

L3_Project

Release 0.5

Corcoral Cetre Barbat

Feb 09, 2021

1	Application main	1
2	Application generate_csv	5
3	Application get_model	7
4	Preparation Cnn model	9
5	Preparation extract infos	11
6	Preparation extraction feature	13
7	Preparation find best epoch	15
8	Preparation format data	17
9	Preparation model builder	19
10	Preparation prediction	21
11	Preparation progress bar	23
	Python Module Index	25
	Index	27

Application main

class application.app.**AffichageRes** (*can, prediction_label, prediction, resultats, resultats_label*) Show the results of the prediction

class application.app.**AffichageSon** (*can, show_audio, show_spec, show_mfcc, play_btn, wav_pic, process_pic, open_test_but, run_test_but*) Display all the informations about the sounds

class application.app.**Aide** (*main_can*) This class is used to print the help / instructions should the user press the help button in the bottom right corner

class application.app.**Footer** (*can, but, help_button*) Small part who display the version num and the two help buttons

class application.app.**Header** (*can, icon*) Small part to display the program icon

class application.app.**InfosMenu** (*can_menu, text, label, epoch, label_epoch, ratio, ratio_spinbox, rs, Side rs_spinbox, label_rs, label_ratio, save_model_but, name_model, name_entry, save_data_but, name_data, name_data_entry, best_epoch_but, val*) Menu who display the informations about the preparation of the model and the saves

class application.app.**Menu** (*can, quit_pic, run_pic, folder_pic, open_but, run_but, quit_but, csv_pic, open_csv_but, format_data_but, format_pic, generate_csv_but, save_csv_img, import_pic, import_model_but, import_data_pic, import_data_but*) Main menu with all

the
buttons
to
manage
the
program

class application.app.**RecapSelect** (*can, data_path_label, data_path_var, csv_path_label, csv_path_var*) Recap the choices for the csv path and the data path

application.app.**change** (*new*) This allows the estimated accuracy to be changed using the value in parameters

application.app.**choose_dir_data** () This is used to indicate the path to the dataset

application.app.**choose_path_csv** () This is used to indicate the path to the .csv

application.app.**choose_test_path** () This is used to indicate the path to the tested file

application.app.**clear_folder** (*folder*) This is used to clear the paths to the different

application.app.**copy_floder_content** (*origin, dest*) Copy all the content of a repository (origin) to another one (dest) :param origin: path of the original repository :param dest: path of the destination repository :return: 0 if everything ok -1 if error

application.app.**create_help** (*path_help*) Aims to create the help section in english

application.app.**find_best_epoch** () Call the function, aims at find best epochs

application.app.**format_data** () This is used to format the data using the different paths indicated

application.app.**generate_csv** () This is used to generate a .csv (Spreadsheet) using the Data set indicated

application.app.**import_data** () Import the data by asking the user

application.app.**import_model** () Import a model by asking the user

application.app.**init_aide** (*win*) Is used to create the help window

application.app.**init_footer** () This is used to initialize the footer for the window

application.app.**init_header** () This is used to initialize the header for the window

application.app.**init_infos_menu** () This is used to indicate all the needed informations to

write the text on the buttons

`application.app.init_menu ()` This function creates the buttons needed to indicate the paths of the different data needed / processing the data

`application.app.init_model ()` This is used to initialize the model using `model.json` and the obtained accuracy

`application.app.init_recap_selec ()` Initialise the show data path and csv path section

`application.app.init_resultats ()` This is used to return the results of the prediction

`application.app.init_sons ()` This is used to create the buttons used to see the graphic representations of a sound

`application.app.init_window ()` This function is used to initialize the window, and lock its size

`application.app.leave ()` This is used to quit the application

`application.app.load_model ()` Used to load a pre-existing model into the application

`application.app.predict ()` This is used to get the prediction of what bird / bat species the test sound corresponds to, and then print it

