Friedrich-Alexander-Universität
Department Artificial Intelligence in Biomedical Engineering

IDEA Lab
Image Data Exploration and Analysis Lab

Segment Anything for Orange Segmentation

Exploring Zero-Shot, Finetuning and Comparing with U-Net

Seminar Paper

in the context of the seminar “Project Representation Learning”

at Friedrich-Alexander-Universität Erlangen-Nürnberg
at the Department Artificial Intelligence in Biomedical Engineering (AIBE)
Image Data Exploration and Analysis (IDEA) Lab

Principal Supervisor:	Prof. Dr. Bernhard Kainz
Associate Supervisor:	Mischa Dombrowski

Author:	Ramkrishna Acharya Laubanerstr. 91056 Erlangen +49 15758289545 qramkrishna.acharya@fau.de 23148551

Submission:	30th September 2024

Abstract

Models trained on large domains often exhibit poor zero-shot performance on unseen datasets, even within the same domain. The Segment Anything Model (SAM), known for its strong zero-shot ability across a broad image domain, makes it an ideal candidate to explore its strengths, weaknesses, and performance after fine-tuning. This project aimed to evaluate SAM's zero-shot performance, compare it with the standard ResNet-based U-Net and fine-tune SAM's decoder. Results showed that SAM (mean intersection over union, mIoU, of 0.93 for box prompts and 0.76 for point prompts) and its variant SAM2 (mIoU of 0.93 for box prompts and 0.72 for point prompts) outperformed U-Net (mIoU of 0.881) in box-prompt tasks. Even though SAM outperformed U-Net significantly in box-prompt tasks, U-Net's performance was found to be stronger than SAM's point-prompt. In addition to that, fine-tuning SAM led to a 6% improvement in box-prompt performance, with a mIoU of 0.987. The code used in this project is available on GitHub: https://github.com/q-viper/Orange-Segmentation-With-SAM.

Contents

Figures

Abbreviations

CNN	Convolutional Neural Networks
IoU	Intersection over Union
LLM	Large Language Model
NLP	Natural Language Processing
OIM	Orange Infection Mask
SAM	Segment Anything Model
SOTA	State Of The Art
ViT	Vision Transformer

1 Introduction

Segment Anything (Kirillov et al., 2023) is a state-of-the-art image segmentation research from Meta AI that mainly focuses on three research goals. Firstly, to find a task that allows a prompt-based zero-shot generalization. Second, to find a corresponding model architecture, and finally, to find data that can power this task and model. In light of that, the authors have added the following major contributions through this research:

- Open-source pretrained model, that can segment stuff and things on various visual data.
- Large scale segmentation dataset i.e. SA-1B dataset with a valid license.
- Inference code with Apache 2.0 license.

Having a strong model capability, SAM has been fine-tuned for different applications like medical image segmentation (K. Zhang & Liu, 2023) and (Chai et al., 2023) and has shown better results than standard U-Net (Ronneberger et al., 2015), and transformer-based autoencoders like (Cao et al., 2021).

Considering SAM being a strong segmentator, this project is done on the Orange Infection Mask Dataset (Pokharel & Acharya, 2024) (OIM Dataset) collected from fields of Palpa, Nepal using a mobile phone. The dataset contains a mask for about 1500 infected and fine oranges combined. The labeled dataset contains an infected and healthy image and their corresponding segmentation mask. None of these datasets are known to SAM prior. The major goals of this project are as follows:

- Evaluate the zero-shot ability of SAM and SAM2 Ravi et al., 2024on OIM.
- Train a U-Net on OIM and compare its result with SAM's zero-shot results.
- Fine-tune SAM's decoder part and compare its results with SAM's zeroshot results.

Among the 3 aforementioned goals, the most important is the second one because U-Nets are still widely popular in the industry for being faster and requiring fewer parameters than State Of The Art (SOTA) methods. But, as it is dependent on Convolutional Neural Networks (CNN) LeCun et al., 1998, a single layer of it can only capture local features. However, SAM is based on transformers Vaswani et al., 2023 which uses self-attention and it can capture both local as well as global information. In addition to that, SAM is trained on a wide range of data and different prompts hence would be interesting to see where SAM fails and whether it can be used as a replacement of U-Net for Instance Segmentation tasks. Furthermore, this project contributes:

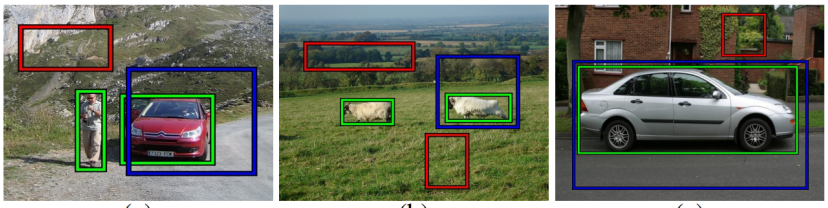
- Results of SAM's zero shot ability on a completely different domain.
- Finetuned SAM model, U-Net model, and the codes to reproduce results.

2 Background and Related Work

SAM being the latest and SOTA method depends on several previous research for inspiration as well as concepts. In this section, we look into some of the related works.

2.1 Object Proposal Generation

Object proposal generation aims at predicting how likely an image is to contain an object. The following figure shows an example of such a task and is referenced from the *What is an object?* (Alexe et al., 2010).



(a) (b) (c)

Fig. 1: **Desired behavior of an objectness measure.** *The desired objectness measure should score the blue windows, partially covering the objects, lower than the ground truth windows (green), and score even lower the red windows containing only stuff or small parts of objects.*

2.2 Semantic Segmentation (2015)

This task is not much different than object proposal generation except for the part that output on this task would usually be a probability mask with a size equal to the input image. Here, each pixel will be classified into one of the classes. Figure 1 shows an example of a model architecture powering such a task.

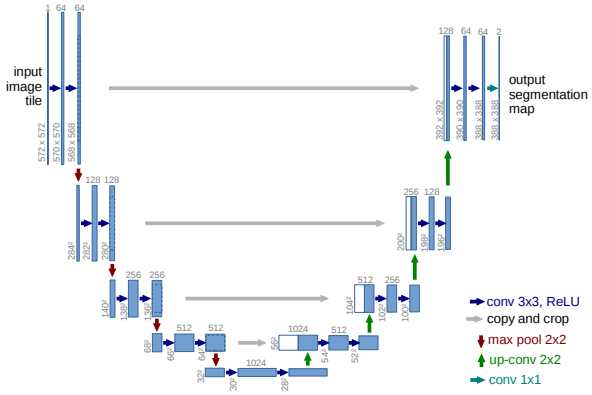


Figure 1 U-Net Architecture (Ronneberger et al., 2015)

2.3 Instance Segmentation

Instead of segmenting objects based on semantics, instance segmentation performs segmentation based on an instance of any object. It can be seen in figure 2.

