

Article

ENSeg: A Novel Dataset and Method for the Segmentation of Enteric Neuron Cells on Microscopy Images

Gustavo Zanoni Felipe ^{1,2}, Loris Nanni ^{2,*}, Isadora Goulart Garcia ³, Jacqueline Nelisis Zanoni ³
and Yandre Maldonado e Gomes da Costa ¹

¹ Department of Informatics, State University of Maringá, Maringá 87020-900, Brazil; gzf1996@gmail.com (G.Z.F.); yandre@din.uem.br (Y.M.e.G.d.C.)

² Department of Information Engineering, University of Padova, 35131 Padova, Italy

³ Department of Morphological Sciences, State University of Maringá, Maringá 87020-900, Brazil; ra126805@uem.br (I.G.G.); jnzanoni@uem.br (J.N.Z.)

* Correspondence: loris.nanni@unipd.it

Featured Application: The dataset and methods presented here represent significant advancements, facilitating progress in Enteric Nervous System imaging analysis and broader biomedical research.

Abstract: The Enteric Nervous System (ENS) is a dynamic field of study where researchers devise sophisticated methodologies to comprehend the impact of chronic degenerative diseases on Enteric Neuron Cells (ENCs). These investigations demand labor-intensive effort, requiring manual selection and segmentation of each well-defined cell to conduct morphometric and quantitative analyses. However, the scarcity of labeled data and the unique characteristics of such data limit the applicability of existing solutions in the literature. To address this, we introduce a novel dataset featuring expert-labeled ENC called ENSeg, which comprises 187 images and 9709 individually annotated cells. We also introduce an approach that combines automatic instance segmentation models with Segment Anything Model (SAM) architectures, enabling human interaction while maintaining high efficiency. We employed YOLOv8, YOLOv9, and YOLOv11 models to generate segmentation candidates, which were then integrated with SAM architectures through a fusion protocol. Our best result achieved a mean DICE score (mDICE) of 0.7877, using YOLOv8 (candidate selection), SAM, and a fusion protocol that enhanced the input point prompts. The resulting combination protocols, demonstrated after our work, exhibit superior segmentation performance compared to the standalone segmentation models. The dataset comes as a contribution to this work and is available to the research community.

Keywords: cell segmentation; enteric nervous system; instance segmentation; segment anything model



Academic Editors: Hyeonjoon Moon and Lien Minh Dang

Received: 10 December 2024

Revised: 9 January 2025

Accepted: 15 January 2025

Published: 21 January 2025

Citation: Felipe, G.Z.; Nanni, L.; Garcia, I.G.; Zanoni, J.N.; Costa, Y.M.e.G.d. ENSeg: A Novel Dataset and Method for the Segmentation of Enteric Neuron Cells on Microscopy Images. *Appl. Sci.* **2025**, *15*, 1046. <https://doi.org/10.3390/app15031046>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Enteric Nervous System (ENS) is a specialized division of the autonomic nervous system, intricately woven throughout the gastrointestinal tract. Often referred to as the “second brain”, the ENS operates independently of the central nervous system, showcasing its remarkable autonomy. Tasked with regulating digestive processes, it coordinates a variety of movement patterns within the digestive tract. These include rapid propulsion of contents (peristalsis), mixing movements (segmentation), slow propulsion, and retropulsion, the latter aiding in the expulsion of harmful substances, such as during vomiting [1].

The ENS consists of two primary networks, commonly referred to as neural plexuses. The first, the myenteric (or Auerbach's) plexus, is situated between the longitudinal and circular muscle layers. The second, the submucosal (or Meissner's) plexus, resides within the submucosa. The myenteric plexus governs gastrointestinal motility, while the submucosal plexus primarily regulates secretion and local blood flow within the gastrointestinal tract [1]. Within the ENS, two main cell types predominate: Enteric Neuron Cells (ENCs) and Enteric Glial Cells (EGCs). These cells are essential for maintaining the homeostasis of gastrointestinal tract (GIT) functions [2].

In preclinical research, the analysis of ENS cells is primarily conducted to evaluate new methodologies and techniques in animal models before progressing to human trials. This approach helps mitigate the risk of mortality or permanent disabilities in human subjects. Neurogastroenterology researchers often rely on images of ENCs and EGCs to examine the visual effects of various diseases and assess the efficacy of emerging medical treatments. Images of enteric cells are particularly valuable in such studies due to the significant impact of chronic degenerative diseases (e.g., diabetes mellitus [3–5], rheumatoid arthritis [6,7], cancer [8–11], etc.) on these cells. This versatility enables researchers to use the same source material across multiple research disciplines, broadening the scope and applicability of their findings.

Chronic degenerative diseases significantly affect both the shape and quantity of ENCs and EGCs. For that reason, researchers conduct morphometric and quantitative analyses to evaluate the health status of target animals. However, these processes are predominantly manual, labor-intensive, and highly repetitive, making them both exhausting and time-consuming. This highlights the critical need for developing computational models that can perform these tasks automatically and efficiently [12].

The described problem here can be reduced to an instance segmentation task. By segmenting each cell individually, we could ease the process of analyzing them on both their shape and quantity. Although previous works have been conducted aiming to classify images from the ENS [12,13], the lack of available data with expert-annotated segmentation masks increases the challenge of automatically performing such a task.

Keeping that in mind, in this work, we present a novel dataset from the ENS, called Enteric Nervous System-Segmentation (ENSeg). This one represents an upgraded subset of the ENS dataset published in [13], containing images of ENC labeled for instance segmentation. In total, 187 images were annotated by the same laboratory in which they were originally obtained, presenting 9709 individually tagged ENCs. Datasets annotated for segmentation tasks in the ENS domain are notably scarce in the literature. To the best of our knowledge, this is the first dataset to be openly accessible to the research community.

In addition to the dataset, we present a novel approach that aims to deal with the proposed task and present baseline results to ENSeg. Our method consists of using instance segmentation models, e.g., YOLOv8 [14], for candidate generation and using user prompts to select them. The generated bounding boxes of the selected candidates are then used to enhance the quality of input prompts fed to the prompt-guided segmentation models, e.g., Segment Anything Model (SAM) [15,16]. Four different protocols are projected and tested aiming to combine both models' outputs, to refine the final segmentation masks further.

Our results are promising, by fine-tuning a YOLOv8 and using SAM, we reached a final mDICE of 0.7877 for our six test sets (split by leave-one-out cross-validation), a value which represented a mean increase of 26.74% and 19.14% from our baselines, i.e., standalone SAM point prompt and YOLOv8. Such results were obtained by a fusion protocol that segmented candidates selected through point-prompting to automatically enhance the prompt quality, later fed to SAM-b.

In this work, we address a significant gap in the literature by proposing and introducing a novel contribution to the ENS research domain. Specifically, we present the following key advancements:

1. **A Novel Dataset (ENSeg Dataset):** We propose the ENSeg, the first publicly available expert-annotated dataset for segmentation tasks in the ENS domain. This dataset fills a critical void in the field, providing a foundational resource for researchers seeking to develop and benchmark segmentation techniques. Dataset and sources will be made available at <https://github.com/gustavozf/seg-lib> (accessed on 9 December 2024).
2. **Baseline Results Using State-of-the-Art Methods:** To establish a benchmark for future research, we provide baseline results obtained using modern state-of-the-art segmentation methods. These methods leverage advanced architectures and techniques, ensuring that the results are both relevant and representative of current technological capabilities.
3. **A Hybrid Approach for Enhanced Segmentation:** We introduce a novel approach that combines the strengths of instance segmentation models with prompt-guided segmentation methods. This hybrid technique is designed to exploit the complementary benefits of both methodologies, paving the way for more robust and adaptable solutions in segmentation tasks.

By addressing the scarcity of datasets and leveraging cutting-edge methods, our work not only closes an existing gap in the ENS domain but also lays the groundwork for further innovation and exploration.

2. Related Works

This section presents a brief overview of the literature. Some of the state-of-the-art methods and architectures are presented for both traditional and promptable medical image segmentation architectures.

2.1. Segmentation in Medical Images

Different studies have been proposed in the literature aiming to develop semantic segmentation architectures for medical images. The U-Net [17] was an influential work that explored a convolutional encoder–decoder neural architecture for the semantic segmentation of medical images. A later work [18] showed that it could be naturally extended to cell counting, detection, and morphometry. Different variations have been proposed in the literature for the U-Net for several unique medical image tasks, such as Attention U-Net [19], U-Net++ [20], ResUNet [21], ResUNet++ [22], DoubleU-Net [23], CU-Net [24], ADU-Net [25], etc.

Following the great success of Transformer networks [26], authors started to explore their usage on a series of new applications. As presented in the work of TransUNet [27], Transformer encoders could achieve a great level of representation for medical image segmentation. Therefore, the current state of the art is based on the usage of Transformer-based architectures, highlighting the ones that deviated from Pyramid Vision Transformers [28,29], e.g., Polyp-PVT [30], PVT-covid19d [31], CAFE-Net [32], PVT-CASCADE [33], etc.

To identify individual objects in semantic segmentation masks, connected components may be labeled based on the existing neighborhood of pixels in an image. Although such an approach may work for some of the medical segmentation masks (e.g., polyp, lung, etc.), in scenarios where smaller objects overlay, the best approach would be to use instance segmentation models. A recent survey [34] shows a tendency to use You Only Look Once (YOLO) architectures in medical images. Even though its first version [35] was proposed for object detection, more recent iterations of YOLO (e.g., YOLOv8 [14], YOLOv9 [36], and YOLOv11 [37]) extended their capabilities to instance segmentation.

2.2. Segment Anything Model (SAM)

The usage of prompt inputs in computer vision has gained a lot of attention due to the success of generative networks and methods, e.g., Stable Diffusion [38]. In text-to-image applications, prompt-engineered inputs enable the user to be in control of the generated images. This same concept could be extended to image segmentation. For instance, when dealing with medical images, the final user (in this scenario: researchers, physicians, medical staff, etc.) would have the opportunity to select target objects on an image, filter out the unwanted ones, and obtain a more coherent segmentation mask. Some architectures were projected to have this capability, such as the Segment Everything Everywhere All at Once Model (SEEM) [39] and, more notoriously, the Segment Anything Model (SAM) [15].

SAM is a Transformer-based architecture that aims to perform segmentation while being capable of using sparse (points, boxes, and text) and dense (masks) prompt inputs. The original model was trained on the SA-1B, with over a billion available image masks that include a huge diversity of objects, landscapes, people, animals, etc. According to the authors, due to such a variety of training data, it presents a strong zero-shot performance in many segmentation tasks. The structure of SAM takes into account three main components: (1) the image encoder; (2) the prompt encoder; and (3) the mask decoder. Figure 1 presents a representation of its architecture. It is worth noting that while text and mask input prompts are supported by the original architecture, they are not depicted in the figure.

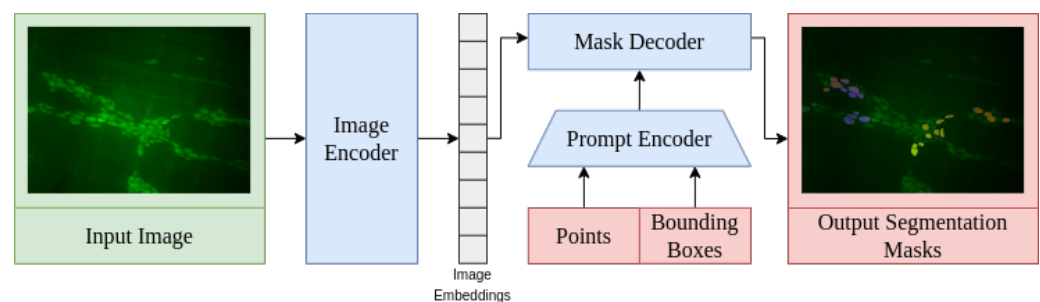


Figure 1. Representation of the SAM architecture.

