Artificial Intelligence in Geosciences 6 (2025) 100105

Contents lists available at ScienceDirect

Artificial Intelligence in Geosciences

journal homepage: www.keaipublishing.com/en/journals/artificial-intelligence-in-geosciences

Original research articles

Optimizing zero-shot text-based segmentation of remote sensing imagery using SAM and Grounding DINO

Mohanad Diab [a], Polychronis Kolokoussis [b], Maria Antonia Brovelli [a,*]

[a] Department of Civil and Environmental Engineering, Politecnico di Milano, Milano, 20133, Italy
[b] School of Rural, Surveying and Geoinformatics Engineering, National Technical University of Athens, Athens, 15780, Greece

ARTICLE INFO

Keywords:
Foundation models
Multi-modal models
Vision language models
Semantic segmentation
Segment anything model
Earth observation
Remote sensing

ABSTRACT

The use of AI technologies in remote sensing (RS) tasks has been the focus of many individuals in both the professional and academic domains. Having more accessible interfaces and tools that allow people of little or no experience to intuitively interact with RS data of multiple formats is a potential provided by this integration. However, the use of AI and AI agents to help automate RS-related tasks is still in its infancy stage, with some frameworks and interfaces built on top of well-known vision language models (VLM) such as GPT-4, segment anything model (SAM), and grounding DINO. These tools do promise and draw guidelines on the potentials and limitations of existing solutions concerning the use of said models. In this work, the state of the art AI foundation models (FM) are reviewed and used in a multi-modal manner to ingest RS imagery input and perform zero-shot object detection using natural language. The natural language input is then used to define the classes or labels the model should look for, then, both inputs are fed to the pipeline. The pipeline presented in this work makes up for the shortcomings of the general knowledge FMs by stacking pre-processing and post-processing applications on top of the FMs; these applications include tiling to produce uniform patches of the original image for faster detection, outlier rejection of redundant bounding boxes using statistical and machine learning methods. The pipeline was tested with UAV, aerial and satellite images taken over multiple areas. The accuracy for the semantic segmentation showed improvement from the original 64% to approximately 80%–99% by utilizing the pipeline and techniques proposed in this work. **GitHub Repository:** MohanadDiab/LangRS.

1. Introduction

With the advancements in the fields of remote sensing (RS) and computer science (CS), RS images have been generated at an accelerated rate, which increased the possibilities of enhanced earth surface monitoring, which in turn facilitated the birth of RS big data (Ma et al., 2015; Li et al., 2016). This explosion of data sources and enhancements in data quality posed the question of automation possibilities of the interpretation and analysis of RS big data (Chi et al., 2016). This attracted the attention of researchers to incorporate computer vision techniques on RS images, which aims to assign labels (e.g., tree, water) to RS images in a classification, to a region denoted in a bounding box (object detection), or the labeling of individual pixels to classes (semantic segmentation) using Artificial Intelligence (AI) methods (Li et al., 2018; Kampffmeyer et al., 2016), due to their wide range of usage in crop assessment (Kussul et al., 2017; Ozdarici-Ok et al., 2015), environmental monitoring (Yu et al., 2018), intelligent traffic (Zhang et al., 2018), urban planning (Shi et al., 2015; Kampffmeyer et al., 2016), etc.

Along with the advancement of the visual prowess of AI models, natural language processing (NLP) models have also been growing rapidly in parallel. Advances in the field gave birth to attention mechanisms and transformers (Vaswani et al., 2017), which paved the way for powerful NLP models such as GPT (OpenAI et al., 2024) and Llama (Touvron et al., 2023). The transformer architecture inspired researchers in the computer vision field to adopt this methodology and develop visual transformers (ViT) (Wu et al., 2019), which in turn led to powerful models capable of storing vast amounts of innate knowledge (Khan et al., 2022).

The evolution of AI model architectures, which allowed them to grow larger to store more knowledge linearly, along with the exponential growth in computing power, enabled researchers to build general-knowledge models known as Foundation Models (FM). These models are trained in a purpose-agnostic manner on vast amounts of data (Bommasani et al., 2021), with examples including GPT-4 (OpenAI

* Correspondence to: Politecnico di Milano, Department of Civil and Environmental Engineering, Italy.
E-mail address: maria.brovelli@polimi.it (M.A. Brovelli).

https://doi.org/10.1016/j.aiig.2025.100105
Received 18 September 2024; Received in revised form 22 December 2024; Accepted 7 January 2025
Available online 13 February 2025

et al., 2024) and the Segment Anything Model (SAM) (Kirillov et al., 2023). They have shown capabilities applicable across multiple domains such as the medical field (Li et al., 2024), environmental monitoring (Biswas, 2023), and notably, the RS field (Osco et al., 2023a).

However, since these models are designed to handle large volumes of data across various domains, they are considered jacks of all trades but masters of none. They sometimes struggle with zero-shot inference, which refers to models processing data not seen during training, leading to potential issues with the quality, accuracy, or relevance of their responses (Kocoń et al., 2007; Yao et al., 2023; Xu et al., 2024).

For this reason, specialized FMs have emerged. With the RS field producing large amounts of data, FMs trained on multi-band satellite imagery, such as Prithvi-100M (Jakubik et al., 2023), have been introduced. These models, adopting a ViT architecture, have demonstrated adaptability by fine-tuning to specialized fields like flood detection and burnt area segmentation.

Considering this history, this paper focuses on implementing FMs in RS pipelines to explore their ability to automate processes fully or partially with high accuracy. For instance, the work presented in Osco et al. (2023b) showcases segmenting RS imagery using vision–language models (VLMs) in zero-shot and one-shot settings, drawing inspiration from Liu et al. (2023a), which suggests using the Grounding Dino open-set detection model to generate bounding boxes. These boxes are then used as input prompts for the SAM model.

Fine-tuning FMs to create specialized models requires substantial amounts of annotated training data and computational resources, which can be challenging for individual researchers and small businesses to afford (Patil and Gudivada, 2024). Therefore, this paper proposes pipelines that build applications on top of general-use FMs within RS pipelines, making them efficient in an off-the-shelf manner. The main contributions of this paper can be summarized as follows:

- This paper proposes the novel use of specialized applications built on top of multiple multi-modal models within RS segmentation pipelines.
- Supplement the zero-shot inference shortcomings of FMs with general pre-processing and post-processing methodologies based on local statistical indices and machine learning algorithms.

The pipelines developed in this paper are mostly automated and require only the input satellite imagery, natural language prompts and a few additional parameters. In this manner, these pipelines would enable users of little or no experience to successfully perform RS tasks such as object detection and segmentation with high accuracy and proficiency.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the methodologies. Section 4 reports the results. Finally, the conclusion and potential future work directions are presented in Section 5.

2. Related work

In this section, a review of some related works from the fields of visual, language, and multi-modal models and their applications in RS fields is presented.

