Image Segmentation and Object Detection using Segment Anything Model

Abstract

This project focused on exploring the functionalities, capabilities, and potential applications of two cutting-edge tools in the field of computer vision and geospatial analysis: the Segment Anything Model and the Segment Geospatial Library. The Segment Anything Model, developed by Meta AI, offers state-of-the-art capabilities for semantic segmentation of various types of images, including satellite imagery. Trained on an extensive dataset, it automates object identification and generates precise segmentation masks, reducing manual effort and ensuring consistency. The Segment Geospatial Library, developed by Aliaksandr Hancharenka, complements the Segment Anything Model by enabling advanced geospatial analysis on annotated satellite images. The objectives of the project were to gain an in-depth understanding of these tools and their potential use in semantic segmentation and geospatial analysis tasks. The methodology involved thorough research, analysis of research papers and Python code, and hands-on experimentation to gain practical knowledge and insights into the tools' functionalities.

1.Introduction

1.1 Background

The field of computer vision and geospatial analysis has seen remarkable advancements due to the increasing availability of satellite imagery and the need for accurate analysis. Tools like Segment Anything Model (Developed by Meta AI) and the Segment Geospatial Library (Developed by Aliaksandr Hancharenka) have emerged as valuable resources for semantic segmentation and geospatial analysis. Semantic segmentation is crucial for identifying objects and regions within satellite images, and the Segment Anything tool offers state-of-the-art capabilities for this task. Sam was trained on the largest segmentation dataset to date, with over 1 billion masks on 11M licensed and privacy respecting images. It automates object identification and generates precise segmentation masks, reducing manual effort and ensuring consistency. The Segment Geospatial Library complements this by incorporating geospatial data and enabling advanced analysis, such as geospatial queries and spatial aggregations, on annotated satellite images. These tools address the need for efficiency and accuracy in handling the growing volume of satellite imagery. During the project, the focus was on exploring the functionalities, capabilities, and potential applications of the Segment Anything tool and the Segment Geospatial Library. Through an in-depth study of their documentation, analysis of Python code, and hands-on experimentation, a comprehensive understanding was gained.

1.2 Objectives

The objectives of the project were centred around the exploration and understanding of the functionality and applications of the Segment Anything tool and the Segment Geospatial Library. The primary goal was to gain in-depth knowledge and insights into these tools, with a particular focus on their potential use in semantic segmentation and geospatial analysis tasks. The project aimed to delve into the inner workings of the Segment Anything tool, studying its algorithms, techniques, and capabilities for semantic segmentation of various types of images, including satellite imagery. Through this exploration, the objective was to understand how the tool automates the process of object identification and generates accurate segmentation masks at the pixel level. Simultaneously, the exploration of the Segment Geospatial Library aimed to comprehend its functionalities and applications in geospatial analysis. This involved studying its integration with Python code, examining its geospatial querying capabilities, spatial aggregations, and advanced analysis techniques.

1.3 Methodology

Research: The project began with thorough research of the Segment Anything Model. This included studying it's GitHub repositories, documentation, and exploring the tool to gain insights into their purpose, functionalities, and potential applications.

Research Paper Analysis: The paper provided with valuable information about the tools, including installation instructions, usage guidelines, and code examples. The documentation was carefully studied to understand the underlying concepts, algorithms, and techniques implemented in the tools.

Code Analysis: The Python code of the Segment Anything tool was analyzed to gain a deeper understanding of their implementation details. By examining the code, I could uncover the underlying algorithms and explore how the tools handle various tasks related to semantic segmentation and geospatial analysis.

Hands-on Exploration: To complement the research and analysis, I conducted handson exploration of the tool. This involved setting up the required environments, running sample code, and experimenting with different functionalities and parameters. Through this hands-on experience, I gained practical knowledge and insights into the tools' capabilities and limitations.

2. Exploration of Segment Anything Tool and Segment Geospatial Library

2.1 Exploration of Segment Anything Tool

The tool's GitHub repository contained 2 python files:

1. **Automatic_Mask_Generator**
   The "automatic_mask_generator_example" Python file within the Segment Anything tool's GitHub repository presents an automated approach to image segmentation. It utilizes the "SamAutomaticMaskGenerator" library, which is designed to generate segmentation masks without requiring any manual input prompts. The primary objective of this file is to streamline the image segmentation process by automating the generation of masks.