According to the authors, certain input prompts (e.g., points) can introduce ambiguity during inference. To address this, SAM offers a multi-mask output, generating three possible masks per inference along with an estimated Intersection over Union (IoU) score. Each mask represents a potential interpretation of the provided visual prompt and input image.

Finally, SAM's latest iteration, SAM 2 [16], introduced significant enhancements, broadening its functionality and efficiency. A key upgrade was the integration of new memory components into the architecture, enabling robust video segmentation support. While the original SAM relied on Vision Transformers (ViTs) [40,41] as its image encoder, SAM 2 adopted a more compact Hiera encoder [42]. This transition not only streamlined the model but also significantly reduced latency for both image and video processing tasks, marking a substantial improvement in its performance and applicability.

2.3. SAM in Medical Imaging

Since its publishing, many variations of SAM have been proposed in the literature. The survey presented by [43] compiles many variations, showing different studies that tried to further expand SAM's capabilities. The works of [44–46] aimed to increment SAM in order to predict labels for the segmentation masks. As [47–49] worked to use SAM in inpainting tasks, i.e., tasks designed to automatically mask objects in an image and use them to perform some artificial modification (e.g., remove the masked objects, add a new background, etc.) using a secondary model. SAM was also adapted to object tracking tasks, as in [50–52]. Finally, the works of [53–56] expanded SAM to work with 3D input data.

Recent surveys [43] detailed a thorough examination of SAM's various applications and associated architectures. Among these, particular attention is given to its utilization in medical imaging studies, showcasing its promising prospects in this domain. The complexity of working with medical images necessitates expertise in labeling for training deep learning models. The authors note that existing deep networks tailored for medical images often focus on specific tasks, lacking the capacity for broader generalization. In response, recent research endeavors have adapted SAM specifically for medical image segmentation, resulting in notable enhancements in performance within this context.

Regarding the aforementioned application domain, SAM-Med2D [57] was introduced as an adaptation of SAM tailored for medical imaging. In their study, the authors emphasize the efficacy of incorporating learnable adapter layers within the image encoder to acquire domain-specific knowledge. Additionally, the model undergoes fine-tuning of both the prompt encoder and mask decoder to optimize parameters for medical imaging tasks. Notably, this model distinguishes itself from previous works primarily through its training dataset. The SA-Med2D-20M dataset, introduced by [58], comprises a substantial collection of 4.6 million 2D medical images and 19.7 million corresponding masks. The authors highlight the dataset's comprehensiveness, covering various anatomical regions (over 200 categories) and encompassing diverse image modalities such as computerized tomography scans, magnetic resonance imaging, X-rays, ultrasounds, and microscopy, among others. Even though different variations were proposed [59–62], at this time, this is still one of the most complete versions available in the literature.

3. ENSeg Dataset

The dataset used here represents an enhanced subset of the ENS dataset [13]. According to the authors, the ENS dataset comprises image samples extracted from the ENS of male adult Wistar rats (*Rattus norvegicus*, albus variety), specifically from the jejunum, the second segment of the small intestine. The original dataset consists of two classes: sick animals with cancer, induced with Walker-256 Tumor (WT), and control animals (C). Image acquisition involved 13 different animals, with 7 belonging to the C class and 6 to the WT class. Each animal contributed 32 image samples obtained from the myenteric plexus. All images were captured using the same setups and configuration, and saved in PNG format, with a spatial resolution of 1384×1036 pixels. The overall process of obtaining the images and performing the morphometric and quantitative analyses takes approximately 5 months, performing the following steps:

1. Treatment, euthanasia, and preparation of animal samples (1 month);
2. Dissection of the small intestine (2 weeks);
3. Immunostaining of tissues (2 weeks);
4. Image capture (1 month);
5. Quantitative and morphometric analyses (2 months).

It is essential to note that the dataset was developed under ethical principles outlined in Brazilian federal law (Law 11,794 (October 2008) and Decree 66,689 (July 2009)), as established by the Brazilian Society of Science on Laboratory Animals (SBCAL). All procedures were submitted to and approved by the Standing Committee on Ethics in Animal Experimentation at the State University of Maringá (Protocol number 062/2012). Following the experimental period, the animals were euthanized and the jejunum was collected for the immunostaining process. Finally, the animals were frozen and subsequently incinerated. More details about the dataset, including the obtaining process and protocols, may be found in the original papers [12,13].

Our dataset version considers expert-annotated labels for 6 animals, tagged as 2C, 4C, 5C, 22WT, 23WT and 28WT. The image labels were created by members from the

same laboratory in which the images originated, i.e., researchers from the Enteric Neural Plasticity Laboratory of the State University of Maringá (UEM). Labels were designed using labelme [63] and consisted of polygons annotated around each body cell. To keep the labeling paired with the laboratory's analysis quality standards, only the neuron cells with well-defined borders were considered in the final label masks. The labeling process lasted 9 months (executed from November 2023 to July 2024) and was iteratively reviewed by the lab lead researcher.

The generated files were later adapted for training and evaluating segmentation models. In the case of instance segmentation models based on the generation of point coordinates (e.g., YOLOv8), a list of x, y points per object in the image, normalized by its original height and width, is required for training. Individual segmentation masks are created for each object to extract metrics from the segmented objects. Such individual masks are created following a similar pipeline, as described in SA-Med2D-20M [58]. But, different from this one, we do not remove small masks, considering that we aim to preserve the expert's annotations' integrity.

After executing this process, the full dataset contains a total of 187 images and 9709 cell annotations. Table 1 presents the total number of image samples and ENCs per animal tag. Due to the restricted quantity of animal samples and the natural split of images, it is recommended that during training, a leave-one-out cross-validation method is used to have more reliable results. Keeping that in mind, 6 training sessions are executed per experiment. In each one, images from the same subject are isolated and used for testing the resulting model, while the remaining images are used for training.

Table 1. Total number of image samples and masks per animal tag.

Animal Tag	# of Images	# of Neurons	Average *	# of Vertices (Avg.) **
2C	32 (17.11%)	1590 (16.37%)	49.68	13
4C	31 (16.57%)	1513 (15.58%)	48.80	13
5C	31 (16.57%)	2211 (22.77%)	71.32	13
22WT	31 (16.57%)	1386 (14.27%)	44.70	10
23WT	31 (16.57%)	1520 (15.65%)	49.03	11
28WT	31 (16.57%)	1489 (15.33%)	48.03	11
Total	187 (100%)	9709 (100%)	-	-

* Number of neurons divided by the number of images. ** Sum of the number of vertices per neuron, divided by the total number of neurons.

It is worth mentioning that the usage of cross-validation is recommended, but not mandatory. Future works may present new training schema as the number of annotated subjects (rats) increases. It is only indicated that images from the same source (animal) are kept in the same data split, to avoid biased results. Even though those samples are randomly selected from the animals' original tissue, we can achieve a higher level of credibility this way.

Some image samples taken from the dataset may be seen in Figure 2. In this one, the labeled neuron cells may be highlighted in the overall image. We reinforce that the total quantity of annotated neuron cells differs from the total quantity present in the image. Such samples represent a microscopic view of the randomly selected patches obtained from the animal's small intestine. Therefore, the total number of cells per image may vary.

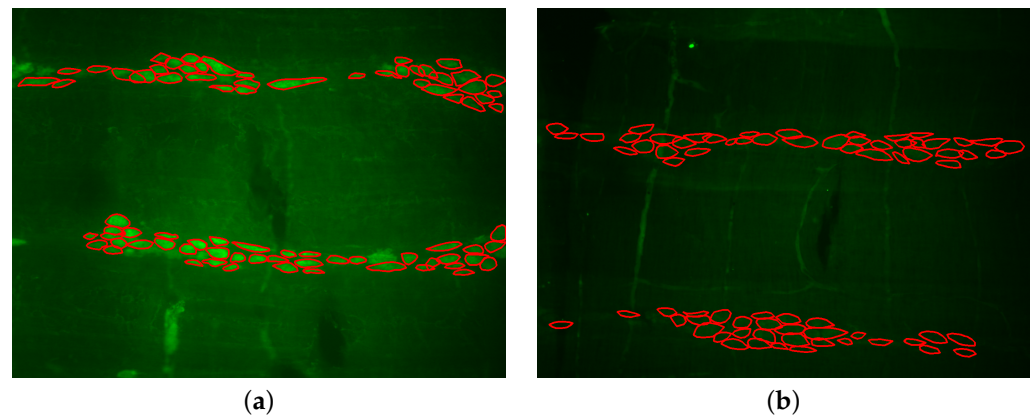


Figure 2. ENSeg image samples, with the polygon annotations delimited as red contours: (a) 5C; (b) 22WT.

Figure 3 illustrates the specific types of noise most commonly observed in the image samples from the ENSeg dataset. The noise level in a given subject (rat) can vary depending on the image acquisition process. Similar variations may also affect the brightness and contrast of the images, as noted in [13]. Examples of such noises include the following:

- Blur noises caused during the immunostaining process (Figure 3a). This is the most common noise type, found in most image samples. Blur decreases the visibility of the ENC boundaries, directly affecting their separability, and therefore the segmentation performance.
- Tear of the tissue (Figure 3b). This noise may compromise the integrity of the contained cells, affecting their shape and possibly the overall quantity.
- Presence of blood vessels (Figure 3c). Blood vessels often overlap the ENC on the images, blocking its visibility. Additionally, due to their brightness level, it may mislead the segmentation models in some scenarios.
- Cells overlapping (Figure 3d). This occurs when ENCs are captured in a deeper layer of the tissue. This characteristic gives them a low-brightness aspect and decreases their visibility. In the case of one cell overlapping another, the uppermost cell is always preferred for analysis, since its full shape is visible.

Finally, to the best of our knowledge, it is important to highlight that this is the first dataset publicly available to the research community containing expert-annotated ENCs for object segmentation. Research on the ENS remains a niche field with significant potential for further automation. The process of capturing these images, which spans extended periods, poses challenges in generating large datasets, making high-volume data collection particularly difficult. This limitation underscores the size of our dataset and highlights the need for innovative, highly adaptive methods, including zero-shot techniques, to address these constraints effectively. The ENSeg dataset will be made publicly available, alongside the developed source code, through a project on the authors' main GitHub repository.

