Deep Learning for Computer Vision

Transformers for Image Segmentation

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Recall: Image Segmentation

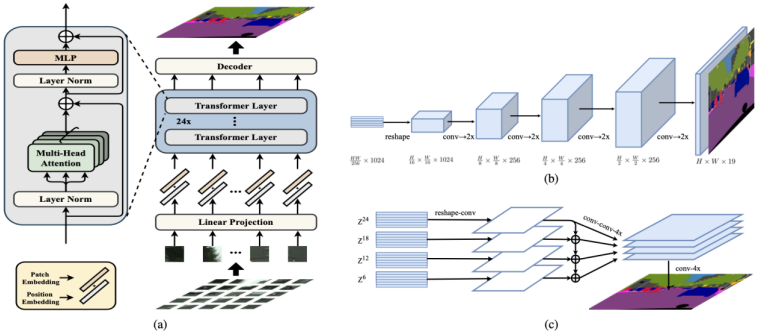
- Image segmentation seeks to partition images into multiple image segments or regions

Recall: Image Segmentation

- Image segmentation seeks to partition images into multiple image segments or regions
- CNNs have achieved remarkable success in image segmentation tasks (recall SegNets, DeepLab, Mask R-CNN), but in recent state-of-the-art approaches, transformers have provided simpler and robust solutions

From DETR to SETR[1]

SETR replaces stacked convolution layer-based encoder with a pure Transformer which gradually reduces spatial resolution

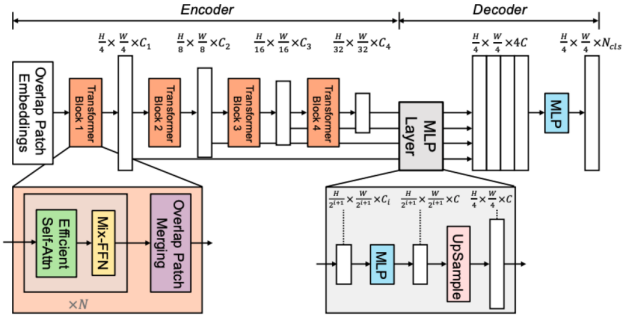


- (a) SETR encoder consists of a standard Transformer
- (b) SETR-PUP decoder with a progressive up-sampling design
- (c) SETR-MLA decoder with a multi-level feature aggregation

[1] Zheng et al, Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, CVPR 2021

The Segformer[2]

Uses a hierarchical Transformer encoder for feature extraction and a lightweight MLP decoder for predicting the final mask



- Uses a patch size of 4×4 in contrast to ViT which uses a patch size of 16×16
- Has an overlapped patch merging process to maintain local continuity around patches
- Introduces Positional-Encoding-Free design as a key feature!

[2]Xie et al, Segformer: Simple and efficient design for semantic segmentation with transformers, NeurIPS 2021

MaskFormer[3]

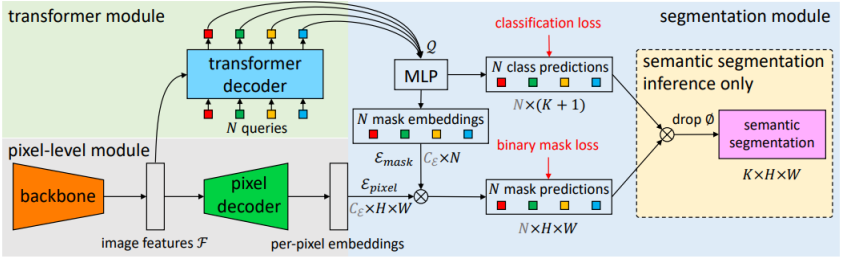
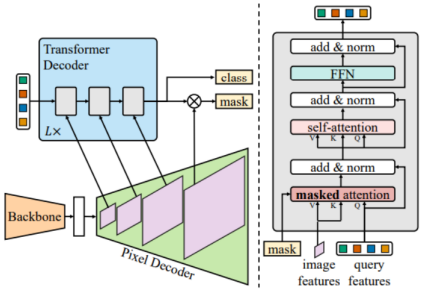


