AstroLoc: Robust Space to Ground Image Localizer

# AstroLoc: Robust Space to Ground Image Localizer

Gabriele Berton[1,2*] Alex Stoken Carlo Masone[1]
[1]Politecnico di Torino [2]Amazon

https://astro-loc.github.io/

## Abstract

*Thousands of photos of Earth are taken every day by astronauts from the International Space Station. Localizing these photos, which has been performed manually for decades, has recently been approached through image retrieval solutions: given an astronaut photo, the goal is to find its most similar match among a large database of geo-tagged satellite images, in a task called Astronaut Photography Localization (APL). Yet, existing APL approaches are trained only using satellite images, without taking advantage of the millions of open-source astronaut photos. In this work we present the first APL pipeline capable of leveraging astronaut photos for training. We first produce full localization information for 300,000 manually weakly-labeled astronaut photos through an automated pipeline, and then use these images to train a model, called AstroLoc. AstroLoc learns a robust representation of Earth's surface features through two objective functions: pairing astronaut photos with their matching satellite counterparts in a pairwise loss, and a second loss on clusters of satellite imagery weighted by their relevance to astronaut photography through unsupervised mining. AstroLoc achieves a staggering 35% average improvement in recall@1 over previous SOTA, reaching a recall@100 consistently over 99% for existing datasets. Moreover, without fine-tuning, AstroLoc provides excellent results for related tasks like the lost-in-space satellite problem and historical space imagery localization.*

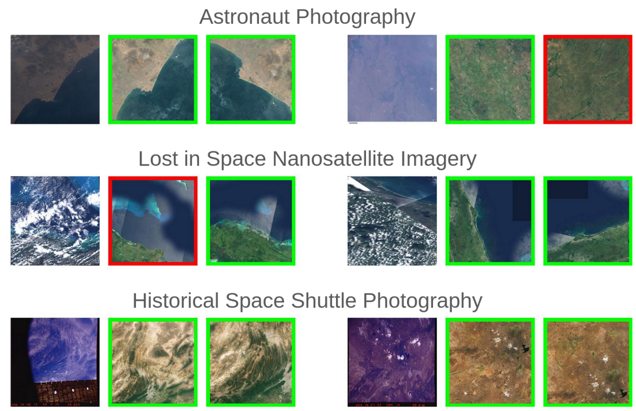


Figure 1. **One model, many space to ground applications.** We train a single model, AstroLoc, that succeeds in multiple space-based image retrieval settings: astronaut photography localization, "lost in space" orbit determination, and historical (Space Shuttle) photography localization. In this figure, each group of 3 images represent a query and its top-2 predictions from searching over a worldwide database of millions of satellite images. Correct predictions in green, wrong predictions in red.

## 1. Introduction

Earth observation is the process of collecting information about the Earth's surface, which is vital for monitoring the state of our planet. Among the multitude of systems used to acquire Earth observations data, most of which rely on automatic collection by nadir-facing (straight-down) satellites, manual acquisition of photos by astronauts aboard the International Space Station (ISS) is a clear outlier. This imagery has distinctive properties including (1) a wide range of spatial resolutions (up to 2 meters per pixel), (2) the potential for oblique perspectives to observe topography and height (*e.g.*, measure the height of a volcanic eruption to divert flights), (3) various illumination conditions (whereas satellite imagery is typically sun-synchronous, always capturing the same area at the same time of day), and (4) offers the highest resolution open source Earth observations data. Furthermore, due to the ISS's 90 minute repeating orbit, astronauts can be tasked with imaging for near real time disaster response, and they can use human intuition to focus on relevant areas and events, making the photographs very information-dense, whereas satellites commonly collect images systematically, regardless of semantic value. All these characteristics make this data source a perfect complement to satellite data and a crucial component for climate science [25], atmospheric phenomena [18, 38], urban planning [9, 14, 28, 30], and, most importantly, disaster management and response [31][1].

*Work was done outside of Amazon

[1]https://storymaps.arcgis.com/stories/947eb734e811465cb0425947b16b62b3

Although over 5 million photos have been collected from the ISS since the beginning of its operations, unfortunately their full potential has yet to be unlocked. The complication is that astronaut photos, unlike satellite imagery, are not automatically geolocated w.r.t. the Earth's surface. Even though the position of the ISS can be determined from the timestamp of the photo and the ISS orbit path, an astronaut can point their camera (a standard hand-held DSLR) toward any location on Earth within their vast visibility range, which spans roughly 20 million square kilometers (sqkm)[2]. Therefore, localizing a photo that covers an area of 100 sqkm (*i.e*. 0.0005% of the visible/search area) is like finding a needle in a haystack. For example, when an astronaut is orbiting above Texas and photographs a wildfire, the burning area could be anywhere from Canada to Mexico depending on camera orientation, and the only way to know where is by geolocating the image.

The potential value these photos hold when geolocated has led to a concerted manual localization effort, with experts and citizen scientists localizing over 300,000 of these images – a process defined by NASA as a "monumentally important, but monumentally time-consuming job"[3]. Nevertheless, manually geolocating a single astronaut photo can take hours (hundreds of thousands of human-hours have been spent on hand-labeling 300k of them), effectively making this procedure inadequate given the ever-growing number of images that are being collected (up to 10k per day). This problem has prompted recent investigation into automating the task of **Astronaut Photography Localization (APL)** with deep learning, combining image retrieval and image matching techniques [5, 33, 34]. Image retrieval is used as a first step, to search within a database of precisely geo-referenced satellite images for those that are most similar to the astronaut photo (called the *query*). With such a system, a query can be coarsely localized in less than a second, making it a viable solution for such a large scale problem. Afterwards, the retrieval candidates may be refined using more computationally demanding matching techniques. Although these pipelines have shown satisfactory speeds for APL, their performance show a wide margin for improvement, and we hypothesize that this is due to existing APL models being trained *only* on satellite images [6]. This is a critical limitation, considering that the end goal is to georeference astronaut photos, not satellite images.

We hypothesize that it would be beneficial to exploit the available 300,000 hand-labeled open-source astronaut photos to train the image retrieval model. The difficulty with doing so is that these photos are only weakly annotated, *i.e*., the manual label only provides the geographic location (latitude,longitude) of a random point close to the center of the image. Therefore, these weakly labeled photos cannot be directly used in a metric learning framework, where the model must be provided with pairs of queries (astronaut photos) and database (satellite) images from the same location. In order to meet this criteria, we need these photos to be labeled with their full footprint (*i.e*. precise GPS location of its four corners), so that we can exactly determine which images from the database can be used as positive samples (*i.e*., if they have a substantial overlap w.r.t. the training queries). The concept of weak and full labels is visualized in Fig. 2.

