Discrete Hierarchical Continual Learning for Single View Geo-Localisation

Aldrich A. Cabrera-Ponce[1] Martin Martin-Ortiz[1] Jose Martinez-Carranza[2]

[1] Benemerita Universidad Autonoma de Puebla (BUAP), Puebla, Mexico.
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico.
carranza@inaoep.mx

Abstract

GPS-based aerial localisation presents a challenge for Unmanned Aerial Vehicles (UAVs) due to signal loss caused by weather conditions. As a result, vision-based methods have been developed to address this issue using the cameras onboard UAVs. The main challenge is to achieve UAV localisation during a flight mission using aerial images and Convolutional Neural Networks (CNNs). To solve this, we propose an aerial localisation methodology based on the sub-mapping concept using continual learning and a multi-model approach. We evaluate and compare our results with ORB-SLAM2, keyframe searching using colour histogram, and with a single model. Additionally, we show that our approach can find the corresponding sub-map and get the camera localisation from a single aerial image with an average accuracy of 0.77 and a processing speed of 69 fps.

1. Introduction

Aerial localisation is a challenge for UAVs that require GPS coordinates to carry out flight missions in outdoor scenarios. The dependence on GPS devices often impedes the acquisition of pose estimation, leading to the development of several vision-based methods. These methods include feature matching [6], Visual Odometry (VO) [5, 10], Simultaneous Localisation and Mapping (SLAM) [4, 12], and deep learning [2, 3, 11]. For the latter, the PoseNet architecture proposed by [8] is used to regress the camera pose and estimate the UAV position using a single image. However, training a model using a large dataset can be computationally expensive and time-consuming. To tackle this issue, an alternative approach is to use continual learning methods to train a CNN with small dataset samples, thereby avoiding catastrophic forgetting when new information arrives.

The continual learning method known as *latent replay* can prove helpful in training models for flight missions [1], particularly in scenarios with limited resources or unpredictable environments [7, 14, 15]. Using this method, CNN models can adapt and improve without re-training, continually updating the model with new information and prior knowledge. This method involves storing the data patterns in external memory and repeating subsequently with incoming data [9, 13]. Moreover, dividing the area into different sections can be advantageous, allowing a more focused and targeted approach to learning.

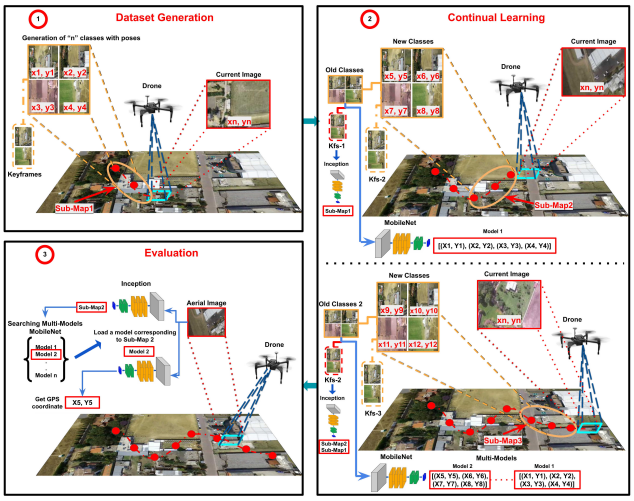

Figure 1. Hierarchical continual learning for aerial localisation consists of three stages: 1) Dataset generation with poses assigned as classes; 2) Continual learning using MobileNet and aerial images with flight coordinates and InceptionV4 using keyframes for sub-map recognition; 3) Multi-model testing using the sub-map searching.

Motivated by the above, we propose a novel methodology for hierarchical learning using two networks that employ a latent replay strategy. Firstly, sub-maps are generated throughout the UAV's trajectory, and representative keyframes are extracted for each sub-map to train the first network. Subsequently, the second network is trained in a

multi-model fashion, with each model representing one sub-map and containing flight coordinates information. Thus, camera poses information is obtained by identifying the relevant sub-map from a testing dataset and loading the corresponding model. Figure 1 presents an overview of our proposed framework.

