The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W9-2025
ISPRS Intl. Workshop “Photogrammetric and computer vision techniques for environmental and infraStructure monitoring, Biometrics and Biomedicine”
PSBB25 , 9–11 June 2025, Moscow, Russia

Neural Network-Driven UAV Course Correction Using Camera Images

Ivan Alyrchikov[1], Nikolai Moiseev[1], Vladimir Knyaz[1,2]

[1] State Research Institute of Aviation System (GosNIIAS), 125319 Moscow, Russia - (alyrim,moikolia)@gosniias.ru
[2] Moscow Institute of Physics and Technology (MIPT), Moscow, Russia - kniaz.va@mipt.ru

Keywords: UAVs navigation, UAV Course Correction, landmark Recognition, limited Data Learning, semantic segmentation, synthetic data generation

Abstract

This paper presents a UAV course correction algorithm leveraging a neural network and the analysis of a video stream from the onboard camera. The algorithm is designed to identify key landmarks efficiently and requires only a limited set of training images. Significantly, it demonstrates operational capabilities at viewing angles and altitudes that differ from those used during training. Experimental results indicate that the algorithm achieves satisfactory landmark recognition accuracy even with substantial perspective deviations, thus enhancing the robustness and effectiveness of UAV operations.

1. Introduction

Currently, airspace is becoming an increasingly relevant area of application for commercial UAVs, and the need for uninterrupted UAV positioning in various natural and man-made environments is also growing.

However, signal loss or GPS positioning inaccuracies are a common problem faced by UAVs, which can lead to collisions with infrastructure, deviations from the route, or loss of payload. This can be caused by poor weather conditions or interference from other signal sources. Therefore, UAVs require an additional motion correction system in situations where a stable GPS signal is lost.

This work proposes an algorithm that allows for the identification of reference areas, which can be used to correct the direction of UAV movement, provided that there is a very limited number of images of reference objects and that the angles and flight altitudes that need to be corrected differ from the available images.

Part of the work involved determining the minimum necessary average number of such images required per object, as well as the limitations on the change in approach angle.

2. Related work

Autonomous navigation and orientation of Unmanned Aerial Vehicles (UAVs) in challenging environments is an area of interest. Traditional methods often rely on Global Navigation Satellite Systems (GNSS), such as GPS; however, their reliability decreases in indoor environments, urban settings, and other areas with limited sky visibility. Consequently, visual methods using onboard cameras have emerged as a compelling alternative to estimate orientation and facilitate navigation.

Optical flow, representing the apparent motion of image patterns, serves as a valuable source of information regarding camera movement, and thus, UAV motion.

Early works by (Horn and Schunck, 1981) and (Lucas and Kanade, 1981) explored the application of optical flow to estimate speed and direction of motion. These approaches frequently employ hand-crafted or classical computer vision techniques for optical flow computation, which are susceptible to noise, variations in lighting conditions, and geometric distortions.

In contrast, (Kim, 2021) and (Kumar et al., 2022) proposed a novel map-based navigation system for UAVs. This system prioritizes label-to-label matching, as opposed to image-to-image matching, between aerial imagery and a map database. Through the use of semantic segmentation, ground objects are labeled and their spatial configuration is used to locate the corresponding position within the map database. Despite these advancements, there are notable limitations associated with utilizing optical flow for UAV orientation, including the method’s reliance on the accuracy and dependability of the semantic segmentation algorithm, the requirement for a comprehensive and accurate map database containing labeled objects, scalability challenges in extensive and intricate regions, and discrepancies between simulated and real-world conditions caused by noise, distortions, and other factors affecting image quality.

(Sundar et al., 2017) propose an approach to UAV navigation and routing in the absence of a GPS signal, based on a combination of cooperative localization and routing. They validate the effectiveness and performance through simulations. The approach is characterized by its novelty, and the formulation of the problem as a combinatorial optimization problem allows for the use of rigorous methods to find optimal solutions. However, a drawback of the proposed method is that combinatorial optimization problems are often computationally complex. The paper does not discuss the scalability of the proposed approach for large areas and complex scenarios. The optimality of the resulting paths is only guaranteed within the framework of the mathematical model used, and real-world conditions may present unmodeled factors that could impact system performance.

