Improving existing segmentators performance with zero-shot segmentators

Loris Nanni *, Carlo Fantozzi, Alberto Pretto, Daniel Fusaro

Posted Date: 26 July 2023

doi: 10.20944/preprints202307.1729.v1

Keywords: segmentation; deep learning; ensemble; SAM zero-shot segmentator

Article

Improving Existing Segmentators Performance with Zero-Shot Segmentators

Loris Nanni *,‡, Daniel Fusaro ‡, Carlo Fantozzi and Alberto Pretto

Department of Information Engineering, University of Padova, Padova, Italy;
{loris.nanni,carlo.fantozzi,alberto.pretto}@unipd.it; fusarodani@dei.unipd.it
* Correspondence: loris.nanni@unipd.it
‡ These authors contributed equally to this work.

Abstract: This paper explores the potential of using the SAM segmentator to enhance the segmentation capability of known methods. SAM is a promptable segmentation system that offers zero-shot generalization to unfamiliar objects and images, eliminating the need for additional training. The open-source nature of SAM on GitHub allows for easy access and implementation. In our experiments, we aim to improve the segmentation performance by providing SAM with checkpoints extracted from the masks produced by DeepLabv3+, then merging the segmentation masks provided by these two networks. Additionally, we examine the "oracle" method (as upper bound baseline performance), where segmentation masks are inferred only by SAM with checkpoints extracted from ground truth. In addition, we tested in the CAMO datasets an ensemble of PVTv2 transformers; combining the ensemble and SAM yields state-of-the-art performance in that dataset. The results of our study provide valuable insights into the potential of incorporating the SAM segmentator into existing segmentation techniques. We release with this paper the open-source implementation of our method.

Keywords: segmentation; deep learning; ensemble; SAM zero-shot segmentator.

1. Introduction

The task of semantic image segmentation aims to assign each pixel in an image to a specific object class, enabling a more fine-grained understanding of visual content. Over the years, deep learning approaches have significantly advanced the field, demonstrating remarkable achievements in accurately segmenting objects within complex scenes. Among these methods, DeepLabv3+ [1] has garnered substantial attention due to its ability to capture detailed object boundaries while maintaining computational efficiency.

However, a fundamental challenge faced by DeepLabv3+ and other existing methods lies in their ability to generalize to unfamiliar objects and images. When confronted with novel or unseen classes during inference, these models often struggle to produce accurate segmentations, as they lack the necessary knowledge to effectively recognize and segment such objects. This limitation restricts the practical deployment of segmentation models in real-world scenarios, where encountering novel objects is a common occurrence. Recently, two cutting-edge promptable segmentation systems, SAM [2] and SEEM [3], have been proposed. They offer zero-shot generalization capabilities to unfamiliar objects and images without requiring additional training. SAM and SEEM leverage the powerful concept of prompting, which allows users to input specific instructions or hints to guide the model's behavior. We propose to leverage SAM and SEEN alongside DeepLabv3+ to extend its segmentation accuracy when dealing with novel, unconventional objects belonging to known classes. While DeepLabv3+ may not currently represent the state-of-the-art (SOTA) in semantic segmentation (current SOTA is obtained by transformers as [4]), it remains a highly popular and widely used segmentator, serving as a valuable baseline for evaluating the performance of the SAM and SEEM models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

Our approach involves extracting checkpoints from the segmentation masks produced by DeepLabv3+ and utilizing them as prompts for SAM and SEEM. By integrating a zero-shot segmentator (SAM or SEEM) into the segmentation pipeline, we aim to enhance the segmentation capabilities, particularly in scenarios involving unfamiliar objects. To provide a baseline for comparison, we also investigate the method of using checkpoints extracted from ground truth segmentation masks, which we refer to as the "oracle" method.

In this paper, we present a comprehensive analysis of the proposed integration, assessing its impact on segmentation quality, generalization to unfamiliar objects, and computational efficiency. We conduct experiments on benchmark datasets, comparing the performance of DeepLabv3+ with and without SAM and SEEM integration. The best performance is obtained by combining SAM with segmentator masks. Only for the CAMO dataset [5], due to computational problems, we also ran an ensemble of Pyramid Vision Transformer Version 2 (PVTv2) transformers [4], the fusion of which with SAM obtains the new state-of-the-art performance in that dataset. This result is very interesting because it shows that our idea can be used to improve even the current state-of-the-art segmentator. Additionally, we evaluate the effectiveness of the "oracle" method to provide insights into the potential benefits of leveraging ground truth information.

We release with this paper the open-source implementation of our method, freely available at https://github.com/LorisNanni.

The remainder of the paper is organized as follows. Section 2 provides an overview of related work in the field of semantic segmentation and zero-shot learning. Section 3 outlines the methodology, including the architecture of DeepLabv3+, the promptable segmentation systems SAM and SEEM, and the proposed integration approach. Section 4 presents the experimental setup. Section 5 displays the results of the different methods, while in Section 6 a discussion of these results is carried out. Section 7 concludes the paper.

2. Related Work

The related work section provides an overview of the existing literature in the field of semantic segmentation, focusing on three key aspects: deep learning-based segmentation methods, zero-shot learning in segmentation and combining continuous outputs of different classifiers to improve the individual performance. These areas of research have contributed significantly to advancing the field and addressing challenges related to accurate and efficient segmentation.

2.1. Deep Learning-Based Segmentation Methods

Deep learning-based segmentation methods have emerged as powerful techniques for pixel-level object classification in images. These methods leverage the capabilities of deep neural networks to capture intricate details and contextual information, enabling precise segmentation of objects within complex scenes. U-Net [6] is a pioneering deep learning architecture for image segmentation. It comprises a contracting path, which captures context information, and an expansive path, which refines spatial details. U-Net's skip connections facilitate the fusion of feature maps from different resolutions, aiding in accurate pixel-wise predictions and enabling its successful application in medical image segmentation tasks. SegNet [7] is another popular semantic segmentation model, designed to balance segmentation accuracy with computational efficiency. It uses an encoder-decoder architecture with a trainable decoder for pixel-wise predictions. SegNet is known for its compact structure, making it suitable for real-time applications in various domains. DeepLabv3+ [1] is an extension of the DeepLab family, featuring atrous spatial pyramid pooling (ASPP) and encoder-decoder modules. ASPP captures multi-scale contextual information, while the encoder-decoder module refines segmentation boundaries. DeepLabv3+ is widely recognized for its strong performance in large-scale and real-world segmentation tasks. Vision Transformers [8] introduced the Transformer architecture to image classification tasks and have since been adapted to image segmentation. ViT models process images in a patch-based manner and employ self-attention mechanisms to capture long-range

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

dependencies. Despite being initially designed for classification, they have shown promising results in semantic segmentation as well. Many transformer-based segmentation methods have been proposed in recent years, e.g. the Pyramid Vision Transformer (PVT) v2 [4], it is an extension of the ViT architecture, designed to improve efficiency and scalability. PVTv2 combines the advantages of both convolutional and transformer models, leveraging multi-scale representations through pyramid structures. This allows PVTv2 to achieve competitive performance in various vision tasks, including semantic segmentation.