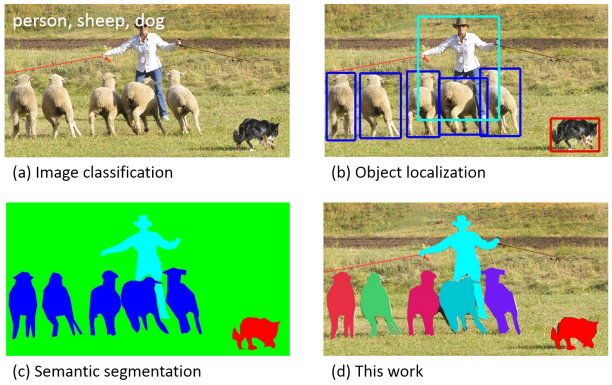


Figure 2 Object Localization vs Semantic Segmentation vs Instance Segmentation (Lin et al., 2015)

2.4 Interactive Segmentation

While interactive segmentation is slightly different than previous tasks, in essence, interactive segmentation also focuses on segmenting things based on the user prompts. *Deep Interactive Object Selection* is the foundational task for it. and the figure 3 shows an example of such task.

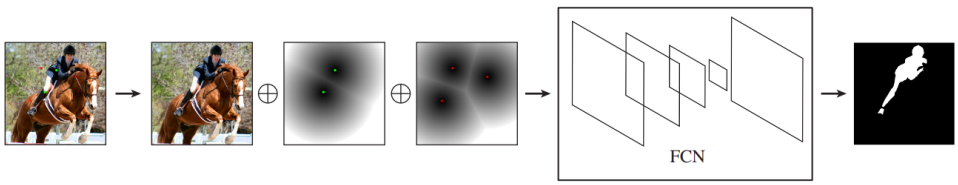


Figure 1: The framework of learning our FCN models. Given an input image and user interactions, our algorithm first transforms positive and negative clicks (denoted as green dots and red crosses respectively) into two separate channels, which are then concatenated (denoted as $\oplus$) with the image's RGB channels to compose an input pair to the FCN models. The corresponding output is the ground truth mask of the selected object.

Figure 3 Deep Interactive Object Selection(Xu et al., 2016)

2.5 Panoptic Segmentation

Panoptic segmentation unifies the typically distinct tasks of semantic segmentation (assign a class label to each pixel) and instance segmentation (detect and segment each object instance). The following figure shows an example of such a task's results.

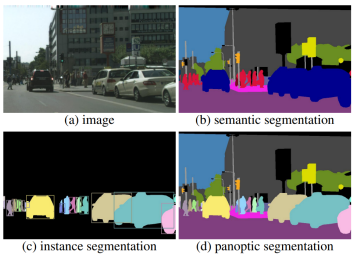


Figure 1: For a given (a) image, we show *ground truth* for: (b) semantic segmentation (per-pixel class labels), (c) instance segmentation (per-object mask and class label), and (d) the proposed *panoptic segmentation* task (per-pixel class+instance labels). The PS task: (1) encompasses both stuff and thing classes, (2) uses a simple but general format, and (3) introduces a uniform evaluation metric for all classes. Panoptic segmentation generalizes both semantic and instance segmentation and we expect the unified task will present novel challenges and enable innovative new methods.

Figure 4 Panoptic Segmentation Example (Kirillov et al., 2019)

2.6 Transformer

Transformers are very powerful foundational models for the present day's Large Language Model (LLM) (e.g. Radford et al., 2018, Devlin et al., 2019, Brown et al., 2020).

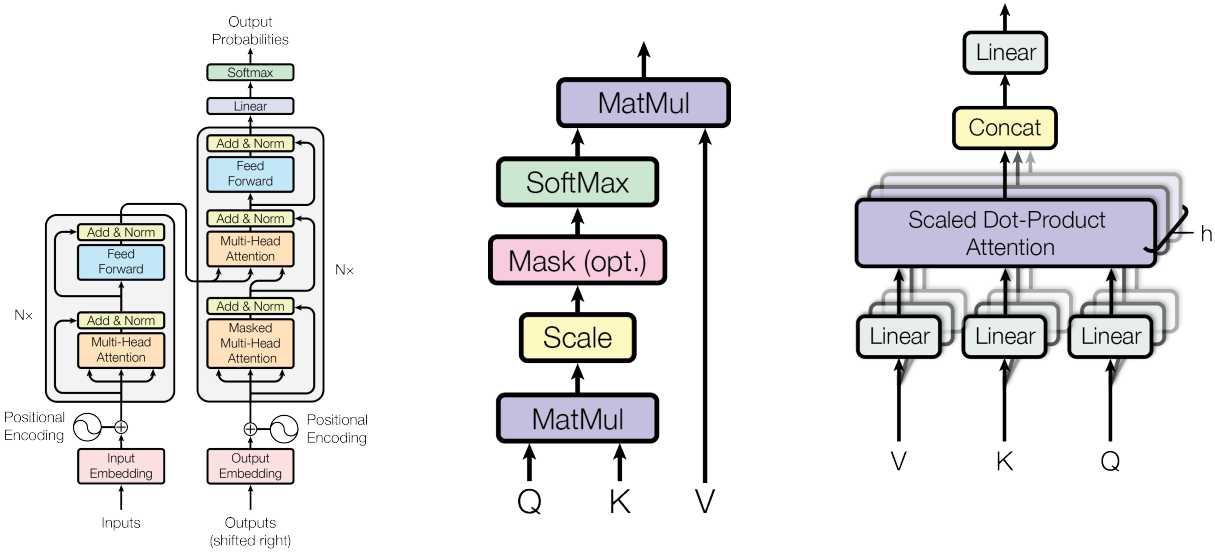


Figure 5 Transformer Architecture (Left) Scaled Dot-Product Attention (Center) Multi-Headed Attention (Right) (Vaswani et al., 2023)

The authors have explained them as:

- Scaled Dot-Product Attention = $\frac{softmax(Q.K^T)}{\sqrt{dim(K)}}V$
- Multi-Headed Attention = $concat(Atn(QW_i^K, KW_i^K, VW_i^V)...)W^O$

2.7 Vision Transformer

In addition to being powerful in LLM, transformers are also strong in vision tasks (e.g. Dosovitskiy et al., 2021, Liu et al., 2021). Figure 6 shows the architecture of such a model.