Figure 2: **MaskFormer overview.** We use a backbone to extract image features $\mathcal{F}$. A pixel decoder gradually upsamples image features to extract per-pixel embeddings $\mathcal{E}_{\text{pixel}}$. A transformer decoder attends to image features and produces N per-segment embeddings $\mathcal{Q}$. The embeddings independently generate N class predictions with N corresponding mask embeddings $\mathcal{E}_{\text{mask}}$. Then, the model predicts N possibly overlapping binary mask predictions via a dot product between pixel embeddings $\mathcal{E}_{\text{pixel}}$ and mask embeddings $\mathcal{E}_{\text{mask}}$ followed by a sigmoid activation. For semantic segmentation task we can get the final prediction by combining N binary masks with their class predictions using a simple matrix multiplication (see Section 3.4). Note, the dimensions for multiplication $\otimes$ are shown in gray.

[3]Cheng et al, Per-Pixel Classification is Not All You Need for Semantic Segmentation, NeurIPS 2021

Mask2Former[4]

Universal architecture to perform all segmentation tasks including panoptic, instance, and semantic segmentation



- Model consists of a backbone feature extractor, a pixel decoder, and a Transformer decoder
- Multi-scale deformable attention Transformer (MSDeformAttn) used as a pixel decoder
- Uses a masked attention operator which restricts the cross-attention to the foreground region of the predicted mask and then extracts the localized features. This makes the attention mechanism more efficient
- Needs to be trained separately for each tasks; a common limitation of universal architectures for segmentation tasks

[4] Cheng et al, Masked-attention mask transformer for universal image segmentation, CVPR 2022

Other Transformer-Based Methods

- Segmenter[5]
- Pyramid Vision Transformer (PVT): v1[6], v2[7]
- Dense Prediction Transformer (DPT)[8]
- HRFormer[9]

[5] Strudel et al, Segmenter: Transformer for semantic segmentation, CVPR 2021
[6] Wang et al, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, CVPR 2021
[7] Wang et al, Pvt v2: Improved baselines with pyramid vision transformer, Computational Visual Media 2022
[8] Ranftl et al, Vision transformers for dense prediction, CVPR 2021
[9] Yuan et al, Hrformer: High-resolution vision transformer for dense predict, NeurIPS 2021

Segment Anything Model (SAM)[10]

- Segment Anything Model (SAM) is the first **foundation model** (model trained on broad data that can be used for different tasks with minimal fine-tuning) for image segmentation

[10] Kirrilov et al, Segment Anything, ICCV 2023

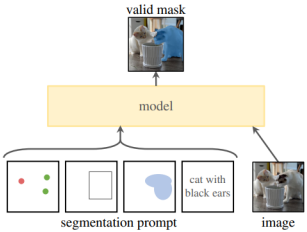
Segment Anything Model (SAM)[10]

- Segment Anything Model (SAM) is the first **foundation model** (model trained on broad data that can be used for different tasks with minimal fine-tuning) for image segmentation
- It is a **promptable model** pre-trained on a **broad dataset** using a **task** that enables powerful downstream generalization

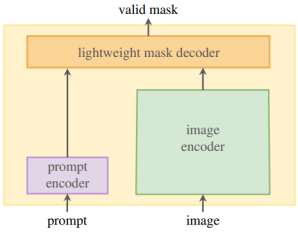
[10] Kirrilov et al, Segment Anything, ICCV 2023

Segment Anything Model (SAM)[10]