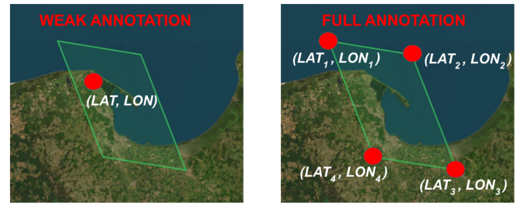


Figure 2. **Visual example of weak (manual) annotation and full annotation of an astronaut photo.** Weak annotation is the geographic coordinates of a single point, which does not provide information about the image's size (*i.e*. it could cover a town or an entire continent), whereas full annotation provides coordinates for all 4 corners (called *footprint*), from which the coordinate of any pixel within the image can be easily calculated (pixel-wise label).

In light of these considerations, we first produce a precise annotation of these 300,000 photos, leveraging their weak annotation and a state-of-the-art matching pipeline to label their full footprint. Then, we demonstrate the importance of using this newly annotated data for training in two ways. First, we create pairs of matching astronaut-satellite images, and use them in a pairwise contrastive loss that takes advantage of the fact that the two images within each pair come from different domains. The creation of these pairs relies on our new annotations, *i.e*., we select a pair if the two images have enough intersection over union on Earth's surface. Even with this loss that takes advantage of the newly labeled 300,000 astronaut photos, it is still important to fully leverage the much larger set of all 5.3M geo-referenced satellite images (not just the ones overlapping astronaut photos). However, we observe that the satellite images are uniformly distributed on the surface of the Earth, whereas the photos from astronauts tend to be unevenly distributed, with higher concentrations in salient areas like volcanoes, glaciers and lakes. Therefore, we propose a new mining technique, which allows for a weighted sampling of the large satellite image collection according to the *distribution* of astronaut photos, which we call Unsupervised Mining, and, to the best of our knowledge, is the first mining technique of its kind. We then make of use of this data with a second contrastive loss.

We find that our training pipeline, yielding a model which we call **AstroLoc**, leads to a model so powerful that

[2]Considering that the ISS orbits at an altitude of 415 kilometers.

[3]We highly encourage the reader to view this clip `https://www.youtube.com/watch?v=drrP_Iss0gA&t=295s`

it saturates existing test sets (recall@100 above 99%), and brings us to propose new, more challenging test sets that better reflect the task of APL in the real world. Furthermore, AstroLoc is useful in the real world, and it has already been used to localize hundreds of thousands of images available here[4]. Thanks to AstroLoc, in a few months the backlog of non-localized astronaut photos will be nearly empty for the first time since the launch of the ISS, with the exception of a small fraction that still result in failure cases. Finally, we note that AstroLoc also shows great potential for other applications like ”lost-in-space” satellite orbit determination and historical space image localization.

## 2. Related Work

**Astronaut and satellite imagery localization** The task of localizing photos taken from Earth's orbit has been studied in a number of domains. Straub and Christian [35] examine the possibility of matching the coastline of imagery from satellites to enhance their autonomous navigation capabilities. Schockley and Bettinger [29] take a similar approach, but instead propose using terrestrial illumination matching. The problem is more recently addressed by McCleary et al. [22], which selects a set of landmarks on Earth to achieve the same goal. They note that common hardware solutions for localization are expensive (∼10k$) for nanosatellite developers, making a software solution a game changer for the industry. For the similar task of localizing astronaut photography, which was historically performed by hand [33], Stoken and Fisher proposed an automated solution leveraging image matching [33], in a framework that was then expanded to localize night-time imagery [34].

While these solutions have advanced the field of Earth from space localization, they are limited, as the matching-based methods can not achieve the scalability and speed required to localize astronaut photographs in real time. Berton et al. [6] proposes approaching the task as an image retrieval problem, where the database is composed of satellite images of known location. This leads to EarthLoc, the first APL-focused retrieval model. In addition, AnyLoc [19], a universal place recognition model, has been shown to produce strong performance and robustness to the query-database domain gap.

**Unmanned aerial vehicle localization** A closely related task is that of unmanned aerial vehicle (UAV) localization. Similar to Astronaut Photography Localization (APL), it is approached through cross-domain image retrieval, where the database is often made of satellite imagery and the queries are photos collected by a drone. The goal is to efficiently find the best prediction for a given query in order to localize the drone in real time [40]. Among notable examples from the literature, Bianchi and Barfoot [7] introduced a method utilizing autoencoded satellite images, significantly reducing storage and computation costs while maintaining robust localization performance. Dai et al. [8] proposed an end-to-end framework, FPI, that directly identifies UAV locations by matching UAV-view images with satellite-view counterparts, streamlining the localization process. Li et al. [20] developed a transformer-based adaptive semantic aggregation method, whereas He et al. [16] make use of foundation models to get an estimate of the location to then refine it with visual-inertial odometry algorithms.

Further improvements in UAV localization are achieved by orientation-guided methods, as shown by Deuser et al. [11], who use orientation-guided contrastive learning to improve UAV-satellite image alignment. Zhu et al. [41] proposed a modern backbone architecture optimized for efficient UAV geo-localization, enhancing both speed and accuracy for UAV applications.

| Type | Queries | Database |
|---|---|---|
| Acquisition | manual, by astronauts | automatic, by satellites |
| Number (*Annotation*) | 5M (*none*), 300k (*weak*) | 5.2M (*full*) |
| Extent | 40 to 1,357,493 sqkm | 346 to 391,776 sqkm |
| Extent percentile (5 / 95) | 239 / 39,729 sqkm | 465 / 16,026 sqkm |
| Occlusions | Yes (ISS hardware, clouds) | No |
| Obliquity & distortion | Yes | No |
| Illumination changes | Yes (*e.g.* day/dawn/sunset) | No |
| Years | 2000 to 2024 | 2018 to 2021 |
| Source | https://eol.jsc.nasa.gov/ | https://s2maps.eu |
| Coverage | scattered, biased on glaciers, volcanoes etc. | dense, worldwide |

Table 1. **Overview of data.** Information reported about satellite imagery refers specifically to the data used in this project, not satellite imagery in general. The labeling refers to the *weak* and *full* labeling shown in Fig. 2.