(Miller et al., 2017) present an innovative approach to UAV navigation. Instead of traditional methods based on GPS or inertial sensors, they propose using the analysis of “optical flow” - the movement of pixels in images captured by an onboard video camera. A key feature of their approach is the direct relationship between optical flow and the UAV’s motion elements

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W9-2025-1-2025

(velocities). This allows them to calculate the aircraft’s current speed and direction of movement by analyzing image changes. Decomposing the problem into calculating optical flow for a given motion (the forward problem) and reconstructing motion from optical flow (the inverse problem) helps structure the solution. Applying statistical estimation methods to solve the inverse problem can improve the reliability and robustness of the algorithm to noise and data errors. However, the assumption of a linear relationship between optical flow and motion elements may be an oversimplification and not always hold true in real-world conditions with perspective and camera distortions.

A research team led by (Qian et al., 2023) developed a new real-time UAV navigation technology based on image scanning and object recognition, utilizing MapBotix software and adaptive algorithms (Faster R-CNN and Mask R-CNN). A key aspect is the application of a modified Mask R-CNN algorithm (with a specific layer structure) for analyzing scanned object images. The emphasis on real-time navigation is important because it enables UAVs to respond quickly to changing conditions. Using object recognition for navigation allows UAVs to orient themselves in space by understanding their environment, rather than relying solely on coordinates. The inclusion of change detection expands the UAV’s capabilities and makes it useful for monitoring and inspection tasks.

(Zalesky and Shuvalov, 2017) developed an autonomous navigation algorithm for UAVs that allows them to return to the starting point or follow a route solely based on data from an onboard video camera, without relying on GPS or other external navigation systems. The algorithm compares the current image with previously stored frames or maps of the area to determine its position and correct its course. This makes navigation independent of external signals, increasing its robustness in conditions where GPS is unavailable.

(Deuser et al., 2023) propose a method for geolocating UAV images without GPS, based on orientation-guided training. They utilize hierarchical localization to estimate the orientation of UAV images relative to satellite imagery. A lightweight orientation prediction module, leveraging contrastively learned embeddings, is developed. This orientation prediction enhances training and outperforms existing methods. Data augmentation via rotation of satellite images improves generalization. The orientation module is not required during inference, reducing computational costs. The approach is dependent on the quality and availability of satellite imagery.

While these approaches demonstrate potential, they are not sufficient for fully performing the navigation task when control signal problems arise.

3. Algorithm

The research aims to develop an algorithm, based on neural networks, that provides data for correcting the UAV’s flight path. The input for the algorithm is a video stream from the UAV’s camera. The output is the detection of the desired reference areas in the input images, based on which, using various methods, a decision is made to correct the direction of movement.

For reliable identification and detection during flight, it is necessary to pre-collect a minimally sufficient number of images of the reference areas, taken from altitudes and perspectives that differ from those of the verification flights, within an acceptable range. Threshold values for these ranges and the composition of the necessary database are discussed later in the article.

3.1 Database collection

To train the neural network, a comprehensive and high-quality database/dataset needs to be collected. For this task, three-dimensional modeling in Blender and Unreal Engine 5 was used in this work. Blender allowed us to build three-dimensional models and then export them to Unreal Engine, where a scene with the selected area is assembled. The approach has shown promising results when applied to this kind of problem in the work (Alyrchikov et al., 2024).

In the initial stage, three areas are identified that are suitable in terms of the number of required reference regions. The main features of such areas are their distance from each other, as well as the heterogeneity or complexity of the placement pattern. An example of one of these areas is shown in Figure 1.

