Politecnico di Torino

Department of Control and Computer Engineering

Master degree in Data Science and Engineering

Internship report

Annotating Remote Sensing Images for Disaster Management using Foundation Models

Academic tutor:
Carlo MASONE
Company tutor:
Edoardo ARNAUDO

Intern:
Jacopo Lungo Vaschetti

October-January 2024

Contents

1 Summary

During my internship, I initiated the development of FMARS, a large satellite imagery dataset with automatic semantic segmentation masks of roads, buildings and high vegetation. The images are sourced from the Maxar Open Data program. GroundingDINO and SAM are the foundation models used for automatic labelling. Although the dataset remained incomplete by the end of my internship, we obtained promising results, making it worth completing the project.

2 Introduction

2.1 LINKS Foundation

The LINKS Foundation is a research and development hub founded by the Compagnia di San Paolo Foundation and the Polytechnic University of Turin. It operates in applied research, innovation and technology transfer. It acts as a bridge between basic research and the market. The foundation's main focus areas include Artificial Intelligence, connected systems and IoT, cybersecurity, advanced calculation systems, and satellite systems. All this is then applied to make innovative projects in many application fields: industry 4.0, Intelligent Mobility, Agritech, Space Economy, Smart infrastructures, Cultural Heritage. I interned in the Artificial Intelligence division.

2.2 The project

The purpose of the project is to leverage open-source foundation models to automatically label satellite images. Very-High Resolution (VHR) remote sensing (RS) imagery is increasingly accessible via open resources, but they often lack complementary annotations for effective machine learning applications. At the same time, recent foundation models like GroundingDINO and Segment Anything Model (SAM) provide unprecedented opportunities to exploit them to automatically generate annotations. Combining the two, we introduce FMARS (Foundation Model Annotations for Remote Sensing images), a dataset leveraging VHR imagery and foundation models for robust annotation.

3 FMARS

3.1 Data

We leveraged images provided by the Maxar Open Data program [6] since they are VHR and focus on natural disasters, including pre and post-event snapshots. Inspired by current state-of-the-art disaster management datasets [3], our dataset construction process emphasizes infrastructures, specifically buildings and roads, which are often the focus in post-event damage assessment. We incorporated Microsoft's open data sources to detect these features.

3.1.1 Images

Since we are focusing on fine-grained segmentation in the context of disaster management, VHR images are necessary. Data sources such as Copernicus Sentinel-2 [2] do not provide enough image content to

characterize objects of interest, such as buildings and roads. To this date, the largest source of disaster-related VHR imagery is represented by the Maxar Open Data Program [6]. This initiative provides pre- and post-event RGB images from more than 100 major crisis events since 2017 worldwide, with a total surface coverage of more than 2.6M km^2. Among these resources, we selected the 28 most recent events, from 2021 to 2023, as displayed in Table 1, summing up to an area of 431,077 km^2.

Event name	**Year**	**Area** (km^2)
Tonga volcano	2021	6,235.5
Indonesia volcano	2021	5,716.5
Marshall wildfires	2021	193.9
Sudan flooding	2022	693.1
Indonesia earthquake	2022	1,742.9
Kentucky flooding	2022	2,507.9
Hurricane Fiona	2022	4,204.6
Yellowstone flooding	2022	5,921.8
Ghana explosion	2022	4,114.0
Hurricane Ian	2022	78,162.5
Kalehe flooding	2022	341.1
Pakistan flooding	2022	48,251.3
Afghanistan earthquake	2022	7,195.5
Cyclone Emnati	2022	16,745.8
South Africa flooding	2022	20,017.6
Gambia flooding	2022	987.1
Italy (Emilia) flooding	2023	4,004.6
Libya floods	2023	2,350.7
India floods	2023	598.3
Georgia landslide	2023	197.6
New Zealand flooding	2023	481.5
Turkey earthquake	2023	37,037.9
Hurricane Idalia	2023	12,611.8
Cyclone Mocha	2023	3,763.8
Hawaii wildfires	2023	1956.6
Canada (BC) wildfires	2023	447.7
Canada (NWT) wildfires	2023	1,465.6
Morocco earthquake	2023	163,129.8

Table 1: List of events included in the FMARS dataset, including its year and total surface coverage derived from VHR imagery.

