

Earthloc: Astronaut photography localization by indexing earth from space

*Original*

Earthloc: Astronaut photography localization by indexing earth from space / Berton, G., Stoken, A., Caputo, B., Masone, C.. - (2024), pp. 12754-12764. (IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) Seattle (USA) 16-22 June 2024) [10.1109/CVPR52733.2024.01212].

*Availability:*

This version is available at: 11583/2989655 since: 2024-06-18T14:09:31Z

*Publisher:*

IEEE

*Published*

DOI:10.1109/CVPR52733.2024.01212

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

# EarthLoc: Astronaut Photography Localization by Indexing Earth from Space

Gabriele Berton<sup>1</sup> Alex Stoken<sup>2</sup> Barbara Caputo<sup>1</sup> Carlo Masone<sup>1</sup>  
<sup>1</sup>Politecnico di Torino <sup>2</sup>Jacobs Technology, NASA Johnson Space Center  
 gabriele.berton@polito.it

## Abstract

Astronaut photography, spanning six decades of human spaceflight, presents a unique Earth observations dataset with immense value for both scientific research and disaster response. Despite their significance, accurately localizing the geographical extent of these images, which is crucial for effective utilization, poses substantial challenges. Current, manual localization efforts are time-consuming, motivating the need for automated solutions. We propose a novel approach – leveraging image retrieval – to address this challenge efficiently. We introduce innovative training techniques which contribute to the development of a high-performance model, EarthLoc. We develop six evaluation datasets and perform a comprehensive benchmark comparing EarthLoc to existing methods, showcasing its superior efficiency and accuracy. Our approach marks a significant advancement in automating the localization of astronaut photography, which will help bridge a critical gap in Earth observations data. Code and datasets are available at <https://github.com/gmberton/EarthLoc>.

## 1. Introduction

Astronaut photography of Earth is a unique remote sensing dataset that spans 60 years of human spaceflight, offering a rare perspective on our planet to the public and valuable data to Earth and atmospheric science researchers. This dataset contains over 4.5 million images and is growing by the tens of thousands per month, as astronauts are continually tasked with taking new photographs that enable scientific research as well as assist in natural disaster response efforts in the wake of events like floods and wildfires. To effectively use these images, the geographical area depicted in them needs to be identified. Unfortunately, this task - Astronaut Photography Localization (APL) - is very challenging. For each photo, only a coarse estimate of location is known, given by the point on Earth directly under the International Space Station (ISS) at the time the photo is taken. This point – called the **nadir** – can be easily computed using the image’s timestamp and the ISS’s orbit path. However, two

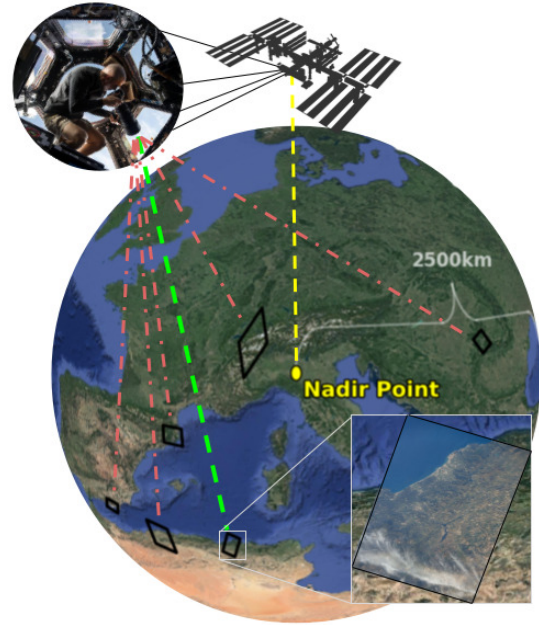


Figure 1. **Overview of the astronaut photography localization task.** Astronauts take hundreds of photos a day from the International Space Station (ISS) cupola (top-left) with hand-held cameras. For each image (example bottom right), the geographic location depicted is not known, and needs to be searched for across a huge area centered at the ISS’s (known) *nadir* point at the moment that the photo is taken. A simulated view of the astronaut’s perspective when the ISS is above Europe is shown. The goal of our paper is to automate the task of localizing these images, which could be anywhere within the view. In the figure’s example, the photo the astronaut took is indicated by the green line and shown in inset – other possible photo extents are in red, illustrating the wide array of potential locations to search.

images taken with the same nadir can be thousands of kilometers apart, as even a slight inclination of the astronaut’s hand-held camera can move the image’s location hundreds of kilometers in any direction, as depicted in Fig. 1. Localization must thus be performed over a wide area, and is additionally complicated by (i) astronauts using hand-held cameras and a variety of zoom lenses, (ii) the large, 2500

kilometer (km) visibility range in all directions, (iii) most photographs being oblique, and (iv) the Earth’s appearance changing over time, mostly due to weather or seasonal variations. Despite these challenges, the high value of properly localized imagery for Earth science research and disaster response has led to more than 300,000 [21] astronaut photos being manually localized by identifying the geographic coordinates of the photo’s center pixel. This process can take minutes to hours per photo for experts in NASA’s Earth Science and Remote Sensing Unit (ESRS) and citizen scientists through ESRS’s Image Detective<sup>1</sup> program. The resulting geo-located images have led to a multitude of peer reviewed papers, technical reports, and articles<sup>2</sup> about light pollution and urban planning[15, 22, 47, 53], atmospheric phenomena [28, 69], changes in land usage and glacial extent [40], and other Earth science topics [27, 48, 68]. Most importantly, astronaut photography has a fast response time, crucial for disaster management - astronauts are alerted of a disaster, provide photos to ground crew which localizes them, and send them to first responders<sup>3</sup> [56]. For example, in 2013 this protocol was activated in response to Cyclone Haiyan in the Philippines, wildfires in Australia, flooding in China, Russia, Pakistan, and the USA, and multiple other events. [56].

Even with the abundance and importance of satellite imagery in modern applications, astronaut photography fills an unserved gap among other remotely sensed data. Unlike satellites that nominally take top-down imagery in similar illumination conditions at fixed temporal intervals, astronaut photographs can show topography via oblique views from multiple orientations, are taken in various lighting conditions (including nighttime), and vary in focal length to show detail at different resolutions. These qualities produce a complementary data product that would be difficult or impossible to gather from traditional satellites. Additionally, having a human in the loop for data collection allows for real-time response to natural disasters as well as a natural sense to avoid clouds and other obstructions.

Given the uniqueness and importance of astronaut photography, and the large amount of time spent by human experts on geo-localizing them, researchers have been exploring solutions to automate the task through computer vision methods. Previous works have shown promising results using a pairwise matching setup that iteratively compares an astronaut photo to satellite imagery [58]. Yet, such methods suffer from high latency, searching multiple directions sequentially and using compute-heavy, dense correspondences to determine matching. The time required to find the location of all 4.5 million images using these meth-

ods is estimated to be over 20 compute years [58]. Furthermore, low latency is of paramount importance for photography in areas that are affected by natural disasters (*e.g.*, hurricanes or wildfires), for which speedy localization can help real-time, on-the-ground operations.

To overcome the high latency of current methods, we propose to instead localize astronaut photography through image retrieval, by matching each astronaut photograph to a worldwide database containing satellite images of known position. In reformulating this problem as a retrieval task, we come across a range of new challenges, from efficiently training a robust model to achieving low-latency inference, as well as defining a success metric and creating evaluation sets, both of which are crucial pieces when assessing how well a given model will perform when deployed to localize astronaut photography.

The ultimate goal of this paper is twofold: (i) provide an efficient method to localize the archive of 4.5M queries as well as all new, incoming imagery, so that researchers studying a given geographical area can have access to a large amount of imagery spanning many years; and (ii) prove the viability of world-wide APL through image retrieval (which has *never been attempted* and was *believed to be infeasible*<sup>4</sup>) in order to spark a new line of research, with direct benefit to all space-based photography and its users across the globe.

To achieve its goal, this paper makes the following contributions:

- we propose approaching APL through image retrieval;
- we develop novel training techniques which show large quantitative improvement on the task;
- we provide six evaluation sets and a large benchmark of results with methods from a variety of relevant domains;
- we show that a model trained with these new techniques, **EarthLoc**, allows us to localize a large number of images, widely outperforming all other methods while being fast and efficient.

Finally, we note how EarthLoc is being used to localize ISS astronaut photographs, which are publicly available and conveniently searchable at <https://eol.jsc.nasa.gov/ExplorePhotos/>.