2.2. Zero-Shot Learning in Segmentation

Zero-shot learning plays a crucial role in segmentation tasks, especially when faced with unfamiliar objects during inference. Traditional segmentation models often struggle to generalize to novel or unseen object classes, as they lack the necessary knowledge to effectively recognize and segment such objects. SAM, an image segmentation model [2], stands out as an innovative approach in promptable image segmentation. Trained on a vast dataset comprising over 1 billion segmentation masks, SAM exhibits impressive zero-shot generalization capabilities. It excels in producing high-quality masks even from a single foreground point. The HQ-SAM model [9] is an extension of SAM that introduces a learnable high-quality output token. This addition enhances the model's effectiveness across various segmentation domains, resulting in improved performance. Although SAM may not provide high-quality segmentation directly for medical image data ([10–12]), its masks, features, and stability scores can be utilized to improve medical image segmentation models. SAMAug [13] is a method that leverages SAM to augment image input for commonly-used medical image segmentation models boosting the performance of both CNN and Transformer common models. Another work focusing on medical images SAM performance is [14] which modifies only the SAM conditioning encoder part (mask or set of points). A new encoder is placed at the beginning, trained using the gradients provided from the frozen SAM subsequent architecture, and state-of-the-art levels are reached in many datasets.

2.3. Combining Continuous Outputs

Several approaches have been proposed to combine continuous outputs in the field of image segmentation. For a comprehensive list of combining approaches, please refer to [15,16].

One commonly used technique is the weighted-rule, which aggregates the predicted probability maps or logits from multiple models or methods. This rule has shown effectiveness in various segmentation tasks. Many fusion-based methods have been proposed in recent years; here we describe two to make it more clear how they work. In [17], a multi-label classifier system based on CNN and LSTM networks for ATC prediction is employed. A 1D feature vector from a compound is extracted and transformed into 2D matrices. A CNN model is trained using these matrices to extract a set of new features. In parallel, an LSTM model is trained on the original 1D vector to extract complementary features. These features are then fed into two general-purpose classifiers specifically designed for multi-label classification. Finally, the outputs of the classifiers are fused using the average rule to generate the final prediction results. The average rule is a weighted rule in which each classifier as the same weight.

Another study, [18], focuses on pedestrian classification using deep learning techniques with data from a monocular camera and a 3D LiDAR sensor. The outputs from individual different CNNs are combined by means of learning and non-learning (average, minimum, maximum, and normalized-product) approaches. From the experimental results, the fusion strategies obtains the best results in comparison with the individual CNNs. In particular, the average rule obtains promising results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

3. Methodology

Motivated by the challenges of the datasets used and the capabilities of the zero-shot semantic segmentation methods SAM and SEEM, we attempt to improve the performance of supervised segmentation approaches on such datasets.

In this section, we first illustrate the architecture of the used segmentators used and then we describe the proposed integration approach.

3.1. DeepLabV3+ Architecture

DeepLabv3+ is a state-of-the-art semantic segmentation model that has demonstrated impressive performance in accurately segmenting objects within images. Its architecture builds upon the original DeepLab framework, incorporating several key components to improve both segmentation quality and computational efficiency.

At its core, DeepLabv3+ utilizes a fully convolutional network (FCN) structure, enabling end-to-end training and inference on arbitrary-sized images. The network architecture consists of an encoder-decoder structure that leverages atrous convolutions and atrous spatial pyramid pooling (ASPP) to capture multi-scale contextual information.

The encoder module of DeepLabv3+ is typically based on a pre-trained backbone network, such as ResNet or Xception, which extracts high-level feature representations from the input image. These features are then processed by atrous convolutions, also known as dilated convolutions, which introduce controlled spatial sampling gaps to maintain a large receptive field without excessive downsampling. By using multiple parallel atrous convolutions with different rates, DeepLabv3+ captures multi-scale contextual information, allowing for accurate segmentation of objects at various scales.

The ASPP module further enhances the contextual understanding of the network by incorporating features at multiple scales. It consists of parallel atrous convolutions with different dilation rates, followed by global average pooling. The outputs of these convolutions are then fused to obtain multi-scale feature representations, effectively capturing both local and global contextual information.

To restore the spatial resolution of the feature maps, DeepLabv3+ employs an upsampling decoder module. This module uses bilinear interpolation followed by a 1x1 convolutional layer to upsample the feature maps to the original input resolution. This process ensures that the final segmentation maps are aligned with the original image dimensions.

DeepLabv3+ also introduces a skip connection from the encoder to the decoder module to incorporate low-level details from early layers of the network. This skip connection helps to refine the segmentation boundaries and improve the localization accuracy of the segmented objects.

Overall, DeepLabv3+ combines the strengths of atrous convolutions, ASPP, and skip connections to achieve state-of-the-art segmentation results. Its architecture makes it possible to capture detailed object boundaries while maintaining computational efficiency, making it an excellent candidate for integration with the SAM segmentator.

3.2. Pyramid Vision Transformer Architecture

The Pyramid Vision Transformer (PVT) [4] stands as a transformer network devoid of convolutions. Its core concept revolves around acquiring high-resolution representations from finely-detailed input. The network's depth is paired with a progressively narrowing pyramid, enabling a reduction in computational burden. Additionally, to further curtail the computational overhead, the system incorporates a spatial-reduction attention (SRA) layer. Each PVT network is trained for 50 epochs with a batch size of 8, AdamW is used as optimizer. In this work, we use an ensemble of six nets, combined by average rule, constructed as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

- we apply two different data augmentation, defined in [19]: DA1, base data augmentation consisting in horizontal and vertical flip, 90° rotation; DA2, this technique performs a set of operations to the original images in order to derive new ones. These operations comprehend shadowing, color mapping, vertical or horizontal flipping, and others.
- we apply three different learning strategy: learning rate of 1e-4; learning rate of 5e-4; learning rate of 5e-5 decaying to 5e-6 after 15 epochs.

3.3. SAM Architecture

SAM (Segment Anything Model) [2], is a state-of-the-art vision foundation model specifically designed for promptable image segmentation. It has been trained on the extensive SA-1B dataset, which includes 11 million images and more than 1 billion masks, making it the largest segmentation dataset to date. This vast training set enables SAM to demonstrate exceptional zero-shot generalization capabilities when applied to new data. SAM has proven its ability to generate high-quality masks even with just a single foreground point and has shown robust generalization across various downstream tasks, such as edge detection, object proposal generation, and instance segmentation.

The SAM model consists of three main components: an image encoder, a flexible prompt encoder, and a fast mask decoder. The image encoder utilizes a Vision Transformer (ViT) backbone to process high-resolution 1024x1024 images and generate a 64x64 image embedding. The prompt encoder handles both sparse prompts (e.g., points, boxes, text) and dense prompts (e.g., masks) by converting them into c-dimensional tokens. Finally, the lightweight mask decoder combines the image and prompt embeddings to produce segmentation masks in real-time. This design allows SAM to efficiently handle diverse prompts with minimal computational overhead.

In our study, we evaluated two versions of SAM models: ViT-H and ViT-L. The ViT-Huge and ViT-Large models vary in the complexity of the input image vision transformer-based encoder, with the former having 632M parameters and the latter having 307M parameters.

3.4. SEEM Architecture

SEEM is a promptable, interactive model for Segmenting Everything Everywhere all at once in an image, as described in [3]. The system aims to predict masks and semantic concepts based on the interactions between the input image and multi-modal prompts. To do this, it encodes points, masks, text, boxes or even a similar referred region of another image in the same joint visual-semantic space.

SEEM employs a generic encoder-decoder architecture, which consists of an image encoder that extracts features from the input image, which are then used by the SEEM decoder to predict masks and semantic concepts. Learnable queries interact with visual, text, and memory prompts through a self-attention mechanism.

It is important to note that SEEM model panoptic and interactive segmentation parts are trained with COCO2017 [20] with panoptic segmentation annotations.