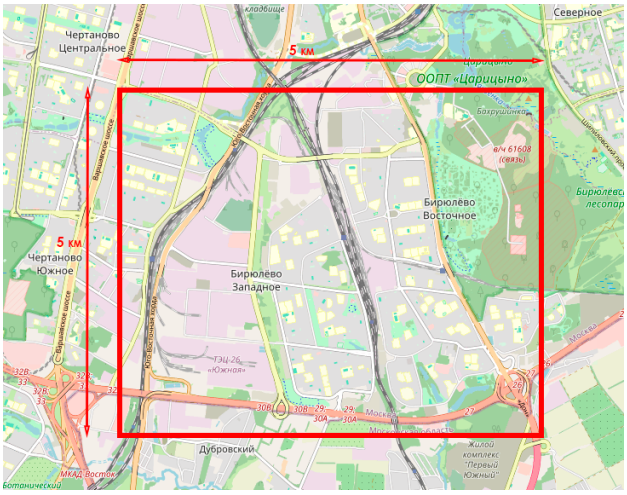

Figure 1. Example of a selected district for creating a scene in Unreal Engine 5

For scene creation, the open-source mapping data from OSM (Open Street Map) was used. Infrastructure objects were loaded from OSM into Blender in primitive form and then transferred to Unreal Engine. As a result of this approach, three scenes were created for training. An example of such a scene, measuring 5x5 km, is shown in Figure 2.

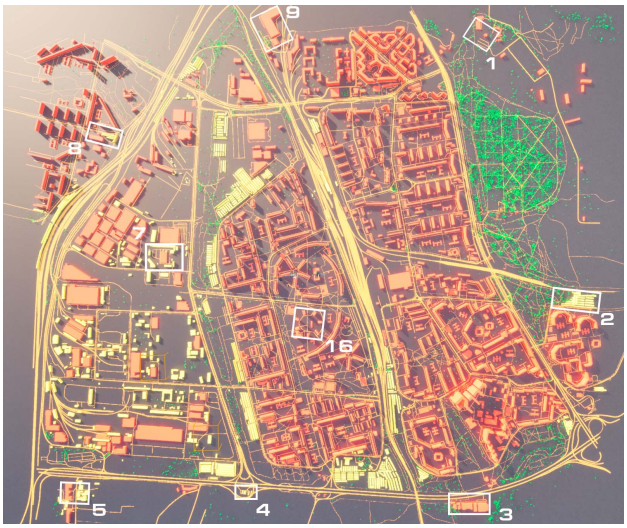

Figure 2. Example of creating a 3D scene

For each reference area, a main trajectory was defined, simulating the flight of a UAV toward the area. To simulate various viewing angles characteristic of real-world shooting conditions, additional trajectories were generated, deviating from the main trajectory by angles within ±2.5, 5, 8, 10, 12, and 15 degrees. In addition to changing the approach angles, the UAV altitude was also varied within ±10 percent of the initial flight altitude, which made it possible to account for the influence of changes in object scale and perspective distortions on the quality of recognition. All flights towards the reference area were conducted at a distance of 1 to 2 kilometers from the target area. An example reference area and the main UAV approach trajectory, as illustrated in Figure 3 (top-down view).

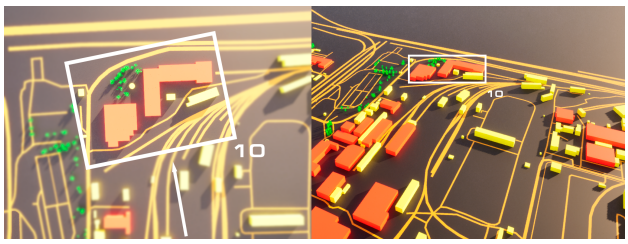

Figure 3. The left part of the image is an example of a support area and the direction of the main trajectory of the UAV approaching it from a top view, on the right is a picture obtained by the UAV during the flight

In this work, the task of detecting and identifying reference areas is solved using semantically segmented images. The use of semantic segmentation improves the accuracy and quality of the algorithm, as separating the task into semantic segmentation and subsequent region search allows for the use of features that would be difficult to access when analyzing the original images. Within the scope of this work, it is assumed that the semantic segmentation step is performed automatically; therefore, the specific methods and algorithms used for this are not considered in detail. Objects in the scenes are divided into 5 classes:

1. Buildings
2. Garages
3. Trees
4. Roads
5. Ground

To create the simulated database of semantically segmented images, a special algorithm was developed in Unreal Engine to generate images already containing semantic segmentation information. An example of a segmented image is shown in Figure 4.

