



GeoImageNet: a multi-source natural feature benchmark dataset for GeoAI and supervised machine learning

Wenwen Li¹ · Sizhe Wang¹ · Samantha T. Arundel² · Chia-Yu Hsu¹

Received: 23 August 2021 / Revised: 5 April 2022 / Accepted: 17 August 2022 /

Published online: 3 September 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022, corrected publication 2023

Abstract

The field of GeoAI or Geospatial Artificial Intelligence has undergone rapid development since 2017. It has been widely applied to address environmental and social science problems, from understanding climate change to tracking the spread of infectious disease. A foundational task in advancing GeoAI research is the creation of open, benchmark datasets to train and evaluate the performance of GeoAI models. While a number of datasets have been published, very few have centered on the natural terrain and its landforms. To bridge this gulf, this paper introduces a first-of-its-kind benchmark dataset, GeoImageNet, which supports natural feature detection in a supervised machine-learning paradigm. A distinctive feature of this dataset is the fusion of multi-source data, including both remote sensing imagery and DEM in depicting spatial objects of interest. This multi-source dataset allows a GeoAI model to extract rich spatio-contextual information to gain stronger confidence in high-precision object detection and recognition. The image dataset is tested with a multi-source GeoAI extension against two well-known object detection models, Faster-RCNN and RetinaNet. The results demonstrate the robustness of the dataset in aiding GeoAI models to achieve convergence and the superiority of multi-source data in yielding much higher prediction accuracy than the commonly used single data source.

Keywords GeoAI · Deep learning · Object detection · RetinaNet · Remote sensing

1 Introduction

The world has been experiencing increasing changes in its natural and built environment; addressing pressing issues, such as global climate change, infectious disease, and more frequent and damaging natural disasters, requires the synthesis of data and knowledge across traditional disciplinary boundaries. Fortunately, advances in earth observing techniques

✉ Wenwen Li
wenwen@asu.edu

¹ School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ 85287, USA

² United States Geological Survey, Center of Excellence for Geospatial Information Science (CEGIS), Rolla, MO 65401, USA

and progress in smart city initiatives have resulted in a large, interconnected physical- and cyber-infrastructure that instantaneously amasses data about nature and human activity [1, 2]. Despite this progress, traditional data analytical techniques, such as object-based image analysis, which are tailored to process low-quantity and high-quality data, have shown significant limitations in processing the big data available today [3].

Artificial intelligence and machine learning, especially the recent revolution in deep learning, have established a new domain for the analytics of geospatial big data [4]. Convolutional neural networks (CNNs) are prominent as a deep learning approach because of their outstanding ability to extract discriminative features from raw data (such as optical images) and perform classification or prediction tasks effectively. Its unique use of the convolutional module with localized connections also makes the neural network easy to parallelize and run on a multi-processor Graphic Processing Unit (GPU), which has recently decreased in cost while enjoying increased computational power. Geospatial artificial intelligence (GeoAI) is an emerging research realm that integrates artificial intelligence into geospatial data analytics for solving big data problems across environmental and social science domains [5].

Geospatial research has immediate need for GeoAI and deep learning, because of the vast amount of data that are currently available in image form, such as high-resolution remote sensing imagery, digital maps, and the output of numerical simulation models. Many remote sensing applications, such as land use and land cover classification [6], disaster management [7], natural feature detection [8], and semantic segmentation of geospatial objects [9, 10], have demonstrated the successful application of GeoAI in support of image analysis and computer vision. Similar to other supervised machine learning tasks, a key factor for ensuring the success of GeoAI research is the availability of high-quality training data, without which a machine can hardly deduce useful knowledge and gain sufficient decision intelligence.

The geospatial community has actively pursued the development and publishing of benchmark datasets for training deep learning models. Depending on the GeoAI task, the level of difficulty in preparing training datasets varies. For instance, deep-learning applications in image analysis can be categorized into image-level classification, object detection, and pixel-wise segmentation [11]. Image-level classification aims to predict content or object type in an image scene. Only class labels (e.g., urban, agriculture, building) are needed to annotate an image, there is no need to pinpoint the location of the object. Hence, this task requires the least labeling effort. Object detection goes one step beyond determining object class, because with geographic extent (expressed in a bounding box or BBOX), the machine can make predictions about object location within an image scene. Therefore, two types of labels, object class and object BBOX, are required for training an object detection model. The preparation of BBOX labels is much more labor intensive than the classification task [12]. Pixel-wise segmentation, in comparison, requires the machine to make decisions about the classes of each individual pixel. To accomplish this, the machine needs the same level of detail (pixel-level) during model training. Training data preparation for the segmentation task is therefore the most challenging.

At present, many familiar benchmark datasets supporting geospatial applications, such as the popular UC Merced Land-Use dataset [13], WHU-RS19 [14] and NWPU-RESISC45 [15], have been developed to support image-level classification. These datasets do not contain labels for individual elements of interest. For object detection tasks, in addition to the widely used datasets in the computer vision community, such as ImageNet Object Detection Dataset [16], Pascal VOC (Visual Object Classes; [17]), and COCO (Common Objects in Context; [18]), the earth observation community has prepared benchmark datasets using

remote sensing and aerial images. For instance, VEDAI (Vehicle Detection in Aerial Imagery; [19], and DLR-MVDA (German Aerospace Center – Multi-class Vehicle Detection and Orientation in Aerial Imagery; [20]) datasets are created for vehicle detection. The DOTA dataset (Detecting Objects in Aerial Images; [21] contains multi-class labels for detecting popular objects in an urban scene. The DIOR (Object Detection in Optical Remote Sensing) dataset is another large-scale dataset containing 20 object classes from over 20,000 images collected from Google Earth [1, 2].

Table 1 lists the main characteristics of popular benchmark datasets for image classification and object detection using remote sensing images as the primary data source. A few other well-known datasets, such as Pascal VOC, COCO and ImageNet, which use images taken primarily from cameras, are also listed. All of these datasets provide labels for things and objects, and mostly man-made features, such as a plane or ship. Very few benchmark datasets contain bounding box labels, which are essential for natural feature detection and terrain analysis. GeoNat v1.0 [22] may be the first dataset to support natural feature detection and serves as the basis of our work. However, it only contains object annotations (in the form of polygons) and does not contain associated image data ready for GeoAI and deep learning tasks. Also, all of these datasets prepare only single-source image data for machine learning; there is no location information available to help identify where the images are located or where the photos were taken. However, it is widely acknowledged that geospatial applications could significantly benefit from multi-source earth observation data as they jointly provide a much richer context about the study area and object than single-source data [23, 24]. Hence, having location information explicitly represented in the dataset could facilitate training data validation, expansion, and multi-source data fusion.

Because natural features tend to have vague boundaries due to both physical processes, such as erosion and deposition and the cognitive conceptualization of these fiat features, they are much more challenging to detect than man-made features. In addition, natural features often possess high intra-class variation and inter-class similarity [22], making them difficult to classify. Another challenging trait is their likelihood of varying significantly in both size and appearance across diverse landscapes. Moreover, many natural features, such as ridges and valleys, reside in mountainous regions and express more dramatic elevation change than do built features. These challenges make multi-source data and multi-source learning a hugely valuable solution, as each individual dataset captures for study unique and different characteristics of an object and they jointly contribute to a more comprehensive understanding of the object. Building a GeoAI benchmark dataset that uses multi-source data to support analysis of natural features will also promote the application of cutting-edge AI in landform and terrain analysis, increasing the scientific value of AI methods.

To achieve this, our paper introduces a novel, multi-source benchmark dataset, GeoImageNet, to support natural feature detection and recognition using GeoAI and deep learning. All the other popular datasets, listed in Table 1, contain only single-source data. Another distinction of our dataset is that it contains location information for each image scene so geographic validation and training data expansion is easy to achieve. For instance, it would be feasible to add additional datasets when they become available if the geographical framework (location, extent and spatial reference) is provided. It is also possible for users and dataset reviewers to “fly-in” to the region on Google Earth based on the available location information and gain more information about the study area and to verify the dataset and annotations.