3.5. Checkpoints Engineering

We devised several methods for generating checkpoints (prompts). The goal of checkpoint engineering is to investigate whether a specific prompt-generation method can enhance the performance of a prompt-based segmentator. Our system takes as input an image along with its corresponding segmentation mask. The segmentation mask specifically identifies a particular class of objects by separating them from the remaining pixels, which represent the background, note that in this work we are dealing with only two classes of data (background and foreground). This segmentation mask can either be the ground truth mask or the output of a segmentation model. Throughout this paper, we will refer to this segmentation mask as the “source image mask.” It is important to note that source image masks may be composed of several regions ("blobs"), disconnected from each other, masking several portions of the image belonging to the same class of interest (refer to Figure 3 for a visual example). Therefore, our first step is to determine the number of blobs present in the source

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

image mask. Subsequently, we aim to extract at least one checkpoint for each blob in the source image mask, excluding blobs with fewer than 10 pixels that we consider as trivial blobs.

We devised four different methods to generate checkpoints starting from a source image mask, which we refer to as “A”, “B”, “C”, and “D”.

A selects the average coordinates of the blob as the checkpoint. While simple and straightforward, a drawback of this method is that checkpoints may occasionally fall outside the blob region.

B determines the center of mass of the blob as the checkpoint. It is similar to Method A and is relatively simple, but we observed that the extracted checkpoints are less likely to lie outside the blob region.

C randomly selects a point within the blob region as the checkpoint. The primary advantage of this method is its simplicity and efficiency. By randomly selecting a point within the blob, it allows for a diverse range of checkpoints to be generated.

D enables the selection of multiple checkpoints within the blob region. Initially, a grid is created with uniform sampling steps of size b in both the x and y directions. Checkpoints are chosen from the grid if they fall within the blob region. We also applied a modified version of this method that considers eroded (smaller) masks. In Table 1, this modified version is referred to as "bm" (border mode). Erosion is a morphological image processing technique used to reduce the boundaries of objects in a segmentation mask. It works by applying a predefined kernel, in our case, an elliptical-shaped kernel with a size of 10x10 pixels, to the mask. The kernel slides through the image, and for each position, if all the pixels covered by the kernel are part of the object (i.e., white), the central pixel of the kernel is set to white in the output eroded mask. Otherwise, it is set to background (i.e., black). This process effectively erodes the boundaries of the objects in the mask, making them smaller and removing noise or irregularities around the edges. In certain cases, this method (both eroded or not) may fail to find any checkpoints inside certain blobs. To address this, we implemented a fallback strategy described in Algorithm 1, in which Method D is considered in the modified version. The objective of the fallback strategy is to shift the grid of checkpoints horizontally first and then vertically, continuing this process while no part of the grid overlaps with the segmentation mask.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

Table 1. The table presents the evaluation results of different methods on four datasets using the Intersection over Union (IoU) and Dice similarity coefficient (Dice) as evaluation metrics. The methods compared in the table include variations of the ViT-H and ViT-L models, as well as the SEEM method. In the top the results of the DeepLabV3+ model (our baseline) and the state-of-the-art method PVTv2 are reported. In the "Method" column, the word "bm" denotes whether the masks from which checkpoints were extracted were eroded (to avoid checkpoints too close to the borders) or not.

		CAMO						Portrait				Locust-mini				VinDr-RibCXR			
		IoU			Dice			IoU		Dice		IoU		Dice		IoU		Dice	
baseline (DLV3+)		60.63			71.75			97.01		98.46		74.34		83.01		63.48		77.57	
PVTv2		**71.75**			**81.07**														
		oracle		DLV3+		PVTv2		oracle		DLV3+		oracle		DLV3+		oracle		DLV3+	
Model	**Method**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**	**IoU**	**Dice**
SAM - ViT-L	A	48.88	57.63	48.57	58.34	49.19	58.81	79.72	84.96	78.23	83.68	40.58	50.68	37.05	46.99	27.46	42.54	27.23	42.49
	B	48.98	57.69	48.93	58.78	49.12	58.62	80.13	85.38	78.54	83.91	39.47	49.43	37.77	47.68	27.15	42.18	27.23	42.49
	C	47.94	56.06	47.94	57.64	45.73	55.35	79.71	84.10	79.65	83.28	34.74	44.61	33.74	43.39	25.36	40.26	27.30	42.58
	D 10	46.51	58.96	45.65	57.67	44.79	56.77	22.39	31.29	23.14	31.98	36.86	49.41	32.54	44.54	26.53	41.70	26.54	41.71
	D 30	66.01	76.32	60.19	70.37	61.37	71.40	93.08	96.28	92.76	96.09	61.27	72.06	53.87	64.90	26.80	42.08	26.99	42.33
	D 50	67.26	76.54	60.23	69.89	**62.36**	71.70	95.84	97.85	95.65	97.74	63.88	74.13	57.70	67.54	28.06	43.52	27.75	43.23
	D 100	62.56	71.13	53.62	62.27	57.81	66.45	96.55	98.23	96.46	98.18	52.88	62.26	50.01	58.46	27.44	42.82	27.76	43.14
	D 10 bm	52.05	64.15	47.62	59.38	47.30	59.02	23.63	32.74	24.35	33.40	43.35	55.84	36.11	48.20	27.17	42.54	26.54	41.70
	D 30 bm	**68.25**	**77.88**	60.76	**70.66**	62.27	**72.08**	93.20	96.35	92.92	96.19	**65.09**	**75.67**	55.68	66.67	**28.36**	**43.94**	26.98	42.32
	D 50 bm	67.48	76.52	**60.89**	70.47	62.29	71.59	95.88	97.87	95.69	97.76	64.26	74.34	**58.24**	**68.07**	28.03	43.43	**27.84**	**43.36**
	D 100 bm	61.13	69.74	53.44	61.95	57.28	66.06	**96.57**	**98.23**	**96.48**	**98.18**	53.16	62.48	50.23	58.59	26.70	41.95	27.76	43.14
SAM - ViT-H	A	51.96	59.89	49.79	58.78	50.76	59.23	76.60	81.69	75.36	80.27	40.16	50.34	36.42	46.32	26.40	41.41	25.99	40.94
	B	51.98	59.97	50.60	59.50	50.40	58.93	76.18	81.30	76.05	80.93	40.37	50.47	36.89	46.63	26.40	41.41	25.98	40.93
	C	50.20	58.30	49.60	58.35	50.31	59.09	70.77	76.56	70.51	75.45	36.33	45.68	35.77	45.42	25.43	40.27	25.51	40.39
	D 10	68.67	78.72	60.49	71.54	63.00	73.76	91.35	95.25	91.21	95.16	47.17	59.87	39.59	51.94	27.87	43.32	28.03	43.50
	D 30	**77.39**	85.28	**65.90**	**75.58**	**68.69**	**77.74**	95.01	97.41	94.79	97.28	68.47	78.38	63.42	72.69	30.71	46.70	30.67	46.70
	D 50	76.21	83.92	63.69	72.82	68.24	76.87	95.82	97.84	95.65	97.75	**70.21**	**80.00**	**66.21**	**75.67**	32.18	47.97	**31.85**	**47.78**
	D 100	67.53	75.35	55.89	64.06	60.73	68.71	**95.90**	**97.87**	**95.82**	**97.82**	53.50	62.72	50.52	59.00	26.45	41.54	26.10	41.11
	D 10 bm	71.73	81.40	61.41	72.28	63.79	74.33	91.45	95.31	91.29	95.21	52.76	65.02	43.85	55.76	29.76	45.64	28.14	43.65
	D 30 bm	77.27	**85.45**	65.58	75.31	68.31	77.43	95.05	97.43	94.83	97.30	69.73	79.55	63.69	73.02	**32.97**	**48.97**	30.66	46.69
	D 50 bm	74.27	82.33	63.05	72.31	67.68	76.42	95.83	97.85	95.69	97.77	68.65	78.51	65.95	75.47	29.44	44.71	31.72	47.61
	D 100 bm	65.74	73.94	55.50	63.51	60.52	68.54	95.89	97.86	95.81	97.81	53.62	63.02	50.02	58.60	26.23	41.25	26.10	41.11
SEEM	A	48.24	55.46	38.58	44.58	38.12	44.52	93.52	95.73	92.94	95.23	39.20	47.81	35.87	43.31	32.13	48.42	32.12	48.41
	B	48.24	55.46	38.37	44.38	37.82	44.21	93.52	95.73	92.94	95.23	39.20	47.81	35.87	43.31	32.13	48.42	32.12	48.41
	C	44.64	51.09	41.65	47.76	33.97	39.67	92.31	94.45	89.84	92.14	32.98	40.61	26.25	32.73	31.58	47.78	31.90	48.18
	D 10	57.77	65.56	53.82	61.56	45.39	52.46	95.90	97.88	95.87	97.86	**63.93**	**72.13**	58.21	65.93	32.15	48.46	32.05	48.35
	D 30	57.18	64.76	53.37	61.08	52.74	60.16	95.89	97.87	95.86	97.86	61.97	69.96	**59.54**	**67.14**	32.13	48.43	32.12	**48.42**
	D 50	55.09	62.23	51.64	58.74	50.91	58.10	95.85	97.85	95.84	97.84	59.30	67.12	58.35	65.94	31.98	48.26	32.05	48.33
	D 100	51.92	58.68	49.14	55.89	50.38	57.16	95.80	97.83	95.80	97.82	47.42	55.07	43.45	50.74	31.79	48.03	31.83	48.08
	D 10 bm	**58.89**	**66.79**	**54.26**	**62.11**	**52.97**	**60.75**	**95.92**	97.89	95.91	97.88	62.39	70.80	57.33	65.59	**32.17**	**48.48**	**32.12**	48.42
	D 30 bm	57.57	65.25	53.27	61.03	52.87	60.29	95.92	**97.89**	**95.91**	**97.88**	60.79	68.74	58.10	65.68	32.07	48.35	32.11	48.42
	D 50 bm	55.11	62.38	51.62	58.65	51.55	58.77	95.89	97.87	95.89	97.87	58.16	66.20	57.05	64.71	31.94	48.19	32.05	48.33
	D 100 bm	51.78	58.62	48.83	55.53	49.86	56.67	95.81	97.83	95.79	97.82	47.67	55.43	43.82	51.14	31.71	47.94	31.82	48.06
FUSION	D 30	-	-	**64.23**	**74.83**	**73.57**	**82.22**	-	-	97.16	98.54	-	-	75.43	83.74	-	-	61.28	75.90
	D 30 bm	-	-	64.02	74.71	73.31	81.99	-	-	97.16	98.54	-	-	75.41	83.74	-	-	**61.28**	**75.89**
	D 50	-	-	63.72	74.19	73.31	81.88	-	-	97.18	98.55	-	-	75.36	83.77	-	-	60.87	75.59
	D 50 bm	-	-	63.66	74.13	73.14	81.77	-	-	**97.18**	**98.55**	-	-	**75.38**	**83.78**	-	-	60.85	75.58

