BioScience, 2026, 76, biag074
https://doi.org/10.1093/biosci/biag074
Biologist's Toolbox

OXFORD

Manage your image database with computer vision—Lessons learned at Encyclopedia of Life

K. Wolcott [1,2,*], J. Hammock[1], K. Schulz[1] and T.M. Orrell[1]

[1]National Museum of Natural History, Smithsonian Institution, Washington, D.C., 20560, United States
[2]Florida Museum of Natural History, University of Florida, Gainesville, FL 32611, United States

*Corresponding author. E-mail: k.wolcott@ufl.edu Note: KW, JH and KS contributed as current or former indepent contractors to the Encyclopedia of Life project.
Katherine Wolcott (k.wolcott@ufl.edu) is a Postdoctoral Associate at the Florida Museum of Natural History in Gainesville and a former independent contractor for Encyclopedia of Life (EOL) at the National Museum of Natural History (NMNH) in Washington, D.C., both in the United States. She studies integrative biology approaches for natural history museum data using computer vision for 2D and 3D images. Jennifer Hammock and Katja Schulz are independent contractors for EOL. Jennifer Hammock is interested in data connectivity and scalable methods to increase discoverability of biodiversity information, particularly for little-known taxa. Katja Schulz is an independent biodiversity data scientist with experience in life science research, biodiversity data management, and the coordination of global scientific collaboratives. Thomas M. Orrell is a research biologist and the director for EOL at the NMNH. He is interested in developing a knowledge network that integrates taxonomic information, species traits, species occurrences, omics data, research findings, and literature to make global biodiversity data accessible and actionable for research.

Abstract

Natural history museums house invaluable biodiversity archives, but discoveries are often limited by the accessibility of collections data. As digitization of specimens and metadata accelerates, automated methods are needed to organize image databases and enable large-scale analyses. Beyond species identification, computer vision (CV) can automate image database management by standardizing gallery displays through intelligent cropping and extracting tags to improve search functionality. Encyclopedia of Life is a biodiversity database that aims to document all ∼2.2 million known species. Using EOL images and crowdsourced data, combined with pretrained and custom-trained models, we present 14 CV pipelines in three categories: Object Detection for Image Cropping, Object Detection for Image Tagging, and Classification for Image Tagging. Pipelines are open source, available on GitHub (https://github.com/EOL/computer-vision-with-EOL-images), and documented for nonexperts. We share models, datasets, and methodological insights demonstrating how CV can unlock novel information from over 1.5 million images.

Keywords **computer vision, database management, biodiversity informatics, machine learning, natural history collections**

Natural history museums document Earth's biodiversity across millions of years, yet their scientific potential is limited by the accessibility of collections and specimen data (Hedrick et al. 2020, Miralles et al. 2020, Hardisty et al. 2020, Lendemer et al. 2020). Increasing digitization efforts have made hundreds of millions of specimen images and metadata available online, but transforming heterogeneous data into searchable resources for downstream research and analysis remains a major challenge. As biodiversity research moves toward global scale synthesis of trends through climatic periods and evolutionary bottlenecks, approaches to streamline data organization and extraction are needed.

Computer vision (CV) is increasingly used in biodiversity research, especially to automate tasks like identifying some combination of species, individuals, and/or their respective traits from field and collection images (Gaston and O'Neill 2004, Goëau et al. 2017, Joly et al 2017, van Horn et al 2018, Wäldchen et al. 2018, Spiesman et al. 2021, Blair et al. 2022, Patton et al. 2023, Poveda-Cuellar et al. 2025, Xie et al. 2025; reviewed in Lürig et al. 2021, Borowiec et al. 2022, Groom et al. 2023, East et al. 2025). Analyses that combine metadata with specimen traits to study global patterns are in their early stages, but show exciting results (Weaver and Smith 2023, Folk et al. 2024, reviewed in Lürig et al. 2021, East et al. 2025, Guo et al. 2025). Using their computer vision pipeline (Mothra) to study over 17,000 digitized butterfly specimens, Wilson and colleagues (2022) showed that caterpillars exposed to higher temperatures became larger adults. Studies of herbarium sheets also showed climate change induced phenological shifts in flower and fruiting times (Lorieul et al. 2019, Pearson et al. 2020, Kommineni et al. 2021, Park et al. 2023, Ordoñez et al. 2025). In contrast to research-based approaches, CV's use for image database management (for tasks such as standardizing displays, enriching metadata, and im-

Received: 8 April 2025. **Revised:** 1 April 2026. **Accepted:** 7 April 2026
© Smithsonian Institution 2026. Published by Oxford University Press on behalf of the American Institute of Biological Sciences. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact reprints@oup.com for reprints and translation rights for reprints. All other permissions can be obtained through our RightsLink service via the Permissions link on the article page on our site-for further information please contact journals.permissions@oup.com

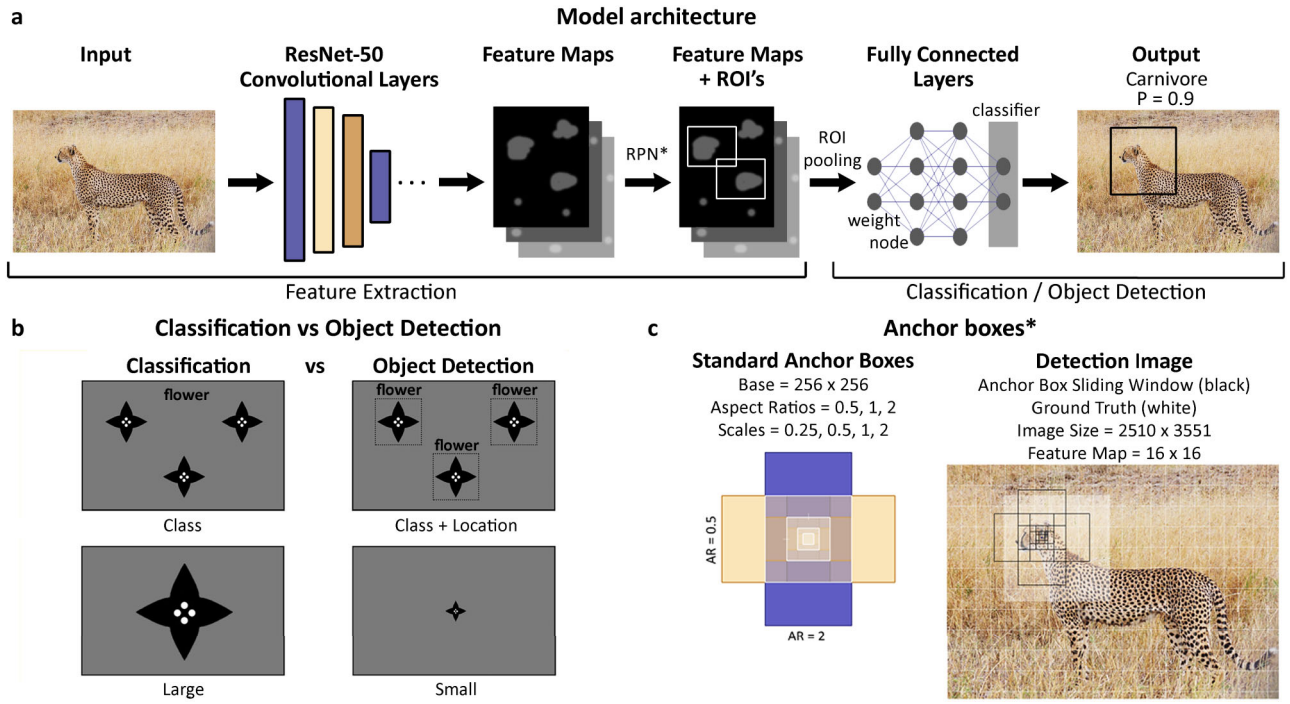


Figure 1. Introduction to concepts of CV using CNNs. (a) Sample object detection model built on a Faster-RCNN meta-architecture (Ren et al. 2017) with a ResNet-50 feature extractor (He et al. 2015; see details in box 1). (b) The different outputs generated by classification (image class) versus object detection (class and location of objects in image) are shown in the top row and appropriate relative subject sizes for each task are shown below. (c) Illustration of default anchor box dimensions used in the Region Proposal Network (see RPN∗ in **a**) step of Faster-RCNN. Anchor boxes are used as a sliding window to make "objectness" proposals for each image region within the boxes before final bounding boxes are determined.

proving discoverability) has been relatively limited. These tasks are essential for improving the experience of users and facilitating computational analysis, yet they are often performed manually, or not at all.

Several tools exist that use CV to extract text data from labels of digitized plant and insect specimens (known as optical character recognition; Barber et al. 2013, Drinkwater et al. 2014, Kirchhoff et al. 2018, Weaver et al. 2023, Stenhouse and Thrall 2025, Turnbull et al. 2025; reviewed in Tessler and Little 2025). CV can also be used to organize images into categories that help researchers work with only their desired data types, yet only few published approaches describe such methods. For example, Ávila and colleagues (2025) trained a classifier, Herbariograph, to identify botanical collection images as pressed specimen, live plant, illustration, transmission light micrograph, and other categories. For broad analysis of botanical images, such categories are a prerequisite to any subsequent questions about phenology, surface structures, or other phenotypic changes.

Biodiversity image databases like the Encyclopedia of Life [EOL 2026], along with iNaturalist [iNaturalist 2026], and iDigBio [iDigBio 2026] provide a wealth of opportunities for developing CV tools. EOL is an online biodiversity resource that seeks to provide information about all ~2.2 million species known to science, for users including scientists, policy makers, educators, formal and informal learners, and the public (Parr et al. 2014). Content is derived from sources that are creative commons licensed or in the public domain, is community curated, and is available for download through the EOL API [Application Programming Interface; EOL API 2026]. For the latest version of EOL (v3), our goal was to use CV to manage our older, less structured image content to improve the accessibility of our images and facilitate cutting-edge biodiversity research.

Here, we present a set of CV workflows developed to manage a large biodiversity image database, using EOL as a case study. Rather than focusing on species identification or trait extraction, we emphasize practical applications of CV for database functionality and extract general lessons applicable to museums, data aggregators, and biodiversity informatics initiatives. These tasks focus on image-wide properties that improve database usability and facilitate downstream analysis.

We highlight three challenges faced by biodiversity database managers, which are well-suited to CV-based approaches:

1) Standardizing image displays, especially generating consistent thumbnails across heterogeneous images.
2) Enhancing image discoverability through image tagging, including the presence of organisms, life stages, or reproductive structures not annotated in existing metadata.
3) Categorizing image types, such as distinguishing specimen photographs from field images, illustrations, or maps and indicating image quality.

Applying CV to EOL images

EOL aggregates images from diverse sources, including standardized museum digitized collection images, user-uploaded field images, historical illustrations, scientific figures, satellite imagery, camera trap images, and more. Although taxonomic identifications are always available for our images, they lack standardized crops, quality indicators, and descriptive tags that would allow users to search through our data more effectively. To address these gaps, we present 14 pipelines

built to test applications of CV to EOL images. The CV approaches we use are based on either image classification or object detection (see details in box 1). In image classification, the entire image is inspected, and a class label is assigned (figure 1b). In object detection, object class and location are determined, resulting in an image with labeled boxes around objects of interest (figure 1b).

Box 1. A brief primer on CV approaches.

We briefly describe the basics of CV to allow the reader to understand our choices of models and hyperparameters and to develop their own pipelines. Convolutional neural networks (CNNs) are a type of deep learning algorithm inspired by the connectivity of visual cortex neurons (Fukushima 1980, LeCun et al. 1999). They are universal function approximators, which are able to recognize complex patterns in multidimensional datasets when they are configured correctly and provided with sufficient training examples (Hornik et al. 1989, LeCun and Bengio 1995, LeCun et al. 2015). CNNs consist of single stage meta-architectures, like Single Shot Multibox Detector (SSD, Liu et al. 2016), YOLO (v3 in Redmon and Farhadi 2018), or two-stage methods, like Faster Region-based CNN (Faster RCNN; Ren et al. 2017, reviewed in Agarwal et al. 2018). Meta-architectures are combined with feature extractors, like DarkNet-19 (Redmon and Farhadi 2017), ResNet-50 (He et al. 2015), and Inception-v2 (Szegedy et al. 2016) to form CV frameworks of varying speed and accuracy. Benchmark datasets, like ImageNet, provide a universal standard to compare model performance (Krizhevsky et al. 2012).

Images are fed into CNNs and transformed through a series of filters (or kernels, eg: resizing, pooling) for different convolutional layers of the network, known as feature extraction (figure 1a). Each layer applies an activation function (like ReLU, Softmax, Sigmoid), which maps a probability to the filtered output (known as a feature map) before passing it to the next layer (Nwankpa et al. 2018). Features are transformed by fully connected layers into the final model predictions for object class and/or location. In image classification, the entire image is inspected, and a class label is assigned (figure 1b). In object detection, object class and location are determined, resulting in an image with labeled boxes around objects of interest (figure 1b). During object detection, there are region proposal, feature extraction, and classification steps before the final labeled bounding boxes are produced (figure 1a). During region proposal, the image is divided into sub-images which are scanned using a sliding window (anchor boxes) to find regions likely to contain objects (figure 1c). When multiple Regions of Interest (ROIs) overlap above a defined Intersection over Union value (IoU), the objects are pooled, located, and classified. With image classification, there is no region proposal step, so feature extraction and classification are carried out on the entire image.

