Autoencoder Satellite Image Matching for UAV Geolocation in Long-Range High-Altitude Missions

Lucas B. V. Cordova*, Stephanie L. Brião*, Felipe G. Oliveira*†,
Rodrigo S. Guerra* and Paulo L. J. Drews-Jr*
*Centro de Ciências Computacionais – C3. Universidade Federal do Rio Grande – FURG.
Rio Grande, RS, Brazil
Email: {lucasbenedetti.ppgmc, stephanie.loi, rodrigo.guerra, paulodrews}@furg.br
†Instituto de Ciências Exatas e Tecnologia – ICET. Universidade Federal do Amazonas – UFAM.
Itacoatiara, AM, Brazil
Email: felipeoliveira@ufam.edu.br

Abstract*—Vision-based geolocation is a promising way to overcome the vulnerabilities of Global Navigation Satellite System (GNSS) methods, which are subject to signal degradation, intentional interference, and environmental obstacles. This paper presents a novel approach to Unmanned Aerial Vehicle (UAV) geolocation in long-range and high-altitude missions using satellite imagery. Our method is based on the matching of encoded vector representations in embedded space, demonstrating robust performance to changes in vegetation and landscape. The neural network is used to encode satellite images of a reference map into embedding representations. Image matching is performed in this embedded space using cross-correlation. We evaluated the accuracy and processing time of the proposed model by querying images along a 200 km northbound path at high altitude, covering an area larger than twenty thousand square kilometers. We also evaluated the network's generalization capability on an unknown map. Reference and query images are sourced from satellite images captured at different acquisition times to evaluate robustness due to appearance variations. The results demonstrate that the method can achieve up to 96.83% accuracy on a known map, while experiments on an unknown map averaged 90% accuracy. The processing time to match encoded images is 0.05 ms. These findings suggest the feasibility of integrating the method into more complex vision-based geolocation systems.

I. Introduction

The use of UAVs is growing due to their ability to cover large areas quickly. Applications include agricultural irrigation, firefighting, package delivery, search and rescue, monitoring, and military operations [1]. For georeferenced positioning, UAV navigation depends heavily on GNSS, such as the Global Positioning System (GPS). However, GNSS-based systems face signal issues such as blocking, interference, or inaccuracies [2] and [3]. In this sense, exploring alternative geolocation systems that provide more resilience and autonomy is thus relevant.

Geolocating aerial vehicles involves estimating a UAV's position and attitude in six degrees of freedom (6DOF) [4], including three translation degrees (x, y, z) and three orientation degrees (ϕ, θ, ψ). Furthermore, GNSS methods typically provide global translation data, while Inertial Measurement Units (IMUs) offer orientation parameters. The IMUs provide geolocation only briefly in GNSS failures, given its limitations [1]. Computer vision systems are viable alternatives for addressing the localization problem with more affordable cameras and embedded systems with integrated GPUs. Visual Inertial Odometry (VIO) tracks the camera movements using image sequences and IMU data. However, this strategy accumulates integration errors over time [5]. Another technique uses 3D city maps, as shown in [6], where building edges are used to locate the UAV, although it is costly and limited to urban areas. Abundant satellite image databases like Landsat [7], MODIS [8], and Sentinel [9] have made 2D reference map approaches viable for UAV localization systems.

This work proposes an image-based autoencoder approach for aerial vehicle geolocation on large-scale maps, with robustness to appearance changes. The main contributions of this study are:

- Development of an innovative approach for creating georeferenced maps using the Earth Engine platform;
- Evaluation of a convolutional neural network-based image encoding method to create latent space representations of satellite images captured at **6,000 m** altitude;
- Demonstration of the method over a **200 km** path, covering an area of **20,643 km²**, and evaluation of generalization on an unknown map;
- Proposal of a fast method for measuring similarity between embeddings, around **0.05 ms**, based on cross-correlation.

The model was trained on satellite images from 2022 and evaluated with images from 2023 to test its robustness to appearance variations [10]. Evaluated for accuracy and processing time, the model outperformed feature-based methods. To the best of our knowledge, this is the first work that explores matching autoencoder images with appearance variations in satellite images and unknown maps for aerial vehicle localization in long-range high-altitude missions.

The authors thank the FAPERGS, CNPq and FINEP for the financial support.

