**SegmentR: Deep Learning in R for Automated Segmentation**

James D. Boyko12   
1 Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, Michigan 48109, USA  
2 Michigan Institute of Data Science, University of Michigan, Ann Arbor, Michigan 48109, USA

**Abstract**

Citizen science platforms like iNaturalist generate biodiversity data at an unprecedented scale, with observations on the order of hundreds of millions. However, extracting phenotypic information from these images, such as color of organisms, at such a large scale poses unique challenges for biologists. Some of the challenges are that manual extraction of phenotypic information can be subjective and time-consuming. Fortunately, with the maturation of computer vision and deep learning, there is an opportunity to automate large parts of the image processing pipeline. Here, I present SegmentR, a user-friendly software package that leverages two state-of-the-art deep learning models - GroundingDINO and SegmentAnything - to enable automated segmentation and color extraction from images. The SegmentR package provides an R-based interface, making it more accessible to evolutionary biologists and ecologists who may not have extensive coding experience. The SegmentR pipeline allows users to load images, automatically segment them based on text prompts, and extract color information from the segmented regions. The package also includes visualization and data summarization functions to facilitate downstream analysis and interpretation of the results.

**Key Words**

color, color pattern, citizen science, image segmentation, deep learning, iNatauralist

**1. INTRODUCTION**

Citizen science initiatives like iNaturalist generate data in the form of crowdsourced observations and images, centralizing vast quantities of biodiversity data and providing opportunity to study color variation across the tree of life and at large spatial scales. Yet, the sheer scale of the data generated and the fact that it is typically not standardized, make it difficult to automate data collection and extraction. As such, most attempts to utilize resources like iNaturalist rely on manually extracting information image by image. Though this approach almost certainly results in high-quality datasets, for even relatively simple tasks like color quantification, extracting information from images will be time-consuming and likely require large teams of willing annotators. Furthermore, manual color extraction can be subject to measurement error, e.g. what one observer calls white another may call light green. Fortunately, the maturation of computer vision and deep learning techniques offers a promising solution to this challenge. By leveraging state-of-the-art machine learning models, it is now possible to automate large parts of the image processing pipeline, allowing biologists to extract phenotypic information from citizen science data at unprecedented scales.

Many of the most successful deep learning applications in recent years have been for computer vision tasks (LeCun et al. 2015; Goodfellow et al. 2016). Two key areas of computer vision that are particularly relevant for extracting color data from biodiversity images are object detection and instance segmentation. Object detection algorithms aim to locate and classify distinct objects within an image (Redmon et al. 2016; Ren et al. 2017), while instance segmentation models, delineate pixel-level boundaries of each instance of an object (He et al. 2017; Kirillov et al. 2020). By combining these capabilities, it becomes possible to focus color analysis only on the specific organisms of interest, rather than the entire scene. However, training a specialized computer vision model from scratch would require large, annotated datasets. This would be prohibitively time-consuming to collect, especially for diverse taxonomic groups. Fortunately, progress in computer vision has led to the development of powerful pre-trained models that can be effectively leveraged for a variety of tasks with minimal fine-tuning (Zhai et al. 2022).

In this work, I utilize two such pre-trained models: GroundingDINO (Liu et al. 2024) for object detection and a data efficient version of the Segment Anything Model (SAM) (Kirillov et al. 2022; Chen et al. 2024) for instance segmentation. To provide an accessible entry point for ecologists and evolutionary biologists, I have implemented this deep learning-powered pipeline as both a set of Python scripts and an R package called SegmentR. The R framework gives users who are already familiar with the R programming language an easier integration with their existing workflows and analytical tools. Additionally, the Python scripts are available for those researchers who may prefer a more customizable, low-level approach to image processing and analysis. The base SegmentR package performs object detection and instance segmentation allowing for individual instances of focal taxa to be extracted from an image. These individual instances then allow for the extraction of color pixel-by-pixel. I demonstrate the SegmentR workflow for several examples, showing also how one would assess the quality of the segmentation and color extraction. Finally, I outline potential pitfalls researchers may experience when using this and other automatic object detection and segmentation software.

