Package ‘animl’

October 12, 2022

Title A Collection of ML Tools for Conservation Research

Version 1.0.0

Description Functions required to classify subjects within camera trap field data. The package can handle both images and videos. The authors recommend a two-step approach using Microsoft's 'MegaDector' model and then a second model trained on the classes of interest.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2.9000

Imports pbapply, jpeg, keras, reticulate, tfdatasets, parallel, exifr, av, data.table, dplyr, stats, imager

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Kyra Swanson [aut, cre] (<`https://orcid.org/0000-0002-1496-3217`>), Mathias Tobler [aut]

Maintainer Kyra Swanson <`tswanson@sdzwa.org`>

Repository CRAN

Date/Publication 2022-03-23 09:00:15 UTC

R topics documented:

classifyImageMD 2
classifyImagesBatchMD 3
classifySequence 4
classifyVideo 5
convertMDresults 5
convertMDResultsSimple 6
decode_img_full 7
decode_img_resize 7
extractAllBoxes 8
extractAllBoxesFromFlat 9

extractBoxes 10
extractBoxesFromFlat 11
extractFiles 12
imagesFromVideos 12
loadMDModel 13
multispecies 14
parseMDjson 14
parseMDsimple 15
plotBoxes 16
plotBoxesFromFlat 16
resize_pad 17

Index **18**

`classifyImageMD` *Run MD on a single image*

Description

Returns the MD bounding boxes, classes, confidence above the min_conf threshold for a single image. #' Requires a an mdsession is already loaded (see loadMDModel()) and the file path of the image in question.

Usage

```
classifyImageMD(mdsession, imagefile, min_conf = 0.1)
```

Arguments

`mdsession`	should be the output from loadMDmodel(model)
`imagefile`	The path for the image in question
`min_conf`	Confidence threshold for returning bounding boxes, defaults to 0.1

Value

a list of MD bounding boxes, classes, and confidence for the image

Examples

```
## Not run:
 images<-read_exif(imagedir,tags=c("filename","directory","DateTimeOriginal","FileModifyDate"),
 recursive = TRUE)
 colnames(images)[1]<-"FilePath"
 mdsession<-loadMDModel(mdmodel)
 mdres<-classifyImageMD(mdsession,images$FilePath[1])

## End(Not run)
```

`classifyImagesBatchMD` *Run MegaDetector on a batch of images*

Description

Runs MD on a list of image filepaths. Can resume for a results file and will checkpoint the results after a set number of images

Usage

```
classifyImagesBatchMD(
  mdsession,
  images,
  min_conf = 0.1,
  batch_size = 1,
  resultsfile = NULL,
  checkpoint = 5000
)
```

Arguments

`mdsession`	should be the output from loadMDmodel(model)
`images`	list of image filepaths
`min_conf`	Confidence threshold for returning bounding boxes, defaults to 0.1
`batch_size`	Process images in batches, defaults to 1
`resultsfile`	File containing previously checkpointed results
`checkpoint`	Bank results after processing a number of images, defaults to 5000

Value

a list of lists of bounding boxes for each image

Examples

```
## Not run:
 images<-read_exif(imagedir,tags=c("filename","directory","DateTimeOriginal","FileModifyDate"),
 recursive = TRUE)
 colnames(images)[1]<-"FilePath"
 mdsession<-loadMDModel(mdmodel)
 mdres<-classifyImagesBatchMD(mdsession,images$FilePath,
                             resultsfile=mdresultsfile,checkpoint = 2500)

## End(Not run)
```

`classifySequence` *Leverage sequences to classify images*

Description

Images must be within maxdiff seconds of each other to be considered in sequence. This function retains "Empty" classification even if other images within the sequence are predicted to contain animals. Classification confidence is weighted by MD confidence.

