Gesellschaft für Informatik e.V. (GI) (Hrsg.): INFORMATIK 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021

Carpe Diem: A Lifelong Learning Tool for Automated Wildlife Surveillance

Implementing Active and Incremental Learning for Object Detection

Clemens-Alexander Brust[1] Björn Barz[1] Joachim Denzler[1,2]

Abstract: We introduce Carpe Diem, an interactive tool for object detection tasks such as automated wildlife surveillance. It reduces the annotation effort by automatically selecting informative images for annotation, facilitates the annotation process by proposing likely objects and labels, and accelerates the integration of new labels into the deep neural network model by avoiding re-training from scratch.

Carpe Diem implements active learning, which intelligently explores unlabeled data and only selects valuable examples to avoid redundant annotations. This strategy saves expensive human resources. Moreover, incremental learning enables a continually improving model. Whenever new annotations are available, the model can be updated efficiently and quickly, without re-training, and regardless of the amount of accumulated training data. Because there is no single large training step, the model can be used to make predictions at any time. We exploit this in our annotation process, where users only confirm or reject proposals instead of manually drawing bounding boxes.

Keywords: Lifelong Learning; Object Detection; Automated Monitoring

1 Introduction

Large-scale biodiversity studies are an important tool in coordinating a global response to the biodiversity crisis [Car+12; VHS09]. While camera traps are less invasive than conventional capture-mark-recapture approaches [AMM10], the resulting image data still requires analysis. When the scale of camera trap data makes manual analysis infeasible, there are several alternatives. Citizen scientists can be employed to distribute the analysis workload [Swa+15] if the task is reasonably attractive and not too difficult for laypeople. Automatic analysis using machine learning can help overcome the scale limitations. Such methods successfully generalize from smaller amounts of human-annotated data to large-scale processing [Bru+17; Nor+17; GSV17; DKB18; Gir+19].

However, it can still be cost-prohibitive to acquire sufficient labeled training data for machine learning analysis if (expensive) domain expertise is required and funding is inadequate. Active learning [Set09; BKD19] can substantially reduce the number of annotated images

[1] Friedrich Schiller University Jena, Computer Vision Group, Jena, Germany firstname.lastname@uni-jena.de
[2] DLR Institute of Data Science, Jena, Germany firstname.lastname@dlr.de

CC BY SA

required to train a machine learning model with a given accuracy. It intelligently decides which unlabeled images should be annotated by a human, and which should not, eliminating redundancies.

Ideally, active learning is integrated in a feedback loop such that only a very small number of images is selected and annotated at a time. After each batch of annotations, the model is updated quickly with the new data using an incremental learning method [Käd+16a]. Then, the loop repeats. New unlabeled images are selected using active learning and presented to the human annotator. Because active learning considers the model's current knowledge, it selects different images in each iteration. This setup is called lifelong learning [Käd+16b].

In this work, we present a Carpe Diem, GUI that implements the lifelong learning loop. It enables efficient annotation and is a machine learning tool supporting all object detection tasks intended for use by biodiversity researchers. When a model is trained to satisfactory performance, a batch prediction mode can be used to perform analysis on large amounts of images.

2 Active and Incremental Learning Methods

We implement a selection of active learning methods that can be grouped in two major categories [BKD20]. First, a method that is in principle compatible with any object detection system where predicted class probabilities are available for individual detected instances. We evaluate an uncertainty-based active learning metric for classification such as margin sampling or entropy sampling [Set09] on each instance. The resulting set of metrics is then aggregated into a whole-image score by computing the sum, maximum or average.

Second, two heuristics that are specific to the YOLO object detector [Red+16]. One compares compares the confidence scores predicted by the classification part and the "objectness" scores and considers mismatches valuable, as they can result from unknown classes or missed detections. The other applies margin sampling to each YOLO grid cell individually and computes a sum weighted by the respective objectness scores. The user can select from the aforementioned implementations which are shown to improve mAP on their own by up to 5.7% in a representative fast exploration scenario. They change the methods at any time.