Computer vision has come a long way since its inception, with the introduction of convolutional neural networks (CNN) and advancements in their architectures and applications (Krizhevsky et al., May 2012; Dosovitskiy et al., 2020; He et al., 2016). Visual Transformers (ViTs) were then introduced, paving the way for larger and more robust models (Han et al., 2023). Subsequently, powerful vision foundation models (FMs) such as Grounding Dino (Liu et al., 2023b) and SAM (Kirillov et al., 2023) were introduced as state-of-the-art models upon their release. The integration of computer vision in RS fields evolved rapidly from being a theoretical concept to becoming standard practice, performing tasks such as clustering, segmentation, and labeling (Ball et al., 2017).

For our specific application, we are interested in the use of vision FMs for RS images, exemplified by the work of researchers in Osco et al. (2023b), who demonstrated satisfactory results using SAM on various RS data sources, including satellite, drone, and aerial imagery. They presented their findings on performing zero-shot and one-shot inference on these RS images.

Transitioning from vision to language models, this field has experienced a significant research surge over the past decade. The development of models and tools capable of generating accurate outputs based on natural language prompting has enabled researchers from various domains to integrate general-knowledge language FMs into their fields (Radford et al., 2019; Brown et al., 2020). Researchers in the RS field have similarly introduced innovative methods to integrate large language models (LLMs) into RS domains, such as location extraction from text (Mehta et al., 2023) and geographical question answering from databases (Bhorge et al., 2023).

The development of vision and language models eventually converged, giving rise to multi-modal models known as vision–language models (VLMs). These models can process inputs of both images and natural language, producing outputs accordingly. Models such as GPT-4V (OpenAI et al., 2024) and Gemini (Team et al., 2024) have demonstrated their ability to provide feedback on images through a Q&A modality. However, performance varies depending on the domain, image complexity, and prompts. Consequently, researchers in the geospatial domain have been developing applications using multi-modal models and evaluating their capabilities for tasks such as mapping from street-level images (Juhász et al., 2023) and visual Q&A for satellite imagery (Bhorge et al., 2023; Chen et al., 2024).

The successful integration of FMs into RS has led researchers to focus on building geospatial models, supported by vision papers and opportunity-risk evaluations (Mai et al., 2023; Fernandez and Dube, Oct. 2023; Rao et al., 2023). Nevertheless, the practical implementation is still in its early stages, underscoring the need for methodologies to incorporate established FMs into RS. For instance, researchers in Li and Ning (2023) developed a tool integrating the GPT model into geospatial data processing tasks, providing visual and language solutions for queries based on natural language.

3. Materials & methodology

In this section, we present the technologies and methods used to build an aerial image semantic segmentation pipeline that is promptable via natural language, fully automated, and requires no fine-tuning whatsoever from the user by using FMs in an off-the-shelf manner. The structure of this section is as follows; the first sub-section presents the datasets used. The limitations and shortcomings of the utilized models are discussed in Section 3.2, and relevant solutions are proposed. Then, our full pipeline, designed to overcome the mentioned shortcomings, is presented in Section 3.3.

3.1. Materials

The datasets used in this study consist of satellite and aerial imagery with 5 to 60 cm spatial resolution over two different areas in Greece and one in the UK. The images cover multiple unique features and areas. The first area is a small region of Anavyssos, in Attica, Greece, for which one aerial and one satellite image have been used. The aerial image is an RGB image that has a spatial resolution of 25 cm, while the satellite image is also an RGB image acquired from the Quick-Bird satellite with a spatial resolution of 60 cm over the same region. The second area is a part of Kala Nera, a small residential area near the town of Volos in Greece. The area is depicted in a RGB aerial image that has a spatial resolution of 12.5 cm. Finally, the third area is a residential block in the South of Manchester, UK, captured using an unmanned aerial vehicle (UAV). In this case, the aerial RGB image has a spatial resolution of 5 cm (Reading, 2024). The results of the pipeline for all the input imagery are presented in detail in Section 4.

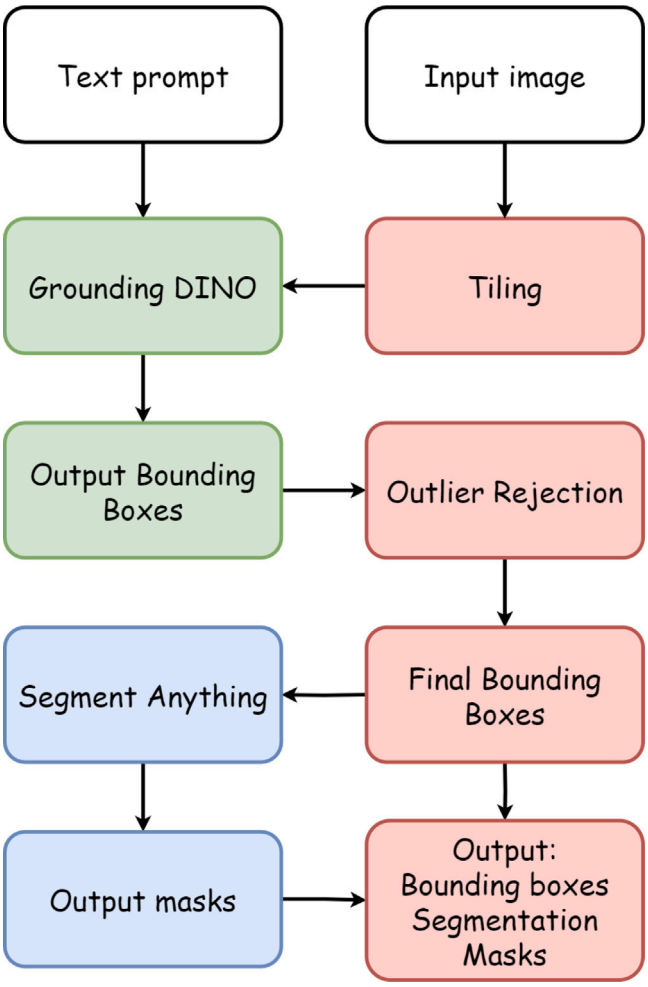

Fig. 1. High-level pipeline for the semantic segmentation of RS imagery. Boxes in green represent the DINO model and its output, the ones in blue represent the SAM model and its output, the pink ones represent our suggested additions to the pipeline.

3.2. Utilized model limitations & suggested solutions

This sub-section outlines a high-level pipeline formulation for performing semantic segmentation of RS imagery utilizing the segment anything model (SAM) and grounding DINO through a VLM. The most important part is to ensure the automation capabilities and the multi-modal nature of this pipeline to overcome the limitations of the utilized models, thus improving the zero-shot performance for current common practices. The pipeline shall be designed in a way such that it can produce output from multi-modal input. The input of the model should be designed to take both an image and a textual representation of the desired class/ classes to be segmented. Its output should provide the output masks/ bounding boxes of the input image. To achieve the goal at hand, a high-level pipeline is presented in Fig. 1.

The proposed pipeline showcases the general workflow, where the image and the prompt are passed as inputs. Go After going through the pre-processing phase (tiling process). After that, the object detection takes place. Then, the outputs preliminary results are post-processed, and the image is segmented producing the object masks using as input the bounding boxes, generated from the initial object detection phase, are filtered, thus eliminating wrong detections.

The experimentation for problem solving and fine-tuning of the pipeline has been carried out mainly using the aerial image of Anavyssos.