## 2. Related Work

**Image Retrieval** Image retrieval involves searching a database for images similar to a query image. Traditionally, methodologies combined hand-crafted local features like SIFT [37], SURF [8], RootSIFT [3] into a global embedding through means like Bag of Words [13], Fisher Vectors [42] or VLAD [29], to allow for fast retrieval through

<sup>4</sup>in [59] (Sec. 2.1) scientists at NASA claim that retrieval for APL is infeasible because *we cannot precompute a database of reference features* due to their belief that the database must be query-specific. Results show that our pipeline can overcome this issue.

<sup>1</sup><https://eol.jsc.nasa.gov/BeyondThePhotography/ImageDetective/>

<sup>2</sup><https://eol.jsc.nasa.gov/AboutCEO/PubList.htm>

<sup>3</sup><https://storymaps.arcgis.com/stories/947eb734e811465cb0425947b16b62b3>

kNN. With the rise of deep learning, local features have been replaced by CNN-derived features [6], significantly improving retrieval performance. Since then, most of the attention of the retrieval community has been focused on how to best aggregate feature maps and efficiently train robust neural networks. For feature map aggregation, some of the most notable examples in literature include Max pooling [44], Regional Max pooling (R-MAC) [61] and Generalized Mean (GeM) [43]. To train these retrieval models, a number of losses have been proposed, which can be generally grouped into two categories: stateful losses, which have weights whose size depends on the number of classes, like the Large-Margin Softmax [35], SphereFace [36], CosFace [65] and ArcFace [16]; and their stateless counterparts, like the Contrastive and Triplet losses, and the more recent Lifted Structure Loss [54], NTXentLoss [64], Multi-Similarity Loss [67], FastAP [11], Supervised Contrastive [31] and the Circle Loss [60].

**Geo-localization and retrieval with Aerial and Remote Sensing Imagery** To the best of our knowledge, only two previous works have specifically addressed the localization problem for astronaut photography. Both rely on local features and compute pairwise comparisons between astronaut photography and geo-located reference imagery. Find My Astronaut Photo [58] focuses on daytime imagery, presenting an evaluation of multiple methods as well as the note that pre-trained self-supervised models were not suitable for a retrieval-based approach to this task. Schwind and Storch [49] instead address nighttime imagery, using synthetically generated street light maps as reference. Both works discuss the importance of proper orientation for matching and the high cost of pairwise comparisons.

More commonly, studies of aerial imagery focus on data from unmanned aerial vehicles (UAVs) and satellite platforms. Aerial geo-localization is an important aspect of UAV navigation systems. The University-1652 dataset [71] and associated challenge encourage work in cross-view geo-localization for drone-satellite imagery, with strong approaches modifying different parts of the general pipeline, from backbone [73] to training setup [18], to feature partitioning [66]. Most use contrastive and self-supervised losses in training. Other works address cross-view geo-localization in more extreme view point differences, like that between aerial and street views [17, 20, 50–52, 74, 75].

Remotely sensed data from satellites, on the other hand, often comes with reliable geo-location information from the sensor itself. Extracting features from satellite imagery is still a common task, with such features most often trained on a pretext objective before use on downstream tasks [5, 12, 39, 45]. Occasionally, features are used for retrieval itself [55], to identify areas of the Earth that are similar in appearance and thus might have similar proper-



Figure 2. **Astronaut photo query examples**, showcasing the large variability in covered area and appearance.

ties to be studied together. In this paper we take a separate direction from previous literature, and aim at using image retrieval techniques for APL.

**Other localization tasks** Various other localization challenges are approached through image retrieval, including visual place recognition (VPR) and visual localization. The former aims at coarsely localizing a given query by matching it to a database of geo-tagged photos, and is mostly studied within urban environments using street-view imagery [7, 63]. Performance is improved either by the use of smart aggregation (e.g. NetVLAD [4]), attention layers (e.g. CRN [32]), or through large-scale training [1, 2, 9, 34]. More recently, the task of universal visual place recognition has been proposed by AnyLoc [30], whose authors provide a model that performs competitively on a large range of scenarios, including aerial imagery. The task of visual localization is focused on finding the precise camera pose of an image, and is commonly approached through feature matching techniques [7, 38, 46, 59]. It can also be used for visual place recognition [24, 76], although this leads to a noticeable increase in runtime when compared to pure retrieval.

### 3. Dataset

In this section we discuss the datasets that we use in this work. Specifically, in Sec. 3.1 we introduce the images that we need to localize, called queries; in Sec. 3.2 we outline the collection of the database, made of satellite images (*i.e.*, taken automatically, not with a hand-held camera) of known location; and in Sec. 3.3 we describe the creation of the evaluation datasets that we use to understand the capabilities of our models, combining queries and database in a way that will reflect the real-world use case of our method.

#### 3.1. Queries

Queries are taken from the Gateway to Astronaut Photography of Earth<sup>5</sup>, a collection of over 4.5 million photos of Earth taken by astronauts on the International Space Station (Fig. 2). This unique photography setting yields only coarse

<sup>5</sup><https://eol.jsc.nasa.gov/>

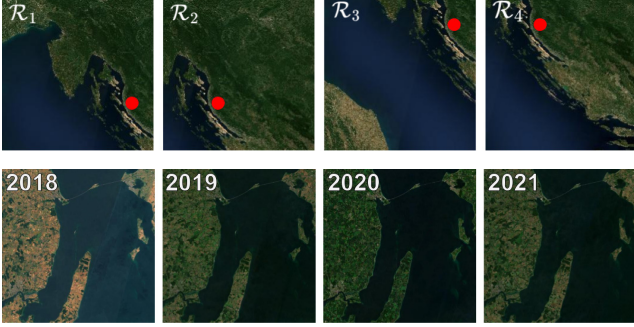


Figure 3. **Examples of database images.** **Top:** four images from different *regions*, with 25% or 50% overlap between any pair. The red dot in each image represents the same geographic point. **Bottom:** 4 images from the same *region* across different years.

location information for each photograph, namely the position on Earth below the ISS (the **nadir** point) at the time the photo was taken. We seek to find the geographic area on the Earth that each photo encompasses (the photo’s **location**), which can be thousands of kilometers away from the ISS nadir point. This distance of maximum visibility (horizon) can be quickly computed as

$$d_{\text{visible}} = \sqrt{2Rh + h^2} = 2436 \text{ km} \quad (1)$$

where  $R = 6317$  is Earth’s radius and  $h = 450$  is the ISS maximum orbiting altitude. We follow common practice and round this number to 2500 kilometers [58].

**Challenges to APL** Some astronaut photo queries, taken with high focal length lenses, can cover only a few kilometers within the almost 20 million sq. km area visible to the astronaut taking the photo, which makes localizing the images extremely challenging. Additional hurdles presented by the imagery acquisition process include the varying quality of the images themselves due to motion blur from the fast moving space station, and occlusion of unique, location-identifying features by cloud cover, shadow, or other portions of the ISS (Fig. 2). More queries are shown in Sec. 10 of the Supplementary.

From the 4.5 million photos available, we consider only the day-time images that have a ground truth obtained by FMAP [58], and keep only those which cover areas between 5000 and 900,000 sq. km, in order to match the areas of database images (see Sec. 3.2). This leads to 17,764 queries which we can use for validation and testing.

### 3.2. Database

To estimate the location of the queries through image retrieval, we need a worldwide database of images, where each image is labeled with its position. We build such a database from an open-source, composited, cloudless col-

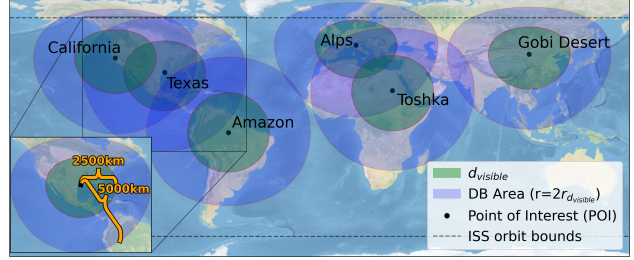


Figure 4. **To create evaluation sets** we choose all images that could contain a Point of Interest (POI) - all photos with nadir point within  $d_{\text{visible}}$ . To localize all photos within this range, even if they do not contain the POI, we create a database that contains all areas visible from the nadir points of all selected photos, yielding a database area of about 5000km<sup>2</sup> per POI.

lection of Sentinel-2 satellite imagery<sup>6</sup>. This collection contains imagery of worldwide landmasses at resolutions up to 15 meters per pixel. Some examples are shown in Fig. 3.

