ADDITIONAL INFORMATION

Beyond Traditional AI: Foundation Models Improving Polyp Detection & Segmentation

Uxue Delaquintana-Aramendi*, Leire Benito-del-Valle*, Aitor Alvarez-Gila, Javier Pascau, Luisa F Sánchez-Peralta, Artzai Picón, J Blas Pagador, Cristina L Saratxaga,

I. Text Prompt Analysis

In object detection models, the choice of input prompts plays a critical role in determining the accuracy and relevance of the predictions. This section examines the impact of text prompts on detection performance for two foundation detection models: YOLO-World [1] and GroundingDINO [2]. Through this analysis, we explore how prompt specificity and natural language capabilities influence the effectiveness of object identification within a medical imaging context, specifically in colonoscopy datasets. The evaluation was conducted using the publicly available PICCOLO colonoscopy dataset [3]. The primary objective was to identify the optimal text prompt for the polyp detection task.

A. *YOLO-World*

In the case of YOLO-World several context-specific prompts were tested, polyp, tumor, and lump, but none of them succeeded in predicting any bounding boxes (Figure 1).

Having seen the previous results, the inference was run without any text prompt to access the model's intrinsic natural language. By doing so, proper object detection and bounding boxes were obtained, even if the given labels were not coherent in the colonoscopy context: "apple", "cake", "pizza", "hot dog"... as seen in Fig. 2.

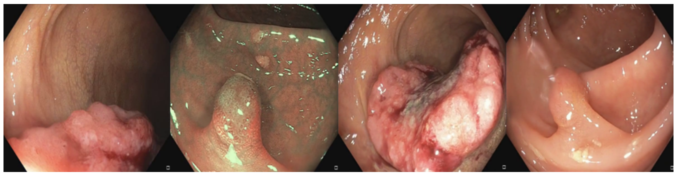

Fig. 1: **Examples of YOLO-World detection with polyp prompts on the PICCOLO dataset.**

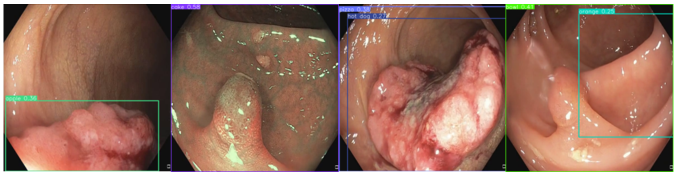

Fig. 2: **Examples of YOLO-World detection without specific prompts, using natural language, on the PICCOLO dataset.**

With this setting, the model iterated over the test subset of PICCOLO and predict the bounding boxes for the objects found. The specifications and functions needed to run the prediction were sourced from the Ultralytics repository and adapted accordingly [4].

B. *GroundingDINO*

For GroundingDINO the prompt "polyp" successfully performed the expected detection. However, utilizing context-specific yet more general prompts like "tumor" or "lump" enhanced the detection: see Table I. As such, the most effective prompt for evaluation was chosen: "lump".

TABLE I: Grounding DINO Detection Metrics by Prompt

Prompt	Average Precision ↑	Average Recall ↑
'polyp'	0.114	0.427
'tumor'	0.115	0.446
'lump'	**0.133**	**0.520**

II. SAM vs MedSAM

MedSAM [5] was introduced as a specialized version of SAM (Segment Anything Model) [6] for medical segmentation tasks. Seeing that endoscopy images comprise the 22% (around 240,000 images) of the entire training dataset and that an accuracy of 98.5% was reported for polyp segmentation, it was only natural to verify the validity of these assertions.

Three versions of the SAM model are available with different backbone sizes: SAM Base (B), SAM Large (L) or SAM Huge (H). The smallest, SAM B, is chosen for evaluation because MedSAM's fine-tuning stems from it.

Submited 2025/03/27.

* Equal contribution.

This work was supported in part by the Elkartek Programme, Basque Government (Spain) (BEREZ-IA, KK-2023/00012 and IKUN, KK-2024/00064).

U. DelaQuintana-Aramendi was with TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia, 48160 Spain and with Universidad Carlos III de Madrid, Madrid, 28911 Spain. She is now with Artificial Intelligence in Science Institute (AISI) at University of California Irvine, CA 92617 USA (e-mail: udelaqui@uci.edu).

