Semantic Template Matching for Efficient Map-Based UAV Localization

Yogev Ofir[1] Moshe Nahshon[2]

Ariel University
{yogev.ofir, moshe.nahshon}@msmail.ariel.ac.il

Abstract

This paper introduces a novel framework for UAV localization that merges deep semantic segmentation with classical template matching to achieve efficient, real-time map-based Localiza. Our approach utilizes a UNetFormer-based network to transform raw UAV imagery into robust semantic segmentation maps, capturing high-level environmental features such as roads, buildings, and vegetation. These semantic maps reduce sensitivity to transient factors like illumination changes, seasonal variations, and occlusions. By decoupling the segmentation process from the localization step, we apply a correlation-based template matching algorithm guided by UAV metadata to accurately align the segmented view with a pre-annotated reference map. This integration not only minimizes computational overhead but also compensates for challenges such as camera tilt, enabling precise localization even in GNSS denied environments. Experimental results demonstrate that our hybrid method reliably determines UAV positions in real time, making it well-suited for dynamic and complex operational scenarios. Overall, the proposed framework advances UAV navigation by combining the high-level scene understanding of modern deep learning with the efficiency and interpretability of classical matching techniques.

Keywords: UAV, Localization, deep learning, semantic segmentation, UNetFormer, template matching

Contents

1 Introduction

Unmanned aerial vehicles (UAVs) have increasingly become indispensable tools in various domains, including military operations, environmental monitoring, and disaster response. Imagine a drone flying over a city during a disaster response mission its task is critical, yet knowing its exact location remains a challenge. While drones are now essential tools in fields ranging from environmental monitoring to emergency services, accurately pinpointing their position becomes difficult, especially in areas where satellite signals are weak or jammed.

1.1 Limitations of Traditional Navigation Methods

Most drones today rely on a mix of Global Navigation Satellite Systems (GNSS) and inertial sensors. GNSS gives a global fix, and inertial sensors track movement through acceleration data [8, 18]. Together, they work well under normal conditions. But when GNSS signals are disrupted whether by intentional interference or unexpected obstacles the accuracy of these systems can drop dramatically.

1.1.1 Terrain-Aided Navigation (TAN) and Terrain-Referenced Navigation (TRN)

Terrain-Aided Navigation (TAN), also known as Terrain-Referenced Navigation (TRN), presents a promising alternative by leveraging radar altimeter measurements to infer position through comparisons with a digital elevation model (DEM)[17]. This approach, which remains effective under diverse weather conditions, relies on high-precision radar altimeters that, while accurate, tend to be costly and bulky factors that can pose challenges for smaller UAVs.

Furthermore, because the system captures only one-dimensional elevation data, it may encounter ambiguities in terrain matching in regions where elevation changes are minimal. Complicating matters further, errors introduced by slant range uncertainties and the wide-beam characteristics of radar measurements can affect the overall precision of localization

1.1.2 Vision-Based Navigation

Vision-based navigation has emerged as an attractive alternative, capitalizing on lightweight onboard cameras to derive positional information. Early systems in this field extracted visual cues such as edges, corners, and textures from captured images, and then matched these features with corresponding digital terrain models to ascertain altitude and orientation [11, 17].

In practice, however, variability in visual data due to changes in illumination, seasonal shifts, or occlusions can alter the appearance of the terrain, while traditional feature matching techniques like SIFT and SURF may struggle under these conditions, sometimes resulting in mismatches and localization errors. Moreover, the computational demands of processing high-resolution images in real time pose a significant challenge on resource-limited UAV platforms.

1.1.3 Map-Based Navigation

Building on vision-based methods, map-based navigation leverages publicly available, geo-referenced aerial or satellite imagery such as those provided by Google Maps or OpenStreetMap to localize UAVs. In this paradigm,

images captured by the UAV are directly matched with corresponding map imagery [16, 21], eliminating the need for onboard map-building.

However, differences in scale, orientation, and perspective between the UAV's images and the aerial maps present significant registration challenges. Additionally, the infrequent updates of public maps can lead to temporal discrepancies, as seasonal variations or recent changes in the landscape may not be reflected accurately. Traditional image matching techniques further complicate the process by struggling with these real-world variations, occasionally resulting in unreliable localization outcomes.

1.2 Deep Learning and Semantic Approaches

Recent advances in deep learning have transformed UAV navigation by introducing semantic segmentation methods that "understand" an environment at a high level. Traditional methods, which rely on handcrafted features and direct image matching, often struggle with variations in lighting, seasonal changes, and occlusions. In contrast, modern deep learning models especially those based on UNet-like architectures and Transformer modules can segment scenes into meaningful categories such as roads, buildings, vegetation, and more. This semantic abstraction means that even if low-level image details change due to different weather or lighting conditions, the essential structure of the scene remains recognizable.

These deep learning methods provide a robust way to extract consistent environmental cues and have been successfully applied to many vision-based navigation tasks. However, many existing approaches[5] tend to rely on end-to-end models that directly localize the UAV from raw images. While these methods have improved accuracy, they can also be computationally intensive and may require extensive training data to handle all possible variations in the operating environment.

1.3 Our Proposed Approach

Our approach takes a different route by combining the strengths of modern deep semantic segmentation with a classical template matching technique. First, we use a UNetFormer-based network[19] to generate detailed semantic segmentation maps from the UAV's live camera feed. By transforming raw images into high-level semantic representations, we filter out transient effects like illumination changes or minor occlusions, focusing on the stable, structural components of the scene.