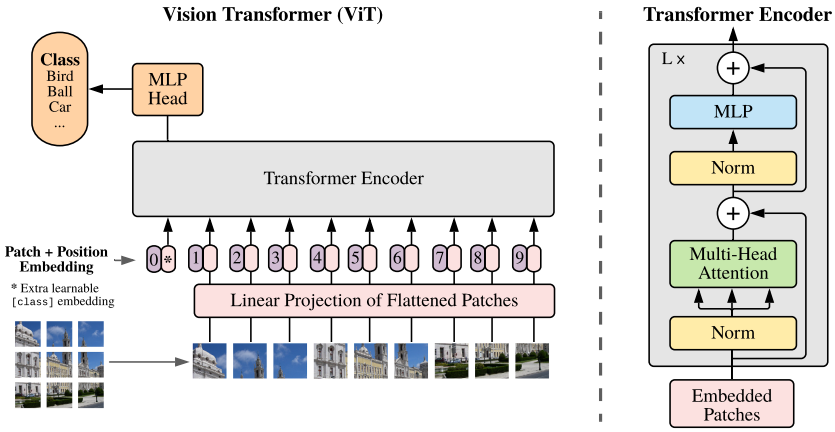


Figure 6 Vision Transformer (Dosovitskiy et al., 2021)

Authors (Dosovitskiy et al., 2021) have provided variants of models based on layers, hidden size, multilayer perceptron size, and number of heads. Such a configuration is provided in table 1.

Model	Layers (L)	Hidden Size (D)	Attention Heads (H)	MLP Size	Parameters (M)
ViT-Base (B)	12	768	12	3072	86
ViT-Large (L)	24	1024	16	4096	307
ViT-Huge (H)	32	1280	16	5120	632

Table 1 Vision Transformer (ViT) Model Configurations

2.8 Masked Auto-Encoder

MAE (Masked Auto Encoder) uses ViT as the encoder and Segment Anything uses MAE as an image encoder.

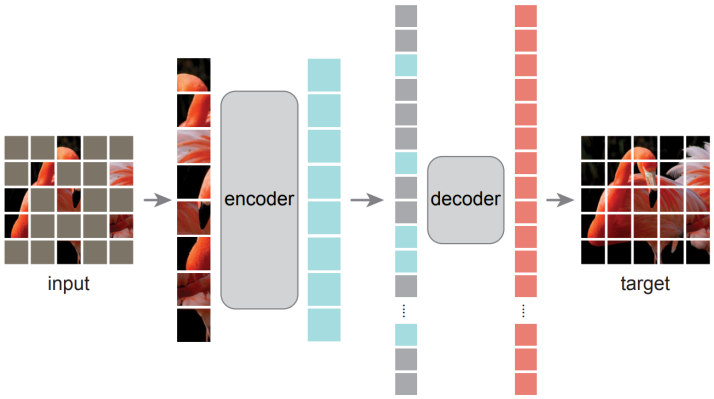


Figure 7 Masked Auto Encoder (He et al., 2021)

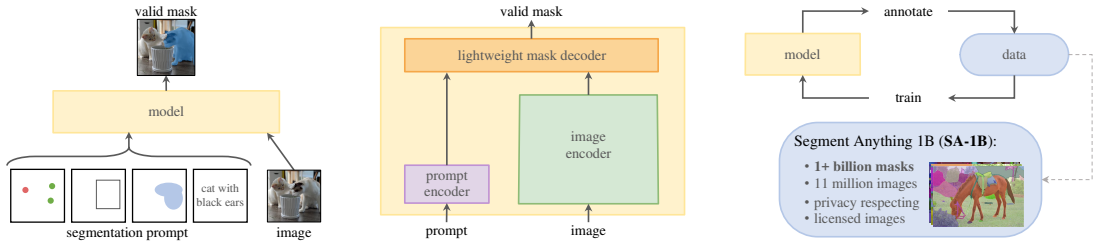
3 Methodology

SAM being a large vision model, is trained gradually in model model-assisted manner.

3.1 SAM: Task, Model, and Data

3.1.1 SAM Task

Prompting is a technique widely used in Natural Language Processing (NLP). The *promptable segmentation task* has a goal of returning a valid segmentation mask given any valid segmentation prompt. A prompt can simply aid in what to segment or identify. Figure 8 shows an example of such a task.



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

Figure 8 Source: SAM Authors

3.1.2 SAM Model

To handle a promptable segmentation task, a model should have the ability to process such prompts. As the model can have multiple types of prompts, there is also a need for ambiguity handling. Furthermore, a model should have a powerful image encoder to process the input image and then a decoder that returns a mask. Figure 8 shows a high-level overview of such a model.

Image Encoder

SAM uses pre-trained MAE Vision Transformer (ViT) Huge with a resolution of 1024 by 1024. The rescaling was done by padding the shorter side. The image encoder runs once per image and is independent of the prompt encoder.

Prompt Encoder

SAM allows two sets of prompts: *sparse* (points, boxes, and text) and *dense* (masks). While points and boxes are represented by positional encodings (Tancik et al., 2020)(which

passes input points through Fourier feature mapping for MLP) text prompt is represented by the encoder of CLIP (Radford et al., 2021) (jointly trains encoders in text to image task). Then dense prompts i.e. masks however are embedded through convolutions. While positional encodings from points and box prompts are summed with embeddings from the text encoder, embeddings from mask prompt is summed element-wise with image embedding.

Mask Decoder

The mask decoder is designed to efficiently map the image embedding, prompt embedding, and an output token to a mask. Hence it uses a modified transformer block with a mask prediction head. It uses prompt self-attention and cross-attention to update all embeddings in prompt-to-image embedding and vice versa. After two blocks, unsampled image embedding is passed to MLP to get foreground probability at each image location.

To handle ambiguity, the model was modified to predict multiple output masks for a single prompt, and during backpropagation, only the minimum loss over masks was returned.

3.1.3 Data

The SA-1B data was created in multiple stages: (1) a model-assisted manual annotation stage, (2) a semi-automatic stage with a mix of automatically predicted masks and model-assisted annotation, and (3) a fully automatic stage in which the model generates masks without annotator input.

In a model-assisted manual annotation stage, a model was trained on a common public segmentation dataset. Then after sufficient data annotation, SAM was retrained. In the end, 4.3M masks from 120k images were collected in this stage.

In a semi-automatic stage, a diversity of masks was added to improve the model's ability. At the end of this stage, an additional 5.9M masks were collected from 180k images.

In the fully automatic stage, annotation was fully automatic and generated 1.1B masks from 11M images.

