

utilized by governments, businesses and individuals.

The building locations data not only supports urban planning but also aids emergency services in responding promptly. Additionally it plays a role, in the development of city initiatives. Furthermore precise building maps can improve delivery services, assist in disaster management efforts and contribute to preservation endeavors.

Recent studies have primarily focused on

using static surveillance cameras or aerial imagery for urban mapping. However, these methods are often constrained by the limited field of view and the inability to capture real-time changes in urban landscapes. For instance, aerial imagery based studies, like the one conducted in [1], provided broad overviews but were limited in terms of real-time data acquisition.

To overcome these limitations, our research presents a novel approach that utilizes moving smartphone cameras, leveraging the ubiquity and mobility of these devices. This method aligns with the recent trend of using moving cameras for dynamic data acquisition as seen in studies such as [2], where vehicle-mounted cameras were used for traffic flow analysis.

This research makes several key contributions to the field of urban building mapping:

- **A Large-Scale Building Detection Dataset:** We developed a comprehensive dataset consisting of more than 7,000 instances of buildings in Tokyo metropolitan area, focused on three distinct types of buildings: low-rise residence, apartment, and high-rise residence.
- **Building Detection Model:** Utilizing the advanced capabilities of YOLOv8, we trained a model that achieves a 76% mAP in building detection.
- **Integration of AI with Geospatial and Photogrammetry Techniques:** Our approach combines AI algorithms with geospatial data

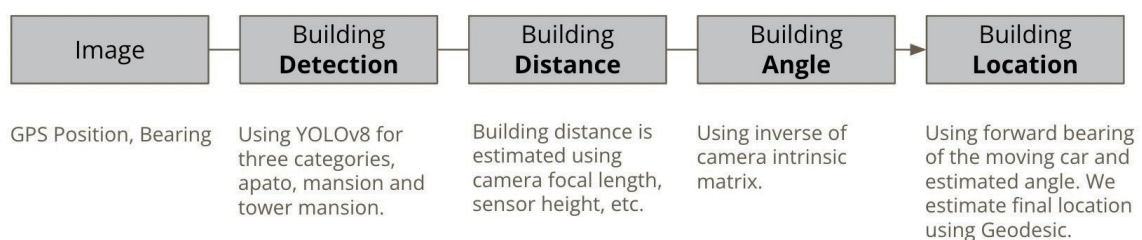
and photogrammetry to localize buildings accurately. This methodology represents an advancement in the precision of urban mapping technologies.

- **Evaluation of Localization Accuracy:** The accuracy of our building localization method is rigorously evaluated, showing an error margin of up to 5.22 meters, which is crucial for understanding the reliability and potential applications of our approach.
- **Error Sources in Localization:** By analyzing the sources of errors in our localization process, we provide insights into potential areas for improvement and pave the way for future enhancements in urban mapping techniques.

Our methodology not only allows for real-time urban mapping but also addresses some of the limitations faced by previous research. Unlike static camera methods, our approach captures the dynamic changes in urban environments. Moreover, by analyzing and identifying the sources of errors in building localization, this study provides insights for future enhancements in urban mapping technologies.

## 2 Methodology

We show the overall framework for building localization in Figure 1, which has components, such as building detection, building distance estimation, building angle estimation and building localization. In the next few sub-sections, we describe each component in detail.



**Fig. 1:** Illustration of building detection and localization using a moving smartphone camera.



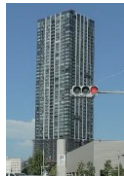
### 2.1 Building Detection Dataset

For the localization of individual buildings, it is necessary to detect them first. There are several ways of building detection, such as using object detection method, instance segmentation, semantic segmentation and panoptic segmentation, etc. Out of all these methods, we selected, object detection method to detect buildings as rectangular bounding boxes. We select this method due to the simplicity of the development of the dataset annotation and

resources required for training and computation.

