An algorithm for operational navigation in urban development using reinforcement learning

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: Semantic segmentation, Reinforcement learning, UAVs navigation, Dijkstra algorithm

Abstract

This paper presents a navigation system for unmanned aerial vehicles (UAVs) operating in urban environments, where the main challenges include changes to the city layout that may not be reflected on existing maps. To effectively address this issue, we have employed semantic segmentation algorithms based on deep convolutional neural networks, which enable accurate identification and classification of urban objects from visual data. This segmentation plays a crucial role in real-time environmental perception, allowing UAVs to distinguish objects such as buildings and vehicles. Initial routes are calculated using an enhanced Dijkstra's algorithm, which determines the shortest path through the urban landscape. However, these routes may require adjustments due to the absence of certain objects on the maps. Along with semantic segmentation and the enhanced Dijkstra's algorithm, reinforcement learning methods are utilized to adjust the navigation routes generated by the algorithms. The reinforcement learning model continuously learns from the UAV's interactions with the environment, optimizing the route by considering safety and efficiency factors. The training and debugging of the algorithm were conducted in a developed synthetic 3D scene. Through simulation and testing in the constructed scene, the proposed navigation system demonstrated improvements in route safety and adaptability compared to routes generated by the enhanced Dijkstra's algorithm.

1. Introduction

In today's world, the speed of information or goods delivery significantly enhances business efficiency and the quality of life for ordinary residents. The use of ground infrastructure for delivery has reached its natural limits, necessitating the development of new modern approaches that do not rely on the overcrowded infrastructure of cities. One of the most promising directions is the development of aerial goods transportation.

In recent years, unmanned aerial vehicles have found increasingly wide applications across various fields, from logistics and delivery to security. However, despite their advantages, navigating UAVs in urban environments presents a complex challenge due to high building density, dynamic obstacles, power lines, and a variety of architectural structures. These factors limit the feasibility of traditional navigation methods based on static maps and require the development of more sophisticated solutions. One potential solution to this problem is the application of deep learning methods, particularly semantic segmentation algorithms based on convolutional neural networks. These technologies enable high accuracy in object recognition, which is critically important for UAVs operating in densely populated areas. Semantic segmentation helps to partition visual data into various object classes, including buildings, roads, and vehicles, thus enhancing overall situational awareness.

However, semantic segmentation is just one part of a comprehensive solution. An important aspect of navigation is also the ability of UAVs to adapt their routes in real-time. To address this challenge, the implementation of reinforcement learning methods is proposed, allowing UAVs to optimize their routes based on experience and interaction with the environment. These methods enable the system to self-learn, making it more resilient to changes in flight conditions.

Thus, the goal of this research is to develop an integrated navigation system for UAVs that combines semantic segmentation, reinforcement learning, and public access maps. The scheme of the developed framework is shown in Figure 1 and described in detail in the next sections.

The proposed approach will not only enhance flight safety and efficiency but also expand the applicability of UAVs in urban environments, opening new horizons for various sectors such as emergency services, delivery, and urban infrastructure monitoring.

2. Related work

In contemporary research, various approaches exist for addressing UAV navigation tasks, with a particular emphasis on navigation within urban environments. Presently, a wealth of continuously updated electronic cartographic data is available, which enables more precise route planning.

The approach proposed by (Castelli et al., 2016) involves loading GIS data of the flight area to subsequently ascertain the shortest path, facilitating the formation of pre-flight tasks aligned with cartographic data.

A notable example of automatic terrain navigation is a solution that applies neural network analysis of camera information for UAV control (Amer et al., 2019).

The system described by (Padhy et al., 2018) is designed for the autonomous navigation of quadcopters in corridor conditions, aiming to replicate the decision-making capabilities of a human pilot in real-time contexts.

(Xiao, 2023) analyzes tree height imagery to determine distances to the trees and measures the width of the space between them to identify the widest passage.

