Binary Networks and Continual Learning for Pose Estimation from a Single Aerial Image

Aldrich A. Cabrera-Ponce[1], L. Oyuki Rojas-Perez[2], Manuel Martin-Ortiz[1], and Jose Martinez-Carranza[2]*

[1] Benemérita Universidad Autónoma de Puebla (BUAP), Puebla, México
[2]Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México

ABSTRACT

Pose estimation using aerial images captured by Unmanned Aerial Vehicles (UAVs) allows the localisation in GPS-denied scenarios. Several methods based on deep learning approaches with convolutional neural networks (CNN) have become tools for estimating localisation from images. However, building a model that can estimate the pose from a single image needs a large dataset and training time to obtain a result. Besides, the model can be inappropriate in assessing the correct pose in dynamic scenarios with multiple changes. Therefore, we propose a methodology using a binary network with a Continual Learning (CL) strategy to create an estimation model during the same flight mission. Also, we use a submap scheme and multiple models to acquire the UAV's localisation into different parts of the trajectory. Finally, we use PoseNet, ORB-SLAM2 and single-model for comparison purposes in four scenarios, achieving a percentage error of 14% of the total trajectory and a processing time of 51 ms with our proposed approach.

1 INTRODUCTION

Pose estimation has been implemented in diverse applications within robotics using LiDAR sensors, IMU, and monocular cameras. The latter has gained significant relevance due to the use of images captured by UAVs, providing better information about the environment during a flight mission. With these images, deep learning (DL) methods using convolutional neural networks (CNN) have leveraged visual information to create pose estimation models and thus enable localisation from images. Nowadays, these methods have been employed in UAV tasks such as inspection [1], crop area monitoring [2], visual mapping [3], among others.

PoseNet [4] was the first CNN designed for camera localisation using a single image. Its success inspired the community to develop CNN architectures to improve training time and accuracy, especially in domains such as geo-localisation applications with UAVs [5]. However, these architectures typically demand extensive datasets and training time to establish a reliable model. Consequently, new strategies have been explored to generate estimation models during the same flight mission. Continual learning has emerged as an efficient method for creating these models, requiring less time and data.

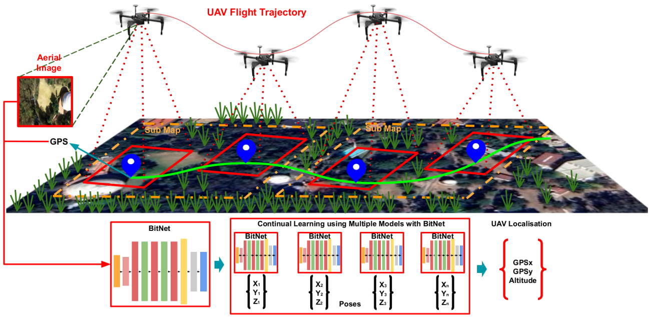

Figure 1: BitNet for aerial localisation: We propose using a binary network to accelerate the training process. In addition, we implement continual learning based on progressive training to extend the knowledge into a multiple-models approach and estimate the pose from a single aerial image.

The continual learning strategy known as latent replay is well-suited for training and updating a model with new information without forgetting prior knowledge. This strategy has proven effective in real-world robotics applications, allowing continuous training while accumulating additional information. However, while continual learning provides efficient pose localisation and time efficiency, accelerating and improving camera pose estimation can be achieved using binary networks. These networks reduce the number of processes, parameters, and calculations in training through weight and activation binarisation. For instance, in [6,7], authors reduced the size of language model weights to one-third of their original size by transitioning from 32-bit to 8-bit representations.

In this work, we propose a methodology that utilises continual learning with a binary network to expedite the training process for obtaining an estimation model. We aim to achieve UAV localisation as a contingency measure in scenarios where GPS signals fail, ensuring uninterrupted flight

*Email address: carranza@inaoep.mx

missions. Based on prior research and the SLAM systems' sub-mapping approach, we support using multiple models along the UAV's flight trajectory. This strategy leverages data acquisition to develop localised models for each flight path segment. Our methodology provides new insights into aerial localisation, by integrating a binary network with continual learning for pose estimation during flight missions. Figure 1 illustrates our aerial localisation methodology using BitNet and continual learning.

This paper is organised as follows. Section 2 shows related works based on continual learning methods; Section 3 outlines the methodology employed, dataset generation and binarised continual training; Section 4 discusses experiments and results obtained. Finally, conclusions and future work are mentioned in Section 5.

2 Related Work

Aerial localisation has been challenging in scenarios with recurrent changes, especially for UAVs conducting flight missions in previously visited environments. Therefore, various methods have been developed to acquire position from an aerial image and use it to locate the UAV when the GPS signal fails. CNN architectures such as PoseNet [4] have enabled significant advancements in camera pose estimation in metres using only one monocular camera, which can be applied to the development of geo-localisation systems [5]. Some compact architectures of PoseNet have been developed to acquire position quickly in outdoor scenarios, such as CompactPN [8], and in indoor environments with DeepPilot4Pose [9].