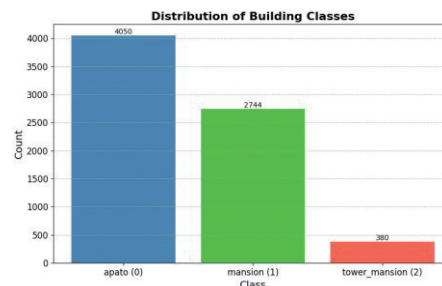
We carry out a driving experiment in Tokyo metropolitan area and record videos using smartphone mounted on the dashboard of a car. We manually annotate buildings in the extracted frames from the video by drawing rectangular bounding boxes and categorize them into three categories based on their height, such as low-rise buildings (up to four floors), mid-rise building (four to fifteen floors) and high-rise buildings (above fifteen floors), as shown in Table 1.

**Table 1:** Building categories for annotation and approximate height characteristics

Type of Building	Number of Stories (approx.)	Example image
Low-rise residence	Up to four	
Apartment	Five to Fifteen	
High-rise residence	Above twelve	



(a) Example annotation of three classes of buildings



(b) Distribution of building classes in the training set

**Fig. 2:** Building annotation example and distribution in the training dataset

In total, we annotate 7,174 buildings for training and 304 buildings for validation. An example annotation along with the distribution of the three types of buildings are shown in Figure 2. We use this dataset for training building detection model.

## 2.2 Training Building Detection Model

We use the building detection dataset consisting of 7,147 buildings to train an object detection network YOLOv8 [3]. YOLOv8 is the latest addition to the YOLO (You Only Look Once) series of real-time object detectors, offering remarkable performance. For improved feature extraction and object detection performance, YOLOv8 features advanced backbone and neck architectures.

In contrast to anchor-based approaches, YOLOv8

utilizes a split Ultralytics head that is anchorfree, enhancing detection accuracy and efficiency. It optimizes the accuracy-speed tradeoff, ensuring suitability for real-time object detection in diverse areas. YOLOv8 also offers a variety of pre-trained models, catering to different tasks and performance requirements.

YOLOv8l, one of the larger models in the YOLOv8 series, is particularly noteworthy for its performance metrics. It achieves a balance between speed and accuracy, standing out as a preferred choice for real-time detection tasks. We use YOLOv8l model using the initial pre-trained weights on the COCO dataset [4] to train the building detection model for 100 epochs and select the best weights on the validation set for further calculations. The hyperparameters used during training are shown in Table 2.

**Table 2:** Top 5 Hyperparameters for Training YOLOv8

Hyperparameter	Value	Detail
Initial Learning Rate (lr0)	0.01	Sets the starting learning rate, crucial for the early stages of the training process.
Weight Decay	0.0005	Helps prevent overfitting by penalizing large weights, important for model regularization.
Momentum	0.937	Accelerates SGD in the right direction, critical for faster and more stable convergence.
Nominal Batch Size (nbs)	64	Influences the speed and stability of the training process, as well as the generalization ability of the model.
Mosaic Augmentation (mosaic)	1.0	Data augmentation technique, enhances the model's ability to generalize and recognize objects in various scenarios.

## 2.3 Building Localization

Building localization refers to the estimation of latitude and longitude of the detected buildings from the smartphone camera images. To calculate the GPS coordinates of identified buildings from the known GPS location of the ego vehicle, two key parameters are essential: the distance  $\mathcal{D}$  to the detected buildings from the ego vehicle, and the bearing  $\beta_{rel}$ . This bearing  $\beta_{rel}$  denotes the direction in which we must travel the distance  $\mathcal{D}$  from the ego vehicle to arrive at the GPS location of the buildings.  $\beta_{rel}$  is obtained by considering the bearing of ego vehicle  $\beta_{fwd}$  and the angle the detected vehicles make with respect to the smartphone camera or ego vehicle. In the next three sub-sections, we explain the estimation of building, angle estimation and localization of building using the GPS position of the ego vehicle.

### 2.3.1 Building Distance Estimation

To calculate the distance to detected buildings, we employ a photogrammetry method proposed in research conducted by Kumar et al. [2, 5]. This technique is based on the principle that the ratio of an object's actual size to its size in an image is equal to the ratio of the object's distance from the camera sensor to the camera's focal length. This relationship is mathematically expressed as:

$$\frac{H_{actual}}{H_{image}} = \frac{D}{f} \quad (1)$$

Here,  $H_{actual}$  represents the real height of the building,  $H_{image}$  is the building's height in the image,  $D$  denotes the distance to the building, and  $f$  is the focal length of the camera. We focus on measuring the height of buildings instead of their width due to the consistent nature of height in varying perspectives and distances in images. Buildings are categorized into types such as low rise, mid rise, and high rise,

each with a predefined average height, as explained before.

