# *OSEA*, a deep learning-based bird classification tool, with pre-trained model, mobile and command line applications

**Running title: *OSEA*, deep learning-based bird classification**

**Jiao Sun**[123*]
[1] CAS Key Laboratory of Plant Germplasm Enhancement and Specialty Agriculture, Wuhan Botanical Garden, Chinese Academy of Sciences, Wuhan 430074, China;
[2] Center of Conservation Biology, Core Botanical Gardens, Chinese Academy of Sciences, Wuhan 430074, China;
[3] University of Chinese Academy of Sciences, Beijing 100049, China.
* Correspondence: sunjiao19@mails.ucas.ac.cn
ORCID: https://orcid.org/0000-0002-5028-8132

## Abstract

In response to the challenges of traditional biodiversity monitoring methods, we introduce OSEA (Open Species Estimation for Avians), a multi-platform, offline tool for bird species identification. Designed to recognize over 10,000 bird species, OSEA includes both a mobile application and a command-line interface (CLI), facilitating efficient bird species identification. The mobile app, developed using Flutter, offers cross-platform compatibility and integrates a pre-trained ResNet34 model. The CLI, suited for batch processing, allows users to process images offline, with geographic filtering capabilities for enhanced accuracy.

OSEA utilizes the DongNiao International Birds 10000 Dataset (DIB-10K), which undergoes rigorous data cleaning to ensure quality and accuracy. The system's core model leverages the MetaFGNet architecture, trained on high-performance computing resources to achieve 90.8% accuracy on the training set and 87.6% accuracy on the validation set. Additionally, the mobile app and CLI incorporate species distribution data for efficient geographical filter.

OSEA addresses significant challenges in biodiversity research, including the time-intensive nature of manual species identification and the limited availability of offline tools for large-scale image analysis. By offering an accessible, offline solution, OSEA empowers amateur birders, educators, and conservationists, particularly in regions with limited internet access. Furthermore, the tool's compatibility with custom models allows flexibility for broader wildlife applications beyond birds.

In conclusion, OSEA offers a practical, scalable, and user-friendly solution for bird species identification, contributing to the acceleration of biodiversity studies and conservation efforts. Future developments will focus on expanding the dataset, optimizing performance, and incorporating more underrepresented species, further enhancing the tool's robustness and global applicability.

**Key words**: bird identification, biodiversity, command-line interface, deep learning, image classification, mobile application.

## Introduction

In recent years, traditional biodiversity survey methods have undergone significant transformation, largely due to the widespread adoption of camera traps and handheld cameras, as well as the increasing popularity of citizen science. These tools have led to an unprecedented surge in the availability of image data of various species, which has proven invaluable for studying and protecting biodiversity (Mesaglio et al., 2023; Mikula et al., 2018; Newcomer et al., 2019). Traditionally, the analysis of such image data has relied heavily on manual identification, a process that is not only time-consuming but also labour-intensive. This extensive requirement for human intervention could prolong the research cycle (Schneider et al., 2019).

In recent years, the development of machine learning techniques, particularly deep convolutional neural networks (CNNs), has revolutionized the field of computer vision. CNNs, popularized by LeCun et al. (1989), have been widely adopted due to their ability to automatically learn hierarchical feature representations from raw data, making them well-suited for tasks such as image recognition (Zhao et al., 2024). Recent studies have shown that deep learning-based computer vision technologies have significant potential to enhance the efficiency of image-based biodiversity surveys. For instance, methods using CNNs for automatically identifying animals in camera trap images have outperformed traditional approaches, greatly reducing the time required for analysis (Norouzzadeh et al., 2018, 2021; Schneider et al., 2019).

However, the application of these advanced methods often comes with challenges. Large-scale data collection, annotation, and the need for expensive training datasets remain prerequisites for developing robust deep learning models. This requirement can be a barrier for researchers from developing countries or independent researchers, limiting their access to cutting-edge technologies (Khan et al., 2024). Moreover, the repeated training of similar visual models across different research projects can lead to significant resource consumption, as noted by Strubell et al., (2019) in the field of natural language processing.

Despite the availability of some online services for wildlife image analysis, these tools often fall short of addressing the practical needs of researchers. A major limitation is the lack of accessible automated APIs, which forces users to manually process images via graphical user interfaces (GUIs). This makes the software suitable only for small-scale, personal use and impractical for large-scale, batch processing. Furthermore, the use of these services in regions with limited communication infrastructure can be hindered.