## 3. Data

Our goal is to localize the extent of **astronaut photos *(queries)*** given a worldwide map of geo-referenced **satellite images *(database)***. Astronaut photographs represent one of the most diverse and long standing Earth observations data sources. The dataset contains a wide range in spatial resolution, field of view, illumination (including night imagery), and obliquity (astronauts can tilt the camera and take oblique photos). These images have many applications in numerous fields [9, 14, 18, 25, 28, 30, 38], and critically in disaster management and response [31][5].

For database images we follow [6] and use a yearly-composited set of cloudless open-source satellite imagery from Sentinel-2, named *S2*. We use tiles from zoom levels 8-12 [26] (expanding the range of zoom levels used in [6]), to provide a thorough train and test set that reflects the extents of the query images. Characteristics of astronaut and satellite imagery are compared in Tab. 1.

[4]Photos available at: https://eol.jsc.nasa.gov/ExplorePhotos/

[5]https://eol.jsc.nasa.gov/Collections/Disasters/ShowDisastersCollection.pl

### 3.1. Training Set

In contrast to previous work, we propose taking advantage of the 300k manually localized astronaut photos for training. The idea is to pair these photos with matching satellite imagery from the database (*i.e.* with enough Intersection over Union, or IoU) and apply contrastive training to train models that are robust to the domain differences between astronaut and satellite imagery. Unfortunately, the available manual localization is only a weak annotation (see Fig. 2), which does not provide enough information to compute IoU. To overcome this limitation, we estimate the precise footprint for each query image as follows: for each manually localized query, we select as potential positives all the database images which contain the weak label point at each zoom level, and we rotate these potential positives by 90°, 180° and 270° to maximize the chance of finding a similar image among the potential positives. In practice, given five zoom levels, four rotations, and the fact that each point is covered by four images (there is overlap among the tiles), there are $5 \times 4 \times 4 = 80$ potential positives per query. Note that different zoom levels are required because it is impossible to estimate an image extent with only a weak label, and a photo could cover an area as small as a city or as big as a continent. We then perform image matching with SuperPoint [10] + LightGlue [21] and the EarthMatch pipeline [5] to get the footprint coordinates of each query.

This method was successful for 221k queries, with the remaining being not localizable due to either errors in the manual label, too much obstruction from clouds, or the image being of the horizon (*i.e.* Earth limb photograph, with two corners in outer space, hence an invalid footprint). With this now precisely localized collection of astronaut photography, we pair each query with the all database images with an IoU over $t_{iou} = 0.2$ producing 865k query-database training pairs. We then discard any pair containing a query that is included within the evaluation sets to avoid any data leakage.

### 3.2. Evaluation Sets

Evaluation sets are made of queries and the associated database to localize them. Six evaluation sets were proposed in [6], covering geographical areas across five continents that mirror astronaut photography use cases like land change study and flood monitoring. However, as detailed in their limitations [6], the test sets only contain queries whose projected area on Earth's surface is between 5,000 and 900,000 sqkm, which represents only 22% of all astronaut photo queries, as shown in Fig. 3.

We therefore propose new evaluation sets which include *all* available geolocated queries within an evaluation area, as well as add zoom levels 8 and 12 to the database to match the wider range of query areas. To avoid introducing complexity, we use the same regions as the test sets from [6].

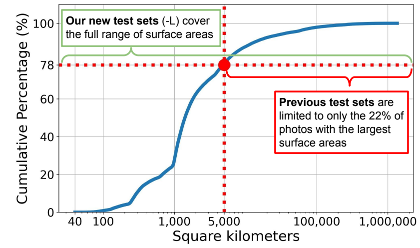


Figure 3. **Distribution of queries by covered area.** The red mark at 5,000 sqkm shows that 78% of astronaut photographs cover an area lower than 5,000 sqkm. Thus, the test sets in Tab. 2 do not contain this vast majority of queries, leading us to propose test sets containing all queries used for experiments in Tab. 3.

These datasets were named after the geographic location of their center, like Texas, Gobi, and Amazon, so we call these new extended versions Texas-L (L for Large), Gobi-L, etc.

## 4. Method

### 4.1. Setting

**Task** Astronaut Photography Localization (APL) is the task of estimating the geographic location covered by a photo (query) taken by astronauts. APL can be approached through image retrieval: for each query, we can estimate its location by finding the most similar match in a large database of geolocated images of Earth.

**Formalization** Throughout this paper, we rely on three sets of images:

- The **database** $\mathcal{D} = \{d_1, d_2, \dots, d_{N_\mathcal{D}}\}$ is a set of $N_\mathcal{D}$ *satellite* images, each characterized by their RGB content and the coordinate label of their four corners, called the *footprint*. Formally, the footprint for a generic image $d_i \in \mathcal{D}$ is denoted as $F_i = \{lat_1, lon_1, lat_2, lon_2, lat_3, lon_3, lat_4, lon_4\}$ where $lon_k$ and $lat_k$ denote the longitude and latitude of the $k$-th corner, respectively (Fig. 2).
- The **training queries** $\mathcal{Q} = \{q_1, q_2, \dots, q_{N_\mathcal{Q}}\}$ is a set of $N_\mathcal{Q}$ astronaut photos, with RGB content and footprints (localized as explained in Sec. 3.1).
- The **test queries** $\mathcal{T} = \{t_1, t_2, \dots, t_{N_\mathcal{T}}\}$ is another set of $N_\mathcal{T}$ astronaut photos, disjoint from $\mathcal{Q}$, each characterized by its RGB content and timestamp. The timestamp is crucial in determining the location of the ISS when the photo was taken, which enables narrowing down the search space from the entire world surface area (510M sqkm) to the local area visible from the ISS at that time (20M sqkm). Note that due to cosmic radiation bit flips, the timestamp can sometimes be incorrect: in such cases, it is necessary to conduct a worldwide search (see Sec. 5.5).