In our GeoImageNet dataset, besides the use of remotely sensed imagery, each labeled image is paired with the Digital Elevation Model (DEM) dataset and its derived parameters

Table 1 A comparison of popular benchmark datasets using remote sensing imagery as the primary data source for image classification and object detection

Dataset	Description	Image size	Applications	Multi-source?	Location information
UC Merced Land-Use dataset	Overhead remote sensing image dataset from USGS for land use classification (Example classes: agriculture, airplane, building, beach)	2100 images (256*256), 21 classes	Image-level classification	No	Not available
WHU-RS19	Remote sensing image dataset from Google Earth for scene classification (Example classes: airport, bridge, forest, meadow)	1005 images (600*600), 19 classes		No	Not available
NWPU-RESISC45	Remote sensing scene classification (Example classes: tennis court, commercial area, palace, church, and river)	31,500 images (256*256), 45 scene classes		No	Not available
Pascal VOC 2012	Natural optical images for object detection (Example classes: person, animal, vehicle, indoor)	11,530 images (averaged 469*387), 20 classes	Object detection	No	Not available
COCO	Natural optical images for object detection (Example classes: person, bicycle, car)	200,000 images (640*480), 80 classes	Object detection and segmentation	No	Not available
ImageNet Object Detection Database	Natural optical images for object detection (Example classes: dog, horse, piano)	500,000 images (averaged 482*415), 200 classes	Object detection	No	Not available

Table 1 (continued)

Dataset	Description	Image size	Applications	Multi-source?	Location information
VEDAI	Aerial ortho-normal images for vehicle detection (Example classes: plane, boat, camping car, car, tractor, and truck)	1210 images (512*512, 1024*1024), 9 classes	Object (vehicle) detection	No	Not available
DLR-MVDA	Aerial images for vehicle detection (Example classes: car, truck)	20 large-scale images (5616*3744), 2 classes		No	Not available
DOTA v1.0	Aerial images for object detection (Example classes: plane, ship, storage tank, vehicle, soccer ball field)	2,806 images (800*800 and above), 15 classes	Object detection	No	Not available
DIOR	Remote sensing images for object detection (Example classes: airplane, airport, baseball field, bridge, golf course, ship, vehicle)	23,463 images (800*800), 20 classes	Object detection	No	Not available

(such as slope, aspect, and hillshade) to capture terrain characteristics from multiple perspectives. Data enrichment strategies are applied to make numerical data (e.g., DEM) more suitable for consumption in a deep learning paradigm. Currently, our GeoImageNet 2022 dataset contains six complex terrain features, including ridge, valley and basin, which have significant terrain changes, and lake, island, and bay, which themselves are more homogeneous in slope. Experiments were conducted on modern object detection models to verify the effectiveness of this dataset for detecting natural features.

The remainder of the paper is organized as follows. Section 2 introduces a search-and-rescue use case where GeoImageNet and AI play a central role in automating and accelerating its process. Section 3 describes data source, selection and preparation of GeoImageNet. Section 4 provides statistics of GeoImageNet. Section 5 reports deep learning-based object detection experiments leveraging the new multi-source benchmark dataset to demonstrate its superior object detection performance as compared to a single remote sensing-based data source. Section 6 concludes the paper with a discussion of possible research directions.

2 A search-and-rescue use case

GeoImageNet and GeoAI can be integral to the workflow of search and rescue operations. Figure 1 simulates the scenario in which a tourist or military troops get lost in a mountainous region. There is no GPS signal, so an accurate location cannot be reported. A rough description about the local terrain is provided: I am down in a valley with an elongated lake in front of me and a tall mountain sits on one side. There are shrubs near the foothills and snow on the summit. A lower mountain sits on the opposite side of the lake.

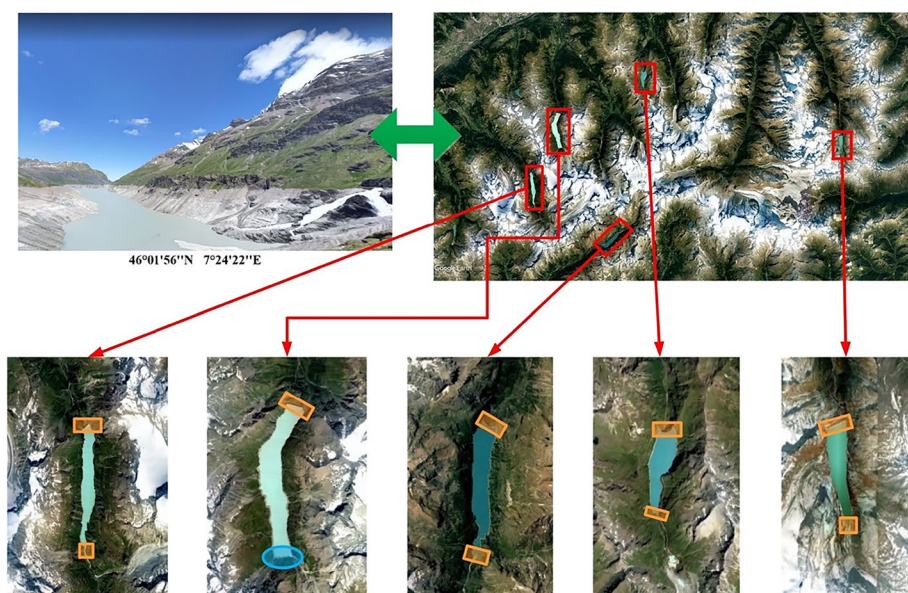


Fig. 1 Search and rescue case in the Alps. Images were retrieved from Google Earth. The satellite image was taken in December 2016, and the street-view photo (upper left photo) was uploaded in July 2016. The center of the photos is located at $46^{\circ}01'56''\text{N}$, $7^{\circ}24'22''\text{E}$. Images listed in the second row show potential locations with the cyan ellipse referring to actual location of the lost persons

Instead of relying solely on manual searches of potential locations, a natural feature dataset and a multi-source learning framework will empower machine intelligence and enable an automatic search process. At the first level of localization, GeoImageNet images can be incorporated into a GeoAI model to detect and extract prominent terrain features (results shown in the photo at the upper-right of Fig. 1). This will result in a list of candidate scenes that need to be further investigated (bottom photos of Fig. 1). Then, GeoImageNet can be further leveraged to extract detailed information about the terrain features (such as location, extent, and shape) in each candidate scene. To facilitate locating the correct lakeside, the elevation dataset in GeoImageNet will play a key role. By comparing the characteristics of the terrain features within a candidate scene with the description from those who are lost, the target location can be identified. Hence, the multi-source dataset is capable of empowering the machine to gain more geospatial intelligence and automation.

3 GeoImageNet: data source, selection and preparation

3.1 Data preparation and labeling method

Data preparation followed several principles: (1) Geographic representativeness and feature variety. According to Tobler's First Law of Geography, nearby things are more alike than are distant things [25]. If features are selected from a single geographic region, their representativeness of external regions will be weakened. The transferability of a GeoAI model trained on such datasets will also be negatively affected. (2) Scale issues. Unlike built features, natural features have a much broader range in their sizes, even within a single class. The complexity in structure and diversity in appearance of natural terrain features make their detection much more challenging than identifying artificial features [26]. Therefore, to ensure the machine's best predictive ability, the size of each feature with regard to the image scene should not be too small (so that they remain visually inspectable from the image), because otherwise their representative characteristics will be difficult to capture with the GeoAI model. Hence, feature rescaling is needed to better present small objects. (3) Data alignment. When multi-source data are used in a deep learning pipeline, they should be aligned in terms of spatial resolution, spatial reference, and geographical extent. Figure 2 demonstrates the data processing workflow.

The first step is to download a local copy of the Geographic Names Information System (GNIS) database, which contains over 3 million named features (both natural and artificial) of the Earth's surface (<https://www.usgs.gov/u.s.-board-on-geographic-names/download-gnis-data>). From this database, we can obtain category and location information (latitude and longitude) of natural features in the US. Both the DEM data and remote sensing imagery covering the extent of selected natural features are downloaded from the U.S. Geological Survey's (USGS) The National Map (TNM). Specifically, the 1/3 arc-second seamless DEM at about 10 m spatial-resolution is used as the elevation data and the National Agriculture Imagery Program (NAIP) aerial imagery at 1 m-resolution is used as the remote sensing data source.