To address these challenges, we present OSEA: Open Species Estimation for Avians. OSEA is an offline project designed to identify over 10,000 species of birds, following the International Ornithological Congress (IOC) 10.1 taxonomy (Gill et al., 2021). The project includes both a mobile application with a user-friendly GUI and a command-line interface (CLI) tool for batch processing, both of which operate entirely offline. A pre-trained deep learning model built on ResNet34 (He et al., 2016) was embedded within. This solution is not only suitable for biodiversity analysis and transfer learning but also provides birdwatchers with a reliable tool to identify birds in their personal photos, ensuring broad accessibility and utility. Additionally, OSEA can be effectively utilized by law enforcement personnel for identifying birds seized from illegal poaching activities, offering a practical solution in wildlife conservation and protection efforts. For researchers who want to train or have trained their own models, our project will be compatible with other models with little or no modification.

## Material and methods

### *Data processing*

In this study, we utilized the DongNiao International Birds 10000 Dataset (DIB-10K), which comprises over 4.8 million images representing 10,922 bird species, following the IOC 10.1 taxonomy. This extensive dataset encompasses a wide variety of bird species, morphological variants, postures, and gestures (Mei & Dong, 2020). Despite the authors' efforts in manual review and correction, the dataset contains numerous duplicate and erroneous images, necessitating a comprehensive data cleaning process.

For the deduplication work, we employed a perceptual hashing algorithm, specifically the pHash method, which generates hash values representing the visual content of images. This technique enables the detection of near-duplicate images by comparing these hashes (Farid, 2021). Initially, we conducted intraclass analysis by processing each species-specific directory independently, identifying duplicates within the same class, and retaining one representative image from each set. Subsequently, we performed interclass duplicate detection across the entire dataset to identify duplicates spanning different classes, removing all interclass duplicate images to maintain dataset integrity.

To address erroneous images with non-avian subjects, we utilized Faster Region-based Convolutional Neural Network (Faster R-CNN) (Ren et al., 2016), a robust object detection model available in PyTorch's torchvision library with pre-trained weights (Ansel et al., 2024). For images where no birds were detected, we implemented a two-tiered approach: images from high-volume classes (⩾200 images) were automatically deleted to avoid heavy labour, while those from low-volume classes (<200 images) were moved to a separate directory for manual inspection. After a thorough review, valid images were reintegrated into the dataset.

Additionally, we wrote scripts to remove corrupted and unreadable images and fix images with incorrect extension names. Following the aforementioned procedures, the dataset was refined to 4,706,520 images representing 10,916 species. We then randomly partitioned the dataset into training and validation sets with a 9:1 ratio for subsequent model training.

Recognizing the challenges in distinguishing among species complexes, we incorporated avian geographical distribution data from AVONET to enhance classification accuracy (Tobias et al., 2022). By filtering results based on the geographical location where an image was captured, we limited species to those present in the specified and neighbouring regions. To facilitate this, we used a Python script to export data from shapefiles and CSV files into an SQLite database, ensuring compatibility with both mobile and CLI applications (Gaffney et al., 2022). To match species between the AVONET and DIB datasets, we developed a Python script to align scientific names. This script systematically converted AVONET scientific names to their corresponding IOC names according to the "IOC with other lists" Excel file from the IOC website (Gill et al., 2021). This conversion facilitated accurate matching with the DIB species list, ensuring the applicability of the geographic filter.

*Model training*

The task of recognizing over 10,000 bird species is inherently challenging, as it involves distinguishing between closely related species that fall within the same broader category. This is a fine-grained visual categorization (FGVC) problem, where the goal is to identify minor differences between highly similar categories. To tackle this challenge, we adopted the MetaFGNet model proposed by Zhang et al. (2018), which is specifically designed for FGVC tasks. We utilized their pre-trained model “LBird-31_checkpoint.pth” as the foundation for transfer learning on the processed DIB-10K dataset.