For our database we include any land area between latitudes 60° and -60° (*i.e.*, the area traversed by the ISS). The Sentinel-2 imagery is available as map tiles, and we use tiles at zoom levels 9, 10, 11, with resolutions ranging from ~300 to ~75 meters per pixel<sup>7</sup>, to ensure that the database encompasses the large scale range of the query images.

These parameters lead us to define roughly 175k **regions**, where each *region*  $\mathcal{R}$  is the projection of a square with known corner coordinates onto the Earth’s surface. *Regions* have areas ranging from 5000 sq. km to 900,000 sq. km. We take partially (up to 50%) overlapping *regions* so that each query has at least one image in the database with considerable overlap (Fig. 3 top). From each *region* we collect four images, one per year from 2018, 2019, 2020 and 2021, covering temporal changes in a given area (Fig. 3, bottom). This produces a database of 700k images.

Unlike our queries, these database images have undergone post-processing that includes atmospheric corrections as well as cloud pixel minimization via composition of images from multiple days. Additionally, the database images are always taken nadir facing (*i.e.* perpendicular to Earth’s surface) to minimize obliquity effects. Each database image has resolution 1024×1024. Further images from the database are shown in Sec. 10 of the Supplementary.

### 3.3. Evaluation sets

With the end goal of fast and accurate localization of the astronaut photography archive, we propose a deployment environment that takes advantage of the known nadir points and visibility limits associated with each image. So while localization can be performed by matching any given query

<sup>6</sup><https://s2maps.eu>

<sup>7</sup>[https://wiki.openstreetmap.org/wiki/Zoom\\_levels](https://wiki.openstreetmap.org/wiki/Zoom_levels)

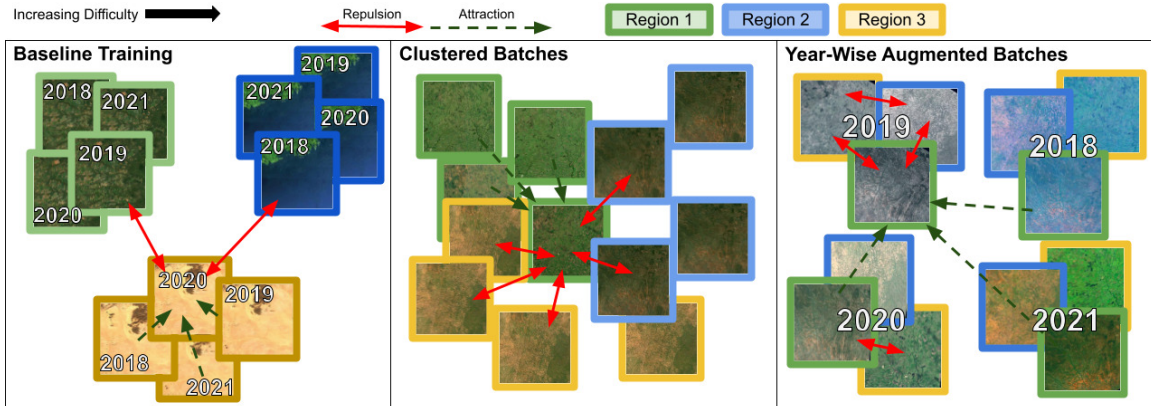


Figure 5. **Training strategy.** Naive batching produces too easy a task for the model to learn robust representations from. We increase difficulty by clustering batches by similarity and adding year-wise augmentations, which move images of different *regions*, but from the same year, closer together in feature space. The model must then learn representations that discern similarity in this more difficult context.

Name	(Lat, Lon)	# queries	# DB	Interest
Texas	(30, -95)	6142	34k	Often photoed (many queries)
Alps	(45, 10)	2394	53k	Sci. interest - glacial change
California	(38, -122)	3568	30k	Disaster response - wildfires
Gobi Desert	(40, 105)	726	54k	Sci. interest - desertification
Amazon	(-3, -60)	682	19k	Sci. interest - deforestation
Toshka Lakes	(23, 30)	2164	63k	Sci. interest - flood monitoring

Table 1. **Evaluation sets.** The six sets cover diverse geographies and relevant areas for research on topics like climate change and disaster management. DB stands for database.

against the whole world-wide database, a more suitable approach is to follow a divide and conquer paradigm by grouping queries according to their nadir, which allows us to use only a subset of the database for each search, increasing speed and improving results.

We seek to create evaluation sets that mirror the situation seen in deployment, so we evaluate on sets of queries that have a nadir within a 2500 km radius from a chosen Point of Interest (POI). We then build a database of images such that the database fully encompasses the area that could have been photographed from each collected query, which is an area  $2 * d_{visible}$  from the original POI, as shown in Fig. 4.

Since we need the queries’ ground truth position for evaluation, we choose queries that have already been localized with exhaustive image matching techniques [58]. For these queries, the full photo extent is publicly available<sup>8</sup>.

To properly evaluate different methods, we choose six POIs that represent various geographies but all have scientific relevance, and center an evaluation set around each. These are summarized in Tab. 1.

## 4. Method

The goal of this work is to estimate the location of each astronaut photo query, which we tackle through image re-

trieval: for any given query, the objective is to find the most similar image(s) from the database, and use the corresponding location labels to confidently localize the query. With this in mind, we aim to train a retrieval model on satellite imagery such that the model is robust to changes in scale, perspective, and color, as well as temporal/seasonal variation. Specifically, the model’s task is to extract features from each image. Then, for each query, the most similar database image is retrieved via kNN in feature space.

In the next subsections we describe how we train our retrieval model using the proposed database.

### 4.1. Baseline Training

We want our model to extract relevant features of satellite imagery, so a straightforward approach would be to train via contrastive learning: for a given batch of images (*samples*), generate a number of augmented views (typically two), and apply a contrastive loss to maximize the distance in feature space between different samples and minimize distance between different views of the same sample. However, this solution would not allow the model to learn robustness to temporal changes, because the augmented views come from the same sample, with no inherent temporal variation.

Therefore, instead of producing multiple views of the same image via synthetic augmentation, we train the model by randomly selecting a number of *regions*, and for each *region* we consider its four images (a **quadruplet**), one per year. Within a batch of quadruplets, the four images from each *region* are mutual positives, whereas any two images from different *regions* are considered negatives: we can therefore consider each *region* as a class. This provides a natural way to address temporal variation. We can then train the model through a multi-similarity loss [67]. This idea is shown in Fig. 5 (left).

<sup>8</sup><https://eol.jsc.nasa.gov/SearchPhotos/PhotosDatabaseAPI/> (mlcoord Table)

## 4.2. Clustered Batches

The baseline paradigm simply creates batches of random *regions* – a single batch of quadruplets could contain samples depicting shorelines, islands, deserts, and mountains. The loss operates batchwise, so large intra-batch variability can lead the model to learn too swiftly to distinguish between such diverse *regions*. In these situations, training converges quickly and the model stops short of learning to discriminate between similar-looking, but distinct, *regions*.

To avoid this issue, we want to train with batches of highly similar images, which are still from different *regions*. To this end, we compute clusters of images with similar representations. Every  $N_{iter}$  training iterations, we extract the features from each *region* with the trained model, using only images from a single year, and group them into  $C$  clusters.

Then, we create training batches using samples from a given cluster, choosing a different cluster for each batch. This increases the difficulty of the task in each batch, ensuring that the model learns to discriminate between similar-looking images of different *regions*. Note that clusters are not geographically related, but related in feature space. For example, a cluster may contain images from the Gobi and the Sahara deserts, which both appear similar in feature space despite being geographically distant. This procedure, which can be seen as an offline mining approach, increases training time by less than 5% but improves recall by roughly 15% on average (see Sec. 5.3), a worthwhile trade off.

This concept is illustrated in Fig. 5 (middle), and qualitative examples of images that are clustered together are shown in Sec. 11.2 of the Supplementary.

## 4.3. Year-Wise Data Augmentation

Query images span the ongoing 20+ years of ISS operations, as well as multiple camera types, camera operators, and view points, so queries depicting the same location can have a number of natural “transformations”. These range from blueish hues from lens filters, to skewed representations in highly oblique shots, to seasonal variation in ground cover (e.g. snow, autumn colors).

To make the model robust to such transformations, we employ data augmentation. Typically, augmentations are applied per image or per batch, but we argue that to truly learn robustness to the augmentations in this setting, the same augmentation should be performed on images from different *regions*, but from the same year.