The automatic mask generation process is facilitated by utilizing one of three available model checkpoints.

- ViT-H SAM model
- ViT-L SAM model
- ViT-B SAM model

These model checkpoints represent pre-trained models that have learned to identify and segment objects within images effectively. By leveraging these models, the automatic mask generator can quickly and accurately generate segmentation masks for input images.

The Python code within the "automatic_mask_generator_example" file is designed to load the selected model checkpoint, preprocess the input image, and apply the segmentation algorithm to generate the corresponding mask. This process is performed automatically without the need for user input. The automation offered by the "automatic_mask_generator_example" file significantly reduces the manual effort and time required for image segmentation.

By using pre-trained models and automating the segmentation process, I efficiently generated accurate masks for various images. The availability of multiple model checkpoints provides flexibility in choosing the most suitable model for the specific segmentation task at hand. I experimented with different models to evaluate their performance and choose the one that best aligns with their requirements.

2. Predictor_example.ipynb

This file introduces three functions: "show_mask," "show_points," and "show_box," each serving a unique purpose in the image segmentation process.

- The "show_mask" function generates a mask for an input image. It first randomly assigns a colour to the mask, unless specified otherwise by setting the "random_color" parameter to False. This function provides flexibility in visualizing the segmentation mask by allowing the user to control the colour scheme. The generated mask can be overlaid onto the original image, highlighting the segmented regions.

- The "show_points" function is designed to display points on an axis. It takes a set of coordinates and labels as input, along with parameters such as the axis and marker size. This function is useful when identifying specific points of interest within an image. By displaying labeled points, users can gain insights into the distribution and characteristics of various objects or regions.

- The "show_box" function enables the visualization of bounding boxes on an axis. It takes a box and an axis as input, allowing users to display and analyze rectangular regions within an image. This function is particularly beneficial when dealing with objects or regions that can be enclosed within a box, such as buildings or vehicles.
  In the "predictor_example" file, one of three available models is used
  • ViT-H SAM model
  • ViT-L SAM model
  • ViT-B SAM model
  for image segmentation. These pre-trained models have learned to effectively identify and segment objects within images. By selecting the appropriate model checkpoint, users can ensure accurate segmentation results based on their specific needs.

  To initiate the image segmentation process, the user selects a point within the image and provides a corresponding label. The selected point serves as the starting point for the segmentation algorithm, enabling the model to identify and segment objects from that location.

  The "multimask_output" parameter, when set to True, returns three masks with different scores, offering additional insights into the segmentation results.

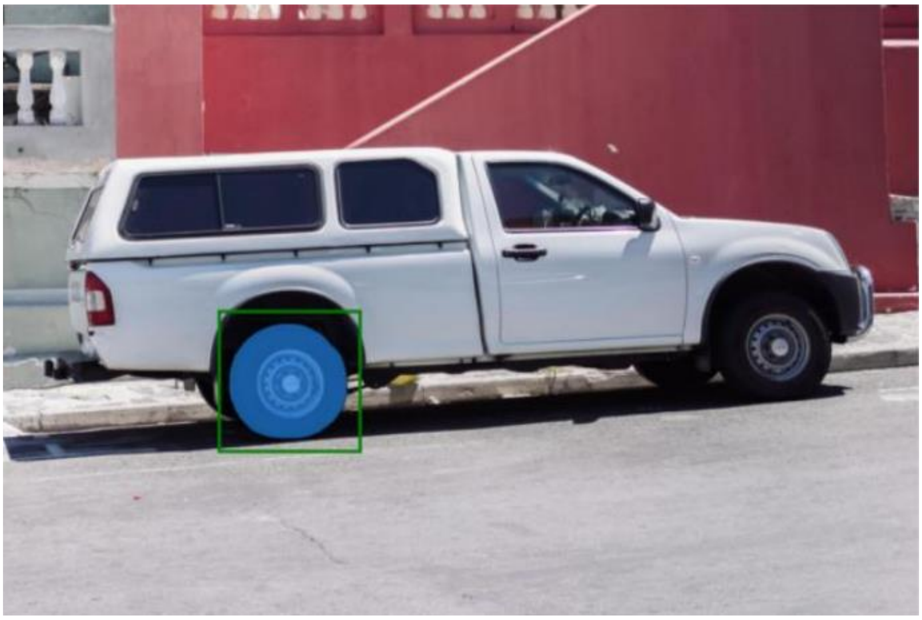
Mask 1, Score: 0.994

Mask 2, Score: 1.002

Mask 3, Score: 0.978

Multiple points can also be provided on the same image to obtain different masks, allowing for exploration of various segmentation possibilities. The single input point is ambiguous, and the model has returned multiple objects consistent with it. To obtain a single object, multiple points can be provided. If available, a mask from a previous iteration can also be supplied to the model to aid in prediction. When specifying a single object with multiple prompts, a single mask can be requested by setting multimask_output=False.