The publication (Soares and Soares, 2016) presents the implementation of an evolutionary algorithm for the control of a robot

Publisher's note: Copernicus Publications has not received any payments from Russian or Belarusian institutions for this paper.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W5-2024-1-2024 | © Author(s) 2024. CC BY 4.0 License.

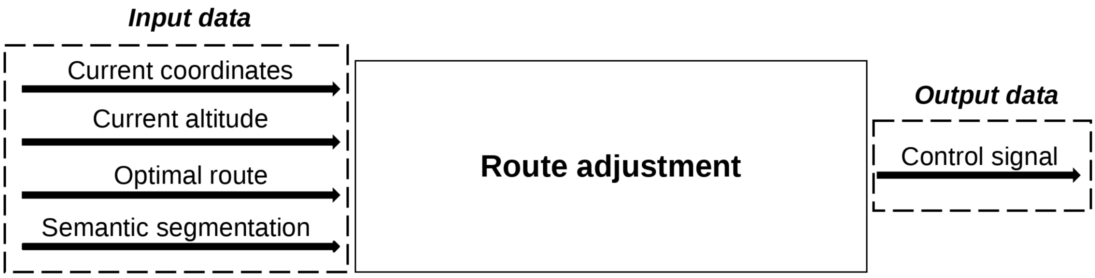


Figure 1. The framework for trajectory correction operation

with autonomous navigation that avoids obstacles. A simulator was developed for testing the algorithm and its configurations within an environment containing a set of barriers detected by sensors.

The use of partially observable Markov decision processes (POMDP) allows for reducing errors associated with GPS signal loss. (Vanegas Alvarez and Gonzalez, 2016) achieved such results concerning the use of unmanned aerial vehicles capable of safely navigating obstruction-free environments through ground control stations that plan routes via a set of GPS coordinates.

(Ali and Sadekov, 2023) explores methods for UAV navigation in GNSS-denied environments using computer vision. The algorithms involve image matching using various approaches, such as pixel-by-pixel matching and neural networks.

Another object analysis approach is described by (Lalak and Wierzbicki, 2022), focusing on object detection and the determination of their geometric characteristics through dense point clouds derived from low-altitude images.

The article (Tanchenko et al., 2020) discusses an original autonomous correction algorithm for the UAV navigation system, based on the comparison of images of the terrain obtained using an onboard machine vision system and vector topographic maps.

The study (Wicaksono and Shin, 2023) proposes a global positioning system (GPS) that supports visual simultaneous localization and mapping (SLAM), named GO-SLAM, for adaptive navigation of unmanned aerial vehicles (UAVs). The GO-SLAM system utilizes a GPS sensor and a camera to implement visual SLAM, providing local positioning information for UAV navigation.

The study (Knyaz and Kniaz, 2020) proposes the approach for UAV navigating in complex environment based on single camera observations. The proposed technique exploits the deep learning approach for image segmentation and depth map estimation using an image of the observed scene. The developed convolutional neural network model is capable to predict depth map of the observed scene along with scene segmentation according the predefined object classes.

(Back et al., 2020) implemented a convolutional neural network (CNN)-based system to govern UAV movement along a specified path while maintaining its position near the center of that path.

The proposal by (Amer et al., 2017) introduces a novel concept termed ”Deep Urban Signatures,” employing deep convolutional neural networks to compute unique characteristics of urban squares or districts based on their architectural and landscape visual perceptions.

(Moteir et al., 2019) presents an intelligent urban navigator for drones that employs a convolutional neural network (Faster R-CNN).

(Cabrera Ponce and Martinez-Carranza, 2022) discuss the utilization of popular CNNs to tackle the geolocation problem based on a single aerial photograph, comparing top-performing CNN architectures in this domain and introducing a compact architecture to expedite inference without degrading estimation accuracy.

In the work of (Konovalenko et al., 2015), a modified method for bias-free pseudo-range estimation is proposed for evaluating UAV positioning. Based on this estimate, a control algorithm was developed to ensure tracking of the reference trajectory in the presence of external perturbations and angular measurement errors.