During training, optimization algorithms (eg: Adam, gradient descent) help the model minimize its loss functions (like cross-entropy and mean squared error) to achieve the best accuracy (Kingma and Ba 2017, Dogo et al. 2018). Activation functions, optimization algorithms, and loss functions are all model hyperparameters that can be fine-tuned between training runs to optimize prediction accuracy. Learning rate (how fast a model learns), batch size (how many images to study at a time), and dropout (how much a model forgets during each iteration over the dataset, known as an epoch) can also be adjusted. Training is an iterative process and often includes many training runs where parameters are fine-tuned and the training data is curated until the desired behavior is achieved. If a model learns patterns for a particular dataset, their ability to make accurate predictions for novel data is not guaranteed, so careful inspection of results is necessary (Geirhos et al. 2020, Greener et al. 2022).

Our pipelines fit into three classes—Object Detection for Image Cropping (figure 2a), Object Detection for Image Tagging (figure 2b), and Classification for Image Tagging (figure 2c; table 1) [Computer Vision with EOL Images GitHub 2026]. They are open source, built in python, run in the cloud using Jupyter Notebooks in Google Colaboratory, hereafter referred to as Colab Notebooks, and use two major CV frameworks—Tensorflow Abadi et al. 2015, Huang et al. 2017) [TensorFlow Object Detection API GitHub 2026] and YOLO (Redmon et al. 2016, Redmon and Farhadi 2017, Redmon and Farhadi 2018, Bochkovskiy et al. 2020, Wang et al. 2022) [(Darkflow GitHub 2026)].

We provide documentation and designs made for new users with limited background knowledge. All major parameters are adjustable through form fields. All code can be run as demos using a sandbox environment in the Google Colab runtime without needing to connect to a local filesystem. For extending our pipelines, we describe how to repurpose crowdsourced datasets to train models. In cases where models perform differently than expected, we also provide methods to preprocess, analyze, and postprocess results.

Methods overview: building and evaluating CV pipelines

Here, we discuss the five main steps of building a CV pipeline to manage an image database, with specifics of how we applied them to EOL images (figure 3). To accelerate model training using high-powered virtual GPUs (Nvidia H100s, G4s, T4s, L4s, A100s, and various TPUs) and store our training data in a shared cloud drive, we purchased a Google Colab Pro subscription ($9.99/month, as of Mar 2026) and 2 TB of Google Drive storage [$99.99/year, as of Mar 2026; Google Colab Pro Signup].

Select model(s)

Before building a CV pipeline, existing models that can be used out-of-the-box or with slight modifications for the desired task were searched for and evaluated. Known as "model zoos," there are several websites that contain pretrained models with code examples that can be used for specific tasks and datasets [see for examples, TensorFlow Hub, PyTorch Hub, Hugging Face Models, Kaggle 2026].

Assemble training data

Models need to be trained on a large, diverse set of images, known as the training dataset (figure 3-2). The training dataset should encompass the full range of variation within your database (a combination of extreme and "average" examples) and needs to be annotated either into classes for image classification (indicating what type of image they are) or with bounding boxes for object detection (indicating what and where objects of interest are contained within the image; figure 1b).

Typically, the larger and more diverse the training dataset, the more robust and accurate the resulting model will be (He et al. 2015, LeCun et al. 2015). There are many general image datasets suited for training machine learning models, which can be searched to see if

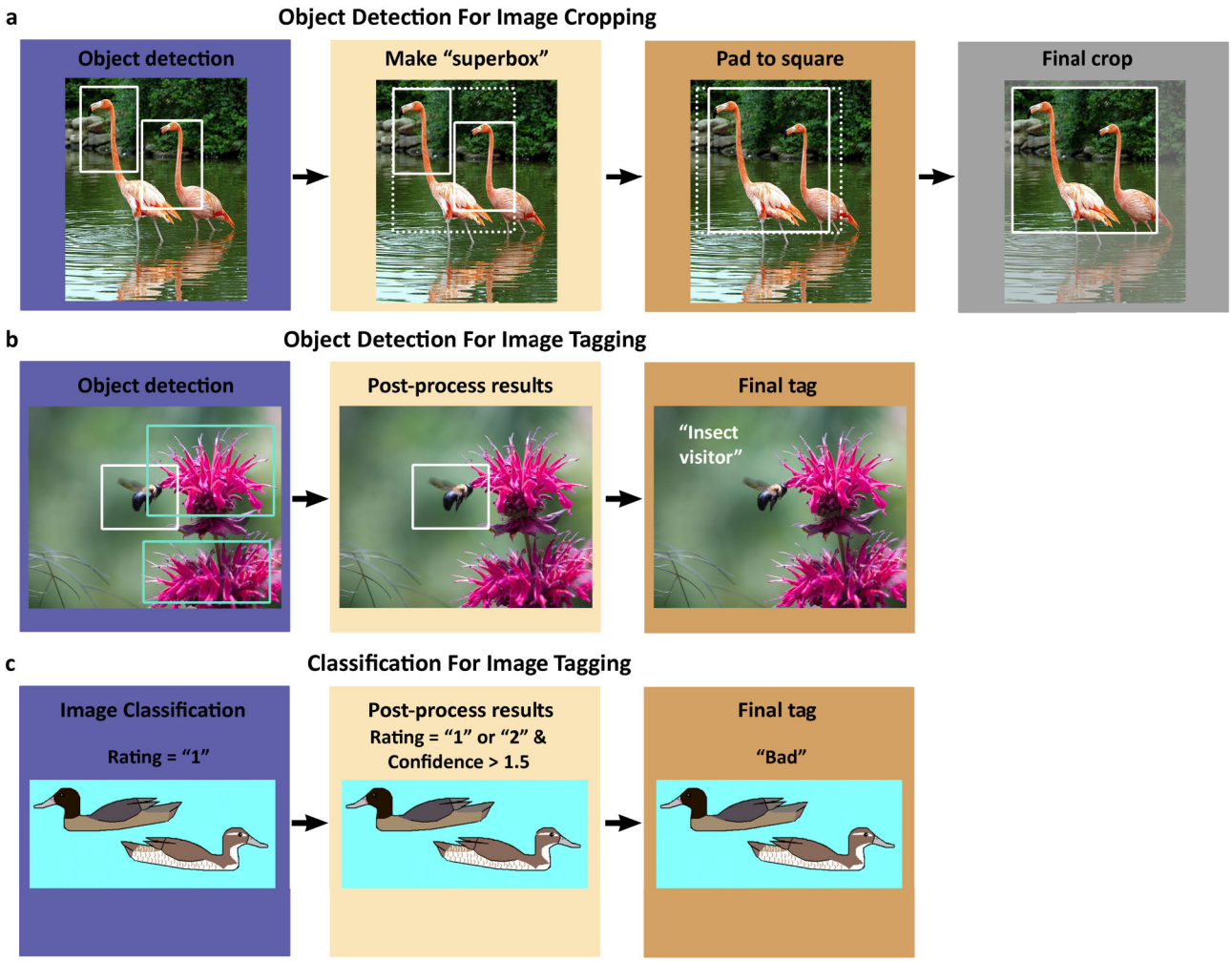


Figure 2. Pipeline examples for Object Detection for Image Cropping (a), Object Detection for Image Tagging (b), and Classification for Image Tagging (c). (a) For Object Detection for Image Cropping, after object detection, all detection boxes are merged into a "superbox," before being padded to square, and producing the final cropping coordinates. (b) For Object Detection for Image Tagging, model predictions are postprocessed for desired results (in this case, an insect present in an Angiosperm image), and converted into a final tag, "Insect visitor." (c) For classification for image tagging, model predictions are postprocessed (in this case, ratings of "1" or "2" with confidence $>$ 1.5 are kept), and converted into a final tag, "Bad."

any images match the desired task [see for examples, Kaggle, Google Open Images, Image-Net 2026], but often biological tasks (like identifying diverse plant reproductive structures of varying shapes, sizes, and colors) require highly curated datasets with some degree of expert review. Recently, some curated, large datasets of museum specimens have been published for use in training machine learning models, although such approaches are in their early stages (Ávila et al. 2025, Barkmann et al. 2025, Stevens et al. 2024). The Tree of Life Dataset is the largest machine learning ready dataset of the photos and taxonomic positions of biological organisms currently available, containing 10 million images for 450,000 taxa (Stevens et al. 2024). Barkmann provides 540,000 images of 185 Austrian butterfly and moth species (2025), and Ávila provides 86,000 botanical images from 43 institutions labeled into 7 categories.

Until more biological datasets curated for machine learning applications are published, data annotation and cleaning remain one of the main bottlenecks for training new CV models. In these cases, crowdsourcing is an invaluable way to incorporate the community into scientific research and create large training datasets, which would not be possible or practical for experts to accomplish alone. Such approaches have been successfully used via mobile phone apps and interactive web interfaces at EOL, as well as Zooniverse, Biodiversity Heritage Library (BHL), and DigiVol, amongst others (Ahn et al. 2012, Simpson et al. 2014, Osawa et al. 2017, Mika et al. 2017, (Alony et al. 2020, Azmy et al. 2024, Stewart et al. 2025).

If available, existing EOL crowdsourced datasets were repurposed to train models (Table S1). Crowdsourced datasets were collected via tools in the EOL web interface between 2009 and 2018. They were originally built to manually extract information to improve our gallery displays by adding tags for image quality and best square thumbnail coordinates. For this project, we decided to repurpose them to scale our approaches from hundreds of thousands to millions of images. When crowdsourced datasets were not available, training datasets were manually built for desired tasks.

Training datasets had 600 to 7000 images per class, depending on the task and number of images available (Table S1). Imbalance was addressed by randomly applying data augmentation to increase the number of images for all classes, followed by randomly undersam-

Table 1. Summary of EOL computer vision approaches for our three main types of pipelines—Object Detection for Image Cropping, Object Detection for Image Tagging, and Classification for Image Tagging.

Task	Type	Target group(s)	Target output	Achieved output	Train images/class	Train image source(s)	Images processed to-date
Crops	Object detection	Aves	Square, centered cropping coordinates	Square, centered crop around animal(s)	NA	NA	405,087
		Chiroptera	Square, centered cropping coordinates	Square, centered crop around animal(s)	600	EOL	17,556
		Lepidoptera	Square, centered cropping coordinates	Square, centered crop around animal(s)	600	EOL	614,779
		Anura, Carnivora, Coleoptera, Squamata	Square, centered cropping coordinates	Square, centered crop around animal(s)	600	EOL	387,959
Tags	Object detection	Insecta	Insect life stage tags	Tag of "adult" or "juvenile" present	NA	NA	20,000
		Hexapoda, Angiospermae	Insect visitor tags	Tag of "insect visitor" in angiosperm images	NA	NA	20,000
		Aves, Amphibia, Reptilia, Mammalia	Scat/footprint present tags	No meaningful output generated	800	EOL, iNaturalist	–
		Angiospermae	Flower/fruit present tags	Tag of "reproductive structures present"	NA	NA	20,000
Tags	Classification	Any/All	Image quality rating tags	Tag of "bad" quality images	7000	EOL	20,000
		Angiospermae, Hexapoda	Flower present tags	No meaningful output generated	600	PlantCLEF 2016	–
		Angiospermae	Flower/fruit present tags	No meaningful output generated	850	EOL	–
		Any/All	Image type tags	Tags of "herbarium sheet," "non-photographic image"	2000	EOL, Wikimedia Commons, Flickr BHL	20,000