**Training with Astronaut Photos (Queries)** While various approaches have been used for APL, no previous work has taken advantage of the huge dataset of 300k astronaut

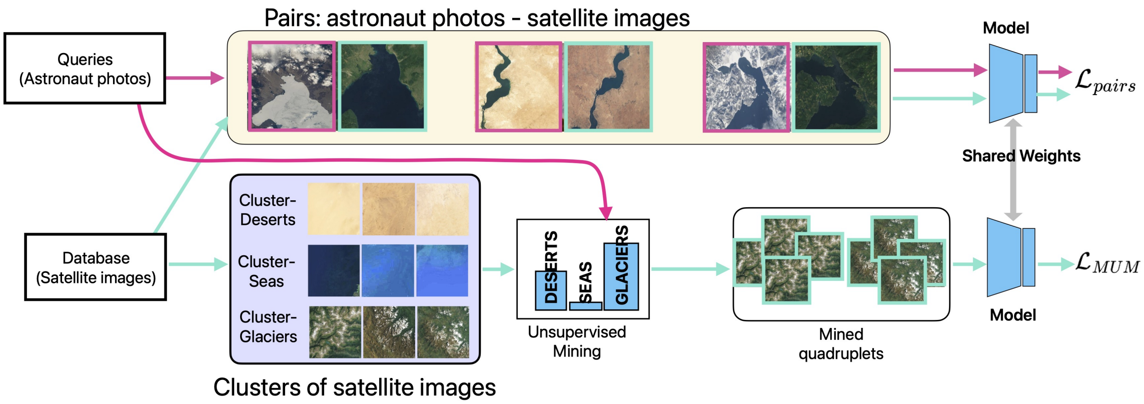


Figure 4. **AstroLoc's training pipeline.** The upper branch feeds pairs of matching query-database images to the pairwise loss (Sec. 4.2). The lower branch (Sec. 4.3) first creates clusters of satellite images, then queries are assigned to these clusters, which are sampled according to how many queries are assigned to each cluster (unsupervised mining). A training batch (mined quadruplets, *i.e*. tuples of 4 images) is sampled from a single cluster and fed to the Multi-similarity Unsupervised Mining (MUM) loss. **Queries are not fed to the MUM loss, and are not used to compute the clusters** — they are only used to sample training data from a closer distribution to the queries'.

photos labeled by human experts. We posit that their use at training time can lead to sizeable improvements in performance, and we present two novel techniques to use this data. In Sec. 4.2 we show how we use pairs of matching query-database images (*i.e*., depicting the same location) to implement a *pairwise loss* (see Fig. 4, upper branch). Additionally, we use the photos a another time, to inform a second loss this time computed on tuples of satellite images that are sampled using the distribution of the queries (see an overview in Fig. 4 bottom). This allows us to use the entire set of database images for training (whereas the *pairwise loss* can be applied only on areas covered by astronaut photos) while still leveraging the queries (see Sec. 4.3).

## 4.2. Query-Satellite Pairwise Loss

Given the cross-domain nature of the problem, we aim to take matching images from the different distributions and applying contrastive learning to encode their relationship in feature space. To this end, we create a batch $\mathcal{P} = \{p_1, p_2, \dots, p_B\}$ of query-database pairs $p_i = (q_i, d_i) \in (\mathcal{Q}, \mathcal{D})$ that have an intersection over union (IoU) higher than a threshold $t_{iou}$[6] (see examples in Fig. 4 top). We ensure that within a batch there are no two pairs with geographic overlap, so that for a pair $p_i \in \mathcal{P}$ all the images from other pairs $p_j \in \mathcal{P} \backslash \{p_i\}$ are negative examples.

With this setting, we define a contrastive loss that comprises two terms: an attraction term $\mathcal{L}_{pos}$ that acts between images in a matching pair, pulling their representations closer, and a repulsion term $\mathcal{L}_{neg}$, that is applied between images from different pairs (*i.e*., non matching), pushing them apart. Formally, given a similarity measure $\mathcal{S}(I_1, I_2)$ between the features of two image (*e.g*., the cosine similarity), we define the positive loss term as

$$\mathcal{L}_{pos} = \frac{1}{\alpha_1 B} \sum_{i=1}^{B} \gamma(\alpha_1, q_i, d_i) \tag{1}$$

where $\alpha_1 > 0$ is a gain and

$$\gamma(x, y, z) = \log \left[1 + e^{-x \times \mathcal{S}(y,z)}\right] \tag{2}$$

is the attraction function.

For the negative loss term, let us first denote with $\mathcal{P}_Q = \{q_1, q_2, \dots, q_B\}$ the set of all queries in the batch $\mathcal{P}$, and with $\mathcal{P}_D = \{d_1, d_2, \dots, d_B\}$ the set of all database images in $\mathcal{P}$. With this notation, we define $\mathcal{L}_{neg}$ as

$$\begin{aligned}\mathcal{L}_{neg} = \frac{1}{\beta_1 B} \sum_{i=1}^{B} \Big[&\varphi(\beta_1, q_i, \mathcal{P}_Q \backslash \{q_i\}) + \varphi(\beta_1, q_i, \mathcal{P}_D \backslash \{d_i\}) \\ &+ \varphi(\beta_1, d_i, \mathcal{P}_Q \backslash \{q_i\}) + \varphi(\beta_1, d_i, \mathcal{P}_D \backslash \{d_i\})\Big]\end{aligned} \tag{3}$$

where $\beta_1 > 0$ is a gain and

$$\varphi(x, y, \mathcal{Z}) = \log \left(1 + \sum_{j=1}^{|\mathcal{Z}|} e^{x \times \mathcal{S}(y, \mathcal{Z}_j)}\right) \tag{4}$$

is the repulsion function. The overall loss is

$$\mathcal{L}_{pairs} = \mathcal{L}_{pos} + \mathcal{L}_{neg} \tag{5}$$

## 4.3. Unsupervised Mining

Although the use of the Query-Satellite Pairwise Loss by itself leads to state-of-the-art results (see Sec. 5), it does not fully exploit the large quantity of available satellite imagery

[6]For the sake of readability, we abused the notation by implying that a matching query and database images from the sets $\mathcal{Q}$ and $\mathcal{D}$ have the same index as the index of the pair they belong to. This trick to simplify the notation is equivalent to have preliminarily sorted the sets $\mathcal{Q}$ and $\mathcal{D}$.