3. Algorithm

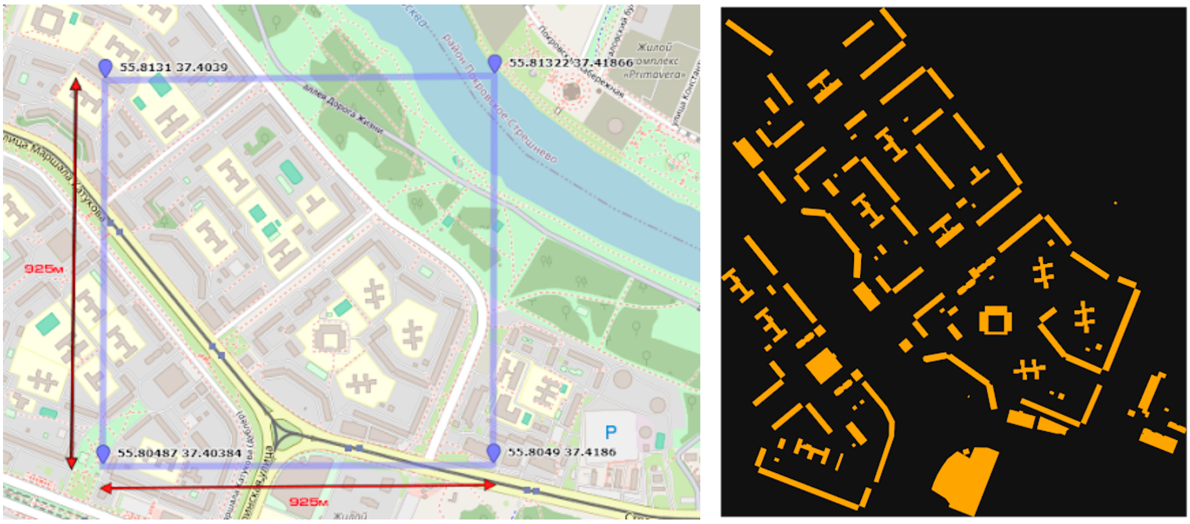
The task of the algorithm is to navigate the unmanned aerial vehicle from the starting position to the destination point. The input data for this task includes OpenStreetMap maps (Figure 2), as well as the coordinates of the start and destination locations. Based on this input, an optimal route trajectory is generated.

The resulting route may contain errors, as because OSM maps may not contain information about all objects or may contain outdated data. To address this issue, a camera is installed on the UAV, which analyzes the environment through semantic segmentation. The output data from this process, along with the optimized route trajectory obtained from the OSM maps and the current coordinates and altitude of the UAV are fed into a secondary neural network that performs route correction.

3.1 Algorithm for optimal path search

A crucial characteristic of the information collection process is the availability of sources that must contain the necessary and sufficient amount of information for subsequent processing. Given the nature of the task, which requires constructing the shortest route in an urban area, the availability of such data under limited access conditions may be confined to open sources. This work utilizes ”OpenStreetMap” (OSM), with the condition that the UAV cannot fly over the rooftops of buildings.

OSM maps provide information regarding the location of urban infrastructure elements, object heights, number of floors, as well as the position of roads and green spaces.



(a) The polygon describing the area of interest

(b) Data processing stage of OSM (extraction of navigation-relevant objects)

Figure 2. (a) area of interest (Strogino, Moscow, Russia) in OSM interface and (b) OSM data processing.

For the debugging phase, the selected area is Strogino, Moscow, Russia. An area of interest, as shown in Figure 2(a), is delineated that reflects the necessary diversity for training the neural network and testing the algorithm.

The next step involves preparing data for subsequent processing and searching for the shortest flight route for the UAV from point A to point B. To this end, the data for the area is uploaded into Python, where the OSMnx library is utilized to extract objects that potentially influence navigation within urban settings, as it is shown in Figure 2(b).

Subsequently, all coordinates of the object vertices are recorded into a separate array (for simplification, it is assumed that all corner points have angles no less than 90 and no more than 175 degrees), marked with a special ID denoting belonging to the same object and information regarding the connection sequence between points.