For the training process, we rented a server equipped with an RTX 4090D GPU, 90GB of memory, and a 15-core Intel(R) Xeon(R) Platinum 8474C processor from the AutoDL platform. This high-performance setup allowed for efficient processing and model training. The training procedure was conducted for 32 epochs, with the following command to execute the training:

```
python main.py --batch_size 256 --momentum 0.9 --weight_decay 1e-4
--data_path /root/autodl-tmp/dib/ --dataset dongniao \
             --pretrain --freeze --print_freq 1 --epochs 32 --schedule
15 23 --newfc --numclass_old 10320 --numclass_new 11000  --workers 64 \
             --pretrained_model
/root/autodl-tmp/LBird-31_checkpoint.pth.tar --resume
/root/autodl-tmp/LBird-31_checkpoint.pth.tar
```

*Mobile Application*

The mobile application was developed using Flutter, a cross-platform framework by Google that enables deployment on Android and iOS platforms. To facilitate efficient on-device inference, the trained model was converted into the Open Neural Network Exchange (ONNX) format and subsequently quantized. This procedure reduces the model's size and computational requirements, making it more suitable for mobile devices. The ONNX model is executed on the device using ONNX Runtime, which provides a high-performance engine for running machine learning models across various platforms (ONNX Runtime Team, 2021).

For object detection tasks, such as identifying and localizing birds within an image, the application employs a pre-trained Single Shot MultiBox Detector (SSD) with a MobileNet backbone, provided in the ONNX Model Zoo (Liu et al., 2016). This model efficiently detects objects in images, allowing the application to crop the detected bird regions for further classification. Users also have the option to crop images if desired manually.

When the user enables geographic filtering, the application cross-references the inference results to include only the bird species known to inhabit the specified location. This step enhances the accuracy of species identification by narrowing down the potential species based on location. Subsequently, the application applies the softmax function to the filtered or unfiltered inference results, converting the model's output logits into probabilities. The top three species with probabilities exceeding 1% are then presented to the user. Depending on user preferences, the application can display the scientific names, common names, or both for the identified species.

*Command-Line Interface (CLI) Software*

The command-line application utilizes PyTorch's built-in pre-trained Faster R-CNN model for object detection (Ansel et al., 2024; Ren et al., 2016). The software automatically crops the detected regions for subsequent classification. The cropped images are then processed using the previously trained classification model to determine the bird species present. If the user provides specific geographic coordinates, the software filters the results as in mobile applications. In the same way as above, the softmax function is applied here as well.

Results with probabilities equal to or exceeding the user-specified threshold are directly written to the output file, indicating high-confidence identifications. For results below this confidence threshold, the corresponding images and their inferred species are saved to a separate file for user review and manual verification. This approach ensures that high-confidence identifications are efficiently processed, while lower-confidence cases receive additional scrutiny to maintain overall accuracy.

## Result

The model achieved impressive performance with 90.8% accuracy on the training set and 87.6% accuracy on the validation set, alongside a top-5 accuracy of 94.2%. The following examples illustrated how to use the model for analysis in the mobile and command line application.

*Mobile example*

Assuming the user has taken a photograph containing one or more birds, they can transfer the image to their mobile device using the camera manufacturer's application or by recapturing the screen with the smartphone's camera.

Opening the application is the first step to identifying the birds in the photo. If the image is already on the device, they tap the "gallery" icon to select the picture; otherwise, they can click the "camera" icon to recapture it. Once an image is selected, the app automatically performs bird object detection and identification. If multiple bird objects are detected, navigation buttons allow the user to switch between different birds. In cases where the object detection model fails to detect a bird, the user has the option to manually crop the image to focus on the area of interest.

For location-based filtering, the user can choose to automatically filter species based on their current geographical location or manually select a location on a map.

This process ensures that users can efficiently identify bird species in their photos, enhancing their bird survey work or birding experience.

*Command-Line Interface (CLI)* example

Assuming the user has photos taken by a camera trap, located in the folder "Documents/trap/camera_1," the user can run the OSEA CLI tool with the following command:

```
python osea.py --input_folder "Documents/trap/camera_1" --output_folder "Documents/trap/camera_1_out" --model_path "model20240824.pth" --class_number 11000 --db_path "avonet.db"
```

The analysis results will be saved in the specified output folder, "Documents/trap/camera_1_out," which contains a subfolder with images annotated with bounding boxes and species names. Additionally, a CSV file will be generated, where each row records the recognition result for a photo in the format: "image filename, species name, confidence score."

