

AstroLoc: Robust Space to Ground Image Localizer

Supplementary Material

1. Supplementary

In this Supplementary we provide additional information on the Lost-in-Space satellite problem (Sec. 1.1), the Historical space imagery localization problem (Fig. 2), and finally we showcase a large number of qualitative results Sec. 1.3.

1.1. Lost-in-Space satellite problem

Overview The goal of “lost in space” is to identify the location/orbit of nanosatellites using computer vision. Existing solutions either involve bulky and expensive ($\sim 10k\$$) GPS receivers, months-long tracking via radio communications through ground stations [4] (a considerable fraction of CubeSats remain unidentified more than 250 days after launch [3]), or use the recent VINSat [3], a computer vision solution that localizes the satellite only if/when it flies above a set of predefined landmarks. With our more general approach to localization, we can instead localize the queries (collected in real time by the satellite) anywhere on Earth, without being constrained to a set of predefined locations or landmarks.

We apply AstroLoc by reformulating the task as image retrieval, using the dataset from McCleary et al. [3]. This dataset contains Sentinel-2 mosaics, which are not cloudless or seamlessly processed, leading to sharp boundaries and occasional oblique views (see Fig. 3, row 1 col 2). Though Sentinel 2 captures images nadir-facing (straight down), some images in this set (Fig. 3, row 1 col 2) appear to have been transformed to mimic oblique views. These conditions increase the challenge of this dataset. All images cover an area between 35k and 55k sqkm, meaning that our database can be built with only satellite images of zoom level 9, reducing the number of images required for worldwide coverage to just 12k. We therefore construct an image retrieval task with all 2500 images from [3] as queries, and 12k database images.

Results Tab. 4 (of the main paper) shows AstroLoc’s superior performance in an out-of-distribution task, outperforming AnyLoc and EarthLoc by ~ 50 R@1. These gains highlight the impact of AstroLoc’s training setup improvements. Additionally, even the smaller AstroLoc-tiny model outperforms AnyLoc with only 2% of the parameters.

Embedded Use To better understand if this model could be actually used on a nanosatellite, we also provide results with a tiny version of AstroLoc, based on the smallest version of DINO-v2 and with output dimension of 512. This version achieves 36.7% R@1 and 74.5% R@100 with only 27M parameters (25% of the full AstroLoc), making it lightweight

enough to fit on an embedded system in a nanosatellite. The memory required to store database features with AstroLoc-tiny is $12k \times 512 \times 4 \times 4 = 98$ MB: 12k images with 512 dimensional features, each repeated 4 times due to the test time rotation augmentation and each element taking 4 bytes due to float32 encoding. Further memory reduction can be achieved through compression with methods like product quantization [2] for the features or by quantizing or pruning the model itself. Given the smaller memory footprint of this model and database, it is feasible to use AstroLoc to run real time localization of nanosatellites. Combined with the state estimation techniques from [3], AstroLoc can power a fast, accurate, and low cost orbit determination solution.

1.2. Historical space imagery localization

Historical imagery of Earth from space represents a unique source of data to understand how Earth has changed over decadal time spans. Similar to photographs taken by astronauts aboard the ISS, lots of historical imagery from space, taken from the Space Shuttle, lacks localization information. Although efforts at manually localizing these photos have been made, we note that (1) a large number of these images still lack location information, and (2) these images are only weakly labeled with a single location of any pixel within the image, which is often noisy and does not represent its full extent.

Given these shortcomings, we seek to understand the performance of APL systems in localizing early photographs (1981-1984) from the Space Shuttle, which were originally taken with film cameras and then digitized. As such, they have different photometric characteristics from more recent photography, and are known¹ to often require color correction (note the blue hue in many images in Fig. 2). We first precisely localize 704 images with the pipeline described in Sec. 3.1, and then compute localization results, reported in Tab. 5. We empirically find that AstroLoc achieves strong results even on these old photographs, showcasing its robustness to various types of domain changes. Comparing against performance on APL evaluation sets (Table 2 and Table 3 of the main paper), AstroLoc has the smallest drop in recall (at all N) on the out-of-distribution historical imagery out of all methods tested.

1.3. Qualitative Results

Fig. 1, 2, 3 show qualitative AstroLoc results on astronaut photos, historical Earth from space imagery from the Space Shuttle, and satellite imagery from the McCleary[3] dataset,