However, these networks require a lengthy process and an extensive dataset to obtain an estimation model. Therefore, techniques of continual learning have been explored to accelerate the learning process and obtain a model while continuing to collect information. One of the most used strategies is latent replay [10], which involves storing previous information to combine later with new information in external memory. This method avoids catastrophic forgetting and is employed in real-time robotic tasks. In [11], an analysis of continual learning techniques is presented, focusing on human teaching styles. It concludes that teaching experiences do not significantly affect the learning style performed by a robot. Another application of continual learning is presented in [12], where a traversability estimation model adapts a robot to new environments using experience replay with uncertainty, continuously updating the model with new data.

In robotics, systems like Visual Odometry (VO) and SLAM (Simultaneous Localisation and Mapping) have been employed alongside continual learning. Visual Odometry, for instance, is a localisation technique based on camera movement, utilised in continual learning to enable UAV localisation through LiDAR systems in GPS-denied environments [13]. When combined with experience replay, VO facilitates the creation of efficient models using images and dual networks to achieve localisation in diverse scenarios [14]. However, these methods still face the challenge of catastrophic forgetting, as discussed in [15], which examines VO-based continual learning models. This study identifies information loss with minor scene changes, endorsing latent replay strategies that better preserve information without forgetting previous knowledge.

On the other hand, unlike VO, SLAM systems, such as ORB-SLAM2 [16], create a map of the environment, which has been utilised to expand the map of scenarios and achieve UAV localisation in unknown environments [17]. Another approach, Continual SLAM [18], leverages point clouds, images, and loop closure to predict new trajectories using dual networks. Additionally, in [19], researchers utilise dual networks as feedback with linear optimisers to improve the camera pose's visual prediction by identifying BIAS correspondences. However, implementing dual networks and latent replays can incur significant time consumption during the creation and update of estimation models as new information is acquired.

Continual learning finds diverse applications, particularly in localisation systems. In [20], recent research explores continual learning approaches for camera localisation using image inputs. The study emphasises the retention of previous information, the integration of new experiences, and various strategies tailored for localisation, including the use of dictionaries [21]. Furthermore, the research delves into aerial localisation with UAVs, showcasing advancements such as topological localisation based on mean poses [22] and hierarchical search techniques employing sub-maps and multiple models [23]. These studies demonstrate effective UAV localisation using single images, employed for classification-based localisation and recovering poses close to ground truth.

Motivated by previous works, we propose leveraging continual learning with multiple models and submap search combined with BitNet, a binary network initially designed to enhance efficiency and reduce computational costs in Large Language Model (LLM) [6, 7]. This approach opens new possibilities for extending BitNet's capabilities to UAV localisation. By binarising convolutional layers and integrating continual learning, BitNet aims to enrich knowledge and maintain precise position information across diverse scenarios, even in GPS-denied environments.

3 Methodology

This work aims to develop a localisation system for UAVs using continual learning and a binary network. The methodology involves progressively learning incoming information to expand knowledge through multiple models alongside trajectory. In this way, we generate the dataset and sub-maps for the training process using continual learning strategies. Thus, our methodology aims to estimate aerial poses to localise a UAV within dynamic or GPS-denied scenarios.

3.1 Dataset Generation

The dataset used for this work is the same presented in [22], which consists of 4 trajectories with fewer than 300 images for the first three of these. In contrast, the fourth trajectory contains 7,000 images with continuous coordinates different from the first ones, whose coordinates are not constant. For this dataset, we used the UAV Matrice 100 connected with the Robot Operating System (ROS) to establish communication between the ground control station (GCS) and the onboard computer. Thus, our configuration enables real-time acquisition while the UAV performs a mission flight.

The training scenarios consist of transversal lengths of 1.0 km for Trajectory 1, 3.0 km for Trajectory 2, 4.6 km for Trajectory 3 and 6.0 km for Trajectory 4. We captured the aerial images with an HD resolution, GPS coordinates, and 50 to 100 metres altitudes. For training purposes, we resized the images to 224 × 224 and converted the GPS coordinates to metres to handle the flight coordinates better. Besides, we create sub-maps along the trajectory, dividing it into ten sections where each one has 20 to 30 images with coordinates in x, y, and z.

We decided to create sub-maps along the trajectory to train a model for each one, allowing us to manage the localisation in case the UAV loses its position. Thus, we create a sub-map when the UAV moves between 50 to 100 metres from its preceding coordinate, achieving between 5 to 20 flight coordinates for each sub-map. In addition, we generate three keyframes representing the sub-map using indexes, which will be used in the training to identify the corresponding one using the current UAV image. In Figure 2, we show a representation of dataset generation, the creation of sub-maps, the keyframes, and the trajectory of the UAV.