Rather than relying solely on a deep network for localization, our method decouples the segmentation process from the matching process. The next step applies a correlation-based template matching algorithm to these semantic maps. This matching is guided by auxiliary metadata such as the UAV's orientation and approximate position which pre-aligns the template with a pre-annotated reference map[7]. This strategy dramatically reduces computational demands compared to a full-scale end-to-end deep learning approach and improves robustness against common issues like camera tilt or slight misalignments.

By merging these two stages, our hybrid method leverages the high-level scene understanding of deep networks while retaining the efficiency and simplicity of classical matching techniques. This balance makes our system particularly suitable for real-time UAV localization in challenging, GNSS-denied environments, offering improved accuracy and adaptability without incurring excessive computational costs.

2 Related Work

Over the past decade, map-based navigation for unmanned aircraft has evolved from classical sensor-fusion and handcrafted feature matching methods to sophisticated deep-learning systems that leverage semantic scene understanding. Early map-based navigation systems relied on the integration of GNSS with inertial sensors and simple map-matching techniques. For example, early frameworks such as the one proposed by Senlet and Elgammal (circa 2011) [14] used stereo imagery in conjunction with road and satellite maps to provide global localization. These methods, however, largely depended on low-level features and manually engineered matching algorithms.

Building on these ideas, cross-view image geolocalization emerged as a promising direction. In Lin, Belongie, and Hays' seminal work on "Cross-View Image Geolocalization" (ICCV 2013)[9], the authors demonstrated that despite dramatic viewpoint differences images captured from ground or aerial perspectives could be matched to

satellite imagery by learning discriminative feature representations. Workman et al (2015). further advanced the field by employing aerial reference imagery to bridge the gap between low-altitude UAV views and high-altitude maps. The availability of large, standardized datasets such as the University-1652 benchmark introduced by Zheng et al (ICCV 2020)[20]. provided a common platform for comparing these methods and pushed the field toward more robust, learning-based approaches. As deep convolutional networks began to dominate computer vision research, they quickly found application in map-based navigation. Researchers replaced handcrafted features (e.g., SIFT or SURF) with learned representations that were more invariant to changes in scale, rotation, and illumination. Early deep-learning–based systems still relied on matching raw image features against aerial maps; however, they were often sensitive to appearance changes caused by seasonal variations or lighting conditions.

More recently, the integration of semantic segmentation into map-based navigation has attracted considerable attention. Semantic segmentation models such as the U-Net architecture introduced by Ronneberger et al (2015)[12]. provided pixel-level class predictions that could abstract away low-level appearance differences. Patents such as US10558864B2[6] demonstrated early efforts to localize images by identifying object regions (e.g., roads and buildings) through semantic segmentation. These ideas were refined further in the past few years by works that combined attention mechanisms and part-based feature learning (e.g., Ren et al., 2024; Quan Chen et al., 2024)[3], which are capable of adapting to non-central targets and scale variations while matching drone-captured views with geo-referenced maps.

Early patents and studies suggested that converting raw images into semantic label maps (where objects such as roads, buildings, and vegetation are identified) could reduce the localization problem to a label-to-label matching task. Kim et al on "Aerial Map-Based Navigation" (2021)[7] exemplified this shift by proposing a system that matches semantic labels between the UAV image and a pre-stored map using relative geometric configurations without relying on altitude or camera models. A further advancement came with approaches that integrate end-to-end deep template matching. For instance, the recent work presented in "Drone Referring Localization" (2022) employs a transformer-based network with an attention mechanism to extract robust, multi-scale semantic templates from UAV images and geo-referenced maps. This method achieves state-of-the-art localization by encoding scene semantics into a low-dimensional feature space that is robust to variations in orientation, scale, and illumination [15]. Arafat et al (2023)[1] provide a detailed review of vision-based navigation techniques for UAVs, categorizing them into map-independent, map-dependent, and map-building-based systems. Their survey discusses the merits and challenges of each approach including optical flow methods, SLAM-based techniques, and map-dependent systems that use public imagery and highlights issues such as computational complexity, sensitivity to environmental changes [1], and the need for robust data fusion. This review underscores the benefits of using high-level semantic cues to overcome many of the limitations inherent in traditional vision-based methods.

Our proposed system falls within the map-dependent paradigm and builds directly on these recent advances. We employ the UNetFormer architecture to generate detailed semantic segmentation maps from the UAV's camera feed. Unlike Kim et al. [7], which relies on polar-coordinate–based label matching, and the deep transformer-based template matching approach of "Drone Referring Localization" [4], our method decouples segmentation from matching. We extract object segmentation from the UAV's camera feed and then apply a template matching algorithm to match these objects against a pre-annotated semantic map of our university campus. This modular design leverages the robustness of deep semantic segmentation while keeping the matching process computationally light. It offers several advantages:

Computational Efficiency, By using traditional template matching on normalization of square difference rather than a heavy end-to-end transformer network, our approach is well suited for resource-constrained UAV platforms. Robustness, The use of high-quality semantic segmentation minimizes the effects of variations in illumination. Simplicity and Interpretability the decoupled pipeline allows easier debugging and adaptation to new environments compared to complex deep template matching systems.

In summary, while recent contributions such as Kim et al. (2021) and the transformer-based method in "Drone Referring Localization" (2022) push the envelope on deep-learning–based map matching, our approach strikes a balance between robustness and efficiency [4]. Moreover, the comprehensive review by Arafat et al. (2023) [1] reinforces the need for semantic methods in UAV navigation an insight that our system directly builds upon by combining advanced semantic segmentation with efficient classical template matching for reliable, real-time localization in GNSS-denied environments.