979-8-3503-7603-6/24/$31.00 © 2024 IEEE

II. Related work

Several works have proposed relevant approaches addressing the image-matching problem for UAV geolocation. Ren *et al.* [11] highlight the challenges of global robot localization using computer vision due to appearance variations from perspective changes, scene content, and lighting between camera images and database images. Mantelli *et al.* [12] emphasizes that these systems need low computational cost and execution time for real-time image matching. Ali *et al.* [1] reviewed methods for aerial vehicle localization using images, concluding that feature extraction algorithms are most used for their ability to extract essential and distortion-resistant information. Descriptors extract image information such as points, lines, edges, corners, pixels, colors, histograms, and geometric entities [13] for subsequent image matching.

Mantelli *et al.* [12] proposed the abBRIEF descriptor, based on the Binary Robust Independent Elementary Features (BRIEF) [14], for matching drone-captured images to a georeferenced global map, combined with an optimized particle filter algorithm. The method differs from the BRIEF description in using color images instead of grayscale images and employing a quantization process instead of a Gaussian filter to reduce noise. This approach resulted in low execution time and high image-matching accuracy in tests. The most extended experiment covered 2400 m on a 1.16 km^2 map.

Another approach is based on Convolutional Neural Networks (CNNs) to identify critical elements for classification. Ren *et al.* [11] presented an object detection system using RPN (Region Proposal Network) to propose candidate regions before detection, significantly reducing total process time and improving the object detection accuracy. Cunha *et al.* [15] proposed a patch-based CNN approach for landmark recognition, dividing images into patches to improve classification accuracy.

Bianchi *et al.* [16] used georeferenced grayscale satellite images to propose a UAV localization approach regarding low altitudes (about 40 m), based on a CNN autoencoder model [17]. This method is faster and less computationally expensive than a Mutual Information (MI) approach [18]. The authors compared their method with [18] and reported positive results in processing time optimization (0.26 ms vs. 109 ms), maintaining the image matching accuracy. The authors' [16] approach requires retraining the autoencoder for new maps and did not explore the system's robustness with respect to seasonal variations. In contrast to [16], our work mitigated both the generalization of the method on unknown maps and seasonal variation between reference and search map images by using color satellite images, which bring more information into the modeling process [19].

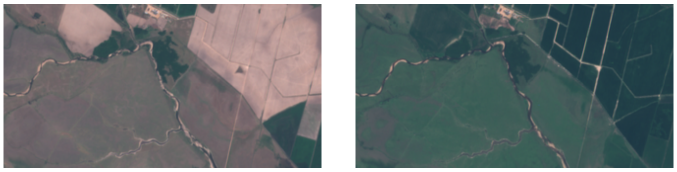
Unlike the mentioned works, [20] addresses the UAV localization problem at medium altitudes. The authors used an Orion-E UAV for a 3,000 m altitude experiment lasting 150 s and covering a 7 km region. They used a vector topographic map as a reference for real-time UAV image comparison. The captured image was segmented using a U-Net [21] to highlight roads, rivers, and background. The Scale Invariant Feature Transform (SIFT) detector [22] then extracted keypoints from the segmented image, and the RANSAC [23] calculated the homography matrix. The proposed method corrected the location of the UAV with an accuracy of approximately 100 m during the loss of GNSS signal.

In [24] addressed the global UAV localization challenges on large-scale maps at low altitudes. The authors introduced a UAV localization approach capable of handling natural variations and ambiguities in drone-captured images over a 100 km² map without prior information about the UAV's initial pose. The CapsNets [25] descriptor is adopted, compacting learned information from satellite mission maps. CapsNets were chosen due to their ability to create stable and robust representations against input perturbations [25]. In the experiments, the UAV achieved precision from 12.6 to 18.7 m on maps with seasonal differences between UAV-obtained images and the reference map.

The presented works highlight a lack of approaches for localizing aerial vehicles over thousands of square kilometers at medium altitudes, as proposed in this work. Satellite images undergo significant appearance changes at medium altitudes due to vegetation, road creation, and construction, as illustrated in Figure 1. These changes are not perceptible in urban zones and altitudes close to sea level, as in the contexts of [12], [16], and [24].