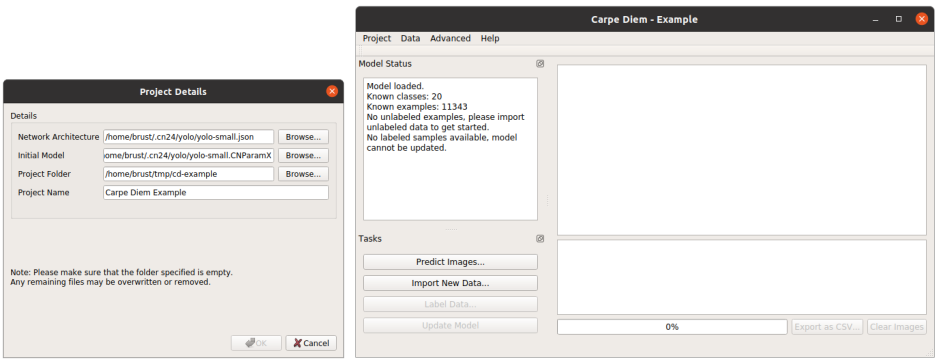
For incremental learning, we implement deep fine-tuning [Käd+16a], which is shown to alleviate catastrophic forgetting. It combines minibatches of previously seen and newly labeled examples in a fixed ratio λ, which defaults to 0.5. We allow the user to configure λ as well as the number of gradient steps per model update to trade off accuracy and runtime.

The combined system, including the weakly supervised annotation scheme proposed in [Pap+16] which further improves annotation time efficiency by a factor of five to nine, is validated in Brust, Käding, and Denzler [BKD20], including a case study in a biodiversity monitoring setting.

3 User Experience

Carpe Diem is built around a lifelong learning workflow. This workflow considers two major characteristics of large-scale automated monitoring projects. First, more data becomes available over time, as new images are transmitted from camera traps or observers. Second, the availability of annotators also changes over time, with funding inconsistencies and volunteers working in their spare time. Lifelong learning adapts to changing environments efficiently with incremental learning, and active learning strategies can intelligently select images with novel or interesting content to save annotation time.

Project Setup and Initial Model Before the lifelong learning process can begin, the user creates a new *project* (see Fig. 1a). A project captures the state of the machine learning component, i. e., the current model parameters, as well as the collected images and annotations. When creating a project, the user can choose from a selection of object detection models, e. g., YOLO [Red+16]. Because the lifelong learning process requires a working initial model, we include weights and the dataset that is used to train the respective model.



(a) Creating a new project. (b) Initial state after creating a new project.

Fig. 1: Project setup and initial state.

Data Management The data stored in a project is divided into labeled and unlabeled data. Initially, only labeled data (from the initial model) is available. As indicated in a status message (see Fig. 1b), the next step is importing unlabeled images, e. g., from camera traps. However, it is also possible to import labeled images and fine-tune the model at any time. All data stored in the project can be reviewed and modified at all times (see Fig. 2), e. g., to remove faulty annotations. The user can rename, remove and add object classes known to the model.

Selection, Annotation and Model Update The lifelong learning cycle runs for as long as the project contains unlabeled images. Whenever the user is ready to annotate a small batch of images, they press the "Label Data..." button. Carpe Diem processes all unlabeled images and selects a user-configurable number of valuable examples. The selection is performed according to the active learning strategy chosen by the user. We implement several active learning methods as described in Sect. 2.

This batch is then presented to the user for annotation. There is no need for manual drawing of bounding boxes. Instead, the user only confirms, modifies, or rejects proposals from the object detection model (see Fig. 3a). This further increases annotation efficiency [Pap+16].

After each annotated batch, the model has to be updated to ensure a tight feedback loop. This results in fewer redundant annotations and a faster increase in accuracy. We use the incremental learning technique presented in Käding et al. [Käd+16a] to perform the model update quickly. Model snapshots can be stored to preserve prior states and revert to them in the event of errors, e. g., divergence during optimization.