4 Evaluation

4.1 SAM: Training Procedure from Authors

SAM was trained gradually with increasing annotation and samples. However, the final model was trained on the SA-1B dataset once all data was prepared.

The model was trained with a linear combination of loss functions: 20 * Focal loss (Lin et al., 2018) + Dice loss (Sudre et al., 2017) + mean-square-loss (MSE). Where dice loss and focal loss were used for mask prediction and MSE for IoU prediction head. If p_t is the predicted probability then the Focal loss can be defined as:

$$\text{Focal Loss}(p_t) = -(1 - p_t)^{\gamma} \log(p_t) \tag{4.1}$$

If y is a target value and $\overline{p}$ is a predicted value, then the Dice loss can be defined as:

$$\text{Dice Loss}\,(y, \overline{p}) = 1 - \frac{(2y\overline{p} + 1)}{(y + \overline{p} + 1)} \tag{4.2}$$

The model was trained with Weighted Adam (AdamW) (Loshchilov & Hutter, 2019) optimizer $beta_1 = 0.9$, $\beta_2 = 0.999$. Additionally, the learning rate warm-up was added for 250 iterations, and the step-wise learning rate decay was added too.

Let $\theta_{t,k}$ be a parameter k^{th} parameter at step t, then it's gradient term is $g_{t,k}$ and it's momentum terms can be defined as:

$$m_{t,k} = \beta_1 m_{t-1,k} + (1 - \beta_1) g_{t,k}$$
$$v_{t,k} = \beta_2 v_{t-1,k} + (1 - \beta_2) g_{t,k}^2$$

As m_t (first moment of gradients) and v_t (second moment of gradients) are initialized as vectors of zeros, they tend to bias towards 0 in early timesteps when decay rates are small. Hence authors proposed bias-corrected moment estimates.

$$\hat{m}_{t,k} = \frac{m_{t,k}}{1 - \beta_1^t}$$
$$\hat{v}_{t,k} = \frac{v_{t,k}}{1 - \beta_2^t}$$

Then weighted Adam can be explained as:

$$\theta_{t+1,k} = \theta_{t,k} - \eta \left(\frac{1}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t + w_{t,k} \theta_{t,k} \right), \forall t \tag{4.3}$$

The model was trained for 90k iterations (2 SA-1B epochs) and decreased the learning rate by a factor of 10 at 60k iterations and again at 86666 iterations. While the batch size was 256, the input image size was 1024 by 1024 and no augmentations were applied. The model was trained for 65 hours with 256 Nvidia A100 with 80 GB each.

4.2 U-Net Training

In this project, standard U-Net with ResNet18 (He et al., 2016) as an encoder is trained. ResNet18 is the lightest variant in this family. The residual connection from the encoder to the decoder side makes them strong in their task. The encoder contains multiple layers with decreasing output size as it goes down and the output also is passed to the decoder layer on the other side. Figure 1 shows it in detail. The shallow layers could be seen as a pyramid of features at different scales. U-Net having a significantly smaller number of parameters and still being widely used in industry due to its lightweight nature is a strong candidate to compare again SAM in this dataset.

The input size to the model is 448 height by 224 width and both have to be divisible by 32 to work with ResNet18. While labeling the data, a polygon was used to draw the contour. To make a segmentation mask, the polygon was drawn on a blank image and then filled with 255 while the background was 0. Furthermore, augmentation techniques horizontal flip, vertical flip, Gaussian noise, and perspective transform were also applied. The model was trained for 150 epochs with a batch size of 256 using an Adam optimizer with a learning rate of 1E-4. A single epoch loops through 5 batches of at max 256 samples and at the end of each batch, parameters were updated. With the hope of finding a better model, the model was trained with Dice and Focal loss functions separately. While 80 percent of the labeled images were used for training, the remaining were used for validation. Softmax activation was used on the last layer of the model to predict each pixel as one of foreground or background. While the dataset consisted of masks for infected oranges as well as healthy images, this project combined both masks into a single one by calling it an orange mask. The reason behind doing this is to make comparison easier with SAM. The training was done using a single Nvidia A100. Metrics like training loss, validation loss, and training Intersection over Union (IoU) and validation IoU were also logged.

4.3 Finetuning SAM

This project also fine-tuned the SAM using the box prompt. As SAM authors also used pre-trained ViT, we also use it the same way. Finetuning the model consists of freezing the image encoder and prompt encoder and then only training the mask decoder part.

The input size to the image encoder is 1024 by 1024. The bounding box is extracted from the labeled polygon and passed as a prompt to the mode. This project only used a batch size of 32. Again each batch consists of a maximum of 32 samples and a single epoch consists of 35 batches. This experiment also used Nvidia A100. Metrics like loss and IoU score for both training and validation samples were logged and used for evaluation.

While the U-Net training was done with an input size of 448 by 224, this experiment used 1024 by 1024. However, resizing is done only on the longest side.

For both fine-tuning and training U-Net, the same data is used for training and validation.

5 Results

5.1 Zeroshot Results from SAM Authors

5.1.1 SAM vs RITM on Zero-Shot Single Point Mask Detection

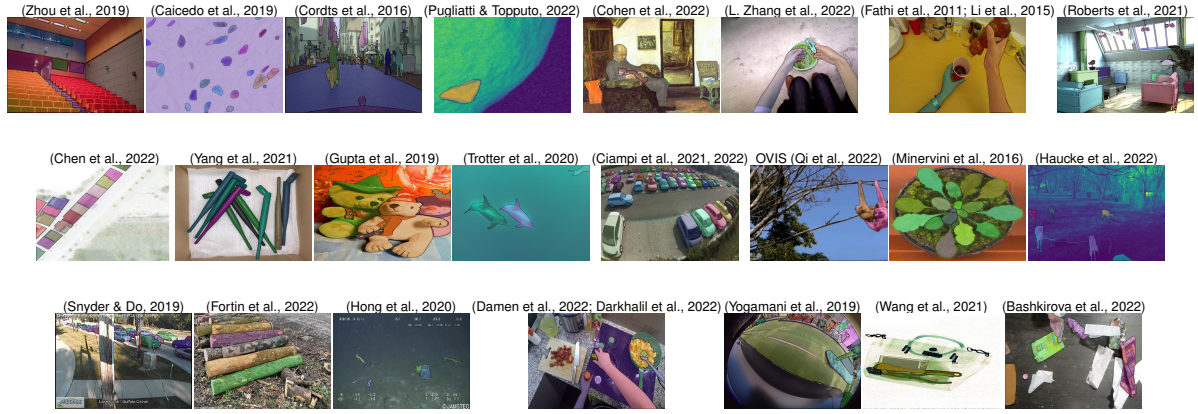
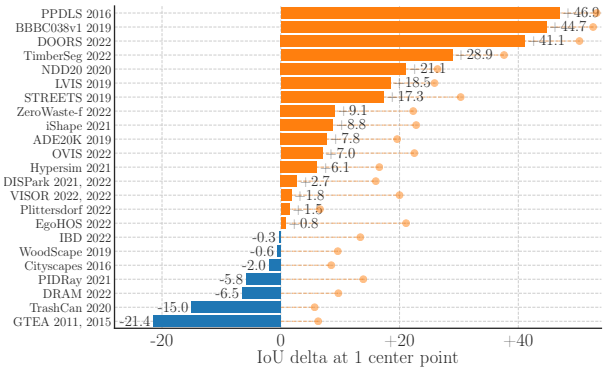