III. Methodology

In the localization problem, image matching can estimate the aircraft pose [24] or act as the primary navigation system [16]. As shown in Figure 1, this can be challenging due to the appearance variations between the database and the images captured by the aircraft [26]. The studies [16], [2], and [24] achieved high success rates by encoding satellite images using discriminative vector representation, *i.e.* embeddings.



(a) Image from January 2022.

(b) Image from January 2023.

Fig. 1. Appearance variation between satellite images at the same location and different acquisition times [10].

An autoencoder [17] is a CNN designed to learn dense vectors that efficiently encode discriminative representations of an image, allowing it to reconstruct the input image at the output [17]. Our autoencoder's architecture consists of three main parts, illustrated in Figure 2. The first part, the encoder, receives the input image and compresses it, preserving discriminative information in a small latent space (bottleneck). This latent representation contains the necessary information to represent the input image. The third part, the decoder, reverses the encoding process, reconstructing the input image from the latent space vector representation. We will use the term embedding to refer to the image representation in the latent space for simplicity.

Our method consists of training an autoencoder with reference satellite images covering the search region. After that, the encoder is used offline to compute the embeddings that represent each of the reference map images (see Figure 3).

To match the encoded image vectors in the embedded space, we used cross-correlation [27]. First, the data are normalized by subtracting the mean and dividing by the standard deviation. Then, given the normalized element values of the two vectors $\bar{x}_i$ and $\bar{y}_i$, the cross-correlation is computed as:

$$r_{\bar{x},\bar{y}} = \sum_{i=0}^{N-1} \bar{x}_i \bar{y}_i, \tag{1}$$

where N is the data number, x_i is the i-th element of the first data series, and y_i is the i-th element of the second data series.

A reference map with images from January 2022 (Figure 1a) was used to train the autoencoder and create the pre-encoded embeddings. We used a different set of images from the same region for querying, captured a year later, in January 2023 (Figure 1b). The method was validated via experiments on different map sizes, measuring accuracy and processing time.

The central idea is to match the encoded representations of images in latent space (see Figure 4). The process begins by randomly selecting [1] an image from the query map. This image is then encoded into an embedding using the trained autoencoder. The cross-correlation is employed to measure the similarity between this query image embedding and each of the pre-computed reference image embeddings from the pre-trained map. The matching is done efficiently through a simple vector-matrix multiplication. The UAV's current location is assigned to the coordinates associated with the reference map embedding with the highest similarity score.

Additionally, we evaluated the method by searching an ordered sequence of images in a continuous path, and we tested the network's generalization capability by assessing its performance on a different map from the one trained.

A. Georeferenced Map

Using the Google Earth Engine [10] Python API, we created a reference map with images from January 2022 and a query map from January 2023, covering an arbitrarily large area of Brazil. For this study, we used Sentinel-2 Level-1C data [9], which offers global coverage with a revisit time of 5 days.

Figure 5 illustrates our procedure for creating the query and reference map images. The whole dataset contains four full-size raw satellite images. These images created a georeferenced grid of clipped 320×160 pixel images. Each grid image was given a unique ID and then indexed to its centroid coordinates, computed using WGS-84 coordinate system [28].

The resulting dataset comprised two maps, one composed of images from 2022 and a second set with images from 2023 of the same region. Each map contains 8064 images and covers an area of $41,287.68$ km² of the same region.

The map with images from 2022 was divided into 85% for the training set and 15% for the test set. Subsequently, each image from the 2022 set is encoded into an embedding to be used as a reference map in the validation stage and experiments. The images from 2023 were used as a query map to simulate a validation environment during the experiments and demonstration of the method as a navigation algorithm.

As illustrated in Figure 4 an image from the query map is self-encoded to later be compared through cross-correlation with each embedding from the reference map, which, at the end of the algorithm, returns the most similar reference image to infer the location estimate. Figure 1 shows reference and query images, highlighting visual differences.

B. Modeling and Training of the Neural Network

We adopted an autoencoder architecture similar to [16]. However, our work explored using the RGB color space, bringing more information into the modeling process [19]. We also explored investigating the network's ability to learn the extraction of representative information, even in ambiguous scenarios.

The code is implemented using the PyTorch framework. Figure 2 presents the neural network architecture and loss function. The input image is compressed into a 1000-element embedding in the encoder stage. The decoder opposes the encoder, attempting to reconstruct the input image.