In step 2, candidate features are randomly sampled from the GNIS database as long as they meet the minimum requirement that they have been identified on the historical topographic map. To ensure geographical representativeness, the selected features were spaced out geographically by setting a distance threshold of 48 km (approximately 0.5 degree at 30 degree north). This means the distance between any two selected features should be larger than this set threshold. Once a feature is picked, a distance calculation operation will be invoked; if its distance to any other features that are already included in the feature

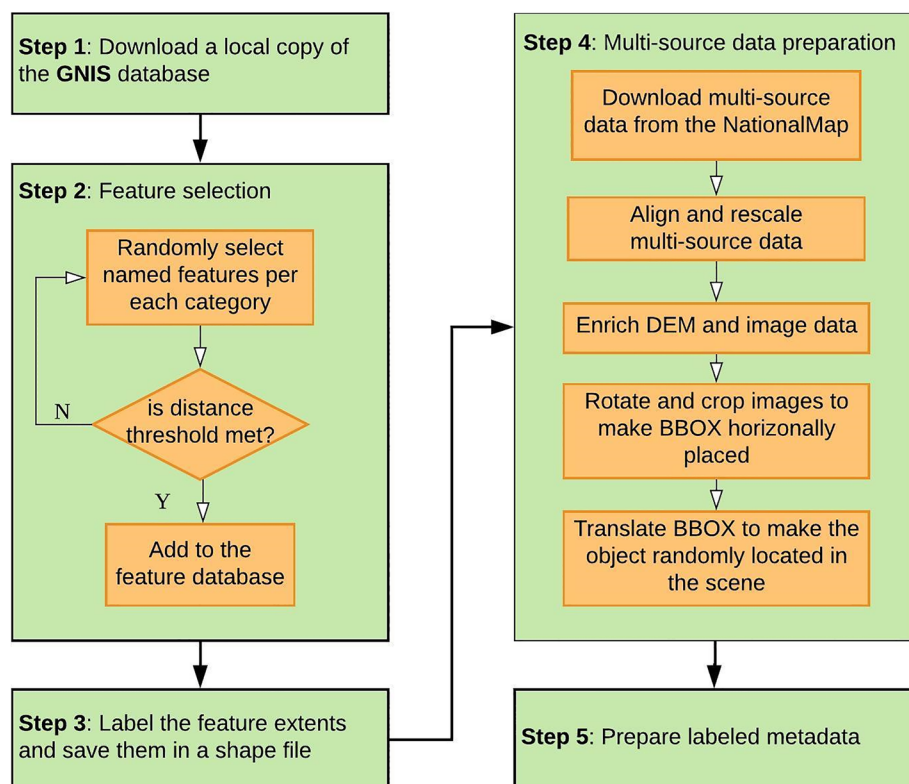


Fig. 2 Data processing workflow for training data preparation

database does not meet the distance threshold, the feature will be disregarded. The search process continues until the number of features per category has reached a certain size. In the 2021 version of GeoImageNet, the expected number is set to be 150 per category.

Step 3 is to label the BBOX of each selected feature. Because the locations of these features are given by a single latitude and longitude in the GNIS database, we need to further provide the BBOX label for each feature. Labeling is completed through visual inspection based on the USGS Historical Topographic Map Collection (HTMC) dataset, aided by NAIP imagery. The HTMC contours are especially helpful in inspecting the basin, valley, and ridge features, which have more significant terrain changes but less obvious variations in color or texture. Note that the BBOX is not labeled in the upright position (Fig. 3). Instead, it is labeled along the longest axis of the feature to make the BBOX as close to the feature's actual extent as possible. The labeling process was conducted in a GIS software (for example, ArcMap). A multi-step data labeling and validation strategy is adopted. Graduate students with background in GIS and terrain analysis are first trained by domain experts from USGS on feature inspection and boundary determination. After the first round of labeling, the results are reviewed and cross-validated by domain experts. For features with semantic ambiguity, USGS's official cartographical mapping principles are followed to finalize the annotation. The resultant feature boundaries are saved as an Esri shapefile.

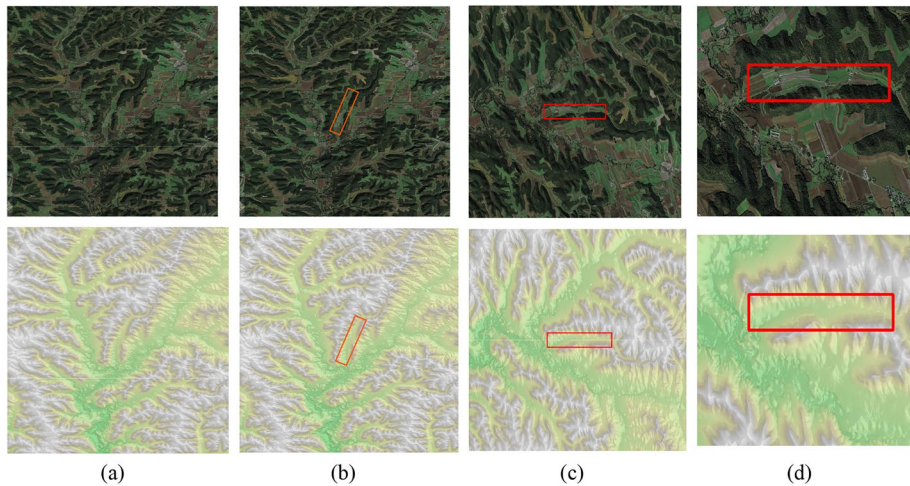


Fig. 3 An illustration of the processing steps for training data preparation. The scene shows Florin Valley, Wisconsin, which is a member of the GNIS class *valley*. The top row depicts the NAIP images and the bottom row shows the corresponding DEM covering the same extent. The columns illustrate an example of **a** the original NAIP data and the enriched DEM data after synthesizing elevation, slope, and hillshade, **b** the labeled results, **c** the resulting image and BBOX after rotation, and **d** the rescaled and translated feature and image ready for model training

Step 4 involves downloading data and preparing the training dataset. The USGS National Map’s TNMAccess API (<https://tnmaccess.nationalmap.gov/api/v1/>) is used to enable automated data downloading. We use the center of each BBOX to download the remote sensing images (NAIP) and the corresponding DEM data. Once the data are downloaded, the resolutions of the remote sensing imagery and the DEM are aligned by rescaling both onto a grid to be the same size. Specifically, bi-cubic resampling is used to up-sample the DEM data (10 m) and down-sample the remote sensing image (1 m) to the same image scale. Because the DEM data provide elevation information in numerical values, its value distribution is therefore very different from the optical remote sensing images, which consist of smoother color or texture changes. To adapt the DEM to the deep learning framework, a data enrichment strategy is performed to render the DEM into a color-scheme image. This image is fused with other terrain data, including slope and hillshade as grey-scale images, to reproduce a final elevation product as the final training image (Fig. 3a, bottom image).

Next, the images are rotated to render the BBOX vertical or horizontal relative to the image edges (Fig. 3b and c). The rotated images are then cut to the desired image size (Fig. 3d). To avoid missing data in the rotated scene, a large margin is chosen when downloading images. The length of any edge of the downloaded image scene is at least four times as long as the longer edge of the BBOX. Specifically, to rotate an image i , the angle α between one arbitrary edge l of the BBOX (see examples in Fig. 2b) and the horizontal edge of the image scene are first calculated. The horizontal edge is in parallel with the x axis of the geographic coordinate system (GCS) of the multi-source data. Next, a rotational matrix based on α is applied to the coordinates (x_{geo}, y_{geo}) of the vertices of the original BBOX to derive the new location (x'_{geo}, y'_{geo}) of the BBOX vertices in the rotated image. Mathematically, the transformation can be expressed as:

$$\begin{bmatrix} x'_{geo} \\ y'_{geo} \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_{geo} \\ y_{geo} \end{bmatrix}$$

with:

$$\alpha = \arctan\left(\frac{\Delta y_{geo}}{\Delta x_{geo}}\right)$$

where Δx_{geo} and Δy_{geo} indicates projection lengths of the selected edge l in the x and y axes in the GCS, respectively. After the image rotation, the BBOX is placed in the upright direction in the rotated image i' (Fig. 3c).