3.1.2 Buildings footprints

Regarding building detection, we considered two open data collections, from Microsoft and Google:

- Microsoft's Building Footprints[1]
- Google's Open Buildings[2][9]

In both datasets, polygons are generated by applying deep learning models on VHR satellite imagery.

Microsoft's Building Footprints: it encompasses 999M building footprint polygon geometries available in GeoJSON format, organized by a country-quadkey partitioning. Precision and recall metrics are

[1] `https://github.com/microsoft/GlobalMLBuildingFootprints`
[2] `https://sites.research.google/open-buildings`

available at continent/sub-continent level, with average values of 94.85% and 79.02% respectively. For some regions, confidence scores are available for each building. The images used for detection were taken from Bing Maps between 2014 and 2023 including Maxar, Airbus, and IGN France imagery.

Google's Open Buildings collection: This collection comprises 1.8 billion building detections, across an inference area of 58M km^2 within Africa, South Asia, South-East Asia, Latin America and the Caribbean. Each building in the collection is associated with a polygon, the Google's Plus Code [3] of the centre of the structure and a confidence score. Also, overall confidence scores for each region are provided. The images used in the creation of this collection are from Google Maps and the freshness of those varies from region to region.

3.1.3 Roads footprints

Concerning road detection we chose Microsoft's Road Detection collection[4]. It contains a total road length of 48.9M km. Also in this case, neural network and Bing Images were used for the mining process. The estimated precision is around 85.24%.

3.2 Foundation Models

Foundation models are large, general-purpose models that can adapt to different downstream tasks with minimal to no effort. For this project, we make use of models of this kind, such as GroundingDINO [5] and Segment Anything Model (SAM)[4]. These models can provide robust object detection and segmentation capabilities in a wide range of contexts without further fine-tuning. However, these solutions have often been employed in the context of natural images, with only a few attempts at applying them extensively on RS data at scale [10].

SAM and its derivatives are standard transformer-based segmentation networks that have been trained using promptable segmentation. In contrast with other segmentation objectives, this task receives two inputs: an image and a prompt. While the former is processed using a large and robust image encoder, the second is embedded into the decoder using a prompt encoder, and exploited as query by a lightweight mask decoder that produces segmentation masks. To resolve ambiguities, SAM can predict multiple outputs with its associated confidence for the same inputs. The prompts can be extremely flexible, ranging from sparse inputs such as a single point, a bounding box, or text, to dense arrays such as a binary mask. While points and boxes are encoded as simple positional embeddings, text is processed using off-the-shelf models such as CLIP [3], and masks are combined with the encoded image using a series of convolutions and element-wise sums.

GroundingDINO [5] is a zero-shot object detection model, built on top of the DINO [1] model. It introduces cross-modal fusion between a text prompt and an image to provide open-set object detection capabilities, using CLIP as text processor [8] and DINO with grounded pre-training as image encoder. Even if its output might be less accurate than human annotations, it allows us to detect objects, such as vegetation, not having a ground truth. In this way, we can use this model as a prompt generator for SAM to obtain the segmentation mask.

[3] `https://maps.google.com/pluscodes/`

[4] `https://github.com/microsoft/RoadDetections`

3.3 Methodology

3.3.1 Organising the data

Images - To obtain satellite imagery provided by Maxar we utilize the Python library LeafMap[5] which provides a convenient and efficient API for accessing these datasets. The images from this source are organized in *events*, which represent instances of natural disasters. An individual event comprises multiple *mosaics*, where each mosaic is a collection of *tiles* - each one is a tif file - capturing a defined geographical region within a brief span of time. For our research objectives, discerning pre and post-disaster mosaics is crucial. Therefore, since this division is not done at priori, we undertake a manual approach, comparing the two dates of the mosaic and the natural disaster for accurate categorisation. In our code, to better model these data and optimise shared heavy filter operations, we create the corresponding classes `Event` and `Mosaic`. When an `Event` is defined we store the coordinates of the rectangle that maximally inscribes all the mosaics it contains. This is useful to perform a first filtering on buildings and roads at event level, allowing us to work with much less irrelevant data at mosaic level, fastening all the subsequent filters.