L. Benito-del-Valle is with TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia, 48160 Spain (e-mail: leire.benitodelvalle@tecnalia.com).

A. Alvarez-Gila is with TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia, 48160 Spain (e-mail: aitor.alvarez@tecnalia.com).

J. Pascau is with the Department of Bioengineering at Universidad Carlos III de Madrid, Madrid, 28911 Spain and with Instituto de Investigación Sanitaria Gregorio Marañón, Madrid, 28009 Spain (e-mail: jpascau@ing.uc3m.es).

L. F Sánchez-Peralta is with Jesús Usón Minimally Invasive Surgery Centre, Cáceres, 10071 Spain and AI4polypNET Thematic Network, Barcelona, 08193 Spain (e-mail: lfsanchez@ccmijesususon.com).

A. Picón is with TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia, 48160 Spain and University of the Basque Country, Plaza Torres Quevedo, 48013 Bilbao, Spain (e-mail: artzai.picon@tecnalia.com).

J. Blas Pagador is with Jesús Usón Minimally Invasive Surgery Centre, Cáceres, 10071 Spain and AI4polypNET Thematic Network, Barcelona, 08193 Spain (e-mail: jbpagador@ccmijesususon.com).

C. L Saratxaga is with TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia, 48160 Spain (e-mail: Cristina.Lopez@tecnalia.com).

The inference code used to run the experiment was sourced from the MedSAM GitHub repository [5] and adapted for the polyp segmentation task (instance segmentation) and the PICCOLO public colonoscopy dataset [3]. Switching between SAM and MedSAM models only requires modifying the path to initialize the corresponding weights.

PICCOLO's validation set (897 images) was used to run the experiments. To ensure compatibility, the code performs a minor preprocessing of the images before feeding them into the model. Images are resized into 1024x1024 and grayscale images (2 channels) are converted to RGB domain (3 channels). Lastly, the pixel values are normalized to match the expected intensity range [0,255]. The same process is applied to the bounding box prompts to ensure proper alignment.

Although both SAM and MedSAM are designed for interactive, real-time segmentation with user-provided prompts, for testing automated prompts had to be generated. At first, boxes were simulated following the authors' training specifications [5]: for each bounding box derived from the ground truth masks, a random perturbation between 0-20 pixels was added to each side. The results, as shown in Table II, indicated poor performance under these initial conditions. The values were not as high as expected and MedSAM performed worse than SAM, which was not logical given the context. It was concluded that the randomness of this process prevented a consistent evaluation of the models.

TABLE II: Bounding Box Suitability Evaluation for SAM and MedSAM

Model	Random Box mIoU ↑	Fixed Box mIoU ↑
SAM	0.842	**0.856**
MedSAM	0.809	**0.928**

The authors did not specify the design of bounding boxes for testing that yielded such high results. Therefore, in order to replicate their values, the closest variation to the ground truth boxes was implemented by adding a fixed augmentation of 10 pixels per side. It was observed (table II) that the fixed boxes approach not only improved the overall performance but, contrary to the random boxes, it effectively enhanced the performance of MedSAM over SAM, fulfilling the expectations. Additionally, an accuracy of almost 93% was achieved, which aligns with the 98.5% accuracy reported by the authors [7].

III. Dataset Splits and Image Selection

A. PICCOLO

For the PICCOLO dataset, the splitting criteria was extracted from the original article [3]. The 3,433 annotated images were distributed into the train, validation, or test set, with 2,203 (64.17%), 897 (26.13%), and 333 (9.70%) frames, respectively. In order to ensure patient independence between sets, lesions originating from the same colonoscopy exploration video were assigned to the same set.

B. PolypSegm-ASH

Similar to PICCOLO, for PolypSegm-ASH [8] the distribution created by the authors was directly employed. The original paper defines three stratified splits with 788, 113 and 224 images for train, validation and test respectively. Special attention was paid to ensure that images from the same polyp were all included in the same set. As noted in the main section of this article, PolypSegm-ASH is part of the iSMIT 2024 Polyp Segmentation Challenge; which is why at the time of access for this study, annotations for the test set were not publicly available. Since professionally annotated masks are essential for the accurate evaluation of this study, the validation set was repurposed for testing.