Figure 9 Samples from the 23 diverse segmentation datasets used to evaluate SAM's zero-shot transfer capabilities.

While SAM was not trained on this dataset, SAM outperformed RITM the strongest single point segmented on most of the datasets. While RITM outperformed SAM on some of the datasets, overall SAM outperformed RITM and on 3 datasets, SAM outperformed RITM by more than 40 mIoU.



(a) SAM vs RITM Sofiiuk et al., 2021 on 23 datasets

Figure 10 Mean IoU (58.1) of SAM and the strongest single point segmenter, RITM Sofiiuk et al., 2021.

5.1.2 SAM Zero-Shot Edge Detection

Figure 11 Zero-shot edge prediction on BSDS500. SAM was not trained to predict edge maps nor had access to BSDS images or annotations during training. Source: SAM Authors

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

Table 2 Zero-shot transfer to edge detection on BSDS500. Source: SAM Authors

SAM was not trained for edge detection tasks and also not seen these data but still, authors performed this evaluation to see SAM's ability. Surprisingly SAM showed strong results.

5.1.3 SAM Zero-Shot Object Proposal

Method	mask AR (average recall) @1000						
	all	small	med.	large	freq.	com.	rare
ViTDet-H 2022	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 3 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. Source: SAM Authors

SAM outperforms ViTDet-H on medium and large objects, as well as rare and common objects.

5.1.4 SAM Zero-Shot Instance Segmentation

Method	COCO 2014				LVIS v1 2019			
	AP	AP^S	AP^M	AP^L	AP	AP^S	AP^M	AP^L
ViTDet-H 2022	51.0	32.0	54.3	68.9	46.6	35.0	58.0	66.3
zero-shot transfer methods:								
SAM	46.5	30.8	51.0	61.7	44.7	32.5	57.6	65.5

Table 4 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.

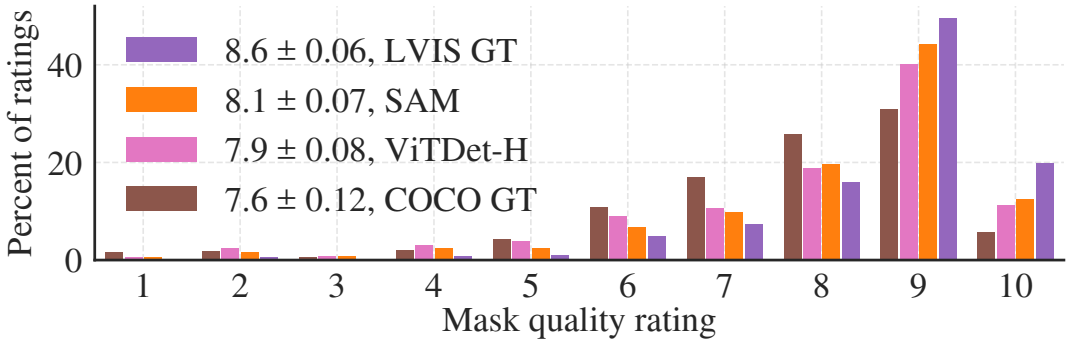


Figure 12 Mask quality rating distribution from human study for ViTDet and SAM, both applied to LVIS ground truth boxes. The legend shows rating means and 95% confidence intervals.

5.1.5 SAM Zero-Shot Text to Mask

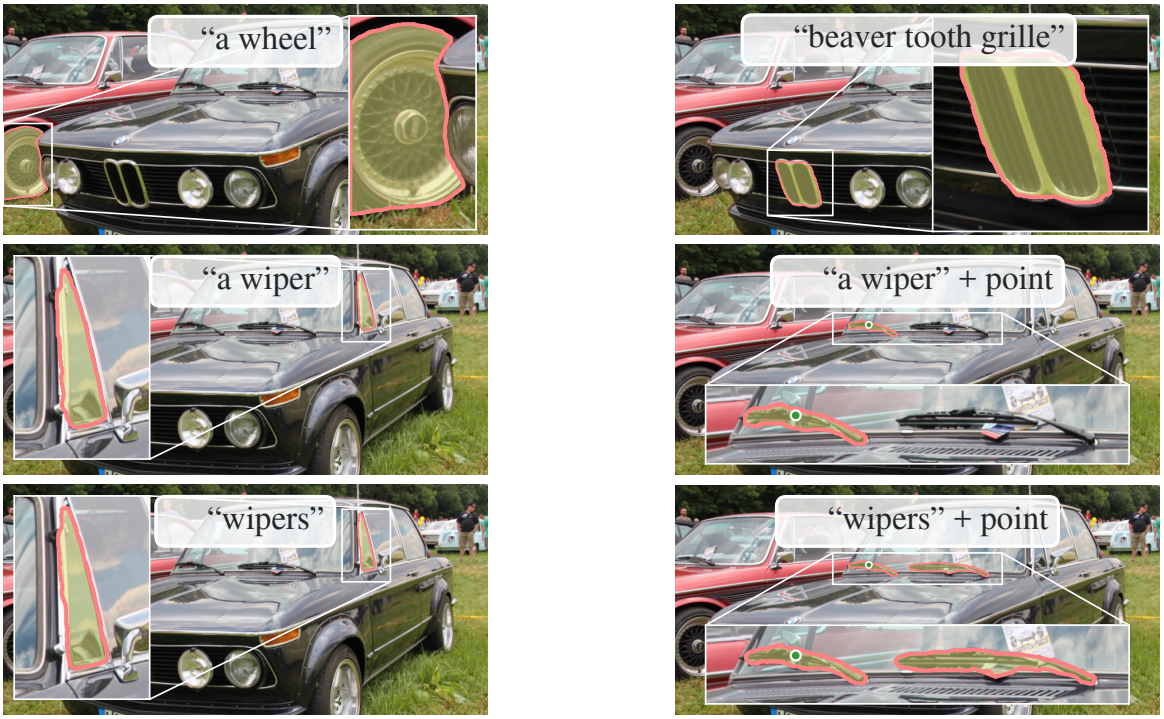


Figure: Zero-shot text-to-mask. SAM can work with simple and nuanced text prompts. When SAM fails to make a correct prediction, an additional point prompt can help. Source: SAM Authors

