Package 'birdnetR'

July 22, 2025

Type Package

Title Deep Learning for Automated (Bird) Sound Identification

Version 0.3.2

Description
Use 'BirdNET', a state-of-the-art deep learning classifier, to automatically identify (bird) sounds.
Analyze bioacoustic datasets without any computer science background using a pre-trained model or a custom trained classifier.
Predict bird species occurrence based on location and week of the year.
Kahl, S., Wood, C. M., Eibl, M., & Klinck, H. (2021) <doi:10.1016/j.ecoinf.2021.101236>.

License MIT + file LICENSE

URL https://birdnet-team.github.io/birdnetR/,
https://github.com/birdnet-team/birdnetR

Depends R (>= 4.0)

Imports reticulate (>= 1.41)

Suggests arrow, curl, devtools, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

BugReports https://github.com/birdnet-team/birdnetR/issues

NeedsCompilation no

Author Felix Günther [cre],
Stefan Kahl [aut, cph],
BirdNET Team [aut]

Maintainer Felix Günther <felix.guenther@informatik.tu-chemnitz.de>

Repository CRAN

Date/Publication 2025-04-30 11:50:06 UTC

Contents

`available_languages` *Get Available Languages for BirdNET Model*

Description

Retrieve the available languages supported by a specific version of BirdNET.

Usage

```
available_languages(version)
```

Arguments

`version`	character. The version of BirdNET to use (default is "v2.4", no other versions are currently supported).

Value

A sorted character vector containing the available language codes.

Examples

```
## Not run: available_languages("v2.4")
```

`birdnet_model_load` *Initialize a BirdNET Model*

Description

The various function of the `birdnet_model_*` family are used to create and initialize diffent BirdNET models. Models will be downloaded if necessary.

- `birdnet_model_tflite()`: creates a tflite-model used for species prediction from audio.
- `birdnet_model_custom()`: loads a custom model for species prediction from audio.
- `birdnet_model_protobuf()`: creates a protobuf model for species prediction from audio that can be run on the GPU (GPU support so far only implemented on Apple Silicon).
- `birdnet_model_meta()`: creates a meta model for species prediction from location and time.

Usage

```
birdnet_model_tflite(
  version = "v2.4",
  language = "en_us",
  tflite_num_threads = NULL
)

birdnet_model_custom(
  version = "v2.4",
  classifier_folder,
  classifier_name,
  tflite_num_threads = NULL
)

birdnet_model_meta(
  version = "v2.4",
  language = "en_us",
  tflite_num_threads = NULL
)

birdnet_model_protobuf(
  version = "v2.4",
  language = "en_us",
  custom_device = NULL
)
```

Arguments

`version`	character. The version of BirdNET to use (default is "v2.4", no other versions are currently supported).

`language` character. Specifies the language code to use for the model's text processing. The language must be one of the available languages supported by the BirdNET model.

`tflite_num_threads`
integer. The number of threads to use for TensorFlow Lite operations. If NULL (default), the default threading behavior will be used. Will be coerced to an integer if possible.

`classifier_folder`
character. Path to the folder containing the custom classifier.

`classifier_name`
character. Name of the custom classifier.

`custom_device` character. This parameter allows specifying a custom device on which computations should be performed. If `custom_device` is not specified (i.e., it has the default value None), the program will attempt to use a GPU (e.g., "/device:GPU:0") by default. If no GPU is available, it will fall back to using the CPU. By specifying a device string such as "/device:GPU:0" or "/device:CPU:0", the user can explicitly choose the device on which operations should be executed.

Details

Species Prediction from audio

Models created from `birdnet_model_tflite()`, `birdnet_model_custom()`, and `birdnet_model_protobuf()` can be used to predict species within an audio file using `predict_species_from_audio_file()`.

Species prediction from location and time

The `birdnet_model_meta()` model can be used to predict species occurrence at a specific location and time of the year using `predict_species_at_location_and_time()`.