Usage

```
classifySequence(
  imagesallanimal,
  mlpredictions,
  classes,
  emptycol,
  maxdiff = 60
)
```

Arguments

`imagesallanimal`	subselection of all images that contain MD animals
`mlpredictions`	classifier predictions
`classes`	list of all possible classes
`emptycol`	integer that represents the empty class
`maxdiff`	maximum difference between images in seconds to be included in a sequence, defaults to 60

Value

reclassified imagesallanimal dataframe

Examples

```
## Not run:
predictions <-classifyCropsSpecies(images,modelfile,resize=456)
imagesallanimal <- classifySequences(images,predictions,classes,17,maxdiff=60)

## End(Not run)
```

`classifyVideo` *Select Best Classification From Multiple Frames*

Description

Select Best Classification From Multiple Frames

Usage

```
classifyVideo(mdanimals, how = "count")
```

Arguments

`mdanimals`	dataframe of all frames including species classification
`how`	method for selecting best prediction, defaults to most frequent

Value

dataframe with new prediction in "Common" column

Examples

```
## Not run:
mdanimals <- classifyVideo(mdanimals)

## End(Not run)
```

`convertMDresults` *Convert the MD output into a data frame*

Description

Each row is a MD bounding box, there can be multiple bounding boxes per image.

Usage

```
convertMDresults(mdresults)
```

Arguments

`mdresults`	list of bounding boxes for each image

Value

A dataframe with one entry for each bounding box

Examples

```
## Not run:
 images<-read_exif(imagedir,tags=c("filename","directory","DateTimeOriginal","FileModifyDate"),
                   recursive = TRUE)
 colnames(images)[1]<-"FilePath"
 mdsession<-loadMDModel(mdmodel)
 mdres<-classifyImagesBatchMD(mdsession,images$FilePath,
                              resultsfile=mdresultsfile,checkpoint = 2500)
 mdresflat<-convertMDresults(mdres)

## End(Not run)
```

`convertMDResultsSimple` *Quickly flatten output from MegaDetector*

Description

Returns a dataframe where each row is a MD bounding box, does not take into consideration potential collisions with the outer edge of the image.

Usage

```
convertMDResultsSimple(mdresults)
```

Arguments

`mdresults`	list of bounding boxes for each image

Value

A dataframe with one entry for each bounding box

Examples

```
## Not run:
 images<-read_exif(imagedir,tags=c("filename","directory","DateTimeOriginal","FileModifyDate"),
                   recursive = TRUE)
 colnames(images)[1]<-"FilePath"
 mdsession<-loadMDModel(mdmodel)
 mdres<-classifyImagesBatchMD(mdsession,images$FilePath,
                              resultsfile=mdresultsfile,checkpoint = 2500)
 mdresflat<-convertMDResultsSimple(mdres)

## End(Not run)
```

`decode_img_full` *Load image with TF, full size for MD*

Description

Load image with TF, full size for MD

Usage

```
decode_img_full(image_filepath)
```

Arguments

`image_filepath`	the file path of the image

Examples

```
## Not run:
  dataset<-tfdatasets::dataset_map_and_batch(dataset,decode_img_full,batch_size)

## End(Not run)
```

`decode_img_resize` *Load image with TF 35722 and resize for MD*

Description

Load image with TF 35722 and resize for MD

Usage

```
decode_img_resize(image_filepath, height = 299, width = 299)
```

Arguments

`image_filepath`	the filepath of the image
`height`	resize height, defaults to 299 px
`width`	resize width, defualts to 299 px

Examples

```
## Not run:
  dataset<-tfdatasets::dataset_map_and_batch(dataset,decode_img_resize,batch_size)

## End(Not run)
```

`extractAllBoxes` *Extract bounding boxes and save as new image from a batch of images*

Description

Extract bounding boxes and save as new image from a batch of images

Usage

```
extractAllBoxes(
  images,
  min_conf = 0,
  buffer = 2,
  save = FALSE,
  resize = NA,
  outdir = "",
  quality = 0.8,
  parallel = FALSE,
  nproc = parallel::detectCores()
)
```

Arguments

`images`	list of images, raw MD output
`min_conf`	Confidence threshold (defaults to 0, not in use)
`buffer`	Adds a buffer to the MD bounding box, defaults to 2px
`save`	Toggle to save output cropped, defaults to FALSE
`resize`	Size in pixels to resize cropped images, NA if images are not resized, defaults to NA
`outdir`	Directory in which output cropped images will be saved
`quality`	Compression level of output cropped image, defaults to 0.8
`parallel`	Toggle to enable parallel processing, defaults to FALSE
`nproc`	Number of workers if parallel = TRUE, defaults to output of detectCores()

Value

a flattened dataframe containing crop information

Examples

```
## Not run:
images<-read_exif(imagedir,tags=c("filename","directory"), recursive = TRUE)
crops <- extractAllBoxes(images,save=TRUE,out)

## End(Not run)
```