For the loss function, we calculated the Mean Squared Error (MSE) of the photometric difference between input and reconstructed images. Then, we added the MSE between the corresponding intermediate layers (L1-L5) of the encoder and decoder, with a weighting value α equal to 0.01. In [16], the authors pointed out that these intermediate layer losses help the decoder learn the reverse path of the encoder, improving the network's learning performance. Equation 2 describes the following loss function used:

$$Loss = LF + \alpha(L1 + L2 + L3 + L4 + L5). \tag{2}$$

We used a 16 GB RAM computer for the autoencoder training and a 12 GB NVIDIA GeForce GTX Titan X GPU. The network was trained with satellite images from the reference map for 200 epochs, with a learning rate of 1×10^{-4}.

C. Experimental setup

The entire validation process was implemented in Python using the Google Colab platform. The codes, model, and georeferenced images are available in our repository[2]. The process begins with an offline preparation stage where each reference map image is encoded into an embedding, as presented in Figure 3. The process allows us to build a $1000 \times N$ matrix with the embeddings of the reference images in each column.

The search begins by randomly selecting an image from the query map and computing its corresponding embedding. Subsequently, the resulting 1×1000 embedding vector multiplies

[1] In a real flight, these images would be captured in sequence, and methods based on continuity and aircraft models would be used to discard clear outliers.

[2] https://github.com/benedettilucas/image-matching.git.

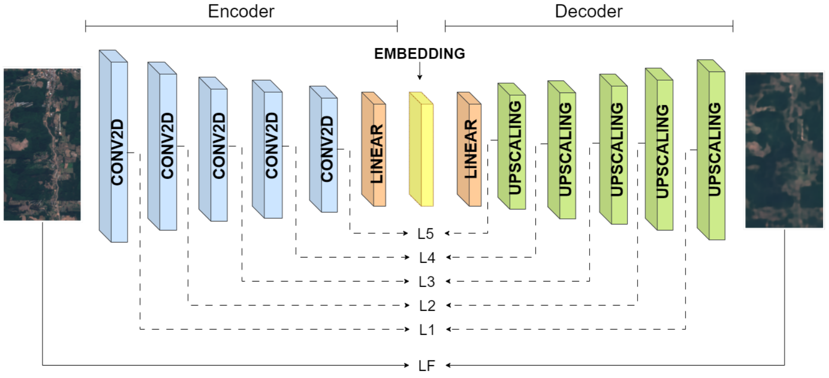


Fig. 2. Autoencoder architecture. LF corresponds to the photometric loss between the input and reconstructed images. The losses between the intermediate layers are L1, L2, L3, L4, and L5. Source: Author.

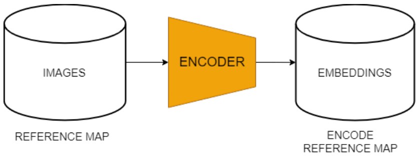


Fig. 3. Reference map encoding.

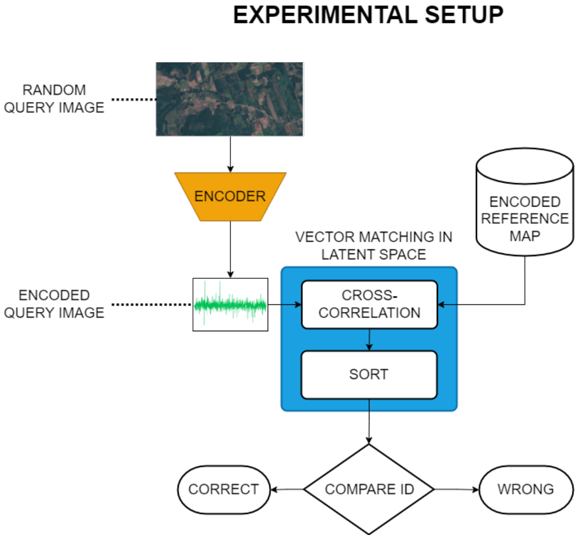


Fig. 4. Flowchart of the experiments for validating the image matching model.

the $1000 \times N$ reference embedding matrix, resulting in a $1 \times N$ vector registering the cross-correlations for each reference embedding. These results are sorted to identify the corresponding ID to the highest similarity, and the correspondence is verified against the ground truth.