Value

A BirdNET model object, which is an S3 object of class `birdnet_model` and specific subclasses (e.g., `birdnet_model_tflite`, `birdnet_model_v2_4`). This object is a list containing:

`py_model` The underlying Python BirdNET model object.

`model_version` The version string of the model (e.g., "v2.4").

... Additional elements specific to the model type:

- **For** `birdnet_model_tflite` **and** `birdnet_model_meta`**:**
  - `language`: The language code used (e.g., "en_us").
  - `tflite_num_threads`: The number of TFLite threads used (or `NULL`).
- **For** `birdnet_model_custom`**:**
  - `classifier_folder`: Path to the custom classifier folder.
  - `classifier_name`: Name of the custom classifier.
  - `tflite_num_threads`: The number of TFLite threads used (or `NULL`).
- **For** `birdnet_model_protobuf`**:**
  - `language`: The language code used (e.g., "en_us").
  - `custom_device`: The custom device specified (or `NULL`).

Note

Currently, all models can only be executed on the CPU. GPU support is only available on Apple Silicon.

See Also

available_languages() predict_species_from_audio_file() predict_species_at_location_and_time()

Examples

```
# Create a TFLite BirdNET model with 2 threads and English (US) language
## Not run:
birdnet_model <- birdnet_model_tflite(version = "v2.4", language = "en_us", tflite_num_threads = 2)

## End(Not run)
```

---

get_top_prediction *Get the top prediction by confidence within time intervals*

Description

This convenience function retrieves the row(s) with the highest confidence value within each time interval. It can also limit the results to a specific time interval if specified.

Usage

```
get_top_prediction(data, filter = NULL)
```

Arguments

data	A data frame with columns 'start', 'end', 'scientific_name', 'common_name', and 'confidence'. This data frame is typically the output from predictions_to_df.
filter	A list containing 'start' and 'end' values to filter the data before calculation. If NULL, the function processes all time intervals.

Value

A data frame containing the rows with the highest confidence per group or for the specified interval.

Examples

```
# Example data
data <- data.frame(
  start = c(0, 0, 1, 1, 2, 2),
  end = c(1, 1, 2, 2, 3, 3),
  scientific_name = c(
    "Species A",
    "Species B",
```

```
    "Species A",
    "Species B",
    "Species A",
    "Species B"
  ),
  common_name = c(
    "Common A",
    "Common B",
    "Common A",
    "Common B",
    "Common A",
    "Common B"
  ),
  confidence = c(0.1, 0.2, 0.5, 0.3, 0.7, 0.8)
)
data

# Get top prediction for each time interval
get_top_prediction(data)

# Get top prediction for a specific time interval
get_top_prediction(data, filter = list(start = 1, end = 2))

# The same thing can be done using dplyr
## Not run:
 data |>
    dplyr::group_by(start, end) |>
    dplyr::slice_max(order_by = confidence)

## End(Not run)
```

`install_arrow` *Install Apache Arrow*

Description

This helper function installs Apache Arrow for both R and Python.

Usage

```
install_arrow()
```

Value

Invisible TRUE if successful, stops with error message if installation fails

Examples

```
## Not run: install_arrow()
```

labels_path *Get Path to a Labels File*

Description

This function retrieves the file path to the BirdNET labels file on your system corresponding to a specified language. This file contains all class labels supported by the BirdNET model.

For a custom model, the path of the custom labels file is returned.

Usage

```
labels_path(model, ...)

## S3 method for class 'birdnet_model_custom'
labels_path(model, ...)

## S3 method for class 'birdnet_model_tflite'
labels_path(model, language, ...)

## S3 method for class 'birdnet_model_protobuf'
labels_path(model, language, ...)
```

Arguments

model	A BirdNET model object.
...	Additional arguments passed to the method dispatch function.
language	character. Specifies the language code for which the labels path is returned. The language must be one of the available languages supported by the BirdNET model.