3 Proposed Map-Based Navigation System

The overall system, depicted in Fig. 1, is composed of three primary modules: (A) Image Processing, (B) Pattern Matching and, (C) Localization. The output of these modules is a robust estimate of the UAV's horizontal position, which when fused with inertial measurements yields accurate real-time estimates of position.

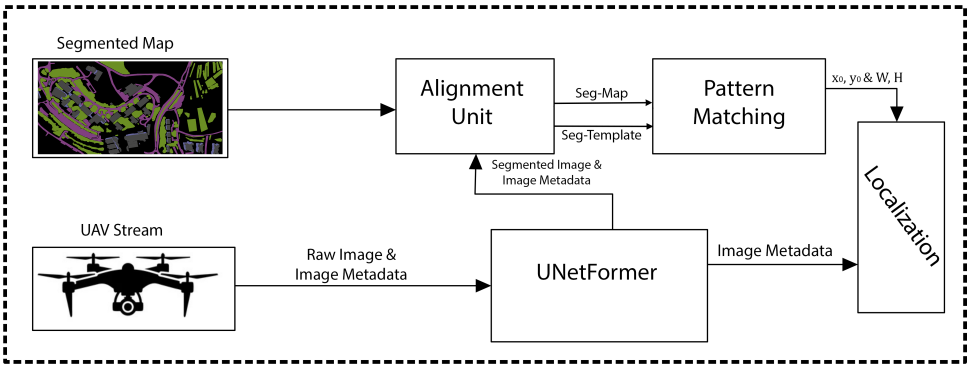

Figure 1: Diagram of the proposed map-based navigation system. The *Alignment Unit* adjusts the template image scale to match its corresponding size on the larger map. Additionally, it rotates the map to align with the UAV's orientation, ensuring accurate template matching.

3.1 Image Processing

The image processing module serves as the first step in our navigation pipeline. Its primary role is to extract high-level semantic information from raw aerial images captured by the UAV. For this purpose, we utilize the UNetFormer architecture [19], a state-of-the-art hybrid Transformer–UNet model. UNetFormer combines the strengths of convolutional neural networks (CNNs) with Transformer modules to achieve a robust semantic segmentation. Like classical U-Net, it consists of an encoder-decoder structure with skip connections. The encoder progressively extracts hierarchical features from the input image, while the decoder reconstructs the spatial resolution using these features. The Transformer modules in UNetFormer enhance the global context modeling, enabling the network to capture long-range dependencies.

The network is trained to classify each pixel of an aerial image into distinct semantic classes (e.g., building, road, vegetation). This process results in a semantic segmentation map that abstracts the raw visual data into high-level labels, which are more robust to variations in illumination, viewpoint, and seasonal changes. In our system, the segmentation output is post-processed to generate a "meta image," where each ground object is represented as group of pixels with a unique color.

By converting the complex raw image into a simplified semantic representation, our system can focus on the spatial configuration of key objects rather than low-level textures. This abstraction not only improves robustness (variations in illumination, viewpoint and seasonal changes) but also reduces the subsequent computational load in the matching phase.

3.2 Pattern Matching

Once the semantic segmentation is obtained, the next step is to determine the UAV's location by matching the generated image against a pre-annotated, geo-referenced semantic map of our operating environment (e.g., a university campus).

Our approach uses a classical template matching algorithm, specifically the `TM_SQDIFF_NORMED` metric provided by OpenCV. In template matching, a smaller template image (in our case, the meta image from the UAV) is slid over a larger search image (the semantic map). At each location, the algorithm computes the normalized squared difference between the template and the corresponding sub-window of the search image.

The metric is defined so that lower values correspond to higher similarity. Mathematically, for a given position

(x, y) in the search image, the `TM_SQDIFF_NORMED` score is computed as

$$R(x,y) = \frac{\sum_{i,j} \left(T(i,j) - I(x+i, y+j)\right)^2}{\sqrt{\sum_{i,j} T(i,j)^2 \cdot \sum_{i,j} I(x+i, y+j)^2}},$$

where $T(i, j)$ is the pixel value of the template and $I(x + i, y + j)$ is the pixel value of the search image. This normalization does indeed reduce the sensitivity to absolute intensity differences.

It is important to note that the `TM_SQDIFF_NORMED` metric itself is not inherently invariant to changes in scale or rotation. In our application, we achieve robustness to these variations not by the matching metric alone, but through preprocessing steps 4.4. Specifically, we convert the raw images into semantic segmentation maps simplified representations that capture the structural layout of the scene (e.g., roads, buildings, vegetation). These semantic maps are then pre-aligned using UAV metadata (yaw angle and approximate position) to compensate for differences in orientation and scale between the template and the reference map.

Once the images are pre-aligned, the template matching algorithm scans the map to find the position where the matching score $R(x, y)$ is minimized, which is then taken as the estimated horizontal location of the UAV. Given that the generated semantic image is a simplified representation with only a few salient features, the template matching process is computationally lightweight, making it well-suited for real-time applications on resource-constrained UAV platforms.

3.3 Localization

While the best-match location from template matching gives the position corresponding to the center of the template, this is only directly valid under the assumption of a nadir (directly downward) camera view. In practice, if the UAV's camera is tilted (i.e., not pointed exactly downward), the projection of the image center on the ground will be shifted relative to the true UAV location. To accurately determine the UAV's ground position, our method incorporates the following theoretical steps:

The best-match region is defined by a rectangle on the geo-referenced map. The center of this rectangle obtained by adding half of the template's width and height to its top-left coordinates is converted from pixel coordinates into map coordinates using a known scale (meters per pixel) and the map's origin.