`application.app.run_model ()` This is used to run the model using `cnn_model.py` after formatting the data

`application.app.run_test_audio ()` This is used to play the sound so you can hear it. If it's an mp3, it will use `pygame` to play it, otherwise, it will use `playsound`

`application.app.save_as_data_format ()` This is used to save the data for later

`application.app.save_as_model ()` This is used save the current model in the "local save" folder to be able to use it later without re-creating every step

`application.app.set_rep (win)` Get the model from a repository

`application.app.set_rep_data (win)` Get the data from a repository

`application.app.set_zip (win)` Get the model from a zip file

`application.app.set_zip_data (win)` Get the data from a zip file

`application.app.show_aide ()` Display the french version of the help

`application.app.show_audio_representation ()` This is used to show the audio representation corresponding to the sound that is being tested

`application.app.show_help ()` Display the english version of the help

`application.app.show_mfccs ()` This is used to show the mfcc corresponding to the sound that is being tested

`application.app.show_spectrogramme ()` This is used to show the spectrogram corre-

sponding to the sound that is being tested

`application.app.test ()` This function prints the current paths to the csv and data for the test

Application generate_csv

`application.generate_csv.deep_search (full_path, local_path)` Is used to get all the folders inside the indicated folder, to get the species of the animals

`application.generate_csv.generate (path)` Generates a .csv using a folder. Inside the folder, there must be other folders named after the species, and containing the sounds of said species

Application `get_model`

`application.get_model.get_model (full_path)` Is used to get all the folders inside the indicated folder, to get the species of the animals

`application.get_model.local_unzip (full_path, tmp_path)` Is used to un-zip a compressed folder in a tmp_folder

Preparation Cnn model

`application.preparation_v2.cnn_model.run_model (epoch=10)` Load all the data from local saves, read the labels of the prepared audio, then build the model, fit and evaluate it. Finally the function save the model in local files

Preparation extract infos

`application.preparation_v2.extract_infos.generate_labels (path_csv, path_txt)` Generate the class_label file with all the labels

`application.preparation_v2.extract_infos.get_infos (path_csv)` Get all the infos from the csv file through our norm

`application.preparation_v2.extract_infos.read_labels (file_path)` Read line by line a .txt file

Preparation extraction feature

`application.preparation_v2.extraction_feature.preprocess_audio (audio_path)` Extract audio values and process noise reduction

`application.preparation_v2.extraction_feature.process_audio (audio_path)` Get the mfccs, cut into 1 second length and select most noisy parts

Preparation find best epoch

```
application.preparation_v2.find_best_epoch.get_best ( )
```

Load the data, load the model and search the best number of epoch thanks to the test values (the model doesn't know them)

Preparation format data

```
application.preparation_v2.format_data.get_the_data ( data_path, csv_path, Get the  
label_text_path, ratio=0.1, rs=42 ) data from  
a specific  
path with  
a csv
```

Preparation model builder

`application.preparation_v2.model_builder.builder (size)` The neural network responsible for the data analysis

Preparation prediction

```
application.preparation_v2.prediction.print_prediction(file_name, model)
```

Prints the estimated specie for the sound and the general percentages

Preparation progress bar

`application.preparation_v2.progress_bar.close_loading(fig)` Just close the progress bar which is a figure

`application.preparation_v2.progress_bar.create_loading()` Create the loading bar by analysing the plot

`application.preparation_v2.progress_bar.update_loading(ax, name, percent)` Update the percent of the progress bar

a

application
 application.app, 1
 application.generate_csv, 5
 application.get_model, 7
 application.preparation_v2.cnn_model,
 9
 application.preparation_v2.extract_infos,
 11
 application.preparation_v2.extraction_feature,
 13
 application.preparation_v2.find_best_epoch,
 15
 application.preparation_v2.format_data,
 17
 application.preparation_v2.model_builder,
 19
 application.preparation_v2.prediction,
 21
 application.preparation_v2.progress_bar,
 23