The OSEA CLI tool provides several configurable arguments to meet users' needs:

- --input_folder: This argument specifies the path to the folder containing the bird images to be processed. It is a required field.
- --output_folder: This defines the path to the folder where the annotated images and results will be stored. It is also required.
- --model_path: The path to the .pth model file that is used for classification. If not specified, the default model will be used.
- --location: (Optional) This argument allows users to specify geographical coordinates (latitude and longitude) for species filtering based on their location. If not provided, no location-based filtering will be applied.
- --classification_model: (Optional) The name of the classification model used for identifying species. The model name must exactly match that in the Pytorch model file.
- --label_path: (Optional) The path to the label file, which contains the list of species names. If not provided, the default labels will be used.
- --class_number: (Optional) This parameter defines the number of species to be classified by the model. It must match the number of classes in the pre-trained model.
- --detection_model: (Optional) The name of the object detection model used to locate birds in the images. As with the classification model, it must match exactly the model name in Pytorch.
- --detection_model_path: (Optional) The path to the .pth file of the object detection model. This allows users to specify custom detection models.
- --detection_class_number: (Optional) This defines the number of classes for the detection model, specifying how many categories the detection model will identify.
- --db_path: (Optional) The path to the distribution database file (e.g., AVONET database). If this parameter and the below one are both omitted, no filtering will be applied.
- --species_list: (Optional) The path to the manually specified species list. It contains a file with one category label in each line. If this parameter and the below one are both omitted, no filtering will be applied.
- --classification_threshold: (Optional, Default = 0.85) This threshold determines the minimum confidence level required for species classification. Results with lower confidence will not be considered.
- --detection_threshold: (Optional, Default = 0.85) This threshold sets the confidence required for object detection. Detections with lower confidence will be ignored.
- --detection_target: (Optional) This argument allows users to specify a target class for object detection, which can be useful for focusing on specific species or groups of birds.

The CLI tool is highly customizable, including work with other animal groups, like mammals. In that case, the user could provide their own custom detection model and classification model, as well as any necessary weights and label maps. This flexibility allows the CLI tool to be compatible with various object detection and image recognition models, broadening its utility beyond just bird species identification.

## Discussion

In this study, we have introduced OSEA: Open Species Estimation for Avians, a multi-platform tool designed to facilitate the identification of over 10,000 bird species worldwide. OSEA offers both a mobile application and a command-line interface (CLI), enabling users to perform offline bird species identification efficiently. The mobile application, developed using Flutter, ensures cross-platform compatibility, while the CLI provides a robust solution for batch processing of images.

One of the primary advantages of OSEA is its accessibility to users with different backgrounds. By providing a user-friendly interface and pre-trained models, individuals without a background in machine learning can effectively utilize the tool for bird identification. This democratization of technology empowers amateur birders and educators to engage in biodiversity studies.

The significance of OSEA extends beyond individual use; it addresses critical challenges in biodiversity monitoring. Traditional methods of species identification are often time-consuming and require expert knowledge, which can delay conservation efforts. By automating the identification process, OSEA accelerates data analysis. Moreover, the offline functionality ensures that users in regions with limited internet connectivity can still benefit from the tool, promoting inclusivity in global biodiversity studies.

Future development of OSEA will focus on several key areas. Expanding and updating the dataset through further data collection and processing to correspond to more recent taxonomic changes. Model training based on updated datasets will enhance the accuracy of species identification. Incorporating additional images, particularly for underrepresented species in the current dataset, will improve the model's robustness and applicability across different ecological contexts. Optimizing application performance is crucial to ensure a seamless user experience. This includes reducing the computational load to facilitate faster inference times on mobile devices.

In conclusion, OSEA represents a significant advancement in the open tools available for bird species identification. By combining advanced machine learning models with user-centric design, it bridges the gap between complex technology and practical application, making a meaningful contribution to global biodiversity monitoring efforts.

## Abbreviations

CLI: command-line interface

CNN: convolutional neural network
DIB: DongNiao International Birds 10000 Dataset
Faster R-CNN: Faster Region-based Convolutional Neural Network
FGVC: fine-grained visual categorization
GUI: graphical user interface
IOC: International Ornithological Congress
ONNX: Open Neural Network Exchange
SSD: Single Shot MultiBox Detector

## Code accessibility

Code for data processing and CLI software: https://github.com/sun-jiao/osea

Code for mobile application: https://github.com/sun-jiao/osea_mobile

The modified MetaFGNet code to adapt to the data structure of DIB-10K: https://github.com/sun-jiao/MetaFGNet

Some binary files are not included in git, they can be found in the release section.



## Acknowledgement

We appreciate Yabin Zhang for permitting us to use their pre-trained weights.