Formally, at each training iteration we choose a set of 4 random augmentations, one for each year  $A_{2018}, A_{2019}, A_{2020}, A_{2021}$ . We then apply the corresponding augmentation to all images from that year (i.e.  $A_{2018}$  is applied to all images from 2018, across all *regions*). Such “year-wise” augmentation has the effect of moving images from a given year closer in feature space to each other, as shown in Fig. 5 (right). In the loss, images from the same

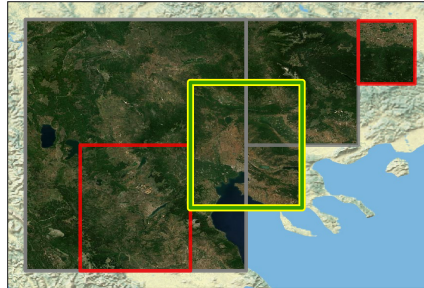


Figure 6. **Dataset collection from satellite imagery and handling in Neutral-Aware Multi-Similarity Loss.** For a given image (yellow), images with no overlap are negatives (red), images from the same *region* are positives (green, overlapping yellow), and images with partial overlap are neutral (grey). Note that the database is made of images at 3 different zooms (each double in size than the next), which explains the *regions* of different size.

*region*, but different years, are treated as positives, while images from different *regions* are treated as negatives. This has the effect that images that receive the same augmentation are assigned different classes – forcing the model to learn to ignore such augmentations – and instead pull together images from the same *region*, across different years.

Furthermore, given that the same augmentation is applied to multiple images, augmentation can be batched and performed on the GPU, making training slightly faster (by 5% on average, although this depends on the hardware).

## 4.4. Neutral-Aware Multi-Similarity Loss

The powerful Multi-Similarity loss is designed to take as input multiple images from a number of classes, along with the corresponding class indices, and compute a loss from the positives and negatives for each image.

However, it does not take into account that in some cases, pairs of images are neither positives nor negatives, and should be considered “neutral”. While the concept of neutral images does not exist in standard image retrieval tasks, as an image either belongs to class A or class B, in APL it must be taken into consideration. For example, should two images, which have an intersection over union of 25%, be considered positives or negatives? This is precisely the case when two images of the same area, but from different zoom levels, are batched together, as well as overlapping images at the same level (see Fig. 6). We believe the answer is neither, so we add the “neutral” case to the loss.

Formally, we model the neutral case with the indicator function  $\mathcal{I}_{NA}$ , which is 0 when the *regions*  $\mathcal{R}_i$  and  $\mathcal{R}_j$  intersect but are not equal, and 1 otherwise.

$$\mathcal{I}_{NA}(\mathcal{R}_i, \mathcal{R}_j) = \mathbb{1}_{(\mathcal{R}_i \cap \mathcal{R}_j) \wedge (\mathcal{R}_i \neq \mathcal{R}_j)} \quad (2)$$

Method	Type of Training Imagery	Texas (validation)			Alps			California			Gobi Desert			Amazon			Toshka Lakes		
		R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@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
NetVLAD [4]	Urban VPR	3.6	13.4	34.8	5.2	15.2	36.2	4.3	14.4	39.0	2.1	7.7	24.5	6.6	19.9	46.3	9.6	22.5	45.6
SFRS [23]	Urban VPR	6.8	17.4	40.9	7.0	17.8	41.4	5.4	17.2	44.4	5.2	14.7	37.6	9.4	23.8	52.5	10.8	20.4	42.4
Conv-AP [1]	Urban VPR	3.7	13.5	37.7	2.0	8.8	25.0	3.8	15.4	41.2	3.7	9.0	24.2	7.0	21.1	47.7	5.0	12.3	32.6
CosPlace [9]	Urban VPR	4.9	15.2	42.0	5.9	19.3	45.2	4.8	15.2	43.3	7.9	17.5	39.5	8.5	23.9	54.4	9.8	25.4	53.0
MixVPR [2]	Urban VPR	4.5	15.3	40.1	3.1	8.6	22.6	4.4	17.5	43.2	4.0	10.1	24.5	9.8	28.7	58.7	5.9	15.4	36.5
EigenPlaces [10]	Urban VPR	6.4	21.6	52.4	8.7	21.4	50.3	6.0	22.8	54.3	8.1	20.0	40.9	10.9	26.5	57.9	14.3	30.9	60.4
AnyLoc [30] (DINOv2 [41] + NetVLAD)	Universal VPR	<u>44.1</u>	<u>68.7</u>	<b>87.8</b>	<u>40.7</u>	<u>70.8</u>	<b>92.0</b>	<u>48.7</u>	<b>75.0</b>	<b>91.6</b>	<u>28.7</u>	<u>57.0</u>	<u>81.7</u>	<u>38.6</u>	<u>63.8</u>	<b>86.2</b>	<u>63.7</u>	<b>84.5</b>	<b>96.3</b>
TorchGeo [57] (ResNet50 w MOCO [26])	Satellite	1.0	3.2	11.6	0.3	1.3	5.7	1.1	4.3	16.4	0.4	2.8	8.3	1.9	6.6	19.4	0.7	2.9	9.9
TorchGeo [57] (ResNet50 w SeCo [70])	Satellite	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 [57] (ResNet50 w GASSL [5])	Satellite	<u>9.7</u>	<u>22.8</u>	<u>46.4</u>	<u>9.1</u>	<u>23.1</u>	<u>50.5</u>	<u>13.3</u>	<u>31.4</u>	<u>58.8</u>	<u>6.3</u>	<u>17.5</u>	<u>45.4</u>	<u>8.3</u>	<u>20.3</u>	<u>40.1</u>	<u>20.4</u>	<u>38.6</u>	<u>64.2</u>
OGCL UAV-View [18] (ConvNeXt-XXLarge)	UAV	16.2	<u>35.3</u>	<u>65.8</u>	14.4	<u>34.1</u>	<u>64.7</u>	19.6	42.1	71.0	5.5	20.5	45.3	10.0	26.3	54.7	21.1	38.4	63.4
OGCL UAV-View [18] (ViT-L/14) [19]	UAV	<u>17.6</u>	33.2	55.9	<u>14.6</u>	33.2	63.9	<u>22.8</u>	<u>48.1</u>	<u>74.9</u>	<u>7.6</u>	<u>22.8</u>	<u>50.1</u>	<u>20.4</u>	<u>39.1</u>	<u>62.5</u>	<u>31.8</u>	<u>51.8</u>	<u>74.4</u>
MBEG [73] (ViT-L/14)	UAV	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
EarthLoc (Ours)	Satellite	<b>54.6</b>	<b>72.1</b>	<u>87.5</u>	<b>53.9</b>	<b>71.9</b>	<u>87.2</u>	<b>55.9</b>	<u>74.6</u>	<b>91.6</b>	<b>46.8</b>	<b>65.0</b>	<b>82.9</b>	<b>45.6</b>	<b>66.6</b>	<u>82.4</u>	<b>67.6</b>	<u>80.3</u>	<u>91.9</u>

Table 2. **Results from different methods on our query sets.** Methods are grouped into (1) naive, (2) VPR, (3) Remote Sensing (UAV+Satellite), (4) other baselines trained on our database with clustered batches (Sec. 4.2). For all methods we use  $4x90TTA$ . VPR stands for Visual Place Recognition, UAV for Unmanned Aerial Vehicle. Best results overall in **bold**, best per-group underlined. EarthLoc outperforms other models, and performs competitively with AnyLoc, while having 50 times faster features extraction and 10 times smaller features.

With this, the neutral-aware multi-similarity loss becomes

$$\mathcal{L}_{NAMS} = \frac{1}{BS} \sum_{i=1}^{BS} \left\{ \frac{1}{\alpha} \log \left[ 1 + \sum_{k \in \mathcal{R}_i} e^{-\alpha(S_{ik} - \lambda)} \right] + \frac{1}{\beta} \log \left[ 1 + \sum_{k \notin \mathcal{R}_i} \mathcal{I}_{NA}(\mathcal{R}_i, \mathcal{R}_k) e^{-\beta(S_{ik} - \lambda)} \right] \right\} \quad (3)$$

where  $BS$  is the number of images in a batch,  $\mathcal{R}_i$  is the region of image  $i$ ,  $S_{ik}$  is the similarity between image  $i$  and  $j$ ,  $\lambda$  is a margin, and  $\alpha, \beta$  are hyperparameters.

We believe that the neutral-aware multi-similarity loss is applicable to any task where it is not straightforward how to enforce full positivity or negativity between any pair of images, including localization tasks like VPR.