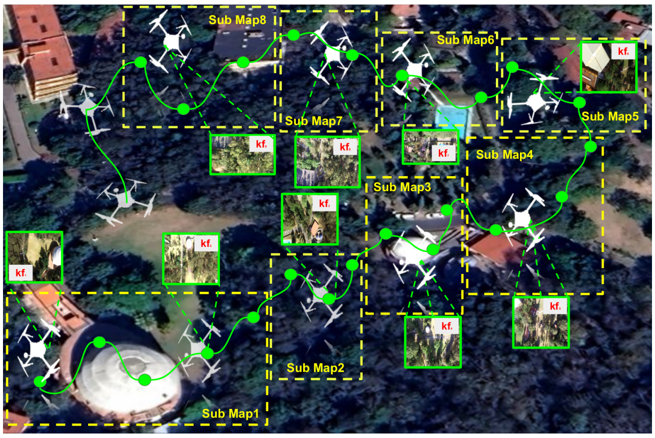

Figure 2: Representation of the UAV trajectory, sub-maps and keyframe generation during a flight mission. We show the UAV trajectory in green, coordinates in green circles, and keyframes in green outlines.

In this way, we generated ten sub-maps for each trajectory, with 30 keyframes for training purposes. In contrast, we generated five keyframes for each sub-map in trajectory four because it is the most extended trajectory, increasing the keyframes for better representation. In this way, we have two datasets, one with aerial images and flight coordinates to train the binary network and the second with keyframes to train a classification network. In Table 1, we show the information about the datasets and the sub-maps generated with the number of images for training.

Table 1: Datasets generated: aerial images include flight coordinates, and keyframes include sub-map indexes.

Traj.	**Aerial Images**		**Keyframes**	
	Train	**Test**	**1 Sub-Map**	**10 Sub-Maps**
1	220	144	3	30
2	188	117	3	30
3	236	84	3	30
4	7826	607	5	50

3.2 Binary Network

BitNet is a neural network architecture designed to improve the efficiency and scalability of deep learning models, particularly in applications like large language models [6, 7]. It uses 1-bit representations instead of higher precision formats such as 16-bit or 32-bit, significantly reducing memory requirements and computational load. BitNet employs advanced quantisation techniques to lower the precision of model weights and activations, optimising performance without compromising accuracy. Additionally, it integrates dequantisation techniques to recover information lost during quantisation, preserving as much detail as possible. In some cases, BitNet also employs linear layers instead of convolutional layers to make models lighter and more efficient regarding computational resources.

In this work, we follow the methodology proposed in [6] to implement a binary network, where the binarisation of weights $W \in \mathbb{R}^{n\times m}$ is formulated as:

$$\tilde{W} = \text{Sign}(W - \alpha) \tag{1}$$

Each weight W_{ij} is transformed into $+1$ or -1 based on whether it is greater or less than α, where α represents the average of the weights.

$$\text{Sign}(W_{ij}) = \begin{cases} +1, & \text{if } W_{ij} > 0, \\ -1, & \text{if } W_{ij} \le 0 \end{cases} \tag{2}$$

$$\alpha = \frac{1}{nm}\sum_{i,j} W_{ij} \tag{3}$$

To ensure that the gradient during backpropagation remains consistent with the original function, we implemented the Straight-Through Estimator (STE) method for $\tilde{W}$. This allows the gradients to be calculated as if no binarisation had occurred, making the training more stable and effective.

The binarisation of activations is described as follows:

$$\tilde{x} = \text{Quant}(x) = \text{Clip}(x \times \frac{Q_b}{\gamma}, -Q_b + \epsilon, Q_b - \epsilon) \quad (4)$$

$$\text{Clip}(x, a, b) = \max(a, \min(b, x)), \quad \gamma = \|x\|_\infty \quad (5)$$

where $Q_b = 2^{b-1}$ for a quantisation of b bits (in our case 8 bits), γ is the maximum absolute value of x, Clip ensures that x is in the range [a, b], and ϵ is a small floating-point number (set to 1×10^{-5}) that prevents overflow during clipping. With the above quantisation equations, the matrix multiplication can be written as:

$$y = \tilde{W}\tilde{x} \quad (6)$$

3.3 *Continual Learning*

We carried out two continual learning processes for model training using a rehearsal and architectural strategy with aerial images, flight coordinates and sub-maps indexes. The first training was based on an architectural strategy called progressive learning, which focuses on expanding the knowledge within a network architecture. The second training is based on a rehearsal strategy called latent replay, which consists of saving the previous information and combining it with the new one using external memory. Therefore, we use both processes to train dual networks simultaneously during the UAV flight mission.