2. Methodology

We use two deep network architectures for hierarchical continual learning methodology: MobileNet and InceptionV4. Firstly, aerial images were collected from a monocular camera mounted on the UAV associated with GPS coordinates during the flight. Subsequently, we generated a set of keyframes for each segment of the trajectory, which we consider as sub-maps of the entire path. Finally, we continually trained the networks to produce a model containing pose information and another containing sub-map information.

Dataset Generation: The dataset was conducted in two stages: capturing aerial images with GPS coordinates and keyframes representing sub-maps. The latter was facilitated through the Robotic Operating System (ROS), which enabled communication with the drone to get image streams and GPS data to our Ground Control Station (GCS). First, we carried out four flight missions for image acquisition with a resolution of 128×128, and GPS coordinates converted to metres. Afterwards, we generated small samples with data augmentation of flight coordinates divided into classes, resulting in around 50 classes for each trajectory. Additionally, we defined sub-maps using information from five flight coordinates, creating a new sub-map when the drone travels a distance of 50 metres. Finally, we stored keyframes for each sub-map and used them to train the InceptionV4 network. The information in these keyframes determines the trajectory zone to which they belong.

Figure 2 provides a diagram illustrating creating sub-maps and storing keyframes. The diagram depicts sub-maps partitioning, each consisting of five flight coordinates, and the storage of three keyframes per sub-map. These keyframes correspond to sub-map beginning, intermediate and end, corresponding to classes 1, 3, and 5.

Continual Learning: We adopted the continual learning strategy known as a latent replay to train MobileNet as in [9]. This strategy enables us to train the network on the fly while preserving essential patterns in external memory in the *pool6 layer*, with each pattern set representing distinct classes. Combining new data with old patterns rejuvenates the previously learned weights and consolidates the learning process. Furthermore, we restrict the number of classes to 50 to prevent memory saturation and avoid catastrophic forgetting of the first classes. Thus, we used MobileNet to train the aerial images with the five coordinates of each sub-map to create the multi-models while we stored the keyframes.

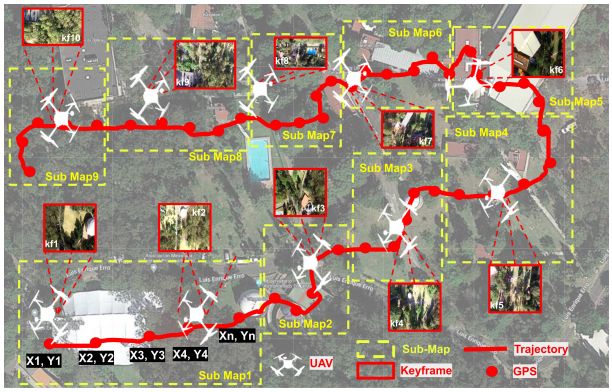

Figure 2. General diagram for the sub-maps creation and keyframes storage. The flight trajectory is shown in red, circles express GPS coordinates, keyframes are in red squares, and sub-maps are in yellow rectangles.

Afterwards, we train InceptionV4 using the stored keyframes and their labels to determine the corresponding sub-map. The training is designed to learn an input image, keeping the features in a vector. Thus, if a new keyframe of the same class arrives, we update the weights in a temporal vector and merge them with the previous ones. Otherwise, the network assigns new features in a new index and expands the features vector with information from both classes. At the end of the training, we concatenate the vector with the temporal vector to join all keyframe features into a single. Figure 3 shows the MobileNet and InceptionV4 training using aerial images and keyframes.

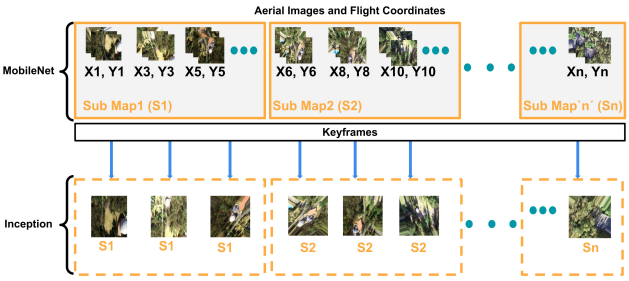

Figure 3. Continual learning with our hierarchical approach. MobileNet is trained using aerial images with five flight coordinates, creating one model for each sub-map. Next, InceptionV4 is trained using the keyframes generated in each sub-map.