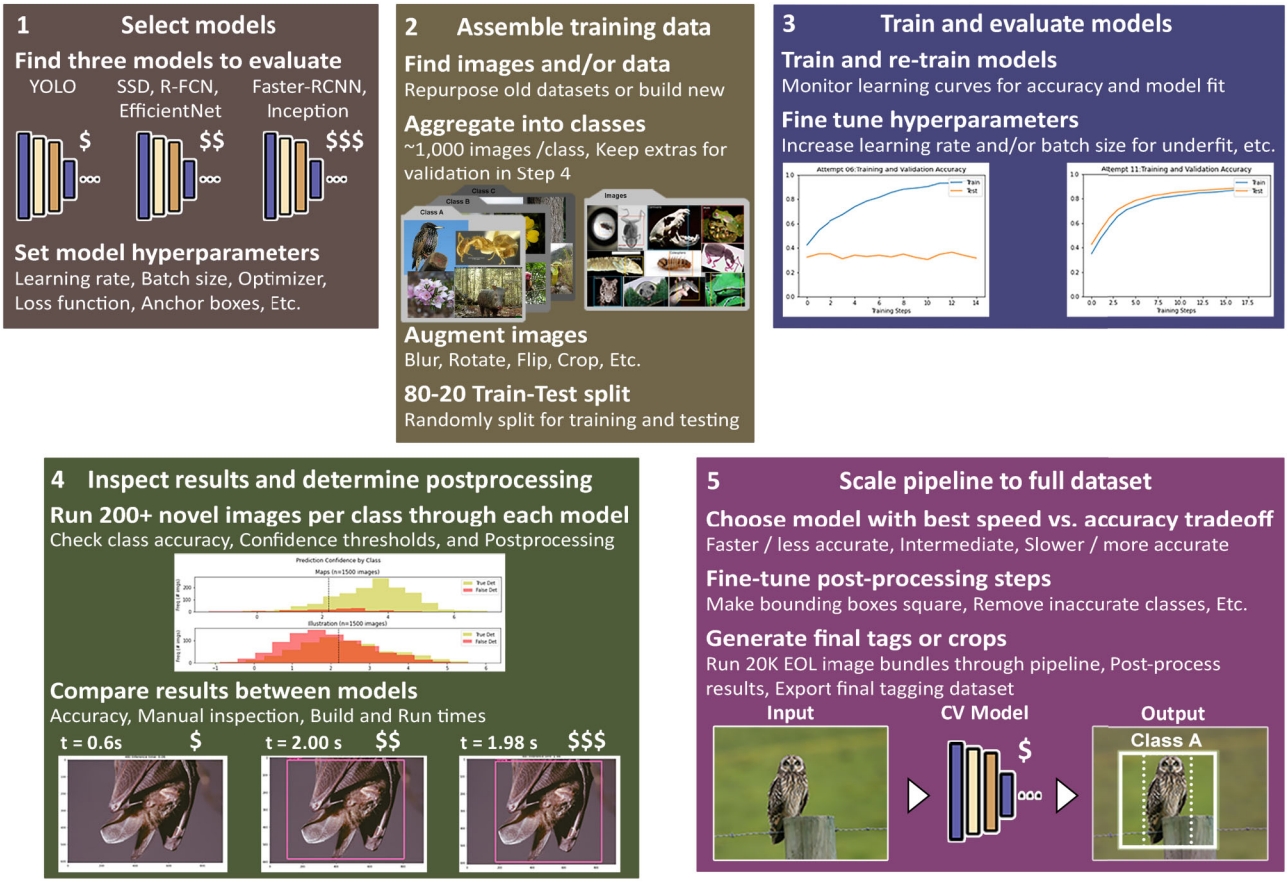


Figure 3. Building and evaluating an EOL computer vision pipeline. (1) Choose three models to evaluate for a given task—a lightweight model with lower accuracy, an intermediate model, and a larger model with higher accuracy. (2) Assemble training data, group into classes, do image augmentation, and split 80–20 into Train–Test. (3) Train models and evaluate performance using learning curves. Adjust hyperparameters and retrain until the highest accuracy without overfitting is achieved. (4) Run novel images through each trained model and compare real-world performance, along with build and run times. (5) Select the model with the best tradeoff between speed and accuracy. Determine post-processing steps (confidence thresholding, etc.) and generate final tags or crops.

pling, leaving all classes with the same number of images before training (Buda et al. 2018). Imgaug [Imgaug GitHub 2026], was used to augment images with filters to randomly crop (by 1–16 pixels), rotate (−25–25°), blur (Gaussian blur, $\sigma = 0$–3), and adjust the color of images (change hue and saturation values by −50–50). At the end of preprocessing, classes were split 80–20 into train and test, where 80% of data was used to train the model and the remaining 20% was used to evaluate performance across epochs. When sufficient data was available, we also prepared a separate validation dataset, which was not used in training or testing, to evaluate model performance on completely novel images.

Train and evaluate model

Two to three models (consisting of different meta-architectures and feature extractors) were evaluated for each CV task before selecting the model with the best tradeoff between speed and accuracy (figure 3-3). Models were selected based on performance tables in Huang and colleagues (2017) and the Tensorflow Object Detection API [TensorFlow Object Detection API GitHub 2026]. Pretrained object detection models were downloaded from the Tensorflow Object Detection Model Zoo [TensorFlow 1 Detection Model Zoo 2026], the darkflow GitHub Repository, which implements YOLOv2 DarkNet-19 in Tensorflow [Darkflow GitHub 2026], and from the darknet GitHub, which provides YOLOv3 pretrained on Google Open Images v4 (Darknet GitHub 2026; Redmon and Farhadi 2018). Some models were used directly for out-of-the-box inference without any retraining, whereas others were custom built and trained.

When not using out-of-the-box options, models were either fully custom trained or trained with transfer learning, where lower layers of the model that extract general image features like shapes are frozen, whereas outer classification layers are retrained (figure 1a). Training progress was monitored using learning curves to ensure that the model fit training data well, while also generalizing to the test data (figure 3-3; figure 4c). If the model was overfitting and learning the train data without generalizing to the test data (figure 4a) or underfitting and not learning the data well (figure 4b), model hyperparameters (e.g., learning rate, dropout, batch size) were adjusted before retraining (figure 4d). Hyperparameters for the models used to generate tags and crops are summarized in table S2. In our model training notebooks, learning curves are automatically created and saved during iterative training attempts. Trained models are available for download on the EOL Model Zoo on Kaggle [EOL Model Zoo 2026].

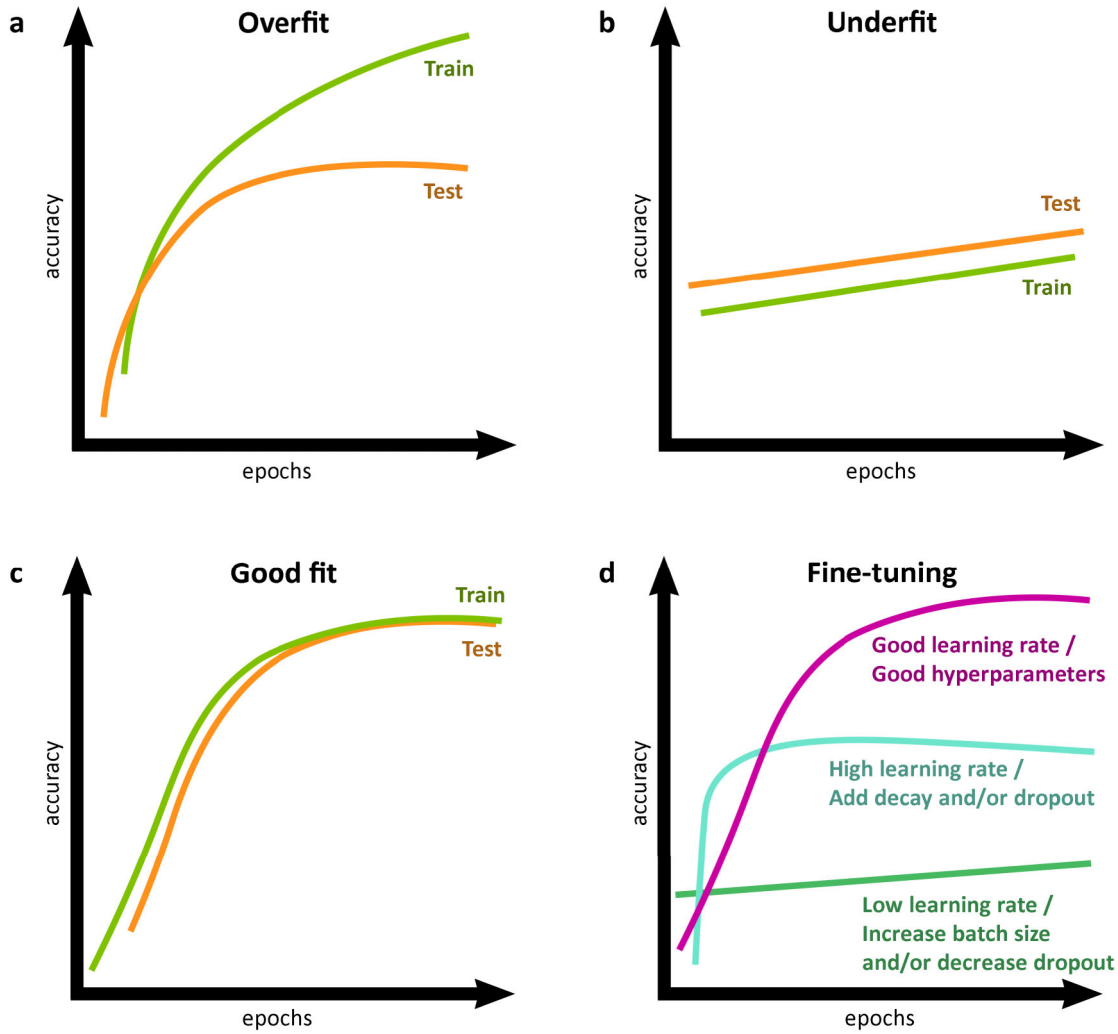


Figure 4. Guide to monitor model training using learning curves. During training, accuracy is measured across epochs for train and test images. Monitoring the progress allows you to evaluate if the model is (a) overfitting to the train data and unable to generalize to the test data, (b) underfitting and not learning the train or test data well, or (c) achieving a good fit, learning train and test data well. (d) Example training patterns and possible hyperparameters to fine-tune, including learning rate decay, batch size, and dropout.

Inspect results and determine postprocessing

To evaluate object detection model performance, inference time, mean Average Precision (mAP), and average recall (AR) were used to compare model performance on the dataset (Everingham et al. 2010, Liu et al. 2016, Ren et al. 2017, Redmon and Farhadi 2018). Inference time measures how long the model takes to make a prediction for an image. mAP and AR measure bounding box prediction accuracy. They were measured at 50% IoU, where the overlap area of the detected box and the ground truth was at least 50% of the total area of both boxes (figure 5a). Precision is a measure of prediction accuracy based on positive results (for model predictions for class "A," how many were actually class "A"), whereas recall is based on negative results (for all class "A" in a dataset, how many did the model predict or miss; figure 5b). Tensorflow Faster-RCNN and SSD models automatically calculate mAP and AR, while for YOLO models, they were calculated using the mAP library [mAP GitHub 2026]. To evaluate image classification model performance, precision, recall, and F1 score (the harmonic mean of precision and recall), were used (figure 5b, 5c). We also inspected results by generating model predictions for novel images not used during training or testing, known as validation images. To achieve the most of model outputs, we used our own custom-built evaluation tools (similar to methods by Wexler et al. 2020) to inspect prediction accuracy per class, at various confidence thresholds, and under different postprocessing scenarios (figure 3-4).

Scale pipeline to full dataset

After selecting the best model and postprocessing steps (confidence thresholding, only keeping certain class predictions, etc.), the pipeline for generating tags and crops was finalized (figure 2; figure 3-5). After generating crops and/or tags for initial bundles of 20,000 images, pipelines were scaled up to all applicable EOL images. Output cropping and tagging datasets are available on Zenodo [EOL CV Datasets on Zenodo 2026].

Pipelines: object detection for image cropping

Square thumbnails are commonly requested by EOL API users, and they improve image galleries by standardizing display of heterogeneous images. To generate square, centered crops, we used a crowd-

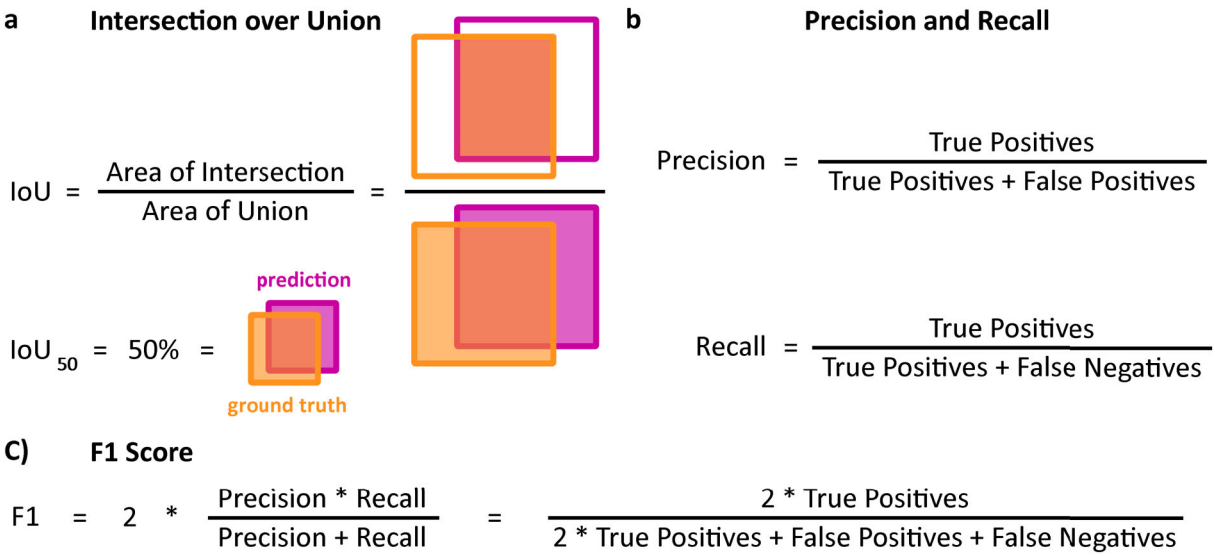


Figure 5. Evaluation metrics used to gauge model performance. (a) IoU measures the overlap area of a predicted bounding box and its ground truth divided by their area of union. Fifty percent IoU is a common benchmark for model performance. (b) Precision is a measure of prediction accuracy (for model predictions for class "A," how many were actually class "A"), whereas recall is a measure of what the model did not find (for all class "A" in a dataset, how many did the model predict or miss). (c) F1 score is the harmonic mean of precision and recall.

sourced dataset of EOL user-determined "best" thumbnail crops to train models (table 1). Between 2009 and 2018, through tools available in the EOL web interface, 19,357 images were cropped 19,500 times (table S1; Ahn et al. 2012). Unlike typical object detection training data (a box around each instance of an object), our cropping data contains one box per image, which can include an animal's head, an entire individual, or many individuals (figure S1). Taxonomic groups were selected based on data availability (taxa with crops for > 500 images, which became datasets of > 600 images per class after augmentation). Repurposing crowdsourced data saved time and incorporated the EOL user community into building our newest release. Our object detection pipelines use coarse taxonomic levels to identify and locate animals within an image, then use taxon-specific postprocessing steps (like adding padding to ensure bird beaks are not cropped out) to generate crops (figure 2a).