5.2 SAM's Zero-Shot on Orange Dataset

5.2.1 SAM's Zero-Shot with Box and Point prompt

The image's original size was Avg. (4160, 1800). Figure 13 shows the results from SAM's zero shot using the Box prompt and point prompt. Each row contains from left

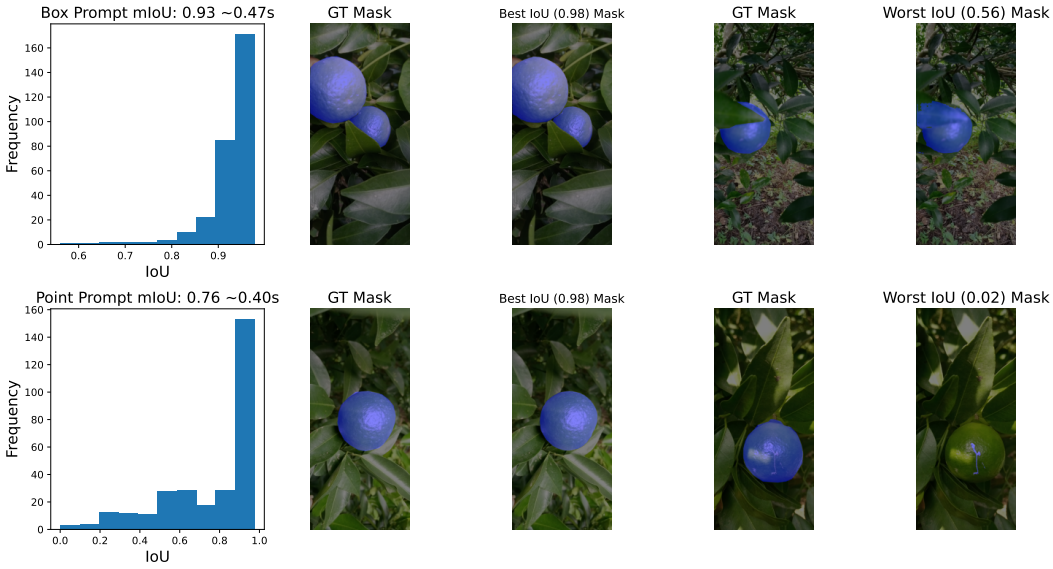


Figure 13 SAM's Zero-Shot Ability on Test Orange Dataset (Pokharel & Acharya, 2024)

to right: the distribution of mIoU on the validation set, the ground truth of the best-predicted mask, the mask from the best predicted, the ground truth from the worst predicted, and the mask from the worst predicted.

For the box prompt, the model shows a mIoU of 0.93, taking nearly 0.47 for each prediction. For the worst prediction, a leaf is behind the orange and SAM predicted the leaf portion as well contributing to a false mask. For the point prompt, the mean IoU is 0.76 and is slightly faster than the box prompt. While both prompts showed an IoU of 0.98 from some, the point prompt made many mistakes and the worst IoU is 0.02. Point prompt seems to be very sensitive to texture in an image and it can be seen on the predicted mask as well.

5.2.2 SAM2 Zero-Shot on Orange Dataset

The same dataset was inference using SAM2 and the plot again contains the distribution of IoU, ground truth mask for best result, best result's mask, ground truth mask for the worst result, and worst result's mask on each row. In both prompting, SAM2 is not better than SAM. Inference was made on Google Collaboratory and shows that box prompting takes slightly more time than point prompting.

