Image matching of satellite and UAV for visual place recognition using YOLO*

Volodymyr Vozniak[1,*,†], Oleksander Barmak[1,†] and Iurii Krak[2,3,†]

[1] Khmelnytskyi National University, 11, Institutes str., Khmelnytskyi, 29016, Ukraine
[2] Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str., Kyiv, 01601, Ukraine
[3] Glushkov Cybernetics Institute, 40, Glushkov Ave., Kyiv, 03187, Ukraine

Abstract

Determining the location of UAVs under challenging conditions plays a crucial role in various applications. In this research, we aim to enhance the accuracy of CNN-based methods for global UAV localization in urban environments using models capable of rapid real-time image processing. By utilizing the YOLO11 model, fine-tuned on a dataset of segmented buildings (achieving an F1-score of 0.722), and employing a proposed statistical distribution alignment method to increase visual similarity between satellite and UAV images, we obtained a Recall@1 metric value of 0.195 with a localization radius of 3 for UAV global localization in urban areas. This result surpasses those obtained by existing CNN-based methods. The outcomes indicate that employing YOLO-generated image vector representations combined with our image preprocessing approach is promising for UAV global localization, with significant potential for further improvement.

Keywords

Visual Place Recognition (VPR), UAV, YOLO, image preprocessing, deep learning, image segmentation

1. Introduction

The accurate determination of Unmanned Aerial Vehicle (UAV) location under challenging conditions plays a decisive role in numerous applications, from rescue operations and agricultural monitoring to mapping and industrial inspections. Typically, Global Positioning Systems (GPS) are utilized to obtain coordinates; however, navigation accuracy significantly deteriorates in areas with substantial radio interference or compromised GPS signals. In such scenarios, methods based on analyzing images acquired from onboard UAV cameras and corresponding satellite imagery of the terrain can assume a critical role.

Due to rapid advancements in deep neural networks over recent years, researchers have demonstrated increased interest in using visual landmarks to determine UAV coordinates. The findings from numerous studies [1] – [16] highlight the potential of deep learning in tasks related to image matching and identifying unique visual landmarks, even in conditions of absent or significantly degraded GPS signals. Additional advantages of this approach include scalability (for instance, through processing extensive satellite imagery and data from various sensors) and the potential to employ hybrid navigation solutions (combining GPS with visual algorithms). Consequently, location determination remains highly relevant for further research as the range of applications requiring reliable and uninterrupted UAV navigation continually expands [1].

The contributions of this research include:

- obtaining vector representations of images from the convolutional layers of a fine-tuned YOLO11 model trained on a dataset of segmented buildings;
- determining the global location of UAVs using averaged cumulative distribution functions of images, which enables more precise matching between UAV imagery and satellite images.

The paper is structured as follows: Section 2 provides a literature review and defines the research objectives and tasks. Section 3 outlines the overall process for UAV location determination, describes

*CITI'2025: 3rd International Workshop on Computer Information Technologies in Industry 4.0, June 11–12, 2025, Ternopil, Ukraine
[1,*] Corresponding author.
† These authors contributed equally.
vozniakvz@khmnu.edu.ua (V. Vozniak); barmako@khmnu.edu.ua (O. Barmak); iurii.krak@knu.ua (I. Krak)
0009-0008-3055-5257 (V. Vozniak); 0000-0003-0739-9678 (O. Barmak); 0000-0002-8043-0785 (I. Krak)

limitations, and proposes a method for aligning statistical distributions between satellite imagery and UAV images to achieve visual similarity. Section 4 details the datasets used, metrics for method evaluation, and experimental methodology. Section 5 presents results, discusses the fine-tuned YOLO model using the dataset with segmented buildings, and compares results for UAV location determination against existing methods.

2. Literature review

In the task of automatic location determination (Visual Place Recognition, VPR), recent years have seen significant methodological progress driven by the extensive adoption of deep learning methods adapted to varying environmental conditions and different imaging angles. Primary approaches rely on global descriptors derived from convolutional neural networks (CNNs). These methods involve neural network-generated image features aggregated into a single compact vector representation, achieving an optimal balance between matching accuracy and retrieval speed. This contrasts with earlier popular methods relying on local descriptors, which are more subjective in matching, mainly when dealing with large datasets, rendering them unsuitable for scenarios with limited computational resources or prone to instability under severe lighting or seasonal variations [1].

A literature analysis indicates that the high performance of most modern approaches can be attributed to the growing size of training datasets designed explicitly for visual localization [2], [3], [4]. NetVLAD [5] combines local feature encoding with the VLAD training layer to create robust global descriptors. Although trained on the extensive Pitts-250k dataset [6], NetVLAD descriptors are high-dimensional (tens of thousands), necessitating increased computational memory storage and potentially hindering real-time UAV applications [7], [8]. Recent variants like Patch-NetVLAD [9] incorporate multi-scale feature aggregation to enhance viewpoint robustness, yet the aggregated representations remain relatively large.

Several studies explored lightweight or multimodal representations. MinkLoc [10] employs sparse 3D convolutions for robust large-scale place recognition, particularly effective with LiDAR or depth-map data. However, due to payload and power limitations, supplementary sensors may be impractical for many UAV platforms. Recently, CosPlace [11] combined classification-based learning using the San Francisco XL dataset, containing 40 million GPS-tagged directional images. Subsequently, MixVPR [12] introduced an MLP-based feature mixer trained on GSV-Cities [13], a specialized dataset comprising 530,000 images from 62,000 global locations. These examples underline the significant volume of specialized datasets utilized in recent studies.