## 5. Experiments

### 5.1. Experimental setting

#### 5.1.1 Training

In training, we use a batch size of 32 quadruplets (128 images), where one quadruplet corresponds to four images within a region. Our model has a MixVPR-style architecture [2], with a ResNet50 [25] backbone and an MLP-Mixer[62] with output dimension 4096. We train for 50k iterations with the Adam optimizer [33] and learning rate 0.0001. Clustering (see Sec. 4.2) is performed every 5k iterations with the number of clusters  $C = 200$ . As augmentation, we use torchvision’s color jittering, random perspective, and random rotation. Training takes between 10 and 12 hours on an A100 GPU for all methods described in Sec. 4.

#### 5.1.2 Inference

Due to the imagery acquisition conditions, we cannot assume any canonical orientation for the photos (in space, there’s no gravity vector, *i.e.*, no “up” or “down”).

This poses a challenge for any retrieval method – to address it, we generate four feature embeddings for each database image, one each after rotating the image for angles  $r \in [0, 90, 180, 270]$ , in a process we call  $4x90^\circ$  test time augmentation ( $4x90TTA$ ). For fairness, we do this for all methods in all experiments, given that the models are not trained to be robust to such large arbitrary rotations. This increases latency by 4x, but the method is still orders of magnitude faster than previous approaches (retrieval takes 0.05 seconds per query with  $4x90TTA$ , while previous works required  $\sim 1$  minute [58] per query). Memory requirements also increase by four times, although this is far from being an issue for a standard workstation, since all database features for the whole Earth fit in less than 12GB of RAM. Experiments without  $4x90TTA$  are shown in the Supplementary in Sec. 7. As an added bonus, since each feature from the database refers to an image with a certain orientation (w.r.t. North), retrieval predictions also give an estimate of the orientation of the query. We use features from the 2021 satellite imagery as our global database.

**Success Criteria** With the final goal of localizing the 4.5M astronaut photograph queries in mind, we define a predicted image as correct if there is non-zero overlap between the predication and the query. As a metric we use Recall@N, defined as the percentage of queries for which at least one of the top-N predictions is correct.

### 5.2. Results

In Tab. 2 we present results from a large number of models from multiple domains. First, we show the recall when naively predicting the query to be directly nadir, proving that the vast majority of images are taken at an angle. Second, we show the poor performance of randomly choosing a prediction from the database - this is a symptom of the large-scale nature of the task (*i.e.*, localizing a 5000 sq.km wide image within a 20 million sq.km area).

Multi-similarity Loss	Using regional quadruplets	Clustered Batches	Year-Wise Data Augmentation	Neutral-Aware Multi-Similarity	Texas (validation)		Alps		California		Gobi Desert		Amazon		Toshka Lakes		Average	
					R@1	R@100	R@1	R@100	R@1	R@100	R@1	R@100	R@1	R@100	R@1	R@100	R@1	R@100
✓					25.5	71.4	24.8	77.3	27.8	76.6	16.8	68.8	23.6	61.5	38.8	80.8	26.2	72.7
✓	✓				45.0	84.1	45.4	87.3	49.5	88.5	35.7	79.9	34.2	80.2	62.9	91.1	45.4	85.2
✓	✓	✓			51.4	87.5	49.6	89.5	53.9	90.6	40.8	84.3	39.3	82.8	64.0	93.0	49.8	88.0
	✓	✓	✓		54.6	87.5	53.9	87.2	55.9	<b>91.6</b>	46.8	82.9	45.6	82.4	67.6	91.9	54.1	87.2
	✓	✓		✓	55.3	88.2	55.1	<b>90.0</b>	57.2	90.3	44.6	85.8	<b>48.4</b>	<b>84.8</b>	68.9	92.9	54.9	88.7
	✓	✓	✓	✓	<b>55.9</b>	<b>88.3</b>	<b>58.4</b>	89.5	<b>58.0</b>	91.4	<b>51.1</b>	<b>86.5</b>	47.2	84.6	<b>72.2</b>	<b>93.3</b>	<b>57.1</b>	<b>88.9</b>

Table 3. **Ablations over test sets.** EarthLoc uses Clustered Batches (Sec. 4.2), Year-Wise Data Augmentation (Sec. 4.3) and Neutral-Aware Multi-Similarity Loss (Sec. 4.4). Best results in **bold**.

Results with methods trained for Urban Visual Place Recognition (VPR) provide, somewhat unsurprisingly, poor results - on the other hand, AnyLoc [30] shows remarkably strong results, correctly predicting the location of the query as the first prediction for roughly 50% of the queries across all evaluation sets, although at the cost of over 50x slower feature extraction and 10x bigger features than its VPR counterparts (AnyLoc uses 49152-D features vs 4096 of NetVLAD).

We then test a number of models trained with self-supervised techniques, like MOCO [26], SeCo [70], and Geography-aware self-supervised learning (GASSL) [5], conveniently provided by the TorchGeo library [57]. Surprisingly, these methods did not provide competitive results, and are on average about equal with Urban VPR methods.

From the Unmanned Aerial Vehicle (UAV) domain, we use the latest state of the art (SOTA) methods, *i.e.*, the winners of the recent challenge in UAV localization at ACM MM (October 2023) [72], namely OGCL [18] and MBEG [73]. Despite being SOTA in the related task of UAV localization, we see that these methods are not competitive with AnyLoc, despite outperforming most other methods.

Finally, EarthLoc outperforms all previous works, localizing 57.1% of queries within the top-1 prediction (Recall@1) averaged on the 6 evaluation sets, while having 50 times faster extraction time than the second best method (AnyLoc) and ten times smaller features.

### 5.3. Ablations

In Tab. 3 we ablate the components of our method to better understand performance. Results show that not only do Clustered Batches, Year-Wise Data Augmentation and Neutral-Aware Multi-Similarity Loss provide significant increases over the baseline, but they are also orthogonal improvements and can be easily combined.

### 5.4. Qualitative Analysis and Failure Cases

A sample of qualitative results from EarthLoc are shown in Fig. 7. The samples show that EarthLoc picks locations that have similar characteristics to the query (top left), is robust to rotations and changes in scale (bottom right), and provides correct predictions that are difficult to correctly match even for the human eye (top left, bottom left). While a number of failure cases arise, we could not identify a repetitive pattern for wrongly localized queries, and believe that many



Figure 7. **Qualitative results.** Each group of four images represents a query (left) and its top-3 predictions. Green and red indicate if the prediction is correct or incorrect.

of these failure cases can be addressed by better architectures and training paradigms. More qualitative results are shown in Sec. 11 of the Supplementary.

### 5.5. Limitations and Future Work

While this work takes a major step forward in the study of APL, EarthLoc focuses on daytime queries, and we did not investigate panoramic or nighttime imagery. The localization of nighttime astronaut photos in particular is important (e.g. to study light pollution [14]), but we believe this to be a separate challenge that should be addressed in future work. Another limitation is due to the zoom limits of the database: we chose database tiles at zoom levels 9, 10, 11 (see Sec. 3.2), which allows us to localize queries of area between 5000 and 900,000 square km. Queries with smaller or bigger covered areas would need finer or coarser database images to be geolocated.

The originality of this work opens up a large number of possible future research directions, like training directly on (some of the) localized queries, using seasonal data, post-processing predictions, obtaining pixel-wise geolocation, applying domain adaptation, using available metadata (like camera lens) during training, et cetera.

## 6. Conclusions

We introduce EarthLoc, a novel method for localizing astronaut photography of Earth. By reframing APL from an image matching task to an image retrieval task, and introducing a novel loss function and training scheme, our method accurately and efficiently determines the geographic location of astronaut photography given only the position of the International Space Station at the time the photo was taken. Our main contributions are the Neutral-Aware Multi-Similarity Loss, Year-Wise Data Augmentation technique, and new astronaut photography-

oriented validation and test datasets to encourage future work on this problem.