The first training was a progressive learning process, and instead of expanding the knowledge within a network architecture, we decided to extend it to multiple models. Thus, we trained each model with aerial images and flight coordinates as labels, maintaining the information without suffering catastrophic forgetting. The network is a 4-layer architecture with convolutional layers binarised and a fully connected layer with a regression with three outputs for x, y, and z coordinates. This learning allows us to have a suitable model in case GPS is lost, with the last model responsible for localising the UAV.

Furthermore, this learning strategy allows us to generate a model and acquire new data to create the next one while the UAV perform the flight. Each model was trained with less than 30 images in each sub-map, with 5 to 20 coordinates depending on the scenario. In addition, we perform traditional training using all the images in the dataset to train the binary network, generating a single model of the entire trajectory for evaluation purposes. We train the binary network using 100 epochs, Adam optimiser, and a learning rate of 0.001 for multiple and single models.

We used the keyframes generated in the dataset generation step for the second training. Thus, we use a classification network called InceptioV4 to determine the index and identify the corresponding sub-map using the current aerial image. Therefore, we train the network using the rehearsal strategy based on the latent replay method, which consists of using external memory to keep the previous information. In this way, InceptionV4 extracts maximum and minimum features from the keyframes and stores them in the sample vector whose dimension is ten indexes, each representing a sub-map.

Subsequently, we use this vector to compare the stored features with the features extracted from the current aerial image, thus obtaining the index of the corresponding sub-map. In this way, InceptionV4 allows us to identify the place corresponding to the image to load the binary network model and estimate the pose. The progressive learning and latent replay methods were implemented using a computer laptop with CUDA 12.2, PyTorch 2.3, 8GB of RAM and a GeForce 960M Nvidia card. Thus, our methodology estimates the pose from aerial images, identifying the corresponding sub-map and obtaining the localisation of the UAV. Finally, we show a representative diagram of the continual learning process using aerial images and keyframes in Figure 3.

4 Experiments and Results

We conducted experiments to evaluate the pose estimation and obtain the aerial localisation from a single aerial image. These experiments assess a two-stage process. The first stage consists of searching for the submap corresponding to the current view of the UAV. The second stage aims to estimate the aerial pose using multiple models and establish the UAV localisation into a scenario. For comparison, we evaluate our methodology across four trajectories to estimate the pose frame-by-frame, comparing the error and processing time results with a neural network, PoseNet, and ORB-SLAM2.

4.1 *Submap Search Stage*

The main idea for the UAV localisation is to find the correct area and estimate the pose using the current aerial image. For that, we present the sub-map search using the features extracted with InceptionV4 and features from a colour histogram. In this way, we find the corresponding submap using the index obtained by matching features between those stored and those extracted from the current image. Thus, if the image matches any keyframe representative of the sub-map, it will return the corresponding index, locating the UAV in a section of the total trajectory.

To show the sub-maps search results, we present a comparison using histogram colour and InceptionV4. In Table 2, we show the number of images in the test dataset used to find the corresponding sub-map, the keyframes found correctly and the accuracy results with each method. We can see that InceptionV4 outperforms the colour histogram due to the better management of the features, with minimums and maximums that grow as the continuous training of the keyframes progresses. Moreover, we argue that using a classification network as a search system can offer advantages due to high-quality features, resulting in the best performance. The results obtained can be used to search the sub-map and load the multiple models of the binary network.