3.2.1. Tiling

The multi-modal models used in this paper were initially designed to handle input of everyday pictures. Grounding DINO was tested and benchmarked on datasets such as COCO Zhang et al. (2018) and proved to be one of the best models to perform open set object detection. But, when feeding these models with images of larger than usual sizes, such as satellite or aerial imagery, they require much more memory and computational power. As such, when testing the models with RS imagery, the system would usually crash or run out of memory. Even when the system does not crash, since large RS images usually contain multiple features at different granularities, it is hard for a model to be able to detect these features at different sizes and resolutions.

Therefore appropriate pre-processing (tiling process) is necessary. The pre-processing consists of loading the input image into a 2D array of size $H \times W$. Then, a square sliding window of size $S \times S$ is defined with an overlap value of O, where $O \leq S \leq H, W$. This window will slide over the image and object detection will be performed for every $S \times S$ part of the image using the Grounding DINO model. This method improves the object detection performance and the overall quality of the process since it does not violate any memory constraints in the system by breaking the input image into smaller parts manageable by the system. Once the object detection is done, the output bounding boxes from each window are joined in one tensor representing the bounding boxes covering the entire image.

The window can be of any size, but from empirical testing of the full pipeline, a recommended window size is 500 × 500 pixels. However, this value is subjective and depends on the size of the desired feature in the satellite image and should be adapted to the task at hand. Overlap is necessary to avoid "cutoff" and double counting of detected features. The overlap is recommended to be set as 20%–40% of the size of the window. These values were obtained from testing on RS images with a spatial resolution ranging from 5–60 cm.

3.2.2. Hyperparameter auto-adjustment

In this step, the tiles resulting from the sliding window process in the tiling operation are fed to the Grounding DINO model, with the input prompt specified in the input. The model usually requires 2 parameters (text_threshold, box_threshold) to be set, which control the granularity of the object detection ranging from 0–1 and representing the relevant confidence levels. When the thresholds are set to zero, the algorithm will segment everything without any rejection, when set to one, the model will only segment features that have 100% confidence based on the log-probabilities (logits) for the feature specified in the input prompt.

For our purposes, two methods have been tested for choosing the initial parameter values. The first method is to allow the user to manually specify a bounding box covering the smallest desired feature which from now on will be called the target_box. The second method does not require the manual intervention of the user. Instead, it sets the parameters to very low values, thus over-segmenting the image, and then filters out the wrong detection using the outlier detection methods described in the next section.

For the first method, the goal of the algorithm is to iteratively reduce the values of the parameters until the target box is captured. The model will run the inference starting from the values of 0.5 for both parameters, producing the output bounding boxes, if one of the output bounding boxes captures the target_box it will exit the loop and save the parameters as best parameters Fig. 2. If the output does not contain a bounding box representing the target box, the algorithm will continuously decrease the values of the parameters by 0.1 up to 0.1, then by 0.025 up to 0 until a successful result is achieved (capturing the target box).

When the values of the hyperparameters are underestimated, meaning that values are higher than the optimum values, the model will not be able to capture smaller features as illustrated in Fig. 3. Inversely, when the threshold values are overestimated, meaning that they are

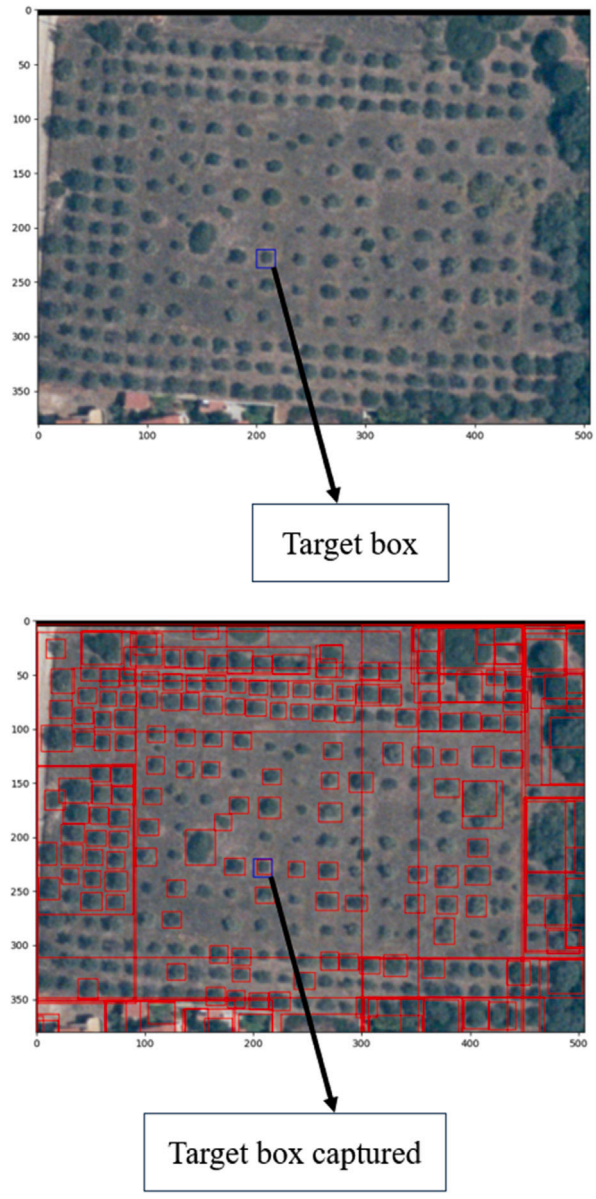

Fig. 2. Hyperparameter tuning method using a target box, convergence at 0.1 threshold (bottom).

smaller than the optimum values, they will over-segment and detect features not relevant to the problem. This means that in a way the model begins to somewhat "overfit" the object detection and starts placing bounding boxes over areas that already cover multiple features, creating bounding boxes that cover other bounding boxes, which are considered falsely placed bounding boxes as shown in Fig. 4.

To navigate through this issue, a second hyperparameter setting method is proposed, which allows the model to estimate the hyperparameters by setting low hyperparameter values from the get-go, leading to an over-estimated object detection as the one seen in Fig. 4, (in this case tree detection), and then; post-process the resulting bounding boxes, to reject the bounding boxes of no relevance to the desired solution.

The overestimation of the bounding boxes is a temporary solution that ensures that even the smallest of the desirable instances are detected using the model, the next sub-section discusses the methodology used to filter the desirable bounding boxes from the undesirable ones.

Fig. 3. Grounding DINO object detection with 0.5 hyperparameter values for the prompt "trees".

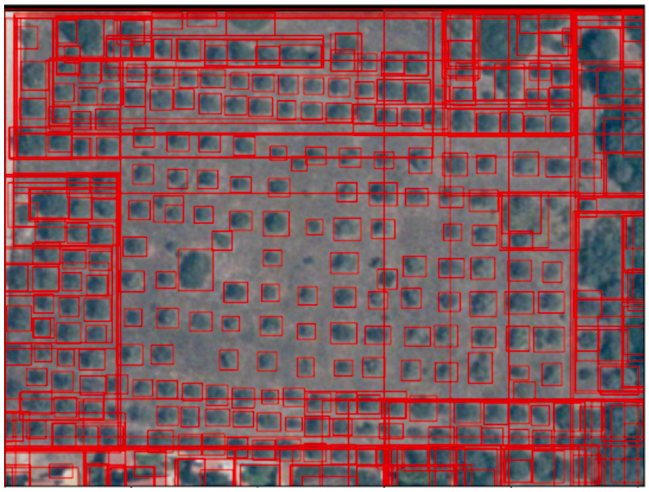

Fig. 4. Grounding DINO object detection with 0.1 hyperparameter values for the prompt "trees".