In digital cameras, such as smartphones, object dimensions are typically measured in pixels. This pixel measurement can be converted to metric units using the formula  $H_{image} = \frac{\mu \times H_{px}}{I_H}$ , where  $\mu$  is the sensor height in millimeters,  $H_{px}$  is the object's height in pixels, and  $I_H$  is the image height in pixels. By substituting these values into Equation 1 and rearranging, we obtain the distance to the building in meters:

$$\mathcal{D} = \frac{f \times H_{actual} \times I_H}{H_{px} \times \mu} \quad (2)$$

We consider different  $H_{actual}$  for different categories of buildings, as shown in Table 3. It should be noted that heights of building is not always fixed and because of this there might be some error in the calculated distance  $\mathcal{D}$ . However, the estimated distance does not have to be precise as building dimensions are also large, which gives extra tolerance for position estimation considering the error in the distance due to fixed height of the buildings.

**Table 3:** Reference building heights  $H_{actual}$

Building Class	Building Height $H_{actual}$
Low-rise (low-rise residence)	30 meters
Mid-rise (apartment)	50 meters
High-rise (high-rise residence)	80 meters

### 2.3.2 Building Angle Estimation

Angle estimation involves calculating the angle between the direction of the vehicle (ego vehicle) and the detected building, using principles of photogrammetry, camera calibration data, and geometric calculations. We calibrate the camera in advance using a chessboard pattern. The

camera calibration coefficients include the camera matrix and distortion coefficients, denoted as  $K$ . These coefficients are critical for correcting lens distortion and transforming 2D image points into 3D world points using the inverse of the camera matrix,  $K^{-1}$ .

For each building detected in the frame, its central pixel coordinates  $(c_x, c_y)$  are determined as the center of the bounding boxes. Using the inverse camera matrix  $K^{-1}$ , we transform these coordinates to a normalized camera coordinate system:

$$R_I = K^{-1} \cdot \begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} \quad (3)$$

This produces a ray  $R_I$  originating from the camera center and intersecting the detected building. The reference direction  $R_C$ , corresponding to the camera's optical axis, is determined using  $K^{-1}$  and the image center. The cosine of the angle between  $R_I$  and  $R_C$  is calculated using their dot product:

$$\cos(\theta) = \frac{R_I \cdot R_C}{\|R_I\| \|R_C\|} \quad (4)$$

To ascertain the precise direction towards the detected building, the angle is determined using the cross product of vectors  $R_C$  and  $R_I$ . The cross product helps in identifying the orientation of  $R_I$  relative to  $R_C$ . The Z-component of the cross product, denoted as  $\vec{v}_z$ , indicates whether  $R_I$  lies to the left or right of  $R_C$ . This is essential for determining the correct sign of the angle. The equation for the cross product is given by:

$$\vec{v} = R_C \times R_I \quad (5)$$

Where  $\vec{v}$  represents the vector resulting from the cross product. Based on the sign of the Zcomponent of  $\vec{v}$ , the angle  $\theta$  is adjusted accordingly:

$$\theta_{corrected} = \begin{cases} -\theta & \text{if } \vec{v}_z < 0 \\ \theta & \text{otherwise} \end{cases} \quad (6)$$

### 2.3.3 Building Position Estimation

The relative bearing to the building is calculated by adjusting the detected angle  $\theta_{corrected}$  with the ego vehicle's forward bearing  $\beta_{fwd}$ :

$$\beta_{rel} = (\beta_{fwd} - \theta_{corrected}) \bmod 360 \quad (7)$$

This relative bearing  $\beta_{rel}$  is used along with the known distance  $D$  to compute the GPS position of the building.

The GPS position of the building  $(\varphi_{bldg_i}, \lambda_{bldg_i})$  is calculated using the geodesic method [6] by considering the current GPS position of the ego vehicle and moving a distance  $D$  at bearing  $\beta_{rel}$ .