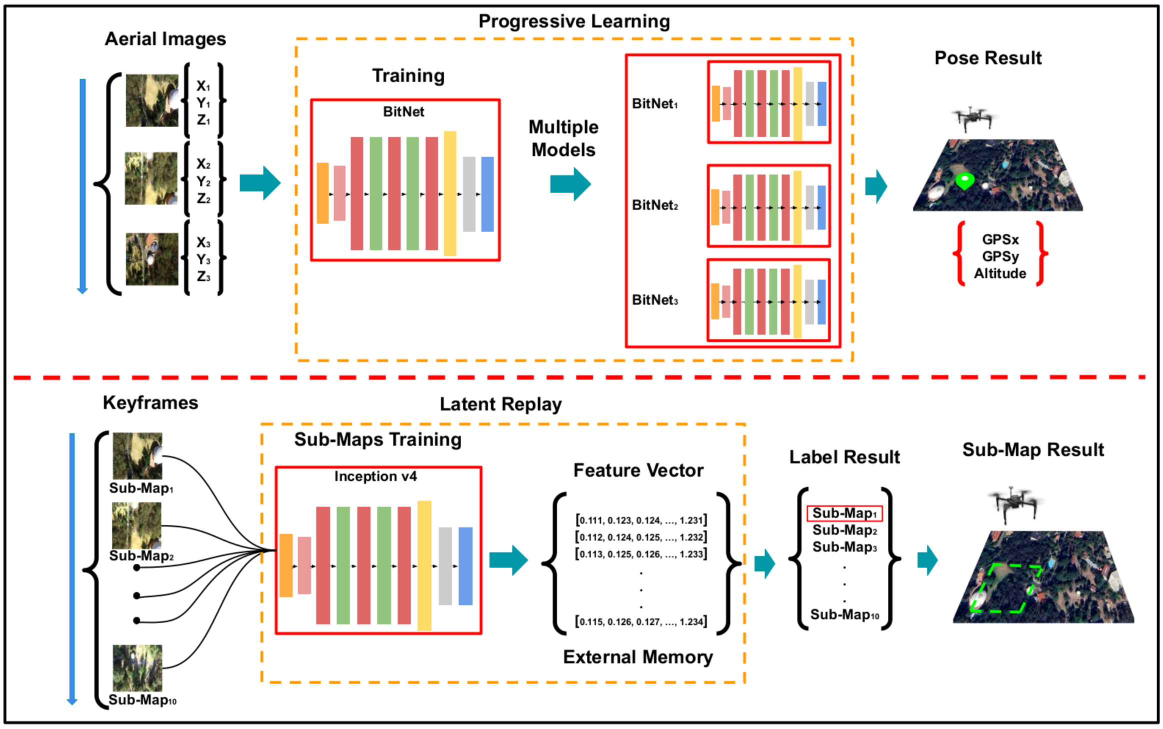

Figure 3: Our training methodology is divided into two parts: Firstly, we train a binary network using aerial images with flight coordinates to create multiple estimation models. Secondly, we train the InceptionV4 network using keyframes to identify the corresponding sub-map.

Table 2: Results of the submap search using colour descriptors and those obtained by the InceptionV4 network.

Traj.	Test	Histogram Colour		InceptionV4	
		Kfs Found	Acc.	Kfs Found	Acc.
1	144	87	0.60	117	0.81
2	117	65	0.55	84	0.71
3	84	56	0.66	60	0.71
4	607	310	0.51	467	0.76

4.2 *Localisation Stage*

This second stage establishes aerial localisation using a single image to estimate the UAV pose. Thus, our methodology works from the moment the UAV images arrive at our localisation system, passing through the InceptionV4 network to find the sub-map index to which it belongs. Once the sub-map is found, the model corresponding to that sub-map is loaded to estimate the pose evaluating the same image. In this way, this process is repeated continuously as each image from the test set arrives. In Figure 4, we illustrate the experimental setup for aerial localisation, showing the sub-maps search, the models' loading and the poses estimation using the input image.

For comparison, we have implemented a 4-layer neural network (NN) to train a single model and multi-model using our methodology. We also used PoseNet and ORB-SLAM2 methods to localise the UAV using a single image. Likewise, we train our binary network using all the images in the dataset to generate a single model and compare it with our multi-models approach. In this way, in Table 3, we present the results obtained with these approaches in the four trajectories, obtaining the Mean Euclidean Distance Error in metres.

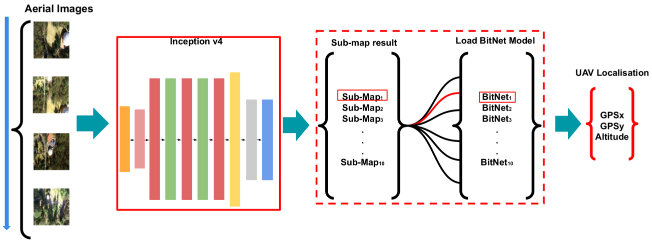

Figure 4: Aerial localisation diagram: The images pass through the InceptionV4 network to identify the corresponding sub-map. Subsequently, we loaded the corresponding estimation model to get the aerial pose.

Similarly, we show the percentage error per axis in Table 4, where a higher percentage consists of a larger error in the pose. These results indicate whether the image obtains the

Table 3: Mean Error Euclidean Distances in metres. The best results are highlighted in bold.

Approach	Trajectory 1			Trajectory 2			Trajectory 3			Trajectory 4		
	x	y	z	x	y	z	x	y	z	x	y	z
PoseNet	47.4	15.3	5.07	94.1	82.3	12.7	59.9	38.3	19.0	146.1	130.8	**2.09**
ORB-SLAM2	-	-	-	**13.04**	**10.68**	6.90	-	-	-	-	-	-
NN-Single	39.4	9.72	**0.10**	82.09	75.7	**5.80**	34.4	37.9	**6.06**	140.7	120.6	7.30
NN-Multiple	12.8	10.9	4.07	45.8	48.47	13.5	27.8	**21.5**	7.05	**52.64**	**52.75**	23.2
BitNet-Single	12.3	11.6	3.40	77.2	61.2	12.5	35.7	29.8	8.58	59.1	64.6	17.7
BitNet-Multiple	**9.63**	**5.38**	2.97	33.8	28.8	7.32	**26.3**	21.9	7.39	57.3	57.3	11.9

Table 4: Percentage Error per Axis (%). The best results are highlighted in bold.