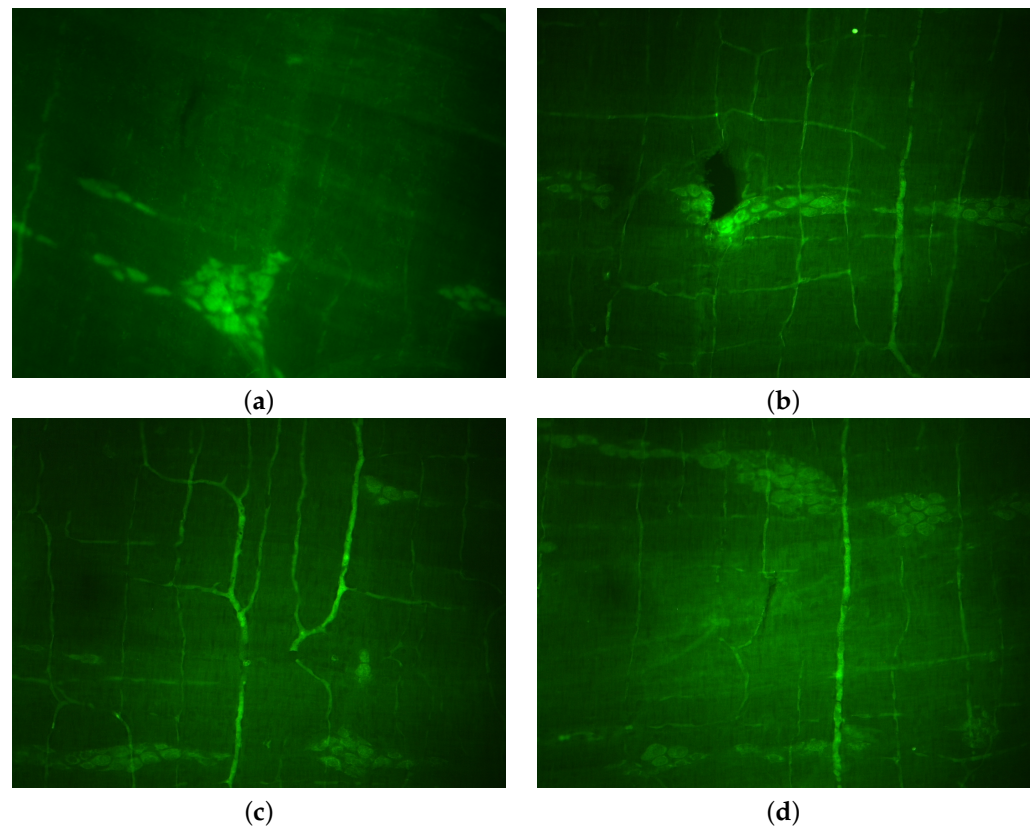


Figure 3. Common noise types found on the images of the ENSeg dataset. (a) Blurred image; (b) tear of the tissue; (c) blood vessels; (d) overlapped neuron cells.

4. Proposed Approach

In this work, we propose an approach for the ENC segmentation task designed to merge the efficiency of instance segmentation models, with the adaptability of prompt-guided segmentation. The goal of such an approach is to enable humans (in our case, researchers of the ENS) to interact with the final system and control the segmentation results. The semi-automatic approach is essential in various medical applications, as it allows the end user to actively engage with the system, providing critical guidance and intervention to shape and refine the final outcomes. Keeping that in mind, our approach is based on the following:

1. Generating candidate cells from the input image, by using a fine-tuned instance segmentation model;
2. Obtaining a single point prompt per target neuron cell, inputted by the user from the same image;
3. Selecting the candidates, based on the intersection of the point prompts and the outputs of the instance segmentation model (more details on Section 4.3);
4. If they intersect, apply a fusion protocol that can use the selected candidates' coordinates to generate bounding box prompts (enhance the prompt from a click to a box), perform the prompt-guided segmentation, and combine both outputs to obtain the final segmentation mask (more details on Section 4.3);
5. If they do not intersect, perform a prompt-guided segmentation, using the point prompt.

In addition to selecting the cells to be segmented, our approach also takes advantage of the zero-shot capabilities of SAM-like architectures, using them to further enhance the final segmentation masks through the fusion/combination of both outputs. More details about each one of the steps of the pipeline may be seen in the following subsections.

A visual representation of the proposed approach may be seen in Figure 4. In this setup, both models operate within a two-step pipeline. Starting from the inputs (image and click/point prompts), the models integrate by forwarding the outputs of one model to the next after processing them through the prompt enhancement mechanism. Although the models function within the same system, this configuration allows the use of different computing resources during inference, such as allocating a separate GPU for each model. It is also worth noting that each individual model pre-processes the input image according to its expected shape and format prior to inference (further details are provided in Section 5.1).

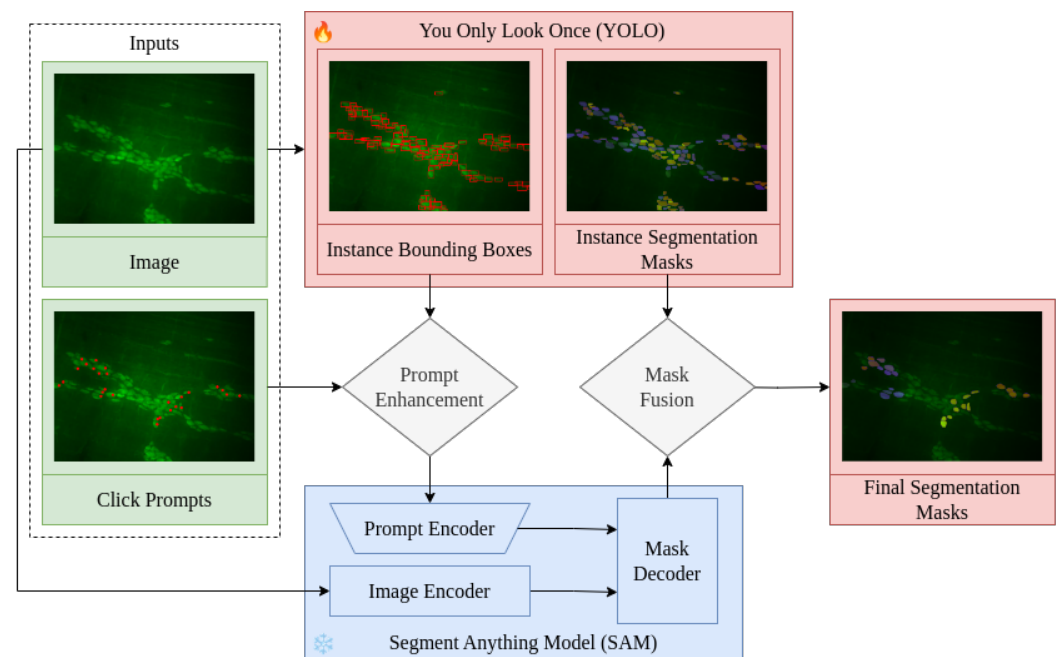


Figure 4. Visual representation of our proposed approach that is based on candidate generation; prompt fusion (or enhancement); and instance mask fusion.

4.1. Candidate Selection

Considering that only a subset of all ENC are used in the morphometric and quantitative analyses, the first step in our approach would be to generate segmentation candidates to be later selected by the user. This step is conducted by running an instance segmentation model trained on ENSeg. Since not all cells are annotated for the training of such models, our goal would be to identify the highest number of ENC, considering that their segmentation could not be optimal.

In our dataset, only ENC that meet the analysis standards are annotated. Therefore, rather than using a weakly supervised approach, we opted for supervised segmentation models to perform the candidate generation step. This approach was chosen to assess the model's ability to avoid segmenting inadequate cells, such as overlapped ENC. The generated candidates are only used in the final segmentation mask if their respective areas intersect with user prompts. This way, we enable the method to choose between (or fuse) segmentation masks in the later stages of the pipeline. Also, we naturally eliminate the unwanted segmented cells that may be rejected by users.

The instance segmentation architectures selected for this study were three of the latest versions from the YOLO family, known for their ability to perform both object detection and segmentation: YOLOv8 [14], YOLOv9 [36], and YOLOv11 [37]. This choice was motivated by their lower latency and superior generalization capabilities, as initially reported by their authors [14,36] and subsequently validated by independent studies in the literature [34]. These advantages are particularly evident in different applications,

including in medical imaging, where YOLO models have demonstrated better performance compared to alternative architectures such as Mask R-CNN [64–66]. In this work, we fine-tuned 12 different versions of the architectures, differing in their backbone size. Nominally, we shall refer to them as YOLOv8- $\{n,s,m,l,x\}$, YOLOv9- $\{c,e\}$ and YOLOv11- $\{n,s,m,l,x\}$.

To evaluate the standalone performance of these models, we measure segmentation metrics (presented in detail in Section 5.2) individually for each ENC annotation (ground truth) in the input images. Therefore, for each annotated ENC, we search for the candidate with the greatest Intersection over Union (IoU). In this paper, if a candidate is found for the target annotation, we call it a match. Once matched, the candidate is removed from the list of candidates and the process continues for all other cell annotations.

4.2. Prompt-Guided Cell Segmentation

As aforementioned, it is of great importance that the developed method allows the final users to guide the final segmentation masks. Having a method with this capability, will increase the overall precision of their final analyses and reduce machine-induced errors, while still drastically reducing the required manual labor.

Keeping that in mind, we find it quite opportune to apply the usage of SAM-based architectures to perform such prompt-guided segmentation. Such a choice is based on recent works in the literature [67]. Therefore, in this work, we approach the usage of all 3 available versions of SAM (i.e., SAM- $\{b,l,h\}$) [15], all 4 versions of SAM 2 (i.e., SAM2- $\{t,s,b+,l\}$) [16], and a fine-tuned SAM focused on medical data. The chosen variation of SAM was SAM-Med2D [57]. This choice is based on the fact that it is one of the most diverse SAM models available for medical imaging. Different from similar models (like SAMed and MedSAM), SAM-Med2D's adaptation of the prompt encoder enables us to experiment with different prompt configurations. Also, considering that it was trained on SA-Med2D-20M, it is expected that it is more susceptible to microscopy images (considering that they compose 7.22% of the dataset).

To evaluate the performance of these architectures, we propose two distinct prompt sampling modes. The first, referred to as “oracle”, represents the upper bound of the method. In this mode, points are sampled from the center-most point of the label image, and bounding boxes are perfectly fitted to the segmentation object. This prompt schema allows us to assess the maximum potential of our developed method. This conclusion is based on the assumption that the sampled points represent an ideally positioned prompt, i.e., an optimal input for the segmentation process. Furthermore, this approach enables a fair comparison of the performance across different SAM architectures.

The second sampling mode, referred to here as “random”, aims to simulate human behavior when providing prompt inputs. In real-world scenarios, humans are more likely to input suboptimal points as prompts to the architecture. In this schema, points are randomly sampled from the objects' label masks. This ensures that at least one point per ENC is selected inside the cell boundaries. By adopting this approach, we aim to obtain metrics that better reflect real-world applications. Since users do not directly input bounding boxes, as mentioned in previous sections, this sampling mode is applied exclusively to input prompts. Its purpose is to evaluate the proposed approach and compare it to the upper boundary results achieved using the “oracle” sampling mode.

It is important to note that each neuron cell is associated with a single point prompt provided by the user. This approach was chosen due to the small size of the neuron cells in the images and to enhance the usability of the final method. While the number of point prompts can influence the segmentation masks generated by SAM, in this study, the point prompt primarily serves as a selection mechanism for the generated candidates. Consequently, most of the input prompts for the SAM architectures are represented by

bounding boxes derived during the candidate generation step. Thus, the point prompt is only utilized by the architecture if a target ENC cannot be identified during the candidate generation process.