`extractAllBoxesFromFlat`

Extract ,cropped images from a processed dataframe

Description

Extract ,cropped images from a processed dataframe

Usage

```
extractAllBoxesFromFlat(
  images,
  buffer = 0,
  resize = NA,
  quality = 0.8,
  outdir = "",
  save = FALSE,
  overwrite = TRUE,
  parallel = FALSE,
  nproc = parallel::detectCores()
)
```

Arguments

`images`	dataframe containing MD output (assumes single row)
`buffer`	Adds a buffer to the MD bounding box, defaults to 2px
`resize`	Size in pixels to resize cropped images, NA if images are not resized, defaults to NA
`quality`	Compression level of output cropped image, defaults to 0.8
`outdir`	Directory in which output cropped images will be saved
`save`	Toggle to save output cropped, defaults to FALSE
`overwrite`	Toggle to overwrite output cropped images if they already exis, defaults to TRUE
`parallel`	Toggle to enable parallel processing, defaults to FALSE
`nproc`	Number of workers if parallel = TRUE, defaults to output of detectCores()

Value

A dataframe containing image and crop paths

Examples

```
## Not run:
crops <- extractBoxesFromFlat(mdresflat,save=TRUE,out)

## End(Not run)
```

`extractBoxes` *Extract bounding boxes for a single image and save as new images*

Description

Requires the unflattened raw MD output

Usage

```
extractBoxes(
  image,
  min_conf = 0,
  buffer = 2,
  plot = TRUE,
  return.crops = FALSE,
  save = FALSE,
  resize = NA,
  outdir = "",
  quality = 0.8
)
```

Arguments

`image`	single image, raw MD output
`min_conf`	Confidence threshold (defaults to 0, not in use)
`buffer`	Adds a buffer to the MD bounding box, defaults to 2px
`plot`	Toggle to plot each crop in the plot window, defaults to TRUE
`return.crops`	Toggle to return list of cropped images, defaults to FALSE
`save`	Toggle to save output cropped, defaults to FALSE
`resize`	Size in pixels to resize cropped images, NA if images are not resized, defaults to NA
`outdir`	Directory in which output cropped images will be saved
`quality`	Compression level of output cropped image, defaults to 0.8

Value

a flattened dataframe containing crop information

Examples

```
## Not run:
images<-read_exif(imagedir,tags=c("filename","directory"), recursive = TRUE)
crops <- extractBoxes(images[1,],plot=TRUE,return.crops=TRUE,save=TRUE)

## End(Not run)
```

`extractBoxesFromFlat` *Extract crops from a single image represented by a processed dataframe*

Description

Extract crops from a single image represented by a processed dataframe

Usage

```
extractBoxesFromFlat(
  image,
  min_conf = 0,
  buffer = 0,
  plot = TRUE,
  save = FALSE,
  resize = NA,
  outdir = "",
  quality = 0.8
)
```

Arguments

`image`	dataframe containing MD output (assumes single row)
`min_conf`	Confidence threshold (defaults to 0, not in use)
`buffer`	Adds a buffer to the MD bounding box, defaults to 2px
`plot`	Toggle to plot each crop in the plot window, defaults to TRUE
`save`	Toggle to save output cropped, defaults to FALSE
`resize`	Size in pixels to resize cropped images, NA if images are not resized, defaults to NA
`outdir`	Directory in which output cropped images will be saved
`quality`	Compression level of output cropped image, defaults to 0.8

Value

A dataframe containing image and crop paths

Examples

```
## Not run:
crops <- extractBoxesFromFlat(mdresflat[1,],save=TRUE,out)

## End(Not run)
```

---

extractFiles *Extract exif data and create dataframe, adjust time if necessary.*

Description

Extract exif data and create dataframe, adjust time if necessary.