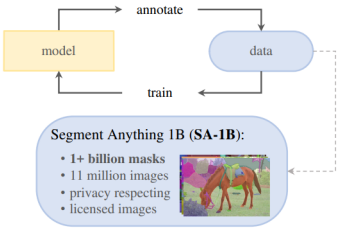
- Segment Anything Model (SAM) is the first **foundation model** (model trained on broad data that can be used for different tasks with minimal fine-tuning) for image segmentation
- It is a **promptable model** pre-trained on a **broad dataset** using a **task** that enables powerful downstream generalization



(a) **Task**: promptable segmentation



(b) **Model**: Segment Anything Model (**SAM**)



(c) **Data**: data engine (top) & dataset (bottom)

[10] Kirrilov et al, Segment Anything, ICCV 2023

The Task

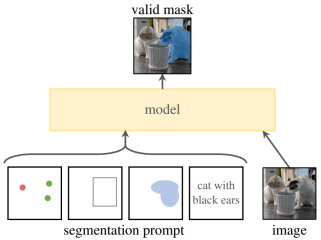
- A task needs to be defined, which is general enough and provides a powerful objective to enable zero-shot generalization to a wide range of downstream applications

The Task

- A task needs to be defined, which is general enough and provides a powerful objective to enable zero-shot generalization to a wide range of downstream applications
- To achieve this, a **promptable segmentation task** is defined, i.e., given an image and a prompt (box, point, text etc), it returns a *valid* segmentation mask (even when a prompt is ambiguous/refers to multiple objects, a mask must be returned for at least one of the objects)

The Task

- A task needs to be defined, which is general enough and provides a powerful objective to enable zero-shot generalization to a wide range of downstream applications
- To achieve this, a **promptable segmentation task** is defined, i.e., given an image and a prompt (box, point, text etc), it returns a *valid* segmentation mask (even when a prompt is ambiguous/refers to multiple objects, a mask must be returned for at least one of the objects)



(a) **Task**: promptable segmentation

The Model

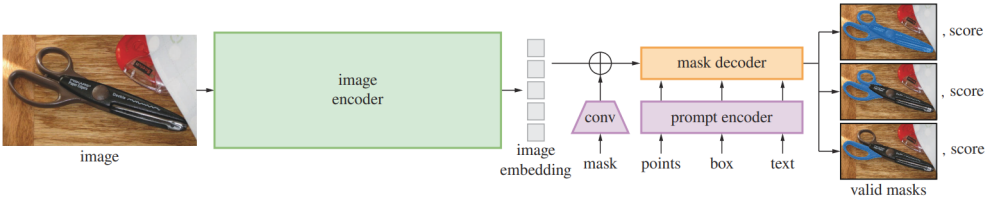
- The model supports **flexible prompts**, computes masks in *amortized real time* (~ 50 ms per mask) for interactive usage and is *ambiguity-aware*

The Model

- The model supports **flexible prompts**, computes masks in *amortized real time* (~ 50 ms per mask) for interactive usage and is *ambiguity-aware*
- The model has three components: an image encoder, a flexible prompt encoder, and a fast mask decoder

The Model

- The model supports **flexible prompts**, computes masks in *amortized real time* (~ 50 ms per mask) for interactive usage and is *ambiguity-aware*
- The model has three components: an image encoder, a flexible prompt encoder, and a fast mask decoder



The Model

- For the image encoder, a pre-trained Vision Transformer is used

[11] Radford et al, Learning Transferable Visual Models From Natural Language Supervision, arXiv 2021

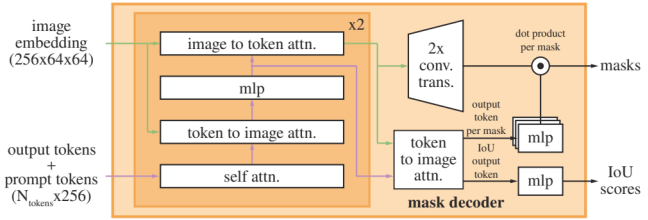
The Model

- For the image encoder, a pre-trained Vision Transformer is used
- For the prompt encoder:
  - box and point prompts are represented using positional encodings
  - text prompts are encoded using the text-encoder from CLIP[11]
  - mask prompts are embedded using convolutions