**2. METHODS AND RESULTS**

**2.1 Object detection and instance segmentation**

Object detection and instance segmentation are fundamental tasks in computer vision (Chollet 2021). The goal of object detection is to draw bounding boxes around particular objects of interest and associate the bounding box with a particular class for a given image. However, most object detection algorithms are limited to a pre-determined set of classes. This is problematic for biological datasets as existing pre-trained models are unlikely to have been trained on all taxonomic groups of interest and adding new classes would require collecting and labeling new data in order to retrain the model. Attempts to address this challenge have focused on combining visual and textual modalities, with Grounding DINO successfully generalizing object detection (Liu et al. 2024). Grounding DINO is a transformed-based architecture that fuses language and vision modalities by linking the closed-set detector, DINO (Zhang et al. 2022), with grounded language-based pre-training (e.g., GLIP; Li et al. 2022). The effect of this design is that Grounding DINO is able to detect arbitrary objects based on diverse text prompts. This model has been extended in several ways, including combining it with the instance segmentation model, SegmentAnything (SAM; Kirillov et al. 2022). SAM, is an image segmentation model trained on the largest segmentation dataset with over 1 billion masks and 11 million images. This allows it to achieve consistently high performance on zero-shot segmentation tasks even when compared to fully supervised models (Kirillov et al. 2022). The combination of Grounding DINO and SAM is called GroundedSAM (Ren. et al 2024) and it uses the bounding box output of Grounding DINO as the input of SAM for high quality instance segmentation. This approach can be further refined by using recently developed efficient versions of SAM such as SlimSAM, which achieve high accuracy while using far less training data (Chen et al. 2024). SlimSAM result in a model a fraction of the size of the original SAM (1.4% of the original parameters) and is ideally suited for biological research as the workflow can be run on moderately powerful personal computers.

**2.2 SegmentR features and examples**

*2.2.1 SegmentR installation*

SegmentR is based on a python implementation of pre-trained deep learning models. As such users must have python3 installed on their computers to use SegmentR. Furthermore, to install SegmentR one must have the conda package and environment manager. Thus, once python is installed, the user should install either Anaconda3 or Miniconda. Note that Miniconda has been reported to work when Anaconda did not. Once these prerequisites are installed, SegmentR can be installed from github with the use of devtools (Wickham et al. 2022) and the command install\_github(“jboyko/SegmentR”). Once the package has been installed, the function setup\_conda\_environment() can be used to install the necessary python dependencies. setup\_conda\_environment which will install all the necessary libraries with either exact version numbers known to work with SegmentR (env\_type= “specific”) or with the minimum constraints on the required libraries (env\_type= “general”). I recommend using env\_type= “general” in most cases, noting that it takes longer to setup the environment than env\_type= “specific” because library versions must be solved to ensure compatibility. During the conda environment setup, a warning message may appear if the user does not have the rust programming language installed on their computer. This is related to the transformers library, but can be safely ignored as the functions being used from transformers for SegmentR do not require rust.

*2.2.2 Description of the SegmentR pipeline*

Using SegmentR requires a user to input an image (specified by the path to the image) and a set of labels. The primary function of SegmentR, grounded\_segmentation\_cli (Table 1), then preforms grounded segmentation based on the input image and labels. This function creates a custom command-line interface (CLI) tool which interacts with a Python back-end. The Python back-end then uses the transformers library, which is itself an API to download and train pre-trained models, to call the segmenter and detector models (Wolf et al. 2020). Note that once the desired models are downloaded and necessary libraries installed, the SegmentR software can be used entirely offline. Results are saved as a JSON file which can then be read into R using the function load\_segmentation\_results. Once the grounded segmentation is complete and the results are loaded into R, the segmentation can be plotted using plot\_seg\_results and subsequent analysis can be conducted on the resulting masks.