Advances in attention-based architectures have recently introduced several transformer-based methods utilizing attention mechanisms for image feature matching. LoFTR [14] matches local features without detectors, exhibiting robustness to moderate viewpoint changes [15]. AnyLoc [16] expands transformer paradigms by integrating self-supervised features from DinoV2 [17] and combining global and local attention modules within a unified framework for place recognition. Despite considerable metric improvements on various VPR benchmarks, transformer architectures' main disadvantages remain their high hardware requirements and low prediction speed. For example, AnyLoc's heavy ViT-Large architecture and 32,000-dimensional VLAD image feature limit its practical application for real-time tasks on resource-constrained devices such as UAVs.

Although computationally efficient, CNN-based visual place recognition methods require training on large, specialized datasets. VLAD-feature methods deliver enhanced performance but require significant memory due to high-dimensional vectors. Finally, transformer-based methods provide high-quality and versatile vector features that are usable without additional fine-tuning but remain considerably slower than CNN models, complicating deployment on UAV platforms.

Surprisingly, using YOLO [18], a CNN-based model renowned for real-time efficiency and high accuracy, has not yet gained popularity in VPR.

Therefore, this study aims to enhance the accuracy of CNN-based methods for UAV global location determination in urban environments using models capable of high-speed real-time image processing.

To achieve this goal, we have formulated the following tasks:

- fine-tune the YOLO11 model [19] on a dataset with segmented buildings to obtain vector representations for UAV global location determination under challenging conditions;

- develop a method for aligning the statistical distributions of satellite imagery and UAV images to achieve visual similarity;
- compare results obtained by the proposed method against existing CNN-based methods across different terrain types.

3. Materials and methods

Determining the location of Unmanned Aerial Vehicles (UAVs) is typically addressed through Visual Place Recognition (VPR). The approach proposed in this study involves comparing a query image obtained from a UAV with a database of georeferenced satellite images. The system determines the UAV's global location by identifying the most similar image (or set of images). Subsequently, a method is applied to determine precise coordinates by aligning the query image with the corresponding satellite image.

Therefore, the considered task can be decomposed into the following sub-tasks:

- determining the global location: identifying the most similar satellite image (or tile) from an extensive database;
- determining precise coordinates: calculating the exact position (latitude and longitude) within the identified tile.

Figure 1 illustrates the general processing scheme for determining UAV location coordinates.

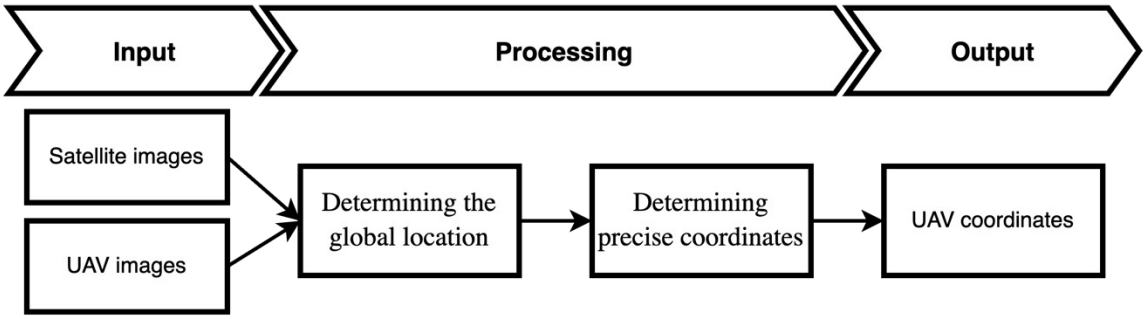

Figure 1: General data processing scheme for determining UAV location coordinates.

Using CNN architectures, this study explicitly addresses determining UAVs' global location in urban environments. Figure 2 shows a detailed scheme of the proposed process.

We introduce the following notation. Let $Q=\{q\}$ represent a query image obtained by the UAV at a specific position and orientation, and $R=\{r_1,r_2,\ldots,r_n\}$ represent a set of pre-prepared images with known coordinates (usually satellite images).

Additionally, we define the following mapping for obtaining vector representations (feature vectors) of images:

$$F: I \to V, \tag{1}$$

where I is the image, V is the feature vector.

Thus, the general task of determining the global UAV location involves identifying the image $r_j \in R$ whose spatial location $l(r_j)$ is closest to the query image $l(q)$. Formally, this can be expressed as:

$$k = \underset{1 \le j \le n}{argmin}\, d\left(F(q), F(r_j)\right), \tag{2}$$

where $d(\cdot,\cdot)$ is a metric (e.g., Euclidean distance) operating on images represented as feature vectors and k is the identified satellite image.