This second hyperparameter setting method proved to provide much better results than the first one and thus it has been adopted in the proposed pipeline. Therefore, the following steps utilize this automatic over-estimation and rejection method.

3.2.3. *Outlier rejection & segmentation*

Following the object detection phase, the model returns a tensor of bounding boxes, the tensor contains the pixel coordinates of each bounding box in the format (x1, y1, x2, y2). Exploiting this information, and referring to the results presented in Fig. 4, we can hypothesize that by visually inspecting the image, most of the bounding boxes with relatively smaller extent are correctly detecting the desired instances of "trees", while on the other hand, the relatively larger bounding boxes are covering multiple already-detected "trees".

To evaluate this hypothesis, we start with the detection example presented in Fig. 4, and apply anomaly rejection algorithms as post-processing techniques to reject bounding boxes of larger sizes. Then, the accuracies of the original segmentation and the segmentation after each post-processing method are compared.

First, the tensor of the bounding boxes is processed and indexed to return the area of each bounding box and its respective index based on its dimensions and order in the tensor respectively. The area

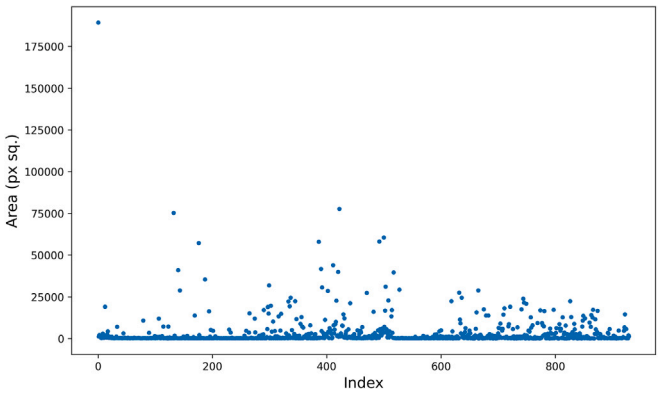

Fig. 5. Bounding box area distribution.

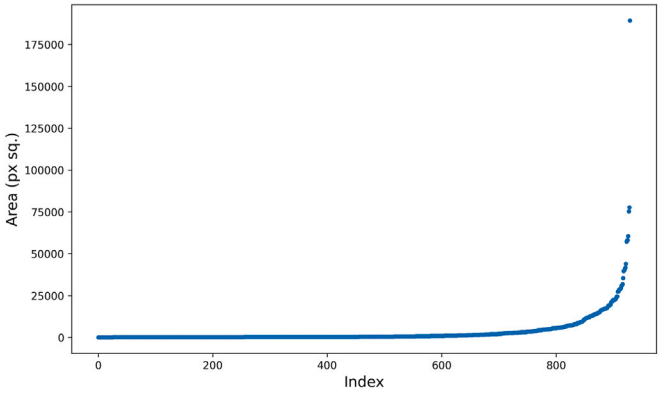

Fig. 6. Bounding box area distribution (ordered by size).

distribution of the indexed bounding boxes for the object detection performed in Fig. 4 is illustrated in Fig. 5. By analyzing the figure visually, we can see the dominance of the values at the bottom of the plot, which in theory represent the bounding boxes encapsulating the single features representing "trees", while as we go higher, the density of the points becomes apparently less dominant.

These boxes are the "outliers" that encapsulate areas containing multiple features. The bounding boxes are then ordered by size and plotted in Fig. 6. This figure shows the ordered distribution of the areas of the bounding boxes, as discussed earlier. Our scope is to reject the bounding boxes of exploding sizes materialized as the exponentially rising tail of the distribution in Fig. 6.

Multiple statistical and machine learning algorithms have been tested on the dataset for their effectiveness in differentiating the inliers from the outliers. These methods were:

- Z-score outliers (threshold = 3)
- IQR outliers (75th interquartile)
- Robust covariance (elliptic envelope)
  - Contamination = 0.1–0.5
- One-class SVM
  - Nu = 0.1
  - Kernel = "rbf"
  - Gamma = 0.1
- One-class SVM with SGD
  - Nu = 0.1
  - Kernel = "linear"

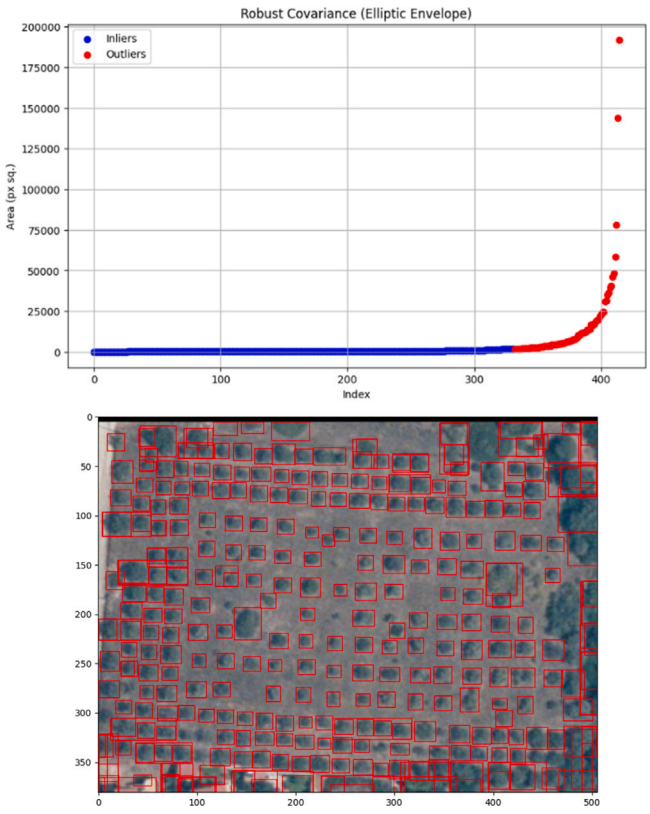

Fig. 7. Outlier rejection visualized.

- Isolation forest
  - Contamination = 0.1–0.5
  - Random_state = 42
- Local outlier factor (LOF)
  - N_neighbors = 20
  - Contamination = 0.1–0.5

The next step was to evaluate the performance and robustness of each outlier rejection method. To evaluate the performance, a confusion matrix was calculated for each method, and from every confusion matrix, the accuracy, precision, recall, and f1 score were calculated (Zhao et al., 2019). The metrics were calculated based on validating the results with manually annotated bounding boxes of the study area considered. The best result of the rejection and the resulting bounding boxes is displayed in Fig. 7.

The results implied that for the distribution at hand, the algorithms that captured the anomalies in an efficient manner were the robust covariance and the isolation forest algorithms.

