L3_Project

Release 0.5

Corcoral Cetre Barbat

Feb 09, 2021

Table of Contents

Chapter 1

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*, *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*) Side 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

Chapter 3

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. Finaly the function save the model in local files

Chapter 5

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*, *label_text_path*, *ratio=0.1*, *rs=42*) Get the 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

L

M

P

R

S

T

U