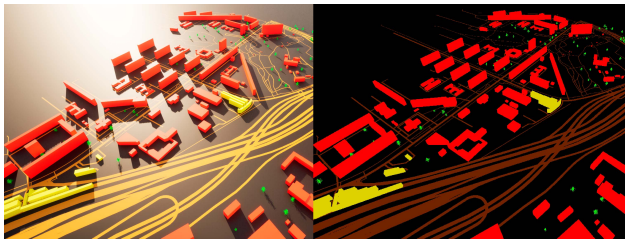

Figure 4. On the left is an image from an Unreal Engine scene, on the right is a segmented image

During the operation of the semantic segmentation algorithm implemented in Unreal Engine 5, two main types of data are generated for each frame. In addition to the directly segmented image, where each pixel is assigned to a specific class, an image is generated on which objects of interest for determining the location of the reference area are highlighted (labeled). This approach enables the creation of a detailed visualization of the target objects in the context of the entire scene, providing clarity to the segmentation results.

A key feature of this approach is the use of a contrasting color scheme to highlight objects of interest. The auxiliary image is created in such a way that the identified reference objects are rendered exclusively in white, while all other elements of the scene that are not of direct interest are displayed in black. This binary color differentiation allows for a clear and unambiguous visual representation of the contours of the target objects, which significantly simplifies the task of their subsequent identification and localization.

Although the use of a contrasting color scheme ensures clear contours, information about the mutual overlap of objects is preserved when rendering the auxiliary image. This means that if the target reference object is partially occluded by another element of the scene, the occlusion will also be displayed in the final image.

Using a developed script that analyzes object location information obtained during rendering, the annotation process is automated. The script analyzes the coordinates of each labeled reference object in the frame, providing accurate data about its position in the image. This information serves as the basis for further training of the neural network. Thus, the combination of semantic segmentation, selective rendering, and scripted analysis provides an efficient and automated process for annotating images for the needs of the navigation algorithm.

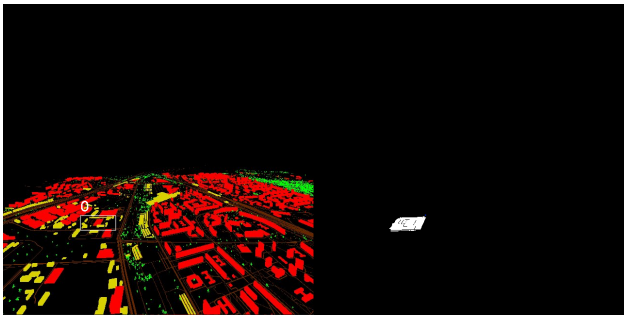

Figure 5. Example of annotation and highlighting of a reference area with a circumscribing rectangle

Twenty locations were selected as reference areas. For training the system, data collected for 16 of these areas was used, including images obtained from various angles and altitudes. The remaining 4 reference areas were reserved for testing and validating the developed algorithm.

3.2 Neural network training

To achieve acceptable training quality with a limited amount of data, a two-stage training approach for the neural network was implemented. In the first stage, pre-training was conducted on an extensive sample of images of reference areas (classes) that were not included in the test area. This allowed the network to identify and learn general features relevant to the recognition of objects in the studied environment. In the second stage, the network was fine-tuned on a limited set of images from the test

areas, which contributed to a more effective adaptation to the target classes and improved the accuracy of their recognition.