After achieving to have robust outlier detection methodologies for our pipeline, the next step was to try to generate the segmentation mask from the object detection output. To this end, the Segment Anything Model (SAM) was used on top of the detected bounding boxes to generate the masks. In a way similar to the Grounding DINO model, SAM takes as input an image and the tensor of bounding boxes which limit SAM to segmenting objects within these bounding boxes only.

The "vit_h" version of the segment anything model was used since it is the biggest and most accurate one thus far. The parameters of SAM were set to the default values proposed in the implementation of the LangSAM class in SAMGeo library, which was used when developing the pipeline (Wu and Osco, 2023).

The output of the mask generation process comes in the form of a semantic segmentation of the input image, where the pixels are assigned with classes or labels based on model's prediction, while irrelevant features (not included in the prompt) are not given a label.

3.3. Pipeline implementation

The image input is expected to be a raster image stored in a Tagged Image File Format (TIFF) format, which is then loaded into the program as a NumPy (Harris et al., 2020) array. Then, the size of the sliding window is defined alongside the overlap, the sliding window will slide over the input image for each row of the image and run the object detection described in the next paragraph.

The SAMGeo package (Wu and Osco, 2023) is then used to load the Grounding DINO model, where the model is used to perform the object detection task with fixed hyperparameters depending on the spatial resolution of the input image and the size of the features; the hyperparameters are set to low values between 0.1–0.3 (automatic setting of the parameters based on the spatial resolution of the image was not considered in this study) to allow the model to over-estimate the bounding boxes to fully capture all objects in the image, and to reject the outliers in the next step. The output of this step is a tensor of the detected bounding boxes, these bounding boxes are indexed, and their areas are calculated using their image coordinates, the list of bounding boxes' areas (areas list) is then ordered by size and passed to the post-processing block.

The post-processing includes the outlier detection/ rejection methods described above. For the statistical methods, NumPy was used to calculate the mean, standard deviation, and interquartile ranges of the ordered areas list. The rest of the outlier detection methods are implemented using the scikit-learn library (Pedregosa et al., 2011).

The outliers are rejected, and the original bounding boxes are then retrieved using the areas list and their respective index. Finally, the list of processed bounding boxes is stored and displayed, and the evaluation metrics are calculated as described in the next section.

The methods described in the pipeline have been applied to the aerial image of Kala Nera for detecting cars, as shown in Fig. 8, where the original image is displayed in Fig. 8(a), the object detection applied to the original image in Fig. 8(b), the object detection using a sliding window in Fig. 8(c), and finally the outlier detection for the object detection overestimation in Fig. 8(d). The figures also highlight the shortcomings of each method. In particular, the sliding window concept was introduced since it helps capture small objects in large images as seen when comparing (b1) and (c1). The outlier rejection comes in handy for detecting objects that do not belong to the desired feature specified in the input prompt (false positives), rejecting these outliers will not only increase the accuracy of the segmentation, it will also reduce the computational load of the segmentation process since it will pass fewer bounding boxes.

4. Results & discussion

This section presents the results of applying the final pipeline to all the datasets presented in Section 3. The section starts with an example that displays the results of varying prompts for the same input image. Then, an overall evaluation is carried out for the pipeline by comparing the segmentation results with the manually crafted ground truth masks of the input imagery. A comparison between the results of this study and a previous relevant study is presented. Finally, a brief discussion of the results is done.

4.1. Text prompts

As discussed in Section 3, SAM by itself does not support text prompting. However, Grounding DINO supports text prompts as inputs, which allows it to do semantic object detection, capturing the features specified in the input text prompt. Using the output bounding boxes from Grounding DINO as inputs for the Segment Anything Model simulates the behavior of semantic segmentation using text prompts. Exploiting that knowledge and by applying the pipeline suggested in Section 3, which improves the performance of the integration of Grounding DINO and SAM to a sample image, the results of applying the pipeline are presented in Fig. 9. The figure shows both the object detection (bounding boxes after outlier rejection) and the segmentation, which is produced when using the resulting bounding boxes as input prompts.

4.2. Evaluation

To evaluate the performance of the pipeline for the segmentation task, the results of its application must be compared to ground truth masks. To obtain the ground truth, we manually annotated the masks for each input image and for each input text prompt; for example, the input image in Fig. 9 has been manually annotated two times, once for the class of "white cars", and another time for the class "blue cars". Then, the images and the text prompts were fed to the pipeline, and the segmentation masks were retrieved. Both the ground truth masks and the resulting segmentation masks are formatted as matrices with the same dimensions as the original image. These matrices use binary Boolean values — true and false — to indicate the presence or absence of the class in each pixel of the input image.

The ground truth masks and the segmentation masks are cross-checked to retrieve the following indices: true positive (TP), where the flags are true in both prediction and ground truth; False Positive (FP), where the prediction flags true and the ground truth false; True Negative (TN), when both prediction and ground truth are false; and finally, False Negative (FN) when the ground truth is true and the prediction is false. These indices are used to calculate the accuracy metrics needed to evaluate the pipeline.

The results of applying the pipeline to the different input images are displayed in Fig. 10.

As previously mentioned, the ground truth consists of manually annotated segmentation masks that cover the objects specified in the input text prompt. The overall results for all the examples and prompts are provided in Table 1 in the form of Accuracy, True Positive Ratio (TPR), and False Negative Ratio (FNR) (Padilla et al., 2020). The equations to calculate the metrics are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$\text{TPR} = \frac{TP}{TP + FN} \tag{4.2}$$

$$\text{FNR} = \frac{FN}{TP + FN} \tag{4.3}$$

Using Eq. (4.1) with the output of the pipeline for the input images to evaluate the performance of our pipeline against the manually annotated ground truth provides the metrics needed to evaluate our pipeline.

4.3. Comparison with previous study

In the work presented by the researchers in Osco et al. (2023b), the performance of semantic textual segmentation using Grounding DINO and SAM has been evaluated using both models in an off-the-shelf manner without any intermediate steps (in zero-shot case). In this section, we compare the performance of zero-shot segmentation of our work, with the performance of applying the models as suggested in the