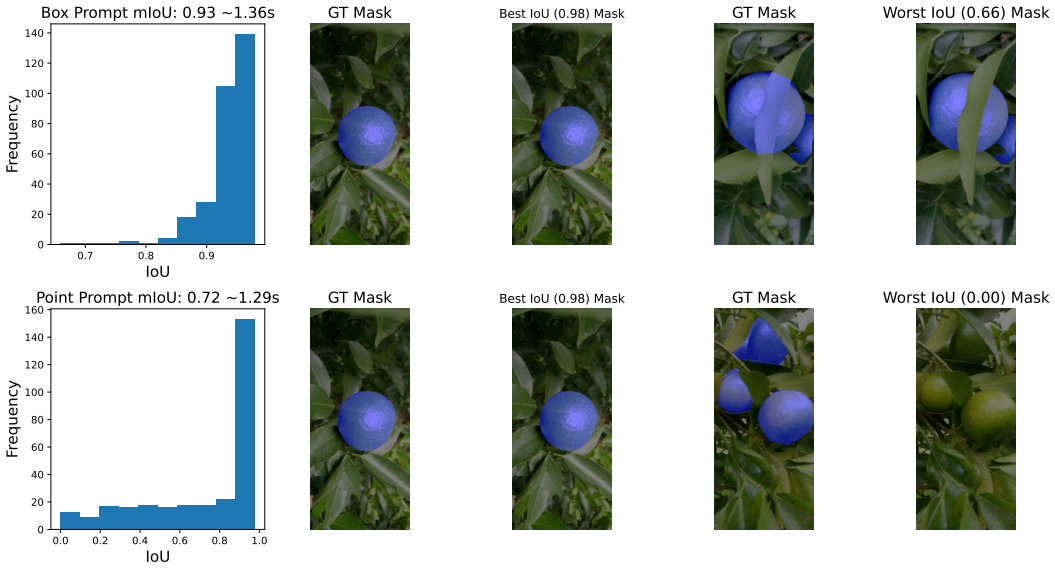


Figure 14 SAM2's Zero-Shot Ability on Test Orange Dataset

5.3 U-Net's Results

5.3.1 Training Curves

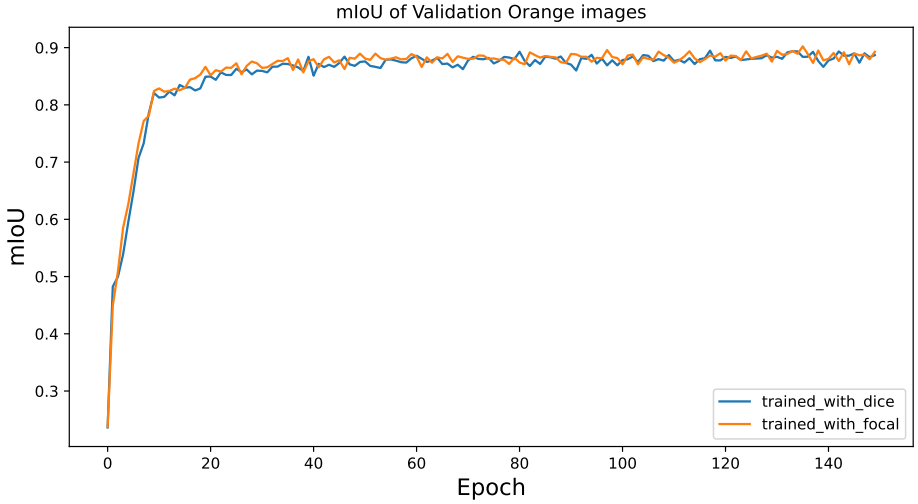


Figure 15 IoU of Validation Dataset using Unet

The same U-Net architecture was trained with different loss functions and showed that their performance was identical.

5.3.2 U-Net's Inference

Figure 16 shows the distribution of IoU for the validation dataset, the ground truth of best prediction, the mask of best prediction, the ground truth of worst prediction, and

Figure 16 U-Net's Performance on the same dataset

the mask of worst prediction. While SAM and SAM2 beat both models on zero-shot using box prompts, U-Net beats the point prompt of SAM and SAM2 by a high margin.

5.4 Finetuned SAM's Results

5.4.1 Training Curves

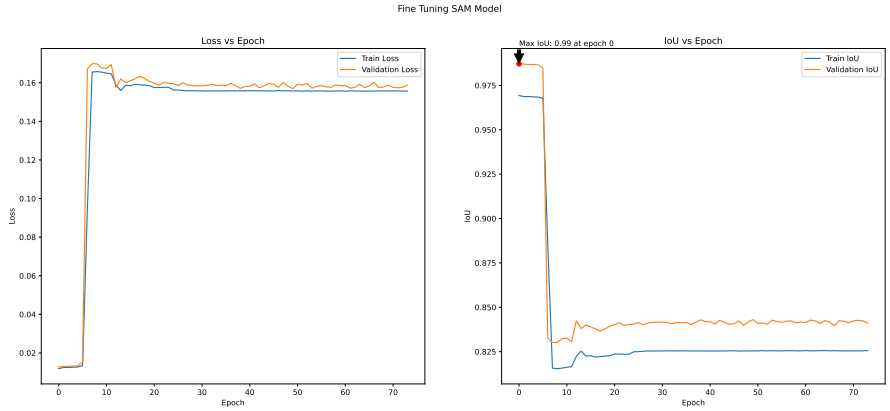


Figure 17 IoU of Validation Dataset using SAM

Even though SAM already performed better than U-Net, SAM's decoder was trained. While the first epoch improved the model's ability to 0.99 mIoU on the validation dataset, it quickly failed to generalize (i.e. started to overfit) and the training was stopped. The entire training took 10+ hours on Nvidia v100 on high-performance computing.

5.4.2 Fine-tuned SAM's Inference

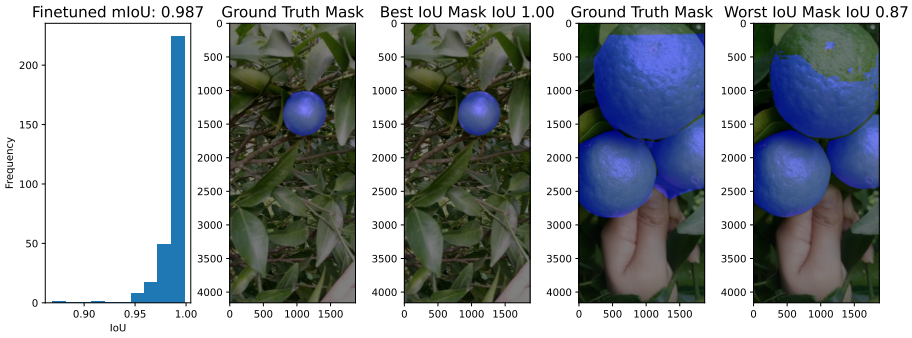


Figure 18 Fine-Tuned SAM on the same dataset

Figure 18 shows the distribution of SAM's predicted IoU on the validation dataset, ground truth for the best mask, predicted best mask, ground truth mask for worst prediction, and worst mask. It seems that the training ground truth itself contained slight mistakes in labeling and that might have been the cause. However, it can be seen by comparing the distribution of IoU (from figure 13, IoU in the range [0.56, 0.93] and 18, IoU in the range [0.87, 1.00] that the fine-tuned model already is better than the original model.

6 Discussion and Conclusion

SAM's zero shot showed promising results on tasks like single-point mask detection, edge detection, object proposal, instance segmentation, and even text-to-mask. Furthermore, SAM's zero shot mIoU based on the box prompt (0.93) was already better than U-Net's mIoU (0.88). However, mIoU from the point prompt (0.76) was smaller than U-Net's mIoU. While SAM2 also showed similar results, it still was not as good as SAM. The reasons for U-Net being outperformed by SAM's zero shot box prompt could be:

- Dataset having some mistakes on labels.
- SAM's input size is 1024 by 1024 while U-Net was trained with an input size of 448 by 224. Having a smaller input size comes with a loss of information.
- SAM has a significantly larger number of parameters than U-Net. SAM has Attention-based encoders/decoders which are better than Convolutional layers for extracting global features from images.

Finetuned SAM significantly improved the mIoU (from 0.93 to 0.987) but the training of a model quickly started to overfit showing the instability in training a huge model with a smaller dataset. While U-Net has significantly lower mIoU than SAM, U-Net is significantly faster due to having a smaller number of parameters and layers than SAM. However, a tradeoff between speed and accuracy is seen by choosing SAM or U-net.

While this project only finetuned SAM on box prompt, in the future, we can fine-tune a SAM using point prompt only or along with box prompt. In addition to that, it might make a model better by training them jointly as well. Considering the input size of U-Net being significantly smaller than SAM's, a fairer comparison between the two might be done after training U-Net with similar input sizes.