Since these were the first CV pipelines developed with EOL images, we were not sure how object detection models would perform when trained using EOL user cropping data instead of traditional training data. To simplify interpreting model training and predictions, we first built models for one taxon at a time. Single taxon models were developed to detect birds (Aves), bats (Chiroptera), or butterflies and moths (Lepidoptera). A combination of fully pretrained models for out-of-the-box inference and custom training were used (table 1). For each single taxon model, three object detectors were evaluated—YOLOv2 with a DarkNet-19 feature extractor (Redmon and Farhadi 2017), SSD with MobileNetv2 (Liu et al. 2016) and Faster-RCNN with ResNet-50 (Ren et al. 2017). The three frameworks differ in speed and accuracy. YOLOv2 was found to be the fastest but least accurate, whereas Faster-RCNN ResNet-50 was found to be the slowest but most accurate, with SSD MobileNetv2 falling somewhere in between (Huang et al. 2017, Lin et al. 2017, Redmon and Farhadi 2018). Faster-RCNN had the best tradeoff between speed and accuracy for all target groups (table 2; table S2). After developing single taxon models, a multitaxon model was trained to simultaneously detect snakes and lizards (Squamata), beetles (Coleoptera), frogs and toads (Anura), and carnivorous mammals (Carnivora). Three object detectors were used—YOLOv2 with DarkNet-19 (Redmon and Farhadi 2017), Faster-RCNN with Inception-v2 (Liu et al. 2016) and Faster-RCNN with ResNet-50 (Ren et al. 2017). Faster-RCNN with Inception-v2 was selected (table 2; table S2).

Since our goal was to emulate human choice of the "best" square crop for an image, our ground truths are subjective (and depend on user-determined "best" square coordinates) compared to typical object detection annotations (that indicate only the location and label of objects; figure S1). As such, our model predictions were also human validated for at least 50 images of each taxon following approaches in Suh and colleagues (2003) and Joly and colleagues (2017).

To evaluate model performance during training, we compared mAP for model predicted bounding boxes versus user generated square crops. To evaluate real world performance on novel images, we compared postprocessed bounding boxes to user generated ground truth crops using mean Intersection over Union (mIoU; figure 5a, 5b; for model predictions versus ground truths displayed on images, see [Ground truths versus model predictions 2026]). At the end of training our models achieved mAP values of 87% (Chiroptera), 96% (Lepidoptera), and 82% (Multitaxon) for object detection bounding boxes versus user generated square crops (table 2). After running novel validation images through and postprocessing model predictions, we achieved mIoU values of 71% (Aves), 68% (Chiroptera), 67% (Lepidoptera), 60% (Multitaxon). Although these metrics are an imperfect measure of model performance because the "best" crop is subjective, they indicate model consistency for novel data.

Pipelines: object detection for image tagging

Image tags can contain information that is not noted in metadata (like plant phenology or image quality) and increase the discoverability and utility of images stored in biodiversity databases. We used object detection to identify objects within specific classes of images (like an insect present in an Angiosperm/flowering plant image), then used

postprocessing steps to achieve the desired tags (like converting the tag of "insect" to "insect visitor"; figure 2b; table 1). Since we did not have any EOL crowdsourced data available for tagging tasks, we reviewed EOL user-requested tags, available and relevant datasets for training, and available pretrained models before determining our tagging category goals—insect life stages, plant-insect co-occurrence, and scat or footprints present.

YOLOv3 pretrained on Google Open Images v4 was used for insect life stage and plant-insect co-occurrence tags (table 1). Google Open Images is a CV benchmark dataset that contains 9.2 million images with 600 classes, some of which are everyday objects, whereas others are taxonomic groups (like Magpie, Beetle, Lynx, Carnivore, etc.; Redmon and Farhadi 2018, Kuznetsova et al. 2020). To add scat or footprint tags, custom training datasets were prepared for transfer learning with YOLOv2 and YOLOv4, but no successful outputs were obtained (Redmon and Farhadi 2018, Bochkovskiy et al. 2020; table 1; table S1).

For insect life stages, upon inspection of available dataset classes in Google Open Images, we noticed that images labeled as "Ant," "Bee," "Beetle," "Butterfly," "Dragonfly," "Insect," "Invertebrate," and "Moths and butterflies" all contained adult invertebrates. It was also observed that images labeled as "Caterpillar," "Centipede," and "Worm" contained the correct classes along with a number of misidentified juvenile invertebrates. We converted model predictions for the former classes to "adult" and the latter classes to "juvenile." Although biologically imprecise, applying a pretrained model and tailored postprocessing steps quickly added useful tags to EOL invertebrate images. Manually curating a dataset of the diverse forms of insect life stages would have taken significant time investment without guaranteed success.

For plant-insect co-occurrence, another postprocessing approach was used with YOLOv3 pretrained on Google Open Images v4 (table 1). Model predictions for "Butterfly," "Insect," "Beetle," "Ant," "Bat (Animal)," "Bird," "Bee," and "Invertebrate" were converted to "insect visitor" when running inference on EOL Angiosperm images. Although we could have chosen a variety of custom training approaches (for e.g., manually identifying flowers and pollinators present within images), we were able to leverage existing frameworks instead. By prioritizing true positives, we ensured that predicted tags contained the desired subject (a tag for "insect visitor" contains an insect), with the tradeoff that some boundary case tags may have been missed as false negatives (some images with an insect present were not tagged as such).

We did not produce summary statistics for Object Detection for Image Tagging pipelines because pretrained models were used and we did not have validation datasets available (figure 2b). After initial builds with pretrained models available for YOLOv3, during production and development phases, we also upgraded pipelines to run using YOLOv8 pretrained on Google Open Images v7 (Reis et al. 2024).

Pipelines: classification for image tagging

Classification pipelines were developed using custom-trained models to generate image tags (figure 2c; table 1). The image quality rating pipeline repurposed an EOL crowdsourced rating dataset, but all other pipelines required assembly of new training datasets. Between 2009 and 2018, through tools available in the EOL web interface, 199,153 images received 2591,501 user ratings and 16,856 images were selected as exemplar in 237,325 actions (table S1; Ahn et al. 2012). Image classification training requires images sorted into

Table 2. Evaluation metrics for final Object Detection for Image Cropping pipelines.

Target group(s)	Target output	No. Classes	Train mAP	Validation mIou	No. Images	Postprocessing	Predictions retained	Model	Model citation
Aves	Square, centered cropping coordinates	1	NA	0.71	180	IoU > 50	137	Faster-RCNN	Ren and colleagues (2017)
Chiroptera	Square, centered cropping coordinates	1	0.87	0.68	53	IoU > 50	24	Faster-RCNN	Ren and colleagues (2017)
Lepidoptera	Square, centered cropping coordinates	1	0.96	0.67	106	IoU > 50	13	Faster-RCNN	Ren and colleagues (2017)
Anura, Carnivora, Coleoptera, Squamata	Square, centered cropping coordinates	4	0.82	0.60	300	IoU > 50	35	Faster-RCNN Inception v2	Ren and colleagues (2017)

Train mAP at IoU = 50% is measured at the end of model training for test images. Validation mean IoU (mIoU) represents real world performance and is measured for novel validation images, which were not seen by the model during training or testing. It is based on bounding boxes after postprocessing to generate one square "superbox" per image (illustrated in figure 2a). Predictions Retained are the number of images retained after filtering predictions by a postprocessing threshold of IoU > 50%. Although Aves models were not custom-trained, we provide evaluation statistics using EOL user generated crops as ground truths. Statistics were generated using "inspect_train_results.ipynb" for each respective target group and are available on our GitHub [Computer Vision with EOL Images GitHub 2026]. See figure 5 for descriptions and illustrations of metrics. Model predictions versus ground truths displayed on images are available on Zenodo [Ground truths versus model predictions 2026].

folders for each desired class (figure S2). We developed classification pipelines that add tags to an image, then use postprocessing steps (like converting a tag for rating = "1" to "bad") to achieve the desired tag outputs (figure 2c; table 1).

For image quality ratings, which improve user options for EOL gallery sorting, an existing EOL crowdsourced rating dataset was used to train models (table 1; table S1). Users assigned over 200,000 image ratings from 1 (bad) to 5 (good), allowing us to implement the largest training dataset of our CV pipelines ($n = 7000$ images per class; figure S2). Preprocessing 35,000 images required additional quality control steps, as bulk image downloads resulted in truncated image files, non-image file downloads (mp3, mp4, etc.), and special characters in filenames. Since manual inspection was not practical for such a large dataset, image files were programmatically validated (detailed steps are available in rating_preprocessing.ipynb [Computer Vision with EOL Images GitHub 2026]).

After initial training runs with image classes 1 (bad)–5 (good), the best obtained training or validation accuracy was 30%. Since prediction accuracy was only slightly better than random chance, we suspected that the subjective nature of crowdsourced ratings may have produced inconsistent image classes. We aggregated rating classes 1 and 2 into "bad" and classes 4 and 5 into "good" and then retrained the models with improved results (60% validation accuracy for Inception v3, 44% for MobileNetSSD v2). Upon further inspection, we realized that EOL users were divided on what defines a "good" image and unified on what makes a "bad" one, with trained models reflecting this inconsistency. Only "bad" predictions with confidence > 1.5 were kept for image tags, resulting in a final prediction accuracy of 82% (table 3; figure S3).

To add tags for flowers present within images, we used PlantCLEF 2016, which was created for a species identification competition and contains images and metadata of European plants (table 1; table S1; Goëau et al. 2017). We used metadata on plant habit and phenology to group images into classes and randomly selected 600 images per class for training (flower, fruit, stem, branch, entire, leaf). Transfer learning was done with MobileNetSSD v2 (Sandler et al. 2018) and Inception v3 (Szegedy et al. 2016) models pretrained on ImageNet 2012 (Russakovsky et al. 2015). Models did not learn well or achieve prediction accuracy better than random chance. They were not further developed.

In hopes that model performance would improve with a larger dataset that was more similar to EOL Angiosperm images, an additional pipeline for classifying images of flowers and fruits was built using manually sorted training classes of EOL images ($n = 850$ images per class; table 1; table S1). Models learned during training, but overpredicted flowers during inspection of results with EOL validation images. Since plant reproductive structures can be small relative to image size, we suspected this task may be better for object detection than classification (figure 1b; figure 2b). We evaluated object detection with YOLOv3 pretrained on Google Open Images v4 (table 1). Detections for "Flower," "Fruit," and "Food" (upon manual inspection, this tag was often applied to fruits and flowers) were converted to "reproductive structures present" during postprocessing. During manual inspection, 2 of 258 predicted positives were false.

To identify image type (herbarium sheet, illustration, map, phylogeny, or none), training datasets ($n = 2000$ images per class) were built from sources within and outside of EOL (table 1; table S1). Transfer learning was done with MobileNetSSD v2 (Sandler et al. 2018) and Inception v3 (Szegedy et al. 2016) models pretrained on ImageNet

Table 3. Evaluation metrics for successfully implemented EOL Classification for Image Tagging pipelines.

Target group	Target output	No. Classes	Train accuracy	Val accuracy	Precision	Recall	F1	No. Images	Post-processing	Predictions retained	Final accuracy	Model	Model citation
All	Image quality rating tags	2	0.60	0.75	0.68	0.82	0.78	2000	confidence > 1.5	708	0.82	Inception v3	Szegedy and colleagues (2016)
All	Image type tags	5	0.90	0.77	0.79	0.77	0.76	7042	mnorm pp = 2 confidence > 1.6	3590	0.92	MobileNet SSD v2	Sandler and colleagues (2018)

Train accuracy is measured at the end of model training on test images. Validation accuracy is measured after training on novel validation images. Precision represents model accuracy for all positive identifications and recall also considers how many true positives the model failed to predict. F1 score is the harmonic mean of precision and recall. Predictions Retained are the number of images whose predictions were retained after filtering by the postprocessing thresholds. postprocessing thresholds for image quality ratings were chosen using histograms in figure S3 and for image type tags in figure 6 and figure 7. Final Accuracy is model accuracy after postprocessing thresholds. Statistics were generated using scikit-learn evaluation metrics in "inspect_train_results.ipynb" for each respective target group and are available on our GitHub [Computer Vision with EOL Images GitHub, Scikit-learn 2026]. See figure 5 for descriptions and illustrations of metrics.