To solve the task of training the neural network to recognize objects, the YOLOv5s architecture was chosen. This choice is due to a number of advantages of YOLOv5s. First, it demonstrates an excellent balance between accuracy and image processing speed, which is critical for tasks that require timely analysis. Second, YOLOv5s, being a one-stage detector, is highly efficient due to the simultaneous prediction of object location and classification, unlike two-stage methods. However, it is worth noting that YOLOv5s, being a relatively small model, may be less sensitive to the detection of very small objects or objects with a high degree of occlusion compared to larger models of the YOLO family or other architectures.

In the first stage of training the model, a representative training sample was formed, covering 16 reference areas. For each area, a database was created containing several hundred images. The ratio of the number of images by class in the training sample is shown in Figure 6. To ensure an objective assessment of the training quality, 10 percent of the total volume of collected data was allocated to form the test and validation samples, which allowed for an independent evaluation of the model’s ability to generalize.

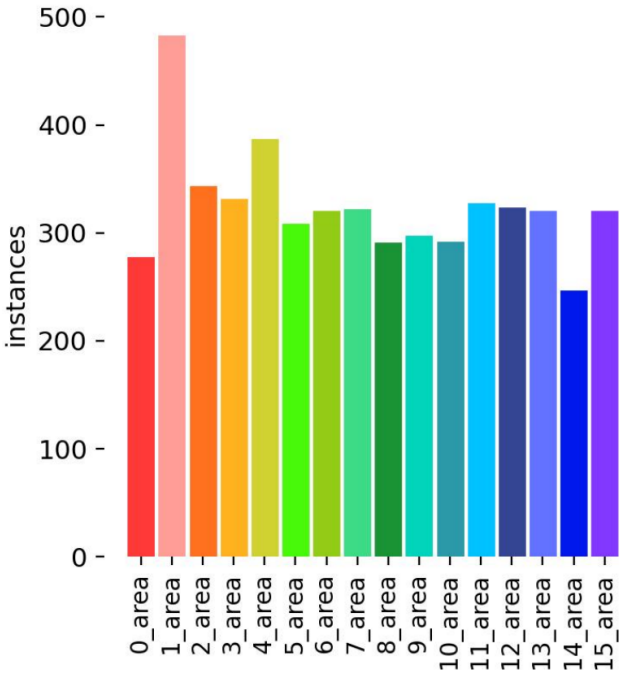

Figure 6. Ratio of the number of images in the training set by class

The training results demonstrate high performance: Precision reached 0.87559, Recall reached 0.90106, and mAP@50 (mean Average Precision at an IoU threshold of 0.5) reached 0.78. Currently, the trained neural network demonstrates confident recognition of target objects within the training sample.

In the second stage, the model was fine-tuned to recognize four new classes, for testing the algorithm on a limited database. The initial training dataset for each of these classes consisted of a limited number of images – only 20 instances. These images were obtained during a single UAV flight over the corresponding reference area, along a straight trajectory. This data collection approach was motivated by simulating the condition of limited availability of training materials, characteristic of real-world application scenarios of the algorithm.