This way, we have 47 flight coordinates for trajectory 1, 50 for the second and third trajectories, and 52 for the last trajectory, representing the classes along the entire path. The length of the traversed trajectory is 0.53 km for trajectory 1, 1.4 km for the second, 2.4 for the third, and 2.9 km

for the last. In addition, we stored 27 keyframes for trajectory 1, 30 for trajectories 2 and 3, and 52 for the last one.

3. Experiments and Results

We conducted two experiments to evaluate the effectiveness of our hierarchical approach to aerial localisation. The first experiment aimed to identify the corresponding sub-map of the aerial image using the InceptionV4 and a colour histogram-based method. In the second experiment, we used the hierarchical scheme to determine the corresponding sub-map label and obtain the pose of an input image. For the sake of comparison, we evaluate our results against a single model, ORB-SLAM2 localisation module, and keyframe searching using the colour histogram.

Sub-Maps Results: We trained InceptionV4 on the fly using keyframes generated during sub-maps creation with indexes assigned as labels. Thus, we evaluate the model into 4 test flights following a trajectory similar to the training data. We determined the number of features corresponding to the nearest keyframe and returned the label with the highest similarity of features corresponding to the sub-map. Furthermore, we compared the performance of our sub-map search method with a colour-based approach. For the latter, we calculated the colour histogram of each test image and compared it to the keyframes using the chi2 metric. A low chi2 value indicates higher similarity. Results of our evaluation are presented in Table 1, which shows the number of correct keyframes and accuracy scores obtained using InceptionV4 and the colour histogram method, respectively.

Table 1. Accuracy results with the test dataset using a colour histogram-based approach and InceptionV4 network to find the keyframes and determine the sub-map. The information in bold shows the best result.

Traj.	SubMap	Histogram		InceptionV4	
		Kfs.	Acc.	Kfs.	Acc.
1	9	87	0.6041	**117**	**0.8125**
2	10	65	0.5555	**84**	**0.7179**
3	10	56	0.6666	**60**	**0.7142**
4	10	310	0.5107	**467**	**0.7693**

Hierarchical Learning Results For this experiment, we conducted hierarchical learning as illustrated in Figure 4. First, we input the image into the InceptionV4 network, which produces the corresponding sub-map index. Next, we load the MobileNet model of the identified sub-map and evaluate the image, resulting in one of the five learned classes. We then compared our results with those obtained through a single learning model, the ORB-SLAM2 localisation module, and a keyframe search approach based on colour histograms for aerial localisation.

The single learning model was developed using the continual learning approach with a latent replay strategy. Nevertheless, we learned and updated all classes in a single model instead of having separate models for each sub-map. In the ORB-SLAM2 re-localisation module, we create the map using the training dataset and save the image poses in a text file. Next, we use the test dataset, deactivate mapping and use feature matching to re-localise the test images. A test image with the same descriptors as a keyframe is automatically re-localised in the map. We also compute the distance between the test frame and the closest keyframe to recover the coordinate corresponding to that keyframe. Finally, we used the colour histogram approach with our hierarchical methodology to evaluate the test dataset.

Table 2 displays the comparison results for an aerial localisation task, presenting information on the poses along the entire path, the number of testing images, and the accuracy of each method. Accuracy is the number of correctly re-localised images from acquiring the sub-map to get the corresponding pose. Our methodology produces satisfactory localisation results, successfully re-localising more images on the first two trajectories with an average accuracy of 0.74. In contrast, ORB-SLAM2 performs better on the last two trajectories with an average accuracy of 0.81. On the other hand, a single model gets a maximum accuracy of 0.43, and using a keyframe search system with a colour histogram achieves an average accuracy of 0.47.