Roads - Microsoft Roads are organised in large tsv files, each one representing an entire continent or just a part. Each row of the file contains the coordinates of the road - represented as a linestring - and the three-letter country code to which it belongs. We manually write a csv file to associate each event to the corresponding tsv file. At inference time, when defining an `Event`, we read the right tsv and filter based on its maximal boundaries. Subsequent filters are then applied to the desired level. We decided not to use the country code since it should have required manual assigning. Also, some events could be located across two countries. Using coordinates is overall faster and more precise.

Buildings - We use only Microsoft buildings. The dataset is organized as a big csv file containing the quadkey code and a link to another file that defines all the buildings in that quadkey. The first building filtering is done using these quadkeys. We build a function to easily switch from the coordinates of a rectangle to the codes of all the quadkeys it intersects. Subsequent filters are then performed using coordinates.

3.3.2 Choosing the right SAM model

During our testing, we calculated that SAM requires around 2.7 seconds to encode a 600×600 px image. Given the magnitude of data to be processed, it would be unfeasible to use this model, even employing batching and other parallelization techniques. For this reason, we searched for faster substitutes, without sacrificing too much the segmentation quality. The most promising models are FastSAM[12] and EfficientSAM[11] (ESAM). Both models drastically reduce the encoding time of the images. However, we prefer ESAM since it produces higher quality segmentation with respect to FastSAM, comparable to the original SAM.

3.3.3 Workflow

We aim to generate segmentation labels for three classes: buildings, roads, and high vegetation. We segment each tile independently of the others. A given tile is divided into square *patches* with edges size

[5] `https://leafmap.org/`

of 600 px. These are sampled in a grid fashion starting from the tile's bottom left corner, with a stride of 400 px (i.e. two patches on the same row overlap by a length of 200 px). Each patch is processed independently. First, the patch has to be encoded by the visual transformer of ESAM. Then, we gather all the prompts for trees and buildings in that patch.
Following previous works [7] and [10], we adopt box prompts as ESAM's inputs for our mask generation process. This also combines naturally with the inputs at our disposal, comprising open data sources for buildings (see Section 3.1.2), and box object detections derived from GroundingDINO for high vegetation. Detection boxes for buildings are given by the Ms Building Dataset as oriented bounding boxes (OBB). We convert them into axis-aligned bounding boxes and enlarge them to be sure that they contain the whole building, before passing them to the model.
ESAM accepts batches of images as well as batches of prompts. A prompt is composed by a `input_point` tensor and a `input_label` tensor. The `input_point` tensor is expected to have dimensions of `[batch_size, max_num_boxes, 2, 2]`. For this reason, for each batch we need to pad the `max_num_boxes` dimension to the number of boxes present in the batch's image that has the most. The `input_label` tensor is expected to have dimensions of `[batch_size, max_num_boxes, 2]`. Its last dimension contains `[2,3]` indicating the top left and the bottom right corner of the prompt bounding box.
Also, for each image, we need to store the indexes of boxes belonging to trees to distinguish the class of the masks. Indeed, the model returns a mask for each prompt (`max_num_boxes` dimension). Merging the masks belonging to the same class, we obtain a single tree mask and a single building mask. We repeat this procedure for all the patches in the tile. Then, collating all the patches' masks we end up having two complete, tile-level, masks, one for the tree class and the other for the building class. Subsequently, generating a buffer from the line string we form the road class segmentation mask. The buffer width must be chosen to be appropriate for the region (e.g. roads in Africa are generally narrower than in the USA). By construction, some pixels may be associated with multiple classes. When this is the case, we enforce the following priority order: buildings, trees and then roads. The masks of buildings are the most reliable. Those belonging to trees surely have some uncertainty but they are crucial in shaping the straight edges of the roads, obtained as line buffers. The complete annotation workflow is summarized in Figure 1.