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

Algorithm 1: Method D with mask erosion and fallback strategy.

```
Input:
mask ;                  // segmentation mask from which to sample the checkpoints
dp ;                                                  // grid sampling step
es ;                                                       // erosion size
Result: checkpoints
checkpoints ← empty list ;                // List to store the selected checkpoints
dx = dy = 0 ;                                  // offsets along x and y directions

/* count the number of non trivial blobs in mask                              */
blobs_num = count_blobs(mask) ;

while checkpoints is empty AND dy < dp do
    /* create a uniformly spaced grid of checkpoints with step dp, horizonal
       offset dx and vertical offset dy                                       */
    grid ← create_uniform_grid(dp, dx, dy) ;
    while checkpoints is empty AND es > 0 do
        /* erode mask with elliptical-shaped kernel of size of 10x10 px       */
        eroded_mask ← erode_mask(mask, 10) ;
        checkpoints ← grid ∩ eroded_mask ;
        /* count the number of blobs having at least a checkpoint inside      */
        extracted_blob_idxs = count_blobs_spanned(checkpoints, mask) ;
        if extracted_blob_idxs ≠ blob_idxs then
            checkpoints ← empty list ;
            es ← es − 1
        end
    end
    dx ← dx + 10 ;
    if dx ≥ dp then
        dx ← dx mod dp ;
        dy ← dy + 10 ;
    end
end
```

4. Experimental Setup

In this section, we describe the datasets used for evaluation, the metrics used for performance evaluation, the baseline extraction process and the implementation details.

Datasets

In our experiments, we have employed four distinct datasets to evaluate the performance of our segmentation methods: CAMO, Portrait, Locust-mini, and VinDr-RibCXR. Each dataset offers unique characteristics and challenges, which ensures a comprehensive evaluation. For each dataset we use split training tests as reported in the literature.

The *CAMO* dataset [5] consists of images with diverse natural scenes containing objects of interest camouflaged in the background. It encompasses various challenging scenarios, such as objects with complex textures and occlusions, making it suitable for evaluating segmentation performance in real-world scenarios. The dataset contains a total of 1250 images, with 1000 for training and 250 for testing.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

The *Portrait* dataset [21] focuses specifically on portrait images of humans. It is designed to evaluate segmentation performance in the context of portrait photography, considering factors such as facial features, skin tones, and background elements. This dataset includes 1447 images for training and 289 images for validation, it can be accessed on https://github.com/HYOJINPARK/ExtPortraitSeg.

The *Locust-mini* dataset [22] contains a collection of 874 images in the training set and 120 test images featuring camouflaged locusts and grasshoppers on various backgrounds. This dataset poses unique challenges due to the complex color patterns and textures of the insects, making it suitable for evaluating segmentation performance in the context of camouflage detection.

The *VinDr-RibCXR* dataset [23] comprises chest X-ray images to detect and segment rib structures. Although it is intended primarily for rib segmentation, we utilized this dataset to evaluate the generalization capability of our proposed methods to medical imaging tasks. This dataset includes a training set of 196 images and a test set of 49 images.

Performance Metrics

To assess the segmentation performance, we employed two commonly used metrics: Intersection over Union (IoU) and Dice similarity coefficient (Dice). For CAMO dataset we computed also Mean Average Error (MAE), weighted F-measure and E-measure, since many papers that segment that dataset also report these performance indicators.

IoU, which was introduced in [24], is defined as:

$$IoU(P,T) = \frac{|P \cap T|}{|P \cup T|} \tag{1}$$

where P is the predicted segmentation mask, T is the ground-truth mask, and the cardinality is the number of pixels. An IoU of 1 corresponds to a perfect prediction, that is, a pixel-perfect overlap between the predicted segmentation mask and the ground truth.

The Dice coefficient [25], it defined as:

$$Dice(P,T) = \frac{2|P \cap T|}{|P| + |T|} \tag{2}$$

measures the overlap between the predicted segmentation mask and the ground truth mask.

The Mean Absolute Error (MAE) metric [26] for 2D image semantic segmentation is a measure of the average absolute difference between the predicted segmentation masks and the ground-truth masks at the pixel level. It is defined as:

$$MAE(P,T) = \frac{\sum_{i=1}^{n} |P_i - T_i|}{n} \tag{3}$$

where n is the number of pixels of an image, and with X_i we indicate the $i-th$ pixel of image X. It quantifies the accuracy of the segmentation model by calculating the average pixel-wise absolute difference between the predicted and true masks for each class in the image. A lower MAE value indicates a better-performing segmentation model with higher accuracy in predicting the correct segmentation boundaries and class labels. MAE is commonly used to evaluate the performance of image segmentation models and compare different approaches in the field of computer vision.