### 2.4 Localization Error Evaluation

This section outlines the methodology employed for estimating the error in the estimated building position  $(\varphi_{bldg_i}, \lambda_{bldg_i})$ . The error is estimated as follows:

1. For each estimated position  $(\varphi_{bldg_i}, \lambda_{bldg_i})$ , its location is first verified on a map.
2. If the estimated position falls directly on the building, the error is considered to be zero meters, indicating a precise estimation.
3. If the estimated position is outside the building's periphery, the nearest point on the building's outline is identified.
4. The error is then calculated as the geodesic distance [7] between the estimated position  $(\varphi_{bldg_i}, \lambda_{bldg_i})$  and the nearest point on the building  $(\varphi_{bldg\_boundary_i}, \lambda_{bldg\_boundary_i})$ . This distance is computed on the Earth's ellipsoidal surface, which offers a more accurate representation than a simple spherical model. Mathematically, the error for the  $i^{th}$  building  $\mathbb{E}_i$  is given by:

$$\mathbb{E}_i = \text{geodesic\_distance}((\varphi_{bldg_i}, \lambda_{bldg_i}), (\varphi_{bldg\_boundary_i}, \lambda_{bldg\_boundary_i}))$$

The mean error ( $\mathbb{E}$ ) over all  $n$  samples is calculated as:

$$\bar{E} = \frac{1}{n} \sum_{i=1}^n E_i \quad (8)$$

### 3 Results

#### 3.1 Building Detection

We evaluate the accuracy of the YOLOv8 model trained on the validation set of the building detection dataset. We present the results of precision, recall and mAP in Figure 3. From Figure 3, we notice that all three metrics precision, recall and mAP increases as the number of epochs increases and reaches a plateau after 60 epochs. We obtain the highest mAP of 0.762 at 85 epoch and use this weight for further detection of buildings.

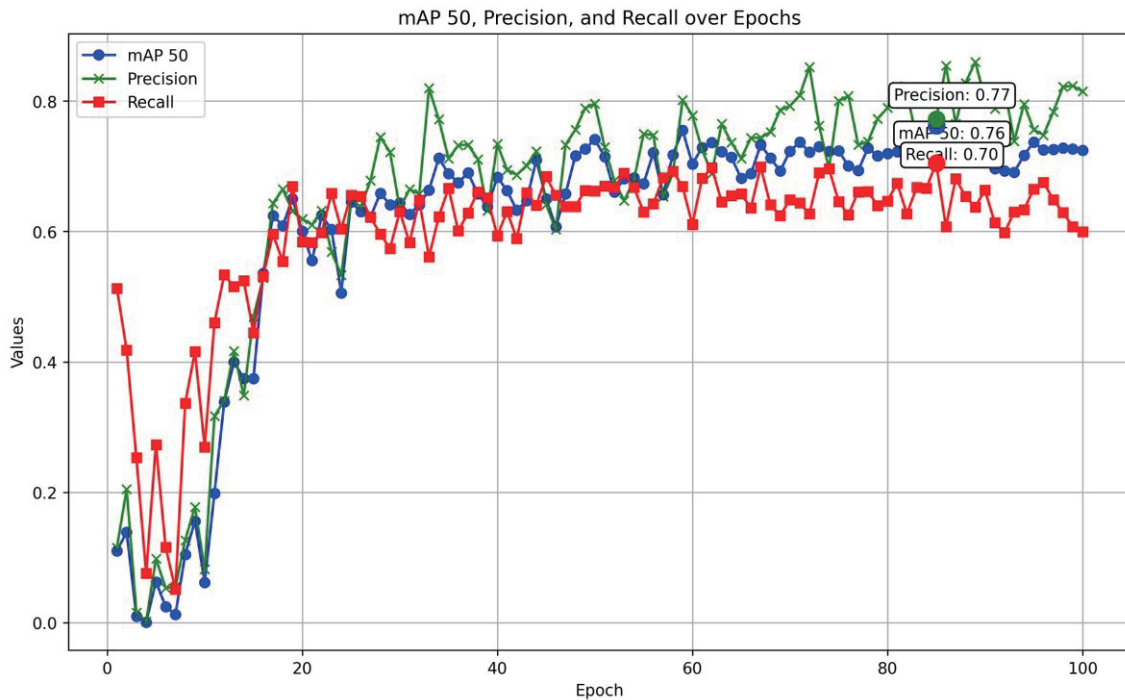
#### 3.2 Building Localization

In Table 4, we show the localization of detected building on the map along with car's GPS location (shown using blue marker). From

different examples in Table 4, we notice that we can estimate location of buildings accurately for various types of buildings, such as mansion, tower mansions, etc.