Approach	Trajectory 1			Trajectory 2			Trajectory 3			Trajectory 4		
	x	y	z	x	y	z	x	y	z	x	y	z
PoseNet	48.7	30.8	9.91	60.4	43.0	12.7	19.0	33.6	18.9	76.9	53.2	**2.11**
ORB-SLAM2	-	-	-	**16.6**	**5.74**	7.56	-	-	-	-	-	-
NN-Single	40.5	19.5	**5.54**	52.6	39.6	**5.70**	10.9	33.3	**6.03**	74.0	49.0	7.26
NN-Multiple	13.1	22.0	8.07	29.4	25.3	13.4	8.82	**18.8**	7.02	**27.7**	**21.4**	12.1
BitNet-Single	12.7	23.31	6.74	49.58	32.0	12.5	11.3	26.1	8.54	31.1	26.2	17.6
BitNet-Multiple	**9.90**	**10.8**	5.89	21.7	15.0	7.29	**8.34**	19.2	7.35	30.1	23.3	11.9

correct aerial localisation, depending on the sub-map identification. Besides, we noted that the z-axis shows the lowest values because all flights are conducted at a consistent height without changes. In contrast, we see a broader percentage of error in the x and y axes due to the low number of images in the dataset, the fast training, and the binary network, which reduces the precision of the training process.

Likewise, we present the total percentage error using each approach in Table 5, which is the sum of the translation error across the entire length of the real trajectory. Thus, we noticed that the ORB-SLAM2 did not obtain the localisation because the aerial images are not continuous, causing the mapping to break. Nevertheless, it achieves the lowest error in trajectory two but does not complete the others, for which we do not obtain a result. In contrast, PoseNet obtained the highest error percentage in all the trajectories due to the low number of images used during the training process. Finally, our multi-model approach with a binary network obtained a mean percentage error of 14.9, showing the lack of precision when performing binary calculations in the training process but being suitable to get the localisation.

Table 5: Total Percentage Error (%). The best results are highlighted in bold.

Approach	Traj.1	Traj.2	Traj.3	Traj.4	Mean
PoseNet	34.3	42.2	22.1	52.0	37.6
Orbslam2	-	11.7	-	-	-
NN-Single	26.3	46.5	14.8	50.0	34.4
NN-Multi	14.1	24.0	10.6	21.9	17.6
BitNet-S	13.8	33.7	13.9	25.3	21.6
BitNet-M	9.11	15.6	10.5	24.6	**14.9**

Finally, we present in Figure 5 the visual results of the localisation using PoseNet, a multi-model approach with a 4-layer network, and our multi-model approach using the binary network. For better visualisation, we show the real trajectory in red and the estimated trajectories in other colours, in which PoseNet is represented in column 1, multiple-models with the 4-layer network in column 2, and our multiple-model approach in column 3. PoseNet presents several estimation jumps, resulting in erroneous localisations, while the 4-layer network approach achieves good estimation results but with some jumps. Finally, our approach estimates the poses closer to the test trajectory, showing better performance than the others.

4.3 Discussions

In order to evaluate the effectiveness of binary networks, it is necessary to know the estimated time to give the pose for each image that arrives. Thus, given the binarisation in the convolutional layers, the training parameters of the network are reduced, making the estimation models have less weight. Furthermore, taking it to a continuous learning process, a model's training and acquisition time can occur during the same UAV flight mission. This process can benefit real-time localisation tasks using images captured with UAVs.

Table 6 presents the total time with the approaches used to estimate the pose and acquire the localisation. Therefore, we evaluate the efficiency of our approach, measuring the processing time in milliseconds (ms) and comparing the entire process from searching the submap to loading the corresponding model to obtain the estimated pose from a single image. PoseNet achieves the slowest processing time of the other approaches, while ORB-SLAM2 only finishes trajec-