The comparison indicates that our methodology performs better than the SLAM system in the first two trajectories, while the latter obtains more poses in the last two. In contrast, a single model approach confirms our assumptions about catastrophic forgetting of the initial data, and even with a continual learning strategy, the network loses knowledge with increasing information. A keyframe search system with a colour histogram and feature extraction may not be advantageous, especially in complex trajectories, and a multi-model-based system has the potential to split the learning of poses into different models. Additionally, the SLAM system's performance drops in complicated scenarios lacking descriptors or with a repetitive pattern, such as Trajectories 1 and 2.

The final result presents the camera poses obtained using a testing trajectory with each method. Thus, if an aerial image is re-localised, we get the sub-map and its pose close to the ground truth. The trajectories' results and the poses recovered using each method are shown in Figure 5. We can see that there are holes in the trajectories, and this is because the process can't obtain the correct location in that zone. In addition, we present the processing speed in fps of each method in Table 3, where ORB-SLAM2 obtains a higher speed but not so far from that obtained with our approach.

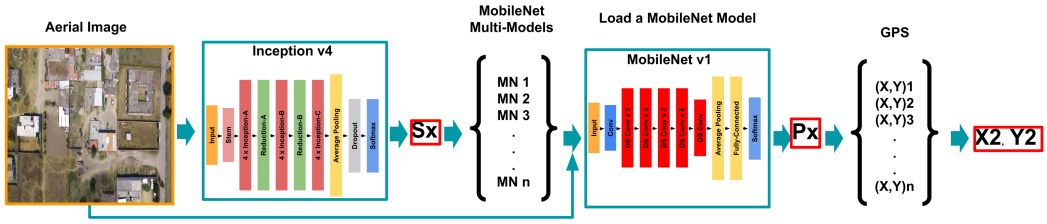

Figure 4. Hierarchical aerial localisation is achieved through a multi-stage evaluation process. Firstly, we use the InceptionV4 network to identify the corresponding sub-map for each image. Then, we load the MobileNet model with information on the 5 flight coordinates specific to that sub-map. Finally, the model evaluates the aerial image to determine the camera pose and obtain the localisation.

Table 2. Accuracy results for aerial re-localisation using a single learning model, the ORB-SLAM2 localisation module, and a keyframe search approach based on colour histograms. The information in bold shows the best result.

Trajectory	Poses	Images	Single Model	ORB-SLAM2	Histrograms	Hierarchical
1	47	144	0.2500	0.1041	0.3055	**0.7083**
2	50	117	0.2564	0.5897	0.3076	**0.7777**
3	50	84	0.2976	**0.9166**	0.6071	0.8928
4	52	607	0.4382	**0.7166**	0.6690	0.7001

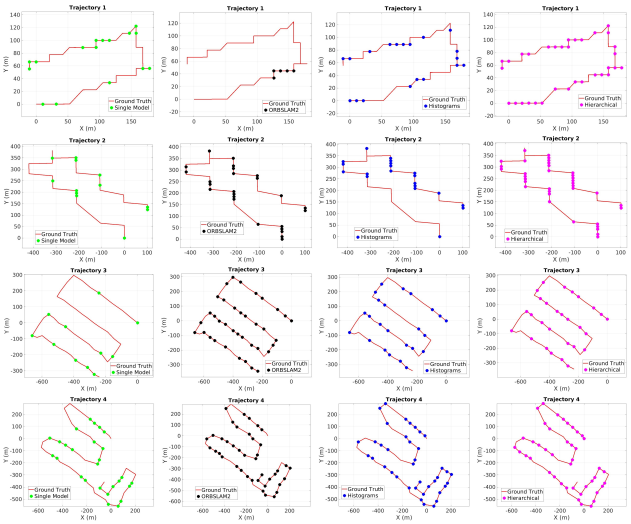

Figure 5. Re-localisation results: Ground-Truth was established using the training trajectory, and circles represent the recovered poses for each evaluated method. The first column shows the evaluation using a single model, the second with ORB-SLAM2, the third using a histogram colour search, and the last with our approach.