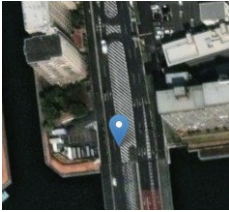


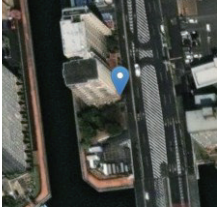
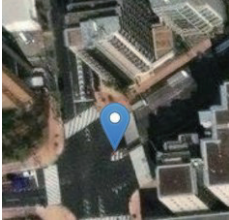


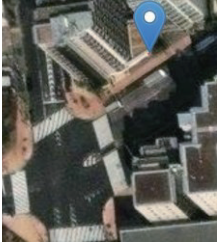
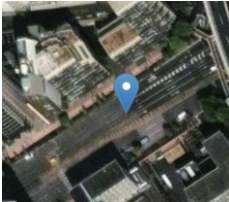


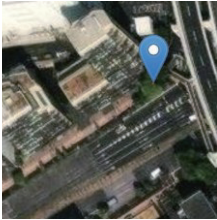
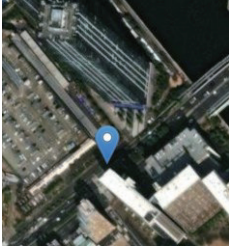


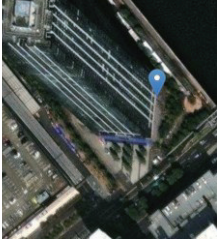
From the results in Table 4, we can see that in every frame, several buildings can be detected. However, we only consider the nearest buildings (shown using green bounding boxes) for localization within a threshold distance of 50 meters.

We evaluate the error (using Equation 8) in localization using 60 samples of buildings from a test drive in the Tokyo metropolitan area and present result in Figure 4. From the histogram presented in Figure 4a, we notice that in several instances the error is 0 meters because the localized position lies on/inside the periphery of the detected building. We obtain an overall error of 5.22 meters.

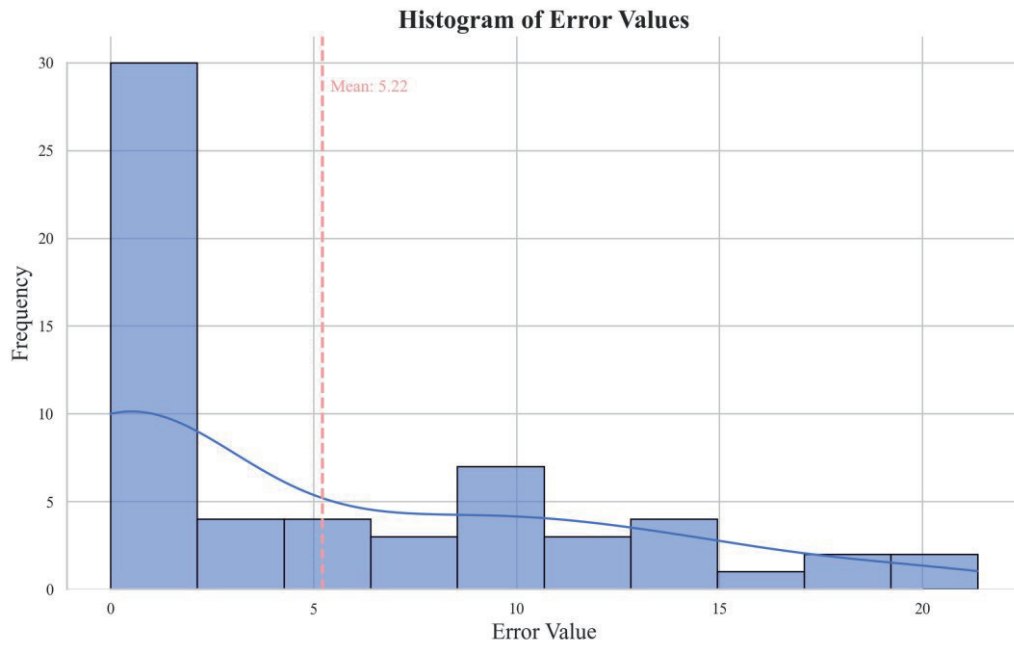


**Fig. 3:** Progression of mAP 50, Precision, and Recall over different epochs with highlighted maximum mAP 50.

**Table 4:** Car's GPS position, detected building location and visualization on map. Image courtesy of USGS Earth Explorer. © 2024/01/21.