[11] Radford et al, Learning Transferable Visual Models From Natural Language Supervision, arXiv 2021

The Model

- For the image encoder, a pre-trained Vision Transformer is used
- For the prompt encoder:
  - box and point prompts are represented using positional encodings
  - text prompts are encoded using the text-encoder from CLIP[11]
  - mask prompts are embedded using convolutions
- Mask decoder uses a modified Transformer decoder block



[11] Radford et al, Learning Transferable Visual Models From Natural Language Supervision, arXiv 2021

The Model: How to predict a mask?

Recall the Maskformer:

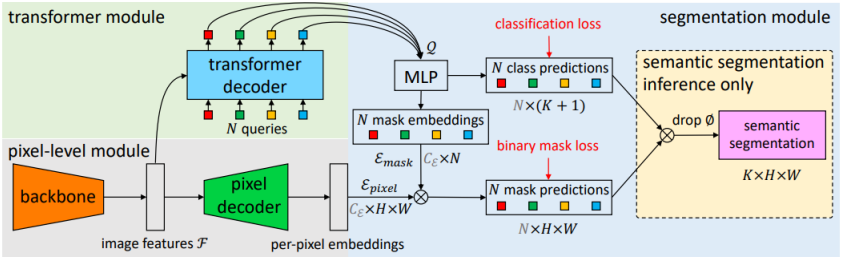


Figure 2: **MaskFormer overview.** We use a backbone to extract image features $\mathcal{F}$. A pixel decoder gradually upsamples image features to extract per-pixel embeddings $\mathcal{E}_{\text{pixel}}$. A transformer decoder attends to image features and produces N per-segment embeddings $\mathcal{Q}$. The embeddings independently generate N class predictions with N corresponding mask embeddings $\mathcal{E}_{\text{mask}}$. Then, the model predicts N possibly overlapping binary mask predictions via a dot product between pixel embeddings $\mathcal{E}_{\text{pixel}}$ and mask embeddings $\mathcal{E}_{\text{mask}}$ followed by a sigmoid activation. For semantic segmentation task we can get the final prediction by combining N binary masks with their class predictions using a simple matrix multiplication (see Section 3.4). Note, the dimensions for multiplication $\otimes$ are shown in gray.

The Data Engine

- To enable strong generalization to new data distributions, SAM needs to be trained on a large and diverse set of masks

The Data Engine

- To enable strong generalization to new data distributions, SAM needs to be trained on a large and diverse set of masks
- A "data engine" was built with model-in-the-loop dataset annotation to create the **SA-1B dataset** with 1 billion masks. This was done in three stages:

The Data Engine

- To enable strong generalization to new data distributions, SAM needs to be trained on a large and diverse set of masks
- A "data engine" was built with model-in-the-loop dataset annotation to create the **SA-1B dataset** with 1 billion masks. This was done in three stages:
  - **Assisted-manual stage:** Human annotators labeled masks using a browser-based interactive tool. SAM was trained using public segmentation datasets and the above collected data.

The Data Engine

- To enable strong generalization to new data distributions, SAM needs to be trained on a large and diverse set of masks
- A "data engine" was built with model-in-the-loop dataset annotation to create the **SA-1B dataset** with 1 billion masks. This was done in three stages:
  - **Assisted-manual stage:** Human annotators labeled masks using a browser-based interactive tool. SAM was trained using public segmentation datasets and the above collected data.
  - **Semi-automatic stage:** This was aimed at increasing diversity of masks. High confidence masks were pre-labeled and human annotators were asked to label any additional objects in the image. SAM was retrained on this data.