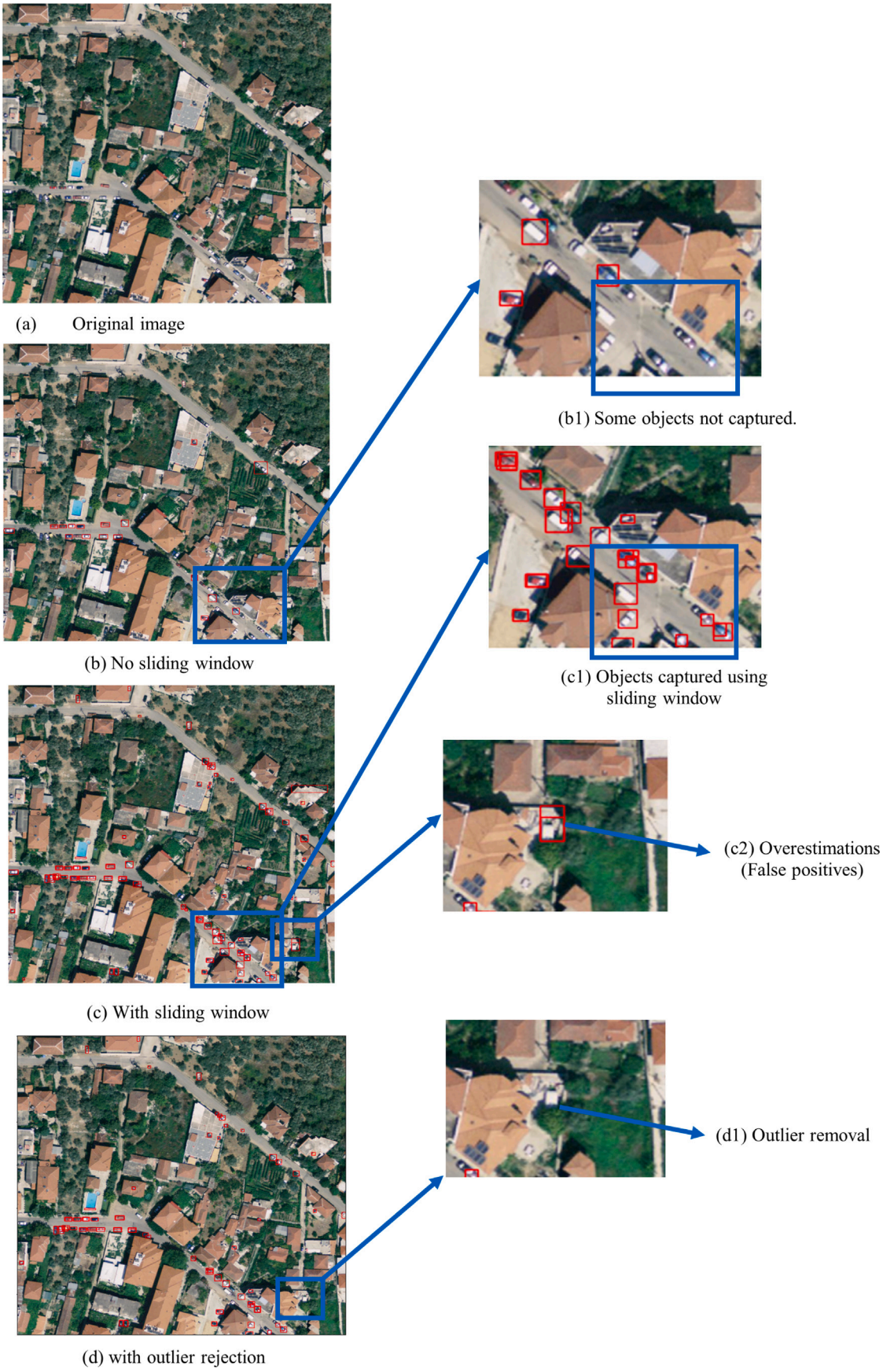

Fig. 8. Pipeline processing methods applied to an aerial image.

Fig. 9. Pipeline segmentation results for a UAV image using different prompts.

Table 1
Model metrics for different models trained on different training datasets.

Area	Platform	Prompt	Resolution (cm)	Accuracy	TPR	TNR
1	UAV	Roof	1	98.9	98	2
2	UAV	Blue car	5	99.95	97	3
2	UAV	White car	5	98.1	96.7	3.3
3	Airborne	Roof	13	96.1	97	3
3	Airborne	Car	13	99.4	84.9	15.1
4	Airborne	Tree	25	91	84.4	15.6
5	Satellite	Tree	60	80	75	25

study.

In this comparison, the Kala Nera image (Fig. 10(b)) has been used for maintaining consistency, as we have already demonstrated in detail the results of our pipeline using this particular image. We utilized this image and used the text prompt "roof" to test our pipeline against the methods described and implemented in Osco et al. (2023b) and Wu and Osco (2023), where the researchers presented their findings in their original work, and then presented the SAMGeo package which implemented the methods and procedures of the original work.

LangSAM, which is a module defined within the SAMGeo package, takes the input text and image and feeds them to Grounding DINO to identify regions in the image that match the text prompt “roof”, which would produce bounding boxes as output. However, these bounding boxes can sometimes be too large, covering multiple roof instances instead of just one as discussed in our study. Next, these bounding boxes are fed to SAM as input prompts, which tries to create segmentation masks of the objects present within the bounding boxes. So, the final output from SAMGeo is a set of these masks showing where the roofs are in the image based on the initial “roof” prompt.

The results of the segmentation and object detection are displayed in Fig. 11. The left image presents the results of SAMGeo and the right image presents the results of our pipeline. By visual inspection, it can be noted that SAMGeo overestimates the segmentation masks. This problem is because Grounding DINO assigns bounding boxes that cover multiple instances of the object as discussed in Section 3, then these boxes are passed to SAM which would, in turn, make SAM overestimate the segmentation masks. On the other hand, our pipeline does not face this issue, since it deals with object detection outliers in the form of bounding box outlier rejection, and it applies the sliding window concept, which allows for a homogeneous distribution of the area size of the resulting bounding boxes.

5. Conclusion

With the recent advancements in the AI field that help automate and simplify already existing and new applications, major attention is being shifted toward integrating AI solutions into the field of RS.

Due the emergence of VLMs and their efficiency in handling problems of multi-modal natures, foundational models, which were trained on vast amounts of general knowledge, exhibit great potential of being integrated into RS applications. However, their general knowledge limits them in terms of trying to discriminate and generate answers and solutions based on unseen data in what is called a zero-shot inference. The pipeline presented in this work tries to minimize the error resulting from the zero-shot prediction by introducing algorithms, tools, and applications on top of already existing FMs to optimize their performance for specific domains instead of building new FMs or fine-tuning existing ones, which is limited to individuals and organizations having the required resources to do so.

The pipeline developed in this work was successful in improving the performance of object detection using Grounding DINO and SAM for aerial and satellite imagery. In retrospect, the pipeline developed