C. SUN-SEG

The SUN-SEG dataset [9] comprises 158,690 colonoscopy images in total, categorized into positive and negative frames based on the presence of polyps.

For positive images, images are further divided into 100 cases, with each case corresponding to a unique polyp. These cases were randomly assigned into training, validation and test sets, resulting in a primary split of 70:20:10. Within each case the number of frames varies, but, all were selected. This resulted in 37,999 images for training, 8,865 for validation and 2,272 for testing. Given the total number of annotated images (49,136), the post-split distribution corresponds to: 77.33% train, 18.04% validation, 4.62% test.

In the case of negative images, 13 different cases can be found. Following the same 70:20:10 split, cases were randomly assigned, yielding approximately 9 cases for training, 3 for validation, and 1 for testing. The total number of images required for each set (37,999 for training, 8,865 for validation, and 2,272 for testing) was divided by the respective number of cases. This calculated the number of images that should be retrieved from each case for equal representation. For clarity, let's call this number the divider. Then, the total number of images inside each case was divided by the divider to obtain a value denoted as the temporal jump. The temporal jump is used to select the desired number of images per set with sufficient spacing to ensure variability and randomness in the dataset; since subsequent frames appeared to be very similar.

For instance, if case 5 was randomly assigned to the training set, the total number of expected training images (37,999) divided by the number of training cases (9) resulted in a divider of 4,222.1, rounded to 4,222. If case 5 contained 16,888 images, then the temporal jump was calculated as 4, meaning every fourth frame was selected.

Finally, the positive and negative splits were merged to form the final dataset with a total of 98,272 images. The dataset includes corresponding mask and bounding box annotations for positive images. The final distribution consists of 75,997 images for training, 17,730 for validation, and 4,544 for testing.

IV. Training

A. YOLOv8

For further training on YOLOv8, the specifications found in the Ultralytics YOLO repository were followed [10].

First and foremost, the dataset employed must take the required format for all YOLO models:

The data was organized into two folders in the training environment: “images” and “labels”. It is crucial that the folders are named as such. Each contained sub-folders for train, test, and validation. Images were added in their original resolution and file name. For the labels, the bounding boxes of all the objects annotated in each mask had to be stored under the same text file. Label files’ names matched their corresponding image’s and the bounding boxes used the format: [class label, x center, y center, width, height]. Lastly, a YAML file was created with the pathways and details needed for the model to access the dataset.

YOLO-based models work with square images and resize them to 640x640 by default. Yet in training, this image size can also be specified by the user. The used resolutions for the three datasets are 608x608 for PICCOLO (600x480 original), 1152x1152 for SUN-SEG (1240x1080 original) and 1440x1440 for PolypSegm-ASH (1280x1024 original). The image size had to be a multiple of the maximum stride size used by the neural network, which was 32 in this case.

To achieve FT (Fine-Tuned) YOLOv8, the YOLOv8 Small architecture was trained for 100 epochs on the default parameters: batch size 16, ”Auto” optimizer, initial learning rate of 0.01 and weight decay 0.0005, among others. See Ultralytics repository for more detail [10].

Once the training was completed, the process was evaluated in terms of the loss functions for the train and validation data (Fig. 3). Monitoring both train and validation losses ensures that the model is not only performing well on the training data but also generalizing effectively to unseen data (Fig. 3).

In this case, multiple loss components are plotted: the box loss, the class loss and the Distribution Focal Loss (DFL). Each component measures the error in the predictions for the bounding boxes, the class labels and the hard-to-classify examples, respectively. Among these, box loss was the most critical for the task at hand but it was equally important to observe a decreasing trend across all of them. Notably, in Fig. 3 the box loss function decreases along with the train loss, indicating that the model is learning useful patterns that generalize well. Regarding differences between validation and training losses, if the validation loss were to increase while the training loss decreases, it would suggest over-fitting. Consequently, the absence of this divergence in Fig. 3 is a positive sign.

B. YOLO-World

The setup for fine-tuning of YOLO-World was almost identical to that of YOLOv8. FT YOLO-World was obtained by loading the pretrained weights of the YOLOv8s-WORLDv2 model and training it on the default parameters (See Ultralytics for details [4]), with the datasets in the YOLO format, also for 100 epochs.