The Data Engine

- To enable strong generalization to new data distributions, SAM needs to be trained on a large and diverse set of masks
- A "data engine" was built with model-in-the-loop dataset annotation to create the **SA-1B dataset** with 1 billion masks. This was done in three stages:
  - **Assisted-manual stage:** Human annotators labeled masks using a browser-based interactive tool. SAM was trained using public segmentation datasets and the above collected data.
  - **Semi-automatic stage:** This was aimed at increasing diversity of masks. High confidence masks were pre-labeled and human annotators were asked to label any additional objects in the image. SAM was retrained on this data.
  - **Fully-automatic stage:** The model was prompted with a 32×32 grid of points to generate masks

Zero-Shot Single Point Valid Mask Evaluation

SAM is evaluated on the task of segmenting an object from a single foreground point on 23 new datasets with diverse data distributions. This task is ill-posed as one point can refer to multiple objects.

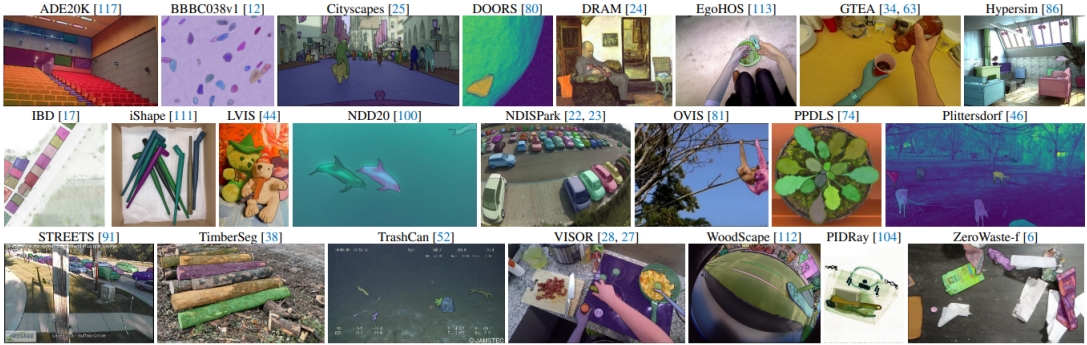
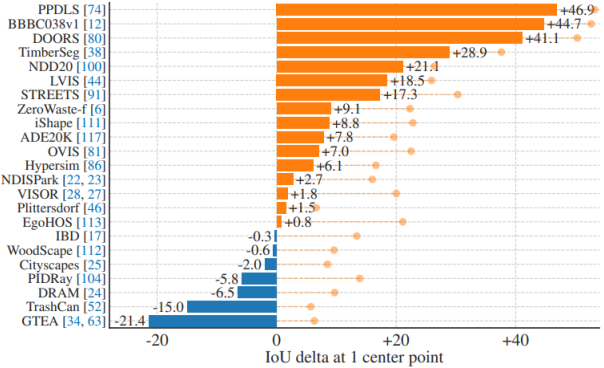


Figure 8: Samples from the 23 diverse segmentation datasets used to evaluate SAM's zero-shot transfer capabilities.

Zero-Shot Single Point Valid Mask Evaluation

The performance of SAM is compared with the state-of-the-art RITM model.[12]



(a) SAM *vs.* RITM [92] on 23 datasets

[12] Sofiiuk et al, Reviving iterative training with mask guidance for interactive segmentation, ICIP 2022

Zero-Shot Text-to-Mask

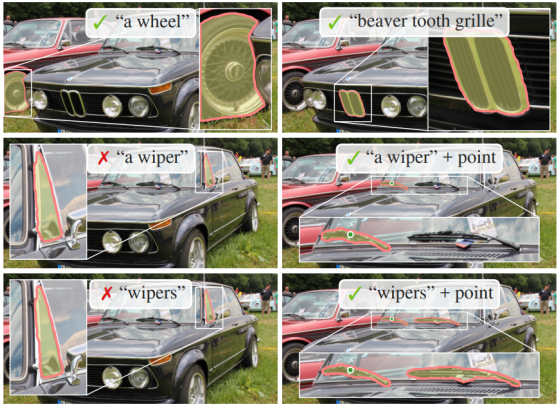


Figure 12: Zero-shot text-to-mask. SAM can work with simple and nuanced text prompts. When SAM fails to make a correct prediction, an additional point prompt can help.