The demonstration of the method as a geolocation algorithm was conducted similarly to the flowchart in Figure 4, with the exception that for the demonstration, the images were not randomly selected but were instead inserted in an ordered manner to form a straight path heading north. It is worth noting that the decoder layers are not necessary during the experiments and demonstrations, as their only functionality is for model training.

IV. Results and Discussion

This section presents the training results of the model regarding the loss function and training time. The autoencoder was trained using images from 2022 (reference map). During the experiments, the accuracy of our method is compared with other image matching methods for UAV localization - SIFT [22], ORB [29] and BRIEF [14]. Following this, a demonstration of our method as a geolocation algorithm is conducted, where images from the 2023 (query map) are sequentially selected to form a path from south to north. In this topic, the model's results are presented in terms of accuracy and robustness to appearance variations compared to classical methods. Finally, similar to the first demonstration, the generalization capability of this method was evaluated across eight paths in an unknown map.

A. Autoencoder Training

Figure 6 presents the error graph related to the loss function, MSE, over epochs. The blue line represents the model's error in the training data, and the green line represents the error in the test data. The autoencoder was trained until no significant improvement was observed in the test data. Empirically, it was assumed that the loss curve for the test data stabilized around 200 epochs. The training time for the model was approximately 11 hours.

B. Validation experiments

Experiments were conducted on ten different map sizes - see Table I - to select the map that best fits this approach. The results are presented in Table II for accuracy values across different image matching methods. Ten experiments were carried out for each map size with fifty samples each.

TABLE I
Map areas used in experiments.

Map	Reduction (%)	Map area (km²)	Number of images
1	89.68	4,259.84	832
2	79.76	8,355.84	1632
3	69.84	12,451.84	2432
4	59.92	16,547.84	3232
5	50.00	20,643.84	4032
6	39.68	24,903.68	4864
7	29.76	28,999.68	5664
8	19.84	33,095.68	6464
9	9.92	37,191.68	7264
10	Original	41,287.68	8064

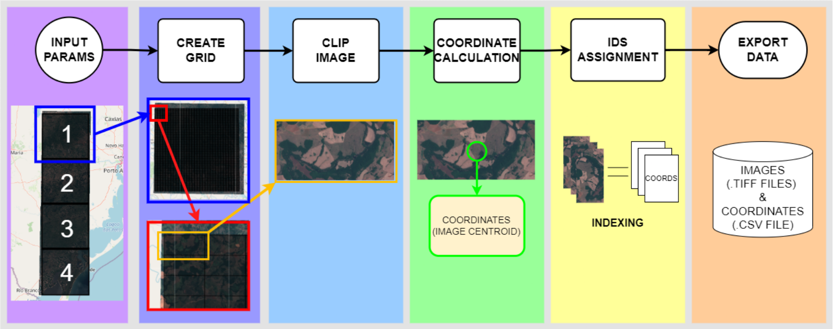


Fig. 5. Stages of the Earth Engine API-Client based algorithm for creating georeferenced maps.

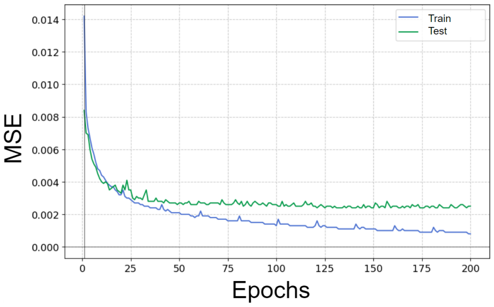


Fig. 6. Loss curve on training and test sets.

TABLE II
AVERAGE ACCURACY OF THE METHODS DURING THE EXPERIMENTS.

Map	Cross-correlation (%)	SIFT (%)	ORB (%)	BRIEF (%)
1	97.40	37.20	17.20	1.40
2	98.40	70.00	40.60	4.60
3	95.00	61.60	35.40	4.60
4	91.20	60.60	34.40	2.00
5	91.00	66.00	38.00	3.40
6	94.40	60.80	35.40	2.80
7	91.20	64.00	32.60	3.00
8	94.40	62.80	32.80	4.60
9	91.00	57.60	27.20	2.80
10	82.40	56.80	27.80	3.80

As this work aims to explore large-scale altitude and area maps, the choice was made to select the largest map that exhibited the highest average accuracy among the applied methods. The optimal choice for the demonstration was map 5.