Value

A character string representing the file path to the labels file for the specified language.

Note

The language parameter must be one of the available languages returned by available_languages().

See Also

available_languages() read_labels()

Examples

```
## Not run:
model <- birdnet_model_tflite(version = "v2.4")
labels_path(model, "fr")

## End(Not run)
```

`predict_species_at_location_and_time` *Predict species for a given location and time*

Description

Uses the BirdNET Species Range Model to estimate the presence of bird species at a specified location and time of year.

Usage

```
predict_species_at_location_and_time(
  model,
  latitude,
  longitude,
  week = NULL,
  min_confidence = 0.03
)

## S3 method for class 'birdnet_model_meta'
predict_species_at_location_and_time(
  model,
  latitude,
  longitude,
  week = NULL,
  min_confidence = 0.03
)
```

Arguments

`model`	birdnet_model_meta. An instance of the BirdNET model returned by `birdnet_model_meta()`.
`latitude`	numeric. The latitude of the location for species prediction. Must be in the interval [-90.0, 90.0].
`longitude`	numeric. The longitude of the location for species prediction. Must be in the interval [-180.0, 180.0].
`week`	integer. The week of the year for which to predict species. Must be in the interval [1, 48] if specified. If NULL, predictions are not limited to a specific week.
`min_confidence`	numeric. Minimum confidence threshold for predictions to be considered valid. Must be in the interval [0, 1.0).

Details

The BirdNET Species Range Model leverages eBird checklist frequency data to estimate the probability of bird species occurrences based on latitude, longitude, and time of year. It integrates actual observations and expert-curated data, making it adaptable to regions with varying levels of data availability. The model employs circular embeddings and a classifier to predict species presence and migration patterns, achieving higher accuracy in data-rich regions and lower accuracy in underrepresented areas like parts of Africa and Asia. For more details, you can view the full discussion here: https://github.com/kahst/BirdNET-Analyzer/discussions/234

Value

A data frame with columns: `label`, `confidence`. Each row represents a predicted species, with the `confidence` indicating the likelihood of the species being present at the specified location and time.

Examples

```
# Predict species in Chemnitz, Germany, that are present all year round
## Not run:
model <- birdnet_model_meta(language = "de")
predict_species_at_location_and_time(model, latitude = 50.8334, longitude = 12.9231)

## End(Not run)
```

`predict_species_from_audio_file` *Predict species within an audio file using a BirdNET model*

Description

Use a BirdNET model to predict species within an audio file. The model can be a TFLite model, a custom model, or a Protobuf model.

Usage

```
predict_species_from_audio_file(
  model,
  audio_file,
  min_confidence = 0.1,
  batch_size = 1L,
  chunk_overlap_s = 0,
  use_bandpass = TRUE,
  bandpass_fmin = 0L,
  bandpass_fmax = 15000L,
  apply_sigmoid = TRUE,
  sigmoid_sensitivity = 1,
  filter_species = NULL,
```

```
  keep_empty = TRUE,
  use_arrow = FALSE
)

## S3 method for class 'birdnet_model'
predict_species_from_audio_file(
  model,
  audio_file,
  min_confidence = 0.1,
  batch_size = 1L,
  chunk_overlap_s = 0,
  use_bandpass = TRUE,
  bandpass_fmin = 0L,
  bandpass_fmax = 15000L,
  apply_sigmoid = TRUE,
  sigmoid_sensitivity = 1,
  filter_species = NULL,
  keep_empty = TRUE,
  use_arrow = FALSE
)
```