Table 1: Primary functions.

|  |  |  |
| --- | --- | --- |
| **Function** | **Arguments** | **Definition** |
| run\_grounded\_segmentation | path | Character string. Path to an input image or directory containing images. E.g., “/home/user\_name/project\_directory/images/” |
| labels | Character vector. Labels to detect in the image.  E.g., c(“a flower”, “a tree”). Labels are automatically appended with a “.” because it substantially improves the performance of groundingDINO. |
| threshold | Numeric. The minimum detection threshold for which detected objects will be segmented (default: 0.3). |
| detector\_id | Character string. ID of the detector model.  default: “IDEA-Research/grounding-dino-tiny” |
| segmenter\_id | Character string. ID of the detector model.  default: “Zigeng/SlimSAM-uniform-77” |
| output\_plot | Character string. Path or directory to save the output plot. |
| output\_json | Character string. Path or directory to save the output JSON. |
| show\_plot | Boolean. Whether a plot should automatically be shown after a segmentation. |
| create\_dir | Boolean. If the output directory doesn't exist, one will be created. |
| pattern | Character string. File pattern to match when path is a directory (default: "\\.(jpg|jpeg|png)$"). |
| recursive | Boolean. Whether to search for images recursively in subdirectories (default: FALSE). |
| conda\_env | Character string. Name of the conda environment to use.  default: “SegmentR-env” created by setup\_conda\_environment |
| load\_segmentation\_results | image\_path | Character string. Path to the original image file. |
| json\_path | Character string. Path to the JSON file containing segmentation results. The JSON file is created by grounded\_segmentation\_cli |
| export\_transparent\_png | input | Either a seg\_results list or image data (cimg object, array, or file path) |
| masks | Optional list of masks or single mask array (if not provided in seg\_results) |
| labels | Optional vector of labels for each mask |
| scores | Optional vector of confidence scores |
| output\_path | Character. Path where files should be saved. |
| score\_threshold | Numeric. Threshold for including results (0-1) |
| remove\_overlap | Boolean. Whether to remove any overlapping regions from masks with different labels. |
| return\_binary | Boolean. Whether only a binary mask should be returned. |
| crop | Boolean. Whether to crop the image to the edges of the segment. |
| prefix | Character. Prefix for output filenames (default: NULL uses "segment") |
| include\_score | Logical. Whether to include confidence score in filename (default: TRUE) |
| id\_padding | Integer. Number of digits to pad mask IDs with (default: 3) |
| plot\_seg\_results | seg\_results | A list containing segmentation results (image, label, score, box, mask). |
| mask\_colors | A named vector of colors for each label, or a color palette name from RColorBrewer. |
| background | One of "original", "grayscale", "transparent", or a specific color. |
| show\_label | Boolean Whether to display labels |
| show\_score | Boolean Whether to display scores |
| show\_bbox | Boolean Whether to display bounding boxes. |
| ... | Additional arguments to be passed to plot. |

*Example 1 – Specifying particular parts of an organism*

The final example demonstrates the power of grounding DINO in specifying and isolating particular parts of an organism for focused color analysis. This capability is important when color patterning is localized to specific areas of an organism. The subject of this analysis is a horned bream, where the main color patterning of interest is located on the body. The challenge lies in excluding the fins, which do not contain the coloration of interest. By leveraging the broad textual understanding provided by grounding DINO, we can use text prompts to detect and exclude various fin types from the analysis (Figure 4). This example highlights both the strengths and limitations of the current implementation. While the algorithm successfully identified and excluded the more prominent caudal and pelvic fins, it struggled with the detection of dorsal, anal, and pectoral fins. This suggests that, despite groundingDINO’s generality, fine-tuning the model on domain-specific datasets will still be necessary in some cases.