Finally, we have counted the images used to retrieve the poses in Table 4, in which our method achieves the highest number of images correctly localised. However, ORB-SLAM2 performs better in re-localising for trajectories 3 and 4 but not for the earlier ones. With the aerial images captured, our methodology could be helpful as a backup localisation method in case the GPS signal is lost.

Table 3. Processing speed in fps with each comparison method. The information in bold shows the best result.

Approach	Traj. 1	Traj. 2	Traj. 3	Traj. 4
Single model	55.30	48.28	55.64	62.76
ORB-SLAM2	**85.47**	**83.33**	**92.57**	**89.28**
Histogram	59.63	62.40	64.80	61.87
Hierarchical	77.44	68.52	67.63	63.37

Table 4. Images correctly located using each method on the 4 flight paths. The information in bold shows the best result.

Approach	Traj. 1	Traj. 2	Traj. 3	Traj. 4
Single model	36	30	25	225
ORB-SLAM2	15	69	**77**	**435**
Histogram	44	36	51	266
Hierarchical	**102**	**91**	75	425

4. Conclusion

We presented a hierarchical continual learning approach to the aerial localisation problem using a single image captured by a UAV. Our methodology utilises two networks to identify a sub-map that best represents the trajectory and a multi-model process for each sub-map divided into the entire path with information on the flight coordinates. We continuously trained the MobileNet and InceptionV4 networks during a flight mission using the latent replay strategy while storing representative keyframes of the sub-map to determine which model to load. As a result, our approach

outperforms the localisation results obtained with single model training and a methodology based on the colour histogram. Additionally, our approach provides localisation results comparable to those obtained with ORB-SLAM. We have demonstrated this approach as a backup localisation method for UAVs with an average accuracy of 0.77 and a performance speed of 69 fps.

References

[1] Aldrich Alfredo Cabrera-Ponce, Manuel Isidro Martin-Ortiz, and Jose Martinez-Carranza. Multi-model continual learning for camera localisation from aerial images. In G. de Croon and C. De Wagter, editors, *13^{th} International Micro Air Vehicle Conference*, pages 103–109, Delft, the Netherlands, Sep 2022. Paper no. IMAV2022-12. 1

[2] Aldrich A Cabrera-Ponce and J Martinez-Carranza. Aerial geo-localisation for mavs using posenet. In *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, pages 192–198. IEEE, 2019. 1

[3] Aldrich A Cabrera-Ponce and Jose Martinez-Carranza. Convolutional neural networks for geo-localisation with a single aerial image. *Journal of Real-Time Image Processing*, 19(3):565–575, 2022. 1

[4] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 1

[5] Ramon Gonzalez, Francisco Rodriguez, Jose Luis Guzman, Cedric Pradalier, and Roland Siegwart. Combined visual odometry and visual compass for off-road mobile robots localization. *Robotica*, 30(6):865–878, 2012. 1

[6] Peter Hansen, Peter Corke, and Wageeh Boles. Wide-angle visual feature matching for outdoor localization. *The International Journal of Robotics Research*, 29(2-3):267–297, 2010. 1

[7] Robin Karlsson, Alexander Carballo, Keisuke Fujii, Kento Ohtani, and Kazuya Takeda. Predictive world models from real-world partial observations. *arXiv preprint arXiv:2301.04783*, 2023. 1

[8] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 1

[9] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR, 2017. 1, 2

[10] Ruben Mascaro, Lucas Teixeira, Timo Hinzmann, Roland Siegwart, and Margarita Chli. Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1421–1428. IEEE, 2018. 1

[11] MS Müller, S Urban, and B Jutzi. Squeezeposenet: Image based pose regression with small convolutional neural networks for real time uas navigation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:49, 2017. 1

[12] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 1

[13] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10203–10209. IEEE, 2020. 1

[14] Ali Safa, Tim Verbelen, Ilja Ocket, André Bourdoux, Hichem Sahli, Francky Catthoor, and Georges Gielen. Learning to slam on the fly in unknown environments: A continual learning approach for drones in visually ambiguous scenes. *arXiv preprint arXiv:2208.12997*, 2022. 1

[15] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. *arXiv preprint arXiv:2203.01578*, 2022. 1