Arguments

`model`	A BirdNET model object. An instance of the BirdNET model (e.g., `birdnet_model_tflite()`).
`audio_file`	character. The path to the audio file.
`min_confidence`	numeric. Minimum confidence threshold for predictions (default is 0.1).
`batch_size`	integer. Number of audio samples to process in a batch (default is 1L).
`chunk_overlap_s`	numeric. The overlap between audio chunks in seconds (default is 0). Must be in the interval [0.0, 3.0].
`use_bandpass`	logical. Whether to apply a bandpass filter (default is TRUE).
`bandpass_fmin, bandpass_fmax`	integer. Minimum and maximum frequencies for the bandpass filter (in Hz). Ignored if `use_bandpass` is FALSE (default is 0L to 15000L).
`apply_sigmoid`	logical. Whether to apply a sigmoid function to the model output (default is TRUE).
`sigmoid_sensitivity`	numeric. Sensitivity parameter for the sigmoid function (default is 1). Must be in the interval [0.5, 1.5]. Ignored if `apply_sigmoid` is FALSE.
`filter_species`	NULL, a character vector of length greater than 0, or a list where each element is a single non-empty character string. Used to filter the predictions. If NULL (default), no filtering is applied.
`keep_empty`	logical. Whether to include empty intervals in the output (default is TRUE).
`use_arrow`	logical. Whether to use Arrow for processing predictions (default is FALSE).

Details

Sigmoid Activation:

When `apply_sigmoid = TRUE`, the raw logit scores from the linear classifier are passed through a sigmoid function, scaling them into the range [0, 1]. This unitless confidence score reflects BirdNET's certainty in its prediction (it is not a direct probability of species presence). Adjusting the `sigmoid_sensitivity` parameter modifies the score distribution:

- Values **< 1** tend to produce more extreme scores (closer to 0 or 1).
- Values **> 1** result in scores that are more moderate (centered around intermediate values). For additional details on BirdNET confidence scores and guidelines for converting them to probabilities, see Wood & Kahl (2024).

Apache Arrow optimization:

By default, predictions from Python are converted to R using basic data structures. For large datasets using Apache Arrow (`use_arrow=TRUE`) can significantly improve performance by reducing memory usage during data conversion and minimizing data copying between R and Python.

When to use Apache Arrow:

- Large audio files (>20 minutes)
- Low confidence thresholds (`min_confidence < 0.1`)
- Memory-constrained environments
- Whenever you encounter an unusual long pause after inference. This is a sign that the data conversion is taking a long time.

Note that using Apache Arrow requires additional dependencies (`arrow` R package and `pyarrow` Python package). You can install them manually using `install_arrow()`.

Value

A data frame with the following columns:

start Start time of the prediction interval.

end End time of the prediction interval.

scientific_name Scientific name of the predicted species.

common_name Common name of the predicted species.

confidence BirdNET's confidence score for the prediction.

References

Wood, C. M., & Kahl, S. (2024). Guidelines for appropriate use of BirdNET scores and other detector outputs. Journal of Ornithology. https://doi.org/10.1007/s10336-024-02144-5

See Also

`read_labels()` for more details on species filtering.

`birdnet_model_tflite()`, `birdnet_model_protobuf()`, `birdnet_model_custom()`

Examples

```
## Not run:
model <- birdnet_model_tflite(version = "v2.4", language = "en_us")
audio_file <- system.file("extdata", "soundscape.mp3", package = "birdnetR")
predictions <- predict_species_from_audio_file(model, audio_file, min_confidence = 0.1)

## End(Not run)
```

`read_labels` *Read species labels from a file*

Description

This is a convenience function to read species labels from a file.

Usage

```
read_labels(species_file)
```

Arguments

`species_file`	Path to species file.

Value

A vector with class labels e.g. c("Cyanocitta cristata_Blue Jay", "Zenaida macroura_Mourning Dove")

See Also

`available_languages()` `labels_path()`

Examples

```
# Read a custom species file
read_labels(system.file("extdata", "species_list.txt", package = "birdnetR"))

# To access all class labels that are supported in your language,
# you can read in the respective label file
## Not run:
model <- birdnet_model_tflite(version = "v2.4", language = "en_us")
labels_path <- labels_path(model, "fr")
species_list <- read_labels(labels_path)
head(species_list)

## End(Not run)
```

Index