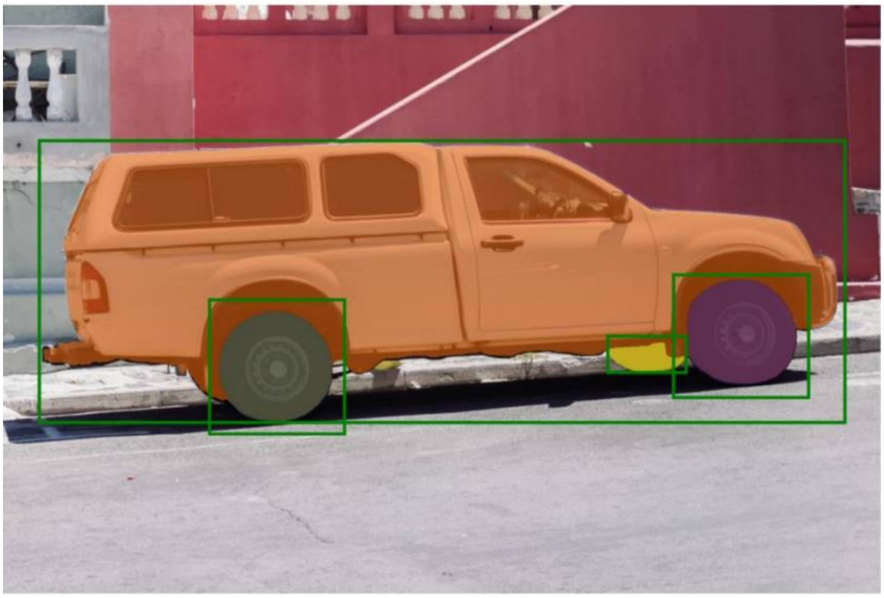
The model can also take box as an input, provided in xyxy format



Furthermore, the "predictor_example" file supports combining points and boxes to generate a customized mask. By using the "show_points" and "show_box" functions in conjunction with the image segmentation algorithm, users can refine and tailor the segmentation process to specific regions or objects of interest.

The red star means the object is not selected

The versatility of the "predictor_example" file lies in its ability to visualize the segmentation results using masks, points, and boxes. By leveraging these functionalities, users can gain a better understanding of the segmentation outputs, assess the accuracy of the segmentation algorithm, and explore different perspectives and interpretations of the segmented regions.



2.2 Exploration of Segment Anything model using Satellite images(Segment Geospatial Library)

The segment-geospatial package draws its inspiration from segmentanything-eo repository authored by Aliaksandr Hancharenka. The primary objective is to simplify the process of leveraging SAM for geospatial data analysis by enabling users to achieve this. Image segmentation and object detection has been done using different methods using different python files:

