

Njobvu-AI: An open-source tool for collaborative image labeling and implementation of computer vision models

Ashwin Subramanian¹, Jonathan S. Koning¹, Cara L. Appel^{2,3}, Christopher M. Sullivan^{1,4}, Lisa Truong⁵, Robyn L. Tanguay⁵, Taal Levi², and Damon B. Lesmeister^{2,3}

¹ Center for Quantitative Life Sciences, Oregon State University, Corvallis, Oregon, USA ² Department of Fisheries, Wildlife, and Conservation Sciences, Oregon State University, Corvallis, Oregon, USA ³ USDA Forest Service, Pacific Northwest Research Station, Corvallis, Oregon, USA ⁴ College of Earth, Ocean, and Atmospheric Sciences, Oregon State University, Corvallis, Oregon, USA ⁵ Department of Environmental and Molecular Toxicology, Oregon State University, Corvallis, Oregon, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [↗](#)

Submitted: 12 March 2024

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

Summary

Practitioners interested in using computer vision models lack user-friendly and open-source software that combines features to label training data, allow multiple users, train new algorithms, review output, and implement new models. Labeling training data is critical for developing accurate object detection algorithms but is often not compatible with many cloud-based services due to limited internet bandwidth in many regions of the world. Desktop tools are useful in remote locations but lack capability to combine multiple projects. Furthermore, many tools offer features for labeling data or using pre-trained models for classification but are limited for creating and applying custom models. Free, open-source, and user-friendly software that offers a full suite of features is desirable to field researchers and conservationists that may have limited coding skills. We developed Njobvu-AI, a free, open-source tool that can be run on both desktop and server hardware using Node.js, allowing users to label data, combine projects for collaboration and review, train custom algorithms, and implement new computer vision models. The name Njobvu-AI (pronounced N-joh-voo AI), incorporating the Chichewa word for “elephant,” is inspired by a wildlife monitoring program in Malawi that was a primary impetus for the development of this tool. Code and documentation for this tool are available at <https://github.com/sullichrosu/Njobvu-AI/>.

Statement of need

Advancements in computer vision and machine learning techniques have led to increasing usage for applied research. Many groups who use images, video, and acoustic data for projects across disciplines are interested in computer vision and machine learning to improve data processing workflows or solve novel problems. Unfortunately, using computer vision and machine learning requires substantial preparation prior to extracting information from unlabeled images, video, and audio data. Researchers typically start by labeling a subset of data with classes of interest. Labeled data are used to train object-recognition models to identify objects in new images.

Currently there are many cloud-based and desktop tools to label training data, but they often have limits around file size, multiple users, and proprietary software. These restrictions limit broad application and offer few options for working outside the web environment. Additionally, few tools allow users to train a custom computer vision model while retaining control over the training parameters and data ownership. Training custom models is therefore limited to

42 experienced users of command-line programming or paid services. Here, we present Njobvu-AI
43 (pronounced N-joh-voo AI), a free open-source tool that can be used both offline and online.

44 Overview

45 Njobvu-AI is built on Node.js, a JavaScript runtime environment, to run in any web browser
46 and utilizes the templating language EJS to render everything on the server side to maintain
47 performance on the user's machine (Koning et al. 2023). All data are uploaded on the server
48 side, which uses SQLite3 for data organization. A global database stores user information and
49 project metadata. Each project has its own database that stores project information (i.e., labels
50 and photos) independent of the user. This compartmentalization allows for fast downloads,
51 easy uploads, and high performance because only smaller databases are accessed for labeling.

52 Njobvu-AI leverages open-source neural network frameworks to enable in-house computer vision
53 model training and image classification. The two frameworks utilized are Darknet (Redmon
54 2016) and TensorFlow (Abadi et al. 2015) with an emphasis on performance through accuracy
55 and speed using the YOLO algorithm (Bochkovskiy et al. 2020). New projects may be created
56 by uploading images and used to organize data by research topic for labeling. Images should
57 span the full scope of the project with all classes to be indexed. New classes and additional
58 training images can be added inside the project configuration page. On the labeling page,
59 a user selects the desired class, draws a bounding box and labels the image. Njobvu-AI
60 maintains proper spatial distances and box size when zooming in on large images. The ability
61 to zoom facilitates compatibility with large images, such as TIFF or LARGE TIFF images from
62 microscopes, and CAT (CT) scans.

63 We created a button to flag images for secondary review by other users. On the configuration
64 page users can customize settings for their needs. Njobvu-AI provides basic statistics about
65 the data for each project and an overview of project progress and balance of labels. On the
66 download page, users can zip all contents for importing to another installation of Njobvu-AI.
67 Users can also download the labeled dataset in TensorFlow, YOLO, COCO, or Pascal VOC
68 formats for training outside Njobvu-AI.

69 Once users have completed labeling their training dataset, they can train a model on the
70 TensorFlow or YOLO pages, which are two popular machine learning frameworks used in
71 computer vision. On the TensorFlow page, a user can add multiple Python paths to accom-
72 modate the user's relevant TensorFlow Python scripts. On the YOLO page, users can upload
73 a pre-trained weights file and begin training using their labeled images. Pre-trained models
74 may be downloaded and used for transfer learning, or users may wish to use previously trained
75 models.

76 Users can apply their trained model to locate and identify objects in new images. Njobvu-AI
77 will upload, process, and label the set of new images. Each image will contain bounding box(es)
78 around predicted objects. Users can review and modify the predicted labels if needed. Users
79 can navigate to validation mode to assess model performance on unlabeled datasets, correct
80 labels, view results, sort by class, bulk class changes, and individual photo statistics.

81 Njobvu-AI is built on Node.js, so users can easily bring the system online using any web server
82 hosting Node.js services. Njobvu-AI can be put into a cloud service and accessed from around
83 the world if needed. Since we created Njobvu-AI as a small, contained package system, users
84 can also install Node.js locally and run the tool on a desktop machine. We have successfully
85 run the system on Windows, Mac, and Linux operating systems as well as x86 and POWER.
86 Njobvu-AI is completely open source, allowing users to add custom formats and guarantee
87 privacy.

88 Acknowledgements

89 We thank Mazen Alotaibi, Thon Chao, and Pankaj Jaiswal for their contributions to the
90 development of this project. Funding was provided by USDA Forest Service Pacific Northwest
91 Research Station and the National Institutes of Health under Award P30 ES030287. The
92 findings and conclusions in this publication are those of the authors and should not be construed
93 to represent any official U.S. Department of Agriculture, National Institutes of Health, or U.S.
94 Government determination or policy. The use of trade or firm names in this publication is for
95 reader information and does not imply endorsement by the U.S. Government of any product or
96 service.

97 References

- 98 Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis,
99 J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, and R.
100 Jozefowicz. 2015. TensorFlow: Large-scale Machine Learning on Heterogeneous Systems.
101 <https://www.tensorflow.org/>.
- 102 Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao. 2020. YOLOv4: Optimal speed and accuracy
103 of object detections. [arXiv.org 2004.10934](https://arxiv.org/abs/2004.10934).
- 104 Koning, J. S., A. Subramanian, M. Alotaibi, C. L. Appel, C. M. Sullivan, T. Chao, L. Truong,
105 R. L. Tanguay, P. Jaiswal, T. Levi, and D. B. Lesmeister. 2023. Njobvu-AI: An open-source
106 tool for collaborative image labeling and implementation of computer vision models. [arXiv.org
107 2308.16435](https://arxiv.org/abs/2308.16435).
- 108 Redmon, J. 2016. Darknet: Open Source Neural Networks in C. <https://pjreddie.com/darknet/>.