3.3.4 GroundingDINO Workflow Details

For better performances, GroundingDINO requires some hyperparameter tuning. The hyperparameter to set are:

- *prompt phrase*: indicating what to detect
- *box_threshold*: the confidence that the box contains the object
- *text_threshold*: the similarity between the prompt and the object in the box

When using GroundingDINO to perform tree detection, we tested a few phrases as prompts such as: "tree", "trees", "vegetation" and others. The best results are obtained using "green trees". The default values for the *box_threshold* and *text_threshold* are 0.35 and 0.25 respectively. We look for a balance between precision and recall. We obtain good results lowering *box_threshold* to 0.15 and increasing *text_threshold* to 0.30. However, since the first parameter is so low, some undesired boxes are outputted. Especially in patches with diffuse vegetation, the model detects the whole image as a tree, leading to

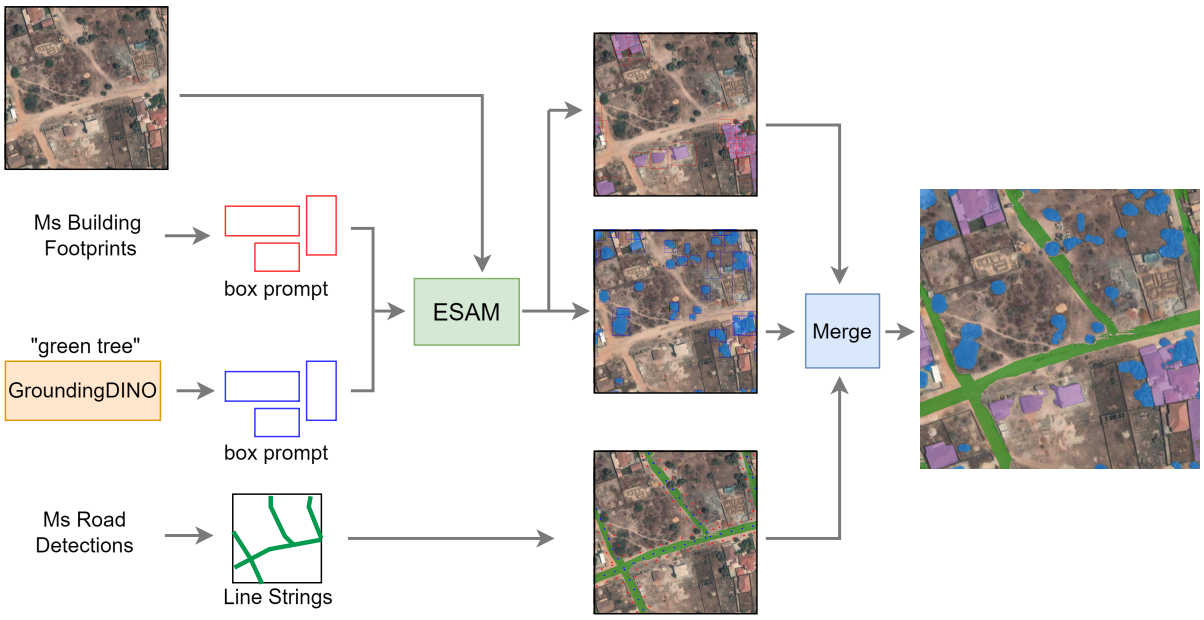

Figure 1: Annotation workflow adopted for the three selected classes. Buildings and trees are segmented via ESAM. Roads are derived from line string buffers. All the masks are merged into a single output.

catastrophic results when prompting that detection to SAM. To solve this problem, using an ecdf plot, we select a reasonable threshold (6000 mt^2) to exclude all the boxes with areas bigger than that.