4.3. Instance Segmentation Fusion

After the process of generating the candidates and obtaining the user prompts, we can perform the inferences by using SAM-based architectures. To take greater advantage of the method and explore its capabilities, we propose four protocols that refine the final outputs by using both models' outputs. Considering I as a given input image; p as an input point prompt; c as a segmentation candidate, where $c \in \text{YOLO}(I)$; and s as SAM's output segmentation mask generated from p (i.e., $\text{SAM}(I, p)$), we propose the following protocols for defining the final segmented object:

1. Candidate-first: if p in c , consider c as the final output and s otherwise;
2. Prompt refinement: if p in c , calculate a bounding box from c and regenerate s . Otherwise, s is the final output;
3. Segmentation fusion: if p in c , combine both c and s using a fusion rule. Otherwise, s is the final output;
4. Prompt refinement with segmentation fusion: similar to Protocol 2, but combining the outputted c and s .

Verifying whether $p \in c$ serves a dual purpose: it functions both as a candidate selection strategy and as a fallback mechanism. Specifically, if YOLO fails to detect or segment a given cell (i.e., p does not intersect with any detected cells), the SAM model can still be employed to segment it. This approach is particularly beneficial for morphometric and quantitative analyses performed by ENS researchers, as it mitigates false negatives while effectively filtering out false positives. Further details on these protocols are provided later in this section, with corresponding pseudo-codes available in Algorithms 1–4.

The first protocol (candidate-first) is described in Algorithm 1. This particular approach allows us to directly evaluate the YOLO's outputs for found candidates. Therefore, it is a direct measure of how the first iteration of segmentation performs, with an increment from the fallback strategy (SAM's outputs). The final segmented objects would have the combination of both c and s (when $p \notin c$).

Algorithm 1 Protocol 1—candidate-first.

```

1: procedure PROTOCOL1( $I, p, c$ )
2:   Input: Image  $I$ , point prompt  $p$ , segmentation candidate  $c$ 
3:   Output: Final segmentation result
4:   if  $p$  is inside  $c$  then
5:     return  $c$ 
6:   else
7:      $s \leftarrow \text{SAM}(I, c)$ 
8:      $s \leftarrow s > \text{SAM.mask\_threshold}$ 
9:     return  $s$ 
10:  end if
11: end procedure

```

The second protocol (prompt refinement) is described in Algorithm 2. This one uses the bounding boxes from the candidate as input to SAM. SAM's output is then used as the final segmentation mask for the object (when $p \in c$). This approach enables the user to interact with the system using a point prompt while having a bounding box performance. According to the authors of [15,16], point prompts may lead to ambiguity during SAM's

segmentation process. Therefore, we follow the premise that bounding boxes can reach better segmentation metrics. In this protocol, all objects are segmented with SAM.

Algorithm 2 Protocol 2—prompt refinement.

```

1: procedure PROTOCOL2( $I, p, c$ )
2:   Input: Image  $I$ , point prompt  $p$ , segmentation candidate  $c$ 
3:   Output: Final segmentation result
4:   if  $p$  is inside  $c$  then
5:      $bbox \leftarrow \text{calculate\_bounding\_box}(c)$ 
6:      $s \leftarrow \text{SAM}(I, bbox)$ 
7:   else
8:      $s \leftarrow \text{SAM}(I, p)$ 
9:   end if
10:   $s \leftarrow s > \text{SAM.mask\_threshold}$ 
11:  return  $s$ 
12: end procedure

```

The third protocol (segmentation fusion) is described in Algorithm 3. Our goal with this protocol is to investigate and understand the possibility of improving the overall performance of those individual models, by complementary objects and masks. To combine two object masks, we perform the boolean operation *AND* on the binary masks. It is worth mentioning that the *AND* operation works as an intersection, inducing the foreground pixel to only exist when both segmentation masks output such pixels as the foreground. It is worth mentioning that in this protocol, no prompt refinement is performed.

Algorithm 3 Protocol 3—segmentation fusion.

```

1: procedure PROTOCOL3( $I, p, c$ )
2:   Input: Image  $I$ , point prompt  $p$ , segmentation candidate  $c$ 
3:   Output: Final segmentation result
4:    $s \leftarrow \text{SAM}(I, p)$ 
5:    $s \leftarrow s > \text{SAM.mask\_threshold}$ 
6:   if  $p$  is inside  $c$  then
7:      $combined \leftarrow c \wedge s$ 
8:     return  $combined$ 
9:   else
10:    return  $s$ 
11:   end if
12: end procedure

```

The fourth and final protocol (prompt refinements with segmentation fusion) is described in Algorithm 4. This one also aims to evaluate the combination of segmentation masks. But, before the mask fusion step, we perform a prompt refinement to convert a p from a point prompt to a bounding box prompt.

In this study, we opted to use point prompts during the inference phase. Images of cells from the ENS often exhibit significant variability in cell shapes and quantities. Typically, these images contain clusters of small cells that frequently overlap with one another. Such characteristics make the use of bounding box prompts less practical and efficient for researchers (users), presenting challenges in accurately defining the regions of interest. Point prompts, on the other hand, can be easily positioned on the ENS cell body. Our approach, particularly in protocols that incorporate prompt refinement (e.g., Protocols 2 and 4), is designed to maintain this ease of input while achieving performance comparable to that of bounding box prompts.

Algorithm 4 Protocol 4—prompt refinement with segmentation fusion.

```

1: procedure PROTOCOL4( $I, p, c$ )
2:   Input: Image  $I$ , point prompt  $p$ , segmentation candidate  $c$ 
3:   Output: Final segmentation result
4:   if  $p$  is inside  $c$  then
5:      $bbox \leftarrow \text{calculate\_bounding\_box}(c)$ 
6:      $s \leftarrow \text{SAM}(I, bbox)$ 
7:      $s \leftarrow s > \text{SAM.mask\_threshold}$ 
8:      $combined \leftarrow c \wedge s$ 
9:     return  $combined$ 
10:  else
11:     $s \leftarrow \text{SAM}(I, p)$ 
12:     $s \leftarrow s > \text{SAM.mask\_threshold}$ 
13:    return  $s$ 
14:  end if
15: end procedure

```

5. Experimental Setup

This section gathers information regarding how the experiments were executed and how the results were validated. Such information is valuable for the results' reproducibility.

5.1. Parameters, Libraries and Implementations

The Segment Anything Model (SAM) (<https://github.com/facebookresearch/segment-anything> (accessed on 9 December 2024)), SAM 2 (<https://github.com/facebookresearch/sam2> (accessed on 9 December 2024)) and SAM-Med2D (<https://github.com/OpenGVLab/SAM-Med2D> (accessed on 9 December 2024)) sources and weights were downloaded from the authors' original repository at GitHub. For inference, their respective predictor classes were employed. Their pre-processing is based on resizing (the SAM/SAM 2 encoder input size is equivalent to 1024×1024 and SAM-Med2D to 256×256), padding, and normalizing the images using pre-established values of mean and standard deviation for each image channel.

The YOLOv8, YOLOv9, and YOLOv11 models were trained using the Ultralytics official library (<https://www.ultralytics.com/> (accessed on 9 December 2024)). Models were trained for 100 epochs with a batch size equivalent to 16. Images were padded, resized to the expected shape (640×640), and normalized to the range of $[0, 1]$. Random data augmentation was applied online during training (e.g., random flipping, scaling, translating, color jittering, etc.). The model was optimized by Adam [68], with an initial learning rate of 0.01, a 3-epoch warm-up, and a weight decay of 5×10^{-4} .

All the developed sources have been made available on GitHub (Microsoft, San Francisco, CA, USA). The resulting models and other artifacts generated from training may also be found in the same repository. All training sessions and experiments were executed on a Python 3 Google Compute Engine backend with a T4 GPU (15 GB dedicated memory), 12 GB RAM, and Intel(R) Xeon(R) CPU (2.00 GHz) (Intel, Santa Clara, CA, USA).

5.2. Evaluation Metrics

To evaluate our results, we recall the usage of a total of two well-known metrics in the literature for both segmentation and object detection. The first one, Intersection Over Union (IoU) [69], may be defined by Equation (1).

$$\text{IoU}(pred, gt) = \frac{|pred \cap gt|}{|pred \cup gt|} \quad (1)$$

In this one, *pred* represents a predicted segmentation mask, as *gt* represents the input's original ground truth (label). By using this metric, we can understand the overlap ratio of the prediction. The maximum value would be equivalent to 1 and would be found only if *pred* perfectly overlays *gt*. It is worth mentioning that the same metric may be extended for bounding boxes.

Equation (2) presents the Dice similarity coefficient [70], a second metric used in this work.

$$\text{DICE}(\text{pred}, \text{gt}) = \frac{2 \times |\text{pred} \cap \text{gt}|}{|\text{pred}| + |\text{gt}|} \quad (2)$$

Similar to an F1-Score (or F-Measure) in classification, these measures verify the harmonic mean between *pred* and *gt*. By calculating the Dice metric between such values, we are able to identify the overlap proportion, normalized by the dimension of both masks/objects. Similar to IoU, a perfect overlap will have a Dice of 1, as the lack of any overlap will have Dice of 0.

It is worth mentioning that, different from semantic segmentation tasks, which perform those metrics on a global level (i.e., considering the whole predicted segmentation mask), on the here-approached task, each segmented object is evaluated separately. By the end of the evaluation process, the mean IoU (mIoU) and Dice (mDice) values of every cell in a test set are presented.

5.3. Validation Schema

The validation schema used here is the same as that proposed in our previous work [13]. Considering the low quantity of available images, we apply *k*-fold cross-validation. In this scenario, each fold is represented by one animal group, composing a leave-one-out cross-validation schema. From that, we assume a total of 6 folds, where during training a *i*-th fold is used for testing, as the remaining ones are used for training.

It is worth mentioning that by following this approach, we keep all images belonging to the same animal source gathered in the same fold. Therefore, we guarantee that our results remain impartial, devoid of any bias towards the inherent characteristics of individuals. Additionally, from such an approach, we are able to visualize and measure each animal's performance individually.

6. Experimental Results

This section presents the results of the executed experiments. In the first moment, we present the baseline results obtained from the standalone models. Later on, the results obtained from the four established combination protocols are introduced.

6.1. Baseline Results

In this section, we evaluate the stand-alone segmentation models to establish baseline results and better assess the results from the proposed approach. Figure 5 shows the metrics obtained by YOLOv8, YOLOv9, and YOLOv11, calculated by averaging the DICE values of each fold, following our validation schema (leave-one-patient-out cross-validation). In this one, it is possible to observe that all values were inferior to 0.7. Since we do not have all cells labeled in our dataset, we can assume that suboptimal models were found during training. Such an assumption is made based on the fact that the architectures could identify ENC outside the annotated set. Therefore, during training, such objects would be identified as wrong defections, causing noise in the calculation of training error. Since YOLOv8-m performed as well as the other architectures while having a smaller standard deviation and acceptable latency (when compared to larger versions), this one was selected to run the candidate selection on experiments presented in Section 6.2.