To further increase the diversity of the data, the feature is also moved out of the center of the image scene to be randomly placed within the image window. A small margin (10% of the image height/width) is reserved between the feature BBOX and the image edge to prevent information loss during the convolution for any part of the BBOX falling along the image edge. The feature relocation is accomplished by cropping the originally downloaded image scene (Fig. 3d), which is much larger (four times on each side) than the desired image size. We define the height and width of the training image i'_{train} to produce h' and w' . A randomization process determines the upper-left coordinates (x'_{ori}, y'_{ori}) of the image window ($h' \times w'$) used to crop the image scene i' . To meet the constraint of the preserved margin, the randomly generated point needs to satisfy the following conditions:

$$\begin{aligned} p_x - x'_{ori} &\geq 0.1w', & \forall p \in B \\ p_y - y'_{ori} &\geq 0.1h', & \forall p \in B \\ x'_{ori} + w' - p_x &\geq 0.1w', & \forall p \in B \\ y'_{ori} + h' - p_y &\geq 0.1h', & \forall p \in B \end{aligned}$$

where B is the collection of all the vertices $\{p\}$ of the feature BBOX. p_x and p_y indicate the coordinates of a vertex p . All of these coordinates use the same coordinate system (i.e., WGS84) as that used in the originally downloaded image. This transformation reduces location dependency in the training data. By increasing the diversity of relevant locations of features in an image scene, the model robustness and adaptability will also increase. This assumption has been validated through experiments in Section 4.

After the training images are generated, the label metadata are created. The transformation from geographic coordinates to image row and column indices can be expressed as follows:

$$\begin{bmatrix} i_{col} \\ i_{row} \end{bmatrix} = \begin{bmatrix} p_x & x'_{ori} \\ -p_y & -y'_{ori} \end{bmatrix} \begin{bmatrix} \frac{1}{R} \\ -\frac{1}{R} \end{bmatrix}$$

where (p_x, p_y) are the coordinates of a vertex of the feature BBOX to be transformed, (x'_{ori}, y'_{ori}) is the geographic coordinate of the upper-left point of the image scene, and R is the spatial resolution of the image. Figure 3d shows the resultant images after data processing.

4 GeoImageNet statistics

4.1 GeoImageNet: an illustration

Table 2 lists the feature type and number of image scenes containing different features in the GeoImageNet 2022 dataset. These features are selected because they are common and popular features for terrain research. In addition, we selected features of two main types: (1) features categorized as “TFe” (e: elevation), including basin, ridge and valley, and (2)

features categorized as “TFc” (c: color), including bay, island, and lake. “TFe” is conceptualized so as to categorize features that have more significant terrain changes, and therefore are more easily detected from elevation data. “TFc” is conceptualized so as to categorize features that reside in relatively flat terrains, and therefore are more easily detected from analyzing remote sensing color imagery. Regardless of their terrain characteristics, each feature is prepared with a pair of image scenes, a remote sensing NAIP image, and a color-rendered DEM covering the same geographical extent in the same image resolution and spatial reference system. There is some variety in the numbers of images for different feature classes because of availability of the high-resolution NAIP or DEM data in the selected area.

Figure 4 shows sample training images that contain pairs of NAIP images and color composite DEMs representing terrain features in the training dataset. The USGS definition of a basin is a “Natural depression or relatively low area enclosed by higher land (amphitheater, cirque, pit, sink).” Many basins are circular or oval in appearance. Common types of basins include drainage basins, structural basins, and ocean basins. For example, the fact that Mickey Basin, east of Steens Mountain in southeastern Oregon, is lower than the surrounding area is more evident in the DEM product than in the remote sensing image (Fig. 4a). USGS defines a ridge as an “Elevation with a narrow, elongated crest which can be part of a hill or mountain.” A ridge normally covers a large extent but appears as a linear feature because its top is narrow relative to its base. Lightning Ridge in the Uinta Mountain region of northeastern Utah is less discernible on the remote sensing image than on the DEM product, which clearly shows the high elevation along the ridgeline and west–east running direction (Fig. 4b). A valley is a “Linear depression in the Earth’s surface that generally slopes from one end to the other,” distinguishing it from the more rounded basin. Because valleys are normally found in mountainous regions, they also have more significant terrain changes. Florin Valley, a small valley that channels runoff to the Mississippi River in western Wisconsin, is also more visible in the DEM product than the remote sensing imagery (Fig. 4c).

Table 2 GeoImageNet feature type and count

Dataset	GeoImageNet	
Feature category	Feature	Image count
TFe	Basin	155
	Ridge	171
	Valley	181
TFc	Bay	93
	Island	106
	Lake	170
Description	Multi-source benchmark dataset which contains remote sensing imagery, DEM, and its deliverables (Slope and Hillshade data)	
Image size	636 × 594 (averaged)	
Applications	Object detection, natural feature detection	
Multi-source?	Yes	
Location information?	Yes	

TFe denotes terrain features that are more easily detected from elevation profile, whereas TFc signifies terrain features that can be better detected from color imagery, such as NAIP data

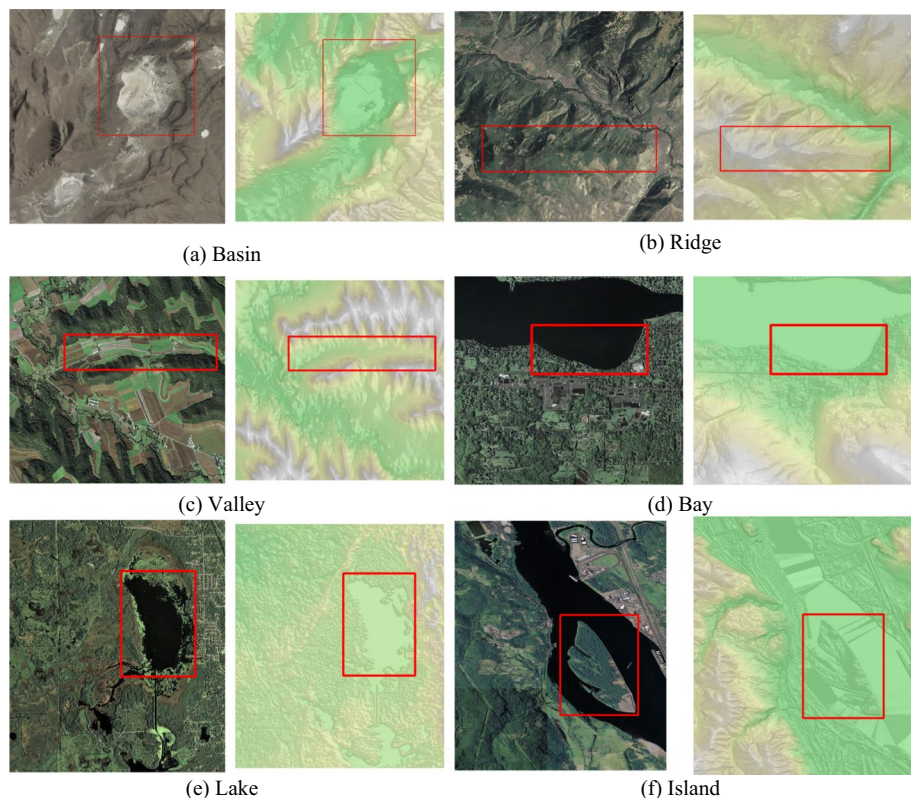


Fig. 4 Sample training images (Left: remote sensing NAIP image. Right: composited color-scheme DEM) with labels (in red BBOX) for the natural features. **a** Basin. **b** Ridge. **c** Valley. **d** Bay. **e** Lake. **f** Island

The remaining features—bay, lake, and island—are related to water. A bay is defined as an “indentation of a coastline or shoreline enclosing a part of a body of water; a body of water partly surrounded by land.” A bay is often demarcated by three land borders, with the remaining open to water. Burtis Bay, just southeast of Lake Erie in western New York, displays this common pattern. A lake, on the other hand, is a “natural body of inland water,” meaning it is surrounded by land, as in South Lake, which is just west of Titusville on the central east coast of Florida (Fig. 4e).” An island, as an “area of dry or relatively dry land surrounded by water or low wetland” can be conceptualized as the inverse of a lake. Sandy Island, OR, in the Columbia River near Longview, WA is classic example of an island (Fig. 4f). The latter three feature classes are more easily detected in the remote sensing imagery than in the DEM because of the strong contrast between water and land.

