Package 'CameraTrapDetectoR'

January 13, 2022

Type Package

Title CameraTrapDetectoR

Version 0.0.2

Author Michael A. Tabak

Maintainer Mikey Tabak <mtabak@west-ulc.ca>

Description This package will run object detection models on camera trap images. There is a general model that detects, counts, and classifies objects as birds, humans, (non-human) mammals, and vehicles, while also IDing empty images. There is another model that goes to the species level, and a third model that classifies to the family level. The `deploy_model` function is most relevent and will run each type of model on your images. The `runShiny` function will launch a Shiny Application to run the `deploy_model` function.

License GPL-2

Encoding UTF-8

LazyData yes

RoxygenNote 7.1.2

Imports torchvision, torch, magick, shiny, shinyFiles, shinyBS, shinyjs, fs, rappdirs

R topics documented:

dataLoader *Create the data loader*

Description

This function loads a batch of data to be passed to the model

Usage

```
dataLoader(file_list, index, w = 408, h = 307)
```

Arguments

file_list	passed from deploy fuction
index	i value from deploy function
w	image width after resizing. Recommend not changing this
h	image height after resizing. Recommend not changing this

dataset *Create the dataset*

Description

This function sets up the dataset

Usage

```
dataset(
  data_dir,
  recursive = TRUE,
  file_extensions = c(".jpg", ".JPG"),
  labeled = FALSE
)
```

Arguments

data_dir	data directory
recursive	boolean
file_extensions	file extensions
labeled	boolean. not functional yet

decode_output *Decode the output from neural network*

Description

This function will convert the output from NN into a format that can be used to make plots and provide a csv with information

Usage

```
decode_output(output, label_encoder, h, score_threshold = 0.7)
```

Arguments

output	this is a subset of the list output from the neural net
label_encoder	passed from deploy model function
h	image height after resizing. Recommend not changing this
score_threshold	threshold score for keeping bounding boxes

deploy_model *Deploy model on camera trap images*

Description

This function deploys a model trained to identify and count the objects in camera trap images.

Usage

```
deploy_model(
  data_dir = NULL,
  model_type = "general",
  recursive = TRUE,
  file_extensions = c(".jpg", ".JPG"),
  make_plots = TRUE,
  plot_label = TRUE,
  output_dir = NULL,
  sample50 = FALSE,
  write_bbox_csv = FALSE,
  score_threshold = 0.6,
  h = 307,
  w = 408,
  lty = 1,
  lwd = 2,
  col = "red",
  labeled = FALSE
)
```

Arguments

data_dir	Absolute path to the folder containing your images
model_type	Options are c('general', 'species', 'family'). The 'general' model predicts to the level of mammal, bird, humans, vehicles. The 'species' model recognizes 58 species. The 'family' model recognizes 25 families.
recursive	boolean. Do you have images in subfolders within your data_dir that you want to analyze, if so, set to TRUE. If you only want to analyze images within your data_dir and not within sub-folders, set to FALSE.
file_extensions	The types of extensions on your image files. Default is c(".jpg", ".JPG")
make_plots	boolean. Do you want to make plots of the images with their predicted bounding boxes?
plot_label	boolean. Do you want the plots to contain the predicted class of object
output_dir	You can specify absolute path to output. Default is 'NULL', and creates a folder within your data_dir. Only specify a path if you want the results stored somewhere else on your computer.
sample50	boolean. Do you want to run the model only on a subset of 50 images from your dataset? This is a good idea if you are experimenting with settings.
write_bbox_csv	boolean. Do you want to create a csv with all of the information on predicted bounding boxes? This csv will include all bounding boxes, even those with low probability.
score_threshold	Confidence threshold for using a bounding box. Default is 0.6. A lower number will produce more bboxes (it will be less stringent in deciding to make a bbox). A higher number will produce fewer bboxes (it will be more stringent).
h	The image height (in pixels) for the annotated plot. Only used if `make_plots=TRUE`.
w	The image width (in pixels) for the annotated plot.
lty	line type for bbox plot. See `?plot` for details
lwd	line width for bbox plot. See `?plot` for details
col	line color for bbox plot. See `?plot` for details
labeled	This is not functional