3.4 Results

As anticipated, the whole dataset is not yet completed. However, we segment some tiles comprised in the `Gambia-flooding-8-11-2022` event. Processing a single tile of dimension 17408 × 17408 px takes around 50 minutes, with a batch size of 2, patch size of 600 px and stride of 400 px, running on a Nvidia GeForce GTX 1080 Ti. On average, a patch of the aforementioned sizes requires around 1.61 seconds to be processed. The most expensive step is the image encoding by the ESAM model. It takes 80% of the whole processing time. Fig. 2 and fig 3 show some qualitative results obtained in our experiments. In particular, fig. 2 are single patches of 600×600 px sampled at runtime. Meanwhile fig. 3 depicts a larger area extracted from the whole tile. Overall, we are satisfied with the outcome, even if there is always room for improvement. For example, it can be seen that sometimes trees are wrongly assigned to the building class or that small random areas are assigned to the tree class.

3.5 FMARS Conclusion

Combining vision foundation models and open data in a novel processing pipeline, we started the creation of FMARS: a remote sensing dataset constructed by VHR imagery and accurate annotations of relevant elements in the context of disaster management, namely buildings, roads, and high vegetation. When completed it will include 28 crisis events and encompass more than $70M$ annotations over a surface of over 200.000 km^2. It could be effectively used for different disaster-related use cases, including damage identification and assessment. An additional work could be converting from semantic segmentation to instance segmentation. This would not be difficult since the masks returned by the model are already

Figure 2: Four patches 600 × 600 px of the event `Gambia-flooding-8-11-2022` displaying the extracted labels for buildings (purple), roads (green) and trees (blue).

Figure 3: Large sample extracted from a segmented tile belonging to the `Gambia-flooding-8-11-2022` event. Buildings are light blue, roads are yellow and trees are red

separated for each detection. Also, despite its size, the dataset only contains three classes. Future works may address this limitation by iteratively adding labels repeating the full annotation workflow with additional categories.

4 Conclusions

This internship at LINKS has been a valuable experience that has broadened my understanding and capabilities of working with remote sensing images and foundation models. I had the opportunity to apply the knowledge and skills I acquired in my course of study in a real-world setting, which has been a great learning opportunity. In terms of specific skills, I have sharpened my capabilities in organizing code for a large project. I also learnt to handle and organize large georeferenced images, visualizing them in ad hoc software like QGIS[6]. I understood how to work with projected coordinates as well as geographic ones. I learned to work with vectorized shapes. I deepen my knowledge of semantic segmentation and zero-shot object detection, especially using SAM and GroundingDINO. Additionally, I have had the chance to collaborate closely with Edoardo and his team, experienced professionals who have mentored me throughout the internship. They have been a valuable resource and I thank them for providing guidance, answering questions and sharing their knowledge on the field.

[6] `https://qgis.org/en/site/`

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[2] Copernicus. Copernicus Programme. `https://www.copernicus.eu`. Accessed: 2024-01-05.

[3] Ritwik Gupta, Bryce Goodman, Nirav Patel, Ricky Hosfelt, Sandra Sajeev, Eric Heim, Jigar Doshi, Keane Lucas, Howie Choset, and Matthew Gaston. Creating xbd: A dataset for assessing building damage from satellite imagery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 10–17, 2019.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[5] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[6] Maxar. Maxar Open Data Program. `https://www.maxar.com/open-data`. Accessed: 2024-01-05.

[7] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[9] Wojciech Sirko, Sergii Kashubin, Marvin Ritter, Abigail Annkah, Yasser Salah Eddine Bouchareb, Yann Dauphin, Daniel Keysers, Maxim Neumann, Moustapha Cisse, and John Quinn. Continental-scale building detection from high resolution satellite imagery, 2021.

[10] Di Wang, Jing Zhang, Bo Du, Minqiang Xu, Lin Liu, Dacheng Tao, and Liangpei Zhang. Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[11] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv:2312.00863*, 2023.

[12] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023.