The acquired data about navigation-significant objects is transferred to a specifically prepared array $a[n][n]$, where $a[i][j].lat$ represents latitude, $a[i][j].long$ represents longitude, and $a[i][j].flag$ denotes a value (0 for absence of an intersection object, 1 for presence of an intersection object). In this array, the points $a[0][0], a[0][n], a[n][0], anda[n][n]$ represent the coordinates of the extreme (corner) points that describe the polygon of the area of interest (Figure 2(a)) and are filled with the corresponding coordinates at a pre-defined interval (varied based on the dimensions of the matrix $n \times n$).

The next step involves filling the resulting object contours with a subsequent "inflation" of the object size by a specific (constant for all objects) increment. This is executed to create a safety zone and avoid situations where the UAV flies too close to objects (an optimal safety zone width is 2 m from the object). Cells not corresponding to navigation-influencing objects are filled with 0.

This array of structures, representing a coordinate grid synthetically created in Unreal Engine 5, allows for visualization in three-dimensional space during various debugging stages. The area of interest is subdivided such that a change in coordinates corresponds to the movement of the quadcopter by 0.5 m. Consequently, the step between adjacent cells of the matrix (neighbouring coordinates) corresponds to a step of 0.5 m.

The next step involves applying an enhanced Dijkstra algorithm (A*) (Sipayung et al., 2023, Wang et al., 2024). The core concept of the A* algorithm is the determination of two values for each vertex in the graph: $V(n)$ – the path length from the starting vertex to the current vertex n. $E(n)$ – a heuristic estimate of the length from the current vertex n to the target. At each step, the A* algorithm (Algorithm 1) selects the vertex with the smallest sum of $V(n) + E(n)$ and explores its neighbours. This process continues until the algorithm reaches the final goal, as it is shown in Figure 3. The heuristic component utilizes the dimensions of the matrix, allowing for analysis of distances by correlating it with geographic coordinates.

Algorithm 1: Dijkstra's Algorithm

Input: Graph $G(V, E)$, source vertex $source$
Output: Distance array $dist$, predecessor array $prev$
for *each vertex* $v \in V$ **do**
  $dist[v] \leftarrow \infty$;
  $prev[v] \leftarrow$ null;
$dist[source] \leftarrow 0$;
$Q \leftarrow V$;
while $Q \neq \emptyset$ **do**
  $u \leftarrow$ vertex in Q with the smallest $dist[u]$;
  $Q \leftarrow Q \setminus \{u\}$;
  for *each neighbor* v *of vertex* u **do**
    $alt \leftarrow dist[u] +$ length$(edge(u, v))$;
    if $alt < dist[v]$ **then**
      $dist[v] \leftarrow alt$;
      $prev[v] \leftarrow u$;
return $dist, prev$;

After determining the shortest distance between the two points, the sequence of coordinates defining the path is recorded in a separate one-dimensional array and subsequently converted into a GeoJSON file for future transmission of flight tasks to the quadcopter.

https://doi.org/10.5194/isprs-archives-XLVIII-2-W5-2024-1-2024 | © Author(s) 2024. CC BY 4.0 License.