Details

This function deploys a model to detect and classify objects in camera trap images. The function will find all files matching the 'file_extension's specified within the 'data_dir' specified and deploy the 'model_type' on these images. If you specify `recusive=TRUE`, the function will find relevant image files within all subdirectories of your 'data_dir'. 'deploy_model' returns a dataframe of predicted number of individuals within each category in each image. This dataframe is also written as a csv file within your 'output_dir'. If you specify `make_plots=TRUE`, the function will plot predicted bounding boxes for each image in your 'output_dir'. If you are working with many images, you may wish to specify `sample50=TRUE` the first time you use this function, which will only deploy the model on 50 of your images. There are three options for `model_type`: 'general' recognizes mammals, birds, humans, and vehicles. 'species' recognizes 58 species. 'family' recognizes 25

families. If you want to see all of the information for each bounding box (including coordinates, labels, and confidence), specify `write_bbox_csv=TRUE` and it will be produced in your 'output_dir'. Additionally, A file called "arguments" will be produced in your 'output_dir' this is a list of all of the arguments you passed to this function for reference.

Value

Returns a dataframe of predictions for each file. The rows in this dataframe are the file names in your 'data_dir'; the columns are the categories in the model. If any of your images were not loaded properly, there will be a column in the dataframe called 'image_error'. Images with a 1 in this column had issues and the model was not deployed on them.

---

download_cache *download files needed to run the mdoel*

Description

This function will download necessary files and store them in the package space

Usage

```
download_cache(
 url = "https://www.dropbox.com/s/m4ojnotd2pev46u/weights_family_cpu.pth?raw=1",
  redownload = FALSE
)
```

Arguments

`url`	location of file to download
`redownload`	boolean. Re-download the file?

---

install_dependencies *Function to install packages*

Description

This function will install packages that are needed in CameraTrapDetectoR

Usage

```
install_dependencies(
 packages = c("torchvision", "torch", "magick", "shiny", "shinyFiles", "shinyBS",
    "shinyjs", "rappdirs", "fs")
)
```

Arguments

packages	packages to be installed, as a vector. If you are planning to use the Shiny App, leave this value as default. If you want to avoid installing the Shiny-specificy dependencies, use `packages=c('torchvision','torch','magick')`

`plot_img_bbox` *Make plots of the image with bounnding box predictions*

Description

Plots original image with predicted bounding box and (optionally) the predicted category

Usage

```
plot_img_bbox(
  filename,
  pred_df,
  output_dir,
  data_dir,
  plot_label = TRUE,
  col = "red",
  lty = 1,
  lwd = 2,
  prop_bbox = FALSE,
  w = 408,
  h = 307
)
```

Arguments

filename	The file containing the image
pred_df	Prediction dataframe that is output from deployment
output_dir	Desired directory to make plots
data_dir	absolute path to images
plot_label	boolean. Do you want the predicted category on the plot?
col	color of the bbox (and label if 'plot_label=TRUE'). See '?plot' for an explanation of 'col', 'lwd', and 'lty'
lty	line type of bbox
lwd	line width of bbox
prop_bbox	boolean. Are the bbox coordinates in proportion instead of exact coordinates? Only 'TRUE' if you are using a different image size
w	The image width (in pixels) for the annotated plot.
h	The image height (in pixels) for the annotated plot. Only used if `make_plots=TRUE`.

runShiny *A wrapper function to run Shiny Apps from* CameraTrapDetectoR.

Description

Running this function will launch a shiny application to deploy object detection models on camera trap images

Usage

```
runShiny(app = "deploy")
```

Arguments

app	The name of the app you want to run. The options are currently 'deploy'.

weightLoader *Load the trained model*

Description

Loads the weights for the pretrained model.

Usage

```
weightLoader(model_type = "general", num_classes = 5)
```

Arguments

model_type	The type of model you want to deploy: c('mammalBirdVehicle', '')
num_classes	The number of classes in the model

Index