When the camera is tilted by a pitch angle θ (measured from the nadir) and has a yaw angle ϕ (which specifies the direction of the tilt relative to the map's coordinate system), the true ground point beneath the UAV does not align with the image center. Instead, the horizontal displacement d is given by:

$$d = h \cdot \tan(\theta),$$

where h is the UAV's altitude. This displacement represents the shift from the nadir point the location directly below the UAV to the point where the camera's optical axis intersects the ground.

The displacement d is then decomposed into its x and y components using the camera's yaw angle:

$$\Delta X = d \cdot \cos(\phi), \qquad \Delta Y = d \cdot \sin(\phi).$$

These components represent the shifts in the map's coordinate system that must be applied to the initially estimated position.

The final UAV ground location is then determined by subtracting these offset components from the initially matched map coordinates. In effect, if $(X_{\text{img}}, Y_{\text{img}})$ represents the position corresponding to the center of the best-match rectangle, the corrected UAV location $(X_{\text{UAV}}, Y_{\text{UAV}})$ is computed as:

$$X_{\text{UAV}} = X_{\text{img}} - \Delta X, \qquad Y_{\text{UAV}} = Y_{\text{img}} - \Delta Y.$$

This theoretical framework ensures that even when the UAV's camera is tilted, the system accurately compensates for the resulting projection error. By combining the robust, invariant template matching (using `TM_SQDIFF_NORMED` on the semantic meta image) with a precise geometric projection model, our method yields an accurate and computationally efficient estimate of the UAV's ground location.

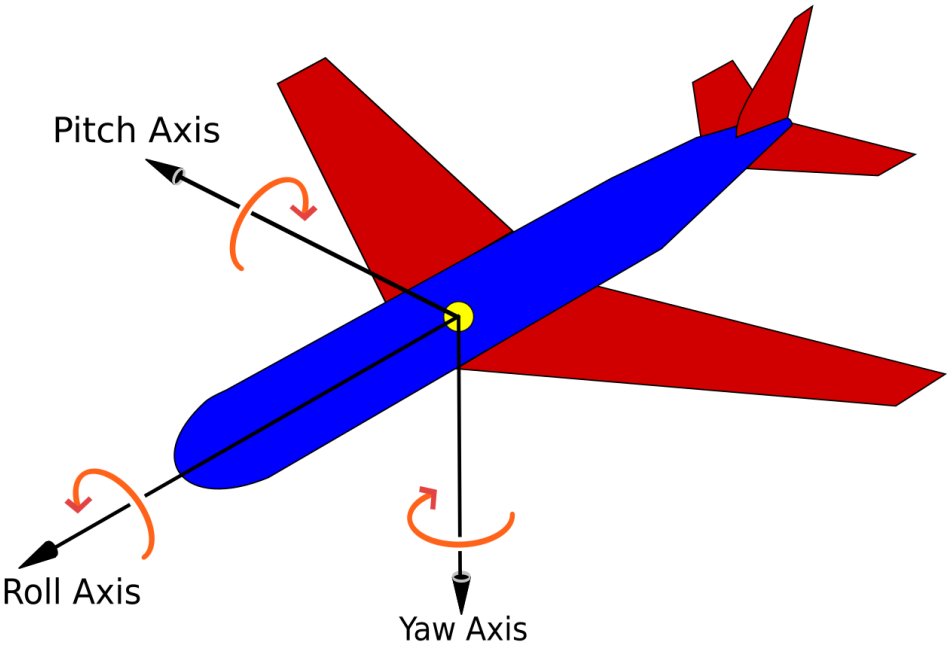

Figure 2: Illustration of the three rotational axes of an aircraft (pitch, roll, and yaw). In our localization framework, the pitch angle θ is measured from the nadir (vertical axis), and the yaw angle ϕ defines the camera's horizontal orientation relative to the map's coordinate system.

4 Experimentation and Results

4.1 Dataset Modifications

Our initial model was trained using the UAVid dataset [10]. However, the segmentation performance on UAVid was suboptimal, yielding an overall mean IoU score of approximately 0.67 when using the UNetFormer model [19]. To address this, we transitioned to the IDD dataset [2], which encompasses UAVid along with VDD and UDD datasets (811 imgaes), providing a more comprehensive labeling scheme. The UAVid classes (Background Clutter, Low Vegetation, Trees, Static Car, Moving Car, Road, Buildings, Human) were re-mapped into refined categories from VDD and UDD (Clutter, Building Wall, Building Roof, Road, Vegetation, Cars, Water). When testing the IDD-trained model on our university data, we observed unsatisfactory results due to significant differences between our university's landscape and the urban environment represented by IDD. The primary issues were: Building Structure, the architectural styles of our university buildings were distinctly different from urban buildings. Road and Sidewalk Patterns, Urban landscapes generally have consistent road and sidewalk structures, whereas in our university, some sidewalks function as roads, and certain roads are occasionally dirt paths. To mitigate these discrepancies, we manually labeled an additional 20 images from our university campus using LabelMe [13] and incorporated them into the IDD dataset. Additionally, we removed the "car" class, as vehicles are dynamic objects and irrelevant to static localization. These modifications led to a significant improvement, raising the overall mean IoU to approximately 0.81.