Weighted F-measure [27], is used to calculate the relationship between the precision and recall (it is a weighted approach, we use the same weights suggested by the authors of CAMO). This means that the F-measure considers the imbalance between classes and provides a more comprehensive evaluation of the segmentation model's performance on different categories. A higher weighted F-measure indicates better overall segmentation accuracy, considering the varying class proportions.

E-measure [28], also known as the Enhanced Dice Coefficient, is a performance metric used in binary semantic segmentation tasks to evaluate the accuracy of the model's predictions. It is an

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

extension of the Dice coefficient and incorporates an additional term to penalize false positives and false negatives differently. This adjustment provides a more balanced evaluation, especially in cases of class imbalance, where the standard Dice coefficient might be biased towards the majority class. A higher E-measure value indicates better segmentation accuracy, considering both precision and recall of the model's predictions.

Baseline Extraction

The baseline performance in our experiments is established by evaluating the results of the DeepLabV3+ model, which was trained end-to-end on each of the four datasets in this study. In addition, the PVTv2 segmentator ensemble is applied to the CAMO dataset.

For our experiments, we employed a DeepLabV3+ model with ResNet101 as the backbone architecture. The model was not trained from scratch. We started the training process from pre-trained weights on Pascal VOC2012 Aug dataset [29], which consists of 513x513 RGB images from various categories, such as airplanes, buses, cars, trains, persons, horses, and more of the original Pascal VOC2012 dataset augmented with extra annotations.

The hyperparameters for the training phase (DeepLabV3+) were as follows: an initial learning rate of 0.01, a total of 10 epochs for training, a momentum value of 0.9, L2 regularization with a coefficient of 0.005, a learning rate drop period of 5 epochs, a learning rate drop factor of 0.2, shuffling of training images at every epoch, and the adoption of the SGD (Stochastic Gradient Descent) optimizer. To increase the diversity and generalization capability of the model, data augmentation techniques were employed. Three operations, namely horizontal flip, vertical flip, and 90° rotation, were applied to augment the training set. These augmentation operations create additional variations of the training samples, thereby improving the robustness and adaptability of the trained network.

The baseline performance provided by the DeepLabV3+ model trained on each dataset offers a reference point for evaluating the effectiveness and enhancements achieved by our proposed methods.

Implementation Details

The pre-trained weights for SAM and SEEM were acquired from the official repositories of the projects, hosted on the popular software development platform GitHub.

To evaluate the effectiveness of our proposed methods, checkpoints were computed for every mask in the four datasets utilized in this study. In this way, exactly the same checkpoint prompts were employed for each model to produce segmented masks, enabling a consistent and fair comparison across the different segmentation models.

Refinement Step Description

To further improve the segmentation results, we incorporated a final refinement step. This step involves combining the logits segmentation masks produced by the SAM model and the DeepLabV3+ model using a weighted-rule approach to obtain a final segmentation mask. For the sake of computation time, only for the CAMO dataset we also combine the logits segmentation masks produced by the SAM model and the state-of-the-art PVTv2 model.

The weighted-rule combines the pixel-wise logits values from both models and applies a thresholding operation to generate a binary mask. The fusion process is formally described in Algorithm 2.We have adjusted the weight of the segmentator model to 2. This modification helps to balance the influence of the segmentator in the overall system.

This process is performed for several reasons.

- Combining Complementary Information: The SAM model and the DeepLabV3+ model have different strengths and weaknesses in capturing certain object details or handling specific image characteristics. By combining their logits through the weighted-rule, we can leverage the complementary information captured by each model. This can lead to a more comprehensive

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

representation of the object boundaries and semantic regions, resulting in a more accurate final segmentation.

- Noise Reduction and Consensus: The weighted-rule approach helps reduce the impact of noise or uncertainties in individual model predictions. By combining the logits, the noise or errors inherent in one model's prediction may be offset or diminished by the other model's more accurate predictions. This consensus-based aggregation can effectively filter out noisy predictions.
- Addressing Model Biases: Different models can exhibit biases or tendencies in their predictions due to architectural differences, training data biases, or inherent limitations. The refinement step enables the combination of predictions from multiple models, mitigating the impact of any model biases and enhancing the overall robustness of the segmentation.
- Enhanced Object Boundary Localization: The weighted-rule of logits can help improve the localization and delineation of object boundaries. As the logits from both models contribute to the final segmentation, the refinement step tends to emphasize areas of high consensus, resulting in sharper and more accurate object boundaries. This can be particularly beneficial in cases where individual models may struggle with precise boundary detection.

It is important to note that to make the outputs of SAM (Segment Anything Model) and the segmentators compatible, we scale both of them by multiplying with 255. After scaling, we save them as gray levels and in .jpg format, for sake of storage space. By multiplying both masks by 255, they are brought to the same intensity range (0 to 255), making them suitable for direct comparison and further analysis. Saving them as .jpg files also allows for efficient storage and visualization, as .jpg is a widely used and compressed image format. However, it is crucial to keep in mind that the segmentators have outputs ranging from 0 to 1, while SAM produces outputs beyond this range. The scaling and saving process ensures that the outputs are transformed into a format suitable for further analysis or visualization. Another implementation detail of the refinement step, is the inversion of SAM output masks values. SAM considers values near zero as background and values near 255 as objects, while our DeepLabV3+/PVTv2 code follows the opposite convention, to ensure compatibility between the models, we invert the SAM mask values.

Algorithm 2: Combining continuous outputs of SAM and segmentator model

Input:
SAM_mask ; `// logits based segmentation mask produced by SAM`
P_mask ; `// logits based segmentation mask produced by a segmentator model`
`/* binary segmentation mask produced by the fusion procedure */`
Result: F_mask
`/* load, convert to single precision and normalize SAM_mask */`
SAM_mask ← single(load(SAM_mask)) - 255 ;
SAM_mask ← abs(SAM_mask) ; `// apply absolute value`
SAM_mask ← uint8(SAM_mask) ; `// convert to uint8 precision`

$$\text{F\_mask} \leftarrow \frac{\text{SAM\_mask} + 2 \cdot \text{P\_mask}}{3} \text{ ; } \quad \text{// apply fusion}$$

`/* binarize */`
foreach $F\_mask_i \in F\_mask$ **do**
  if $F\_mask_i < 128$ **then**
    $\text{F\_mask}_i \leftarrow 1$;
  else
    $\text{F\_mask}_i \leftarrow 0$;
  end
end

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

5. Results

First of all, we must note that the scores obtained from SEEM with all our checkpoint prompting strategies are lower than those obtained from SAM, or, at best, comparable. The only exception is the Dice score on the VinDr-RibCXR dataset, where SEEM overtakes SAM ViT-H by a small margin. However, the score remains markedly lower than the baseline. All in all, we can say that SEEM is less promising than SAM, at least with the prompting strategies and datasets we consider in this paper.