Usage

```
extractFiles(imagedir, timezone_offset = 0)
```

Arguments

`imagedir`	file path
`timezone_offset`	integer to adjust file modify time

Value

images

Examples

```
## Not run:
images <- extractFiles("C:\\Users\\usr\\Pictures\\")

## End(Not run)
```

---

imagesFromVideos *Extract frames from video for classification*

Description

This function can take

Usage

```
imagesFromVideos(
  videos,
  outdir = tempfile(),
  format = "jpg",
  fps = NULL,
  frames = NULL,
  parallel = FALSE,
  nproc = 1
)
```

Arguments

`videos`	dataframe of videos
`outdir`	directory to save frames to
`format`	output format for frames, defaults to jpg
`fps`	frames per second, otherwise determine mathematically
`frames`	number of frames to sample
`parallel`	Toggle for parallel processing, defaults to FALSE
`nproc`	number of processors to use if parallel, defaults to 1

Value

dataframe of still frames for each video

Examples

```
## Not run:
frames <- imagesFromVideos(videos,outdir="C:\\Users\\usr\\Videos\\",frames=5)

## End(Not run)
```

`loadMDModel` *Title*

Description

Title

Usage

```
loadMDModel(modelfile)
```

Arguments

`modelfile`	.pb file obtained from megaDetector

Value

a tfsession containing the MD model

Examples

```
## Not run:
mdmodel<-"megadetector_v4.1.pb"
mdsession<-loadMDModel(mdmodel)

## End(Not run)
```

`multispecies` *Classify images with multiple potential species*

Description

Classify images with multiple potential species

Usage

```
multispecies(imagesallanimal, mlpredictions, classes, emptycol)
```

Arguments

`imagesallanimal`	dataframe of MD animal images
`mlpredictions`	classifier predictions for each box
`classes`	list of all possible classes
`emptycol`	integer value of the empty column

Value

list of images with multiple species, includes number of animals for each class

Examples

```
## Not run:
imagesallanimal <- multispecies(imagesallanimal,mlpredictions,classes,17)

## End(Not run)
```

`parseMDjson` *parse MD JSON results file*

Description

parse MD JSON results file

Usage

```
parseMDjson(json)
```

Arguments

`json`	md output file

Value

list of MD detections

Examples

```
## Not run:
mdres <- parseMDjson("MDresults.json")

## End(Not run)
```

`parseMDsimple` *parse MD JSON results file into a simple dataframe*

Description

parse MD JSON results file into a simple dataframe

Usage

```
parseMDsimple(mdresults)
```

Arguments

`mdresults`	raw MegaDetector output

Value

flattened dataframe of results

Examples

```
## Not run:
mdresults <- parseMDsimple(mdres)

## End(Not run)
```

`plotBoxes` *Plot bounding boxes on image from md results*

Description

Plot bounding boxes on image from md results

Usage

```
plotBoxes(image, label = FALSE, minconf = 0)
```

Arguments

`image`	The mdres for the image
`label`	T/F toggle to plot MD category
`minconf`	minimum confidence to plot box

Value

no return value, produces bounding box in plot panel

Examples

```
## Not run:
mdres<-classifyImageMD(mdsession,images$FilePath[30000])
plotBoxes(mdres,minconf = 0.5)

## End(Not run)
```

`plotBoxesFromFlat` *Plot bounding boxes from MD flat data frame*

Description

Plot bounding boxes from MD flat data frame

Usage

```
plotBoxesFromFlat(image, label = FALSE, minconf = 0)
```

Arguments

`image`	The mdres for the image
`label`	Toggle to plot MD category, defaults to FALSE
`minconf`	minimum confidence to plot box

Value

no return value, produces bounding box in plot panel

Examples

```
## Not run:
mdres<-classifyImageMD(mdsession,images$FilePath[30000])
mdresflat<-flattenBoxesMDSimple(mdres)
plotBoxesFromFlat(mdresflat,minconf = 0.5)

## End(Not run)
```

`resize_pad` *Resize an image with padding*

Description

Resize an image with padding

Usage

```
resize_pad(img, size = 256)
```

Arguments

`img`	the image, read by jpeg library
`size`	new size

Value

returns resized jpeg image

Examples

```
## Not run:
crop<-resize_pad(cropped_image_path,256)

## End(Not run)
```

Index

`classifyImageMD`, 2
`classifyImagesBatchMD`, 3
`classifySequence`, 4
`classifyVideo`, 5
`convertMDresults`, 5
`convertMDResultsSimple`, 6

`decode_img_full`, 7
`decode_img_resize`, 7

`extractAllBoxes`, 8
`extractAllBoxesFromFlat`, 9
`extractBoxes`, 10
`extractBoxesFromFlat`, 11
`extractFiles`, 12

`imagesFromVideos`, 12

`loadMDModel`, 13

`multispecies`, 14

`parseMDjson`, 14
`parseMDsimple`, 15
`plotBoxes`, 16
`plotBoxesFromFlat`, 16

`resize_pad`, 17