4.1.1 Resolution-Based Augmentation

To enhance our dataset further, we introduced images at lower resolutions (720p and 360p) through a downscaling and upscaling process applied to 4K images. This preserved spatial dimensions while reducing details, effectively tripling our dataset size and increasing its diversity. Our motivation stemmed from the need for lower-resolution images to facilitate faster inference.

Resolution	Model	F1	mIoU	Time(s)
4k	1024×1024 + down&up scaling	**89.87**	**81.81**	**14.41**
4k	1024×1024 + fine-tuned	77.16	63.34	14.66
4k	256×256	69.24	54.26	14.79
1080p	1024×1024 + down&up scaling	**84.55**	**73.56**	**6.58**
1080p	1024×1024 + fine-tuned	81.74	69.27	6.59
1080p	256×256	79.61	66.17	6.66
720p	1024×1024 + down&up scaling	76.36	62.72	**4.56**
720p	1024×1024 + fine-tuned	83.43	71.13	4.74
720p	256×256	**83.77**	**71.62**	4.88
540p	1024×1024 + down&up scaling	69.20	54.38	2.13
540p	1024×1024 + fine-tuned	83.15	72.52	2.15
540p	256×256	**84.38**	**73.51**	**2.099**
360p	1024×1024 + down&up scaling	57.86	42.53	**1.89**
360p	1024×1024 + fine-tuned	78.43	64.15	**1.89**
360p	256×256	**79.81**	**66.78**	1.95

Table 1: Performance of Models at Different Resolutions on a batch of 18 images, "Time", refers to inference time.

4.2 Model Variations and Performance

All models in the experiments were implemented with the PyTorch framework on a single NVIDIA RTX 4070 TI Super GPU. Our experimental setup included three distinct models, each designed to improve segmentation performance while balancing inference time. The Scaling Model (1024 + DownUp Scaling) was trained on IDD [2] and our university data (excluding the "car" class). It incorporated a downscaling step to 720p and 360p, followed by upscaling back to 4K, aiming to enhance performance on lower-resolution images. However, the results showed relatively low mean IoU scores for lower-resolution images, indicating that this approach did not significantly aid segmentation performance. To address this limitation, we developed the Fine-Tuned Model (1024 + FT) by building upon the Scaling model and fine-tuning it using resized 256×256 patches over 20 additional epochs. This refinement improved segmentation accuracy on lower-resolution images, as transformer models rely heavily on input size for context representation. By explicitly training on smaller patches, the model adapted better to lower-resolution data. Lastly, we experimented with the 256 Model, which was trained entirely using resized 256×256 patches, excluding the original 4K training weights. This model slightly outperformed the fine-tuned model, likely due to the absence of high-resolution training data, which may have interfered with learning the contextual relationships in lower-resolution images more effectively.

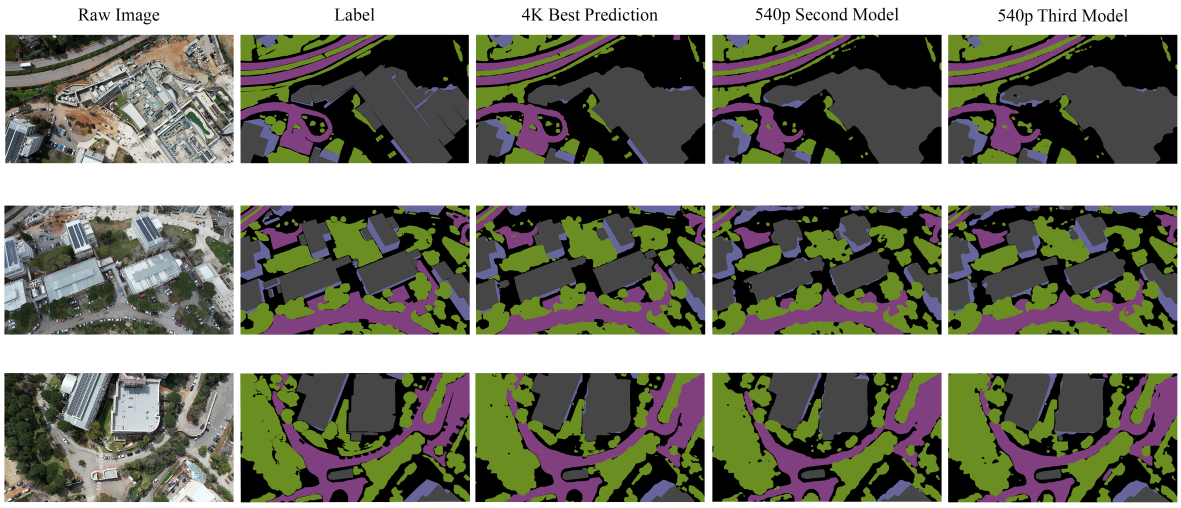

Figure 3: Comparison of ground truth labels with the best segmentation result obtained using the 4K model, alongside the second and third models trained on lower resolutions for real-time applications. Despite minor differences compared to the 4K model, both lower-resolution models are effective for template matching.

4.2.1 Trade-Off Between Resolution and Inference Time

Table 1 highlights a key observation: there exists an inherent trade-off between image resolution and inference speed. Higher-resolution images provide better predictions but significantly increase inference time. Lower-resolution images, on the other hand, facilitate faster inference but degrade segmentation performance. Our objective was to identify the optimal balance achieving adequate segmentation accuracy while maintaining real-time feasibility.