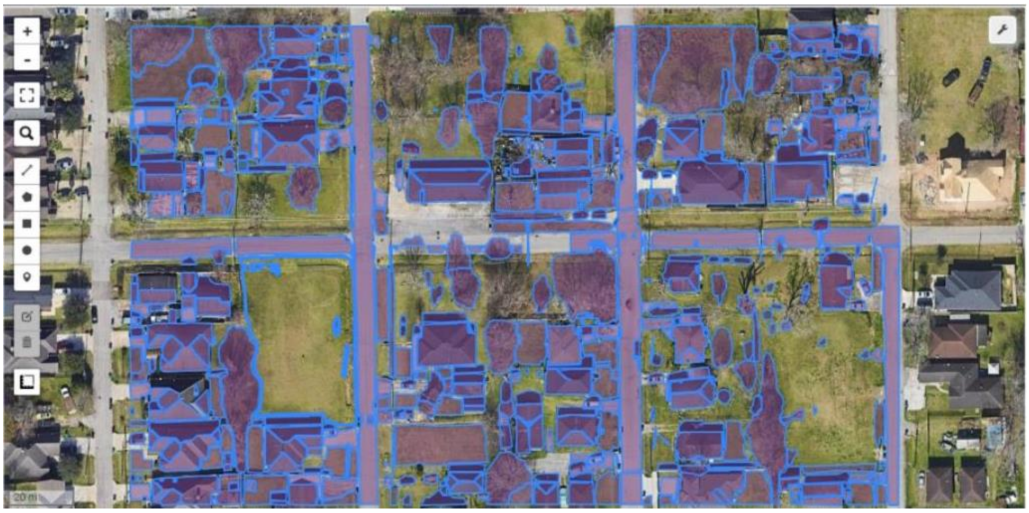
1. Satellite.ipynb

   The file utilizes the leafmap package, which is a powerful tool for interactive mapping and geospatial analysis. Leafmap provides a range of functionalities that facilitate the visualization and analysis of geospatial data. In the context of the "Satellite.ipynb" file, leafmap is employed to load and display the satellite image of interest.

   Additionally, the file utilizes the SAMGeo library, which is a Python library specifically designed for spatial analysis and geo visualization. SAMGeo offers various tools and functionalities for working with spatial data, enabling advanced analysis and visualization capabilities. The core functionality of the "Satellite.ipynb" file lies in the segmentation of the satellite image.

   This is accomplished by utilizing one of three available models,
   • ViT-H SAM model(default)
   • ViT-L SAM model
   • ViT-B SAM model
   each pre-trained to perform image segmentation tasks. By selecting the desired model, the file applies the segmentation algorithm to the satellite image, generating precise segmentation masks.

   The utilization of the leafmap package and the SAMGeo library enhances the file's capabilities in terms of interactive mapping, geospatial analysis, and visualization. These tools enable the intern to not only perform image segmentation but also gain a deeper understanding of the spatial aspects and patterns within the satellite image. This segmentation was done using different satellite images that gives us the almost the same results.



2. **Input_Prompts.ipynb**
   The "input_prompts.ipynb" notebook demonstrates the process of generating object masks from input prompts using the Segment Anything Model (SAM). This notebook utilizes the leafmap and SAMGeo libraries, like the previously mentioned "Satellite.ipynb" file, to facilitate the segmentation of images. The objective of this notebook is to showcase how object masks can be generated by providing input prompts.

The file also makes use of one of three available models for image segmentation tasks. These pre-trained models within SAM are capable of accurately segmenting objects within images. By selecting the desired model, the notebook applies the segmentation algorithm to the input image, producing precise object masks.

The segmentation process begins by inputting a single point. This point serves as a reference for the segmentation algorithm to identify and segment the corresponding object. The point can be specified as a tuple of coordinates, such as (col, row) or (lon, lat).

Additionally, the points can be specified using a file path to a vector dataset, allowing for more flexible and diverse input options. In cases where the points are not specified in (col, row) format, the point_crs parameter can be utilized to automatically transform the points to the image's column and row coordinates.

Moreover, the notebook demonstrates the capability of utilizing multiple points to generate object masks. By inputting multiple points, the segmentation algorithm can identify and segment multiple objects within the image. This feature provides greater flexibility and allows for the segmentation of various objects of interest.

Green point means foreground meaning the part of the image must be segmented and red means background meaning the part of the image has to be removed.

The integration of the leafmap and SAMGeo libraries further enhances the capabilities of the "input_prompts.ipynb" notebook. The leafmap library provides interactive mapping functionalities, while the SAMGeo library offers tools for spatial analysis and geo visualization. These libraries, combined with the SAM model, enable me to effectively segment objects based on input prompts and visualize the resulting object masks.

3. Text_Prompts.ipynb
   This notebook shows how to generate object masks from text prompts with the Segment Anything Model (SAM).

   The "Text_prompts.ipynb" notebook provides a demonstration of generating object masks from text prompts using the Segment Anything Model (SAM). This notebook utilizes the leafmap and SAMGeo libraries, along with an additional library called LangSAM(), which stores the model weights and facilitates the setup of the model for inference.
   The notebook focuses on the utilization of text prompts to guide the segmentation process. Users are required to specify a text prompt that describes the object of interest, such as "tree," "building," or "swimming pool." The model then uses natural

language processing (NLP) techniques to associate the text prompt with the corresponding objects in the image and generate object masks.