Zero-Shot Object Proposals and Instance Segmentation

	mask AR@1000						
method	all	small	med.	large	freq.	com.	rare
ViTDet-H [62]	63.0	51.7	80.8	87.0	63.1	63.3	58.3
zero-shot transfer methods:							
SAM – single out.	54.9	42.8	76.7	74.4	54.7	59.8	62.0
SAM	59.3	45.5	81.6	86.9	59.1	63.9	65.8

Table 4: Object proposal generation on LVIS v1. SAM is applied zero-shot, *i.e.* it was not trained for object proposal generation nor did it access LVIS images or annotations.

	COCO [66]				LVIS v1 [44]			
method	AP	AP^S	AP^M	AP^L	AP	AP^S	AP^M	AP^L
ViTDet-H [62]	51.0	32.0	54.3	68.9	46.6	35.0	58.0	66.3
zero-shot transfer methods (segmentation module only):								
SAM	46.5	30.8	51.0	61.7	44.7	32.5	57.6	65.5

Table 5: Instance segmentation results. SAM is prompted with ViTDet boxes to do zero-shot segmentation. The fully-supervised ViTDet outperforms SAM, but the gap shrinks on the higher-quality LVIS masks. Interestingly, SAM outperforms ViTDet according to human ratings (see Fig. 11).

Zero-Shot Edge Detection

- SAM is evaluated on the classic low-level task of edge detection using BSDS500[a].
- SAM is prompted with a 16×16 regular grid of foreground points resulting in 768 predicted masks (3 per point).
- Redundant masks are removed by non-maximal supression (NMS). Then, edge maps are computed using Sobel filtering of unthresholded mask probability maps and standard lightweight postprocessing, including edge NMS.

[a] Martin et al, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, ICCV 2021

Figure 10: Zero-shot edge prediction on BSDS500. SAM was not trained to predict edge maps nor did it have access to BSDS images or annotations during training.

method	year	ODS	OIS	AP	R50
HED [108]	2015	.788	.808	.840	.923
EDETR [79]	2022	.840	.858	.896	.930
zero-shot transfer methods:					
Sobel filter	1968	.539	-	-	-
Canny [13]	1986	.600	.640	.580	-
Felz-Hutt [35]	2004	.610	.640	.560	-
SAM	2023	.768	.786	.794	.928

Table 3: Zero-shot transfer to edge detection on BSDS500.

Grounded SAM: Open-Vocabulary Detection and Segmentation[14]

- Grounded SAM uses Grounding DINO[13] as an open-set object detector to combine with the segment anything model (SAM)

[13] Liu et al, Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, arXiv 2023
[14] Ren et al, Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks, arXiv 2024

Grounded SAM: Open-Vocabulary Detection and Segmentation[14]

- Grounded SAM uses Grounding DINO[13] as an open-set object detector to combine with the segment anything model (SAM)
- The annotation cost of detection data is relatively lower compared to segmentation tasks, enabling the collection of more higher-quality annotated data

[13] Liu et al, Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, arXiv 2023
[14] Ren et al, Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks, arXiv 2024

Grounded SAM: Open-Vocabulary Detection and Segmentation[14]

- Grounded SAM uses Grounding DINO[13] as an open-set object detector to combine with the segment anything model (SAM)
- The annotation cost of detection data is relatively lower compared to segmentation tasks, enabling the collection of more higher-quality annotated data
- Given an input image and a text prompt, Grounded SAM employs Grounding DINO to generate precise boxes for objects or regions within the image by leveraging textual information as condition. Subsequently, the annotated boxes obtained through Grounding DINO serve as the box prompts for SAM to generate precise mask annotations