Car's GPS	Building Detection	Target Building	Localized GPS on map
 <p>35.660298, 139.8037941</p>			 <p>35.660681378870244, 139.80357100495178</p>
 <p>35.661935, 139.8036792</p>			 <p>35.66229062062282, 139.8039596017458</p>
 <p>35.6625822, 139.8048738</p>			 <p>35.662903031199804, 139.80515298698734</p>
 <p>35.6564844, 139.7991766</p>			 <p>35.65726367288982, 139.79965628513142</p>





(a) Histogram of error values in building localization with mean error



(b) Localized position visualization on map. Green bars show the ground truth while blue bars represent predicted position. Map data: © OpenStreetMap contributors, CC-BY-SA

**Fig. 4:** Histogram of error values in building localization along with estimated and ground truth building GPS position visualized on map for a sample driving experiment

In Figure 4b, we also present visualization of predicted and ground truth positions of buildings. To avoid crowding of points for closer buildings or when the same building is localized from different angles, we consider a distance threshold to only consider a small sample.

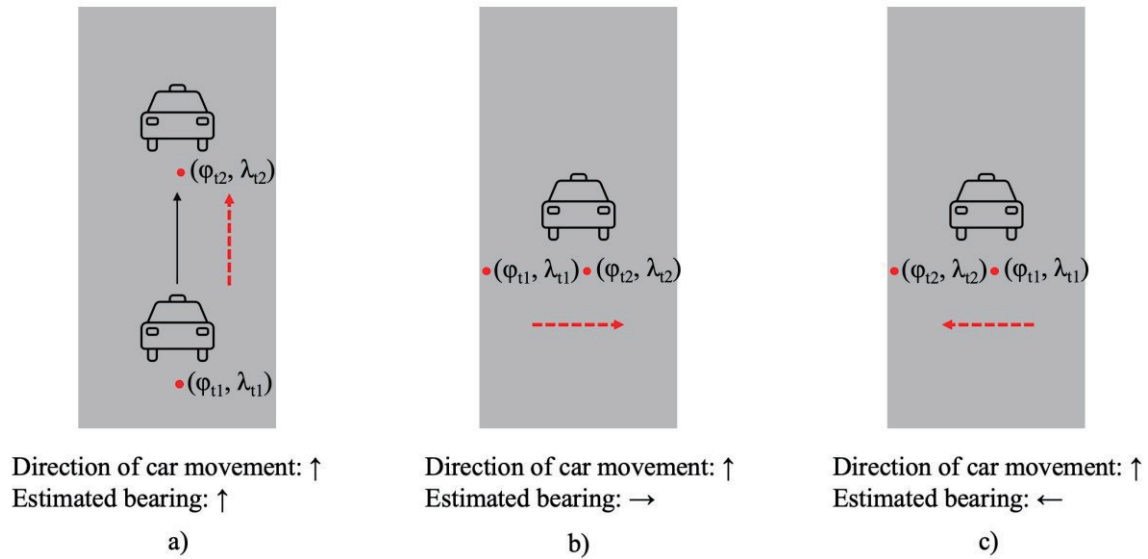
#### 4 Discussions

In the results for building detection, we observed that the mean Average Precision (mAP) value for building detection is 0.762. This indicates that we can detect buildings with high accuracy. It should be noted that building types can vary in shape and size, even within the same category, such as midrise buildings (mansions), which leads to some errors in the classification. In addition to variations within the same category, the appearance of buildings also changes as the ego vehicle approaches them. For example, a tower mansion is detected and categorized correctly when viewed from a distance. However, when the ego vehicle is close to the building, the full view is not visible, and it may be misclassified as a mansion. Misclassification of buildings does not significantly affect the accuracy of localization since this categorization is mainly used for distance estimation, as indicated by the real height of the building ( $H_{actual}$ ) in Equation 2. The reference heights of mansions and tower mansions do not vary significantly, as shown in Table 3.