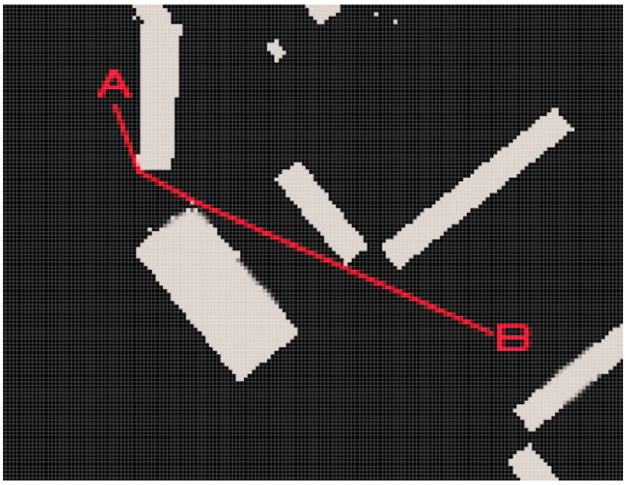


Figure 3. ”Obstacle” matrix with visualization of shortest distance A-B

3.2 Semantic segmentation

Semantic segmentation of objects in an image necessitates a high-quality and extensive database. To collect an appropriate dataset within the Unreal Engine 5 environment, an algorithm has been developed that in real-time captures a stream of frames from a camera mounted on the UAV, along with segmented images, which are depicted in Figure 4, and annotation text files. The camera on the UAV captures images at a resolution of 640×640. The objects present in the scene are categorized into 14 classes: buildings, street and road lighting, soil, concrete/roads, electrical/telephone wires, public transport stops, passenger vehicles, freight vehicles, buses, billboards, navigation signs, tram tracks, and sky.

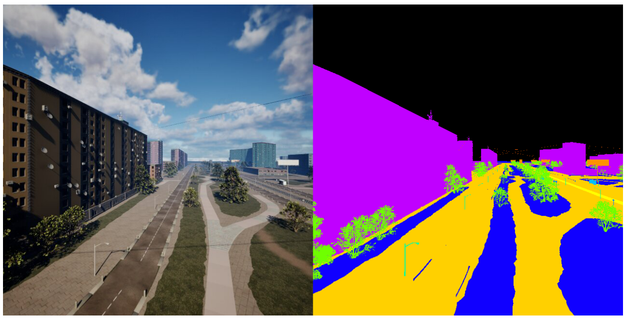


Figure 4. Images for the semantic segmentation training dataset

The selected urban area is divided into two parts: one part is used for collecting the training database, and the other for testing, as shown in Figure 5.

For the task of semantic segmentation, the YOLOv8s neural network architecture has been chosen as a compromise between speed and accuracy.

3.3 Real-time Trajectory Correction

To address the problem of flying around objects with inaccurate locations or complete absence in OSM maps, a reinforcement learning-based algorithm has been developed to adjust UAV movement.

The algorithm (Figure 1) receives as input: the optimal route based on OSM maps, the current coordinates and altitude of the

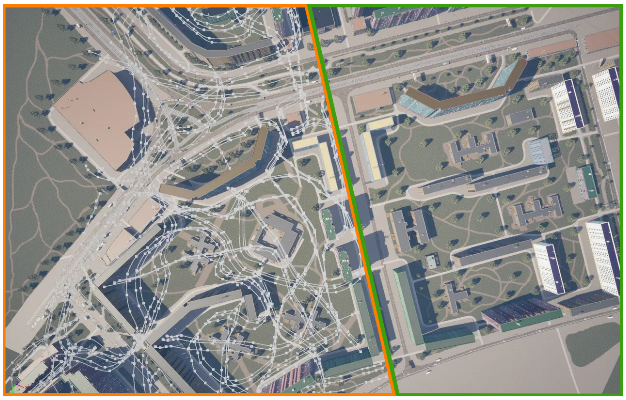


Figure 5. The orange outline indicates the testing area, while the green outline represents the area for collecting the training database

UAV, and the output of semantic segmentation. The output is the direction of UAV movement.

It has been decided to set a target flight altitude of 15m for the UAV. This helps avoid certain obstacles, such as billboards and public transport stops, and eliminates the possibility of collisions with vehicles and pedestrians. The UAV may operate at a different altitude, but the further and longer it remains at that altitude, the greater the penalty it incurs in the reinforcement learning algorithm.

The UAV continuously compares its coordinates with the optimal route based on OSM maps. If it deviates from the route by more than a few meters, it begins to incur penalties. It is assumed that the UAV always accurately knows its current altitude and coordinates.

In Figure 6, the optimal route constructed based on OSM maps is shown as a red line, with the area highlighted in yellow representing a building.