SAM beats the DeepLabV3+ on the CAMO dataset, performs similarly on the Portrait dataset, and is significantly worse than the baseline on the Locust-mini and VinDr-RibCXR datasets. This is basically true for both the ViT-L and ViT-H models and for all the three methods considered (oracle, DLV3+, and PVTv2). Our strongest results in this article are obtained with the CAMO dataset, as highlighted in Table 2. The prompts we extract from DeepLabv3+ masks with method D allow SAM to beat DeepLabv3+, that is, to provide a better mask than the one of DeepLabv3+ itself. Most importantly, the fusion between SAM and the ensemble of PVTv2 outperforms the ensemble of PVTv2, which is a current state-of-the-art segmentation approach. For comparison, we report the performance of Explicit Visual Prompting v2 (EVPv2) [30], which has the best state-of-the-art metrics that are available on the famous benchmark dataset sharing platform Paperswithcode [1] . In other words, fusion between PVTv2 and SAM-based segmentators becomes, to the best of our knowledge, the new state of the art on the CAMO dataset. Several qualitative results are illustrated in Figure 8.

On average over the four datasets, neither SAM (SAM ViT-L, SAM ViT-H) nor SEEM beat DeepLabv3+, regardless of the prompts we provided. Not surprisingly, SEEM is worse on average than both SAM ViT-L and SAM ViT-H.

The A, B, and C prompt generation methods are never the top performers. One reason for this fact is shown in Figure 1. Our case for including methods A to C in this paper is to document our experiments and discuss their points of failure.

(a) (b) (c)

Figure 1. An example from the Portrait dataset demonstrating method A can fail to provide SAM with a prompt that is strong enough. (a) Ground truth. (b) Output of SAM with prompt extracted from the DeepLabv3+ mask by method A: a single checkpoint on the nose results in a segmentation output that can be considered semantically valid, but does not capture what was intended. (c) Output of SAM with prompt extracted from the DeepLabv3+ mask by method D ($b = 100$, no mask erosion): a higher number of checkpoints pushes SAM to provide the intended segmentation mask.

[1] https://paperswithcode.com/sota/camouflaged-object-segmentation-on-camo

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

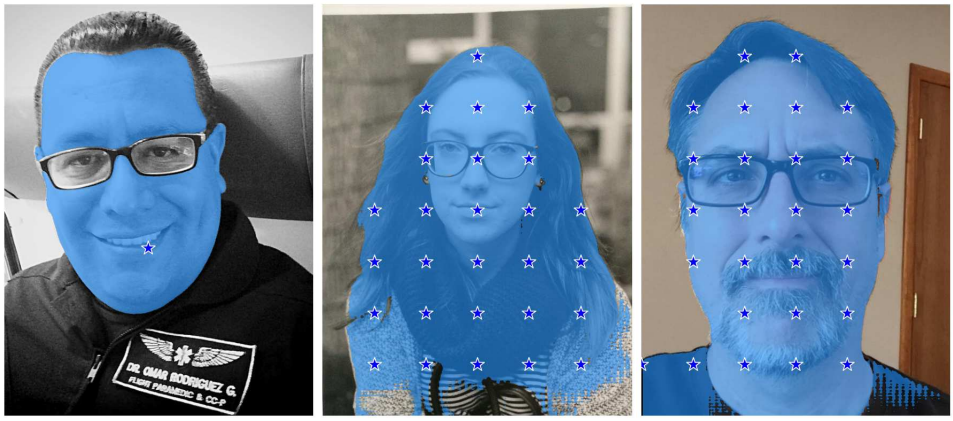
Table 2. Complete results on CAMO dataset. Line 1: DeepLabv3+. Line 2: PVTv2. Line 3: EVPv2 (current State of the Art method on CAMO dataset). Line 4: SAM with prompts obtained from DeepLabv3+ masks with method D ($b = 50$, no mask erosion). Line 5: fusion of the masks just mentioned with DeepLabv3+ masks. Line 6: SAM with prompts obtained from PVTv2 masks with method D ($b = 50$, no mask erosion). Line 7: fusion of the masks just mentioned with PVTv2 masks. ↑ means that higher is better, ↓ means that lower is better.

	IoU ↑	Dice ↑	MAE ↓	F-score ↑	E-measure ↑
DLV3+	60.63	71.75	8.39	75.57	83.04
PVTv2	71.75	81.07	5.74	82.46	89.96
EVPv2 (current SOTA)	-	-	5.80	78.60	89.90
SAM ViT-H D-50 DLV3+	63.69	72.82	12.51	73.86	79.71
SAM ViT-H D-50 DLV3+ fusion	65.00	75.42	7.51	79.17	85.41
SAM ViT-H D-50 PVTv2	68.24	76.87	10.74	77.37	83.05
SAM ViT-H D-50 PVTv2 fusion	**73.31**	**81.88**	**5.60**	**83.32**	**90.00**

No single variation of method D performs consistently the best. Sometimes (e.g., with the CAMO dataset) a low value of the sampling step b works best. Sometimes (e.g., with the Portrait dataset) a higher value of b, which produces fewer checkpoints, is beneficial. Sometimes the best results are obtained with mask erosion and sometimes not, albeit in these cases the difference is smaller and, in some of them, comparable with measurement noise. If a single value of the parameters must be chosen, then $b = 50$ and no mask erosion provide good results on average. This is the value we adopt ourselves in Table 2.

6. Discussion

In this section, we supplement the summary of the results provided in Section 5 with some general remarks about the strengths and failure modes that we encountered in our experiments. The analysis includes a collection of figures that illustrate our assertions, offering a visual demonstration of the capabilities and drawbacks of the proposed prompting methods and the zero-shot segmentators we consider.



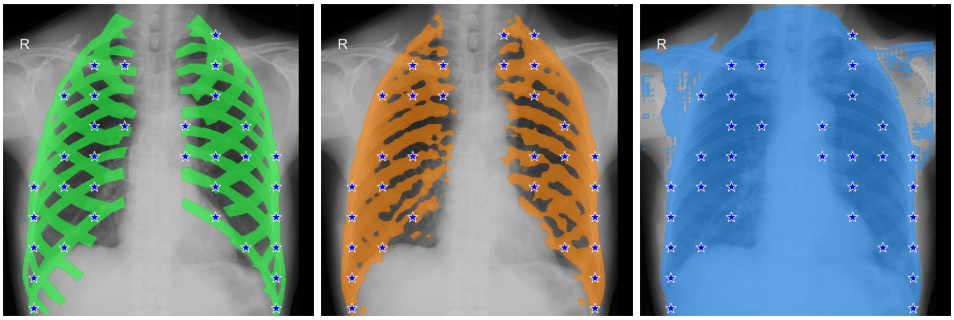
(a) **(b)** **(c)**

Figure 2. Examples from the Portrait dataset where SAM does not provide a semantically reasonable output, regardless of the type of prompting. (a) The output of SAM does not include the portion of skin behind the glasses. (b)(c) The output of SAM incorrectly captures only part of the dress.

The oracle method provides a significant performance boost on the CAMO and Locust-mini datasets, but not on the Portrait and VinDr-RibCXR datasets. This is true for both the SAM and SEEM models. We think the reason may be the same for Portrait and VinDr-RibCXR. The performance on the former dataset is so good that the improvement in prompting with the oracle method produces negligible effects. The performance on the latter dataset is so bad, that is, the models are so inadequate

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

to segment anatomical structures such as ribs (Figure 3), that changes in the prompting do not make a difference.



(a) **(b)** **(c)**

Figure 3. A typical failure mode of SAM on the VinDr-RibCXR dataset: basically, SAM does not capture any anatomical structure. (a) Ground truth and corresponding checkpoints extracted by method D (oracle method, $b = 50$, no mask erosion). (b) Mask from DeepLabv3+ and corresponding checkpoints extracted by method D ($b = 50$, no mask erosion). (c) The output of SAM when prompted with oracular checkpoints from (a). It is apparent that the mask is much worse than that provided by DeepLabv3+.