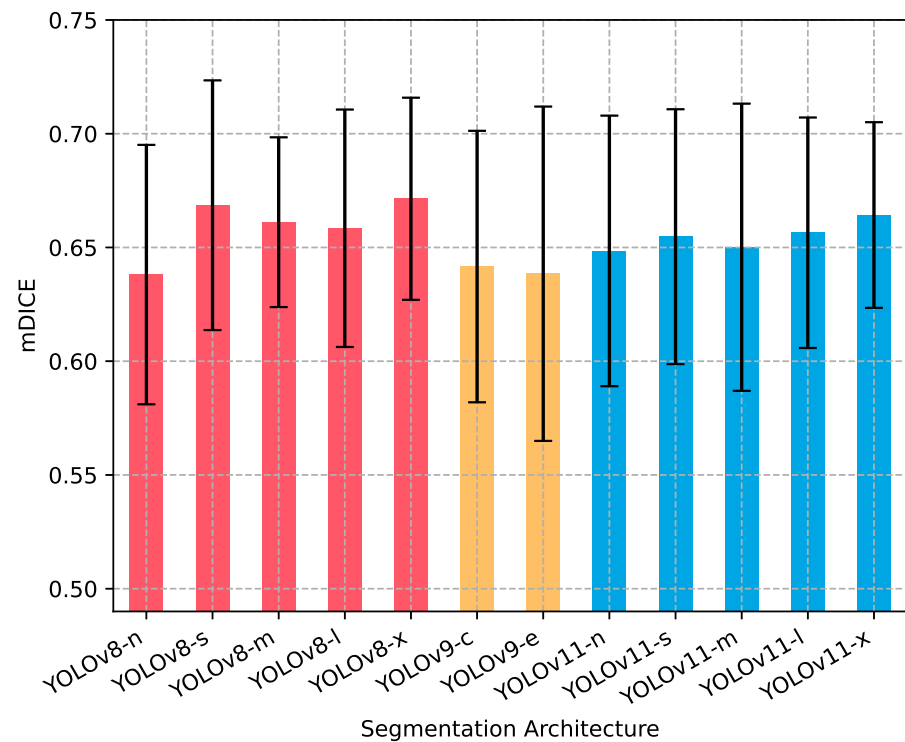


Figure 5. YOLOv8 (red), YOLOv9 (yellow), and YOLOv11 (blue) baseline metrics on annotated cells.

Figure 6 presents the obtained mDICE for SAM, SAM2, and SAM-Med2D, averaged from the oracle evaluation of each validation set. When analyzing it, it is possible to notice that SAM-Med2D underperformed compared to the regular SAM and SAM2 models. Although it was fine-tuned on medical datasets, we presume that its reduced input size and lack of similar microscopy image data influenced this result. This affirmation is based on the fact that ENSeg images differ from regular microscopy images (e.g., the ones with H&E coloration) and therefore should require additional tuning to enhance such representation.

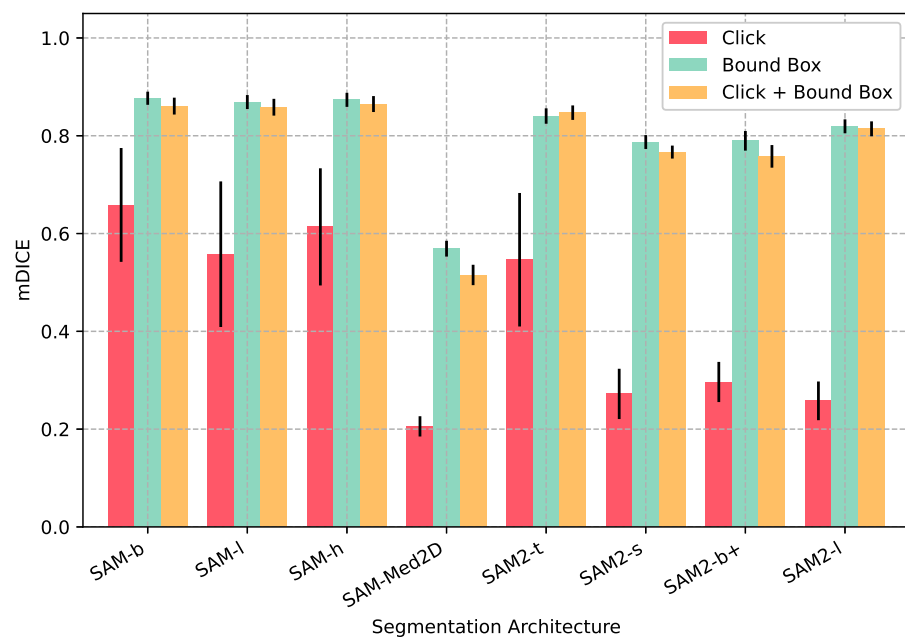


Figure 6. SAM baseline metrics (oracle evaluation method).

When checking the prompt types, it is possible to observe that by using bounding boxes as prompt inputs, we can reach higher mDICE values. Such behavior matches what was reported by the original authors. Also, the combination of bounding boxes and click prompts (by simultaneous input) did not improve the metrics. Additionally, when comparing the versions of SAM, we may notice that although they have different sizes, their performance is similar. The obtained metrics also show that SAM slightly outperformed SAM2. Therefore, throughout the rest of this paper, we choose to use SAM-b, since it presents a better mDICE for both points and bound box prompts and has a smaller latency when compared to the other SAM variations (i.e., SAM-l and SAM-h).

Table 2 presents our final baseline results. Click (point) prompts are obtained with the random evaluation schema since it is closer to human behavior. As reported in previous sections, these are preferable on the ENSeg dataset. The bounding boxes on the other hand have their values presented from the oracle evaluation since they represent an upper limit to our results. In this table, it is possible to observe that the mean difference from the bound box to the click prompt type is equivalent to 0.2448 (mIoU) and 0.2551 (mDICE). This difference in performance justifies our protocols with prompt refinement, in which we try to leverage the performance of point prompts on the here-approached task. It is also possible to notice that the WT subjects have an average mDICE difference of 0.3563, while for the C subjects, it is equivalent to 0.1540 (79.26% smaller). The WT subjects often present a higher level of unwanted noises (as described in Section 3) and lower sharpness when compared to C, increasing the challenge of segmenting ENC's on them.

Table 2. Metric values obtained from SAM-b, by using point (random sampling) and bounding box (oracle sample) prompts.

Prompt Type	Metric	Animal Tag						Mean
		2C	4C	5C	22WT	23WT	28WT	
Point (Random)	mIoU	0.6409	0.6146	0.6816	0.4620	0.4294	0.4178	0.5411
	mDICE	0.7321	0.7016	0.7704	0.5376	0.5009	0.4861	0.6215
Bound Box (Oracle)	mIoU	0.8046	0.8012	0.8070	0.7716	0.7652	0.7658	0.7859
	mDICE	0.8885	0.8872	0.8906	0.8677	0.8630	0.8629	0.8766
Difference	mIoU	0.1637	0.1865	0.1254	0.3095	0.3357	0.3479	0.2448
	mDICE	0.1564	0.1855	0.1201	0.3300	0.3620	0.3767	0.2551

Table 3 shows the total number of objects detected by the trained YOLOv8-m model. It can be observed that, in most evaluation sets (animal tags), YOLOv8 detected more objects than were originally labeled, with an average of 11.26% additional ENC's. In our task, this over-detection is interpreted as generating a variety of segmentation candidates, which will later be filtered during the overall process and excluded from the final mask. This supports our decision to report metrics only on the annotated objects. Furthermore, since only ENC's suitable for analysis are labeled in our dataset, it is crucial that these are successfully identified. As such, at this stage, we prioritize the match rate over other metrics.

The greatest difference in the number of segmented instances may be seen in the 4C animal set, where 26.56% additional ENC's were labeled and 95.77% of their tagged cells had a respective match. Similar to the results presented in Table 2, the C subjects presented a higher match rate (average of 93.95%) and mDICE (average 0.6904) when compared to WT (87.25% and 0.6317, respectively), further supporting our statement on the impact of the noises found in those images. From these results, we can also conclude that the resulting segmentation objects are not optimal. Such a fact becomes clearer when checking that, although 90.60% of the ENC's were matched, we only achieved a final mDICE of 0.6611.

Such a value is 0.2155 smaller than the metric obtained by SAM-b and box prompts (oracle evaluation), showing that we can still improve the masks for the segmented cells.

Table 3. Total quantity of detected cells, matches, and metrics found by the YOLOv8-m model, for each animal tag.

Animal Tag	# of Cells	# Predicted Cells	# Matches	mIoU	mDICE
2C	1590	1612 (1.38%)	1440 (90.56%)	0.5389	0.6582
4C	1513	1915 (+26.56%)	1449 (95.77%)	0.5835	0.7142
5C	2211	2582 (+16.77%)	2112 (95.52%)	0.5666	0.6988
22WT	1386	1378 (−0.57%)	1195 (86.21%)	0.5250	0.6354
23WT	1520	1666 (+9.6%)	1328 (87.36%)	0.5183	0.6371
28WT	1489	1695 (+13.83%)	1313 (88.18%)	0.5035	0.6226
Mean	1618.16	1808 (+11.26%)	1472.83 (90.60%)	0.5393	0.6611

From the presented results, we are able to verify that the final mDICE and mIoU are equivalent to 0.6611 ± 0.0373 and 0.5393 ± 0.0304 for YOLOv8-m. As for SAM-b (random click prompt), the same metrics are equivalent to 0.6215 ± 0.1270 and 0.5411 ± 0.1174 . Therefore, the point-prompt segmentation model presented an mDICE 5.99% smaller and mIoU 0.33% higher when compared to the fully automatic instance segmentation method. Although there is a difference in those metrics, by applying a Wilcoxon statistical test on the mDICE values for each fold, considering both models, we obtain a p -value of 0.3125. Since this value is not smaller than 0.05, we can consider that the difference is not statistically significant.

6.2. Combination Protocols

In this section, we present the results from combining SAM-b and the trained YOLOv8-m using the four protocols presented in Section 4.3. Table 4 presents the results found for each protocol. In this one, we consider the baseline and upper limit metrics as the random point prompts and the bound box (oracle) results using SAM-b, respectively (presented in Table 2).

From these results, it is possible to notice that the greatest improvements were obtained by fusion Protocol #2 (prompt refinement), which uses point prompts to select a candidate and uses the candidate's bounding box as input to the SAM-b's prompt encoder. This protocol, achieved the highest average mDICE when compared to the other protocols, being equivalent to 0.7877. The greatest increase in metrics comes from the WT subjects. Such increments were equivalent to 0.2035 ($\uparrow$ 37.85%), 0.2446 ($\uparrow$ 48.83%) and 0.3016 ($\uparrow$ 62.04%), respectively, for the 22WT, 23WT and 28TW animal tags. The results show that we can improve our segmentation metrics by using prompt fusion enhancement, i.e., converting point prompts to bound box prompts.

Table 4. mDICE values for each animal tag, considering the 4 fusion protocols.

Fusion Protocol	Animal Tag						Mean
	2C	5C	6C	22WT	23WT	28WT	
Baseline	0.7321	0.7016	0.7704	0.5376	0.5009	0.4861	0.6215
1	0.7296	0.7456	0.7382	0.6945	0.6813	0.6733	0.7104
2	0.8156	0.8336	0.8433	0.7474	0.7411	0.7455	0.7877
3	0.7928	0.8193	0.8259	0.7301	0.7208	0.7208	0.7683
4	0.8152	0.8368	0.8458	0.7418	0.7323	0.7406	0.7854
Upper Limit	0.8885	0.8872	0.8906	0.8677	0.8630	0.8629	0.8766

From the results of Protocol #1 (candidate-first), we conclude that it underperformed when compared to other protocols. This protocol selects the instance segmentation cells (YOLOv8-m outputs) to compose the final mask, which are already known to have a lower IoU than the target cells. Although it reached a smaller mDICE when compared to the other protocols, it still outperformed the standalone SAM-b and YOLOv8 models by, respectively, 0.0888 ($\uparrow$ 14.30%) and 0.0493 ($\uparrow$ 07.45%). Such an increase may be also justified by the fact that such an approach includes the ENC's that were not matched during the candidate generation phase.