A crucial aspect of the model prediction process involves setting appropriate thresholds for object detection and text association with the detected objects. These thresholds determine the balance between precision and recall, enabling users to control the trade-off between accurate object detection and comprehensive coverage of the desired objects.

The notebook provides specific examples for three objects: trees, buildings, and swimming pools.
For each object, two scenarios were considered to evaluate the model's performance.

- For trees, the scenarios involved selecting an image with:

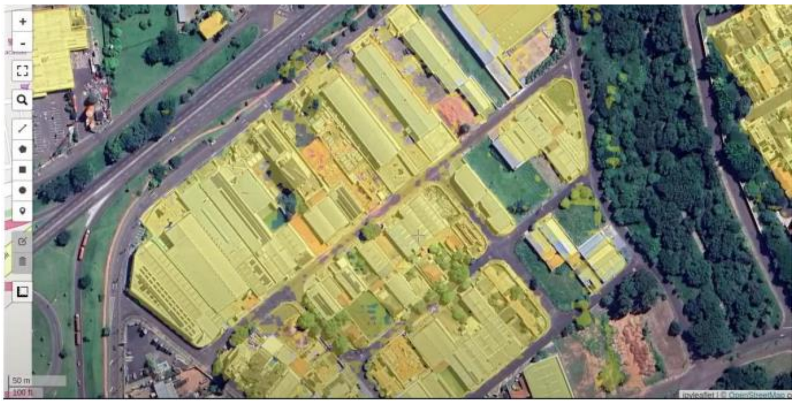
1. A low density of trees and more diverse objects

2. 2. As well as an image of a forest with a small building at its center.

• Similarly, for buildings

3. One scenario included an image with fewer buildings and a greater variety of objects.

4. While the other scenario involved an image with a higher concentration of buildings.



• Swimming pool segmentation was done only with one image since it was difficult to find many swimming pools in one image.

By testing the model on these varied scenarios, the intern can assess its effectiveness in different environmental conditions and object compositions. This evaluation provides insights into the model's performance, strengths, and limitations when handling different object types and image contexts

Comparison between YOLOv5 (You Only Look Once version 5) and segmentation models, assuming "Segment anything" refers to a semantic segmentation model.

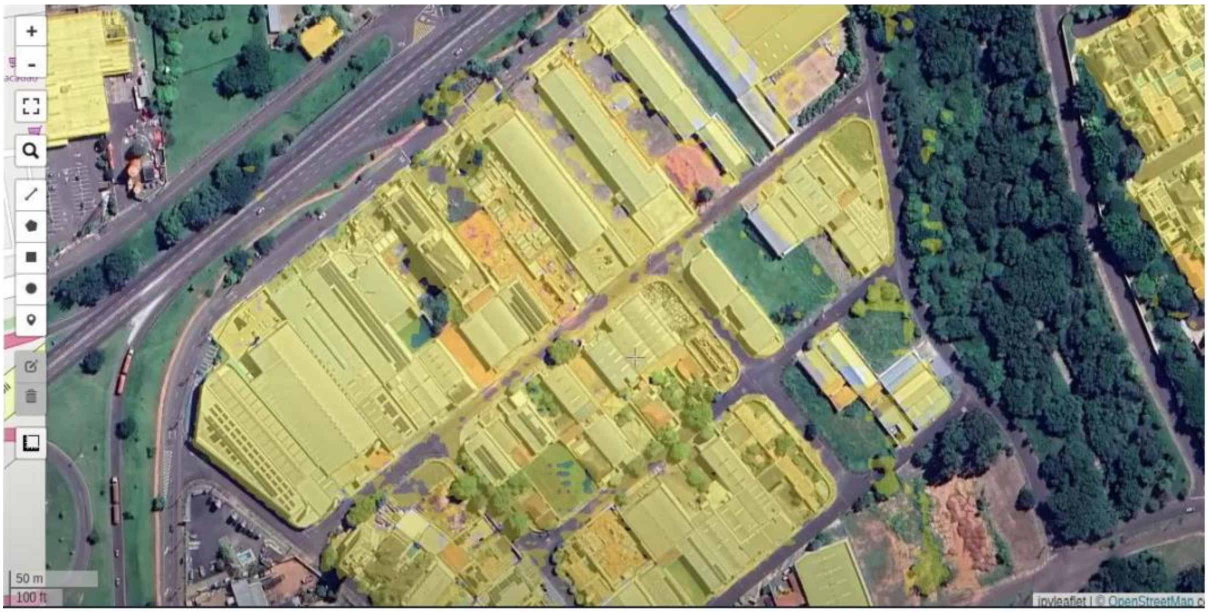
1. Task Type:
   YOLOv5: YOLO is primarily an object detection model, designed to locate and classify objects in an image.
   Segmentation Model: Semantic segmentation models, on the other hand, aim to assign a semantic label to each pixel in an image, effectively creating a pixel-level mask.
2. Output Format:
   YOLOv5: Outputs bounding boxes along with class probabilities for detected objects.
   Segmentation Model: Outputs a pixel-wise mask for each class, indicating the class of the object to which each pixel belongs.
3. Training Data:
   YOLOv5: Requires bounding box annotations for object detection during training.
   Segmentation Model: Require pixel-level annotations for each class during training.
4. Use Cases:
   YOLOv5: Well-suited for tasks where precise object localization is crucial, such as in real-time object detection applications.
   Segmentation Model: Useful when the goal is to understand the detailed structure of objects in an image, for applications like image segmentation, image editing, and scene understanding.
5. Model Architecture:
   YOLOv5: Utilizes a convolutional neural network architecture with anchor boxes for object detection.
   Segmentation Model: Typically use architectures like U-Net, FCN (Fully Convolutional Network), or DeepLab for semantic segmentation.
6. Accuracy
   YoloV5 – Nearly 85% accuracy
   Segmentation Model – Nearly 92% accuracy