C. Demonstration of the method as a geolocation algorithm

The demonstration of the method as a geolocation algorithm was conducted in the area covered by map 5, which encompasses regions 1 and 2 of the raw satellite images (Figure 5) with an area of approximately 20,643.84 km². The proposed path is illustrated in Figure 7 and performs a search for correspondences of 126 images from the query map, moving from south to north along a 200 km straight line (blue line), as all images are oriented to the north. The green point marks the starting point, while the yellow indicates the destination.

In this demonstration, the cross-correlation approach correctly matched 96.83% of the images, surpassing the SIFT, ORB, and BRIEF methods. Table III presents the results regarding accuracy, processing time for feature extraction and matching, for one comparison for each method during the demonstration.

It is worth noting that the extraction step occurs once per captured image, while the number of matching operations is proportional to the size of the reference map.

As previously mentioned, the image captured by the aircraft often differs in texture, brightness, and appearance from the image used as a reference for training the model. Therefore, developing an approach that is robust to these variations is crucial. Table IV below shows some query and reference images that exhibited appearance variations, and our method successfully matched them compared to other methods.

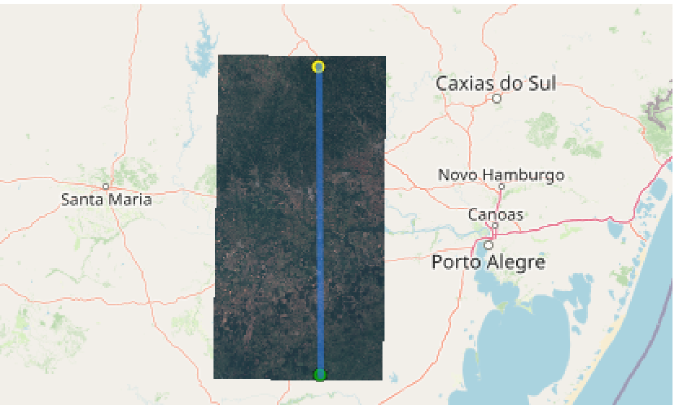


Fig. 7. 200 km path used to demonstrate the algorithm as a geolocation system.

TABLE III
ACCURACY OF METHODS DURING THE DEMONSTRATION.

Method	Accuracy (%)	Extraction (ms)	Matching (ms)
OURS	**96.83**	275.91	**0.05**
SIFT	79.37	14.96	193.66
ORB	47.62	3.43	337.82
BRIEF	6.35	3.03	35.42

TABLE IV
COMPARISON OF PERFORMANCE BETWEEN AUTOENCODER AND OTHER IMAGE MATCHING METHODS FOR UAV LOCALIZATION.

Query image	Autoencoder	SIFT	ORB	BRIEF

D. Generalization

The method's generalization capability in terms of accuracy was investigated regarding eight paths, each approximately 200 km long in an unknown map. The path demonstration followed the procedure used in Map 5. Table V presents the results of the approach for each path in the unknown map.

TABLE V
ACCURACY OF THE METHOD FOR UNKNOWN PATHS.

Path	1	2	3	4	5	6	7	8
Acc (%)	93.1	90.0	90.8	93.8	93.8	82.3	90,8	89.2

V. Conclusion

This work proposes an autoencoder-based algorithm using cross-correlation as an approach to match satellite images in the problem of aerial vehicle geolocation using images. As a contribution, we explored the method's ability in terms of accuracy and generalization to match medium-altitude satellite images that exhibit significant appearance differences. We demonstrated that the implemented method can learn discriminative representations of medium-altitude satellite images, achieving approximately 90% accuracy across all experiments. Moreover, the method showed potential for integration into a geolocation algorithm by correctly matching 96.83% of images along a known map path and an average of 90% across eight paths on an unknown map. Furthermore, this work contributed to an innovative approach to creating georeferenced maps based on Google Earth Engine. Our mapping algorithm can extract large-scale satellite images over time, periodicity, and data volume, marking an essential step in UAV geolocation using satellite images as reference maps.

Future work to enhance the robustness of the proposed approach involves training neural networks on satellite images with more diverse orientations and overlaps to improve matching capability. Additionally, we plan to adopt an aircraft flight model for location belief representation, e.g., particle filter, to narrow the search window, improving the algorithm's performance in terms of response speed and accuracy. Finally, we plan to train and evaluate our method in other regions of the world and different seasons.