¹<https://eol.jsc.nasa.gov/FAQ/#photoQuality>

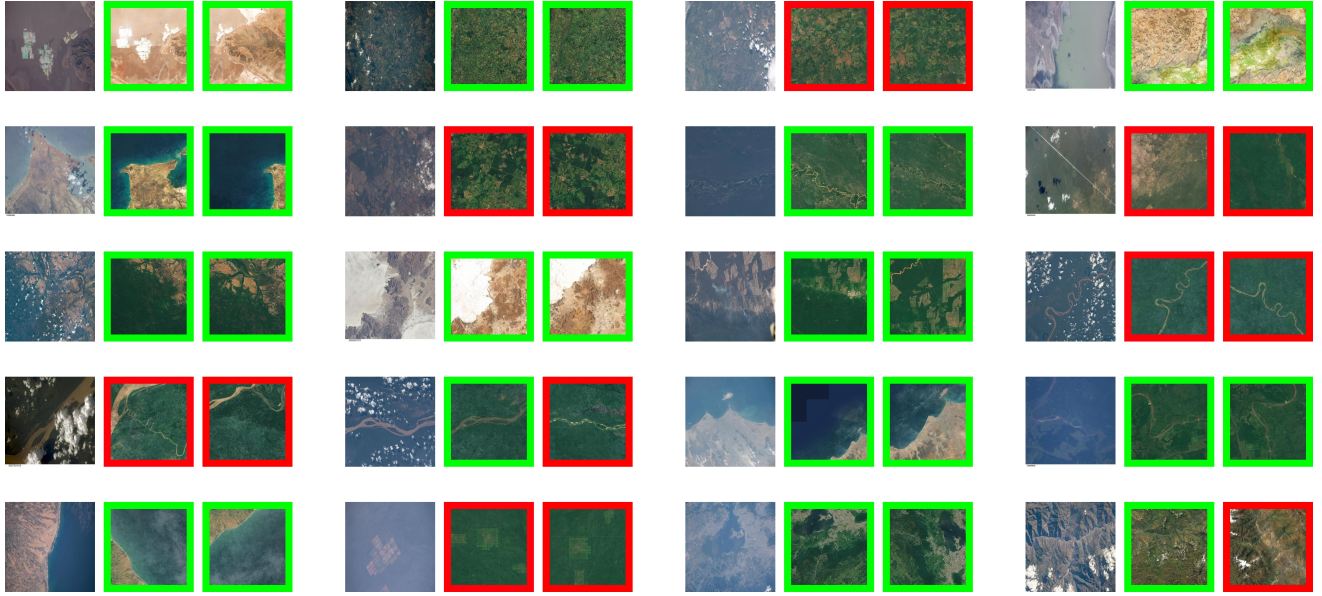


Figure 1. **Qualitative examples from the Amazon-L test set.** Each triplet shows one query and its top-2 predictions, red if wrong and green if correct.

respectively. There are four examples in each row, with each example showing the query photo (leftmost) and the top-2 retrieval results (*i.e.* most similar, as ranked by AstroLoc). Correct predictions, defined as those that have *any* overlap with the query, are outlined in green, and incorrect predictions are outlined in red.

Failure Modes The most common failure modes are (1) in heavily forested areas (Fig. 1), which share similar visual characteristics to other areas in the same (as well as nearby) forests; (2) in coastal regions where very little land is in the image, causing many mostly water database images to be retrieved (Fig. 3, bottom right); and (3) the presence of occlusion, typically in the form of clouds (Fig. 3, many examples). Training specifically to ignore cloud regions may improve performance in the case of occlusions. Database curation, removing images with minimal landmass, can help alleviate the mostly water retrieval results. Further hard negative mining may help disambiguate similar forested regions, though we note that in practice these similar (but not the same) predictions are typically filtered by a local feature-based verification method like EarthMatch [1]. Overcoming these failure modes will be explored in future work.

References

- [1] Gabriele Berton, Gabriele Goletto, Gabriele Trivigno, Alex Stoken, Barbara Caputo, and Carlo Masone. Earthmatch: Iterative coregistration for fine-grained localization of astronaut photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Work-*

shops, 2024. 2

- [2] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011. 1
- [3] Kyle McCleary, Swaminathan Gurumurthy, Paulo RM Fisch, Saral Tayal, Zachary Manchester, and Brandon Lucia. Vinsat: Solving the lost-in-space problem with visual-inertial navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 1, 3
- [4] Mark A. Skinner, Michael Coletti, Matthew C. Voss, Tomas Svitek, John C. Lee, Kerstyn Auman, Hemanshu Patel, and Eamonn J. Moyer. Mitigating cubesat confusion: Results of in-flight technical demonstrations of candidate tracking and identification technologies. *Journal of Space Safety Engineering*, 9(3):403–409, 2022. 1

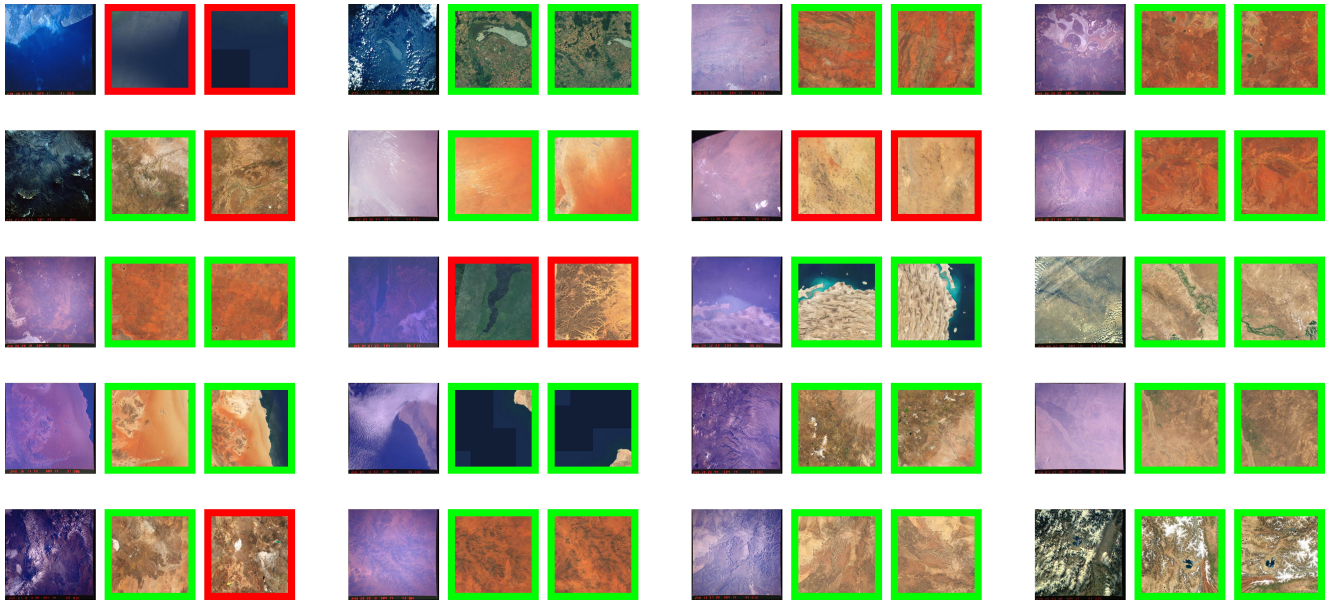


Figure 2. **Qualitative examples from the historical Space Shuttle imagery.** Each triplet shows one query and its top-2 predictions, red if wrong and green if correct. The queries were taken with analog cameras between 1981 and 1984 and then later digitized.

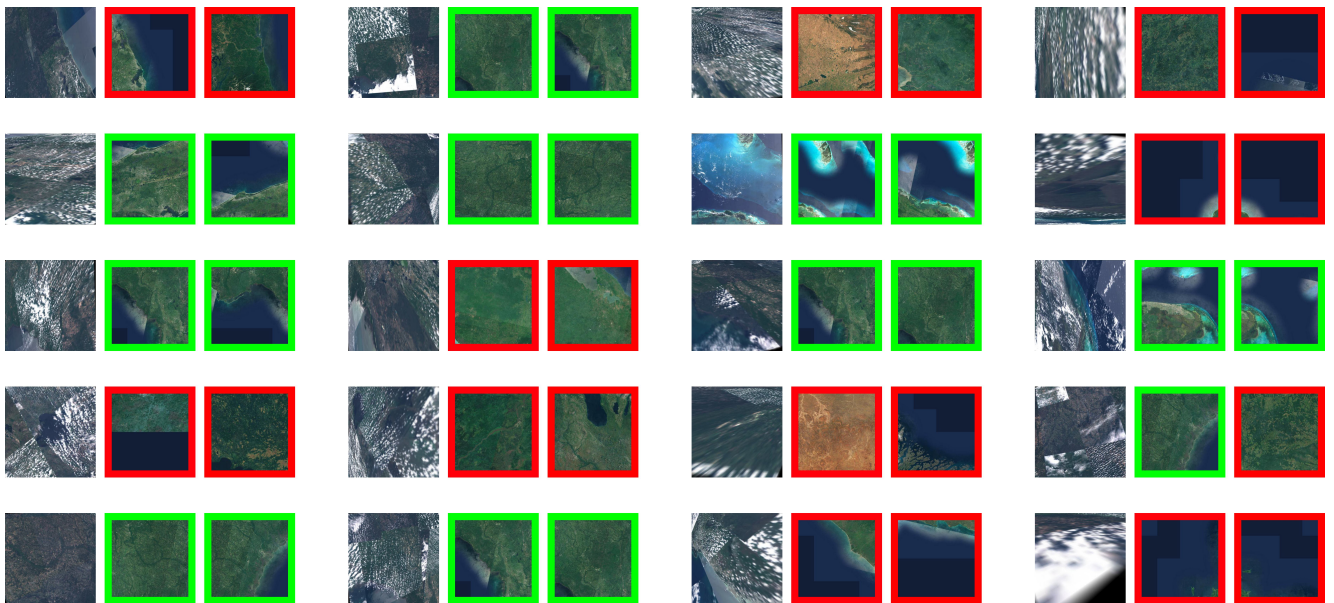


Figure 3. **Qualitative examples from the VINSat dataset [3].** Each triplet shows one query and its top-2 predictions, red if wrong and green if correct. Queries are mosaics of Sentinel 2 imagery.