According to the results in Table 1, the added complexity of the SAM ViT-H model with respect to the SAM ViT-L model does not make a radical difference in the Portrait dataset. As a matter of fact, the smaller model performs slightly better than the larger one. We believe that, similarly to the case we previously discussed, performance on the Portrait dataset is already so high with ViT-L that a wall has been hit. It is difficult to overcome such a wall by simply increasing the complexity of the model. If we look at the other three datasets, however, we observe that SAM ViT-H performs significantly better than SAM ViT-L.

Adding to the previous discussion on the strong performance of SAM and SEEM on Portrait, we must say that another limitation to the growth of the performance metrics on this dataset may be the inaccuracy of the ground truth. Indeed, we found images where

1. the ground truth is wrong (see, e.g., Figure 4a, swimming cap and beard) or, at least, semantically questionable (e.g., Figure 5a, belts and blanket);
2. SAM or SEEM provide a mask that is more accurate than the ground truth (see, e.g., Figures 4b and 5c).

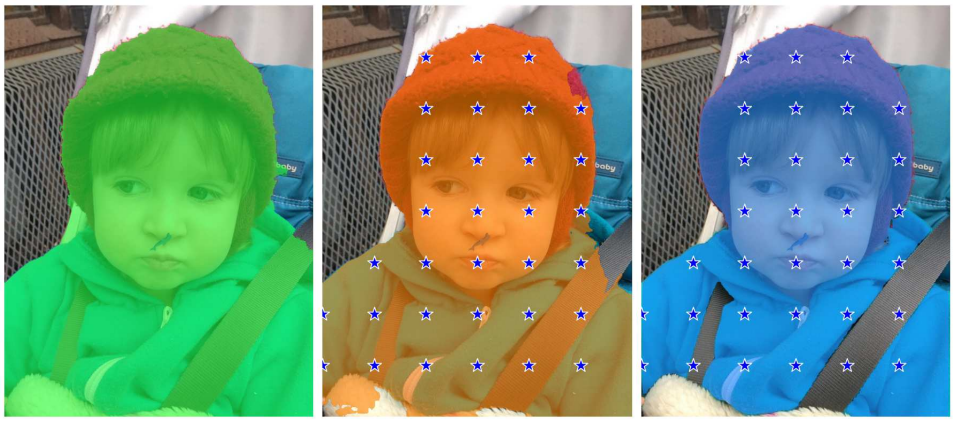
These images lower the IoU and Dice scores for SAM and SEEM.

(a) **(b)**

Figure 4. An example from the Portrait dataset that shows method A operating under the same conditions described in Figure 1, but providing a strong enough hint. (a) Ground truth. (b) Output of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

SAM with prompt extracted from the DeepLabv3+ mask by method A: in this case, a single checkpoint on the nose results in the correct segmentation output by SAM. Indeed, the mask provided by SAM is better than the ground truth in the beard region.



(a) **(b)** **(c)**

Figure 5. An example from the Portrait dataset where the output of SAM, with suitable prompting obtained from the DeepLabv3+ mask, is arguably better than the DeepLabV3+ mask itself. (a) Ground truth. (b) Mask from DeepLabv3+ and checkpoints extracted from such mask by method D ($b = 100$, no mask erosion). (c) Mask provided by SAM. It can be seen that SAM ignores the belts, albeit hinted to include them, and the blanket. It can be argued that this choice is semantically better than the output of DeepLabv3+ and the ground truth, where the mask includes objects that are not part of the person.

(a) **(b)** **(c)**

Figure 6. An example from the CAMO dataset where SAM provides a better mask than DeepLabv3+. (a) Ground truth. (b) Mask from DeepLabv3+ and corresponding checkpoints extracted by method D ($b = 30$, no mask erosion). (c) The output of SAM when prompted with the aforementioned checkpoints (not shown for clarity).

(a) **(b)** **(c)**

Figure 7. A failure mode of SAM on the CAMO dataset: despite strong prompting, SAM fails to segment a common pet. (a) Ground truth. (b) Mask from DeepLabv3+ and corresponding checkpoints extracted by method D ($b = 30$, no mask erosion). (c) The output of SAM when prompted with the aforementioned checkpoints (not shown).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2023 doi:10.20944/preprints202307.1729.v1

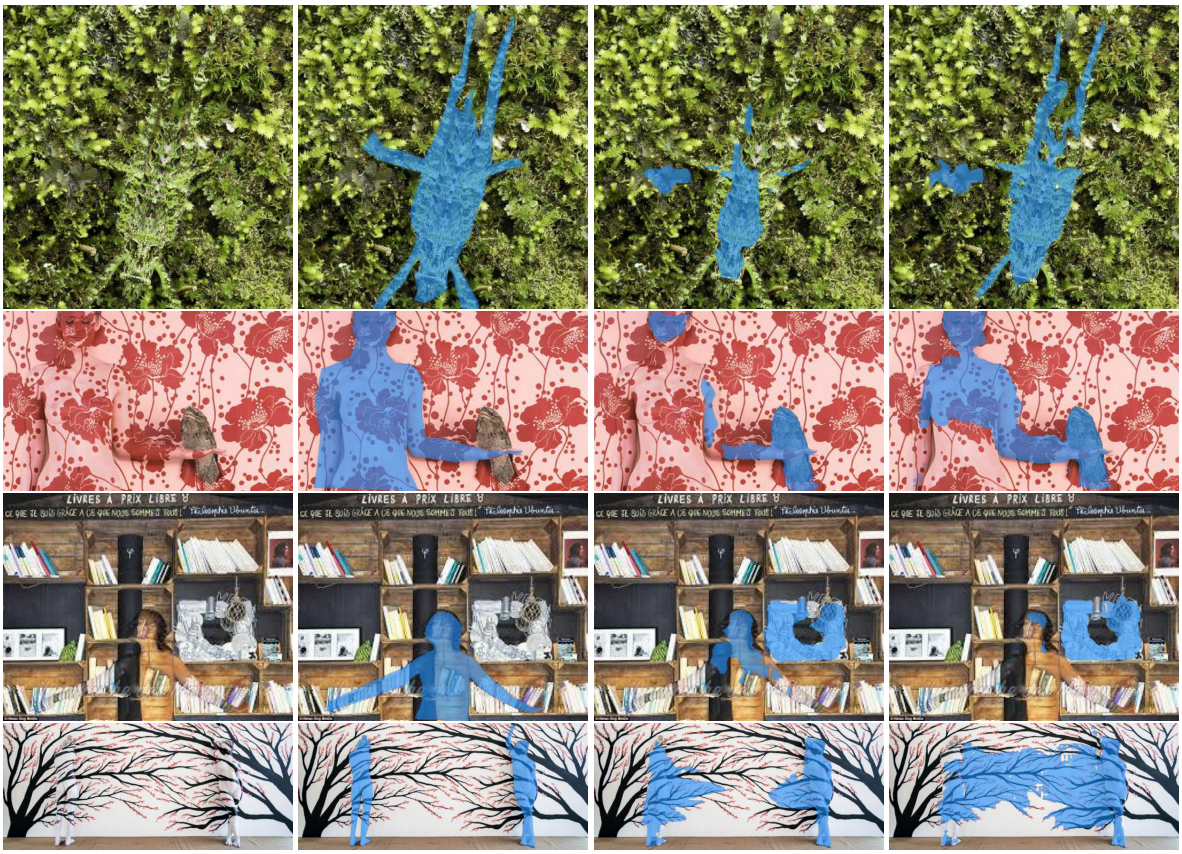


Figure 8. Arranged from left to right are: the source images, the ground truth masks, the binary masks obtained by the PVTv2 method, the binary masks obtained by fusing the logits masks of PVTv2 and SAM with checkpoints sampled from PVTv2. The first two rows demonstrate significant improvements in segmentation, while the last two rows illustrate instances where the fusion process did not yield desired results.