The loss functions for YOLO-World (Fig. 4) exhibit a similar behavior to that observed in YOLOv8 (Fig. 3), confirming that the model is learning effectively for generalization.

C. Mask R-CNN

The training parameters of Mask R-CNN [11] were optimized with a particular focus on the learning rate (LR). The LR is a hyper-parameter that controls the size of the steps the model takes when updating its weights during training. Hence, it determines how quickly or slowly a model learns. The smaller the LR is, the smaller the steps and the slower the learning process will be, which in complex tasks and contexts, like the medical domain, is usually beneficial.

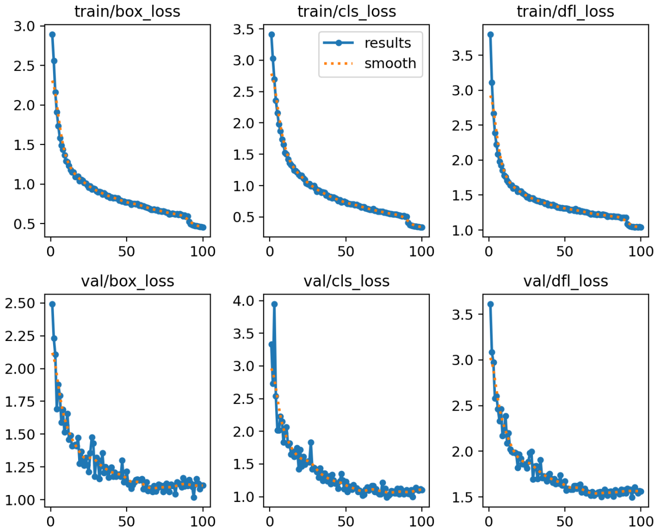

Fig. 3: **Training and validation losses on YOLOv8.** From left to right, box loss, class loss and Distribution Focal Loss (DFL) plots. Above, training losses, below, validation ones.

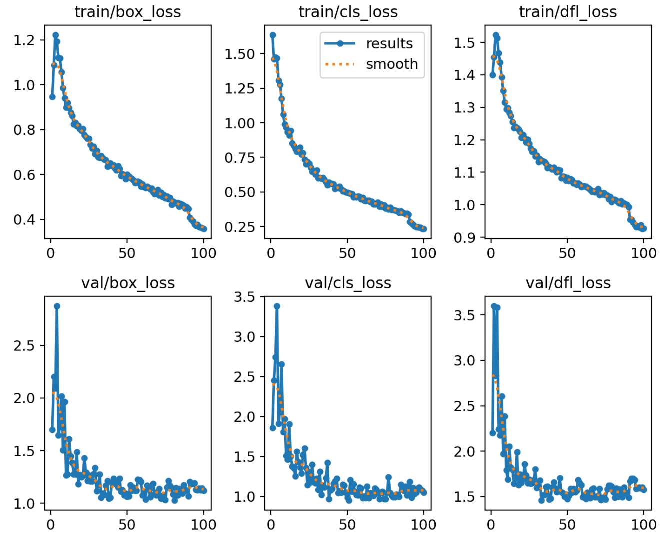

Fig. 4: **Training and validation losses on YOLO-World.** From left to right, box loss, class loss and Distribution Focal Loss (DFL) plots. Above, training losses, below, validation ones.

Through a series of experiments, we identified the optimal initial LR for this task. Fig. 5 for detailed results. Since the model generates both bounding boxes and segmentation masks, we evaluated performance based on detection accuracy, using average precision on the validation set, and segmentation accuracy, measured by mean IoU: Shown in Table III. This analysis determined that an LR of 0.00001 yielded the best results.

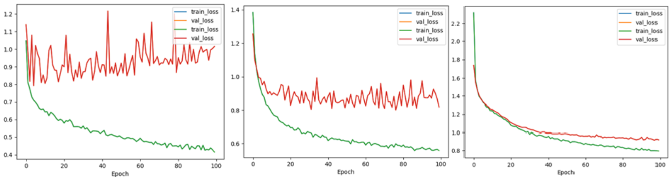

Fig. 5:]
Training and Validation Losses on Mask R-CNN Parameter Optimization. Training curve is shown in green, validation in red.