2012 (Russakovsky et al. 2015). Herbarium sheet, map, and phylogeny classes had few misidentifications, whereas illustrations were inconsistent (figure 6a). Confidence thresholds (confidence > 1.6) and image properties (Manhattan norm per pixel = 2) were leveraged to produce final image type tags (table 3; figure 6). To classify images as photographic or not, images were "cartoonized" using edge creation and bilateral filtering. The difference between original and cartoonized images was measured using Manhattan norm (sum of absolute values of pixel differences) and Zero norm (number of non-zero pixel differences). Values for photorealistic images change more than those for non-photorealistic images (figure 6b). When tags from classification and "cartoonization" agreed for "map," "illustration," "phylogeny," or "herbarium sheet," a tag of "non-photographic image" or "herbarium sheet" was added (figure 7a) otherwise a tag of "NA" was added (figure 7b; detailed steps are available in cartoonization.ipynb [Computer Vision with EOL Images GitHub 2026]).

To evaluate performance for Classification for Image Tagging pipelines, we compared model predictions against known image tags using precision, recall, and F1 scores (figure 5b, 5c). Models achieved a train accuracy of 60% (Image Quality Rating) and 90% (Image Type; table 3). During validation, they achieved accuracy of 75% (Image Quality Rating) and 77% (Image Type). After postprocessing steps, models achieved a final accuracy of 82% (Image Quality Rating) and 92% (Image Type; figure 2c; figure 7; table 3).

Key takeaways for future CV efforts with biodiversity image databases

Using EOL CV approaches

We provide a user guide for EOL computer vision pipelines, including inputs and outputs for object detection in figure 8a and image classification in figure 8b. CV for EOL images code and documentation are available on GitHub [Computer Vision with EOL Images GitHub 2026]. Information needed to download raw training datasets used to develop our pipelines is available in table S1. Training settings and hyperparameters for the best resulting model from each CV pipeline are found in table S2. To download trained models, see the EOL Model Zoo on Kaggle [EOL Model Zoo 2026]. To download produced cropping and tagging datasets, see the EOL Computer Vision Pipelines on Zenodo [EOL CV Datasets on Zenodo 2026].

Image tags can be generated using object detection or image classification. For tasks with localized ROI (for example, finding insects within plant photos), object detection should be used (figure 1b; figure 2a, 2b). Object detection training dataset preparation takes more time (bounding boxes must be drawn on each training image), but smaller datasets can be used (600–800 images per class; figure S1). For image-wide problems, image classification should be used (figure 1b; figure 2c). Preparing datasets is faster (images only need to be sorted into folders by class), but larger datasets are required (>1000 images per class; figure S2). Either task can be crowdsourced with quality control measures in place to ensure consistency.

Datasets, model training, and evaluation

When no existing datasets or models are available, building a sufficiently large and diverse training dataset is important, with extra curation needed for test images (figure 3-2). Van Horn and colleagues (2015) found that when ImageNet bird labels had an error rate of 5%, model performance was only negatively affected if false images were in the test dataset. Even with large datasets, ensuring consistency still matters. Despite having > 35,000 images for training the EOL image quality rating classifier, initially, only 44% accuracy was achieved with MobileNetSSD v2 and 60% with Inception v3 (figure S3; table 3). Upon closer inspection of model predictions and our crowdsourced training data, we determined that users were consistent on which images they did not like, but conflicted on which images they liked. This is known as negativity bias, a psychological phenomenon where negative impressions are stronger than positive ones (Baumeister et al. 2001). Instead of manually curating datasets to retrain the model, we used postprocessing to only keep predictions for "bad" images with confidence > 1.5, resulting in a final accuracy of 82%.

For future efforts, we recommend producing large enough training datasets to be able to split data into 80–15–5 into train–test–validation. Just because a model achieves near 100% accuracy on training examples, it is unlikely it will achieve the same accuracy on real world data. Reserving 5% of curated data to not be seen by models during training or testing allows objective quantification of real world performance and fine-tuning to achieve the best outputs. Tools like DivShift can be used to quantify bias in crowdsourced biodiversity data to best leverage available resources and interpret model predictions accordingly (Sierra et al. 2025). Although using PlantCLEF 2016 images to custom train a plant phenology classifier did not produce useful tags for our pipelines, as more ML-ready biological image datasets are published (see for example, Stevens et al. 2024; Tree of Life Dataset 2026), building custom classifiers with limited upfront investment in data curation will be simplified and streamlined for better results. Further performance improvements could be gained by using a CNN + RNN (Recurrent Neural Network) hybrid approach to account for multiple, hierarchical tags per image, instead of presence or absence for independent categories (for example, "daisy" is nested within Asteraceae which is nested within Angiospermae; Wang et al. 2016).

We found significant model performance increases with minor adjustments to hyperparameters (table S2). For each model, we started with default hyperparameters and then systematically fine-tuned them, beginning with batch size, epochs, learning rate, and dropout. Monitoring learning curves across training attempts is invaluable for tracking how each change affects model performance (figure 3-3; figure 4d). Batch size should be as large as GPU RAM allows to speed up training attempts. Epochs should be high enough for learning to plateau. To avoid overfitting, learning rate should be adjusted so the model achieves the highest accuracy possible for train and test data (figure 4a). If needed, dropout and L2 regularization (via optimizers and kernel regularizers/epsilon values) can be added to prevent overfitting. Epsilon is a small constant value used by optimizers to improve stability during gradient scaling. Anchor box dimensions can be adjusted to better fit target object dimensions (illustrated in figure 1c). For instance, detecting long, narrow organisms (like needlefish, Family: Belonidae) may benefit from narrower bounding boxes than the default dimensions. After logging hyperparameters and training graphs across attempts, tools like Google Facets can identify hyperparameter combinations that achieve the highest per-class prediction accuracy under different thresholding scenarios (Wexler et al. 2020). All pipelines found in our GitHub log hyperparameters and training graphs, and they include form fields (dropdowns, sliders, or raw input boxes) to easily adjust hyperparameters for retraining.

For cases where model outputs were not as accurate or reliable as desired, we found flexible approaches using confidence thresholds to maximize coverage and minimize error (figure 3-4; figure S3). We also successfully incorporated image properties (colorspace, pixelwise

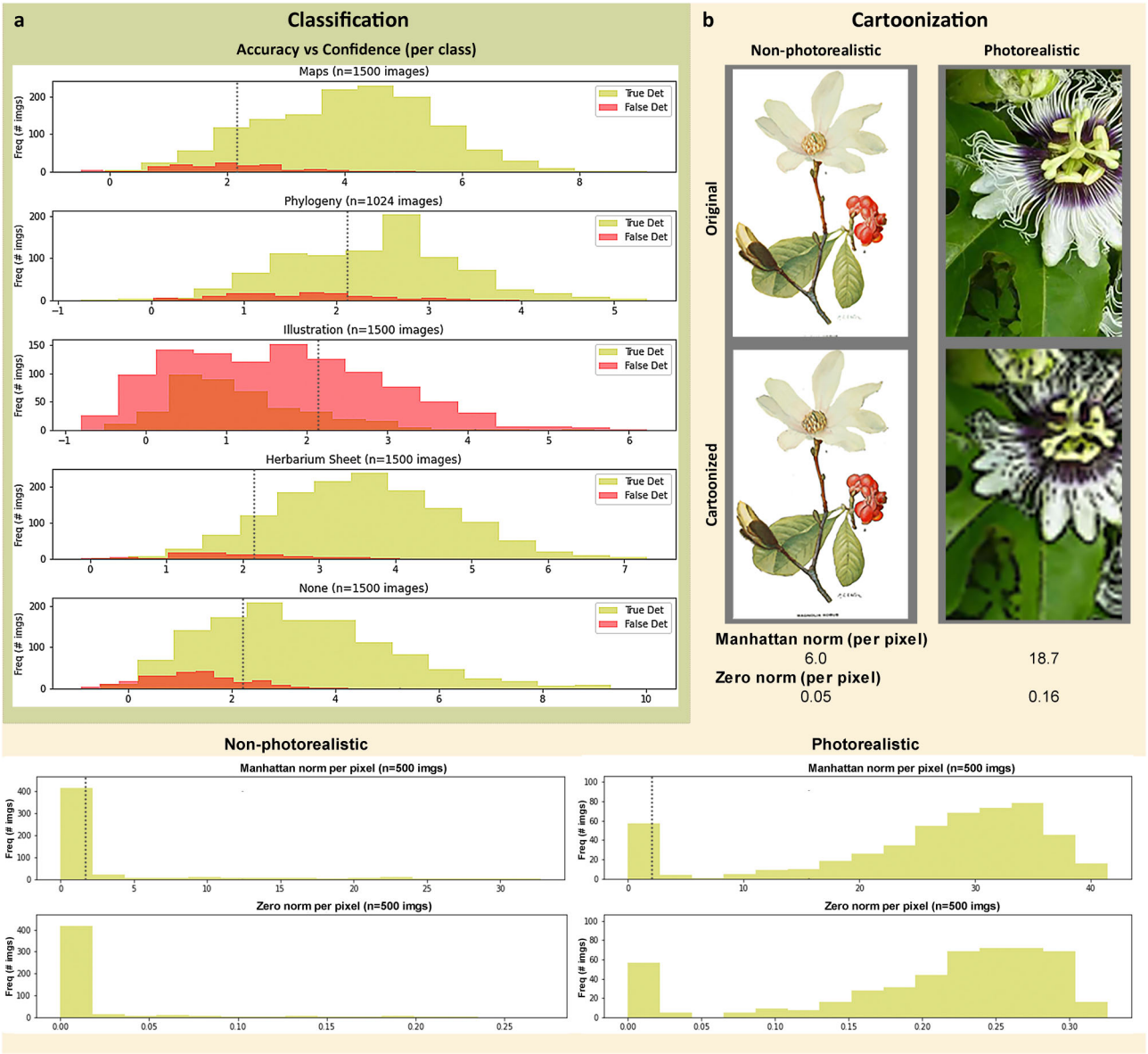


Figure 6. Classification (a; shown in green) and "cartoonization" (b; shown in beige) methods used to determine postprocessing steps and add tags for image type. (a) To filter initial model predictions, a confidence threshold of 2.2 (shown as a dotted line) was used. Since some classes (illustrations and herbarium sheet) still had low accuracy, cartoonization was used as a secondary filter. (b) Cartoonization leverages image color properties and image processing techniques to classify as photorealistic (Herbarium Sheet, None) or non-photorealistic (Map, Phylogeny, Illustration). Following edge creation and bilateral filtering, pixelwise differences between original and cartoonized images were calculated. Photorealistic images change more from being "cartoonized" than non-photorealistic images (b, top row), meaning they have higher Manhattan norm (sum of absolute values of pixel differences) and Zero norm (number of non-zero pixel differences) values. Histograms were used to identify Manhattan norm per pixel > 2 (shown as a dotted line) as the threshold for classifying an image as photorealistic (b, bottom row).

color differences, edge properties) as a secondary method of image tagging to predict image types (figure 6; figure 7).

Advances and outlook for CV and biodiversity research

Since the time our CV experiments were carried out, the field has grown rapidly with frameworks, like PyTorch (Paszke et al. 2019), and models, like BIOCLIP (Stevens et al. 2024), gaining popularity for biological tasks. We will briefly provide an update on these exciting new directions of the field and their implications for museums and biodiversity research. Specifically, we will touch on advances in multimodal models, foundation models, and optical character recognition (OCR), as well as challenges associated with model interpretability.

Vision foundation models are large, pretrained AI models that can reason about a broad range of domains without task-specific fine-tuning. Dinov3 is a self-supervised foundation model that was released in August 2025 by MetaAI (Siméoni et al. 2025). Like other foundation models, Dinov3 learns high–quality image features without needing manual annotations. Since there are no image classes, bounding boxes, or segmentation maps, Dinov3 is a versatile model that achieves state of the art performance without any fine–tuning.

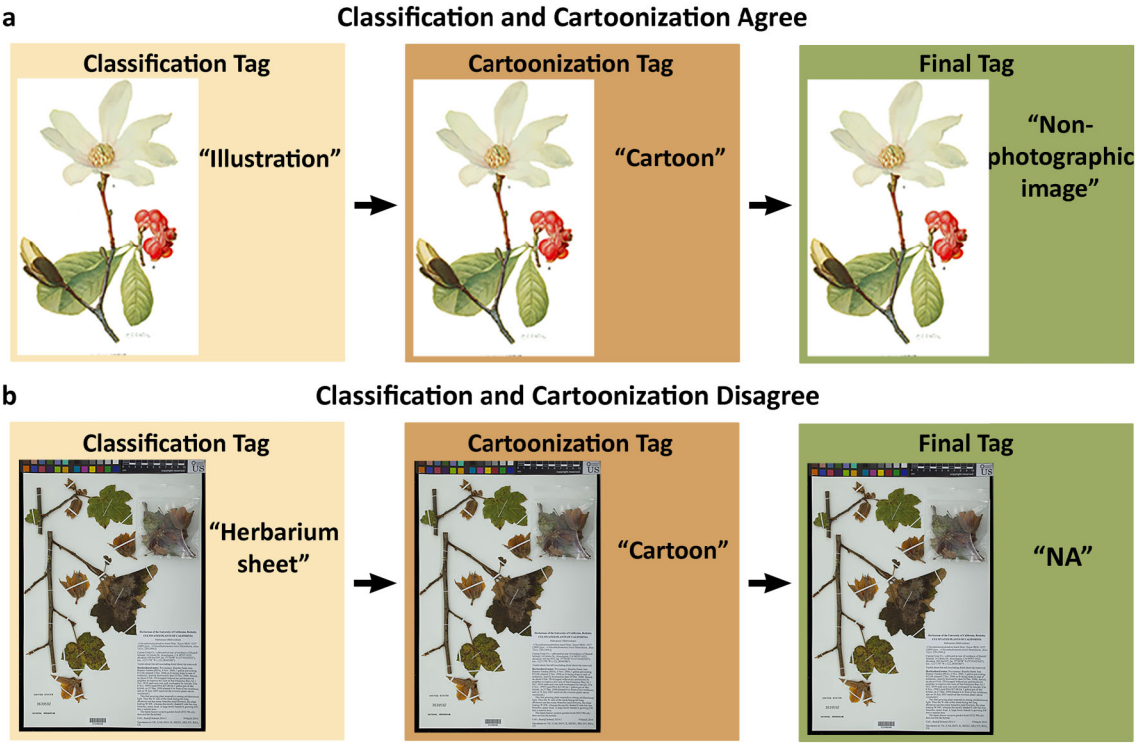


Figure 7. Classification for image tagging pipeline output for image type pipelines using postprocessing with "cartoonization." To maximize useful outputs from the pipeline, we added a secondary tagging step that used "cartoonization" described in figure 6b, to identify images as "cartoon" or "photographic." (a) When tags from classification and cartoonization agree for "map," "illustration," "phylogeny," or "herbarium sheet," a final tag of "non-photographic image" or "herbarium sheet" is added. (b) When tags from classification and cartoonization do not agree, a final tag of "NA" is added.