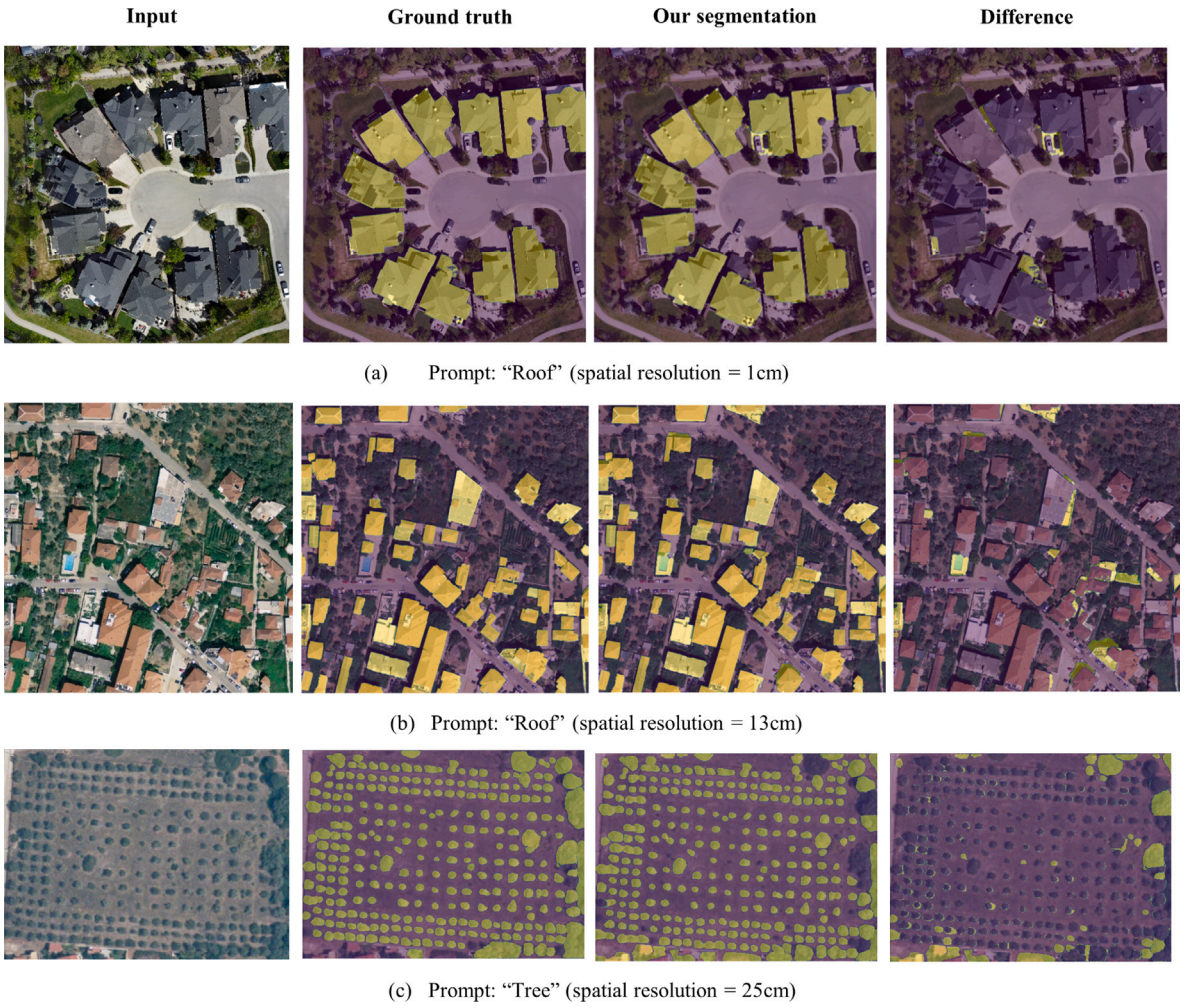

Fig. 10. Suggested pipeline applied to input imagery of varying spatial resolutions compared against the manually annotated ground truth masks, on the far right the masks difference is presented.

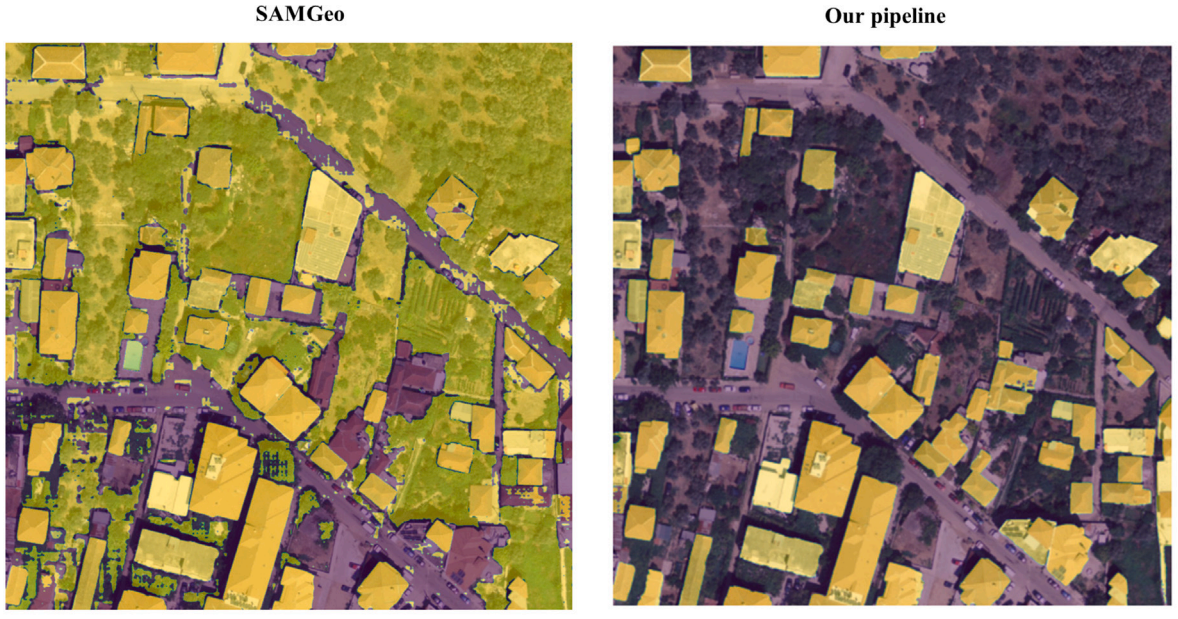

Fig. 11. "The segmentation masks of applying the SAMGeo and our pipeline for the prompt Roof".

in this study was able to detect features and segment them based on the semantic description provided in the input textual prompt as shown in the results in Section 4.1. It also demonstrated its accurate segmentation potential when compared to the ground truth masks as shown in Section 4.2. Finally, in comparison with the current way that the models are being used, our pipeline displayed improvements to the zero-shot potential of the foundation VLM's by deploying applications and algorithms on top of the models.

In the future, we plan to further evolve the research on this front by incorporating specialized LLMs using retrieval-augmented generation (RAG), more FMs for object detection and Q&A, and finally, to build an open-source interface that implements the pipeline, which would help people of no or little experience in detecting and identifying features in RS imagery to automate such tasks.

CRediT authorship contribution statement

Mohanad Diab: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Polychronis Kolokoussis:** Writing – review & editing, Validation, Supervision, Resources, Conceptualization. **Maria Antonia Brovelli:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for [ISPRS - International Journal of Geoinformation] and [Taylor & Francis - International Journal of Digital Earth] and was not involved in the editorial review or the decision to publish this article.

References

Ball, J.E., Anderson, D.T., Chan, C.S., 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. J. Appl. Remote Sens. 11, 1.

Bhorge, S., Rane, M., Rane, N., Patil, M., Saraf, P., Nilgar, J., 2023. Visual AI for satellite imagery perspective: A visual question answering framework in the geospatial domain. In: 2023 IEEE 8th International Conference for Convergence in Technology. I2CT, pp. 1–6. http://dx.doi.org/10.1109/I2CT57861.2023.10126467.

Biswas, S.S., 2023. Potential use of chat GPT in global warming. Ann. Biomed. Eng. 51.