Similar behavior was also found when applying Protocol #3 (segmentation fusion). As we fused the output masks of the point-prompted SAM-b and YOLOv8-m, it was possible to notice that the intersection rule increased the results obtained by the standalone models by 0.1467 ($\uparrow$ 23.62%) and 0.1071 ($\uparrow$ 16.21%). This increment in the evaluation metrics indicates that we benefit from a late mask fusion, leading to the thought that the generated segmentation masks are complementary.

Protocol #4 (prompt refinement with segmentation fusion), on the other hand, presented similar results to Protocol #2. They both share the same feature of converting the prompt enhancement (click to bound box). But, Protocol #4 also performs mask fusions. This protocol slightly enhanced the 5C and 6C subject metrics, representing an increment of 0.0032 ($\uparrow$ 0.38%) and 0.0025 ($\uparrow$ 0.29%). Although the final mDICE outperformed the other fusion protocols, it failed to surpass the segmentation metrics calculated for Protocol #2. This last one obtained a statically similar mDICE (supported by the Wilcoxon statistical test, with a p-value above 0.3) and requires less computational power for inferring the results.

By the end of this work, it is possible to conclude that the developed method was able to increase both YOLOv8-m and SAM-b (point prompt) results individually. The final mDICE obtained by the protocol that applied prompt enhancement (#2) was equivalent to 0.7877. Such value is 26.74% greater than the baseline (SAM-b with point prompts) and 19.14% greater than the standalone candidate selection model (YOLOv8-m). But, we reinforce that our method still has room for improvement. This affirmation is based on the fact that our best mDICE is 10.14% smaller than the upper limit (i.e., SAM-b with perfectly fit bound boxes). Future works will aim to decrease such differences by further exploring new models for candidate selection, testing new fusion approaches, and working on a personalized ensemble learning pipeline.

6.3. Final Remarks and Limitations

To evaluate the final performance of the proposed approach, Figures 7–11 illustrate each step of the process using the same input image. These results were obtained by applying the pipeline that achieved the best overall metrics: YOLOv8-m for candidate selection, SAM-b for segmenting the ENC, and the prompt refinement fusion protocol. Figure 7a shows the annotated input image, containing a total of 70 labeled ENC's. Figure 7b represents the simulation of user-prompted inputs, where one point prompt is provided per cell. It is worth noting that the points are not centered due to the random validation schema employed, in which points are randomly sampled from within the labeled cell polygons.

Figure 8a depicts the segmentation results from the candidate generation step, where 68 cells were pre-segmented. While the majority of cells in the image were successfully identified, the model encountered difficulties in accurately defining the boundaries of individual cells. The resulting polygons, averaging 17 points per polygon compared to 12 in the annotations, often displayed a squared shape that lacked the natural roundness typically observed in the cells. Furthermore, some segmented objects exhibited slight margins of error relative to their original counterparts. These findings are consistent with the low standalone segmentation metrics discussed in previous sections.

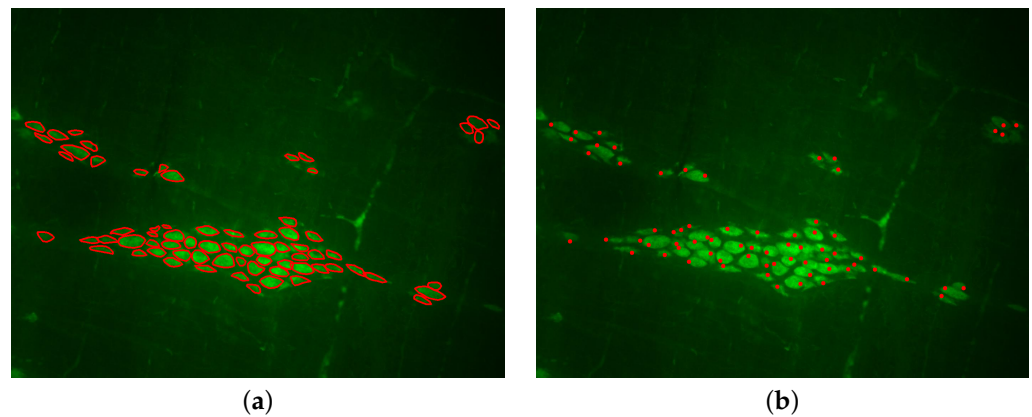


Figure 7. Example of input images and prompt points. (a) Ground truth. (b) Point prompts (random schema).

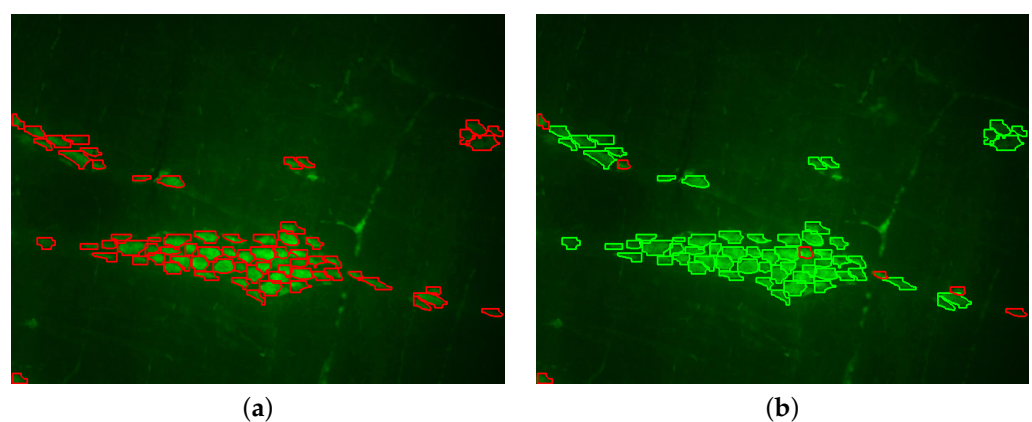


Figure 8. Example of output results from candidate generation step. (a) Generated candidates (YOLOv8). (b) Candidate matches (outlined in green) and non-matches (outlined in red).

A comparison of outputs across different YOLO architectures, as shown in Figure 9, reveals that the previously mentioned behavior persists regardless of the version or size used. This observation suggests that, for the application presented here, these architectures struggled to accurately predict the complex shapes of ENC. One possible explanation for this limitation is the constrained availability of annotated images, which may have hindered effective fine-tuning of the model. Notably, among the architectures evaluated, YOLOv8-m achieved the highest match rate for the visible cluster of cells, with only a single ENC candidate missing.

The input point prompts intersected with 62 of the required cells, achieving a match rate of 88.57%. The matched cells are highlighted with green boundaries in Figure 8b. Following prompt enhancement, the bounding boxes of the matched cells were recalculated, and their corresponding point prompts were updated. Figure 10a illustrates the input prompts provided to SAM-b, which include 62 bounding boxes and eight points. The final segmentation results are shown in Figure 10b.

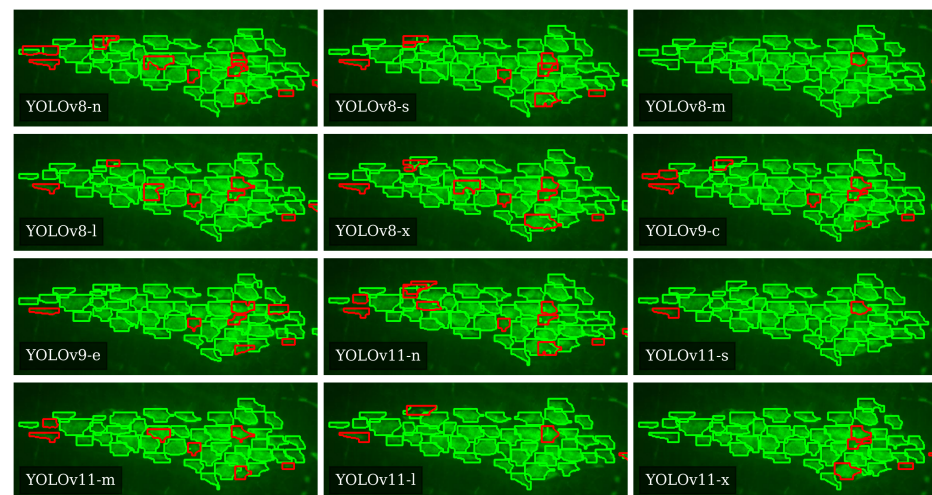


Figure 9. Comparison of YOLO architectures, used for segmentation candidates generation. Cells matched are represented by the green outlines, the ones in red represent non-matches.

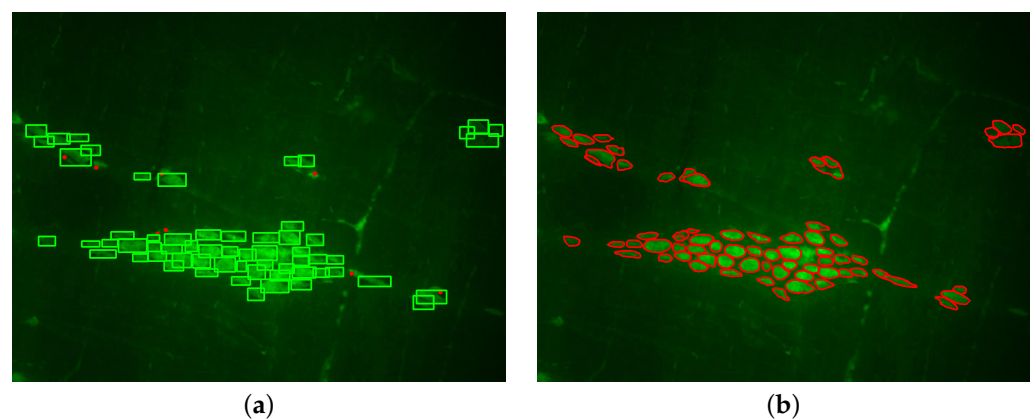


Figure 10. Example of input prompts to SAM and the final segmentation result. (a) Refined input prompts. (b) Final segmentation (SAM).

The final output segmentation mask reveals that the segmented ENC's exhibit shapes closely aligned with the expected contours. Additionally, the boundaries of the segmented objects are more accurately delineated, showing better alignment with the actual objects compared to the results produced by YOLOv8. However, in some cases, the SAM model merged clusters of cells into a single segmentation object, particularly when large bounding boxes were used. This issue could be mitigated by refining the candidate selection method to better isolate individual cells.

Finally, Figure 11 compares the outputs obtained from the tested SAM architectures. From this comparison, it is clear that SAM-Med2D performed poorly in our task, as no individual cells were successfully segmented. Instead, this architecture produced suboptimal masks containing sub-clusters of cells. Additionally, for the input sample presented here, only SAM-b, SAM-l, and SAM2-t yielded satisfactory results for ENC segmentation. The other models, namely SAM-h, SAM2-s, SAM2-b+, and SAM2-l, also segmented additional structures, such as the background and the entire plexus, which led to misleading final results.