Initially, inference on 4K images was performed using patches of size 1152×1024 to reduce inference time. However, this approach led to suboptimal predictions, particularly at patch borders due to the context-based nature of transformer models. Conversely, using full-image patches improved predictions but significantly increased inference time, making it unsuitable for real-time applications. This challenge further reinforced the necessity of reducing image resolution to enhance inference speed.

4.3 Template Matching

To localize the UAV's semantically segmented template image within a larger semantic map, we conducted a series of experiments with different matching techniques. The goal was to find where the small template (representing the UAV's local view) appears in the big map (representing the global environment). We summarize these approaches and their outcomes below.

4.3.1 Homography-Based Matching (Global Search)

The first approach attempted was a feature-based homography alignment. In this method, keypoints from the template and the entire large map were extracted and matched, and a homography (projective transformation) was computed to see if the template could be aligned to a subset of the map. The idea was that if the template truly exists as a sub-image of the map (possibly rotated or scaled), a consistent homography could be found via RANSAC using the matched feature points. However, this homography method failed when applied directly to the full map without any spatial restriction. The large semantic map contained many repetitive patterns and lacked distinctive features (since semantic segmentation images consist of broad uniform regions for each class). As a result, feature matching produced spurious correspondences, and the homography estimation often converged to an incorrect solution. In essence, searching the entire map at once allowed too many false matches, preventing the algorithm.

4.3.2 Homography with Sliding Window

To improve the reliability of the homography approach, we next introduced a sliding window strategy. Instead of searching the whole map in one go, the map was scanned by considering smaller sub-regions (windows), and the homography-based matching was applied locally to each window. The expectation was that by limiting the search area, we might reduce false matches and find a correct alignment in one of the windows. Unfortunately, this strategy also failed. Scanning with a sliding window greatly increased the computational cost (since a homography had to be computed for many overlapping sub-regions), and it still did not guarantee a correct match. The fundamental issue remained that the segmented images did not have enough robust and unique keypoint features for the homography method to latch onto. Even within smaller windows, the feature matching could not consistently distinguish the correct location from other similar-looking areas of the segmentation map. Thus, the homography-with-sliding-window approach was neither efficient nor effective in this scenario.

4.3.3 Contour and Feature Descriptor Methods

We then explored other classic image matching techniques, including shape-based and feature-descriptor-based methods, again often combined with a sliding window search. In the contour-based approach, the outline (contour) of the segmented template was extracted, and we attempted to find a similar contour in the larger map. This involved scanning the map for contours and using shape similarity measures (e.g., Hu moment based shape matching) to compare each region's contour to the template's contour. This method did not succeed: the map contained multiple regions with similar shapes, and the orientation mismatch further confused the shape comparison, leading to no reliable identification of the correct location. We also investigated feature descriptor approaches using Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB). Keypoints and descriptors were computed on the template and on candidate sub-regions of the map (using a sliding window or keypoint search across the map). These feature-based methods also failed to locate the template. The primary reason is that the semantic segmentation images have large uniform colored regions (each representing a class like road, building, etc.), which yield very few distinctive keypoints. SIFT and ORB, which excel on natural images with textured details, found either no matches or many unstable matches on the segmentation maps. Any matches that were found were often false positives scattered around the map, and none produced a coherent transformation to correctly place the template. In summary, neither contour matching nor SIFT/ORB feature matching was effective for this task, especially under the necessary exhaustive search of the large map.

4.3.4 Brute-Force Rotation Search

Given the difficulties above, a more brute-force strategy was attempted wherein we explicitly searched over possible rotations of the template. The UAV's local view could be rotated relative to the orientation of the global map, so this approach involved rotating the template image in fixed increments (we used 5° steps from 0° to 355°) and performing a template match for each rotation. Essentially, for each rotated version of the template, a normalized cross-correlation or difference matching was done over the entire map to find the best match location. This brute-force rotation search had partial success: by trying many orientations, we occasionally found a rotation that approximately aligned the template correctly, yielding a noticeable match in the map. In cases where the UAV's orientation happened to be near one of the tested angles, the template would correlate well with the correct region in the map, and thus the method could identify the right location. However, this approach was extremely inefficient. Scanning the large map for each of dozens of rotations is computationally intensive, leading to very slow performance. Moreover, the success was not guaranteed if the true orientation did not exactly coincide with one of the 5° increments, or if slight scale differences existed, the matching might still fail or produce ambiguous results. In practice, this method was not viable for real-time use, though it demonstrated that incorporating orientation alignment was important for solving the problem.

4.3.5 Metadata-Assisted Template Matching (Final Approach)

The breakthrough came from utilizing the UAV's metadata (auxiliary information about the UAV's state) to guide and constrain the template matching. Instead of blindly searching the entire space of rotations and positions, we used the UAV's known pose to pre-align the images. Specifically, the UAV's heading (yaw angle) was used to

rotate the global map to match the orientation of the UAV's camera. Additionally, the UAV's altitude was used to match the scale between the template and the global map, ensuring consistent resolution.

Once the template and map were thus roughly aligned, we employed OpenCV's *matchTemplate* function with the normalized squared difference method TM_SQDIFF_NORMED. This correlation-based algorithm automatically "slides" the template over the (now properly oriented) map and computes the pixel-wise squared difference between the two at each position. The location where this difference is minimized (after normalization) is identified as the best match.