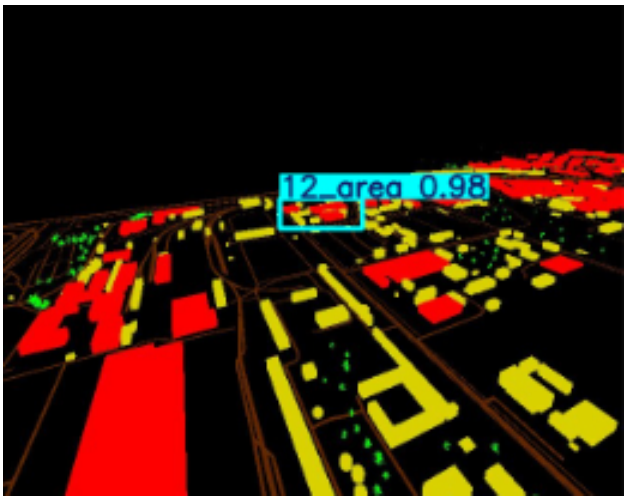

Figure 7. Example of reference area recognition

Since the detector training is based on semantically segmented images, when setting the training parameters, emphasis was placed on certain types of data augmentation, while others were excluded. In particular, during fine-tuning, augmentation methods related to variations in image color characteristics, such as hue, brightness, and saturation, were not used. This decision was made based on the assumption that semantic segmentation, which precedes the detection stage, largely eliminates the influence of changes in the color palette of the images.

Instead, the main focus was on geometric transformations that simulate changes in the observation angle and scale of the images. During fine-tuning, augmentation methods such as scaling (changing the size of objects in the image), rotation (changing the orientation of objects), shifts (moving objects within the image), and perspective transformations (simulating the perspective effect arising from changes in the viewing angle) were widely used. The application of these augmentation methods significantly expanded the effective size of the training sample and increased the model’s robustness to changes in object geometry, which, in turn, contributed to improving the accuracy and reliability of the detection algorithm in various shooting conditions.

4. Results

As a result of the work performed, an algorithm was developed designed to correct the course of an unmanned aerial vehicle (UAV) based on the analysis of images received from the onboard camera. A key feature of the developed algorithm is its ability to detect and identify reference areas of the terrain, while using a minimal amount of prior information about these areas. This is achieved through the effective use of machine learning methods, which allow training the model using a limited number of training images for new, previously unknown, reference areas.

To ensure effective training and validation of the developed algorithm, extensive databases were created. The main database, designed to train the model to recognize general features of the

terrain, included 5462 images for the training sample. In addition, to ensure a reliable assessment of the model’s generalization ability, validation and test samples were formed, each containing 500 images. All images in this database related to 16 different areas used for training.

A separate database was created to fine-tune the model to recognize specific target reference areas used for testing the UAV course correction algorithm. This database contained 80 training images, as well as 100 images each for the validation and test samples. All images in this database related to 4 test reference areas. The limited volume of this database was due to the desire to simulate real-world application conditions, where the number of available images of the target reference areas may be significantly limited.

To optimize the algorithm and assess its robustness to changes in the shooting angle, a series of experiments were conducted by varying the number of images in the training sample and the UAV’s approach angle to the reference object. In particular, the training results using 15 and 8 images in the training sample for each target area were analyzed; the test results according to the mAP@50 metric are presented in Figure 8.

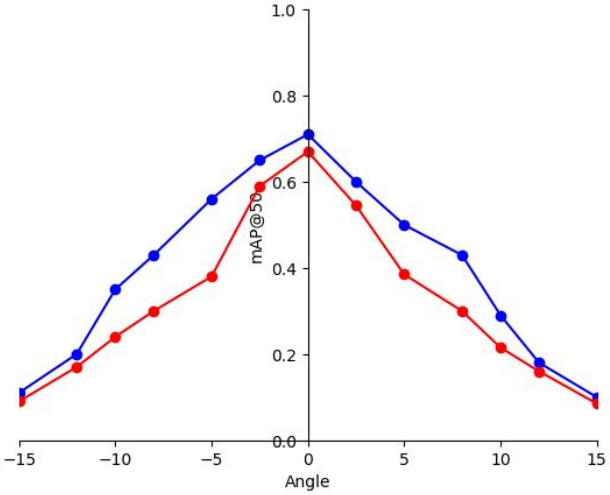

Figure 8. Figure 8 - mAP@50 for various approach angles to the reference object. The red graph shows training with 8 images, and the blue graph shows training with 15 images.

A comparative analysis of the results indicates that reducing the number of images in the training sample has a negative impact on the algorithm’s effectiveness, especially at larger deviation angles from the direct course. The analysis showed that the maximum recognition accuracy and the best mean average precision (mAP50 up to 0.71) are achieved at deviation angles from the direct course in the range of -2.5 to 2.5 degrees. As the deviation angle increases in both directions, there is a consistent decrease in accuracy and mAP50, indicating a decrease in the algorithm’s effectiveness when the shooting angle changes significantly.