4.2 Geographical representativeness

Figure 5 shows the geographical distribution of selected features in the training dataset. Despite the geographically constrained feature sampling plan, some features classes are clustered in particular regions. This is unavoidable due to the environmental conditions required to host particular feature types. For example, lakes are more common in wet regions than dry; named basins fall

in the basin and range region of the western US; and ridges are limited mostly to mountainous regions. Nevertheless, the distance threshold applied during the feature selection process ensures the geographical representativeness of the dataset. For instance, these features are widely distributed in 44 states of the US, including Hawaii, but not in Alaska due to lack of a source of high-resolution data for that region. California and Utah host the most selected features, especially in the categories of basin, valley, and ridge. Bays and lakes are mostly found in the northeastern states, Washington, and Florida. Not surprisingly, islands are located in coastal and Great Lake states.

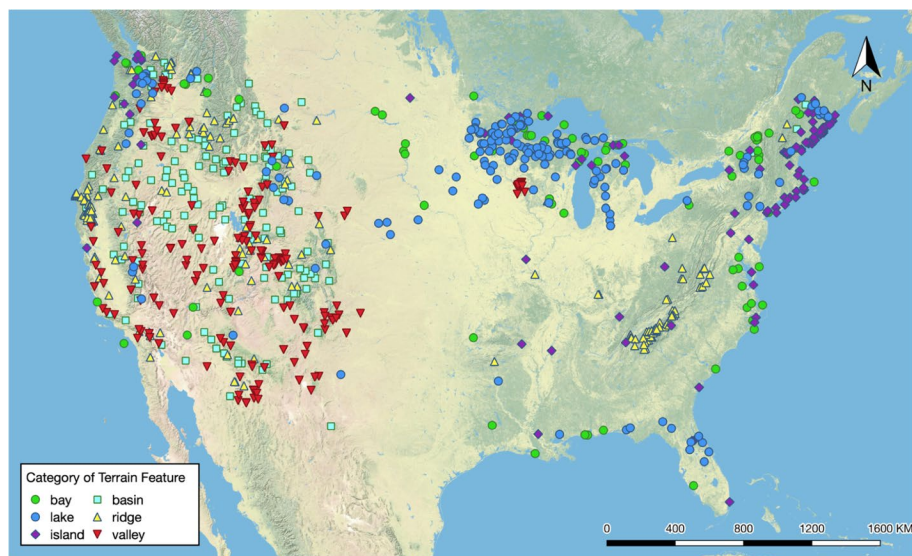
4.3 Dataset robustness

Developing a robust dataset is as important as developing a robust computational model. As the knowledge extracted by the model is largely dependent upon the input data dataset, it is critical to create a dataset that is diverse and representative to ensure model generalizability. The previous section reveals the geographic representativeness of the GeoImageNet. This section discusses additional important issues addressed when preparing the dataset: (1) randomness of relative feature location; (2) running direction of the BBOX labels, and (3) relevant feature scales regarding the image scene.

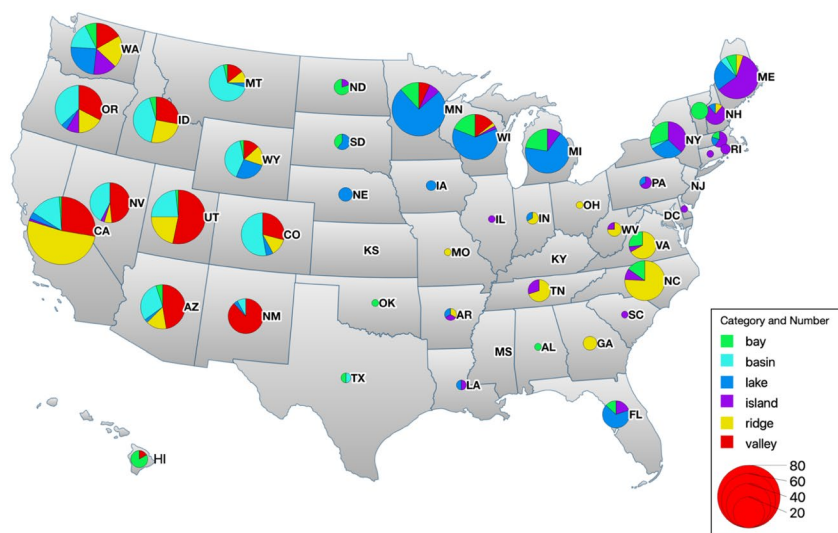
First, it has been increasingly recognized that neural network models are not shift-invariant [27], which means a small shift of the image may result in a dramatic change in results. The model shift-invariance issue is caused mainly by the popular max-pooling strategies used for down-sampling feature maps in the deep learning pipeline. Max-pooling, the strategy to select the maximal value in a sliding window will suppress low-pass frequencies, resulting in a bias in the sampling procedure. When it comes to object detection, the relative location of the labeled BBOX will influence a GeoAI model's predictability. For instance, when the BBOX is always located in the center of the image, or near the edge, the model may create a memory of this location characteristic and will predict a BBOX at those locations, resulting in a location bias in the knowledge gained. Addressing this issue requires careful preparation of the training dataset. Section 3.1 describes in detail on how the GeoImageNet is created. Experiments in Section 5 verify that such datasets will reduce model bias and improve model robustness in handling diverse datasets and prediction tasks.

Second, the horizontal or vertical placement of the BBOX may also affect dataset diversity. To make sure the labeled BBOX is as close to the actual feature extent as possible, it is drawn along the running direction of the feature. For instance, if a ridge runs southeast to northwest, the BBOX will be along that direction. However, because mainstream object detectors require an image edge-aligned BBOX as input, we rotated the BBOX (and the image) so that its edges are in parallel with the border of the image scene. In doing this, we created a random factor to allow the BBOX to be placed either horizontally (when the width is longer than the height) or vertically (the height is longer than the width) instead of only in one fixed direction (Fig. 4).

Third, scale – particularly the size of the target features relative to the entire image scene—is always an important parameter to consider in both terrain analysis and deep learning. Scale research has focused on multi-scale analysis to extract scale-dependent features for classification and object detection, and for small object detection leveraging image pyramids and feature fusion [28–30]. GeoImageNet addresses the second issue by preparing the feature at a recognizable scale regardless of its original size. This is because terrain feature detection has been a very challenging topic compared to the detection of built features, which is compounded by multi-sourcing learning. Therefore, in our 2022 GeoImageNet dataset, we control the relative ratio between the feature size and the image scene to allow both small and large features to be displayed at a detectable scale. This requires proper down-sampling of high-resolution images and up-sampling of low-resolution images, as well as multi-source image alignment in the rescaling process.



(a) Geographic distribution of individual features



(b) State-based pie chart showing terrain feature distribution

Fig. 5 Geographical distribution of training features in GeoImageNet

5 Experiments

5.1 Performance benchmark for object detection leveraging cutting-edge models

To enable detection of natural features leveraging multi-source data, we designed extensions to two popular deep learning-based object detection architectures, Faster-RCNN [31] and RetinaNet [32]. Object detection architectures can be categorized into two-stage

detectors which separate regional proposal generation (for BBOX prediction from object classification. Faster-RCNN is a representative object detection model and it has achieved both a faster detection speed and higher detection efficiency than its predecessors in the RCNN family. On the other hand, one-stage detectors accomplish object detection in a single pass, and models, such as YOLO (You Only Look Once; [33], typically have higher detection efficiency but lower accuracy than the two-stage models. With recent advances in RetinaNet, as a one-stage detector, it is reported to achieve a higher detection accuracy, as well as running efficiency on the Microsoft COCO dataset, due to taking advantage of its focal loss function and the adoption of a feature pyramid network [32].

All experiments utilized two 6-core CPUs at 1.90 GHz with 64 GB RAM and the Nvidia GeForce GTX980Ti GPU with 6 GB memory. The training learning rate was $1e-5$ utilizing an Adam optimizer. The data split was 80% for training and 20% for validation/testing.