ground\_results <- grounded\_segmentation\_cli(

image\_path = "/home/user/project\_file/img.jpg",

labebls = c("a fish.", "the fins of a fish."),

output\_json = "/home/user/project\_file/json/",

output\_plot = "/home/user/project\_file/plot/")

Figure - a) The original image. b) Object detections (bounding boxes) and segmentations (red and blue colored masks) of the original image. c) The recolorized image when the entire “fish” object is used. d) The recolorized image when detected “fins” are removed f

*Example 2 – Batch segmentation of a small dataset*

Figure – outcome of batch processing (39 seconds for object detection, segmentation, and processing)

A potential use… Here I use a dataset of four images sourced from inaturalist containing both flowers in bees. For my analysis, I am interested soley in the color of the flowers which would mean both removing the background of the image and removing the bee. I use SegmentR as a preprocessing step for subsequent color analysis. I compare SegmentR pipeline to the time needed to manually extract the masks using imageJ.

**2.3 Limitations**

The zero-shot object detection capabilities of groundingDINO (Feng et al., 2023) offer significant advantages when processing non-standardized citizen science data, particularly in the context of biodiversity studies. The heterogeneous nature of images on platforms like iNaturalist, where organisms may be positioned anywhere within the frame, camouflaged against diverse backgrounds, or present alongside other species, highlights the value of general text-prompt-based object detection. However, it is important to note that improved object detection algorithms cannot address all inherent biases present in citizen science data. A primary concern is the inconsistency in lighting conditions across images, which can lead to shadow-induced distortions in color values (Szeliski, 2022). While some shadow removal techniques based on luminescence values have been developed (e.g., Murali & Govindan, 2013), their efficacy when applied to the diverse lighting conditions encountered in iNaturalist images has been limited. One potential approach to mitigate these lighting-related issues involves the use of pre-specified color palettes. Within the SegmentR framework, this can be implemented through the custom\_colors argument in the process\_masks\_and\_extract\_colors function. This method clusters pixel colors based on their distance to predefined custom colors rather than employing k-means clustering, potentially offering more consistent results across varying lighting conditions.

Beyond these technical limitations, citizen science datasets are subject to broader constraints that merit consideration. A significant issue is the presence of sampling effort biases (Dickinson et al., 2010). The majority of data is generated from North America, particularly the United States, despite the fact that global biodiversity is concentrated in tropical regions, which remain critically undersampled (e.g. Vasconcelos, 2023). While initiatives to increase sampling efforts in tropical areas exist, current datasets exhibit substantial geographical biases that must be accounted for in analyses utilizing citizen science data (Ward, 2014). Another limitation specific to color analysis is the discrepancy between human color perception (and standard camera capabilities) and the color perception of various organisms (Endler & Mielke, 2005). Many species perceive colors in ways that differ significantly from human vision, often requiring specialized equipment to capture color accurately. The widespread adoption of such specialized imaging technology among citizen scientists contributing to platforms like iNaturalist is unlikely, potentially limiting analyses to the human-perceptible color space. However, it is worth noting that images captured with specialized equipment could still be processed and analyzed using tools like SegmentR, should they become available.

**3. DISCUSSION**

The proliferation of citizen science initiatives has led to an unprecedented accumulation of biodiversity data. However, the heterogeneous nature of these datasets presents significant challenges for standardization and analysis. Recent advancements in computer vision technologies have emerged as a promising solution to address these complexities, offering remarkable flexibility in data processing and quantification. The application of deep-learning to tasks within ecology and evolutionary biology is not new, however, to-date, most applications have focused on models trained from scratch (e.g., Weaver and Smith 2023). A more powerful, accessible, and environmentally responsible way forward for academics may be the utilization of large pre-trained models. These models are trained on extensive datasets which few academic collaborations are able to match (e.g., the billion masks used to train SAM; Krillov et al., 2023). Fortunately, many of the underlying representations within the hidden layers of the deep-learning models are still useful in biological studies and through the use of fine-tuning, can be readily adapted to even the most obscure taxonomic groups.