## Non-scientific notes

The formal name "Open Species Estimation for Avians" was intentionally coined as a backronym of *osea*, the specific epithet of Palestine's national bird, *Cinnyris osea*. This epithet was derived from the Ancient Greek word ὅσια (hosia) that means "holy," referring to Palestine, the Holy Land.

## Conflict of interests

The author has no conflict of interests.

## Author contribution statements

JS designed the project and created the mobile and command-line application.

## References


Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., … Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 929–947. https://doi.org/10.1145/3620665.3640366

Farid, H. (2021). An Overview of Perceptual Hashing. *Journal of Online Trust and Safety*, *1*(1), Article 1. https://doi.org/10.54501/jots.v1i1.24

Gaffney, K. P., Prammer, M., Brasfield, L., Hipp, D. R., Kennedy, D., & Patel, J. M. (2022). SQLite: Past, present, and future. *Proc. VLDB Endow.*, *15*(12), 3535–3547. https://doi.org/10.14778/3554821.3554842

Gill, F., Donsker, D., & Rasmussen, P. (2021). IOC world bird list. *International Ornithologists' Union*, *10.1*. https://doi.org/10.14344/IOC.ML.10.1

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Khan, M. S., Umer, H., & Faruqe, F. (2024). Artificial intelligence for low income countries. *Humanities and Social Sciences Communications*, *11*(1), 1–13. https://doi.org/10.1057/s41599-024-03947-w

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541 – 551. Neural Computation. https://doi.org/10.1162/neco.1989.1.4.541

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21 – 37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Mei, J., & Dong, H. (2020). *The DongNiao International Birds 10000 Dataset* (No. arXiv:2010.06454; Version 2). arXiv. https://doi.org/10.48550/arXiv.2010.06454

Mesaglio, T., Sauquet, H., Coleman, D., Wenk, E., & Cornwell, W. K. (2023). Photographs as an essential biodiversity resource: Drivers of gaps in the vascular plant photographic record. *New Phytologist*, *238*(4), 1685 – 1694. https://doi.org/10.1111/nph.18813

Mikula, P., Hadrava, J., Albrecht, T., & Tryjanowski, P. (2018). Large-scale assessment of commensalistic-mutualistic associations between African birds and herbivorous mammals using internet photos. *PeerJ*, *6*, e4520. https://doi.org/10.7717/peerj.4520

Newcomer, K., Tracy, B. M., Chang, A. L., & Ruiz, G. M. (2019). Evaluating Performance of Photographs for Marine Citizen Science Applications. *Frontiers in Marine Science*, *6*. https://doi.org/10.3389/fmars.2019.00336

Norouzzadeh, M. S., Morris, D., Beery, S., Joshi, N., Jojic, N., & Clune, J. (2021). A deep active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, *12*(1), 150 – 161. https://doi.org/10.1111/2041-210X.13504

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, *115*(25), E5716 – E5725. https://doi.org/10.1073/pnas.1719367115

ONNX Runtime Team. (2021). *ONNX Runtime*. https://onnxruntime.ai/

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137 – 1149. https://doi.org/10.1109/TPAMI.2016.2577031

Schneider, S., Taylor, G. W., Linquist, S., & Kremer, S. C. (2019). Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods in Ecology and Evolution*, *10*(4), 461 – 470. https://doi.org/10.1111/2041-210X.13133

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645 – 3650). Association for Computational Linguistics.

https://doi.org/10.18653/v1/P19-1355

Tobias, J. A., Sheard, C., Pigot, A. L., Devenish, A. J. M., Yang, J., Sayol, F., Neate-Clegg, M. H. C., Alioravainen, N., Weeks, T. L., Barber, R. A., Walkden, P. A., MacGregor, H. E. A., Jones, S. E. I., Vincent, C., Phillips, A. G., Marples, N. M., Montaño-Centellas, F. A., Leandro-Silva, V., Claramunt, S., … Schleuning, M. (2022). AVONET: Morphological, ecological and geographical data for all birds. *Ecology Letters*, *25*(3), 581 – 597. https://doi.org/10.1111/ele.13898

Zhang, Y., Tang, H., & Jia, K. (2018). Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. *Proceedings of the European Conference on Computer Vision (ECCV)*, 233 – 248. http://openaccess.thecvf.com/content_ECCV_2018/html/Yabin_Zhang_Fine-Grained_Visual_Categorization_ECCV_2018_paper.html

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*(4), 99. https://doi.org/10.1007/s10462-024-10721-6