**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 Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. [3](#), [7](#)
- [2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition. In *WACV*, pages 2998–3007, 2023. [3](#), [7](#)
- [3] R. Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012. [2](#)
- [4] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pa-jdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE TPAMI*, 40(6): 1437–1451, 2018. [3](#), [7](#)
- [5] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. *ICCV*, 2021. [3](#), [7](#), [8](#)
- [6] Artem Babenko, Anton Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. *ECCV*, abs/1404.1777, 2014. [3](#)
- [7] Giovanni Barbarani, Mohamad Mostafa, Hajali Bayramov, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Are local features all you need for cross-domain visual place recognition? In *CVPRW*, pages 6155–6165, 2023. [3](#)
- [8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 2008. [2](#)
- [9] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *CVPR*, 2022. [3](#), [7](#)
- [10] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *ICCV*, pages 11080–11090, 2023. [7](#)
- [11] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, pages 1861–1870, 2019. [3](#)
- [12] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *NeurIPS*, 2022. [3](#)
- [13] Gabriela Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004. [2](#)
- [14] Alejandro Sánchez de Miguel, José Gómez Castaño, Jaime Zamorano, Sergio Pascual, M Ángeles, L Cayuela, Guillermo Martín Martínez, Peter Challupner, and Christopher C M Kyba. Atlas of astronaut photos of earth at night. *Astronomy and Geophysics*, 55(4):4.36–4.36, 2014. [8](#)
- [15] 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. [2](#)
- [16] Jiankang Deng, J. Guo, and S. Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4685–4694, 2019. [3](#)
- [17] Fabian Deuser, Konrad Habel, and Norbert Oswald. Sample4geo: Hard negative sampling for cross-view geo-localisation. In *ICCV*, pages 16847–16856, 2023. [3](#)
- [18] 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](#), [7](#), [8](#)
- [19] 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. In *ICLR*, 2021. [7](#)
- [20] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelwagen. Uncertainty-aware vision-based metric cross-view geolocalization. In *CVPR*, pages 21621–21631, 2023. [3](#)
- [21] Kenton Fisher, Sara Schmidt, and Alex Stoken. Crew earth observations: New tools to support your research. In *12th Annual International Space Station Research and Development Conference*. Center for the Advancement of Science in Space, Inc., 2023. [2](#)
- [22] 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. [2](#)
- [23] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV*, pages 369–386, Cham, 2020. Springer International Publishing. [7](#)
- [24] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *CVPR*, pages 14141–14152, 2021. [3](#)
- [25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [7](#)

- [26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020. 7, 8
- [27] Christopher R. Jackson and Werner Alpers. The role of the critical angle in brightness reversals on sunglint images of the sea surface. *Journal of Geophysical Research: Oceans*, 115(C9), 2010. 2
- [28] 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. 2
- [29] Hervé Jégou, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE TPAMI*, 34, 2011. 2
- [30] 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
- [31] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 3
- [32] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geolocalization. In *CVPR*, pages 3251–3260, 2017. 3
- [33] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 7
- [34] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Data-efficient large scale place recognition with graded similarity supervision. *CVPR*, 2023. 3
- [35] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516. PMLR, 2016. 3
- [36] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017. 3
- [37] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 2
- [38] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 3
- [39] Oscar Mañas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodríguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *ICCV*, pages 9394–9403, 2021. 3
- [40] 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. 2
- [41] 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
- [42] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391, 2010. 2
- [43] F. Radenović, G. Toliás, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE TPAMI*, 2018. 3
- [44] A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual Instance Retrieval with Deep Convolutional Networks. *CoRR*, abs/1412.6574, 2015. 3
- [45] Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature Communications*, 12(1), 2021. 3
- [46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3
- [47] 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. 2
- [48] Johannes A. Schultz, Maik Hartmann, Sascha Heinemann, Jens Janke, Carsten Jürgens, Dieter Oertel, Gernot Rücker, Frank Thonfeld, and Andreas Rienow. Diego: A multispectral thermal mission for earth observation on the international space station. *European Journal of Remote Sensing*, 53(sup2):28–38, 2020. 2
- [49] Peter Schwind and Tobias Storch. Georeferencing urban nighttime lights imagery using street network maps. *Remote Sensing*, 14(11), 2022. 3
- [50] Yujiao Shi and Hongdong Li. Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In *CVPR*, pages 17010–17020, 2022. 3
- [51] Yujiao Shi, Xin Yu, Shan Wang, and Hongdong Li. Cvlnet: Cross-view feature correspondence learning for video-based camera localization. In *ACCV*, pages 652–669, 2022.
- [52] Yujiao Shi, Fei Wu, Akhil Perincherri, Ankit Vora, and Hongdong Li. Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer. In *ICCV*, pages 21516–21526, 2023. 3
- [53] Christopher Small. Spectrometry of the urban lightscape. *Technologies*, 10(4), 2022. 2
- [54] H. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. Spotlight presentation. 3
- [55] Weiwei Song, Zhi Gao, Renwei Dian, Pedram Ghamisi, Yongjun Zhang, and Jón Atli Benediktsson. Asymmetric hash code learning for remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. 3
- [56] 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. 2

- [57] Adam J. Stewart, Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, and Arindam Banerjee. Torch-Geo: 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. 7, 8
- [58] Alex Stoken and Kenton Fisher. Find my astronaut photo: Automated localization and georectification of astronaut photography. In *CVPRW*, pages 6196–6205, 2023. 2, 3, 4, 5, 7
- [59] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 3
- [60] Yifan Sun, Changmao Cheng, Yuhang Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. *CVPR*, pages 6397–6406, 2020. 3
- [61] Giorgos Tolias, R. Sivic, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. *CoRR*, abs/1511.05879, 2016. 3
- [62] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, pages 24261–24272. Curran Associates, Inc., 2021. 7
- [63] Gabriele Trivigno, Gabriele Berton, Juan Aragon, Barbara Caputo, and Carlo Masone. Divide&classify: Fine-grained classification for city-wide visual geo-localization. In *ICCV*, pages 11142–11152, 2023. 3
- [64] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 3
- [65] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274. Computer Vision Foundation / IEEE Computer Society, 2018. 3
- [66] Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE TCSVT*, 2021. doi: [10.1109/TCSVT.2021.3061265](https://doi.org/10.1109/TCSVT.2021.3061265). 3
- [67] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pages 5022–5030, 2019. 3, 5
- [68] M. Justin Wilkinson and Yann Gunnell. *Fluvial Megafans on Earth and Mars*. Cambridge University Press, 2023. 2
- [69] 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. 2
- [70] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, 2021. 7, 8
- [71] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. *ACM MM*, 2020. 3
- [72] 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. 8
- [73] 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, 7, 8
- [74] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *CVPR*, pages 3640–3649, 2021. 3
- [75] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. In *CVPR*, pages 1162–1171, 2022. 3
- [76] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *CVPR*, pages 19370–19380, 2023. 3

# EarthLoc: Astronaut Photography Localization by Indexing Earth from Space

## Supplementary Material

### 7. Empirical Investigation of Challenge Modes

The astronaut photography localization task has various challenge modes, where particular photography conditions make localization more difficult for one image compared to another. In this section, we analyze the correlation between two of these conditions (challenge modes) and performance.

**Distance from Nadir** As the distance between the ISS nadir point and photo location increases, so too does obliquity and shear due to the imaging geometry. Oblique imagery is often taken through a thicker column of atmosphere, adding blurriness to the image. Thus, we expect some performance drop to accompany increasing distance, as seen beyond 400km in Fig. 8. Further augmentation to satellite images during training can potentially close this gap, particularly if these augmentations are designed to simulate conditions seen in far-from-nadir imagery.

**Area** We next analyze the correlation between an astronaut photo’s geographic area encompassed (area) and recall (Fig. 9). Here, recall decreases as area increases. Based on our training set construction (see Sec. 3.2), there are fewer images with larger areas required to cover the extent (land area between  $\pm 60^\circ$  latitude), so fewer such images are included in training. We expect including more such images during training will improve performance for this type of imagery.

### 8. Effect of 4x90 TTA

We also study the impact of our *4x90TTA* strategy, with results reported in Tab. 4. In each evaluation set, performance improves due to *4x90TTA*. Based on orientation information from the Gateway to Astronaut Photography of Earth, there are approximate the same number of images across orientation angles in  $[0^\circ, 360^\circ)$ . However, the performance does not increase in proportion to the TTA (i.e., by 4x), indicating that EarthLoc learns some rotation invariance during training. This observation is further supported by multiple rotations of the same image present in the top 3 predictions, examples of which can be seen in Fig. 13. Note that even in such cases, the database image with the closest orientation is usually the top prediction, followed by correct database images with other orientations.

Despite EarthLoc’s ability to retrieve correct images with other orientations, the ablation experiment in Tab. 4 shows that this capability is limited, and that *4x90TTA* significantly boosts recall on all evaluation sets.

### 9. Features visualizations

Given the domain gap existing between database and queries images, we show in Fig. 10 the distribution of their features, as extracted by EarthLoc, through a T-SNE.