It is important to note that parameter optimization was performed on the original ResNet50-FPN backbone [11], and then implemented on the DINOv2 [12]-FPN architecture we created. This likely gives the original ResNet50 backbone an advantage over DINOv2 but aligns with our initial goal of minimal adaptation for foundation models.

TABLE III: Mask R-CNN PICCOLO Detection and Segmentation Metrics by Learning Rate

Learning Rate	Average Precision ↑	Mean IoU ↑
0.0001	0.578	0.671
0.00001	**0.614**	**0.681**
0.000001	0.555	0.651

Apart from the learning rate, other parameters that contributed to training were:

- Epochs = 100: 100 epochs were selected for the fine-tuning of all models.
- Box Detections per Image = 10: Sets the maximum number of detections per image to 10 for all classes.
- RPN Score Threshold = 0.55: Adjusts the model to only return proposals with an object score greater than 0.55.
- RPN NMS Threshold = 0.7: Uses a NMS threshold of 0.7 for postprocessing the proposals. Non-maximum suppression (NMS) is a technique used to eliminate duplicate detections.
- RPN Pre-NMS Top N Train = 20: Keeps a maximum of 20 proposals before applying NMS during training.
- RPN Post-NMS Top N Train = 20: Keeps a maximum of 20 proposals after applying NMS during training.

Mask R-CNN also resizes the input images for training. In this case, inputs size of 600x600 for PICCOLO, 800x800 for SUN-SEG, and 1440x1440 for PolypSegm-ASH, were defined.

D. Grounding DINO

Fine-tuning of GroundingDINO was conducted using the MMDetection implementation with the default training parameters [13]. The model was trained for 100 epochs using the Swin-Tiny backbone for single-class detection ("polyp") with an initial learning rate of 0.00005. The learning rate was adjusted using a linear Warm-up for the first 30 iterations with a factor of 0.001, followed by a multi-step learning rate decay at epoch 15 with a decay factor of 0.1. The optimizer used was AdamW, with different learning rate multipliers applied to various parts of the model: the backbone's learning rate was scaled by 0.1x, while the language model parameters were frozen.

The batch size was set to 16 images and the input images were used in their original resolution. Annotations were provided in COCO format with bounding box coordinates and class labels in JSON files.

References

[1] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[2] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[3] L. F. Sanchez-Peralta, J. B. Pagador, A. Picon, A. J. Calderon, F. Polo, N. Andraka Rueda, R. Bilbao, B. Glover, C. Saratxaga, and F. Sánchez-Margallo, "Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets," *Applied Sciences*, vol. 10, no. 12, p. 8501, Dec. 2020.

[4] Ultralytics, "Yolo-world models," 2024, accessed: 2024-06-04. [Online]. Available: https://docs.ultralytics.com/models/yolo-world/

[5] B. W. Lab, "Medsam: Segment anything model for medical images," https://github.com/bowang-lab/MedSAM, 2024, accessed: 2024-11-17.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[7] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, no. 1, p. 654, Jan. 2024.

[8] Y. Tudela, A. García-Rodríguez, G. Fernández-Esparrach, and J. Bernal, "Towards fine-grained polyp segmentation and classification," in *Clinical Image-Based Procedures, Fairness of AI in Medical Imaging, and Ethical and Philosophical Issues in Medical Imaging*, S. Wesarg, E. Puyol Antón, J. S. H. Baxter, M. Erdt, K. Drechsler, C. Oyarzun Laura, M. Freiman, Y. Chen, I. Rekik, R. Eagleson, A. Feragen, A. P. King, V. Cheplygina, M. Ganz-Benjaminsen, E. Ferrante, B. Glocker, D. Moyer, and E. Petersen, Eds. Cham: Springer Nature Switzerland, 2023, pp. 32–42.

[9] M. Misawa, s.-e. Kudo, Y. Mori, K. Hotta, K. Ohtsuka, T. Matsuda, S. Saito, T. Kudo, T. Baba, F. Ishida, H. Itoh, M. Oda, and K. Mori, "Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video)," *Gastrointestinal Endoscopy*, vol. 93, 07 2020.

[10] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8 documentation," https://docs.ultralytics.com/models/yolov8/, 2024, accessed: 2024-06-09.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 06 2018.

[12] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[13] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.