References

[1] B. Ali, R. Sadekov, and V. Tsodokova, "A review of navigation algorithms for unmanned aerial vehicles based on computer vision systems," *Gyroscopy and Navigation*, vol. 13, no. 4, pp. 241–252, 2023.

[2] D.-K. Kim and M. R. Walter, "Satellite image-based localization via learned embeddings," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[3] M. M. Dos Santos, G. G. De Giacomo, P. L. Drews-Jr, and S. S. Botelho, "Matching color aerial images and underwater sonar images using deep learning for underwater localization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6365–6370, 2020.

[4] Z. Mohajerani, "Vision-based uav pose estimation," Master's Thesis, Northeastern University, Boston, Massachusetts, 2024, a thesis presented to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

[5] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2502–2509.

[6] K. Qiu, T. Liu, and S. Shen, "Model-based global localization for aerial robots using edge alignment," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1256–1263, 2017.

[7] USGS, "Earthexplorer," 2023.

[8] NASA, "Moderate resolution imaging spectroradiometer," 2023. [Online]. Available: https://modis.gsfc.nasa.gov/data/

[9] ESA, "Sentinel," 2023. [Online]. Available: https://sentinels.copernicus.eu/web/sentinel/home

[10] Google, "Google earth engine," https://earthengine.google.com/, 2023.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[12] M. Mantelli, D. Pittol, R. Neuland, A. Ribacki, R. Maffei, V. Jorge, E. Prestes, and M. Kolberg, "A novel measurement model based on abbrief for global localization of a uav over satellite images," *Robotics and Autonomous Systems*, vol. 112, pp. 304–319, fevereiro 2019. [Online]. Available: https://doi.org/10.1016/j.robot.2018.12.006

[13] J. Gómez-Reyes, J. Benítez-Rangel, L. Morales-Hernández, E. Resendiz-Ochoa, and K. Camarillo-Gomez, "Image mosaicing applied on uavs survey," *Applied Sciences*, vol. 12, no. 5, p. 2729, 2022.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision*. Springer, 2010, pp. 778–792.

[15] K. B. da Cunha, L. Maggi, V. Teichrieb, J. P. Lima, J. P. Quintino, F. Q. B. da Silva, A. L. M. Santos, and H. Pinho, "Patch PlaNet: Landmark recognition with patch classification using convolutional neural networks," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018, pp. 126–133.

[16] M. Bianchi and T. D. Barfoot, "Uav localization using autoencoded satellite images," 2021.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[18] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," in *2020 IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2020, pp. 6491–6497.

[19] M. S. Kankanhalli, B. M. Mehtre, and R. K. Wu, "Cluster-based color matching for image retrieval," *Pattern Recognition*, vol. 29, no. 4, pp. 701–708, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0031320395000976

[20] A. Tanchenko, A. Fedulin, R. Bikmaev, and et al., "Uav navigation system autonomous correction algorithm based on road and river network recognition," *Gyroscopy Navig.*, vol. 11, pp. 293–299, 2020.

[21] A. Buslaev, S. Seferbekov, V. Iglovikov, and A. Shvets, "Fully convolutional network for automatic road extraction from satellite imagery," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 197–1973.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[23] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[24] J. Kinnari, R. Renzulli, F. Verdoja, and V. Kyrki, "Lsvl: Large-scale season-invariant visual localization for uavs," *Robotics and Autonomous Systems*, vol. 168, p. 104497, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023001367

[25] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 44–51.

[26] Y. Li, C. Fu, Z. Huang, Y. Zhang, and J. Pan, "Intermittent contextual learning for keyfilter-aware uav object tracking using deep convolutional feature," *IEEE Transactions on Multimedia*, vol. 23, pp. 810–822, 2021.

[27] T. Derrick and J. Thomas, "Time-series analysis: The cross-correlation function," in *Innovative Analyses of Human Movement*, N. Stergiou, Ed. Champaign, Illinois: Human Kinetics Publishers, 2004, pp. 189–205.

[28] EPSG:4326, "World geodetic system 1984, used in gps," 2023. [Online]. Available: https://epsg.io/4326

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.