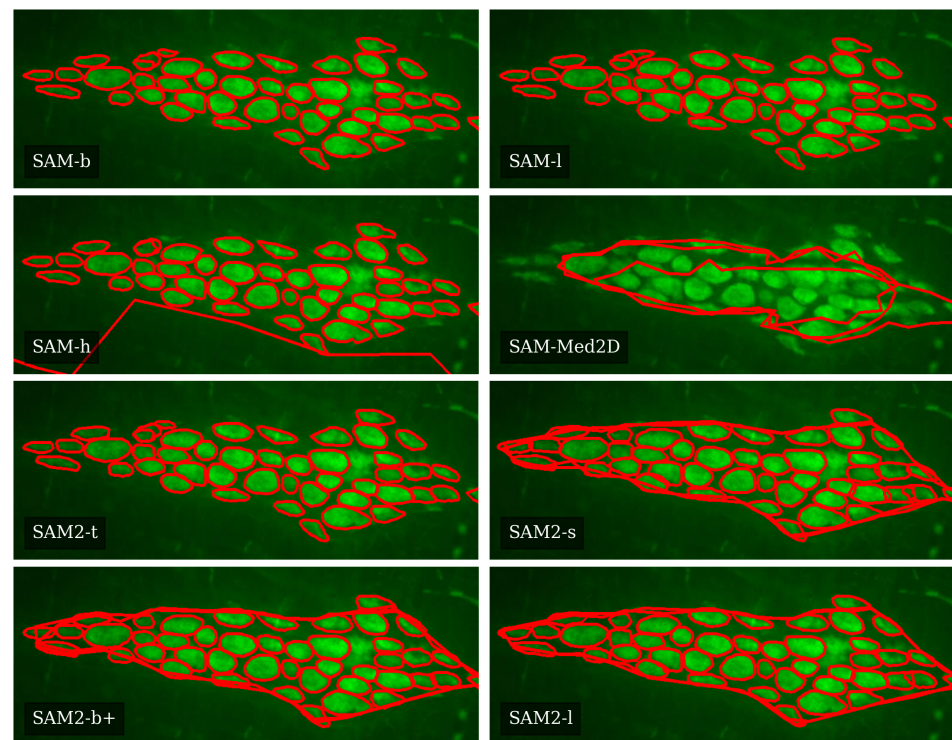


Figure 11. Comparison of SAM architectures, used for generating the final segmentation masks.

7. Conclusions

In this work, we introduced a new dataset called ENSeg. This dataset presented a novel task to the literature, containing different microscopy images obtained from the Enteric Nervous System (ENS). Studying cells from the ENS presents significant challenges due to the limited availability of specialized techniques in existing literature. Researchers must undertake labor-intensive, time-consuming analyses to assess the effectiveness of new methods designed to explore the effects of chronic degenerative diseases on the human body.

To solve such a problem, we developed a new approach that combines the efficiency of automatic instance segmentation models with the versatility of SAM-like architectures. Our goal was to allow human interaction with the system while providing a pipeline that fits the problem we are solving. Therefore, we employed YOLO architectures for generating segmentation candidates that were later combined with SAM architectures, in a schema that may include the fusion of their output instance masks or the input prompts.

Our final mDICE score was 0.7877, achieved using YOLOv8-m for candidate selection, SAM-b for segmentation, and a fusion protocol designed to enhance input point prompts by converting them into bounding boxes. This represents a mean increase of 26.74% and 19.14% over our baselines, namely standalone SAM-b point prompt and YOLOv8-m, respectively. However, this score is 10.14% below the upper limit of our solution, defined by SAM-b results obtained from bounding boxes perfectly fitted to the cells (oracle evaluation). These findings indicate that there is still room for improvement in our approach.

Future work will aim to fully fine-tune SAM models for this task; design task-specific adapter modules for partial SAM fine-tuning; evaluate the usage of new late fusion techniques; train new candidate generation methods/models; develop an ensemble learning training pipeline for a hybrid approach; validate the developed technique on other ENS cells (e.g., Enteric Glial Cells); and validate our method on the resources (e.g., images and hardware) of new laboratories and by different ENS researchers.

Author Contributions: Conceptualization, G.Z.F.; methodology, G.Z.F., L.N. and Y.M.e.G.d.C.; software, G.Z.F.; resources, J.N.Z.; data labeling, I.G.G. and J.N.Z.; writing—original draft preparation, G.Z.F.; writing—review and editing, L.N., J.N.Z. and Y.M.e.G.d.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Programa de Doutorado Sanduíche no Exterior (PDSE) grant number 88881.846308/2023-01.

Institutional Review Board Statement: All procedures regarding the images' obtaining were submitted to and approved by the Standing Committee on Ethics in Animal Experimentation at the State University of Maringá (Protocol number 062/2012, protocol date 3 July 2012).

Informed Consent Statement: Not applicable.

Data Availability Statement: All source code and detailed information regarding the ENSeg dataset, including both images and object annotations, are readily available on our GitHub page: <https://github.com/gustavozf/seg-lib> (accessed on 9 December 2024). It is worth mentioning that the code sources are licensed under Apache 2.0, and the ENSeg dataset is licensed under Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Acknowledgments: The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES)-Finance Code 001, and the National Council for Scientific and Technological Development (CNPq) for partially financing this research. AI or AI-assisted tools were used in parts of the manuscript exclusively for language translation, language editing, and grammar checking.

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

References

1. Furness, J.B. (Ed.) *The Enteric Nervous System*; Blackwell Publishing: Hoboken, NJ, USA, 2007. [\[CrossRef\]](#)
2. Sharkey, K.A. Emerging roles for enteric glia in gastrointestinal disorders. *J. Clin. Investig.* **2015**, *125*, 918–925. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Sehaber-Sierakowski, C.C.; Vieira-Frez, F.C.; Hermes-Uliana, C.; Martins, H.A.; Bossolani, G.D.P.; Lima, M.M.; Blegniski, F.P.; Guarnier, F.A.; Baracat, M.M.; Perles, J.V.C.M.; et al. Protective effects of quercetin-loaded microcapsules on the enteric nervous system of diabetic rats. *Auton. Neurosci.* **2021**, *230*, 102759. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Martins-Perles, J.V.C.; Bossolani, G.D.P.; Zignani, I.; de Souza, S.R.G.; Frez, F.C.V.; de Souza Melo, C.G.; Barili, E.; de Souza Neto, F.P.; Guarnier, F.A.; Armani, A.L.C.; et al. Quercetin increases bioavailability of nitric oxide in the jejunum of euglycemic and diabetic rats and induces neuronal plasticity in the myenteric plexus. *Auton. Neurosci.* **2020**, *227*, 102675. [\[CrossRef\]](#)
5. Vieira-Frez, F.C.; Sehaber-Sierakowski, C.C.; Perles, J.V.C.M.; Bossolani, G.D.P.; Verri, W.A.; Nascimento, R.C.D.; Guarnier, F.A.; Bordini, H.P.; Blegniski, F.P.; Martins, H.A.; et al. Anti- and pro-oxidant effects of quercetin stabilized by microencapsulation on interstitial cells of Cajal, nitrergic neurons and M2-like macrophages in the jejunum of diabetic rats. *Neurotoxicology* **2020**, *77*, 193–204. [\[CrossRef\]](#)
6. Zandoni, J.N.; Piovezana Bossolani, G.D. Does the rheumatoid arthritis affect the enteric nervous system? *Arq. Gastroenterol.* **2019**, *56*, 113–117. [\[CrossRef\]](#)
7. Piovezana Bossolani, G.D.; Silva, B.T.; Colombo Martins Perles, J.V.; Lima, M.M.; Vieira Frez, F.C.; Garcia de Souza, S.R.; Sehaber-Sierakowski, C.C.; Bersani-Amado, C.A.; Zandoni, J.N. Rheumatoid arthritis induces enteric neurodegeneration and jejunal inflammation, and quercetin promotes neuroprotective and anti-inflammatory actions. *Life Sci.* **2019**, *238*, 116956. [\[CrossRef\]](#)
8. Oliveira, A.P.; Perles, J.V.C.M.; de Souza, S.R.G.; Sestak, S.S.; da Motta Lima, F.G.; Almeida, G.H.D.R.; Cicero, L.R.; Clebis, N.K.; Guarnier, F.A.; Blegniski, F.P.; et al. L-glutathione 1 reduces the oxidative stress in the jejunum of rats with Walker-256-bearing tumor. *Neurogastroenterol. Motil.* **2023**, *35*, e14688. [\[CrossRef\]](#)
9. Vicentini, G.E.; Martins, H.A.; Fracaro, L.; de Souza, S.R.G.; da Silva Zandoni, K.P.; Silva, T.N.X.; Blegniski, F.P.; Guarnier, F.A.; Zandoni, J.N. Does l-glutamine-supplemented diet attenuate NO-mediated damage on myenteric plexus of Walker 256 tumor-bearing rats? *Food Res. Int.* **2017**, *101*, 24–34. [\[CrossRef\]](#)
10. Lima, F.G.D.M.; Silva, M.P.A.D.; Sestak, S.S.; Guarnier, F.A.; de Oliveira, A.P.; Kuller, J.V.; Gulbransen, B.D.; Perles, J.V.C.M.; Zandoni, J.N. Cancer-induced morphological changes in enteric glial cells in the jejunum of Walker-256 tumor-bearing rats. *Acta Histochem.* **2024**, *126*, 152146. [\[CrossRef\]](#)