Figure 6. Optimal route according to OSM maps and necessary route correction

The blue area indicates the zone where the UAV does not receive penalties for deviating from the route, and the red dashed

Figure 7. Reinforcement learning training process

line represents the actual shortest route the UAV should take to avoid obstacles. As the UAV deviates further and longer from the OSM-based route, penalties increase, making it necessary to implement an additional incentive zone. This zone is marked in green in Figure 5. When the UAV first enters this area, it can recuperate the penalties accrued while navigating around an obstacle, allowing it to proceed. This mechanism is crucial to ensure that when navigating multiple obstacles, the total penalties do not hinder the learning process.

The result of semantic segmentation from the UAV’s camera is fed into the algorithm, reducing the probability of collisions with objects that are absent in OSM maps. If a collision does occur during the learning process, the UAV receives a significant penalty. In the Unreal Engine 5 scene, multiple UAVs function concurrently (Figure 7), providing real-time data to the reinforcement learning algorithm.

4. Results

Throughout the project, a system was created that enables UAVs to plan routes from point A to point B, utilizing publicly available OSM map data. Additionally, a specialized executable program was developed to process this data for use with the enhanced Dijkstra’s algorithm.

To build the database and refine the system, a synthetic 3D scene was created, which served to compile a dataset of images for training the neural network based on semantic segmentation. The dataset consists of 12,033 training images and 1,500 testing images. It contains 14 classes of interest, the distribution of which is presented in Figure 8.

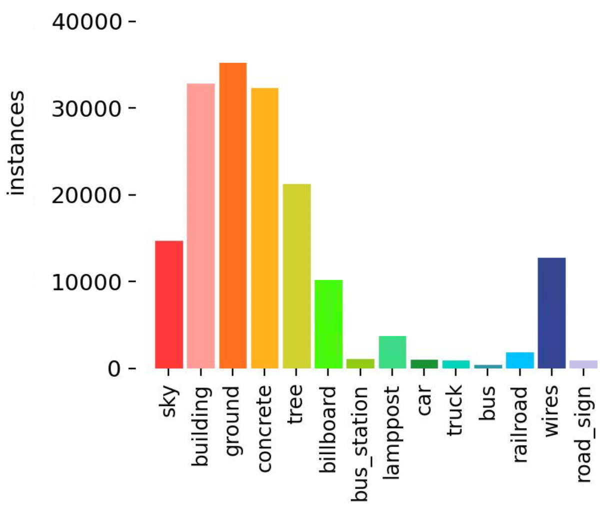


Figure 8. Distribution of labels for each class in the database

A neural network with the YOLOv8s architecture was trained on the collected dataset of urban objects. The training results are not without artifacts, as shown in Figure 9. Most recognition errors occur when defining the boundaries of the ”sky” class and do not impact the performance of the reinforcement learning algorithm for obstacle analysis.

Reinforcement learning was conducted on the developed scene for the UAV control neural network. The neural network analysis of the video stream received from the UAV’s camera enabled route adjustments made by the Dijkstra algorithm, thereby reducing the frequency of collisions with infrastructure and environmental objects absent from the mapping data. Out of 18 trials where motion correction was needed due to objects not represented on the maps, 11 flights were successfully adjusted.

5. Conclusion

A comprehensive navigation system for unmanned aerial vehicles has been developed to enhance navigation efficiency in urban environments. This system utilizes advanced deep learning technologies, such as semantic segmentation, along with an improved Dijkstra algorithm and reinforcement learning methods. Research has demonstrated that this system is capable of real-time adaptation of UAV routes.

Significant attention has been devoted to creating an extensive database for training the neural network and conducting experiments in a synthetic three-dimensional environment, which has shown improvements in the safety and adaptability of routes. Despite the successes achieved, further work is necessary to

Figure 9. Example of semantic segmentation of a frame from the 3D scene

address remaining artifacts in the segmentation algorithms and to expand the system’s capabilities to enhance its reliability in real-world conditions.

Additionally, the general open access cartographic data plays a negative role, as the OpenStreetMap database is publicly available, allowing any user to add or modify existing objects, which introduces issues related to the validity of both entire objects and the accuracy of their positioning. At this stage, the curvature of the Earth has not been considered, which also requires further refinement.