The detection accuracy of the trained models, training efficiency, and how the models converge over time are examined on our dataset using the two detectors. In this experiment, both detectors take the remote sensing NAIP image and the DEM data as input and fuse the information from both sources at the feature (map)-level. For each detector, the two different data sources are fed to two identical CNN-based feature extractors running in parallel. The output feature maps are concatenated through an expansion of the third channel, followed by a dimension reduction layer which reduces the number of expanded channels by half such that the fused feature map can be fitted into the rest of the object detection network. To achieve better accuracy, several data augmentation techniques are adopted in both detectors, including rotation, image inversion, pepper and salt noise injection, and Gaussian noise injection [11].

The best-performing model of the RetinaNet detector has the higher mean Average Precision (mAP) in terms of predictive accuracy within the first 40 training epochs (Fig. 6a) and lower average time cost of processing one pair of color/DEM images (Fig. 6b). One RetinaNet epoch includes 2000 iterations/steps and Faster-RCNN epoch includes 10,000 iterations. mAP is a commonly used performance measure which evaluates predictive ability by combining both precision and recall. It has a value range between 0 and 1, with 1 the highest value. Both detectors achieve a testing mAP in a range between 0.7 to 0.8, indicating that 1) the dataset is well prepared and fits the task for training popular object detection models, and 2) there is still the possibility of gaining higher detection accuracy by developing enhanced models and training strategies. Our experiment also shows that RetinaNet runs twice as fast as Faster-RCNN, because of the reduced complexity of the one-stage model design. The two detectors converge smoothly as the training proceeds, leveraging our GeoImageNet dataset (Fig. 6b). The testing efficiency of both models increases sharply within the first 15 epochs. Thereafter, the accuracy fluctuates and slightly improves over time. The Faster-RCNN model's testing accuracy drops slightly on the last few epochs, but its training accuracy keeps increasing. This is an indication of overfitting. In comparison, RetinaNet shows stable performance due to the introduction of the focal loss function, which can effectively avoid easy cases dominating the learned weights in the model. Experimental results show that RetinaNet reaches its peak mAP at the 22nd epoch and Faster-RCNN gained the highest mAP at the 32nd epoch. From the results of this experiment, we suggest that training GeoImageNet for object detection tasks should generally be controlled within 40 epochs.

Figure 7 shows the breakdown of prediction accuracy of the two detectors. It can be seen that both detectors gain higher AP scores on TFc type of features, including bays, lakes, and islands than the TFe type of features, such as basins, ridges, and valleys (Fig. 7). This is because the TFc features have more obvious boundaries than the TFe features and

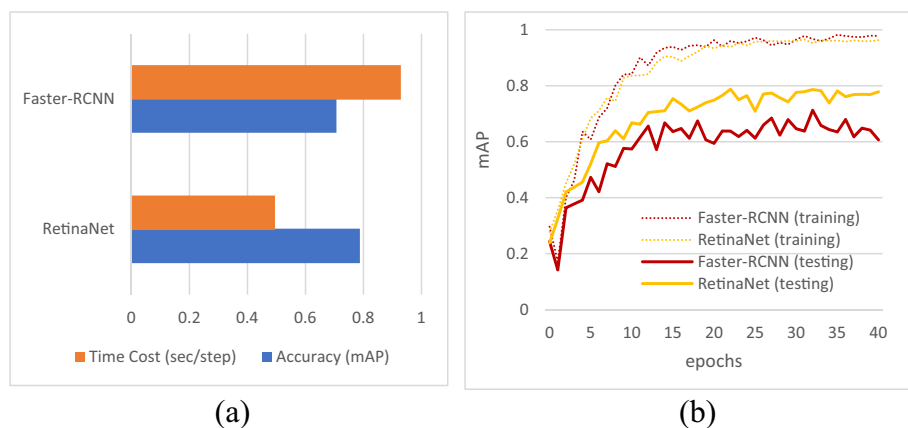


Fig. 6 Performance comparison between Faster-RCNN and RetinaNet. **a** Comparison of mAP and efficiency of the best models trained using both detectors. One step is one iteration that finishes processing one pair of remote sensing and DEM images. **b** Comparison of the two models on an mAP increase during model training, where Y-axis stands for mAP and X-axis stands for number of epochs. Because the iterations within each epoch of Faster-RCNN and RetinaNet are different (2000 for RetinaNet and 10,000 for Faster-RCNN), RetinaNet only uses 1/10 of the time of Faster-RCNN at the same epoch on the x axis. Dotted and solid lines show model training and prediction/testing accuracy, respectively

are usually more easily detected from the optical remote sensing imagery because of the high contrast between foreground (target terrain features) and the background, in terms of the color, texture, and brightness. However, the TFe features often have low contrast in optical imagery; they are deeply embedded and are often considered as the background of an image scene, rendering them difficult to recognize. The strategy of adding DEM data, which can better depict the TFe features by capturing their elevation changes, offers richer information to help better discern natural features. When cross comparing the performance of the two models, the results show that RetinaNet is superior than Faster-RCNN for the detection of most natural features (except valley), demonstrating its prominent performance. For valley prediction, RetinaNet shows slightly lower accuracy (0.07) than Faster-RCNN. Although it is difficult to examine what exactly caused this subtle difference due to the model complexity, we believe that this is at an acceptable level of uncertainty.

The next section further demonstrates the added value of the multi-source GeoImageNet to support object detection in a GeoAI model.

5.2 Performance comparison between the multi-source and single-source object detection

This section presents experiments that compare the performance of single source data (remote sensing imagery) and multi-source data (remote sensing imagery plus DEM) in support of natural feature detection. RetinaNet is used as the object detector in the experiments. The multi-source training follows the same configuration as that described in Section 5.1. Single-source training is conducted with the original RetinaNet model.

Both multi-source and single-source detectors converge smoothly (Fig. 8). Not surprisingly, more epochs are required to flatten the increasing curve of multi-source learning than the single-source version. The single-source RetinaNet model peaked at the 5th epoch,

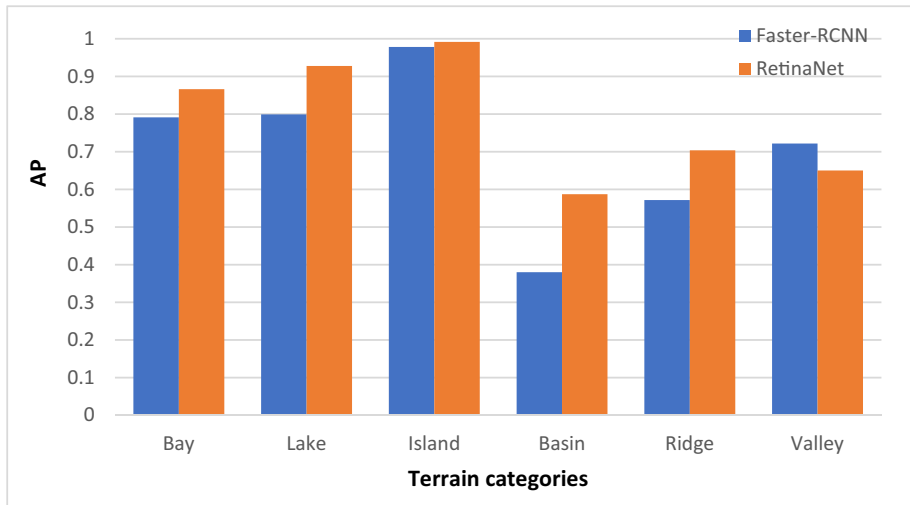


Fig. 7 Accuracy comparison between Faster-RCNN and RetinaNet on different terrain features. AP = Average Precision

as compared to the 21th epoch for the multi-source model. The additional training time is due to the complexity in processing and analyzing more data. Based on model architecture, the introduction of a new feature extractor for processing the DEM data also extends computation time. However, the results (mAP) clearly demonstrate the superiority of the multi-source GeoImageNet over single-source data in improving a GeoAI model’s predictive performance. The predictive accuracy increases from ~50% to nearly 80%, a dramatic improvement.