11. Martins, H.A.; Sehaber, C.C.; Hermes-Uliana, C.; Mariani, F.A.; Guarnier, F.A.; Vicentini, G.E.; Bossolani, G.D.; Jussani, L.A.; Lima, M.M.; Bazotte, R.B.; et al. Supplementation with L-glutamine prevents tumor growth and cancer-induced cachexia as well as restores cell proliferation of intestinal mucosa of Walker-256 tumor-bearing rats. *Amino Acids* **2016**, *48*, 2773–2784. [CrossRef]
12. Felipe, G.Z.; Zanoni, J.N.; Sehaber-Sierakowski, C.C.; Bossolani, G.D.P.; Souza, S.R.G.; Flores, F.C.; Oliveira, L.E.S.; Pereira, R.M.; Costa, Y.M.G. Automatic chronic degenerative diseases identification using enteric nervous system images. *Neural Comput. Appl.* **2021**, *33*, 15373–15395. [CrossRef] [PubMed]
13. Felipe, G.Z.; Teixeira, L.O.; Pereira, R.M.; Zanoni, J.N.; Souza, S.R.G.; Nanni, L.; Cavalcanti, G.D.C.; Costa, Y.M.G. Cancer Identification in Enteric Nervous System Preclinical Images Using Handcrafted and Automatic Learned Features. *Neural Process. Lett.* **2022**, *55*, 5811–5832. [CrossRef]
14. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 9 December 2024).
15. 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. *arXiv* **2023**, arXiv:2304.02643.
16. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. *arXiv* **2024**, arXiv:2408.00714.
17. 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), Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9351, pp. 234–241.
18. Falk, T.; Mai, D.; Bensch, R.; Çiçek, Ö.; Abdulkadir, A.; Marrakchi, Y.; Böhm, A.; Deubner, J.; Jäkel, Z.; Seiwald, K.; et al. U-Net—Deep Learning for Cell Counting, Detection, and Morphometry. *Nat. Methods* **2019**, *16*, 67–70. [CrossRef]
19. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
20. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Proceedings of the 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, 20 September 2018*; Proceedings 4; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
21. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
22. Jha, D.; Smedsrud, P.H.; Riegler, M.A.; Johansen, D.; De Lange, T.; Halvorsen, P.; Johansen, H.D. Resunet++: An advanced architecture for medical image segmentation. In Proceedings of the 2019 IEEE International Symposium on Multimedia (ISM), San Diego, CA, USA, 9–11 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 225–2255.
23. Wang, P.; Chung, A.C.S. DoubleU-Net: Colorectal Cancer Diagnosis and Gland Instance Segmentation with Text-Guided Feature Control. In Proceedings of the Computer Vision—ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; Bartoli, A., Fusiello, A., Eds.; Springer: Cham, Switzerland, 2020; pp. 338–354.
24. Akbas, C.E.; Kozubek, M. Condensed U-Net (Cu-Net): An Improved U-Net Architecture for Cell Segmentation Powered by 4×4 Max-Pooling Layers. In Proceedings of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 3–7 April 2020; pp. 446–450.
25. Saha, S.; Dutta, S.; Goswami, B.; Nandi, D. ADU-Net: An Attention Dense U-Net based deep supervised DNN for automated lesion segmentation of COVID-19 from chest CT images. *Biomed. Signal Process. Control.* **2023**, *85*, 104974. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems, Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
27. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306.
28. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578.
29. 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. *Comput. Vis. Media* **2022**, *8*, 415–424. [CrossRef]
30. Dong, B.; Wang, W.; Fan, D.P.; Li, J.; Fu, H.; Shao, L. Polyp-pvt: Polyp segmentation with pyramid vision transformers. *arXiv* **2021**, arXiv:2108.06932. [CrossRef]
31. Zheng, L.; Fang, J.; Tang, X.; Li, H.; Fan, J.; Wang, T.; Zhou, R.; Yan, Z. PVT-COV19D: COVID-19 Detection Through Medical Image Classification Based on Pyramid Vision Transformer. In Proceedings of the Computer Vision—ECCV 2022 Workshops: Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part VII; Springer: Berlin/Heidelberg, Germany, 2023; pp. 526–536. [CrossRef]

32. Liu, G.; Yao, S.; Liu, D.; Chang, B.; Chen, Z.; Wang, J.; Wei, J. CAFE-Net: Cross-Attention and Feature Exploration Network for polyp segmentation. *Expert Syst. Appl.* **2024**, *238*, 121754. [\[CrossRef\]](#)
33. Rahman, M.M.; Marculescu, R. Medical Image Segmentation via Cascaded Attention Decoding. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 6211–6220. [\[CrossRef\]](#)
34. Qureshi, R.; Ragab, M.G.; Abdulkader, S.J.; Muneer, A.; Alqushaib, A.; Sumiea, E.H.; Alhussian, H.; Al-Selwi, S.M. A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *TechRxiv* **2023**, *12*, 57815–57836. [\[CrossRef\]](#)
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
36. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
37. Jocher, G.; Qiu, J. Ultralytics YOLO11.2024. Available online: https://docs.ultralytics.com/pt/models/yolo11/#__tabbed_3_1 (accessed on 9 December 2024).
38. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv* **2021**, arXiv:2112.10752. [\[CrossRef\]](#)
39. Zou, X.; Yang, J.; Zhang, H.; Li, F.; Li, L.; Wang, J.; Wang, L.; Gao, J.; Lee, Y.J. Segment Everything Everywhere All at Once. *arXiv* **2023**, arXiv:2304.06718. [\[CrossRef\]](#)
40. 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 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
41. Li, Y.; Mao, H.; Girshick, R.; He, K. Exploring Plain Vision Transformer Backbones for Object Detection. In Proceedings of the Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part IX; Springer: Berlin/Heidelberg, Germany, 2022; pp. 280–296. [_17 \[CrossRef\]](#)
42. Ryali, C.; Hu, Y.T.; Bolya, D.; Wei, C.; Fan, H.; Huang, P.Y.; Aggarwal, V.; Chowdhury, A.; Poursaeed, O.; Hoffman, J.; et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR: Brookline, MA, USA, 2023; pp. 29441–29454.
43. Zhang, C.; Liu, L.; Cui, Y.; Huang, G.; Lin, W.; Yang, Y.; Hu, Y. A Comprehensive Survey on Segment Anything Model for Vision and Beyond. *arXiv* **2023**, arXiv:2305.08196. [\[CrossRef\]](#)
44. Chen, T.; Mai, Z.; Li, R.; lun Chao, W. Segment Anything Model (SAM) Enhanced Pseudo Labels for Weakly Supervised Semantic Segmentation. *arXiv* **2023**, arXiv:2305.05803. [\[CrossRef\]](#)
45. Jiang, P.T.; Yang, Y. Segment Anything is A Good Pseudo-label Generator for Weakly Supervised Semantic Segmentation. *arXiv* **2023**, arXiv:2305.01275. [\[CrossRef\]](#)
46. Cen, J.; Wu, Y.; Wang, K.; Li, X.; Yang, J.; Pei, Y.; Kong, L.; Liu, Z.; Chen, Q. SAD: Segment Any RGBD. *arXiv* **2023**, arXiv:2305.14207. [\[CrossRef\]](#)
47. Yao, J.; Wang, X.; Ye, L.; Liu, W. Matte Anything: Interactive Natural Image Matting with Segment Anything Models. *arXiv* **2023**, arXiv:2306.04121. [\[CrossRef\]](#)
48. Lillrank, D.O.; Akiyama, S.; Arulkumaran, K. Zero-Shot Object Manipulation with Semantic 3D Image Augmentation for Perceiver-Actor. 2023. Available online: <https://openreview.net/forum?id=pInZFikIRc9> (accessed on 9 December 2024).
49. Xiao, Z.; Bai, J.; Lu, Z.; Xiong, Z. A Dive into SAM Prior in Image Restoration. *arXiv* **2023**, arXiv:2305.13620. [\[CrossRef\]](#)
50. Zhang, Z.; Wei, Z.; Zhang, S.; Dai, Z.; Zhu, S. UVOSAM: A Mask-free Paradigm for Unsupervised Video Object Segmentation via Segment Anything Model. *arXiv* **2023**, arXiv:2305.12659. [\[CrossRef\]](#)
51. Cheng, Y.; Li, L.; Xu, Y.; Li, X.; Yang, Z.; Wang, W.; Yang, Y. Segment and Track Anything. *arXiv* **2023**, arXiv:2305.06558. [\[CrossRef\]](#)
52. Yang, J.; Gao, M.; Li, Z.; Gao, S.; Wang, F.; Zheng, F. Track Anything: Segment Anything Meets Videos. *arXiv* **2023**, arXiv:2304.11968. [\[CrossRef\]](#)
53. Zhang, D.; Liang, D.; Yang, H.; Zou, Z.; Ye, X.; Liu, Z.; Bai, X. SAM3D: Zero-Shot 3D Object Detection via Segment Anything Model. *arXiv* **2023**, arXiv:2306.02245. [\[CrossRef\]](#)
54. Yang, Y.; Wu, X.; He, T.; Zhao, H.; Liu, X. SAM3D: Segment Anything in 3D Scenes. *arXiv* **2023**, arXiv:2306.03908. [\[CrossRef\]](#)
55. Fan, Z.; Pan, P.; Wang, P.; Jiang, Y.; Xu, D.; Jiang, H.; Wang, Z. POPE: 6-DoF Promptable Pose Estimation of Any Object, in Any Scene, with One Reference. *arXiv* **2023**, arXiv:2305.15727. [\[CrossRef\]](#)
56. Chen, J.; Sun, M.; Bao, T.; Zhao, R.; Wu, L.; He, Z. ZeroPose: CAD-Model-based Zero-Shot Pose Estimation. *arXiv* **2023**, arXiv:2305.17934. [\[CrossRef\]](#)
57. Cheng, J.; Ye, J.; Deng, Z.; Chen, J.; Li, T.; Wang, H.; Su, Y.; Huang, Z.; Chen, J.; Sun, L.J.H.; et al. SAM-Med2D. *arXiv* **2023**, arXiv:2308.16184. [\[CrossRef\]](#)
58. Ye, J.; Cheng, J.; Chen, J.; Deng, Z.; Li, T.; Wang, H.; Su, Y.; Huang, Z.; Chen, J.; Jiang, L.; et al. SA-Med2D-20M Dataset: Segment Anything in 2D Medical Imaging with 20 Million masks. *arXiv* **2023**, arXiv:2311.11969. [\[CrossRef\]](#)

59. Ma, J.; He, Y.; Li, F.; Han, L.; You, C.; Wang, B. Segment Anything in Medical Images. *arXiv* **2023**, arXiv:2304.12306. [[CrossRef](#)] [[PubMed](#)]
60. Zhang, K.; Liu, D. Customized Segment Anything Model for Medical Image Segmentation. *arXiv* **2023**, arXiv:2304.13785. [[CrossRef](#)]
61. Wu, J.; Ji, W.; Liu, Y.; Fu, H.; Xu, M.; Xu, Y.; Jin, Y. Medical SAM Adapter: Adapting Segment Anything Model for Medical Image Segmentation. *arXiv* **2023**, arXiv:2304.12620. [[CrossRef](#)]
62. Lin, X.; Xiang, Y.; Zhang, L.; Yang, X.; Yan, Z.; Yu, L. SAMUS: Adapting Segment Anything Model for Clinically-Friendly and Generalizable Ultrasound Image Segmentation. *arXiv* **2023**, arXiv:2309.06824. [[CrossRef](#)]
63. Wada, K. Labelme: Image Polygonal Annotation with Python. *Zenodo* **2021**. [[CrossRef](#)]
64. Prasetyo, E.; Suciati, N.; Faticah, C. A Comparison of YOLO and Mask R-CNN for Segmenting Head and Tail of Fish. In *Proceedings of the 2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, Indonesia, 10–11 November 2020; pp. 1–6. [[CrossRef](#)]
65. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. [[CrossRef](#)]
66. Thai, T.T.; Ku, K.B.; Le, A.T.; Oh, S.S.M.; Phan, N.H.; Kim, I.J.; Chung, Y.S. Comparative analysis of stomatal pore instance segmentation: Mask R-CNN vs. YOLOv8 on Phenomics Stomatal dataset. *Front. Plant Sci.* **2024**, *15*, 1414849. [[CrossRef](#)]
67. Nanni, L.; Fusaro, D.; Fantozzi, C.; Pretto, A. Improving Existing Segmentors Performance with Zero-Shot Segmentors. *Entropy* **2023**, *25*, 1502. [[CrossRef](#)]
68. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
69. Rahman, M.A.; Wang, Y. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In *Advances in Visual Computing, Proceedings of the 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, 12–14 December 2016*; Bebis, G., Boyle, R., Parvin, B., Koracin, D., Porikli, F., Scaff, S., Entezari, A., Min, J., Iwai, D., Sadagic, A., et al., Eds.; Springer: Cham, Switzerland, 2016; pp. 234–244.
70. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Jorge Cardoso, M. 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, Proceedings of the Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, 14 September 2017*; Cardoso, M.J., Arbel, T., Carneiro, G., Syeda-Mahmood, T., Tavares, J.M.R., Moradi, M., Bradley, A., Greenspan, H., Papa, J.P., Madabhushi, A., et al., Eds.; Springer: Cham, Switzerland, 2017; pp. 240–248.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.