A

AffichageRes (class in application.app), 1
AffichageSon (class in application.app), 1
Aide (class in application.app), 1
application.app
 module, 1
application.generate_csv
 module, 5
application.get_model
 module, 7
application.preparation_v2.cnn_model
 module, 9
application.preparation_v2.extract_infos
 module, 11
application.preparation_v2.extraction_feature
 module, 13
application.preparation_v2.find_best_epoch
 module, 15
application.preparation_v2.format_data
 module, 17
application.preparation_v2.model_builder
 module, 19
application.preparation_v2.prediction
 module, 21
application.preparation_v2.progress_bar
 module, 23

B

builder() (in module application.preparation_v2.model_builder), 19

C

change() (in module application.app), 2
choose_dir_data() (in module application.app), 2
choose_path_csv() (in module application.app), 2
choose_test_path() (in module application.app), 2
clear_folder() (in module application.app), 2
close_loading() (in module application.prepara-

tion_v2.progress_bar), 23
copy_floder_content() (in module application.app), 2
create_help() (in module application.app), 2
create_loading() (in module application.preparation_v2.progress_bar), 23

D

deep_search() (in module application.generate_csv), 5

F

find_best_epoch() (in module application.app), 2
Footer (class in application.app), 1
format_data() (in module application.app), 2

G

generate() (in module application.generate_csv), 5
generate_csv() (in module application.app), 2
generate_labels() (in module application.preparation_v2.extract_infos), 11
get_best() (in module application.preparation_v2.find_best_epoch), 15
get_infos() (in module application.preparation_v2.extract_infos), 11
get_model() (in module application.get_model), 7
get_the_data() (in module application.preparation_v2.format_data), 17

H

Header (class in application.app), 1

I

import_data() (in module application.app), 2
import_model() (in module application.app), 2
InfosMenu (class in application.app), 1
init_aide() (in module application.app), 2
init_footer() (in module application.app), 2

init_header() (in module application.app), 2
init_infos_menu() (in module application.app), 2
init_menu() (in module application.app), 3
init_model() (in module application.app), 3
init_recap_selec() (in module application.app), 3
init_resultats() (in module application.app), 3
init_sons() (in module application.app), 3
init_window() (in module application.app), 3

L

leave() (in module application.app), 3
load_model() (in module application.app), 3
local_unzip() (in module application.get_model), 7

M

Menu (class in application.app), 1
module
 application.app, 1
 application.generate_csv, 5
 application.get_model, 7
 application.preparation_v2.cnn_model, 9
 application.preparation_v2.extract_infos, 11
 application.preparation_v2.extraction_feature, 13
 application.preparation_v2.find_best_epoch, 15
 application.preparation_v2.format_data, 17
 application.preparation_v2.model_builder, 19
 application.preparation_v2.prediction, 21
 application.preparation_v2.progress_bar, 23

P

predict() (in module application.app), 3
preprocess_audio() (in module application.preparation_v2.extraction_feature), 13
print_prediction() (in module application.preparation_v2.prediction), 21
process_audio() (in module application.preparation_v2.extraction_feature), 13

R

read_labels() (in module application.preparation_v2.extract_infos), 11
RecapSelect (class in application.app), 2
run_model() (in module application.app), 3
run_model() (in module application.preparation_v2.cnn_model), 9
run_test_audio() (in module application.app), 3

S

save_as_data_format() (in module application.app), 3
save_as_model() (in module application.app), 3
set_rep() (in module application.app), 3
set_rep_data() (in module application.app), 3
set_zip() (in module application.app), 3
set_zip_data() (in module application.app), 3
show_aide() (in module application.app), 3
show_audio_representation() (in module application.app), 3
show_help() (in module application.app), 3
show_mfccs() (in module application.app), 3
show_spectrogramme() (in module application.app), 3

T

test() (in module application.app), 4

U

update_loading() (in module application.preparation_v2.progress_bar), 23