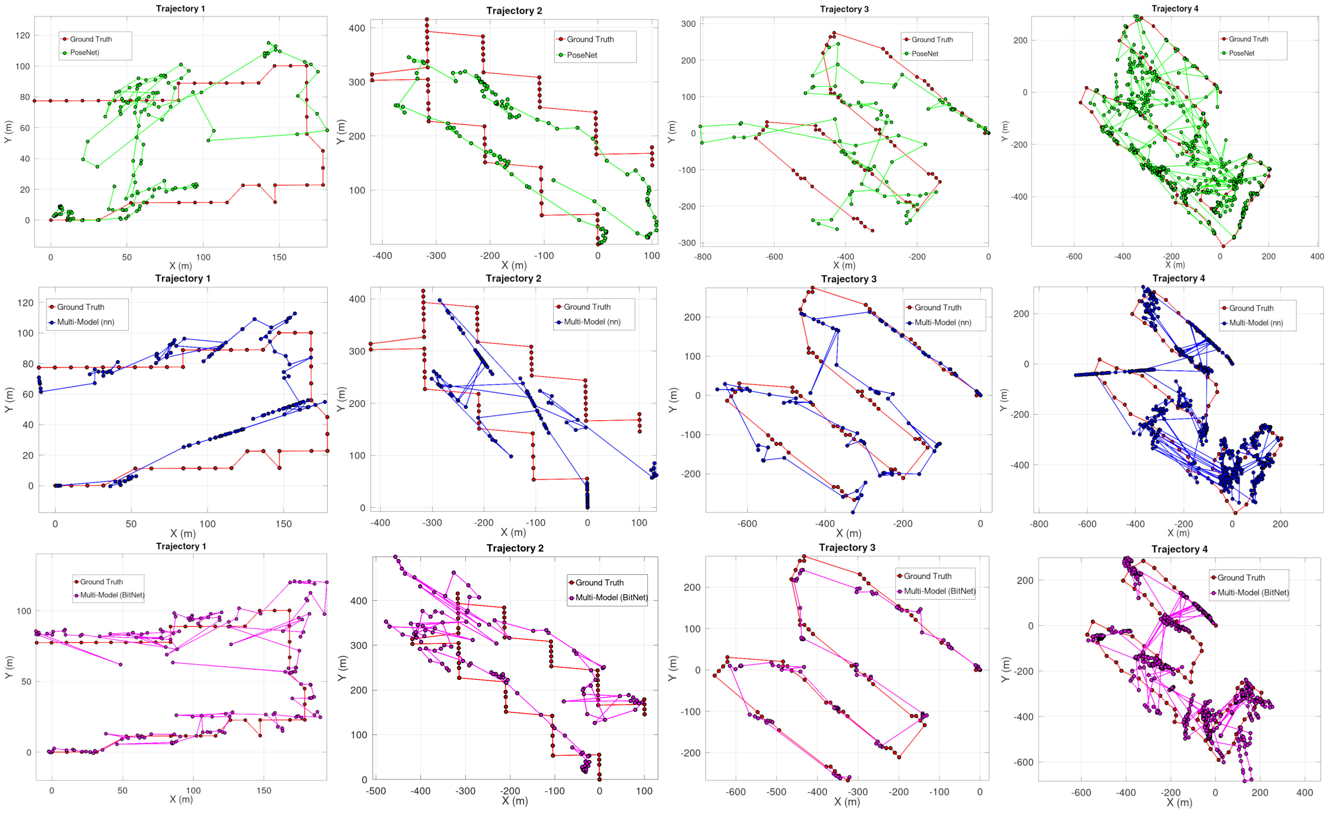

Figure 5: Pose estimation results: we show the testing trajectory in red and circle the ground truth poses. BitNet's multi-model approach obtained better pose estimation results than the other approaches, acquiring the UAV localisation in most images.

tory 2 with a processing time similar to PoseNet. The single model approach with a 4-layer network and BitNet exhibits faster processing time by handling a single model without sub-maps searching to get the aerial pose.

Table 6: Total Processing Time (ms). The best result is highlighted in bold.

Approach	Traj.1	Traj.2	Traj.3	Traj.4	Mean
Orbslam2	-	79.12	-	-	-
PoseNet	71.94	73.74	73.52	72.15	72.83
NN-Single	56.01	50.25	45.22	49.82	50.32
NN-Multi	61.45	61.43	64.60	64.18	62.91
BitNet-S	18.47	19.94	18.73	18.69	**18.96**
BitNet-M	51.80	52.22	49.91	51.85	51.44

Nevertheless, the multi-model approaches with a 4-layer network and BitNet require loading models w.r.t the sub-map index found. Thus, the processing time increases when handling a load of multiple models, even with small 4-layer networks, and the binarisation of the convolutional layers with BitNet. Despite this, our approach achieves good performance and a lower time than PoseNet, which is suitable for real-time localisation tasks with non-continuous and small datasets. Additionally, we can reduce the inference time by handling fewer training bits in the quantisation, but we will take it for future work.

5 Conclusion

We proposed a methodology for aerial localisation based on pose estimation with multiple models using BitNet. We leverage the quantisation to binarise the convolutional layers of BitNet, improving the training process to use 8-bit computes instead of float values. Besides, we implemented a localisation by sub-map identification comparing the features of the current images with those of the representative keyframes. This method allows us to load the corresponding BitNet model to estimate the pose from a single image. Our main contribution is the first implementation of BitNet to localisation tasks using aerial images captured by UAV and adapted to a continual training strategy based on progressive learning.