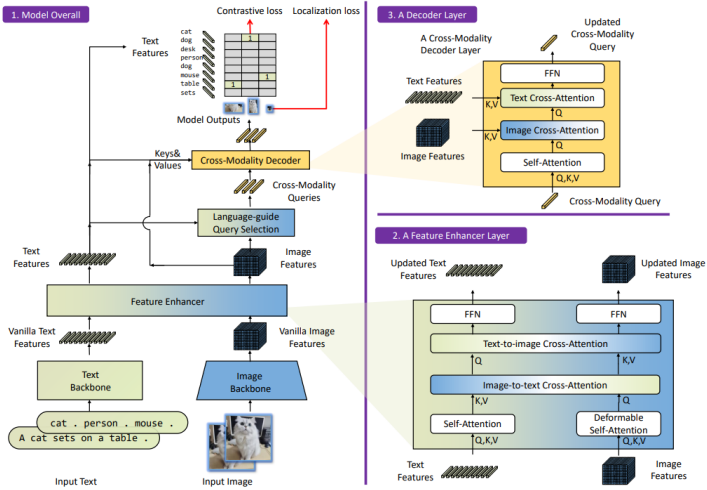
[13] Liu et al, Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, arXiv 2023
[14] Ren et al, Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks, arXiv 2024

Grounded SAM: Open-Vocabulary Detection and Segmentation[14]

- Grounded SAM uses Grounding DINO[13] as an open-set object detector to combine with the segment anything model (SAM)
- The annotation cost of detection data is relatively lower compared to segmentation tasks, enabling the collection of more higher-quality annotated data
- Given an input image and a text prompt, Grounded SAM employs Grounding DINO to generate precise boxes for objects or regions within the image by leveraging textual information as condition. Subsequently, the annotated boxes obtained through Grounding DINO serve as the box prompts for SAM to generate precise mask annotations
- A wide range of vision tasks can be achieved by using the versatile Grounded SAM pipeline.

[13] Liu et al, Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, arXiv 2023
[14] Ren et al, Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks, arXiv 2024

Some Background: Grounding DINO[15]



[15]Liu et al, Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, arXiv 2023

RAM-Grounded-SAM: Automatic Dense Image Annotation

An image-caption model or an image tagging model (like RAM[a]), can be used to generate output results (captions or tags) that can be given as inputs to Grounded SAM to generate precise box and mask for each instance.

RAM-Grounded-SAM: Automatic Dense Image Annotation

An image-caption model or an image tagging model (like RAM[a]), can be used to generate output results (captions or tags) that can be given as inputs to Grounded SAM to generate precise box and mask for each instance.

[a] Zhang et al, Recognize Anything: A Strong Image Tagging Model, arXiv 2023

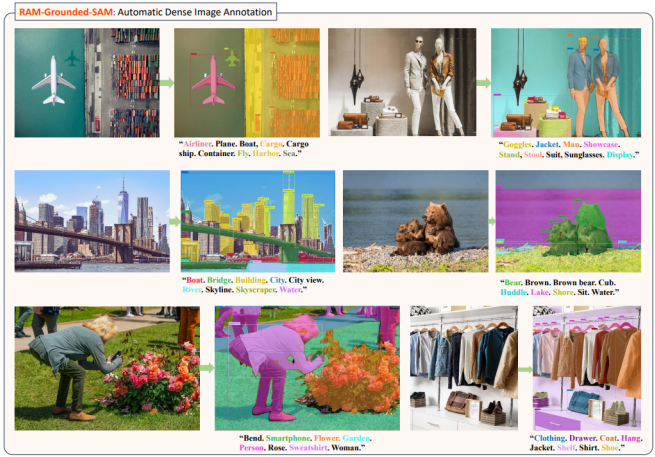


Figure 3: **RAM-Grounded-SAM** combines the robust tagging capabilities of the RAM [83] with the open-set detection and segmentation abilities of Grounded SAM, which enables automatic dense image annotation with only image input (the demo images are sampled from the SA-1B [25] dataset).