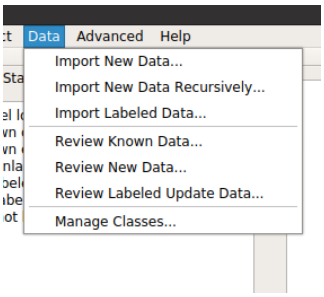
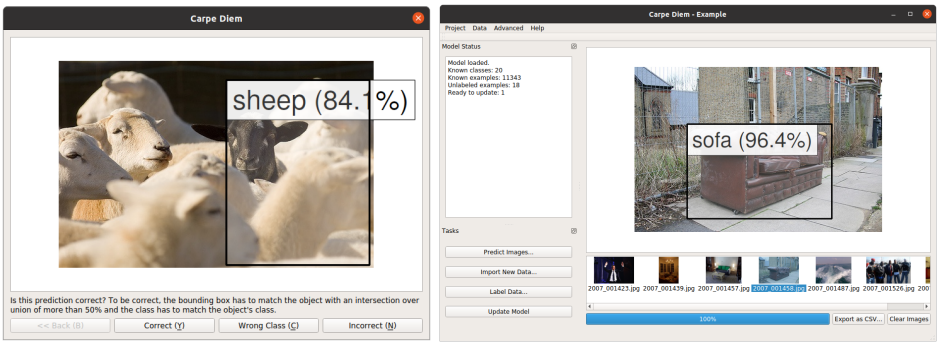


Fig. 2: Menu for data management.



(a) The annotation process with keyboard shortcuts. (b) Browsing predictions in the thumbnail gallery.

Fig. 3: Annotation and prediction.

Predictions Only a small fraction of the unlabeled images should ever be annotated. The main purpose of annotating data and training a model is to use its generalization capability for an analysis of the large-scale unlabeled data. After validating the model performance using a held-out hand-annotated validation set, the model can be used to make predictions. Carpe Diem can process very large batches of images for prediction and export the detected objects in comma-separated value (CSV) format. Before export, the user can browse the predictions in a gallery (see Fig. 3b).

4 Interoperability and System Requirements

Carpe Diem is built on the CN24 deep learning framework [Bru+15]. The framework is binary compatible with darknet[3] models, e. g., Extraction and YOLO [Red+16]. Custom architectures can be specified in a human-readable JSON format. The same is true for image datasets. If needed, a user can easily construct and train an initial model on their own annotated data in CN24 and import it into Carpe Diem for lifelong learning.

Predictions are exported in CSV format for further analysis, with one line representing a single detected object. Each prediction includes the object class, a confidence score, and the coordinates of an axis-aligned bounding box around the object.

Carpe Diem is tested on Windows 10 and Linux (Ubuntu 18.04 LTS, 20.04 LTS) operating systems. It mainly depends on CN24 (which in turn requires OpenCL and clBLAS) and Qt 5. We recommend using a GPU that implements OpenCL 1.2 and is equipped with at least 4 GiB of VRAM. Carpe Diem supports both AMD and NVIDIA GPUs. Training the object detectors, even on the YOLO-Small variant, is not practical on CPUs only.

5 Conclusion

We present Carpe Diem, a fully integrated lifelong learning experience for object detection tasks. It is capable of adapting to changing environments and acquires annotations as efficiently as possible. The underlying methods are empirically validated individually [Käd+16a; BKD19; Pap+16] and as a complete system in a wildlife monitoring setting [BKD20]. The software will be made available publicly under a three-clause BSD license upon formal publication.

Future Work Carpe Diem is undergoing constant improvement. Current development efforts include a client-server separation layer to enable multi-user annotation scenarios. Further, we are enabling support for TensorFlow [Aba+15] models. Plans for the future include alternative annotation methods such as extreme clicking [Pap+17] and methods for integrating domain knowledge [BD19] to further increase label efficiency.

[3] `https://pjreddie.com/darknet/`

References

[Aba+15] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[AMM10] Steven C Amstrup, Trent L McDonald, and Bryan FJ Manly. *Handbook of Capture-recapture Analysis*. Princeton University Press, 2010. ISBN: 978-0-691-08967-6.

[BD19] Clemens-Alexander Brust and Joachim Denzler. "Integrating Domain Knowledge: Using Hierarchies to Improve Deep Classifiers". In: *Asian Conference on Pattern Recognition (ACPR)*. 2019. DOI: 10.1007/978-3-030-41404-7_1.