From the building localization results, we find that buildings can be localized accurately. In some cases, we notice some error in localization,

such as the localized position being either overpasses the detected building or lies in front of the building. Such phenomenon occur mainly due to two reasons. The first reason is the fixed height consideration  $H_{actual}$  in Equation 2. Such problems could be fixed using depth estimation techniques that does not require consideration of object dependent parameters, such as using Depth Anything [8] at the expense of computational cost. The second reason is also due to error inherent in the ego vehicle GPS location. The smartphone's GPS sensor may have positioning error of a few meters, especially in the dense areas due to factors, such as multipath error, signal blockage, etc. [9]. The initial error in the source may lead to error propagation in the estimated distance.

Building position estimation requires the forward bearing of the ego vehicle, which is calculated using GPS position at two timestamps  $t$  and  $t + 1$ . When the vehicle is moving in the forward direction, the GPS positions at two timestamps lie in the direction of the vehicle movement. The problem arises when the vehicle is at rest. In such cases, the recorded positions may not be in the direction of the movement and may lie laterally to the car direction. This causes the change in the direction of bearing leading to incorrect localization, as illustrated in Figure 5. In such cases, it is necessary to consider either moving average of the past GPS traces or consider the bearing of the vehicle only when the vehicle is moving.



**Fig. 5:** Bearing direction estimation when the car is moving (a) and when it is at rest (b) and (c). When the vehicle is at rest, the GPS position of the smartphone at two different times may lie lateral (sideways) to the direction of the vehicle, as shown in b) and c).

## 5 Conclusions

In this research, we introduce a framework for the accurate localization of urban buildings using moving smartphone cameras. We develop a building detection dataset containing more than 7,000 instances of various types of buildings. A key component of our methodology was the training of the YOLOv8 model, which demonstrated high accuracy in detecting a diverse range of building types, such as low-rise residence, apartment, etc. with an mAP of 76%. We integrate the building detection results with distance and angle estimation and estimate building location by considering ego vehicle GPS position with distance and bearing of the detected vehicles. Using our localization method, we achieve accuracy of 5.22 meters. In addition, our research examines the causes of localization errors, providing valuable insights for future improvements. We identified challenges such as the fixed height consideration in distance estimation and the inherent inaccuracies in smartphone GPS data. Addressing these issues

will be crucial for enhancing the accuracy of urban mapping technologies.

Our proposed method exemplifies the potential of integrating AI algorithms with photogrammetry and geospatial data, especially when leveraging the mobility and ubiquity of smartphones. This technique not only enables dynamic mapping of urban landscapes but also addresses the limitations of static observational methods.

The implications of our work extend beyond the realm of urban planning and development. The precise localization of buildings can significantly benefit emergency response systems, navigation applications, and augmented reality experiences. Additionally, the potential applications in delivery services, disaster management, and preservation efforts highlight the societal impact of this research. While the current results are promising, ongoing advancements in AI and geospatial technologies are expected to further refine and expand the capabilities of urban mapping methods in the

future.

#### References

- [1] Boguszewski, A., Batorski, D., Ziembajankowska, N., Dziedzic, T., Zambrzycka, A.: Landcover. ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1102–1110 (2021)
- [2] Kumar, A., Kashiyama, T., Maeda, H., Omata, H., Sekimoto, Y.: Real-time citywide reconstruction of traffic flow from moving cameras on lightweight edge devices. *ISPRS Journal of Photogrammetry and Remote Sensing* **192**, 115–129 (2022)
- [3] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLOv8. <https://github.com/ultralytics/ultralytics>
- [4] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, pp. 740–755 (2014). Springer
- [5] Kendal, D.: Measuring distances using digital cameras. *Australian Senior Mathematics Journal* **21**(2), 24–28 (2007)
- [6] Karney, C.F.: Algorithms for geodesics. *Journal of Geodesy* **87**(1), 43–55 (2013)
- [7] Vincenty, T.: Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review* **23**(176), 88–93 (1975)
- [8] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. arXiv:2401.10891 (2024)
- [9] Arnold, L.L., Zandbergen, P.A.: Positional accuracy of the wide area augmentation system in consumer-grade gps units. *Computers & Geosciences* **37**(7), 883–892 (2011)