Because the images were already well aligned, the template matching could very quickly pinpoint the exact placement of the template on the map. This final approach proved both highly accurate and efficient. In our experiments, it achieved a 100% success rate (in range 30 to 100 meter altitude, with overall vertical-ish angle) in correctly locating the template within the map for all test cases, with computation time dropping below 1/4 second a dramatic improvement over the earlier brute-force and sliding-window methods. By leveraging metadata to simplify orientation and scale alignment, we avoided the pitfalls of false matches and heavy computation, making the template matching reliable enough for real-time use. This successful strategy highlights the benefit of incorporating domain-specific knowledge (UAV pose metadata) into the image-matching process, yielding real-time performance for template localization.

4.3.6 Enhanced Semantic Masking for Improved Matching

In the final iteration of our template matching approach, we incorporated additional semantic pre-processing to further enhance the matching robustness. Specifically, we modified the semantic masks by ignoring the "trees" class and merging the "building_roof" and "building_wall" classes. This change was implemented in our processing pipeline as follows:

Ignoring Trees: Trees often exhibit high intra-class variability and can change appearance with seasonal variations. They also tend to be spatially scattered and less geometrically consistent compared to man-made structures. By excluding the "trees" class, we reduced the amount of noise and variability in the semantic maps. This exclusion ensured that the template matching process focused on more stable and distinctive landmarks rather than being distracted by the fluctuating patterns of vegetation.

Merging Building Classes: In urban or campus environments, buildings are key landmarks for localization. However, the distinction between "building_roof" and "building_wall" is not always critical for the matching task, and maintaining separate masks for each class could lead to redundant or fragmented information. By uniting these two classes into a single "building" mask, we increased the contrast and coherence of the building regions in the semantic maps. This merged representation provided a more consistent and robust geometric structure for the matching algorithm to identify, thereby reducing false positives.

These semantic masking modifications, when combined with the metadata-assisted alignment and the efficient TM_SQDIFF_NORMED-based template matching, resulted in a significant improvement in both accuracy and speed. The refined masks enhanced the signal-to-noise ratio by emphasizing stable, large-scale structures (e.g., roads and buildings) and removing variable elements (e.g., trees), which in turn contributed to the observed 100% success rate in our final experiments.

4.4 Alignment Unit

One of the key challenges in our localization pipeline was ensuring that the UAV's captured template image and the reference map were properly aligned before performing template matching. Initially, we attempted to rotate the template image itself to match the orientation of the map. However, this approach introduced several problems.

First, when the template image is rotated, parts of the content may shift outside the original bounding box. If the matching window remains fixed at its original dimensions, relevant portions of the template might get cut off. Expanding the window size to accommodate the rotated template would introduce another issue empty regions (caused by the rotation) would require padding, which would interfere with the matching algorithm.

To overcome these issues, we shifted our approach and instead rotated the reference map rather than the template. By rotating the large map to align with the UAV's orientation and filling the newly created empty areas with a neutral color (white, which is not part of the segmented classes), we ensured that the template could match the map without being distorted or truncated (see example in fig 4).

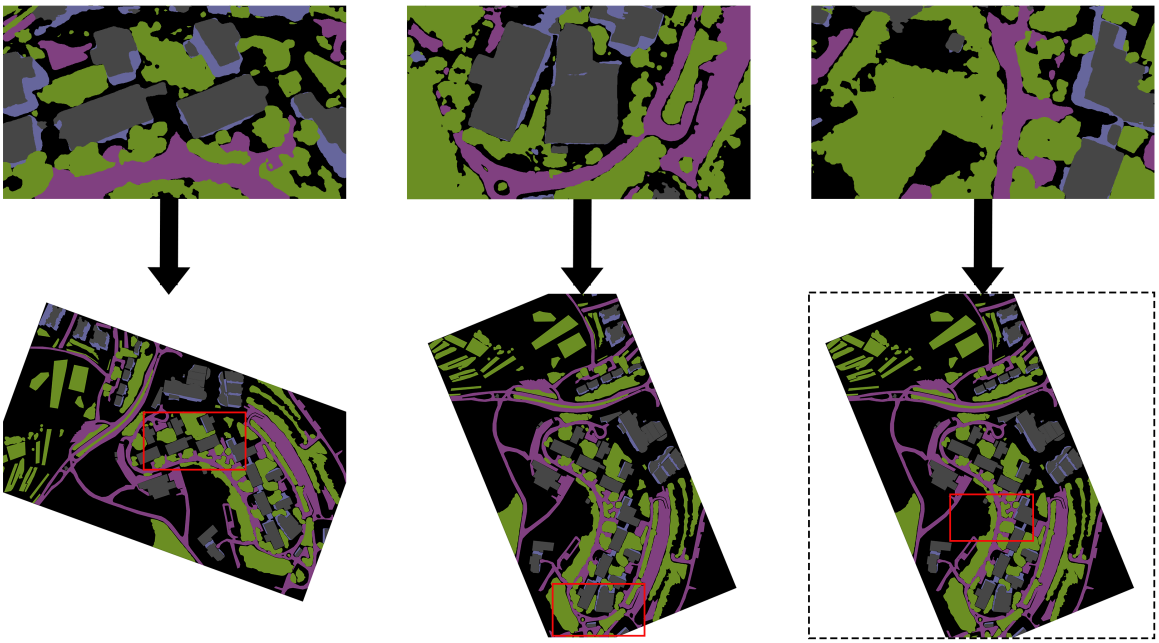

Figure 4: Illustration of the correlation-based template matching pipeline on semantically segmented images after the Alignment Unit. In each example (left, center, right), the UAV's segmented view (top) is aligned with the corresponding region of the reference map (bottom). The dashed rectangle indicates the full extent of the aligned map, ensuring that both orientation and scale match correctly for accurate template matching.