Accuracy drops across all tested terrain features when removing the DEM data from model training (Fig. 9), even, notably, for the TFc features, which rely heavily on the remote sensing imagery. Accuracy falls most significantly among basins, ridges, and valleys, as expected, in which the remote sensing images are not very effective when used alone for their detection. This experiment verifies the effectiveness of the multi-source data in GeoImageNet to support natural feature detection.

5.3 Location sensitivity of the trained model on datasets with different feature positioning strategies

In this section, we further examine the impact of relative locations of a target feature in the image scene on a model’s performance. According to our data processing workflow, the original image scene that contains a named feature is downloaded by setting the feature location as the image’s center location. However, always placing the target feature in the same location of an image in a training dataset may cause the GeoAI model to memorize the feature locations and be biased when inspecting images with a different feature distribution. Our assumption is that the strategy of locating the labeled targets may have an impact on the transferability of the corresponding trained model. Thus, a set of experiments is performed to examine such model sensitivity. By default, the data are downloaded centered around the labeled feature with a certain buffer (referred to as “center dataset”), i.e., each feature is in

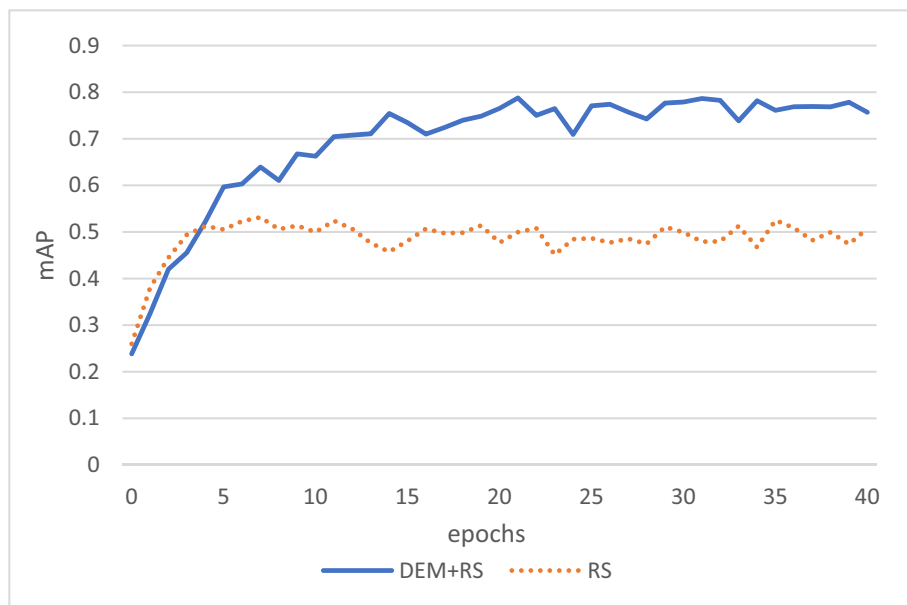


Fig. 8 Accuracy comparison of object detector trained using multi-source (remote sensing imagery + DEM) and single-source (remote sensing imagery only) input of GeoImageNet

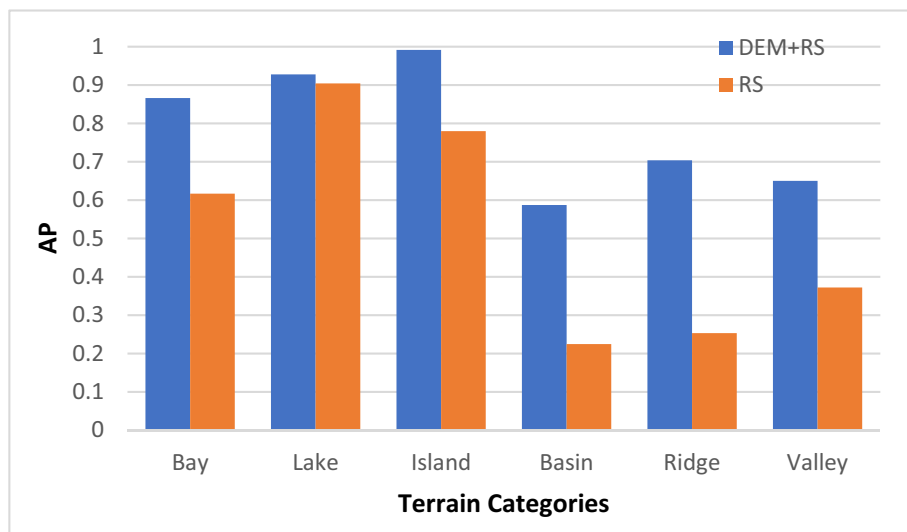


Fig. 9 Accuracy comparison between multi-source (remote sensing imagery + DEM) and single-source (remote sensing imagery only) on different terrain features

the center of the image window. Based on center dataset, we prepared another two datasets with different feature placement strategies: the edge dataset forces the target to locate near a random edge of the image scene with very small padding (i.e., 10% of the image size); and

the random dataset allows the targeted feature to locate anywhere within the image scene. The 10% padding is also applied in the random dataset. We also ensure that each target in the three datasets comprise the same proportion of the image scene. This consistency eliminates the impact of the size ratio between a target and its image scene during model training using different datasets. We randomly split each dataset into training and testing as a ratio of 8:2. The other two datasets then follow the same split. This treatment allows us to test three trained models across different testing sets in order to examine their robustness. We trained three models separately using the multi-source deep learning model based on RetinaNet with each of the three datasets, resulting in nine testing scenarios. The predictive accuracy (in mAP) of each of the nine scenarios is demonstrated in Fig. 10.

It can be observed that when training and testing use the same datasets with exactly the same feature-positioning configuration, the performances of the trained models in terms of testing mAP are very similar (Fig. 10, the first three columns from the left). The one trained and tested on the random dataset (third column from the left) shows a slightly lower mAP than the other two (first and second columns from the left). This is likely due to the greater (location) diversity in the random dataset than in the other two. But the difference is very small and negligible. From this, we can tell that regardless of the challenge level in these datasets (random dataset > edge dataset & center dataset), the trained models can all achieve convergence.

As shown in Fig. 10, the first group of results (left three columns) demonstrates that the model presents little sensitivity to the target location, if the consistency of target location between training set and the data for inferencing is ensured. However, the inferencing data in real applications can be arbitrary. The right group columns demonstrate the results when such consistency no longer apply. The testing scores of the models that are trained by the datasets with targets location in the center (the 4th and 5th columns from the left) or on the edge (the 6th and 7th columns from the left) of the image window decrease noticeably when testing on a different testing set. Those models show poor transferability. On the other hand, the three-chessboard filled green columns (model trained using the random dataset) show consistently higher mAPs when tested on different datasets. The results verify that the random dataset, which presents more location-diversity, is more robust than the center and edge datasets in tackling object detection problems.

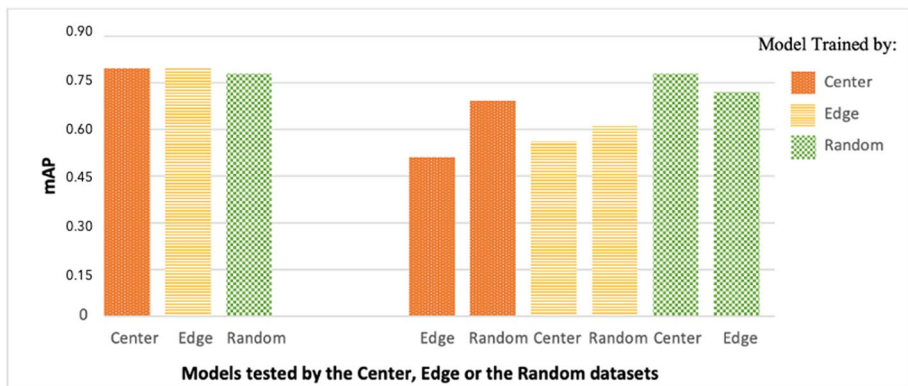


Fig. 10 Location sensitivity of the trained models on datasets with different feature positioning strategies. The dotted orange bars show the prediction mAP of the models trained on the center dataset and tested using different datasets. The horizontal line filled yellow bars show the mAP of the models trained on the edge dataset and the chessboard filled green columns show the mAP of the models trained on the random dataset

6 Conclusion

This paper introduces GeoImageNet, a first-of-its-kind multi-source dataset that supports terrain analysis and natural feature recognition. It addresses the increasing concern over the lack of diverse databases to advance the field of GeoAI and supervised machine learning [34]. Different from existing geospatial benchmark data, this new dataset combines color imagery and DEMs to offer richer spatial and contextual information for GeoAI modeling to achieve better detection accuracy. The availability of location information for each image scene offers great flexibility for geographic referencing and data expansion. Because of this, GeoImageNet is the first “geospatial” benchmark dataset for object detection and image analysis. The principles of geographic representativeness and scale and location sensitivity in preparation of geospatial training datasets are discussed. This dataset is evaluated using two popular and representative object detection models, Faster-RCNN and RetinaNet, and its validity is proved for aiding a GeoAI model to achieve convergence and satisfactory detection performance.