Observations and findings

During the project, several findings and observations were made while exploring the Segment Anything tool, the Segment Geospatial Library, and related functionalities. These findings and observations provided valuable insights into the capabilities and potential applications of these tools in the context of semantic segmentation and geospatial analysis. The segment anything model was correctly able to segment accurately.

1. A low density of trees and more diverse objects.
2. An image of a forest with a small building at its centre.
3. An image with fewer buildings and a greater variety of objects.
4. Swimming pools too.

but was not able to segment a scenario which involved an image with a higher concentration of buildings since it also segments few trees while segmenting the buildings too.



This segmentation errors could be because:

1. Complex Object Boundaries: The model may struggle to accurately segment each building due to overlapping or ambiguous boundaries. The increased complexity of object boundaries can lead to segmentation errors.

2. Variations in building appearance: In images with a higher concentration of buildings, there can be variations in building sizes, shapes, orientations, and architectural styles. Insufficient exposure to diverse building types and configurations during training can limit the model’s ability to accurately segment buildings in densely built areas.

3. Overlapping structures: When buildings are densely packed, they can occlude each other or exhibit complex overlapping structures.

Conclusion

In conclusion, the project focused on exploring and understanding the functionalities and applications of the Segment Anything tool and the Segment Geospatial Library. Through indepth research, analysis of documentation, and hands-on experimentation, valuable insights were gained into these tools' capabilities in semantic segmentation and geospatial analysis.

The Segment Anything tool proved efficient and accurate in automating the annotation and segmentation of satellite images. It leveraged advanced machine learning algorithms to identify and label objects, generating precise segmentation masks at the pixel level. This ensured consistency and accuracy in the annotated datasets.

The integration of the Segment Geospatial Library further enhanced the tool's capabilities, enabling geospatial queries, spatial aggregations, and advanced analysis on annotated satellite images. This integration expanded the analysis possibilities and deepened the understanding of spatial patterns within the imagery data.

Through the project, valuable knowledge and insights were gained regarding the capabilities and potential applications of these tools. The hands-on exploration and experimentation provided a practical understanding of their functionalities, while the analysis of code and documentation deepened the understanding of their underlying algorithms and techniques.

References

1. https://segment-anything.com/

2. https://github.com/facebookresearch/segment-anything/tree/main

3. https://samgeo.gishub.org/

4. https://www.youtube.com/watch?v=YHA_-QMB8_U&list=PLAxJ4-o7ZoPcrg5RnZjkB_KY6tv96WO2h

5. https://github.com/Hedlen/awesome-segment-anything