[BKD19] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. "Active Learning for Deep Object Detection". In: *Computer Vision Theory and Applications (VISAPP)*. 2019. DOI: 10.5220/0007248601810190.

[BKD20] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. "Active and Incremental Learning with Weak Supervision". In: *Künstliche Intelligenz* 34 (2 2020), pp. 165–180. DOI: 10.1007/s13218-020-00631-4.

[Bru+15] Clemens-Alexander Brust et al. "Convolutional Patch Networks with Spatial Prior for Road Detection and Urban Scene Understanding". In: *Computer Vision Theory and Applications (VISAPP)*. 2015. DOI: 10.5220/0005355105100517.

[Bru+17] Clemens-Alexander Brust et al. "Towards Automated Visual Monitoring of Individual Gorillas in the Wild". In: *International Conference on Computer Vision Workshops (ICCV-WS)*. 2017. DOI: 10.1109/ICCVW.2017.333.

[Car+12] Bradley J Cardinale et al. "Biodiversity Loss and Its Impact on Humanity". In: *Nature* 486.7401 (2012), pp. 59–67. DOI: 10.1038/nature11148.

[DKB18] Joachim Denzler, Christoph Käding, and Clemens-Alexander Brust. "Keeping the Human in the Loop: Towards Automatic Visual Monitoring in Biodiversity Research". In: *International Conference on Ecological Informatics (ICEI)*. 2018.

[Gir+19] Jhony-Heriberto Giraldo-Zuluaga et al. "Camera-trap Images Segmentation Using Multi-layer Robust Principal Component Analysis". In: *The Visual Computer* 35 (3 2019), pp. 335–347. DOI: 10.1007/s00371-017-1463-9.

[GSV17] Alexander Gomez Villa, Augusto Salazar, and Francisco Vargas. "Towards Automatic Wild Animal Monitoring: Identification of Animal Species in Camera-trap Images Using Very Deep Convolutional Neural Networks". In: *Ecological Informatics* 41 (2017), pp. 24–32. DOI: 10.1016/j.ecoinf.2017.07.004.

[Käd+16a] Christoph Käding et al. “Fine-tuning Deep Neural Networks in Continuous Learning Scenarios”. In: *Asian Conference on Computer Vision Workshops (ACCV-WS)*. 2016. DOI: 10.1007/978-3-319-54526-4_43.

[Käd+16b] Christoph Käding et al. “Watch, Ask, Learn, and Improve: a lifelong learning cycle for visual recognition.” In: *European Symposium on Artificial Neural Networks (ESANN)*. 2016.

[Nor+17] Mohammed Sadegh Norouzzadeh et al. “Automatically Identifying, Counting, and Describing Wild Animals in Camera-trap Images with Deep Learning”. In: (2017). arXiv: 1703.05830v5 [cs.CV].

[Pap+16] Dim P. Papadopoulos et al. “We Don’t Need No Bounding-boxes: Training Object Class Detectors Using Only Human Verification”. In: *Computer Vision and Pattern Recognition (CVPR)*. June 2016. DOI: 10.1109/CVPR.2016.99.

[Pap+17] Dim P Papadopoulos et al. “Extreme clicking for efficient object annotation”. In: *International Conference on Computer Vision (ICCV)*. 2017.

[Red+16] Joseph Redmon et al. “You Only Look Once: Unified, Real-time Object Detection”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2016. DOI: 10.1109/CVPR.2016.91.

[Set09] Burr Settles. *Active Learning Literature Survey*. Tech. rep. University of Wisconsin-Madison, 2009.

[Swa+15] Alexandra Swanson et al. “Snapshot Serengeti, High-frequency Annotated Camera Trap Images of 40 Mammalian Species in an African Savanna”. In: *Scientific Data* 2 (1 2015). DOI: 10.1038/sdata.2015.26.

[VHS09] Jean-Christophe Vié, Craig Hilton-Taylor, and Simon N. Stuart, eds. *Wildlife in a Changing World: An Analysis of the 2008 Iucn Red List of Threatened Species*. International Union for Conservation of Nature and Natural Resource, 2009. ISBN: 978-84-96553-63-7.