Bommasani, R., Hudson, D.A., Adeli, E., Altman, R.B., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N.S., Chen, A.S., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N.D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M.S., Krishna, R., Kuditipudi, R., et al., 2021. On the opportunities and risks of foundation models. CoRR abs/2108.07258, arXiv:2108.07258, URL: https://arxiv.org/abs/2108.07258.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., 2020. Language models are few-shot learners. In: NeurIPS Proceedings.

Chen, J., Lin, B., Xu, R., Chai, Z., Liang, X., Wong, K.-Y.K., 2024. Mapgpt: Map-guided prompting for unified vision-and-language navigation. arXiv.org.

Chi, M., Plaza, A., Benediktsson, J.A., Sun, Z., Shen, J., Zhu, Y., 2016. Big data for remote sensing: Challenges and opportunities. Proc. IEEE 104 (11), 2207–2219. http://dx.doi.org/10.1109/JPROC.2016.2598228.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Sylvain, G., Uszkoreit, J., Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arxiv.

Fernandez, A., Dube, S., Oct. 2023. Core Building Blocks: Next Gen Geo Spatial GPT Application. (Cornell University), arXiv.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D., 2023. A survey on vision transformer. IEEE Trans. Pattern Anal. Mach. Intell. 45 (1), 87–110. http://dx.doi.org/10.1109/TPAMI.2022.3152247.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585 (7825), 357–362. http://dx.doi.org/10.1038/s41586-020-2649-2.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778.

Jakubik, J., Roy, S., Phillips, C.E., Fraccaro, P., 2023. Foundation models for generalist geospatial artificial intelligence. arXiv.org.

Juhász, L., Mooney, P., Hochmair, H.H., Guan, B., 2023. ChatGPT as a mapping assistant: A novel method to enrich maps with generative AI and content derived from street-level photographs. In: Spatial Data Science Symposium 2023.

Kampffmeyer, M., Salberg, A.-B., Jenssen, R., 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 680–688. http://dx.doi.org/10.1109/CVPRW.2016.90.

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., 2022. Transformers in vision: A survey. ACM Comput. Surv..

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollar, P., Girshick, R., 2023. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 4015–4026.

Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., 2007. ChatGPT: Jack of all trades, master of none. In: PsycCRITIQUES. Vol. 52.

Krizhevsky, A., Sutskever, I., Hinton, G.E., May 2012. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 84–90.

Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14, 778–782.

Li, J., Dada, A., Puladi, B., Kleesiek, J., Egger, J., 2024. ChatGPT in healthcare: A taxonomy and systematic review. Comput. Methods Programs Biomed. 245, 108013.

Li, S., Dragicevic, S., Castro, F.A., Sester, M., 2016. Geospatial big data handling theory and methods: A review and research challenges. ISPRS J. Photogramm. Remote Sens. 115, 119–133.

Li, Z., Ning, H., 2023. Autonomous GIS: the next-generation AI-powered GIS. Int. J. Digit. Earth 16, 4668–4686.

Li, Y., Zhang, Y., Huang, X., Zhu, H., Ma, J., 2018. Large-scale remote sensing image retrieval by deep hashing neural networks. IEEE Trans. Geosci. Remote Sens. 56, 950–965.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L., 2023a. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. arxiv.

Liu, S., Zeng, Z., Tianhe, R., Feng, L., Hao, Z., Jie, Y., Chunyuan, L., Jianwei, Y., Hang, S., Jun, Z., Lei, Z., 2023b. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. arxiv.

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., Jie, W., 2015. Remote sensing big data computing: Challenges and opportunities. Future Gener. Comput. Syst. 51, 47–60.

Mai, G., et al., 2023. On the Opportunities and Challenges of Foundation Models for Geospatial Artificial Intelligence. (Cornell University), arXiv.

Mehta, S., Jain, G., Mala, S., 2023. Natural language processing approach and geospatial clustering to explore the unexplored geotags using media. In: 2023 13th International Conference on Cloud Computing, Data Science and Engineering (Confluence). pp. 672–675. http://dx.doi.org/10.1109/Confluence56041.2023.10048848.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., et al., 2024. GPT-4 technical report. arXiv:2303.08774, URL: https://arxiv.org/abs/2303.08774.

Osco, L.P., Lemos, E.L.d., Gonçalves, W.N., Ramos, A.P.M., Junior, J.M., 2023a. The potential of visual ChatGPT for remote sensing. Remote Sens. 15, 3232.

Osco, L.P., Wu, Q., de Lemos, E.L., Gonçalves, W.N., Ramos, A.P.M., Li, J., Marcato, J., 2023b. The Segment Anything Model (SAM) for remote sensing applications: From zero to one shot. Int. J. Appl. Earth Obs. Geoinf. 124 (103540), http://dx.doi.org/10.1016/j.jag.2023.103540.

Ozdarici-Ok, A., Ok, A.O., Schindler, K., 2015. Mapping of agricultural crops from single high-resolution multispectral images—Data-driven smoothing vs. Parcel-based smoothing. Remote Sens. 7, 5611–5638.

Padilla, R., Netto, S.L., da Silva, E.A.B., 2020. A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing. IWSSIP, pp. 237–242. http://dx.doi.org/10.1109/IWSSIP48289.2020.9145130.

Patil, R., Gudivada, V., 2024. A review of current trends, techniques, and challenges in Large Language Models (LLMs). Appl. Sci. (Basel) 14, 2074.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12 (Oct), 2825–2830.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. In: Semanticscholar. URL: https://api.semanticscholar.org/CorpusID:160025533.

Rao, J., Gao, S., Mai, G., Janowicz, K., 2023. Building Privacy-Preserving and Secure Geospatial Artificial Intelligence Foundation Models (Vision Paper). (Cornell University), arXiv.

Reading, M., URL: https://map.openaerialmap.org/#/-2.185888000000018.

Shi, N.H., Chen, L., Bi, F.-k., Chen, H., Yu, Y., 2015. Accurate urban area detection in remote sensing images. IEEE Geosci. Remote Sens. Lett. 12, 1948–1952.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., et al., 2024. Gemini: A family of highly capable multimodal models. arXiv:2312.11805, URL: https://arxiv.org/abs/2312.11805.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G., 2023. LLaMA: Open and efficient foundation language models. arXiv:2302.13971, URL: https://arxiv.org/abs/2302.13971.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: NeurIPS Proceedings.

Wu, Q., Osco, L.P., 2023. Samgeo: A python package for segmenting geospatial. J. Open Source Softw..

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., Vajda, P., 2019. Visual transformers: Token-based image representation and processing for computer vision. arXiv.org.

Xu, Z., Jain, S., Kankanhalli, M., 2024. Hallucination is inevitable: An innate limitation of large language models. arXiv.org.

Yao, J.-Y., Ning, K.-P., Liu, Z.-H., Ning, M.-N., Yuan, L., 2023. LLM Lies: Hallucinations are not bugs, but features as adversarial examples. arXiv.org.

Yu, B., Yang, L., Chen, F., 2018. Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11, 3252–3261.

Zhang, Y., Lu, Y., Zhang, D., Shang, L., Wang, D., 2018. RiskSens: A multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 1544–1553. http://dx.doi.org/10.1109/BigData.2018.8621996.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2019. Object detection with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 30 (11), 3212–3232. http://dx.doi.org/10.1109/TNNLS.2018.2876865.