Multimodal models learn from multiple types of data, like text and images, and include several recent influential models, like CLIP and ChatGPT (Baltrušaitis et al. 2019; ChatGPT 2023). CLIP models (Contrastive Language–Image Pretraining) are trained on hundreds of millions of image-text pairs to learn contextual relationships using an image encoder and text encoder (Radford et al. 2021). BIOCLIP is a multimodal vision foundation model trained on Tree of Life Dataset (Stevens et al. 2024; Tree of Life Dataset 2026), which is the largest machine learning dataset of biological images (10 million images of > 400,000 taxa, with images sourced from EOL, iNaturalist, and BIOSCAN).

Traditional OCR is one of the most widely adopted CV approaches in museums for its ability to detect and transcribe text data from specimen labels (Barber et al. 2013, Drinkwater et al. 2014, Kirchhoff et al. 2018, Weaver et al. 2023, Stenhouse and Thrall 2025, Turnbull et al. 2025; reviewed in Tessler and Little 2025). In contrast, DeepSeek-OCR is a multimodal approach leveraging a vision encoder and LLM that renders text as images, then transcribes and interprets the contents (Wei et al. 2025). Following the adage, "A picture is worth a thousand words," DeepSeek-OCR achieved 97% accuracy on OCR benchmark datasets, whereas other state-of-the-art deep learning approaches achieve 85–90% accuracy. Although traditional OCR works well for short text strings, like specimen labels, DeepSeek-OCR could be able to parse historical scientific manuscripts containing elaborate tables, figures, and a lengthy text body.

Machine learning approaches are often described as "black boxes," since even experts are not able to mathematically or logically explain why a certain input to a model produces a certain output. As the use of AI increases, there is a new push for "explainable AI," where model results can be better interpreted. For CV, interpreting model predictions often involves some combination of supervised and unsupervised visualization techniques to determine which parts of images are used for model predictions. For example, Jorgensen and colleagues (2025) used quantitative structure-property relationships to map molecular traits to physical properties for interpreting model results instead of relying solely on opaque latent vectors. Class activation mapping (CAM) approaches are also popular for identifying spatial features of images used to distinguish between classes (Zhou et al. 2016). By linking model predictions to image traits, they allow the end user to better interpret model outputs and compare them with human validation results. Although traditional CAMs highlight shared features across multiple classes, Finer-CAM uses contrast-based details to filter feature maps from trained models and hone in on unique details used to discriminate one class from another (Zhang et al. 2025). On the other hand, Prompt-CAM learns class specific traits, which when present, result in class predictions (Chowdhury et al. 2025). It emphasizes distinguishing between classes less and predicting classes based on present traits more.

New implementations using out-of-the box-approaches for EOL images

To highlight how our approaches can be integrated with these new frameworks, we present three additional pipelines that utilize new models to perform image management tasks, which are also available on our GitHub and can serve as a guide for other biodiversity database managers wanting to use CV [Play for Beginners Notebooks 2026].

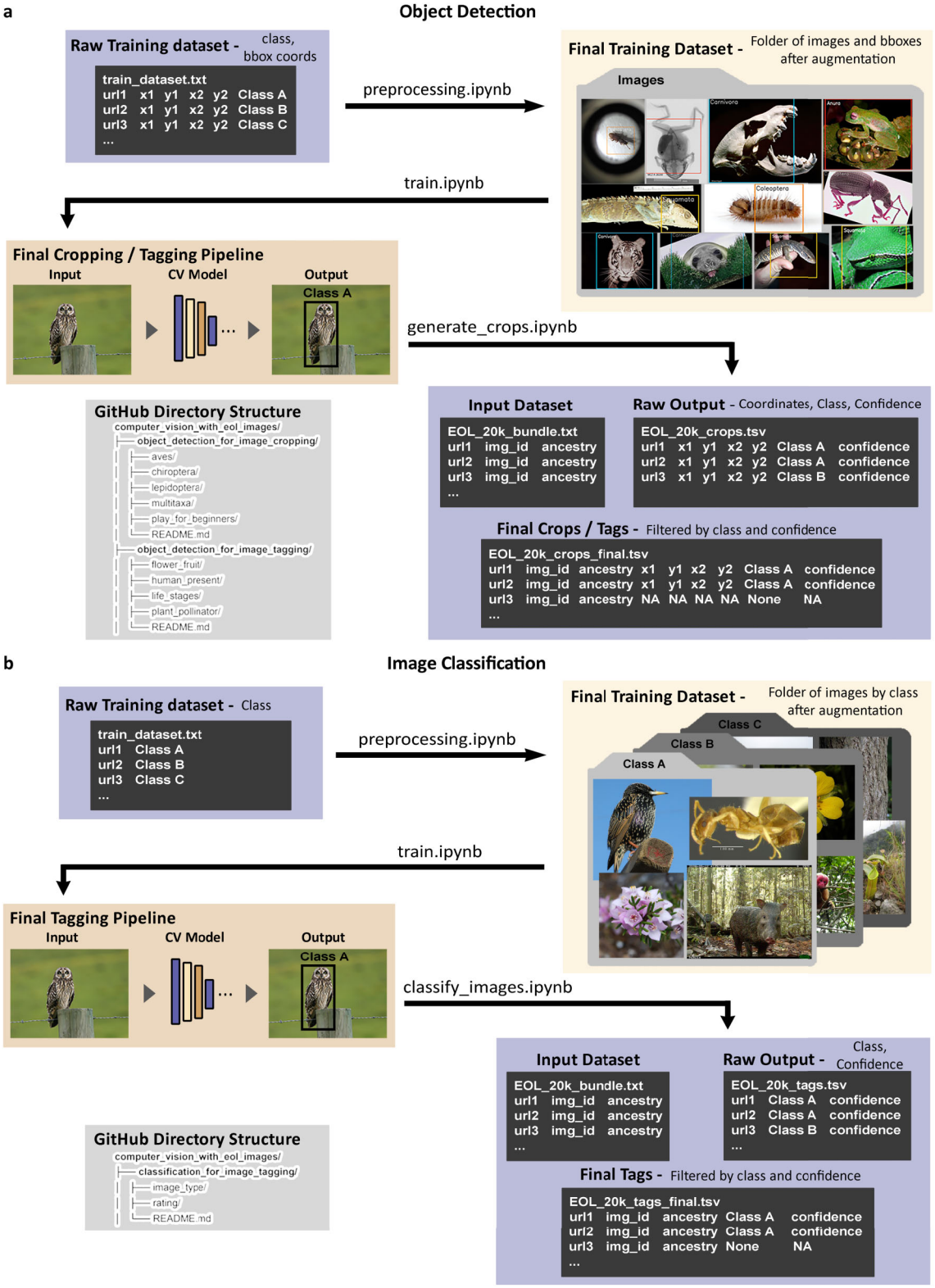


Figure 8. User guide for EOL computer vision pipelines with inputs and outputs for each Google Colab notebook (.ipynb files) shown for object detection (a) and image classification (b), along with GitHub directory structures for all available pipelines. If using our pretrained pipelines for cropping or tagging, only generate_crops.ipynb or classify_images.ipynb are needed. If modifying and retraining for other purposes, follow all steps. Training datasets consist of text files with information about image class and location using bounding box coordinates (a; for object detection) or image class only (b; for image classification). Images are downloaded to the required directory structure for each framework in preprocessing.ipynb scripts. Models are trained in train.ipynb scripts. The resulting models can localize (a) and/or identify (b) image contents. In generate_crops.ipynb/classify_images.ipynb, EOL 20k image bundles are run through trained models. The output datasets are then postprocessed and filtered using confidence thresholds and image class before producing the final cropping (a) or tagging (b) datasets.

YOLOv8 is the 8th version of the You Only Look Once (YOLO) type of real-time object detection models (Reis et al. 2024). It is a lightweight CNN that can be used for object detection in real-time and is implemented in pytorch. YOLOv8 achieves ∼20% higher mAP scores on MS COCO images [MS COCO Dataset 2026], is 2x-4x faster for running inference, and is more easy and lightweight to run in the command line with fewer dependencies. We provide a demo notebook using YOLOv8 (Reis et al. 2024) pretrained on Google Open Images v7 [Google Open Images 2026] as an update to our Object Detection for Image Tagging notebooks [Computer Vision with EOL Images GitHub 2026], which are built on YOLOv3 pretrained on Google Open Images v6 (Redmon and Farhadi 2018). For future image processing and data management at EOL, YOLOv8 could be used for adding tags for species interactions not included in metadata, for quality control or data cleaning, by identifying empty or irrelevant images, or for adding counts of the number of individuals per image.

BIOCLIPv2 is a multimodal model trained on the Tree of Life Dataset (Stevens et al. 2024). We provide a demo that uses BIOCLIPv2 to detect visiting insects within EOL Angiosperm images as a possible replacement for our current plant-insect tagging pipeline, built on YOLOv3 (Redmon and Farhadi 2018). For future approaches, BIOCLIP could be used for tasks like text-to-image search (Natural Language Queries; "images of birds with black and yellow stripes"), as a universal cropping tool based on image focal points (saliency maps), and to add tags for interesting image features (like "insect visitor" tags for EOL Angiosperm images).

Dinov3 is a self-supervised foundation model that learns image features without needing manual annotations (Siméoni et al. 2025). We provide a demo notebook that uses DINOv3 as a universal cropping total that without any required image annotations. To further improve image management, DINOv3 could be used for tasks like image similarity searches (content-based image retrieval), text-to-image queries using multimodal methods (like combining Dinov3 with BIOCLIP), and as a universal cropping tool based on image focal points (saliency maps).

Although our experiments presented here use older frameworks and models, the challenges of managing biodiversity databases persist and CV remains an underutilized resource in our field. We hope our methods of finding datasets, training models, and postprocessing results serve as a helpful guide to increase the speed and depth of biodiversity informatics research of the future. As more approaches are published, museums can work toward a global infrastructure to fully leverage the scientific potential of digitized natural history collections and better understand the planet's biodiversity of the past, present, and future (Groom et al. 2023).

Conclusions

Using CV during the transition from EOL v2 to v3 allowed us to automate management of millions of images. Image cropping adds uniformity to gallery displays and increases functionality for API users who often request square thumbnails. Image tagging can increase the discoverability and utility of our images, enabling users to search for content beyond taxonomy (e.g., plant-insect co-occurrence). Training CV models with existing crowdsourced datasets saved time and resources, while incorporating users into our forthcoming version. Our pipelines are open source and run entirely in the cloud, so they can be used by anyone as-is, or modified for other use cases with guidance from our documentation. We hope that our provided training datasets, trained models, and strategies (table 1; table S1; table S2; figure 3) will serve as a helpful guide for building future biodiversity CV projects.

Although many approaches in biology use CV for species delimitation and feature extraction, efforts to apply these technologies to image management tasks are minimal. Lessons learned through our experimentation with the EOL database can guide future efforts to better manage the increasing volume of biodiversity images. Even with some unexpected results, in nearly all cases, models produced valuable outputs. Future approaches should remain flexible about ways to use a combination of machine learning-ready biological datasets (e.g., Ávila et al. 2025, Barkmann et al. 2025, Stevens et al. 2024), the latest pre-trained models for general approaches (e.g., Stevens et al. 2024, Siméoni et al. 2025, Wei et al. 2025; Reis et al. 2024), and post-processing of outputs to tailor results for specific biological questions (e.g., figure 3-4; figure 6; figure 7). Using CV to manage museum image databases will expedite biodiversity informatics research and facilitate new global discoveries from existing archival data.

Supplemental material

Supplemental data are available at *BIOSCI* online.

Acknowledgments

We thank Eli Agbayani for making image breakdown-download bundles, running images through built pipelines, and uploading produced data files to Zenodo. For administrative support through the duration of the project, we thank David Kanner. We also thank two anonymous reviewers, BioScience Editor-in-Chief Charles Fenster, Senior Editor James Verdier, Handling Editor Hayley Elliot, and the Editorial Board Member for their suggestions that greatly improved the manuscript.