The approach presented in this work may serve as a foundation for further research and development in the automation of UAV navigation, thereby contributing to more efficient and safer utilization of unmanned technologies in urban settings. Test results confirm the system’s effectiveness, opening new opportunities for the application of UAVs in areas such as delivery, urban infrastructure monitoring, and emergency services.

References

Ali, B., Sadekov, R., 2023. A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems. *Gyroscopy and Navigation*, 30, 87–105.

Amer, K., Samy, M., ElHakim, R., Shaker, M., ElHelw, M., 2017. Convolutional neural network-based deep urban signatures with application to drone localization. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2138–2145.

Amer, K., Samy, M., Shaker, M., Elhelw, M., 2019. Deep convolutional neural network-based autonomous drone navigation.

Back, S., Cho, G., Oh, J., Tran, X.-T., Oh, H., 2020. Autonomous UAV Trail Navigation with Obstacle Avoidance Using Deep Neural Networks. *Journal of Intelligent and Robotic Systems*, 100, 1-17.

Cabrera Ponce, A., Martinez-Carranza, J., 2022. Convolutional Neural Networks for Geo-Localisation with a Single Aerial Image. *Journal of Real-Time Image Processing*, 19, 1-11.

Castelli, T., Sharghi, A., Harper, D., Treméau, A., Shah, M., 2016. Autonomous navigation for low-altitude UAVs in urban areas.

Knyaz, V., Kniaz, V., 2020. Object recognition for UAV navigation in complex environment. L. Bruzzone, F. Bovolo, E. Santi (eds), *Image and Signal Processing for Remote Sensing XXVI*, 11533, International Society for Optics and Photonics, SPIE, 115330P.

Konovalenko, I., Miller, A., Miller, B., Nikolaev, D., 2015. UAV Navigation On The Basis Of The Feature Points Detection On Underlying Surface. *Proceedings - 29th European Conference on Modelling and Simulation, ECMS 2015*, 499-505.

Lalak, M., Wierzbicki, D., 2022. Methodology of Detection and Classification of Selected Aviation Obstacles Based on UAV Dense Image Matching. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1-1.

Moteir, I. G. M. I., Ismail, K., Zawawi, F. M., Azhar, M. M. M., 2019. Urban intelligent navigator for drone using convolutional neural network (cnn). *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 1–4.

Padhy, R., Verma, S., Ahmad, S., Choudhury, S., Sa, P., 2018. Deep Neural Network for Autonomous UAV Navigation in Indoor Corridor Environments. *Procedia Computer Science*, 133, 643-650.

Sipayung, L., Sinaga, C., Sagala, A., 2023. Application of Dijkstra’s Algorithm to Determine the Shortest Route from City Center to Medan City Tourist Attractions. *Journal of Computer Networks, Architecture and High Performance Computing*, 5, 648-655.

Soares, A., Soares, T., 2016. An Evolutionary Algorithm for Autonomous Robot Navigation. *Procedia Computer Science*, 80, 2261-2265.

Tanchenko, A., Fedulin, A., Bikmaev, R., Sadekov, R., 2020. UAV Navigation System Autonomous Correction Algorithm Based on Road and River Network Recognition. *Giroskopiya i Navigatsiya*, 28, 32-42.

Vanegas Alvarez, F., Gonzalez, L., 2016. Enabling UAV Navigation with Sensor and Environmental Uncertainty in Cluttered and GPS-Denied Environments. *Sensors*, 16, 666.

Wang, Y., Cheng, X., Sun, X., 2024. Optimization of Cold Chain Path for S company of C city based on Dijkstra’s Algorithm. *Frontiers in Science and Engineering*, 4, 241-258.

Wicaksono, M., Shin, S., 2023. Go-slam: Gps-aided visual slam for adaptive uav navigation in outdoor-indoor environments.

Xiao, S., 2023. Convolutional neural network for UAV image processing and navigation in tree plantations based on deep learning. *Nonlinear Engineering*, 12.