To evaluate the effectiveness of our approach, we compare the poses estimated with PoseNet, ORB-SLAM2, and a 4-layer network. Also, we performed traditional training with all the data to get a single model without sub-map searching. The total percentage error results demonstrated that our approach outperformed the others across all four trajectories, with a percentage error of 14% with multiple models and 21% using a single model. Besides, we achieved a processing time of 51 ms using multiple models and 18 ms with a single model using BitNet.

Finally, implementing a binary network for pose estimation accelerates the processing time using the quantisation method. Furthermore, with progressive learning, we ensure that we retain prior knowledge by using multiple models with flight coordinate information. It should be noted that quantisation reduces the precision of the training, obtaining a suitable result in the UAV localisation. However, the poses help recover the UAV in cases where the GPS signal is lost. In future work, we want to take advantage of and exploit binary networks, improving the processing time and accuracy of pose estimates.

Acknowledgements

The authors Aldrich A. Cabrera-Ponce and L. Oyuki Rojas-Perez thank CONAHCYT in Mexico for their scholarships with numbers 802791 and 924254.

References

[1] Diego Benjumea, Alfonso Alcántara, Agustin Ramos, Arturo Torres-Gonzalez, Pedro Sánchez-Cuevas, Jesus Capitan, Guillermo Heredia, and Anibal Ollero. Localization system for lightweight unmanned aerial vehicles in inspection tasks. *Sensors*, 21(17):5937, 2021.

[2] David Bohnenkamp, Jan Behmann, and Anne-Katrin Mahlein. In-field detection of yellow rust in wheat on the ground canopy and uav scale. *Remote Sensing*, 11(21):2495, 2019.

[3] Jie Qian, Kaiqi Chen, Qinying Chen, Yanhong Yang, Jianhua Zhang, and Shengyong Chen. Robust visual-lidar simultaneous localization and mapping system for uav. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021.

[4] 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.

[5] 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.

[6] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

[7] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024.

[8] 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.

[9] L Oyuki Rojas-Perez and Jose Martinez-Carranza. Deeppilot4pose: a fast pose localisation for mav indoor flight using the oak-d camera. *Journal of Real-Time Image Processing*, 20(1):8, 2023.

[10] 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.

[11] Ali Ayub, Zachary De Francesco, Jainish Mehta, Khaled Yaakoub Agha, Patrick Holthaus, Chrystopher L Nehaniv, and Kerstin Dautenhahn. A human-centered view of continual learning: Understanding interactions, teaching patterns, and perceptions of human users towards a continual learning robot in repeated interactions. *ACM Transactions on Human-Robot Interaction*, 2024.

[12] Hyung-Suk Yoon, Ji-Hoon Hwang, Chan Kim, E In Son, Se-Wook Yoo, and Seung-Woo Seo. Adaptive robot traversability estimation based on self-supervised online continual learning in unstructured environments. *IEEE Robotics and Automation Letters*, 2024.

[13] Jiun Fatt Chow, Basaran Bahadir Kocer, John Henawy, Gerald Seet, Zhengguo Li, Wei Yun Yau, and Mahardhika Pratama. Toward underground localization: Lidar inertial odometry enabled aerial robot navigation. *arXiv preprint arXiv:1910.13085*, 2019.

[14] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Covio: Online continual learning for visual-inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2464–2473, 2023.

[15] Paolo Cudrano, Xiaoyu Luo, and Matteo Matteucci. The empirical impact of forgetting and transfer in continual visual odometry. *arXiv preprint arXiv:2406.01797*, 2024.

[16] 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.

[17] 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.

[18] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. In *The International Symposium of Robotics Research*, pages 19–35. Springer, 2022.

[19] Youqi Pan, Wugen Zhou, Yingdian Cao, and Hongbin Zha. Adaptive vio: Deep visual-inertial odometry with online continual learning. *arXiv preprint arXiv:2405.16754*, 2024.

[20] Aldrich A Cabrera-Ponce, Martin-Ortiz Manuel, and Jose Martinez-Carranza. Continual learning for camera localization. *Machine Learning for Complex and Unmanned Systems*, pages 14–33, 2024.

[21] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3252–3262, 2021.

[22] Aldrich A Cabrera-Ponce, Manuel Martin-Ortiz, and Jose Martinez-Carranza. Continual learning for topological geo-localisation. *Journal of Intelligent & Fuzzy Systems*, 44(6):10369–10381, 2023.

[23] Aldrich Alfredo Cabrera-Ponce, Manuel Isidro Martin-Ortiz, and Jose Martinez-Carranza. Hierarchical continual learning for single image aerial localisation. In D. Moormann, editor, *14th annual International Micro Air Vehicle Conference and Competition*, pages 40–48, Aachen, Germany, Sep 2023. Paper no. IMAV2023-5.