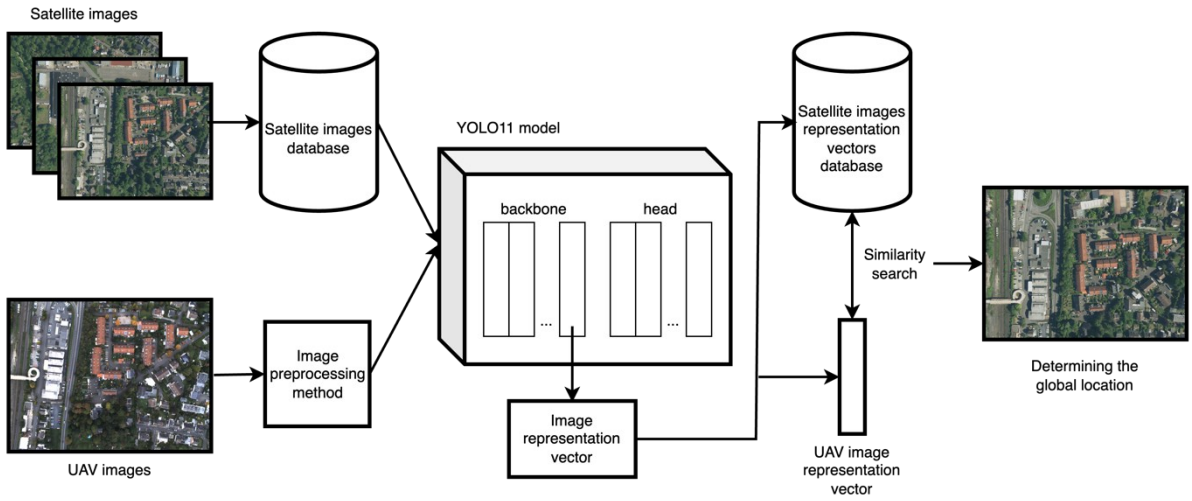

Figure 2: Processing scheme for UAV global localization using an image preprocessing method and a YOLO11 model fine-tuned on a dataset of segmented buildings.

A database of satellite images along a predefined route must be available to determine the UAV's global location. Using deep learning models, image vector representations or feature sets can be obtained (1). In this study, the YOLO11 model is employed and fine-tuned on a dataset of segmented buildings. The image representation vector is obtained from the final backbone layer, which contains the most comprehensive information about the image derived from YOLO11's convolutional layers. Importantly, this model is identical to vector representations of both satellite images and UAV images. Therefore, the fine-tuned YOLO11 model generates vector representations for the satellite image database, facilitating UAV global location determination. In real-time mode, UAV images are fed into the YOLO11 model to obtain their vector representations (1), subsequently identifying the most suitable satellite image (2) by minimizing a similarity search function (e.g., Euclidean distance). The matched satellite image thus represents the UAV's global location.

We propose a method that aligns their statistical distributions to enhance visual similarity between UAV and satellite images. Unlike methods that directly apply cumulative distribution functions (CDF) from satellite to UAV images [20] (which may cause distortions due to distribution mismatch), our approach calculates transformations individually for each UAV frame, considering its unique distribution.

The method leverages probability theory: given a random variable ξ with a known distribution function $F_\xi(x)$, one can obtain a uniformly distributed random variable $\gamma \sim U(0,1)$by applying the distribution function to itself $\gamma = F_\xi(\xi)$. Conversely, applying the inverse distribution function $\gamma \sim U(0,1)$yields a variable ξ with the distribution $F_\xi(x)$. Satellite images collected by the same device share consistent color characteristics. UAV images, acquired under different conditions, exhibit distinct statistical properties. Pixel intensities can thus be treated as discrete random variables. Correspondingly, intensities from UAV and satellite images yield random variables Y, and X precisely calculating $F_X(x)$ from satellite datasets and estimating $F_Y(y)$ for UAV images allows transformation via $F_X^{-1}(F_Y(y))$ distribution alignment. Figure 3 shows the diagram of the proposed method based on the probability theory.

The proposed method comprises two stages:

1. Computing the averaged cumulative distribution function from satellite images;
2. Applying this function individually to UAV images.

Algorithm 1 provides pseudocode for computing the averaged cumulative distribution function from satellite images, and Algorithm 2 details applying this averaged function individually to UAV images.

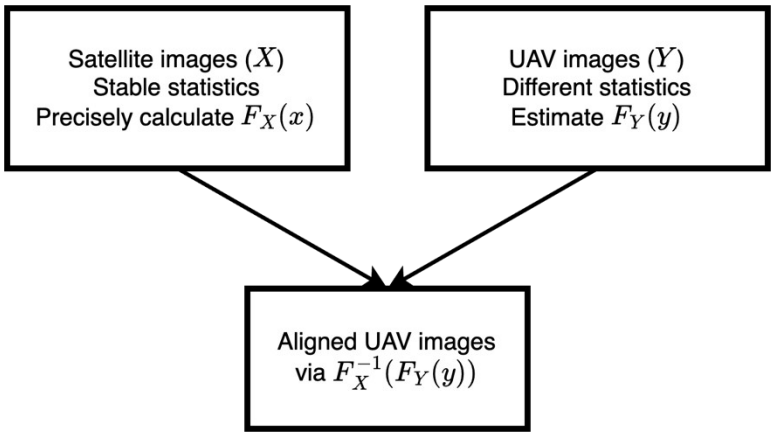

Figure 3: Diagram of the proposed method based on the probability theory.

Algorithm 1: Computing the averaged cumulative distribution function from satellite images

Input: $R=\{r_1, r_2, \ldots, r_n\}$ – n satellite images.

Initialization: $F^{sum} \leftarrow 0_{x \times c}, x \in [0; 255], c \in \{R, G, B\}$.

For i in 1..n

Step 1. Compute the normalized histogram (probability density function) for each color channel $c \in \{R, G, B\}$:

$$E_{c,i}(x) = \frac{H_{c,i}(x)}{\sum_{x=0}^{255} H_{c,i}(x)}, \quad (3)$$

where $H_{c,i}(x)$ is the number of pixels with value x in channel c of the i-th satellite image.

Step 2. Compute the cumulative distribution function (CDF) for each channel:

$$F_{c,i}(x) = \sum_{k=0}^{x} E_{c,i}(k). \quad (4)$$

Step 3. Add the cumulative distribution function values to to later derive the average cumulative distribution function:

$$F_i(x) = \left[F_{R,i}(x), F_{G,i}(x), F_{B,i}(x)\right]; \quad (5)$$

$$F^{sum}(x) = F^{sum}(x) + \mathrm{F}_i(\mathrm{x}). \quad (6)$$

End

Step 4. Compute the averaged cumulative distribution function (CDF):

$$\hat{F}(x) = \frac{F^{sum}(x)}{n}. \quad (7)$$

Output: $\hat{F}(x)$– averaged cumulative distribution function of satellite images, $x \in [0; 255]$.

Algorithm 2: Applying averaged cumulative distribution function of satellite images individually to UAV images.

Input: $Q=\{q_1, q_2, \ldots, q_m\}$ – m UAV images.
For j in 1..m
Step 1. Compute the normalized histogram (probability density function) for each color channel $c \in \{R, G, B\}$:

$$D_{c,j}(y)=\frac{S_{c,j}(y)}{\sum_{y=0}^{255} S_{c,j}(y)}, \quad (8)$$

where $S_{c,j}(y)$ is the number of pixels with value y in channel c of the j-th UAV image.
Step 2. Compute the cumulative distribution function (CDF) for each channel:

$$G_{c,j}(y)=\sum_{n=0}^{y} E_{c,j}(n). \quad (9)$$

Step 3. Define the transformation function for each channel c by finding the value at which the cumulative distribution function (CDF) from the UAV images aligns with the averaged cumulative distribution function of satellite images:

$$M_{c,j}(y)=\widehat{F}_{c,j}^{-1}\left(G_{c,j}(y)\right), \quad (10)$$

where $\widehat{F}_c^{-1}$ is the inverse function of the averaged cumulative distribution function for satellite images in channel c. Because $\widehat{F}_c^{-1}$ might be non-analytical, interpolation is used as an approximation:

$$M_{c,j}(y)=\text{interp}\left(G_{c,j}(y), \widehat{F}_c(z), z\right), \quad (11)$$

where is an interpolation function that finds the corresponding z for each $G_c(y)$, such that $\widehat{F}_c(z) \approx G_c(z)$.
End
Output: $M_{c,j}(y)$ – the resulting function that transforms the input pixels of the *j*-th UAV image for the given color channel *c*, $y \in [0;255]$.

Histogram alignment ensures the similarity of pixel intensity distributions between UAV and satellite images, adjusting brightness, contrast, and tonal characteristics, thus reducing intra-class variation and enhancing feature matching and recognition tasks.

4. Experimental setup

4.1. Datasets

4.1.1. Dataset for YOLO finetuning

Since the default YOLO model does not include building recognition among its classes, this research proposes fine-tuning the YOLO11 model [19] on a dataset of segmented buildings [21] containing only one class – buildings. This dataset consists of 9665 images, predominantly from the following cities: Tyrol (2999), Tripoli (1078), Kherson (1053), Donetsk (999), Mekelle (951), Mykolaiv (739), and Kharkiv (602). Figure 4 shows an example of segmented buildings from an image of the training dataset.

Figure 4: An example of segmented buildings from one image of the training dataset [21].

4.1.2. VPAir

For validation of the proposed method, aligning statistical distributions between satellite and UAV images, the VPAir dataset [22] was selected. This dataset was explicitly created for the challenge of UAV localization under complex conditions. Data were collected during a flight between Bonn and the mountainous Eifel region in Germany, spanning 107 kilometers and encompassing diverse landscapes, including urban, agricultural, and forested areas. Data collection took place on October 13, 2020, using a lightweight aircraft flying at altitudes between 300–400 meters. The equipment included a single-lens color camera (resolution 1600x1200 pixels, reduced to 800x600 pixels for the dataset) and a GNSS/INS system providing highly accurate 6-DoF positions (rotation error: 0.05°, positional error: <1 meter). The dataset comprises 2706 query images captured from the aircraft, 2706 corresponding satellite database images, and 10,000 distractor images from another geographical region near Düsseldorf.

Although the authors of [22] provide metrics comparing different algorithms across various terrain types, this annotation is not publicly available. Therefore, we proposed our annotation of terrain types for this dataset, classifying them into four categories: urban (dominated by buildings and streets), field, forest, and water.

4.2. Evaluation

The following metrics are commonly used to evaluate image segmentation model performance: mAP, Precision, Recall, and F1-score. Precision, Recall, and F1-score [23] are standard metrics applied broadly in machine learning tasks, ranging from binary classification to image segmentation. A distinctive feature of segmentation tasks is the absence of true-negative counts in the confusion matrix, which does not hinder the computation of these metrics.

Additionally, mean Average Precision (mAP) deserves separate mention. Formally, it can be expressed as follows:

$$AP_c = \sum_{n=1}^{N} \left(R_n - R_{n-1}\right) P_n, \tag{12}$$

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c, \tag{13}$$

where P_n and R_n are Precision and Recall at the threshold n with $R_0 = 0$ and $R_N = 1$, C is the number of classes, AP_c – average precision for the class c.

AP_c can also be interpreted as the area under the Precision-Recall curve for class c. Since this study involves segmentation with a single class (buildings), here $mAP = AP_b$.

Recall@N [24] is commonly used to evaluate localization methods. In this metric, search results are considered true-positive for a given query if the corresponding image from the database is within the top N retrieved results:

$$\text{Recall@N} = \frac{M_Q}{N_Q}, \tag{14}$$

where N_Q is the total number of query images, and M_Q is the number of queries with at least one correct match within the top-N results.

This metric is popular within computer vision communities and is suitable for scenarios where subsequent processing may further filter false-positive matches.

A modified definition considers a localization radius, meaning a result is true-positive if the distance between the query and matched database images is within a predefined radius. This radius can be specified either in meters or in a number of frames (as satellite images are often stored sequentially). Such a metric is beneficial in scenarios with overlapping reference images, enabling precise UAV localization even if an exact image match is not found.

4.3. Experimental methodology

4.3.1. YOLO finetuning

The YOLO11 model was fine-tuned on the segmented buildings dataset using the open-source Python library ultralytics [25] on Ubuntu OS with an Nvidia MSI RTX 3060 graphics card. Only one segmentation class (Buildings) was used. The default YOLO11 architecture was employed with 100 epochs and an image size of 640 pixels.

To evaluate the performance of the fine-tuned YOLO model, mAP, Precision, Recall, and F1-score metrics were calculated. F1-score was chosen as the primary metric for this task since both missing a building when present and incorrectly identifying a non-building as a building constitute equally adverse segmentation outcomes. Each metric was computed by performing seven random splits of the dataset into training and testing subsets in an 80%/20% ratio, subsequently calculating average values (Avg) and standard deviations (Std) for model stability assessment.

4.3.2. Image preprocessing method validation

The proposed method was validated by aligning statistical distributions between satellite and UAV images for determining UAV global location using the VPAir dataset [22]. The Recall@1 (14) metric was chosen for evaluation, with a localization radius of 3. This metric was computed across various terrain types: urban, field, forest, and water. This study considered urban terrain the primary environment, as the YOLO11 model was fine-tuned, specifically using segmented building images for generating vector representations of both satellite and UAV images.

Three experiments were conducted to compare the proposed method (using vector representations from YOLO11) with known UAV global localization methods (CosPlace [11]): experiments using unprocessed VPAir dataset [22] images, experiments using grayscale conversions of these images, and experiments employing the proposed image preprocessing method.

5. Results and discussion

5.1. YOLO fine-tuning results

Figure 5 shows an example of building segmentation on an image from the test dataset used for fine-tuning YOLO11. Figure 6 presents an example of building segmentation on an image from the VPAir dataset [22].

Figure 7 displays the loss function plots for the training (train) and validation (val) sets, which were used to fine-tune the YOLO11 model on the optimal training and testing data split.

Architecturally, the YOLO model family employs a weighted sum of various loss functions. For image segmentation tasks, four specific loss functions are utilized:

- box_loss – emphasizes the accurate placement of bounding boxes around detected objects (weight coefficient: 7.5);
- seg_loss – emphasizes the accurate placement of segmentation masks around segmented objects (weight coefficient: 7.5);
- cls_loss – emphasizes the correct classification of objects (weight coefficient: 0.5);

dfl_loss – emphasizes differentiating between objects that are visually similar or difficult to distinguish by better capturing their unique features (weight coefficient: 1.5).

Figure 5: An example of building segmentation on an image from the test dataset used for fine-tuning YOLO11.

Figure 6: An example of building segmentation on an image from the VPAir dataset [22].

The graphs indicate that the model effectively captures underlying patterns from the building segmentation dataset, as the training loss consistently decreases and stabilizes on the validation dataset.

Figure 8 shows the confusion matrix, which does not include true-negative counts since these cannot be clearly defined for segmentation tasks.

Table 1 presents the evaluation metrics obtained from the fine-tuned YOLO11 model on the segmented building dataset.

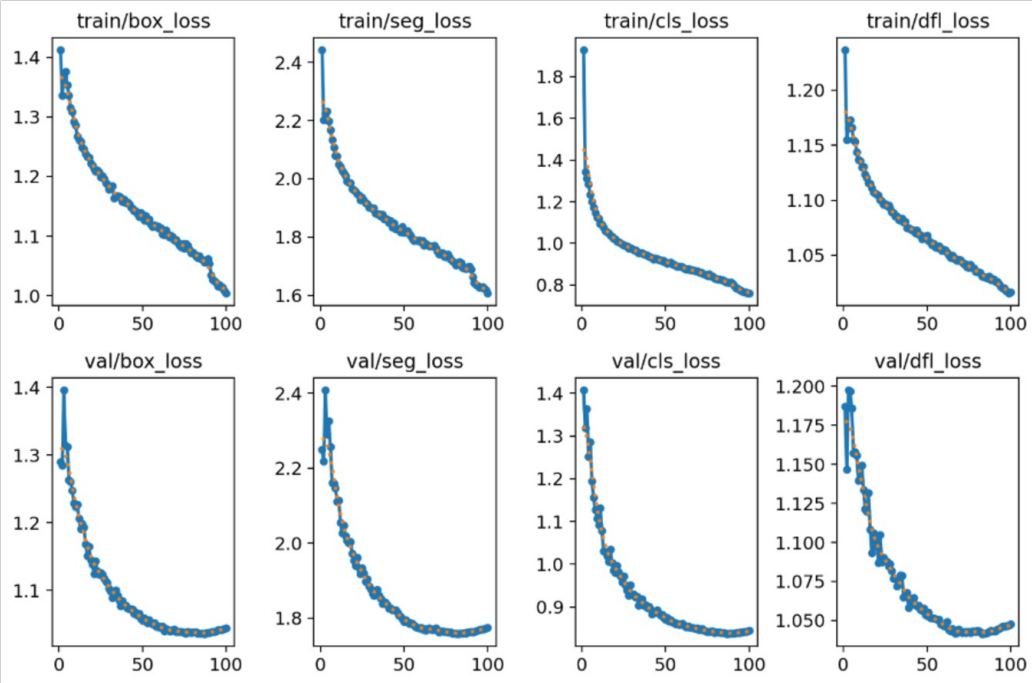

Figure 7: Loss function plots for the training (train) and validation (val) sets.

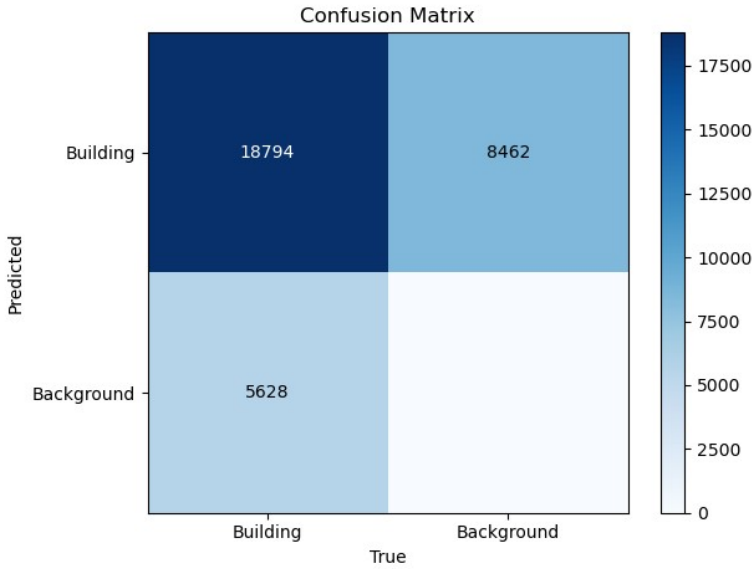

Figure 8: The confusion matrix for the YOLO11 fine-tuned model on the segmented buildings dataset.

Table 1
Evaluation metrics obtained from the fine-tuned YOLO11 model on the segmented building dataset

Metric		Random splitting							Avg	Std
		1	2	3	4	5	6	7		
mAP	Train	0.813	0.821	0.820	0.823	0.813	0.817	0.814	0.817	0.0043
	Test	0.748	0.760	0.757	0.755	0.757	0.753	0.749	0.754	0.0043
Recall	Train	0.727	0.737	0.736	0.738	0.728	0.730	0.728	0.732	0.0047
	Test	0.677	0.685	0.681	0.679	0.681	0.680	0.673	0.680	0.0037
Precision	Train	0.809	0.819	0.815	0.820	0.809	0.813	0.813	0.814	0.0042
	Test	0.764	0.773	0.775	0.768	0.778	0.770	0.769	0.771	0.0047
F1	Train	0.766	0.775	0.773	0.777	0.766	0.769	0.768	0.771	0.0044
	Test	0.718	0.727	0.725	0.721	0.726	0.722	0.718	0.722	0.0037

This table contains values for the metrics mAP, Precision, Recall, and F1-score, calculated across seven distinct dataset splits into training and testing subsets. Average values (Avg) and standard deviations (Std) are also provided to assess model stability.

Figure 9 shows the Precision-Recall curve for the best-performing training and testing dataset split, with an area under the curve (AUC) of 0.76. This curve is informative for image segmentation tasks as it emphasizes accurately identifying the positive class (buildings) in scenarios of class imbalance (buildings vs. background).

In the context of building segmentation, especially on masked images where objects may be partially obscured or possess complex contours, an F1-score of 0.722 for the test dataset can be considered a positive outcome, especially given the real-time speed advantages inherent to the YOLO model family [18]. This indicates that the trained model reliably identifies buildings, which is crucial for generating the vector representations used for UAV global localization. Furthermore, the standard deviation of each metric across both training and testing datasets is below 0.5%, indicating consistent model performance. Future improvements could explore modifications to the YOLO11 neural network architecture and hyperparameter tuning to maximize the F1-score, specifically for building segmentation tasks.

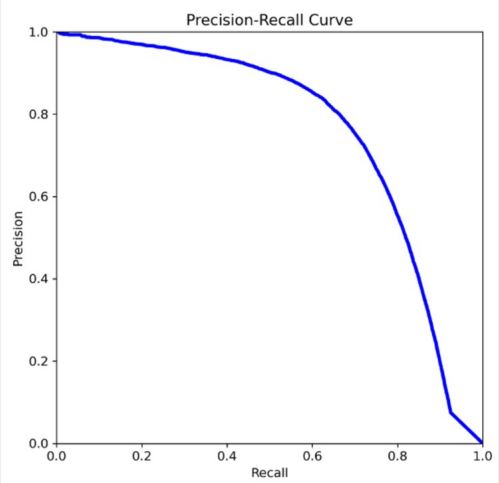

Figure 9: Precision-Recall curve for the YOLO11 fine-tuned model on the segmented buildings dataset.

5.2. Image preprocessing method results

Figure 10 illustrates the visual results of the proposed averaged cumulative distribution function (CDF) method applied to UAV images, aligning their statistical distributions with those of satellite images.

Table 2 provides Recall@1 (14) metric results with a localization radius of 3 for CosPlace [11] and the proposed UAV global localization method using YOLO11 across different terrain types from the VPAir dataset [22], employing various image processing approaches, including experiment without color preprocessing, experiment with greyscale preprocessing and experiment with proposed averaged cumulative distribution function.

The results indicate that CNN-based methods (CosPlace [11], YOLO) perform better with the proposed preprocessing method, confirming its capability to enhance UAV global localization accuracy. Although CosPlace [11] performs better in fields, forests, and water terrains, the proposed YOLO-based method achieves superior results in urban environments and the targeted terrain due to YOLO11's fine-tuning on segmented building data.

Considering the inherent complexity of UAV global localization, where many approaches struggle to maintain high accuracy at top ranks, a Recall@1 value of 0.195 (or 19.5%) with a localization radius of 3 for urban terrain is a promising outcome. This demonstrates that the YOLO11-based method is competitive and robust to variations in input data. Furthermore, the achieved result surpasses existing methods like CosPlace [11], underscoring the effectiveness of the proposed method at this stage of CNN-based localization research.

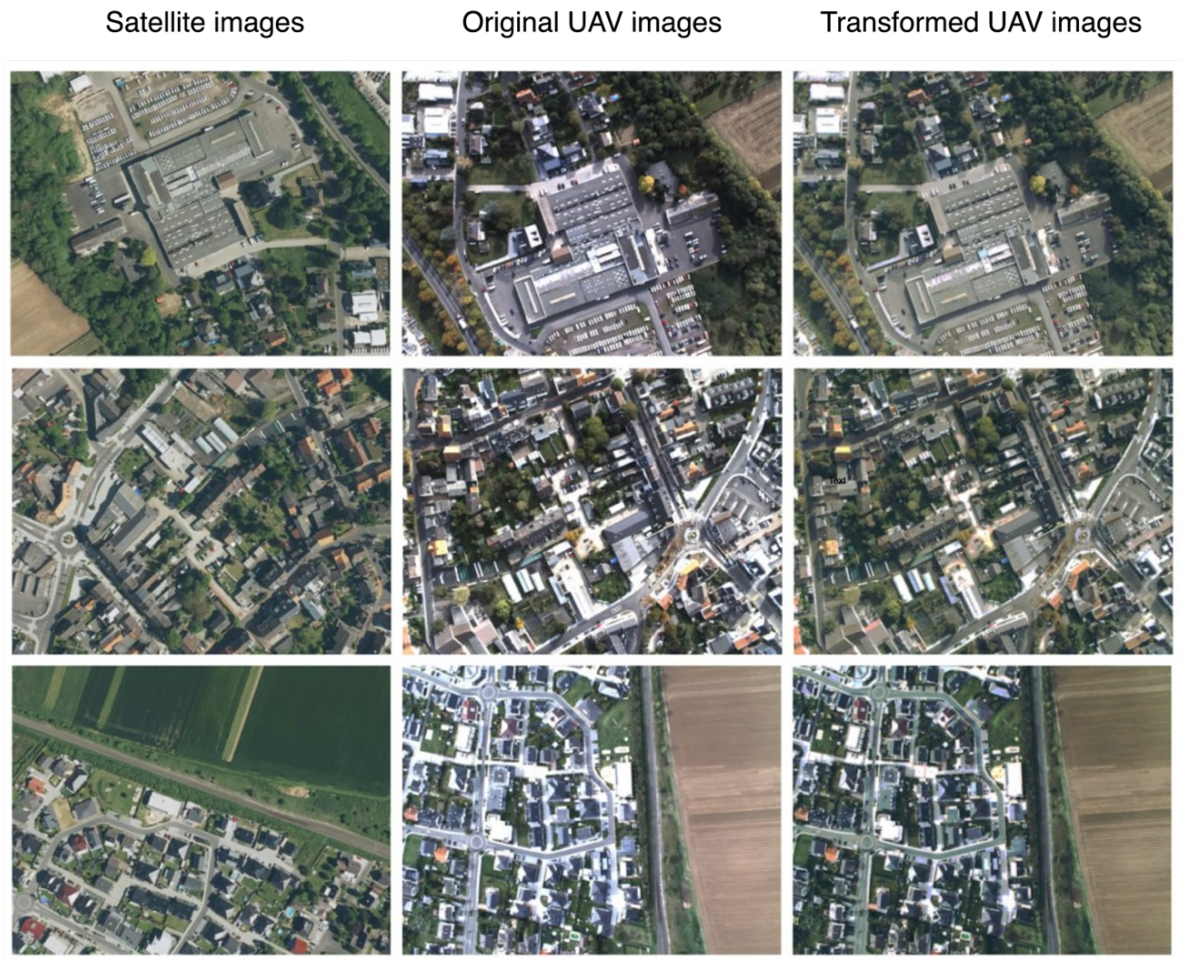

Figure 10: Visual results of the proposed averaged cumulative distribution function (CDF) method applied to UAV images, aligning their statistical distributions with those of satellite images. From left to right: satellite images – original UAV images – transformed UAV images.

The limitations of the proposed UAV global localization method include its applicability restricted to urban terrains during daylight hours without extreme weather conditions, and reliance on a predefined database of satellite images along potential UAV routes.

Table 2
Recall@1 metric results with a localization radius of 3 across different terrain types from the VPAir dataset [22]

Method	Urban	Field	Forest	Water
Without color preprocessing				
CosPlace [11]	0.140	**0.097**	**0.122**	**0.364**
YOLO (Our)	**0.181**	0.049	0.055	0.345
Using greyscale preprocessing				
CosPlace [11]	0.132	**0.093**	**0.114**	**0.366**
YOLO (Our)	**0.175**	0.030	0.038	0.255
Using proposed averaged cumulative distribution function				
CosPlace [11]	0.145	***0.108***	***0.134***	0.374
YOLO (Our)	***0.195***	0.068	0.070	***0.545***

Conclusions

The primary objective of this research was successfully achieved: enhancing the accuracy and robustness of CNN-based methods for UAV global localization in challenging urban environments using models capable of efficient real-time image processing. The fine-tuning of the YOLO11 model

on a dataset of segmented buildings yielded vector representations that significantly improved matching accuracy between UAV images and satellite imagery.

The proposed image preprocessing approach, based on aligning statistical distributions of satellite imagery and UAV-acquired images, demonstrated clear advantages. It enhanced the visual similarity necessary for precise localization, outperforming existing methods such as CosPlace [11], particularly in urban and targeted terrain scenarios. Specifically, the obtained quantitative results include an F1-score of 0.722, demonstrating reliable and consistent building segmentation performance, crucial for effective vector representation generation. Furthermore, a Recall@1 metric of 19.5% with a localization radius of 3 significantly surpasses existing benchmarks in urban terrains, confirming the competitive advantage and robustness of the proposed method.

Key benefits of this research include the ability to perform rapid and accurate UAV localization in GPS-compromised environments, directly addressing critical limitations found in current localization systems. The integration of YOLO11's inherent real-time processing capabilities with improved vector matching techniques represents a practical and efficient solution, particularly suited to real-world applications involving time-sensitive operations, such as rescue missions, urban monitoring, and infrastructure inspections.

Despite these advances, the proposed method's applicability currently remains focused primarily on urban environments during optimal conditions (daylight and favorable weather). Additionally, its effectiveness relies on the availability and quality of a predefined database of satellite images corresponding to potential UAV operational routes.

Future research directions should focus on further enhancing the generalizability and accuracy of the YOLO11 model for broader segmentation and localization tasks through comprehensive image augmentation, architecture modifications, and rigorous hyperparameter optimization. Expanding the scope of localization capabilities to include diverse terrains (e.g., forested areas, fields, and water bodies) would significantly enhance the versatility and operational value of this approach, broadening its practical application spectrum across various UAV mission scenarios.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] C. Masone, B. Caputo, A Survey on Deep Visual Place Recognition, IEEE Access 9 (2021) 19516–19547. doi:10.1109/ACCESS.2021.3054937.

[2] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '10, IEEE Computer Society, 2010, pp. 3304–3311. doi:10.1109/CVPR.2010.5540039.

[3] S. Garg, T. Fischer, M. Milford, Where Is Your Place, Visual Place Recognition?, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, International Joint Conferences on Artificial Intelligence Organization, Montreal, Canada, 2021, pp. 4416–4425. doi:10.24963/ijcai.2021/603.

[4] T. Sattler, et al., Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, IEEE Computer Society, 2018, pp. 8601–8610.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN Architecture for Weakly Supervised Place Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, IEEE Computer Society, 2016, pp. 5297–5307.

[6] A. Torii, J. Sivic, T. Pajdla, M. Okutomi, Visual Place Recognition with Repetitive Structures, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013, IEEE Computer Society, 2013, pp. 883–890.

[7] A. Torii, et al., Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 814–829. doi:10.1109/TPAMI.2019.2941876.

[8] N. Suenderhauf, et al., Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free, in: D. Hsu (Ed.), Robotics: Science and Systems XI, Robotics: Science and Systems Conference, 2015, pp. 1–10.
[9] S. Hausler, S. Garg, M. Xu, M. Milford, T. Fischer, Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, IEEE Computer Society, 2021, pp. 14141–14152.
[10] J. Komorowski, M. Wysoczańska, T. Trzcinski, MinkLoc++: Lidar and Monocular Image Fusion for Place Recognition, in: Proceedings of the 2021 International Joint Conference on Neural Networks, IJCNN 2021, IEEE, 2021, pp. 1–8. doi:10.1109/IJCNN52387.2021.9533373.
[11] G. Berton, C. Masone, B. Caputo, Rethinking Visual Geo-Localization for Large-Scale Applications, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, IEEE Computer Society, 2022, pp. 4878–4888.
[12] A. Ali-bey, B. Chaib-draa, P. Giguère, MixVPR: Feature Mixing for Visual Place Recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, IEEE, 2023, pp. 2998–3007.
[13] A. Ali-bey, B. Chaib-draa, P. Giguère, GSV-Cities: Toward appropriate supervised visual place recognition, Neurocomputing 513 (2022) 194–203. doi:10.1016/j.neucom.2022.09.127.
[14] J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, LoFTR: Detector-Free Local Feature Matching With Transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, IEEE Computer Society, 2021, pp. 8922–8931.
[15] R. Wang, Y. Shen, W. Zuo, S. Zhou, N. Zheng, TransVPR: Transformer-Based Place Recognition With Multi-Level Attention Aggregation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, IEEE Computer Society, 2022, pp. 13648–13657.
[16] N. Keetha, et al., AnyLoc: Towards Universal Visual Place Recognition, IEEE Robot. Autom. Lett. 9 (2024) 1286–1293. doi:10.1109/LRA.2023.3343602.
[17] M. Oquab, et al., DINOv2: Learning Robust Visual Features without Supervision, arXiv:2304.07193 (2024). doi:10.48550/arXiv.2304.07193.
[18] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, IEEE Computer Society, 2016, pp. 779–788.
[19] Ultralytics, YOLO11 NEW, 2025. URL: https://docs.ultralytics.com/models/yolo11.
[20] J. Shao, L. Jiang, Style Alignment-Based Dynamic Observation Method for UAV-View Geo-Localization, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–14. doi:10.1109/TGRS.2023.3337383.
[21] Buildings Instance Segmentation Dataset > Overview, Roboflow, 2025. URL: https://universe.roboflow.com/roboflow-universe-projects/buildings-instance-segmentation/dataset/1.
[22] M. Schleiss, F. Rouatbi, D. Cremers, VPAIR – Aerial Visual Place Recognition and Localization in Large-scale Outdoor Environments, arXiv:2205.11567 (2022). doi:10.48550/arXiv.2205.11567.
[23] O. Rainio, J. Teuho, R. Klén, Evaluation metrics and statistical tests for machine learning, Sci. Rep. 14 (2024) 6086. doi:10.1038/s41598-024-56706-x.
[24] M. Zaffar, et al., VPR-Bench: An Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change, Int. J. Comput. Vis. 129 (2021) 2136–2174. doi:10.1007/s11263-021-01469-5.
[25] ultralytics/ultralytics: Ultralytics YOLO11, 2025. URL: https://github.com/ultralytics/ultralytics.