Looking forward, GeoImageNet should continue to be enriched by appending more feature types and more instances of each, as well as more data sources. We will test how the characteristics learned by the deep learning models using the natural features, primarily located in the US, can be adapted for feature detection in other countries. Deep domain adaptation techniques [35, 36] will be exploited and applied to ensure the terrain knowledge extracted from this work is, to the maximal extent possible, transferable and generalizable. We are also developing new visualization techniques to open-up the black-box of GeoAI models to better understand and explain the models’ reasoning processes.

The GeoImageNet data and model code will be openly accessible to the geospatial community. We hope to make this a community-driven effort toward developing more foundational datasets to support geospatial research and facilitate progress in GeoAI. The scale variety of the features should also increase in the training data and advanced model capabilities need to be developed to address small object detection [37]. The realm of GeoAI research can extend to the study of natural features on other planets and their moons, such as Mars and Phobos and Deimos.

Funding This work is supported in part by the National Science Foundation under awards 2120943, 1853864, and 2230034.

Data availability The datasets generated during and/or analysed during the current study will be made available in the GitHub repository, <https://github.com/ASUcicilab/GeoImageNet>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Disclaimer Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

References

1. Li K, Wan G, Cheng G, Meng L, Han J (2020) Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J Photogramm Remote Sens* 159:296–307
2. Li W, Batty M, Goodchild MF (2020) Real-time GIS for smart cities. *Int J Geogr Inf Sci* 34(2):311–324

3. Li W, Arundel ST (2022) GeoAI and the future of spatial analytics. New thinking in GIScience. Springer, Singapore, pp 151–158
4. Gahegan M (2020) Fourth paradigm GIScience? Prospects for automated discovery and explanation from data. *Int J Geogr Inf Sci* 34(1):1–21
5. Li W (2020) GeoAI: Where machine learning and big data converge in GIScience. *J Spat Inf Sci* 2020(20):71–77
6. Zhang C, Sargent I, Pan X, Li H, Gardiner A, Hare J, Atkinson PM (2019) Joint Deep Learning for land cover and land use classification. *Remote Sens Environ* 221:173–187
7. Demertzis K, Iliadis L, Pimenidis E (2021) Geo-AI to aid disaster response by memory-augmented deep reservoir computing. *Integr Comput Aided Eng* 28(4):383–398
8. Li W, Zhou B, Hsu CY, Li Y, Ren F (2017) Recognizing terrain features on terrestrial surface using a deep learning model: An example with crater detection. In: *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pp 33–36
9. Zhao B, Feng J, Wu X, Yan S (2017) A survey on deep learning-based fine-grained object classification and semantic segmentation. *Int J Autom Comput* 14(2):119–135
10. Torres RN, Fraternali P, Milani F et al (2020) Mountain summit detection with Deep Learning: evaluation and comparison with heuristic methods. *Appl Geomat* 12:225–246. <https://doi.org/10.1007/s12518-019-00295-2>
11. Li W, Hsu CY (2020) Automated terrain feature identification from remote sensing imagery: A deep learning approach. *Int J Geogr Inf Sci* 34(4):637–660
12. Hsu CY, Li W (2021) Learning from counting: Leveraging temporal classification for weakly supervised object localization and detection. *arXiv preprint arXiv:2103.04009*
13. Yang Y, Newsam S (2010) Bag -of-visual-words and spatial extensions for land-use classification. In: *Proc ACM SIGSPATIAL Int Conf Adv Geogr Inform Syst*, pp 270–279
14. Sheng G, Yang W, Xu T, Sun H (2012) High-resolution satellite scene classification using a sparse coding based multiple feature combination. *Int J Remote Sens* 33(8):2395–2412
15. Cheng G, Han J, Lu X (2017) Remote sensing image scene classification: Benchmark and state of the art. *Proc IEEE* 105(10):1865–1883
16. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, ..., Fei-Fei L (2015) Imagenet large scale visual recognition challenge. *Int J Comput Vision* 115(3):211–252
17. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. *Int J Comput Vision* 88(2):303–338
18. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, ..., Zitnick CL (2014) Microsoft coco: Common objects in context. In: *European conference on computer vision*. Springer, Cham, pp 740–755
19. Razakarivony S, Jurie F (2015) Vehicle detection in aerial imagery: A small target detection benchmark. *J Vis Commun Image Represent* 34:187–203
20. Liu K, Mattyus G (2015) Fast multiclass vehicle detection on aerial images. *IEEE Geosci Remote Sens Lett* 12(9):1938–1942
21. Xia GS, Bai X, Ding J, Zhu Z, Belongie S, Luo J, Datcu M, Pelillo M, Zhang L (2018) DOTA: A large-scale dataset for object detection in aerial images. In: *Proc IEEE Int Conf Comput Vision Pattern Recognit*, pp 3974–3983
22. Arundel ST, Li W, Wang S (2020) GeoNat v1. 0: A dataset for natural feature mapping with artificial intelligence and supervised learning. *Transactions in GIS* 24(3):556–572
23. Goodchild MF, Guo H, Annoni A, Bian L, De Bie K, Campbell F, ..., Woodgate P (2012) Next-generation digital earth. *Proc Natl Acad Sci* 109(28):11088–11094
24. Li W, Li L, Goodchild MF, Anselin L (2013) A geospatial cyberinfrastructure for urban economic analysis and spatial decision-making. *ISPRS Int J Geo Inf* 2(2):413–431
25. Tobler WR (1970) A computer movie simulating urban growth in the Detroit region. *Econ Geogr* 46(sup1):234–240
26. Arundel ST, Sinha G (2018) Validating the use of object-based image analysis to map commonly recognized landform features in the United States. *Cartogr Geogr Inf Sci* 46:441–455. <https://doi.org/10.1080/15230406.2018.1526652>
27. Azulay A, Weiss Y (2019) Why do deep convolutional networks generalize so poorly to small image transformations? *J Mach Learn Res* 20:1–25
28. Goodchild MF, Li W (2021) Replication across space and time must be weak in the social and environmental sciences. *Proc Natl Acad Sci* 118(35):e2015759118
29. Hsu CY, Li W, Wang S (2021) Knowledge-driven GeoAI: integrating spatial knowledge into multi-scale deep learning for Mars Crater detection. *Remote Sens* 13(11):2116
30. Wang S, Li W (2021) GeoAI in terrain analysis: enabling multi-source deep learning and data fusion for natural feature detection. *Comput Environ Urban Syst* 90:101715

31. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*, pp 91–99
32. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*, pp 2980–2988
33. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 779–788
34. Hu Y, Gao S, Lunga D, Li W, Newsam S, Bhaduri B (2019) GeoAI at ACM SIGSPATIAL: progress, challenges, and future directions. *SIGSPATIAL Special* 11(2):5–15
35. Wang M, Deng W (2018) Deep visual domain adaptation: A survey. *Neurocomputing* 312:135–153
36. Xu W, He J, Shu Y (2020) Transfer learning and deep domain adaptation. In: *Advances and Applications in Deep Learning*. IntechOpen. <https://doi.org/10.5772/intechopen.94072>
37. Tong K, Wu Y, Zhou F (2020) Recent advances in small object detection based on deep learning: A review. *Image Vis Comput* 97:103910

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.