Data availability

The code associated with this article is available in the CV for EOL Images GitHub (https://github.com/EOL/computer-vision-with-EOL-images). Custom trained models are available on EOL Model Zoo on Kaggle (https://www.kaggle.com/eolorg/models). EOL user generated data used to train models is available on Zenodo (Object Detection for Image Cropping: https://doi.org/10.5281/zenodo.1330556, Classification for Image Tagging—Ratings: https://doi.org/10.5281/zenodo.13305561). EOL image tagging and cropping datasets generated through this study are available on Zenodo (https://zenodo.org/communities/eol/records?q=computer%20vision&l=list&p=1&s=10&sort=bestmatch). Licensing information for images used in all figures is available in Table S3.

Author contributions

K. Wolcott (Conceptualization, Data curation, Formal Analysis, Investigat ion, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing), J. Hammock (Conceptualization, Project administration, Supervision, Writing – review & editing), K. Schulz (Conceptualization, Project administration, Supervision, Writing – review & editing), T.M. Orrell (Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing)

Conflict of interest

The authors have no conflicts of interest to declare.

References cited

Abadi M, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. *OSDI '16: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation; 2–4 November 2016*, Savannah, USA. https://dl.acm.org/doi/10.5555/3026877.3026899

Agarwal S, Terrail JO D, Jurie F. 2018. Recent advances in object detection in the age of deep convolutional neural networks. *arXiv*. https://doi.org/10.48550/arXiv.1809.03193

Ahn JW, Hammock J, Parr C, Preece J, Shneiderman B, Schulz K, Hansen D, Rotman D, He Y. 2012. Visually exploring social participation in Encyclopedia of Life. *Proceedings of the 2012 ASE International Conference on Social Informatics; 14–16 December 2012*, Alexandria, USA. https://doi.org/10.1109/SocialInformatics.2012.10

Alony I, Haski-Leventhal D, Lockstone-Binney L, Holmes K, Meijs LCPM. 2020. Online volunteering at DigiVol: An innovative crowd-sourcing approach for heritage tourism artefacts preservation. *Journal of Heritage Tourism* 15: 14–26. https://doi.org/10.1080/1743873X.2018.1557665

Ávila FA, Park JY, Feder L, Little DP. 2025. Herbariograph: A deep-learning tool to classify specimen images. *New Phytologist*. https://doi.org/10.1111/nph.70447

Azmy SN, Tarmidi Z, Hassan N, Johari SA, Shamsir Omar MS. 2024. Biodiversity data crowdsourcing: A way forward. *12th IGRSM International Conference and Exhibition on Geospatial & Remote Sensing; 29–30 April 2024*, Kuala Lumpur, Malaysia. https://doi.org/10.1088/1755-1315/1412/1/012023

Baltrušaitis T, Ahuja C, Morency L-P. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41: 423–443. https://doi.org/10.1109/TPAMI.2018.2798607

Barber A, Lafferty D, Landrum LR. 2013. The SALIX method: A semi-automated workflow for herbarium specimen digitization. *Taxon* 62: 581–590. https://doi.org/10.12705/623.16

Barkmann F, Lindner A, Würflinger R, Höttinger H, Rüdisser J. 2025. Machine learning training data: Over 500,000 images of butterflies and moths (Lepidoptera) with species labels. *Scientific Data* 12. https://doi.org/10.1038/s41597-025-05708-z

Baumeister RF, Bratslavsky E, Finkenauer C, Vohs KD. 2001. Bad is stronger than good. *Review of General Psychology* 5: 323–370. https://doi.org/10.1037/1089-2680.5.4.323

Blair J, Weiser MD, de Beurs K, Kaspari M, Siler C, Marshall KE. 2022. Embracing imperfection: Machine-assisted invertebrate classification in real-world datasets. *Ecological Informatics* 72: 101896. https://doi.org/10.1016/j.ecoinf.2022.101896

Bochkovskiy A, Wang CY, Liao HYM. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv*. https://doi.org/10.48550/arXiv.2004.10934

Borowiec ML, Dikow RB, Frandsen PB, McKeeken A, Valentini G, White AE. 2022. Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution* 13: 1640–1660. https://doi.org/10.1111/2041-210X.13901

Buda M, Maki A, Mazurowski MA. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106: 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

ChatGPT. ChatGPT. OpenAI. 2023. (27 March 2026; *https://openai.com/index/gpt-4-research/).*

Chowdhury A, Paul D, Mai Z, Gu J, Zhang Z, Mehrab KS, Campolongo EG, Rubenstein D, Stewart CV, Karpatne A, Berger-Wolf T, Su Y, Chao W-L. 2025. Prompt-CAM: Making vision transformers interpretable for fine-grained analysis. *arXiv*. https://doi.org/10.48550/arXiv.2501.09333

Computer Vision with EOL Images GitHub. Computer vision with EOL images. EOL. (27 March 2026; *https://github.com/EOL/computer-vision-with-EOL-images/).*

Darkflow GitHub. 2026. darkflow. thtrieu. (27 March 2026; *https://github.com/thtrieu/darkflow).*

Darknet GitHub. darknet. pjreddie. (27 March 2026; *https://github.com/pjreddie/darknet).*

Dogo EM, Afolabi OJ, Nwulu NI, Twala B, Aigbavboa CO. 2018. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural network. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems; 21–23 December 2018*, Belagavi, India. https://doi.org/10.1109/CTEMS.2018.8769211

Drinkwater RE, Cubey RWN, Haston EM. 2014. The use of optical character recognition (OCR) in the digitisation of herbarium specimen labels. *PhytoKeys* 38: 15–30. https://doi.org/10.3897/phytokeys.38.7168

East A, Campolongo EG, Meyers L, Rayeed SM, Stevens S, Zarubiieva I, Fluck IE, Girón JC, Jousse M, Lowe S, Perry KI, Betancourt I, Charney N, Donoso E, Fox N, Landsbergen KJ, Nepovinnykh E, Ramirez M, Singh P, Thapa-Magar K, Thompson M, Waite E, Berger-Wolf T, Lapp H, Mabee P, Stewart C, Taylor G, Record S. 2025. Optimizing image capture for computer vision-powered taxonomic identification and trait recognition of biodiversity specimens. *Methods in Ecology and Evolution* 16: 2260–2275. https://doi.org/10.1111/2041-210X.70140

EOL API. Encyclopedia of life advanced programming interface (API). EOL. (27 March 2026; https://eol.org/docs/what-is-eol/data-services).

EOL CV Datasets on Zenodo. EOL Computer Vision datasets. Zenodo. (*27 March 2026*; *https://zenodo.org/communities/eol/records?q=%22eol%20computer%20vision%20pipelines%22&l=list& p =1&s=10&sort=bestmatch)*

EOL Model Zoo. EOL Model Zoo. Kaggle. (27 March 2026; *https://www.kaggle.com/eolorg/models).*

EOL. Encyclopedia of Life. EOL. (27 March 2026; *https://eol.org/).*

Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88: 303–338. https://doi.org/10.1007/s11263-009-0275-4

Folk RA, Guralnick RP, LaFrance RT. 2024. FloraTraiter: Automated parsing of traits from descriptive biodiversity literature. *Applications in Plant Sciences* 12. https://doi.org/10.1002/aps3.11563

Fukushima K. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36: 193–202. https://doi.org/10.1007/BF00344251

Gaston KJ, O'Neill MA. 2004. Automated species identification: Why not? *Philosophical Transactions of the Royal Society of London Series B: Biological Sciences* 359: 655–667. https://doi.org/10.1098/rstb.2003.1442

Geirhos R, Jacobsen JH, Michaelis C, Zemel R, Brendel W, Bethge M, Wichmann FA. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2: 665–673. https://doi.org/10.1038/s42256-020-00257-z

Goëau H, Bonnet P, Joly A. 2017. Plant identification based on noisy web data: The amazing performance of deep learning (LifeCLEF 2017). *Workshop Proceedings of Conference and Labs of the Evaluation Forum (CLEF 2017); 11–14 Sep 2017*, Dublin, Ireland. https://ceur-ws.org/Vol-1866/invited_paper_9.pdf

Google Colab Pro Signup. Google Colab. Google. (27 March 2026; *https://colab.research.google.com/signup)*.

Google Open Images. *Open Images Dataset V7 and Extensions* (27 March 2026; *https://storage.googleapis.com/openimages/web/index.html)*

Greener JG, Kandathil SM, Moffat L, Jones DT. 2022. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology* 23: 40–55. https://doi.org/10.1038/s41580-021-00407-0

Groom Q, et al. 2023. Envisaging a global infrastructure to exploit the potential of digitised collections. *Biodiversity Data Journal* 11: e109439. https://doi.org/10.3897/BDJ.11.e109439

Ground truths versus model predictions. Zenodo. Katherine Wolcott. (27 March 2026; *https://doi.org/10.5281/zenodo.14853913)*.

Guo YY, Cai H, Bramley GLC, Atkins HJ, Li B, Theodossiades S. 2025. A review of artificial intelligence in herbarium specimen image analysis. *Artificial Intelligence Review* 58: 1–46. https://doi.org/10.1007/s10462-025-11408-2

He K, Zhang X, Ren S, Sun J. 2015. Deep residual learning for image recognition. *arXiv*. https://doi.org/10.48550/arXiv.1512.03385

Hedrick Brandon P. Heberling J Mason. Meineke Emily K. Turner Kathryn G. Grassa Christopher J. Park Daniel S. Kennedy Jonathan. Clarke Julia A. Cook Joseph A. Blackburn David C. Edwards Scott V. Davis Charles C. 2020. Digitization and the Future of Natural History Collections. *BioScience* 70, 243–251.

Hardisty Alex, Saarenmaa Hannu, Casino Ana, Dillen Mathias, Göddderz Karsten, Groom Quentin, Hardy Helen, Koureas Dimitris, Nieva de la Hidalga Abraham, Paul Deborah, Runnel Veljo, Vermeersch Xavier, van Walsum Myriam, Willemse Luc. 2020. Conceptual design blueprint for the DiSSCo digitization infrastructure - DELIVERABLE D8.1. *Research Ideas and Outcomes* 6: e54280.

Hornik K, Stinchcombe M, White H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21-26 July 2017*, Honolulu, USA. https://doi.org/10.1109/CVPR.2017.351

Hugging Face Models. Hugging Face Models. (27 March 2026; *https://huggingface.co/models)*

iDigBio. iDigBio. (27 March 2026; *https://www.idigbio.org/)*

ImageNet. ImageNet. (27 March 2026; *https://image-net.org/)*

Imgaug GitHub. imgaug. aleju. (27 March 2026; *https://github.com/aleju/imgaug)*.

iNaturalist. iNaturalist. (27 March 2026; *https://www.inaturalist.org/)*

Joly A, Goëau H, Glotin H, Spampinato C, Bonnet P, Vellinga WP, Lombardo JC, Planqué R, Palazzo S, Müller H. 2017. LifeCLEF 2017 lab overview: Multimedia species identification challenges. *Proceedings of the 2017 International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF); 11–14 Sep 2017*, Dublin, Ireland. https://doi.org/10.1007/978-3-319-65813-1_24

Jorgensen C, Lin AY, Vasavada R, Cersonsky RK. 2025. Interpretable visualizations of data spaces for classification problems. *arXiv*. https://doi.org/10.48550/arXiv.2503.05861

Kaggle. Kaggle. (27 March 2026; *https://www.kaggle.com)*

Kingma DP, Ba J. 2017. Adam: A method for stochastic optimization. *arXiv*. http://arxiv.org/abs/1412.6980

Kirchhoff A, Bügel U, Santamaria E, Reimeier F, Röpert D, Tebbje A, Güntsch A, Chaves F, Steinke KH, Berendsohn W. 2018. Toward a service-based workflow for automated information extraction from herbarium specimens. *Database* 2018: 1–11. https://doi.org/10.1093/database/bay103

Kommineni VK, Tautenhahn S, Baddam P, Gaikwad J, Wieczorek B, Triki A, Kattge J. 2021. Comprehensive leaf size traits dataset for seven plant species from digitised herbarium specimen images covering more than two centuries. *Biodiversity Data Journal* 9: 1–41. https://doi.org/10.3897/BDJ.9.e69806

Krizhevsky A, Sutskever I, Hinton G. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems. 1097–1105; 27 March 2026.* https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Kuznetsova A, et al. 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision* 128: 1956–1981. https://doi.org/10.1007/s11263-020-01316-z

LeCun Y, Bengio Y. 1995. Convolutional networks for images, speech, and time-series. Pages 255–258 in Arbib M, ed. *The Handbook of Brain Theory and Neural Networks*. MIT Press. https://dl.acm.org/doi/10.5555/303568.303704

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* 521: 436–444. https://doi.org/10.1038/nature14539

LeCun Y, Haffner P, Bottou L, Bengio Y. 1999. Object recognition with gradient-based learning. Pages 319–345 in Forsyth DA, Mundy JL, Gesu V, Cipolla R, eds. *Shape, Contour, and Grouping in Computer Vision*. Springer Verlag. https://doi.org/10.1007/3-540-46805-6_19

Lendemer J, et al. 2020. The extended specimen network: A strategy to enhance us biodiversity collections, promote research and education. *BioScience* 70: 23–30. https://doi.org/10.1093/biosci/biz140

Lin T-Y, Goyal P, Girshick R, He K, Dollár P. 2017. Focal loss for dense object detection. *arXiv*. http://arxiv.org/abs/1708.02002

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. 2016. SSD: Single shot multibox detector. *Computer Vision—ECCV 2016: 14th European Conference, 11-14 October 2016*, Amsterdam, Netherlands. https://doi.org/10.1007/978-3-319-46448-0_2

Lorieul T, Pearson KD, Ellwood ER, Goëau H, Molino JF, Sweeney PW, Yost JM, Sachs J, Mata-Montero E, Nelson G, Soltis PS, Bonnet P, Joly A. 2019. Toward a large-scale and deep phenological stage annotation of herbarium specimens: Case studies from temperate, tropical, and equatorial floras. *Applications in Plant Sciences* 7: 1–14. https://doi.org/10.1002/aps3.1233

Lürig MD, Donoughe S, Svensson EI, Porto A, Tsuboi M. 2021. Computer vision, machine learning, and the promise of phenomics in ecology and evolutionary biology. *Frontiers in Ecology and Evolution* 9: 642774. https://doi.org/10.3389/fevo.2021.642774

mAP GitHub. mAP. Cartucho. (27 March 2026; *https://github.com/Cartucho/mAP)*.