References

Alexe, B., Deselaers, T., & Ferrari, V. (2010). What is an object? *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 73–80. https://doi.org/10.1109/CVPR.2010.5540226

Bashkirova, D., Abdelfattah, M., Zhu, Z., Akl, J., Alladkani, F., Hu, P., Ablavsky, V., Calli, B., Bargal, S. A., & Saenko, K. (2022). ZeroWaste dataset: Towards deformable object segmentation in cluttered scenes. *CVPR*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., Rohban, M., Singh, S., & Carpenter, A. E. (2019). Nucleus segmentation across imaging experiments: The 2018 data science bowl. *Nature Methods*.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, *PAMI-8*(6), 679–698.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). Swin-unet: Unet-like pure transformer for medical image segmentation. https://arxiv.org/abs/2105.05537

Chai, S., Jain, R. K., Teng, S., Liu, J., Li, Y., Tateyama, T., & Chen, Y.-w. (2023). Ladder fine-tuning approach for sam integrating complementary network. https://arxiv.org/abs/2306.12737

Chen, J., Xu, Y., Lu, S., Liang, R., & Nan, L. (2022). 3D instance segmentation of MVS buildings. *IEEE Transactions on Geoscience and Remote Sensing*.

Ciampi, L., Santiago, C., Costeira, J., Gennaro, C., & Amato, G. (2021). Domain adaptation for traffic density estimation. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*.

Ciampi, L., Santiago, C., Costeira, J., Gennaro, C., & Amato, G. (2022). Night and day instance segmented park (NDISPark) dataset: A collection of images taken by day and by night for vehicle detection, segmentation and counting in parking areas. *Zenodo*.

Cohen, N., Newman, Y., & Shamir, A. (2022). Semantic segmentation in art paintings. *Computer Graphics Forum*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes dataset for semantic urban scene understanding. *CVPR*.

Damen, D., Doughty, H., Farinella, G. M., Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., & Wray, M. (2022). Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *IJCV*.

Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., & Damen, D. (2022). EPIC-KITCHENS VISOR benchmark: Video segmentations and object relations. *NeurIPS*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N.

(2021). An image is worth 16x16 words: Transformers for image recognition at scale. https://arxiv.org/abs/2010.11929

Fathi, A., Ren, X., & Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. *CVPR*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, *59*(2), 167–181.

Fortin, J.-M., Gamache, O., Grondin, V., Pomerleau, F., & Giguère, P. (2022). Instance segmentation for autonomous log grasping in forestry operations. *IROS*.

Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. *CVPR*.

Haucke, T., Kühl, H. S., & Steinhage, V. (2022). SOCRATES: Introducing depth in visual wildlife monitoring using stereo vision. *Sensors*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners. https://arxiv.org/abs/2111.06377

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hong, J., Fulton, M., & Sattar, J. (2020). TrashCan: A semantically-segmented dataset towards visual detection of marine debris. *arXiv:2007.08097*.

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019). Panoptic segmentation. https://arxiv.org/abs/1801.00868

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. https://arxiv.org/abs/2304.02643

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, Y., Chen, Y., Wang, N., Zhang, Z., Sun, J., & Dai, J. (2022). Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*.

Li, Y., Ye, Z., & Rehg, J. M. (2015). Delving into egocentric actions. *CVPR*.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. https://arxiv.org/abs/1708.02002

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). Microsoft coco: Common objects in context. https://arxiv.org/abs/1405.0312

Lin, T.-Y., Roth, M. M., Dollár, P., & Malik, J. (2014). Microsoft coco: Common objects in context. *European Conference on Computer Vision (ECCV)*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. https://arxiv.org/abs/1711.05101

Minervini, M., Fischbach, A., Scharr, H., & Tsaftaris, S. A. (2016). Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*.

Pokharel, D., & Acharya, R. (2024). Orange infection mask dataset. https://doi.org/10.34740/KAGGLE/DSV/7742783

Pu, J., Zhu, S., Dong, W., Xu, C., & Zhou, B. (2022). Edter: Edge detection with transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3748–3757.

Pugliatti, M., & Topputo, F. (2022). DOORS: Dataset fOr bOuldeRs Segmentation.

Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P., & Bai, S. (2022). Occluded video instance segmentation: A benchmark. *ICCV*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. https://arxiv.org/abs/2103.00020

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI preprint*.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*. https://arxiv.org/abs/2408.00714

Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M. A., Paczan, N., Webb, R., & Susskind, J. M. (2021). Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *ICCV*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. https://arxiv.org/abs/1505.04597

Snyder, C., & Do, M. (2019). STREETS: A novel camera network dataset for traffic flow. *NeurIPS*.

Sofiiuk, K., Petrov, I. A., & Konushin, A. (2021). Reviving iterative training with mask guidance for interactive segmentation. https://arxiv.org/abs/2102.06583

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 240–248). Springer International Publishing. https://doi.org/10.1007/978-3-319-67558-9_28

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. https://arxiv.org/abs/2006.10739

Trotter, C., Atkinson, G., Sharpe, M., Richardson, K., McGough, A. S., Wright, N., Burville, B., & Berggren, P. (2020). NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. *arXiv:2005.13359*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need. https://arxiv.org/abs/1706.03762

Wang, B., Zhang, L., Wen, L., Liu, X., & Wu, Y. (2021). Towards real-world prohibited item detection: A large-scale x-ray benchmark. *CVPR*.

Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. *Proceedings of the IEEE international conference on computer vision*, 1395–1403.

Xu, N., Price, B., Cohen, S., Yang, J., & Huang, T. (2016). Deep interactive object selection. https://arxiv.org/abs/1603.04042

Yang, L., Wei, Y. Z., HE, Y., Sun, W., Huang, Z., Huang, H., & Fan, H. (2021). iShape: A first step towards irregular shape instance segmentation. *arXiv:2109.15068*.

Yogamani, S., Hughes, C., Horgan, J., Sistu, G., Varley, P., O'Dea, D., Uricár, M., Milz, S., Simon, M., Amende, K., et al. (2019). WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving. *ICCV*.

Zhang, K., & Liu, D. (2023). Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*.

Zhang, L., Zhou, S., Stent, S., & Shi, J. (2022). Fine-grained egocentric hand-object segmentation: Dataset, model, and applications. *ECCV*.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2019). Semantic understanding of scenes through the ADE20K dataset. *IJCV*.

Declaration of Academic Integrity

I hereby declare that this report and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly attributed. Where I have quoted from the work of others, the source is always given. I am aware that the report in digital form can be examined for the use of unauthorised aid and in order to determine whether the report as a whole or in parts may amount to plagiarism. I am aware that a false assurance fulfils the elements of fraud in accord with § 10 and § 13 ABM-PO/TechFak and will result in the consequences proclaimed there. This paper was not previously presented to another examination board and has not been published.

Erlangen, 28th September 2024

Ramkrishna Acharya