The results of the analysis allow us to conclude about the minimum required number of images in the training sample to achieve an acceptable level of recognition accuracy of the target reference areas. According to the data obtained, using 12 to 15 images in the training sample provides sufficient recognition accuracy in the approach angle range from 0 to 8 degrees. In this angle range, the algorithm demonstrates stable and reliable operation, which allows it to be effectively used for UAV course correction.

Despite the decrease in recognition accuracy at large approach angles, the algorithm retains the ability to periodically detect the target area even in conditions of significant deviations from the direct course. This means that even at approach angles exceeding 8 degrees, the algorithm can identify the necessary area with a certain probability, which allows the information obtained to be used for preliminary course correction of the UAV and a return to a more optimal flight path.

Thus, the developed algorithm provides the possibility of UAV course correction even in conditions of a limited amount of training data and significant changes in the shooting angle. However, to ensure stable and reliable operation of the algorithm, it is recommended to use a training sample containing at least 12-15 images and to take into account the influence of the approach angle on recognition accuracy.

5. Conclusion

This work demonstrates the development and validation of a course correction algorithm for unmanned aerial vehicles (UAVs) based on the analysis of images from an onboard camera, designed to operate in conditions with limited prior information about landmarks. A key achievement is the algorithm’s ability to effectively fine-tune using a small number of images acquired from a fixed altitude and a single approach angle.

The experiments conducted allowed us to evaluate the algorithm’s robustness to changes in approach angle and flight altitude differing from those used during training. Analysis of the results revealed that, while recognition accuracy is highest within the range of approach angles close to those used during training (0-8 degrees), the algorithm retains the ability to periodically detect target areas even with significant deviations in perspective. This makes it possible, using a limited training sample, to enable UAV course correction when shooting from altitudes and angles that differ significantly from those on which the training data was collected.

Therefore, the developed algorithm demonstrates potential for application in real-world conditions where it is often impossible to obtain complete and accurate information about landmarks from various angles and altitudes. Recommendations formulated based on the results of the study regarding the minimum necessary size of the training sample (12-15 images) and consideration of the influence of approach angle on recognition accuracy, can be used to further optimize the algorithm and develop adaptive UAV control systems capable of functioning effectively in a wide range of shooting conditions.

References

Alyrchikov, I., Moiseev, N., Knyaz, V., 2024. An algorithm for operational navigation in urban development using reinforcement learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2/W5-2024, 1-7.

Deuser, F., Habel, K., Werner, M., Oswald, N., 2023. Orientation-guided contrastive learning for uav-view geolocalisation.

Horn, B., Schunck, B., 1981. Determining Optical Flow. *Artificial Intelligence*, 17, 185-203.

Kim, Y., 2021. Aerial Map-Based Navigation Using Semantic Segmentation and Pattern Matching.

Kumar, S., Kumar, A., Lee, D.-G., 2022. Semantic Segmentation of UAV Images Based on Transformer Framework with Context Information. *Mathematics*, 10, 4735.

Lucas, B., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision (ijcai). 81.

Miller, B., Stepanyan, K., Popov, A., Miller, A., 2017. UAV navigation based on videosequences captured by the onboard video camera. *Automation and Remote Control*, 78, 2211-2221.

Qian, B., Said, N., Dong, B., 2023. New technologies for UAV navigation with real-time pattern recognition. *Ain Shams Engineering Journal*, 15, 102480.

Sundar, K., Misra, S., Rathinam, S., Sharma, R., 2017. Routing Unmanned Vehicles in GPS-Denied Environments.

Zalesky, B., Shuvalov, V., 2017. Autonomous navigation of drone by onboard video camera: Algorithm and computer modeling. *Scientific Visualization*, 9, 13-25.