Several R-packages exist for color and color pattern analysis (e.g., Van Belleghem et al., 2018; Maia et al., 2019; van den Berg et al., 2020; Weller et al. 2024). These packages have several novel analysis features and users are encouraged to use SegmentR in conjunction with existing R-packages rather than the final end-point of data collection. Color extraction from natural images poses several challenges including differentiating the signal of the focal organism from the natural background (see 2.4 Limitations). SegmentR uses object detection via groundingDINO to overcome this, but other approaches should be used in conjunction to ensure a high quality dataset (e.g., van den Berg et al., 2020). Other issues, such as the shadow distorting color, are more difficult to resolve. Using a pre-defined color palette with colors that are distant in color space may be one way around this, but shadow removal techniques are also a promising way forward (e.g., Murali & Govindan, 2013).

The implementation of deep learning tools in R is an important step towards making these techniques widely available to biologists. R is one of the most widely used programming language for academic ecologists and evolutionary biologists, but most deep-learning developments take place in Python. Furthermore, the use of light-weight models, such as slimSAM (Chen 2024), allows for advanced deep-learning models to be run even on moderately powerful personal computers. By enhancing the accessibility of these tools, biologists can gain access to an increasing number of data sources. Nonetheless the automation provided by computer vision techniques will need to be balanced with careful verification of the results. Performance is not guaranteed to be the same across all images (see example 2.3.3) and increasing the accuracy of these models on diverse taxonomic groups will likely require some amount of additional data collection and fine-tuning. Furthermore, while SegmentR allows for a more automated collection of citizen science data, it cannot address all inherent limitations of this data source. Researchers must remain cognizant of lighting inconsistencies and sampling biases when interpreting results derived from these datasets.

**4. CONCLUSION**

Color is a crucial phenotype for many organisms, however gathering large interspecific datasets has proven difficult without some amount of automation. Here I have introduced SegmentR, an R package which utilizes an underlying Python framework to automatically detect and segment organisms within citizen science photos. Using the pre-trained models groundingDINO and slimSAM, SegmentR is able to preform object detection and segmentation without training for specific taxa. Using a computer vision pipeline for automated color extraction is just one example of how deep learning techniques can be used to quantify large-scale bio-diversity datasets. SegmentR and other deep-learning software offer many exciting possibilities, but still necessitate careful consideration of the data produced. Researchers must reconcile the fact that though these tools are able to automatically extract vast quantities of data, high quality datasets will only be created if they are rigorously evaluated. As deep-learning techniques continue to integrate with ecology and evolutionary biology, it is important that they are used to complement traditional methods, rather than replace them.

**Acknowledgments**

The author thanks Thais Vasconcelos for her thoughtful feedback that greatly improved this work. This project is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Sciences Futures program.

**Data Availability Statement**

The development version of the package is available on GitHub: https://github.com/jboyko/SegmentR. All images and code used to generate the examples are taken from the vignette associated with the package. Data and image credits can be loaded using SegmentR::load\_segmentr\_example\_data.

**Conflict of Interest Statement**

The author declares no competing interests.

**References**

Chen, Z., Fang, G., Ma, X. & Wang, X. (2024). SlimSAM: 0.1% Data Makes Segment Anything Slim.

Chollet, F. (2021). *Deep learning with Python*. Second edition. Manning Publications, Shelter Island.

Cox, C. & Davis Rabosky, A. (2023). The Integrative Biology of Snake Coloration. pp. 179–217.

Dickinson, J.L., Zuckerberg, B. & Bonter, D.N. (2010). Citizen Science as an Ecological Research Tool: Challenges and Benefits. *Annual Review of Ecology, Evolution, and Systematics*, 41, 149–172.

Endler, J.A. (1993). The Color of Light in Forests and Its Implications. *Ecological Monographs*, 63, 1–27.