After rotation, we adjusted the scale of the template image to match its corresponding size on the map. This step was crucial, as any discrepancies in scale would have led to mismatches in the template matching process. With the template now properly scaled and aligned with the rotated map, the template matching algorithm performed optimally, yielding precise localization results.

5 Conclusion and Future work

In this paper, we introduced a novel framework for UAV localization that integrates deep semantic segmentation with classical template matching. By leveraging a UNetFormer-based network, our method transforms raw UAV imagery into high-level semantic segmentation maps that capture stable environmental features—such as roads, buildings, and vegetation—while being robust to variations in illumination, seasonal changes, and occlusions. This semantic abstraction simplifies the matching process and reduces the computational load, which is critical for real-time applications on resource-constrained UAV platforms.

Our experimental evaluations highlight the trade-offs between image resolution and inference speed, and demonstrate that our metadata-assisted template matching pipeline—when combined with a robust Alignment Unit—can accurately determine the UAV's ground position even in GNSS-denied environments. Through a series of comparative experiments, we showed that traditional matching techniques (such as feature-based homography and contour matching) are less effective on semantic maps, whereas our approach achieves a 100% success rate in identifying the correct location within the reference map.

Looking ahead, there are several promising directions for future work. First, we believe that incorporating a larger and more diverse set of annotated drone images will further enhance the performance of our semantic segmentation model. While our current work provides a strong theoretical foundation and partial implementation of the complete localization system, future efforts should focus on integrating the individual components into an end-to-end, real-time operational pipeline. This system would stream low-resolution UAV images into the segmentation model, process them through the Alignment Unit, and subsequently feed both the segmented map and template into the template matching module. The matching results, combined with UAV metadata, would then drive the final localization module to accurately determine the drone's position.

Moreover, additional research is warranted to evaluate the system's performance on very close-range images (e.g., altitudes below 30 meters) and under extreme camera angles (approaching or exceeding 45° from nadir). Finally, automating the generation of annotated maps from satellite imagery could further streamline deployment by reducing the manual effort currently needed for map creation.

Overall, our work lays a robust foundation for efficient, map-based UAV navigation, and the proposed future directions promise to enhance both the accuracy and real-time applicability of the system in diverse operational scenarios.

References

[1] Muhammad Yeasir Arafat, Muhammad Morshed Alam, and Sangman Moh. Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges. *Drones*, 7(2):89, 2023.

[2] Wenxiao Cai, Ke Jin, Jinyan Hou, Cong Guo, Letian Wu, and Wankou Yang. Vdd: Varied drone dataset for semantic segmentation. *arXiv preprint arXiv:2305.13608*, 2023.

[3] Quan Chen, Tingyu Wang, Zihao Yang, Haoran Li, Rongfeng Lu, Yaoqi Sun, Bolun Zheng, and Chenggang Yan. Sdpl: Shifting-dense partition learning for uav-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[4] Ming Dai, Enhui Zheng, Jiahao Chen, Lei Qi, Zhenhua Feng, and Wankou Yang. Drone referring localization: An efficient heterogeneous spatial feature interaction method for uav self-localization. *arXiv preprint arXiv:2208.06561*, 2022.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[6] Zehua Huang, Pengfei Chen, Panqu Wang, and Ke Xu. System and method for image localization based on semantic segmentation, February 11 2020. US Patent 10,558,864.

[7] Youngjoo Kim. Aerial map-based navigation using semantic segmentation and pattern matching. *arXiv preprint arXiv:2107.00689*, 2021.

[8] Youngjoo Kim, Kyungwoo Hong, and Hyochoong Bang. Utilizing out-of-sequence measurement for ambiguous update in particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 54(1):493–501, 2017.

[9] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[10] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020.

[11] Jeffrey J. Rodriguez and JK Aggarwal. Matching aerial images to 3-d terrain maps. *IEEE Transactions on pattern analysis and machine intelligence*, 12(12):1138–1149, 1990.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[13] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77:157–173, 2008.

[14] Turgay Senlet and Ahmed Elgammal. A framework for global vehicle localization using stereo images and satellite and road maps. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*, pages 2034–2041. IEEE, 2011.

[15] Zachary Seymour, Kowshik Thopalli, Niluthpol Mithun, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Maast: Map attention with semantic transformers for efficient visual navigation. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13223–13230. IEEE, 2021.

[16] Mo Shan, Fei Wang, Feng Lin, Zhi Gao, Ya Z Tang, and Ben M Chen. Google map aided visual navigation for uavs in gps-denied environment. In *2015 IEEE international conference on robotics and biomimetics (ROBIO)*, pages 114–119. IEEE, 2015.

[17] Dong-Gyu Sim, Rae-Hong Park, Rin-Chul Kim, Sang Uk Lee, and Ihn-Cheol Kim. Integrated position estimation using aerial image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 24 (1):1–18, 2002.

[18] Philip Spiegel, Johann Dambeck, and Florian Holzapfel. Slant range analysis and inflight compensation of radar altimeter flight test data. *Navigation: Journal of The Institute of Navigation*, 63(4):493–509, 2016.

[19] Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, and Peter M Atkinson. Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:196–214, 2022.

[20] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM international conference on Multimedia*, pages 1395–1403, 2020.

[21] Xiangyu Zhuo, Tobias Koch, Franz Kurz, Friedrich Fraundorfer, and Peter Reinartz. Automatic uav image geo-registration by matching uav images to georeferenced image data. *Remote Sensing*, 9(4):376, 2017.