Test Time Aug.	Recall@1					
	Texas	Alps	California	Gobi	Amazon	Toshka
None	32.6	30.9	31.1	30.4	27.2	39.1
<i>4x90TTA</i>	<b>54.6</b>	<b>53.9</b>	<b>55.9</b>	<b>46.8</b>	<b>45.6</b>	<b>67.6</b>

Table 4. **Ablation on *4x90TTA***. Performance of EarthLoc on each evaluation set with and without *4x90TTA*. *4x90TTA* approximately doubles Recall@1 across the sets.

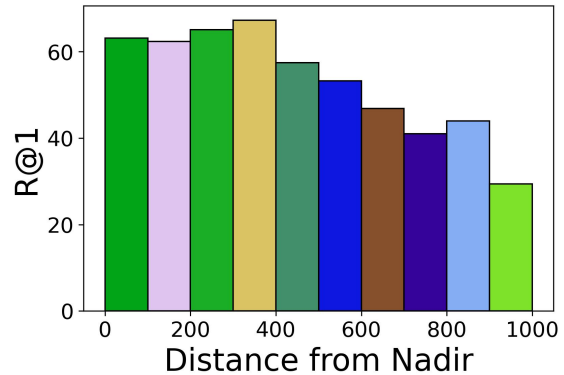


Figure 8. **Recall@1 vs Distance from Nadir (km)**. Performance of EarthLoc as distance from nadir increases. Larger distances are more challenging due to obliquity effects, and this is reflected in the drop in performance.

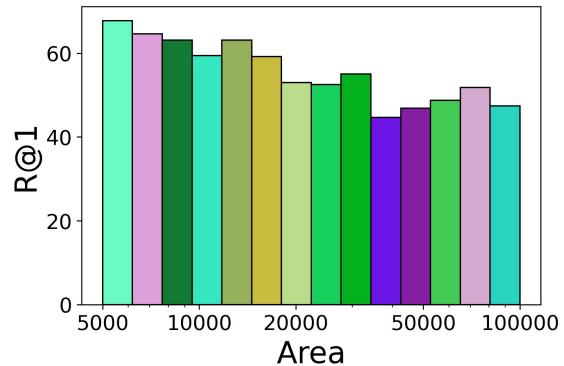


Figure 9. **Recall@1 vs Area (sq. km)**. Performance of EarthLoc on astronaut photo queries with different areas. Decrease in performance for larger area photos is attributed to lower quantities of training data for this regime.

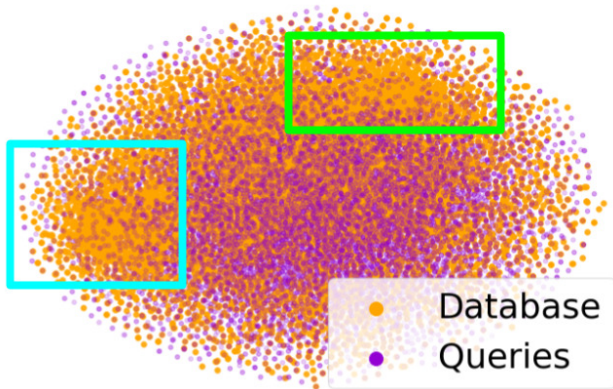


Figure 10. **T-SNE representation.** The two colored boxes represent geography types with few queries (areas rarely photographed by astronauts), namely open sea (light blue box) and plains (green).

## 10. Further Examples of Queries and Database

We present additional examples of astronaut photo queries (Fig. 11) and satellite database images (Fig. 12).

These example queries illustrate the variety inherent in astronaut photography. Some images have occlusion due to spacecraft hardware, some images contain clouds, and all have different orientations with respect to North. There is significant geographic and scale variation, with some photos highlighting neighborhoods of large cities and others showing entire lakes or vast mountain ranges.

The satellite imagery is more regular. In addition to uniform orientation, these images are algorithmically post-processed and thus share similar characteristics, which are somewhat different from those of astronaut photography. Though not all satellite imagery is cloudfree, this particular set is constructed to minimize clouds, and consequently few clouds are seen in the example images.

## 11. Qualitative Results

### 11.1. Examples of Queries and Predictions

Example queries and associated top 3 predictions from EarthLoc are in Fig. 13. Often, multiple orientations of the same database image are within the top 3. In other cases, correct database images with different areas (scale) are retrieved.

In examples where no correct database image is retrieved, similar looking images are often returned (left, third from bottom). In some failure cases, however, predictions do not show significant similarity to the query (bottom left). We have not found any unifying characteristics in such scenarios.

### 11.2. Examples of Clustered Batches

As described in Sec. 4.2, we construct batches from clusters to facilitate training. Samples of such clusters are in Fig. 14. Clusters are built by collecting *regions* that have similar representations in feature space, despite being from potentially far reaching places on Earth. These similar representations often correspond to shared characteristics, and we can assign high level labels to clusters, like “rivers in the forest” to the top cluster and “mountainous deserts” for the second.

During training, the loss function works to separate the representations for different classes/*regions* that are batched together, so building batches from clusters provides a much more challenging optimization task than random batching, as intra-cluster (across the row) images are much more similar than inter-cluster images (down the columns).

### 11.3. Examples of Year-Wise Data Augmentation

We show a subset of a training batch as it is presented to the model (i.e., after augmentation) in Fig. 15. This illustrates our Year-Wise Data Augmentation. Each half-row (4 images) is a *region* quadruplet, with one image from each of the years 2018, 2019, 2020, and 2021. Columns are arranged by year (i.e., each column contains images from the same year). According to our Year-Wise Data Augmentation scheme (see Sec. 4.3), images from the same year (i.e., in each column) have received the same augmentation. Augmentations are color jittering, random perspective, and random rotation.

<sup>9</sup><https://eol.jsc.nasa.gov>

<sup>10</sup><https://s2maps.eu>

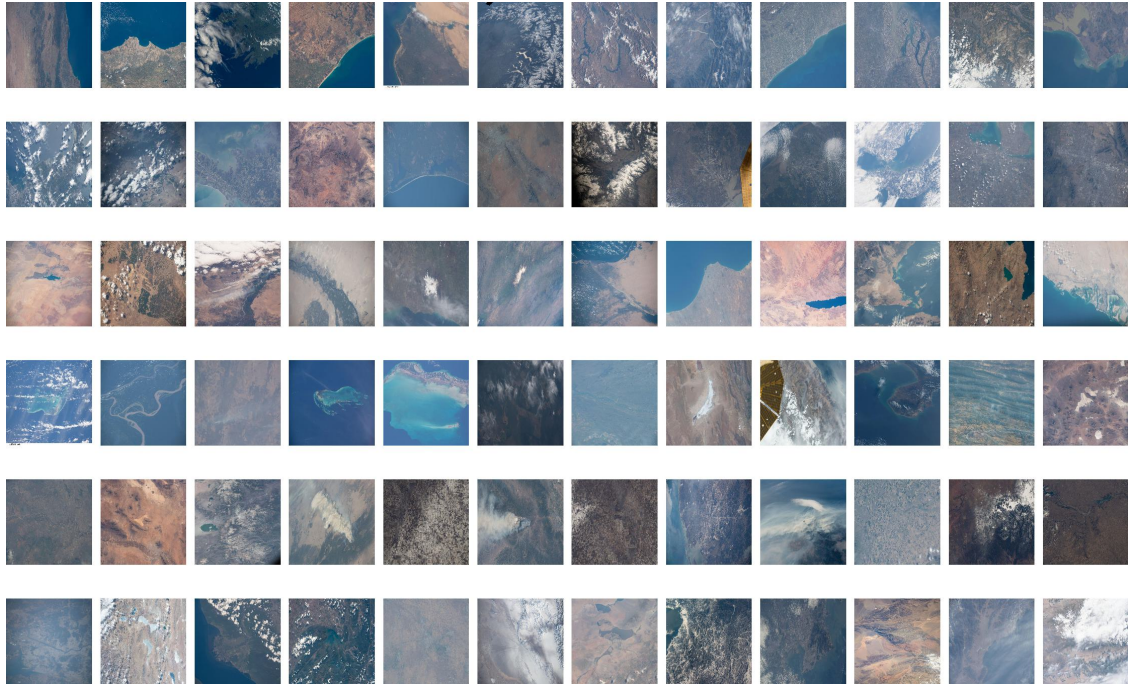


Figure 11. **Random examples of queries.** All queries come from the Gateway to Astronaut Photography of Earth collection<sup>9</sup>. Each row is a randomly selected set of queries from each test set, respectively being Alps, Texas, Toshka Lakes, Amazon, California, Gobi.