Grounded-SAM-SD: Highly Accurate and Controllable Image Editing

- By integrating the powerful text-to-image capability of image generation models with Grounded SAM, a comprehensive framework that enables the creation of a robust data synthesis factory can be created

Grounded-SAM-SD: Highly Accurate and Controllable Image Editing

- By integrating the powerful text-to-image capability of image generation models with Grounded SAM, a comprehensive framework that enables the creation of a robust data synthesis factory can be created
- With the additional capability of an image generation model, highly precise and controlled image manipulation, including modifying image representation, replacing objects, removing the corresponding regions, etc can be achieved.

Grounded-SAM-SD: Highly Accurate and Controllable Image Editing

- By integrating the powerful text-to-image capability of image generation models with Grounded SAM, a comprehensive framework that enables the creation of a robust data synthesis factory can be created
- With the additional capability of an image generation model, highly precise and controlled image manipulation, including modifying image representation, replacing objects, removing the corresponding regions, etc can be achieved.
- In downstream scenarios where data scarcity arises, the system can generate new data, addressing the data requirements for the training of models

Grounded-SAM-SD: Highly Accurate and Controllable Image Editing

Figure 4: **Grounded-SAM-SD** combines the open-set ability of Grounded SAM with inpainting

Grounded-SAM-OSX: Promptable Human Motion Analysis

- Grounded SAM and OSX[16] models can be integrated to achieve a novel promptable (instance-specific) whole-body human detection and mesh recovery system, thereby realizing a promptable human motion analysis system

[16] Lin et al, One-Stage 3D Whole-Body Mesh Recovery with Component Aware Transformer, CVPR 2023

Grounded-SAM-OSX: Promptable Human Motion Analysis

- Grounded SAM and OSX[16] models can be integrated to achieve a novel promptable (instance-specific) whole-body human detection and mesh recovery system, thereby realizing a promptable human motion analysis system
- Specifically, given an image and a prompt to refer to a specific person, Grounded SAM is first used to generate a precise specific human box. Then, OSX is used to estimate an instance-specific human mesh.

[16] Lin et al, One-Stage 3D Whole-Body Mesh Recovery with Component Aware Transformer, CVPR 2023

Grounded-SAM-OSX: Promptable Human Motion Analysis

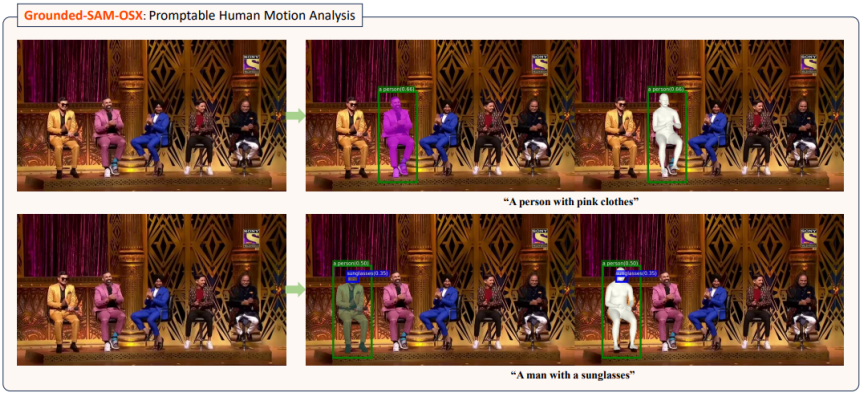


Figure 5: **Grounded-SAM-OSX** merges the text-promptable capability of Grounded SAM with the whole body mesh recovery ability of OSX [33], facilitating a precise human motion analysis system.

More Information

Resources

- HuggingFace Link for Semantic Segmentation
- Semantic Segmentation using Vision Transformers: A survey
- Official Github page for Segment Anything