Mika K, Deveer J, Rinaldo C. 2017. Crowdsourcing natural history archives: Tools for extracting transcriptions and data. *Biodiversity Informatics* 12. https://doi.org/10.17161/bi.v12i0.6646

Miralles Aurélien. Bruy Teddy. Wolcott Katherine. Scherz Mark D. Begerow Dominik. Beszteri Bank. Bonkowski Michael. Felden Janine. Gemeinholzer Birgit. Glaw Frank. Glöckner Frank Oliver. Hawlitschek Oliver. Kostadinov Ivaylo. Nattkemper Tim W. Printzen Christian. Renz Jasmin. Rybalka Nataliya. Stadler Marc. Weibulat Tanja. Wilke Thomas. Renner Susanne S. Vences Miguel. Friedman Matt. 2020. Repositories for Taxonomic Data: Where We Are and What is Missing. *Systematic Biology* 69, 1231–1253.

MS COCO Dataset. Common Images in Context. Microsoft. (27 March 2026; *https://cocodataset.org/)*.

Nwankpa C, Ijomah W, Gachagan A, Marshall S. 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv*. https://doi.org/10.48550/arXiv.1811.03378

Ordoñez JC, Tovar C, Walker BE, Wheeler J, Ayala-Ruano S, Aguirre-Carvajal K, McMahon SM, Cuesta F. 2025. Phenological patterns of tropical mountain forest trees across the neotropics: Evidence from herbarium specimens. *Proceedings of the Royal Society B: Biological Sciences* 292: 20242748. https://doi.org/10.1098/rspb.2024.2748

Osawa T, Yamanaka T, Nakatani Y, Nishihiro J, Takahashi S, Mahoro S, Sasaki H. 2017. A crowdsourcing approach to collecting photo-based insect and plant observation records. *Biodiversity Data Journal* 5: e21271. https://doi.org/10.3897/BDJ.5.e21271

Park DS, Lyra GM, Ellison AM, Maruyama RKB, Reis Torquato D, Asprino RC, Cook BI, Davis CC. 2023. Herbarium records provide reliable phenology estimates in the understudied tropics. *Journal of Ecology* 111: 327–337. https://doi.org/10.1111/1365-2745.14047

Parr CS, Wilson N, Leary P, Schulz KS, Lans K, Walley L, Hammock JA, Goddard A, Rice J, Studer M, Holmes JTG, Corrigan RJ. 2014. The Encyclopedia of Life v2: Providing global access to knowledge about life on earth. *Biodiversity Data Journal* 2: e1079. https://doi.org/10.3897/BDJ.2.e1079

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. 2019. PyTorch: An imperative style, high-performance deep learning library. *arXiv*. https://doi.org/10.48550/arXiv.1912.01703

Patton PT, et al. 2023. A deep learning approach to photo–identification demonstrates high performance on two dozen cetacean species. *Methods in Ecology and Evolution* 14: 2611–2625. https://doi.org/10.1111/2041-210X.14167

Pearson KD, Nelson G, Aronson MFJ, Bonnet P, Brenskelle L, Davis CC, Denny EG, Ellwood ER, Goëau H, Heberling JM, Joly A, Lorieul T, Mazer SJ, Meineke EK, Stucky BJ, Sweeney P, White AE, Soltis PS. 2020. Machine learning using digitized herbarium specimens to advance phenological research. *BioScience* 70: 610–620. https://doi.org/10.1093/biosci/biaa044

Play for Beginners Notebooks. Computer vision with EOL images GitHub. *EOL* (27 March 2026; *https://github.com/EOL/computer-vision-with-EOL-images/tree/master/play_for_beginners)*

Poveda-Cuellar JL, Morantes-Duarte D, Martínez-Carrillo F, García-Melo JE, Marchant S, Reu B. 2025. Deep learning on field photography reveals the morphometric diversity of Colombian freshwater fish. *Zoomorphology* 144: 67. https://doi.org/10.1007/s00435-025-00747-x

PyTorch Hub. PyTorch Hub for Researchers. (27 March 2026; *https://pytorch.org/hub/)*

Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv*. https://arxiv.org/abs/2103.00020

Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 27–30 June 2016*, Las Vegas, USA. https://doi.org/10.1109/CVPR.2016.91

Redmon J, Farhadi A. 2017. YOLO9000: Better, faster, stronger. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21-26 July 2017*, Honolulu, USA. https://doi.org/10.1109/CVPR.2017.690

Redmon J, Farhadi A. 2018. YOLOv3: An incremental improvement. *arXiv*. http://arxiv.org/abs/1804.02767v

Reis D, Kupec J, Hong J, Daoudi A. 2023. Real-time flying object detection with YOLOv8. *arXiv*. https://doi.org/10.48550/arXiv.2305.09972

Ren S, He K, Girshick R, Sun J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39: 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Russakovsky O, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115: 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18–23 June 2018*, Salt Lake City, USA. https://doi.org/10.1109/CVPR.2018.00474

Scikit-learn. Scikit-learn. (27 March 2026; *https://github.com/scikit-learn/scikit-learn).*

Sierra E, Gillespie LE, Soltani S, Exposito-Alonso M, Kattenborn T. 2025. Divshift: Exploring domain-specific distribution shifts in large-scale, volunteer-collected biodiversity datasets. *39th Proceedings of the AAAI Conference on Artificial Intelligence; February 25–March 4, 2025*, Philadelphia, USA. https://doi.org/10.1609/aaai.v39i27.35060

Siméoni O, Vo HV, Seitzer M, Baldassarre F, Oquab M, Jose C, Khalidov V, Szafraniec M, Yi S, Ramamonjisoa M, Massa F, Haziza D, Wehrstedt L, Wang J, Darcet T, Moutakanni T, Sentana L, Roberts C, Vedaldi A, Tolan J, Brandt J, Couprie C, Mairal J, Jégou H, Labatut P, Bojanowski P. 2025. DINOv3. *arXiv*. https://doi.org/10.48550/arXiv.2508.10104

Simpson R, Page KR, De Roure D. 2014. Zooniverse: Observing the world's largest citizen science platform. *WWW'14 Companion: Proceedings of the 23rd International Conference on World Wide Web; 7–11 April 2014*, Seoul, Korea. https://doi.org/10.1145/2567948.2579215

Spiesman BJ, Gratton C, Hatfield RG, Hsu WH, Jepsen S, McCornack B, Patel K, Wang G. 2021. Assessing the potential for deep learning and computer vision to identify bumble bee species from images. *Scientific Reports* 11. https://doi.org/10.1038/s41598-021-87210-1

Stenhouse A, Thrall PH. 2025. SpeciMate: Improving metadata extraction from digitised biological specimens. *Biodiversity Data Journal* 13: e160553. https://doi.org/10.3897/bdj.13.e160553

Stevens S, Wu J, Thompson MJ, Campolongo EG, Song CH, Carlyn DE, Dong L, Dahdul WM, Stewart C, Berger-Wolf T, Chao W-L, Su Y. 2024. Bioclip: A vision foundation model for the tree of life. Seattle, US-AIEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 17–21 June 2024, Seattle, USA. https://doi.org/10.1109/CVPR52733.2024.01836

Stewart RD, Bard NW, van der Bank M, Davies TJ. 2025. Leveraging machine learning and citizen science data to describe flowering phenology across South Africa. *Plants, People, Planet*. https://doi.org/10.1002/ppp3.70059

Suh B, Ling H, Bederson BB, Jacobs DW. 2003. Automatic thumbnail cropping and its effectiveness. *UIST '03: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology, 2–5 November 2003*, Vancouver, Canada. https://doi.org/10.1145/964696.964707

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. 2016. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27–30 June 2016*, Las Vegas, USA. https://doi.org/10.1109/CVPR.2016.308

TensorFlow 1 Detection Model Zoo. Tensorflow Object Detection Model Zoo. Tensorflow. (27 March 2026; *https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md).*

TensorFlow Hub. TensorFlow Hub. (27 March 2026; *https://www.tensorflow.org/hub)*

TensorFlow Object Detection API GitHub. Tensorflow object detection. Tensorflow. (27 March 2026; *https://github.com/tensorflow/models/blob/master/research/object_detection/README.md)*

Tessler M, Little DP. 2025. On herbarium specimen images and artificial intelligence. *New Phytologist*. https://doi.org/10.1111/nph.70312

Tree of Life Dataset. Tree of Life. Hugging Face. (27 March 2026; *https://huggingface.co/datasets/imageomics/TreeOfLife-10 M).*

Turnbull R, Fitzgerald E, Thompson KM, Birch JL. 2025. Hespi: A pipeline for automatically detecting information from herbarium specimen sheets. *BioScience* 75: 637–648. https://doi.org/10.1093/biosci/biaf042

Van Horn G, Aodha O M, Song Y, Cui Y, Sun C, Shepard A, Adam H, Perona P, Belongie S. 2018. The iNaturalist species classification and detection dataset. *IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18–23 June 2018*, Salt Lake City, USA. https://doi.org/10.1109/CVPR.2018.00914

Van Horn G, Branson S, Farrell R, Haber S, Barry J, Ipeirotis P, Perona P, Belongie S. 2015. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7–12 June 2015*, Boston, USA. https://doi.org/10.1109/CVPR.2015.7298658

Wäldchen J, Rzanny M, Seeland M, Mäder P. 2018. Automated plant species identification—Trends and future directions. *PLoS Computational Biology* 14: 1–19. https://doi.org/10.1371/journal.pcbi.1005993

Wang C-Y, Bochkovskiy A, Liao H-YM. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv*. https://doi.org/10.48550/arXiv.2207.02696

Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W. 2016. CNN-RNN: A unified framework for multi-label image classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 27–30 June 2016*, Las Vegas, USA. https://openaccess.thecvf.com/content_cvpr_2016/papers/Wang_CNN-RNN_A_Unified_CVPR_2016_paper.pdf

Weaver WN, Ruhfel BR, Lough KJ, Smith SA. 2023. Herbarium specimen label transcription reimagined with large language models: Capabilities, productivity, and risks. *American Journal of Botany* 110: e16256. https://doi.org/10.1002/ajb2.16256

Weaver WN, Smith SA. 2023. From leaves to labels: Building modular machine learning networks for rapid herbarium specimen analysis with LeafMachine2. *Applications in Plant Sciences* 11: e11548. https://doi.org/10.1002/aps3.11548

Wei H, Sun Y, Li Y. 2025. DeepSeek-OCR: Contexts optical compression. *arXiv*. https://doi.org/10.48550/arXiv.2510.18234

Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viegas F, Wilson J. 2020. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 26: 1–1. https://doi.org/10.1109/TVCG.2019.2934619

Wilson RJ, de Siqueira AF, Brooks SJ, Price BW, Simon LM, van der Walt SJ, Fenberg PB. 2022. Applying computer vision to digitised natural history collections for climate change research: Temperature-size responses in British butterflies. *Methods in Ecology and Evolution* 14: 372–384. https://doi.org/10.1111/2041-210X.13844

Xie G, Xuan J, Luo HR, Liu HY, Renner SS, Li M. 2025. AI-based identification of Chinese vascular plants from herbarium specimens: A tool for all herbaria with Chinese holdings. *New Phytologist*. https://doi.org/10.1111/nph.70443

Zhang Z, Gu J, Chowdhury A, Mai Z, Carlyn D, Berger-Wolf T, Su Y, Chao W-L. 2025. Finer-CAM: Spotting the difference reveals finer details for visual explanation. *arXiv*. https://doi.org/10.48550/arXiv.2501.11309

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. 2016. Learning deep features for discriminative localization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 27–30 June 2016*, Las Vegas, USA. https://doi.org/10.1109/CVPR.2016.319

Received: 8 April 2025. **Revised:** 1 April 2026. **Accepted:** 7 April 2026

© Smithsonian Institution 2026. Published by Oxford University Press on behalf of the American Institute of Biological Sciences. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact reprints@oup.com for reprints and translation rights for reprints. All other permissions can be obtained through our RightsLink service via the Permissions link on the article page on our site-for further information please contact journals.permissions@oup.com