Figure 12. **Random examples of database images.** All database images come from the EOX Sentinel-2 cloudless collection.<sup>10</sup>



Figure 13. **Qualitative Examples of EarthLoc Predictions.** Query image and top 3 predictions. Green indicates a correct prediction, red an incorrect prediction. Each half-row is a separate example.

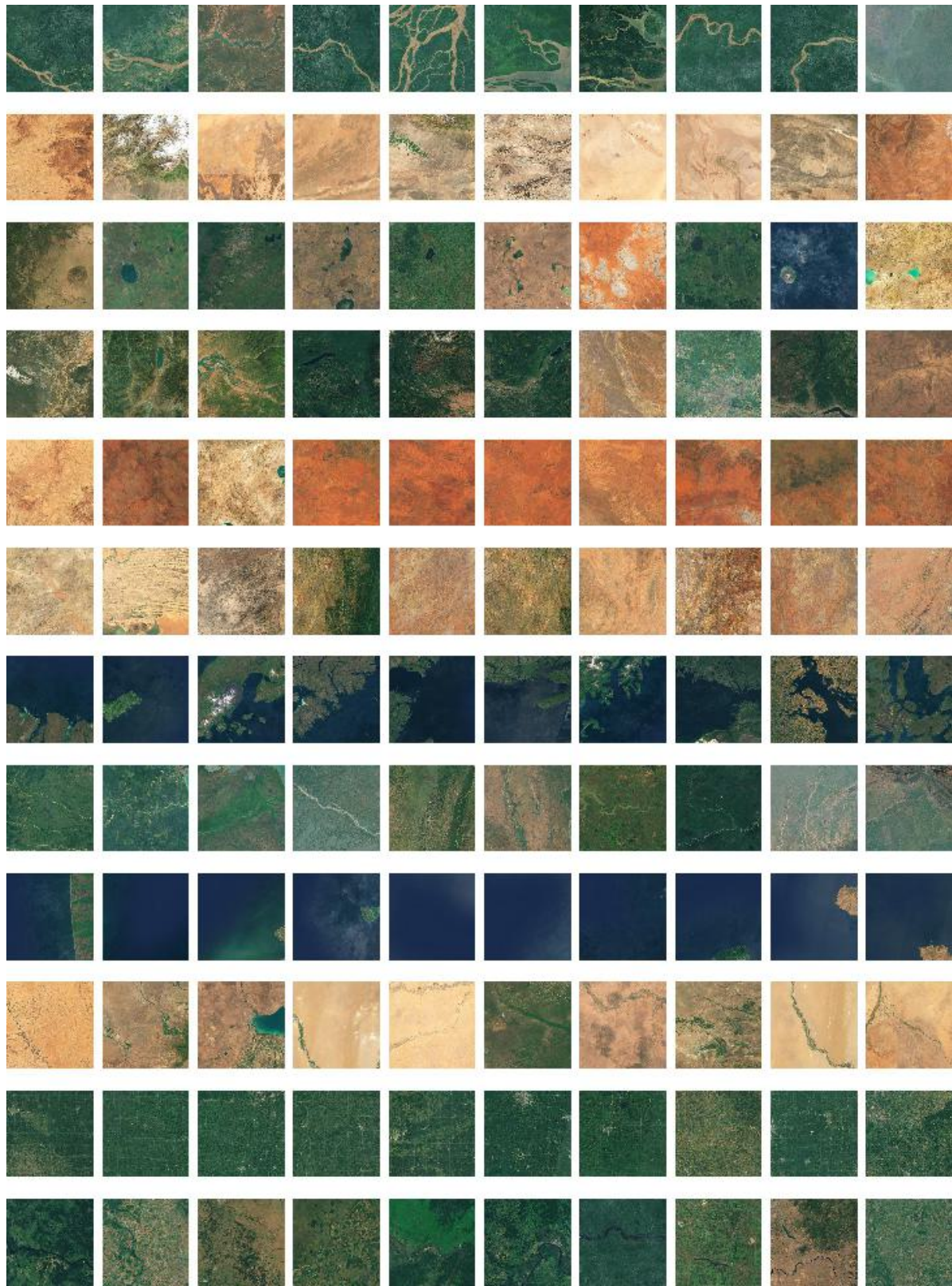


Figure 14. **Examples of clusters, one cluster per row.** Clusters are formed from images with similar representations. Training batches are selected from images within one cluster (see Sec. 4.2).

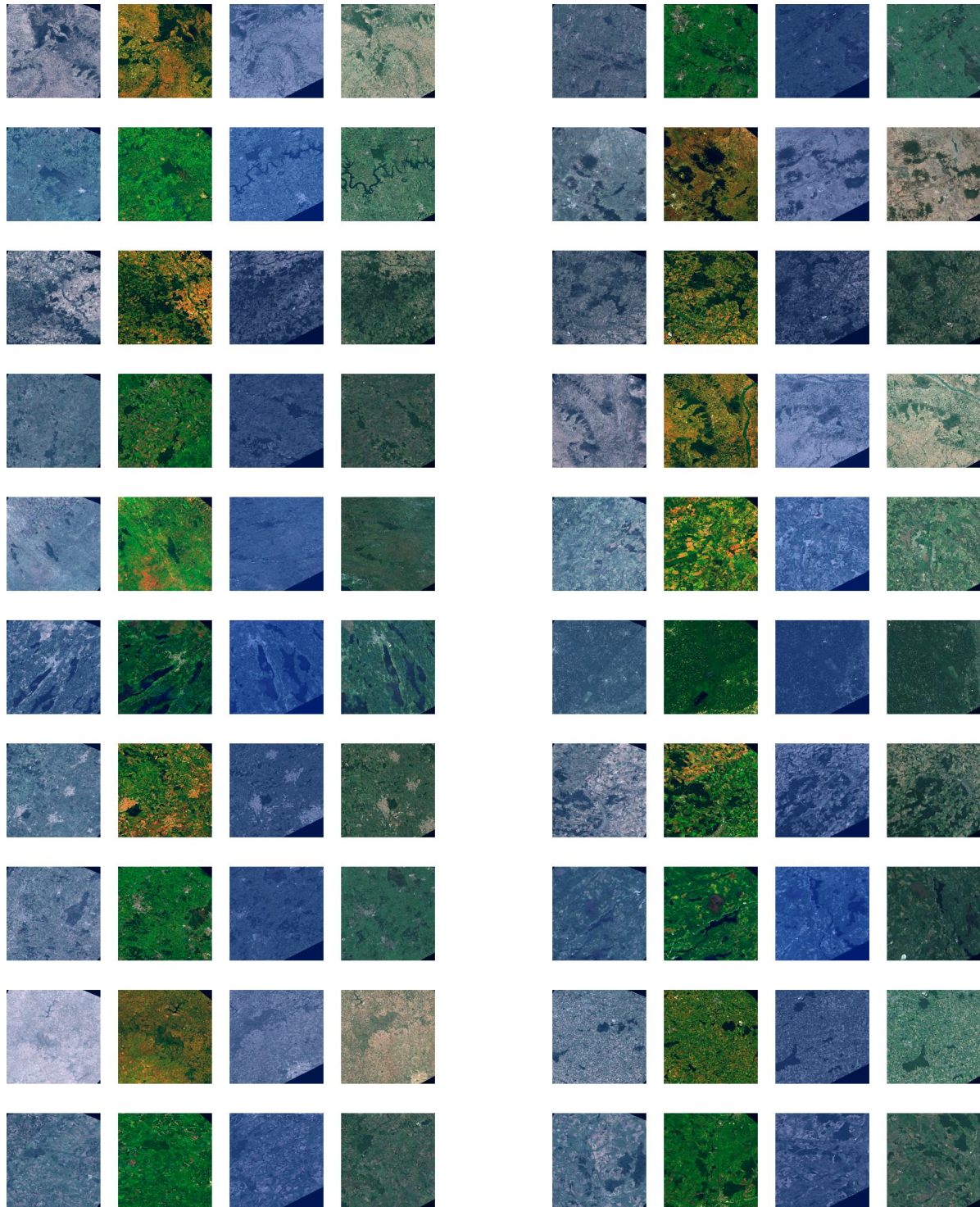


Figure 15. **An example of a batch**, showing 20 out of 32 quadruplets due to space limits. Each half-row of 4 images represents the quadruplet from one *region*. Each *region* has images from 2018, 2019, 2020 and 2021. The same augmentation is applied to all images from a given year, e.g. the images from 2020 receive a blueish color transformation in this batch.