Figure 5. **Examples of training batches, using the three different sampling solutions presented.** For each solution, we show two examples of batches with batch size 12 (*i.e.* 3 quadruplets), so that each image has 3 positives and 8 negatives. **Solution 1** leads to training a non-robust model, because of the lack of hard negatives, as quadruplets are randomly sampled and are often very diverse between each other. **Solution 2** provides hard negatives, but it creates batches with uninformative and feature-less images (*e.g. batch 1* is from a cluster of images of seas next to coasts) which hurt the training process. **Solution 3** allows to simultaneously provide hard negatives, discard uninformative batches, and focus on the most salient imagery, by sampling images from clusters that reflect the queries' distribution.

worldwide, as it only uses images that overlap with astronaut photographs. Therefore, we propose to use a second loss: in the next paragraphs, we explain the design of this innovative technique, starting from a naive implementation and developing into our full-fledged MUM loss.

**Solution 1: naive sampling** The simplest approach to train a retrieval model on large volumes of satellite imagery would be to directly apply a contrastive loss. Common training pipelines rely on using quadruplets [2, 23] (*i.e.* 4 images) from each class, and stacking multiple quadruplets within a batch to ensure that each image has exactly 3 positives (from the same class) and a multitude of negatives. However, this naive approach would not train a robust model, because random sampling of classes would lead to very few hard negatives (see Fig. 5 left), which are crucial to train robust image retrieval models.

**Solution 2: database clustering** To improve over naive sampling we could first compute features for all images within our database, and then cluster them with a k-means algorithm in feature space. This leads to the creation of $K$ clusters that share similar visual characteristics: for example one cluster containing images of forests, another one with images of deserts, and so on. These clusters can then be used to create training batches: for example there would be one batch of forest images, one batch of desert images, et cetera, leading to harder negatives within batches. While this solution leads to better results (see [6]), one issue would arise: some batches may contain only uninformative images, like feature-less images of deserts or seas, (see Fig. 5 center), which can hurt the learning process.

**Solution 3: unsupervised mining** We finally propose to not only create clusters from database images, but also to sample them according to the distribution of the astronaut images: intuitively this means that, as astronauts take very few photographs of deserts, and many photographs of glaciers, the clusters of desert-images will be sampled seldom, while the clusters with glaciers will be sampled more often (see Fig. 5 right).

To formalize this concept, we split the database images into a set of $K$ clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$ using a standard k-means algorithm in feature space, such that

$$\bigcup_{k=1}^{K} \mathcal{C}_k = \mathcal{D} \tag{6}$$

Secondly, we compute features for the training queries (*i.e.* astronaut photos), and assign each to one of the K clusters $C$: we then denote with $b_1, \ldots, b_K$ the size of the bins derived from assigning the features of the training queries to the $K$ clusters. Clusters are then sampled according to

$$k \sim B(Q, 1, k) \tag{7}$$

where $B$ denotes the weighted distribution according to the bins $b_1, \ldots, b_K$, *i.e.* such that

$$Pr(k) = \frac{b_k}{\sum_{i=1}^{K} b_i} \tag{8}$$

meaning that clusters with more queries are sampled more often for the creation of our training batches. We emphasize that the training batches are made only of database images, and that the queries are only used to calculate how often each cluster should be sampled, as shown in Fig. 4. Finally, given a series of quadruplets from cluster $\mathcal{C}_K$ (sampled according to Eq. (7)), we plug them into a contrastive loss (in our case a Multi-Similarity loss [37]). Therefore, the second loss in our pipeline is

$$\begin{aligned}\mathcal{L}_{MUM} = \frac{1}{4H}\sum_{i=1}^{H}\Bigg[\frac{1}{\alpha_2}\sum_{j=1}^{4}\log\Bigg(1+\sum_{d\in h_i\setminus\{h_{i_j}\}} e^{-\alpha_2 \mathcal{S}(h_{i_j},d)}\Bigg)\\ +\frac{1}{\beta_2}\sum_{j=1}^{4}\log\Bigg(1+\sum_{d\in \mathcal{H}_k\setminus\{h_i\}} e^{+\beta_2 \mathcal{S}(h_{i_j},d)}\Bigg)\Bigg]\end{aligned} \tag{9}$$

where $\alpha_2$ and $\beta_2$ are positive hyperparameters, and MUM stands for **M**ulti-similarity with **U**nsupervised **M**ining.

Among the large number of mining techniques in literature, our Unsupervised Mining stands out in two ways:

1. it is the only mining that uses one distribution (*i.e.* $Q$) to learn how to sample from another distribution (*i.e.* $\mathcal{D}$);
2. unlike common mining techniques used in localization and place recognition pipelines [1, 3, 15, 24], our miner does not require labels of queries, so we could potentially use all of the 5M existing astronaut photos (even the unlabeled ones). In practice we use only training queries to ensure avoiding any kind of test data leakage.

Finally, our final loss is the sum of the two losses:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{pairs} + \lambda_2 \mathcal{L}_{MUM} \tag{10}$$

# 5. Experiments

## 5.1. Implementation Details

**Training** We set the hyperparameters as follows: $t_{iou} = 0.2, \alpha_1 = 1, \alpha_2 = 1, \beta_1 = 50, \beta_2 = 50, \lambda_1 = 1, \lambda_2 = 1, K = 50$, batch size = 48 (48 pairs for pair loss, 48 quadruplets for MUM loss), learning rate = 5e-5, Adam optimizer. We use as architecture a DINO-v2-base backbone [27] with SALAD [17] and a linear layer to reduce feature dimensionality (from 8448 to 2048) (*i.e.* a similar model to AnyLoc [19] while being over 10 times lighter than AnyLoc, which uses DINO-v2-giant). Training runs for 30k iterations. As in [6], the Texas dataset is used as validation. The features for our Unsupervised Mining are computed periodically (every 5000 iterations) while the model is undergoing training, as standard practice in virtually every mining procedure [1, 3, 6, 24]. With this setup, mining increases the training time by less than 10%.

**Evaluation** We follow the protocol from [6] by doing image retrieval with an augmented dataset (*i.e.* applying four 90° rotations to each image), and using as metric the recall@N, as the percentage of queries where at least one of the top-N predictions is a correct match to the query.

## 5.2. Results

In Tab. 2 we report experiments on the test sets proposed by [6]. For fairness, we include results with a more powerful version of EarthLoc that uses the same architecture and training data of AstroLoc, which we refer to as EarthLoc++.

Results clearly show that, whereas previously there was no dominant method between EarthLoc and AnyLoc, the introduction of AstroLoc provides a new state-of-the-art model which significantly outperforms all previous, achieving a near perfect (above 99%) recall@100 on all six evaluation sets. Note that the recall@100 is more relevant in the real world than recall@1, since it is common practice to re-rank the top-N predictions with image matching methods [5, 33, 36]. Some qualitative results of queries and their predictions are in Fig. 1, with many more in the supplementary.

**New test sets** Here we compute experiments on the newly extended test sets described in Sec. 3.2, which provide a setting that is more similar to the real world scenario and more challenging. Results, reported in Tab. 3, illustrate that AstroLoc is able to outperform previous models by an even wider margin, and presents recall@100 consistently above 96% even in these more comprehensive and complex cases.

## 5.3. Other Space to Ground Use Cases

In this paper we test the robustness of AstroLoc by performing experiments on two related tasks: the lost-in-space problem and historical space imagery localization. Given the lack of space, we present a thorough explanation of the tasks, motivations, experimental details and results in the supplementary, while only offering a brief summary here.

**The Lost-in-Space satellite problem** has the goal of identifying the location/orbit of nanosatellites through computer vision [22], requiring photos be searched for over the entire planet. We use as queries the images from McLeary et al. [22], and a worldwide database at zoom level 9 (12k images). Results in Tab. 4 highlight the robustness and adaptability of AstroLoc, which is the only method to achieve a R@1 over 50% on the retrieval-formulation of lost in space, outperforming all other methods by at least 45% of R@1.

**Historical space imagery localization** Historical imagery of Earth from space is a unique source of data to understand how Earth has changed over decadal time spans. We perform experiments on 704 queries from early Space Shuttle days (1981-1984), on a worldwide database, and report results in Tab. 5. We empirically find that AstroLoc achieves strong results even on these old film photographs, showcasing its robustness to various types of domain changes.

## 5.4. Ablations

We compute ablations on the losses and mining in Tab. 6, where results clearly show the strong impact from each module in our pipeline. Most importantly, the combination of these components presents the optimal characteristic of orthogonality, in that their mixture has notably higher results than each of the singular elements.

## 5.5. Toward Worldwide Search

Astronaut photographs are associated with a timestamp which, combined with the known orbit of the ISS, can be used to obtain the position of the camera when the photo was taken, allowing the search space to be reduced from the entire extent of the Earth to just 20M sqkm (the area of Earth's surface visible from the ISS). However, due to high-energy cosmic particles causing bit-flips in cameras,

| Method | Texas (6k / 34k) | | | Alps (2k / 53k) | | | California (4k / 30k) | | | Gobi Desert (1k / 54k) | | | Amazon (1k / 19k) | | | Toshka Lakes (2k / 63k) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R@1 | R@10 | R@100 | R@1 | R@10 | R@100 | R@1 | R@10 | R@100 | R@1 | @10 | R@100 | R@1 | R@10 | R@100 | R@1 | R@10 | R@100 |
| Nadir | 2.4 | - | - | 1.2 | - | - | 2.4 | - | - | 1.8 | - | - | 3.1 | - | - | 1.4 | - | - |
| Random Choice | 0.2 | 1.7 | 15.5 | 0.1 | 1.1 | 11.6 | 0.2 | 2.3 | 20.1 | 0.1 | 1.0 | 13.2 | 0.1 | 1.1 | 11.5 | 0.2 | 1.2 | 9.1 |
| TorchGeo [32] (SSL w SeCo [39]) | 6.1 | 15.6 | 41.7 | 7.4 | 20.2 | 49.2 | 5.1 | 14.5 | 37.1 | 3.7 | 14.0 | 38.9 | 4.6 | 13.3 | 32.9 | 5.7 | 15.6 | 38.5 |
| TorchGeo [32] (SSL w GASSL [4]) | 9.7 | 22.8 | 46.4 | 9.1 | 23.1 | 50.5 | 13.3 | 31.4 | 58.8 | 6.3 | 17.5 | 45.4 | 8.3 | 20.3 | 40.1 | 20.4 | 38.6 | 64.2 |
| OGCL UAV-View [11] [12] | 17.6 | 33.2 | 55.9 | 14.6 | 33.2 | 63.9 | 22.8 | 48.1 | 74.9 | 7.6 | 22.8 | 50.1 | 20.4 | 39.1 | 62.5 | 31.8 | 51.8 | 74.4 |
| MBEG [41] | 7.0 | 17.6 | 35.1 | 6.6 | 19.3 | 45.9 | 8.7 | 20.7 | 41.5 | 4.4 | 15.0 | 38.0 | 6.4 | 17.1 | 39.3 | 8.1 | 20.7 | 49.1 |
| AnyLoc [19] | 44.1 | 68.7 | 87.8 | 40.7 | 70.8 | 92.0 | 48.7 | 75.0 | 91.6 | 28.7 | 57.0 | 81.7 | 38.6 | 63.8 | 86.2 | 63.7 | 84.5 | 96.3 |
| EarthLoc [6] | 55.9 | 73.0 | 88.3 | 58.4 | 76.8 | 89.5 | 58.0 | 76.0 | 91.4 | 51.1 | 67.5 | 86.5 | 47.2 | 67.9 | 84.6 | 72.2 | 85.0 | 93.3 |
| EarthLoc++ [6] | 80.0 | 89.4 | 95.9 | 80.6 | 91.2 | 96.7 | 82.9 | 92.0 | 97.9 | 67.6 | 84.6 | 94.6 | 73.6 | 85.5 | 93.1 | 90.1 | 95.3 | 98.2 |
| AstroLoc | **96.1** | **98.7** | **99.7** | **98.1** | **99.5** | **99.8** | **97.4** | **99.2** | **99.8** | **94.6** | **99.2** | **99.9** | **93.0** | **96.9** | **99.1** | **99.0** | **99.6** | **99.9** |

Table 2. **Recalls on APL test sets from EarthLoc [6],** from which we source most of the numbers in the table. AstroLoc comfortably and consistently achieves recall over 90%, outperforming all other methods by a large margin. Numbers next to each test set name (*e.g*. Texas 6k / 34k) represent number of queries / database images. R@N indicates the recall-at-N. Best results **bold**, second best underlined.

| Method | Features dimension | Memory (GB) | Total latency (hh:mm) | Per query latency (ms) | Texas-L (21k / 179k) | | Alps-L (10k / 261k) | | California-L (12k / 166k) | | Gobi Desert-L (2k / 284k) | | Amazon-L (3k / 101k) | | Toshka Lakes-L (9k / 299k) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | R@1 | @100 | R@1 | @100 | R@1 | @100 | R@1 | @100 | R@1 | @100 | R@1 | @100 |
| AnyLoc | 49152 | 235 GB | 23:51 | 291 | 29.2 | 72.2 | 30.9 | 78.9 | 31.7 | 76.3 | 22.9 | 68.2 | 20.4 | 63.5 | 34.9 | 76.8 |
| EarthLoc | 4096 | 19 GB | 1:12 | 21 | 44.0 | 75.0 | 52.9 | 82.6 | 49.1 | 80.4 | 33.9 | 70.1 | 34.9 | 70.6 | 63.3 | 86.5 |
| EarthLoc++ | 2048 | 9 GB | 1:42 | 12 | 72.4 | 91.5 | 79.4 | 94.7 | 75.4 | 92.9 | 62.6 | 87.2 | 62.6 | 89.2 | 83.4 | 95.1 |
| AstroLoc | 2048 | 9 GB | 1:42 | 12 | **91.1** | **98.5** | **94.6** | **99.2** | **92.1** | **98.7** | **84.2** | **96.2** | **83.8** | **96.8** | **94.8** | **99.0** |

Table 3. **Results on our extended versions of the test sets from Tab. 2.** "-L" (for "large") test sets better represent the performance that an APL model would achieve if deployed, as no filtering has been applied on test queries. The two numbers (*e.g*. Texas 21k / 179k) represent the number of queries and database images. *Memory*, *Total latency* and *Per query latency* refer to testing on the Texas dataset. *Memory* is the memory required to store features; *Total latency* is the time to run the test (including database feature extraction); and *Per query latency* is the time needed to localize one query, which consists of query feature extraction and kNN search (kNN is the primary bottleneck). All tests are performed with an A100 and a 32-core CPU, using FAISS [13] for efficient kNN.

| Method | # Params (M) | R@1 | R@5 | R@10 | R@20 | R@100 |
|---|---|---|---|---|---|---|
| AnyLoc | 1136 | 4.4 | 9.6 | 13.4 | 19.0 | 40.0 |
| EarthLoc | 27.6 | 3.3 | 7.0 | 8.8 | 12.0 | 27.6 |
| AstroLoc | 105 | **52.7** | **63.9** | **68.7** | **73.3** | **83.7** |
| AstroLoc-tiny | **27.2** | 36.7 | 49.4 | 55.3 | 61.6 | 74.5 |

Table 4. **Results of Lost-in-Space satellite problem.** Using the dataset from VINSat [22] as queries, AstroLoc achieves good results on this additional task. AstroLoc-tiny is a tiny version more suitable for deployment on a nanosatellite, with 27M model parameters and 512 dimensional features (see supplementary).

| Method | R@1 | R@5 | R@10 | R@20 | R@100 |
|---|---|---|---|---|---|
| AnyLoc | 19.6 | 32.7 | 40.5 | 47.2 | 65.2 |
| EarthLoc | 29.5 | 40.1 | 43.5 | 48.7 | 64.5 |
| AstroLoc | **82.0** | **89.9** | **91.9** | **94.2** | **96.7** |

Table 5. **Results on historical imagery localization.** The 704 queries are 40 year-old photos from the early days of the Space Shuttle. Searched is conducted across the whole globe.

| Pair Loss (Sec 4.2) | Sol. 1 (Sec 4.3) | Sol. 2 (Sec 4.3) | Sol. 3 (Sec 4.3) | Texas-L | | | Alps-L | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | R@1 | R@10 | R@100 | R@1 | R@10 | R@100 |
| ✓ | | | | 83.6 | 93.1 | 97.5 | 87.2 | 95.0 | 98.2 |
| | ✓ | | | 67.6 | 79.2 | 89.3 | 76.5 | 87.1 | 93.9 |
| | | ✓ | | 72.4 | 83.3 | 91.5 | 79.4 | 89.1 | 94.7 |
| | | | ✓ | 82.2 | 91.1 | 97.1 | 86.9 | 94.5 | 97.9 |
| ✓ | ✓ | | | 84.0 | 93.3 | 97.5 | 87.9 | 95.1 | 98.0 |
| ✓ | | ✓ | | 86.5 | 94.1 | 97.7 | 91.5 | 96.5 | 98.6 |
| ✓ | | | ✓ | **91.1** | **96.2** | **98.5** | **94.6** | **97.8** | **99.2** |

Table 6. **Ablation over components and variation of the loss.**

the timestamp can be unreliable, leaving no information about which area of the Earth the photo is taken near.

We therefore propose the task of worldwide astronaut

| Method | R@1 | R@5 | R@10 | R@20 | R@100 |
|---|---|---|---|---|---|
| EarthLoc | 40.3 | 49.5 | 53.2 | 57.1 | 66.6 |
| AstroLoc | 86.9 | 91.9 | 93.4 | 94.6 | 96.8 |

Table 7. **World-wide search.** Using the same query sets as Tab. 3, but with a database encompassing the entire Earth. Results computed for the best performing methods only, except AnyLoc [19], due to its large features (49,152-D vs EarthLoc's 4096-D) requiring 600GB of RAM to store features to compute kNN.

photography localization, where the database covers the entire world with 881k images. Results (Tab. 7) show that AstroLoc achieves a recall@100 of 96.8%, proving its robustness even in this highly challenging scenario.

# 6. Conclusions

We tackle the task of Astronaut Photography Localization (APL), and through newly computed labels and ad hoc training techniques, obtain AstroLoc, which shows impressive results on all APL datasets. Furthermore, we find that AstroLoc can be utilized for other real-world applications, showing robustness to various domains. Finally, we note that we have already used AstroLoc (in conjunction with EarthMatch's post-processing) to provide localization for a staggering 500k photos, and anticipate that in a few months the backlog of non-localized photos will be nearly empty for the first time since the launch of the ISS.

**Acknowledgements.** We acknowledge the Cineca award under the Iscra initiative, for the availability of high performance computing resources. This work was supported by CINI. Project supported by ESA Network of Resources Initiative. This study was carried out within the project FAIR - Future Artificial Intelligence Research - and received funding from the European Union Next-GenerationEU (Piano nazionale di ripresa e resilienza (PNRR) – missione 4 componente 2, investimento 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. European Lighthouse on Secure and Safe AI – ELSA, Horizon EU Grant ID: 101070617

## References

[1] Amar Ali-Bey, Brahim Chaib-draa, and Philippe Giguere. Global proxy-based hard mining for visual place recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 7

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 6

[3] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018. 7

[4] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. *IEEE International Conference on Computer Vision*, 2021. 8

[5] 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) Workshops*, 2024. 2, 4, 7

[6] Gabriele Berton, Alex Stoken, Barbara Caputo, and Carlo Masone. Earthloc: Astronaut photography localization by indexing earth from space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3, 4, 6, 7, 8

[7] Mollie Bianchi and Timothy D. Barfoot. Uav localization using autoencoded satellite images. *arXiv preprint arXiv:2102.05692*, 2021. 3

[8] Ming Dai, Enhui Zheng, Zhenhua Feng, Jiahao Chen, and Wankou Yang. Finding point with image: A simple and efficient method for uav self-localization. *arXiv preprint arXiv:2208.06561*, 2022. 3

[9] Alejandro Sánchez de Miguel, Jonathan Bennie, Emma Rosenfeld, Simon Dzurjak, and Kevin J. Gaston. Environmental risks from artificial nighttime lighting widespread and increasing across europe. *Science Advances*, 8(37): eabl6891, 2022. 1, 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 4

[11] Fabian Deuser, Konrad Habel, Martin Werner, and Norbert Oswald. Orientation-guided contrastive learning for uav-view geo-localisation. In *Proceedings of the 2023 Workshop on UAVs in Multimedia: Capturing the World from a New Perspective*, page 7–11, New York, NY, USA, 2023. Association for Computing Machinery. 3, 8

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv*, abs/2010.11929, 2021. 8

[13] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv*, 2024. 8

[14] Kevin J. Gaston and Alejandro Sánchez de Miguel. Environmental impacts of artificial light at night. *Annual Review of Environment and Resources*, 47(1):373–398, 2022. 1, 3

[15] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Computer Vision – ECCV 2020*, pages 369–386, Cham, 2020. Springer International Publishing. 7

[16] Yao He, Ivan Cisneros, Nikhil Keetha, Jay Patrikar, Zelin Ye, Ian Higgins, Yaoyu Hu, Parv Kapoor, and Sebastian Scherer. Foundloc: Vision-based onboard aerial localization in the wild. *arXiv preprint arXiv:2310.16299*, 2023. 3

[17] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 7

[18] Augustin Jehl, Thomas Farges, and Elisabeth Blanc. Color pictures of sprites from non-dedicated observation on board the international space station. *Journal of Geophysical Research: Space Physics*, 118(1):454–461, 2013. 1, 3

[19] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *arXiv*, 2023. 3, 7, 8

[20] Shishen Li, Cuiwei Liu, Huaijun Qiu, and Zhaokui Li. A transformer-based adaptive semantic aggregation method for uav visual geo-localization. *arXiv preprint arXiv:2401.01574*, 2024. 3

[21] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 4

[22] 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. 3, 7, 8

[23] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision – ECCV 2020*, pages 681–699, Cham, 2020. Springer International Publishing. 6

[24] Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. Pytorch metric learning. *ArXiv*, abs/2008.09164, 2020. 7

[25] Aadil Nathani, Rishi Iyer, Annabelle Wang, Aarnav Chitari, Adele Wilson, and Hannah Norris. Observing Earth From Space: Using Astronaut Photography to Analyze Geographical Climate Patterns. In *AGU Fall Meeting Abstracts*, pages ED44C–06, 2022. 1, 3

[26] OpenStreetMap Contributors. Zoom levels. `https://wiki.openstreetmap.org/wiki/Zoom_levels`, 2023. 3

[27] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 7

[28] Aaron Schirmer, Caleb Gallemore, Ting Liu, Seth Magle, Elisabeth DiNello, Humerah Ahmed, and Thomas Gilday. Mapping behaviorally relevant light pollution levels to improve urban habitat planning. *Scientific Reports*, 9, 2019. 1, 3

[29] Liberty M. Shockley and Robert A. Bettinger. Real-time aerospace vehicle position estimation using terrestrial illumination matching. In *IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace),*, 2021. 3

[30] Christopher Small. Spectrometry of the urban lightscape. *Technologies*, 10(4), 2022. 1, 3

[31] W. L. Stefanov and C. A. Evans. Data collection for disaster response from the international space station. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-7/W3:851–855, 2015. 1, 3

[32] Adam J. Stewart, Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, and Arindam Banerjee. TorchGeo: Deep learning with geospatial data. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–12, Seattle, Washington, 2022. Association for Computing Machinery. 8

[33] Alex Stoken and Kenton Fisher. Find my astronaut photo: Automated localization and georectification of astronaut photography. In *CVPRW*, pages 6196–6205, 2023. 2, 3, 7

[34] Alex Stoken, Peter Ilhardt, Mark Lambert, and Kenton Fisher. (street) lights will guide you: Georeferencing nighttime astronaut photography of earth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 492–501, 2024. 2, 3

[35] Miranda Straub and John A. Christian. Autonomous optical navigation for earth-observing satellites using coastline matching. In *AIAA Guidance, Navigation, and Control Conference*, 2015. 3

[36] Gabriele Trivigno, Carlo Masone, Barbara Caputo, and Torsten Sattler. The unreasonable effectiveness of pretrained features for camera pose refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12798, 2024. 7

[37] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 6

[38] Yoav Yair, Melody Korman, Colin Price, and Eytan Stibbe. Observing lightning and transient luminous events from the international space station during ilan-es: An astronaut's perspective. *Acta Astronautica*, 211:592–599, 2023. 1, 3

[39] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, 2021. 8

[40] Zhedong Zheng, Yujiao Shi, Tingyu Wang, Jun Liu, Jianwu Fang, Yunchao Wei, and Tat-seng Chua. Uavm '23: 2023 workshop on uavs in multimedia: Capturing the world from a new perspective. In *ACM MM*, page 9715–9717, New York, NY, USA, 2023. Association for Computing Machinery. 3

[41] Runzhe Zhu, Mingze Yang, Kaiyu Zhang, Fei Wu, Ling Yin, and Yujin Zhang. Modern backbone for efficient geo-localization. In *Proceedings of the 2023 Workshop on UAVs in Multimedia: Capturing the World from a New Perspective*, page 31–37, New York, NY, USA, 2023. Association for Computing Machinery. 3, 8