7. Conclusions

In summary, the results of our experiments demonstrate that the proposed variations of SAM, based on the ViT-L and ViT-H models, can lead to segmentation improvements over the original DeepLabV3+ and PVTv2 masks, even beating the state of the art on the CAMO dataset. These findings highlight the potential of zero-shot segmentators such as SAM and SEEM to advance the state of the art for semantic segmentation, and provide valuable insights for further analysis. To this aim, future work needs to consider other prompting strategies, chiefly those based on bounding boxes and text prompts, on a larger number of datasets. We remark that prompting SAM with text, albeit explored in [2], is currently inaccessible with the source code that is publicly available.

Author Contributions: "Conceptualization, L.N.; methodology, C.F., D.F. and L.N.; software, C.F., D.F and L.N.; writing—original paper, C.F., D.F., L.N. and A.P. All authors have read and agreed to the published version of the manuscript."

Funding: This research received no external funding.

Data Availability Statement: All the resources required to replicate our experiments are available at https://github.com/LorisNanni (accessed on 22 July 2023

Acknowledgments: Through their GPU Grant Program, NVIDIA donated the GPU that was used to train the CNNs presented in this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision – ECCV 2018; Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y., Eds.; Springer International Publishing: Cham, 2018; pp. 833–851.
2. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything, 2023, [arXiv:cs.CV/2304.02643].
3. Zou, X.; Yang, J.; Zhang, H.; Li, F.; Li, L.; Gao, J.; Lee, Y.J. Segment Everything Everywhere All at Once, 2023, [arXiv:cs.CV/2304.06718].
4. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved baselines with Pyramid Vision Transformer. *Computational Visual Media* **2022**, *8*, 415–424. https://doi.org/10.1007/s41095-022-0274-8.
5. Le, T.N.; Nguyen, T.V.; Nie, Z.; Tran, M.T.; Sugimoto, A. Anabranch Network for Camouflaged Object Segmentation. *Journal of Computer Vision and Image Understanding* **2019**, *184*, 45–56.
6. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015; Navab, N.; Hornegger, J.; Wells, W.M.; Frangi, A.F., Eds.; Springer International Publishing: Cham, 2015; pp. 234–241.
7. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.
8. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, 2021.
9. Ke, L.; Ye, M.; Danelljan, M.; Liu, Y.; Tai, Y.W.; Tang, C.K.; Yu, F. Segment Anything in High Quality. *arXiv:2306.01567* **2023**.
10. Wu, J.; Fu, R.; Fang, H.; Liu, Y.; Wang, Z.; Xu, Y.; Jin, Y.; Arbel, T. Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation. *arXiv preprint arXiv:2304.12620* **2023**.
11. Cheng, D.; Qin, Z.; Jiang, Z.; Zhang, S.; Lao, Q.; Li, K. SAM on Medical Images: A Comprehensive Study on Three Prompt Modes, 2023, [arXiv:cs.CV/2305.00035].
12. Hu, C.; Xia, T.; Ju, S.; Li, X. When SAM Meets Medical Images: An Investigation of Segment Anything Model (SAM) on Multi-phase Liver Tumor Segmentation. *arXiv e-prints* **2023**, p. arXiv:2304.08506, [arXiv:eess.IV/2304.08506]. https://doi.org/10.48550/arXiv.2304.08506.
13. Zhang, Y.; Zhou, T.; Wang, S.; Liang, P.; Chen, D.Z. Input Augmentation with SAM: Boosting Medical Image Segmentation with Segmentation Foundation Model, 2023, [arXiv:cs.CV/2304.11332].
14. Shaharabany, T.; Dahan, A.; Giryes, R.; Wolf, L. AutoSAM: Adapting SAM to Medical Images by Overloading the Prompt Encoder, 2023, [arXiv:cs.CV/2306.06370].
15. Kuncheva, L.I. Diversity in multiple classifier systems. *Information Fusion* **2005**, *6*, 3–4. Diversity in Multiple Classifier Systems, https://doi.org/https://doi.org/10.1016/j.inffus.2004.04.009.
16. Kittler, J. Combining classifiers: A theoretical framework. *Pattern Analysis and Applications* **1998**, *1*, 18–27.
17. Nanni, L.; Brahnam, S.; Lumini, A. Ensemble of Deep Learning Approaches for ATC Classification. In Proceedings of the Smart Intelligent Computing and Applications; Satapathy, S.C.; Bhateja, V.; Mohanty, J.R.; Udgata, S.K., Eds.; Springer Singapore: Singapore, 2020; pp. 117–125.
18. Melotti, G.; Premebida, C.; Goncalves, N.M.M.d.S.; Nunes, U.J.C.; Faria, D.R. Multimodal CNN Pedestrian Classification: A Study on Combining LIDAR and Camera Data. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3138–3143. https://doi.org/10.1109/ITSC.2018.8569666.
19. Nanni, L.; Lumini, A.; Loreggia, A.; Formaggio, A.; Cuza, D. An Empirical Study on Ensemble of Segmentation Approaches. *Signals* **2022**, *3*, 341–358. https://doi.org/10.3390/signals3020022.
20. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014); Springer, , 2014; pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.

21. Kim, Y.W.; Byun, Y.C.; Krishna, A.V.N. Portrait Segmentation Using Ensemble of Heterogeneous Deep-Learning Models. *Entropy* **2021**, *23*. https://doi.org/10.3390/e23020197.
22. Liu, L.; Liu, M.; Meng, K.; Yang, L.; Zhao, M.; Mei, S. Camouflaged locust segmentation based on PraNet. *Computers and Electronics in Agriculture* **2022**, *198*, 107061. https://doi.org/10.1016/j.compag.2022.107061.
23. Nguyen, H.C.; Le, T.T.; Pham, H.H.; Nguyen, H.Q. VinDr-RibCXR: A benchmark dataset for automatic segmentation and labeling of individual ribs on chest X-rays. In Proceedings of the 2021 International Conference on Medical Imaging with Deep Learning (MIDL 2021); , 2021.
24. Rahman, M.A.; Wang, Y. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In Proceedings of the International Symposium on Visual Computing (ISVC 2016); Springer, , 2016; pp. 234–244. https://doi.org/10.1007/978-3-319-50835-1_22.
25. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Cardoso, J.M. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017; Springer, , 2017; pp. 240–248. https://doi.org/10.1007/978-3-319-67558-9_28.
26. Perazzi, F.; Krähenbühl, P.; Pritch, Y.; Sorkine-Hornung, A. Saliency filters: Contrast based filtering for salient region detection. *2012 IEEE Conference on Computer Vision and Pattern Recognition* **2012**, pp. 733–740.
27. Margolin, R.; Zelnik-Manor, L.; Tal, A. How to Evaluate Foreground Maps. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 248–255. https://doi.org/10.1109/CVPR.2014.39.
28. Fan, D.P.; Cheng, M.M.; Liu, Y.; Li, T.; Borji, A. Structure-measure: A New Way to Evaluate Foreground Maps. In Proceedings of the IEEE International Conference on Computer Vision, 2017.
29. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* **2010**, *88*, 303–338. https://doi.org/10.1007/s11263-009-0275-4.
30. Liu, W.; Shen, X.; Pun, C.M.; Cun, X. Explicit Visual Prompting for Universal Foreground Segmentations, 2023, [arXiv:cs.CV/2305.18476].