Finkbeiner, S.D., Briscoe, A.D. & Reed, R.D. (2014). Warning signals are seductive: Relative contributions of color and pattern to predator avoidance and mate attraction in *Heliconius* butterflies: BUTTERFLY VISUAL SIGNALS. *Evolution*, 68, 3410–3420.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press.

He, K., Gkioxari, G., Dollár, P. & Girshick, R. (2018). Mask R-CNN.

Huyghe, K., Vanhooydonck, B., Herrel, A., Tadic, Z. & Van Damme, R. (2007). Morphology, performance, behavior and ecology of three color morphs in males of the lizard Podarcis melisellensis. *Integrative and Comparative Biology*, 47, 211–220.

Internationale Beleuchtungskommission (Ed.). (2004). *Colorimetry*. Publication / CIE. 3rd ed. Comm. Internat. de l’éclairage, Wien.

Kirillov, A., Wu, Y., He, K. & Girshick, R. (2020). PointRend: Image Segmentation as Rendering.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., *et al.* (2023). Segment Anything.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.

Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., *et al.* (2022). Grounded Language-Image Pre-training.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., *et al.* (2024). Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. University of California Press, pp. 281–298.

Maia, R., Gruson, H., Endler, J.A. & White, T.E. (2019). pavo 2: New tools for the spectral and spatial analysis of colour in r. *Methods in Ecology and Evolution*, 10, 1097–1107.

Mitchem, L.D. (2017). The ecology of color: how color affects visual behaviors in aquatic habitats in a sexually dimorphic species (Lucania goodei) and their major predator (Micropterus salmoides).

Murali, S. & Govindan, V.K. (2013). Shadow Detection and Removal from a Single Image Using LAB Color Space. *Cybernetics and Information Technologies*, 13, 95–103.

Narbona, E., Arista, M., Whittall, J.B., Camargo, M.G.G. & Shrestha, M. (2021). Editorial: The Role of Flower Color in Angiosperm Evolution. *Front. Plant Sci.*, 12.

Pohland, G. & Mullen, P. (2006). Preservation agents influence UV-coloration of plumage in museum bird skins. *J Ornithol*, 147, 464–467.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection.

Ren, S., He, K., Girshick, R. & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., *et al.* (2024). Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks.

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer Nature.

Tsuchida, T., Koga, R., Horikawa, M., Tsunoda, T., Maoka, T., Matsumoto, S., *et al.* (2010). Symbiotic Bacterium Modifies Aphid Body Color. *Science*, 330, 1102–1104.

Van Belleghem, S.M., Papa, R., Ortiz-Zuazaga, H., Hendrickx, F., Jiggins, C.D., Owen McMillan, W., *et al.* (2018). patternize: An R package for quantifying colour pattern variation. *Methods in Ecology and Evolution*, 9, 390–398.

Vasconcelos, T. (2023). A trait-based approach to determining principles of plant biogeography. *American Journal of Botany*, 110, e16127.

van den Berg, C.P., Troscianko, J., Endler, J.A., Marshall, N.J. & Cheney, K.L. (2020). Quantitative Colour Pattern Analysis (QCPA): A comprehensive framework for the analysis of colour patterns in nature. *Methods in Ecology and Evolution*, 11, 316–332.

Ward, D.F. (2014). Understanding sampling and taxonomic biases recorded by citizen scientists. *J Insect Conserv*, 18, 753–756.

Weaver, W.N. & Smith, S.A. (2023). From leaves to labels: Building modular machine learning networks for rapid herbarium specimen analysis with LeafMachine2. *Applications in Plant Sciences*, 11, e11548.

Weller, H.I., Hiller, A.E., Lord, N.P. & Van Belleghem, S.M. (2024). recolorize: An R package for flexible colour segmentation of biological images. *Ecology Letters*, 27, e14378.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., *et al.* (2020). Transformers: State-of-the-Art Natural Language Processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, pp. 38–45.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., *et al.* (2022). DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection.