Anna Kondrat, s32252

Dokumentacja projekt PPY

Przepraszam za nieprzyjemny spis treści, Sphinx wyrzucał mi błędy z powodu problemów z biblioteką tensorflow.keras, choć w kodzie działała ona bez problemu. Te same komentarze znajdują się w docstringach w kodzie.

Spis treści

createModel.py

Script for training the model to recognize the number of fingers shown in a photo.
Data source: Kaggle Hub ("koryakinp/fingers")

Tuning.py

Script for tuning an already created model with some more realistic photos, that I took by myself.
It creates a new model - hand_model_finetuned.keras, used both in try.py and Main.py

Try.py

def tryOut(photo)

Function allows to test the prediction of the model on the given photo.
  Args:
    photo (str): path to the photo
  Writes:
    prediction (List[int]): predictions of the model
    predicted class (int)

Anna Kondrat, s32252

Main.py

Program allows the user to answer questions by showing a certain number of fingers to the camera. App class creates the
whole program and uses previously created machine learning models.

Functionalities:
- user can sign in, the program remembers all the answers as well as questions that were answered, so that no question is asked
twice to the same user.
- questions are randomized
- user can turn on the camera whenever ready and take a photo of the answer
- user can draw a different question (repeated randomization)
- questions are drawn until all were answered by the certain user
- user can save the picture taken for the last answer
- user can decide to either see last taken photos on the screen or hide them
- history of user's answers is visible on the screen

def __init__(self):

Constructor. Creates the main screen, sets size, initializes finger recognision model, loads previously prepared questions and turns on the first window to allow for users' login.

def firstWindow(self):

Function allows for logging in.
If a user just signed out, their data is saved.
If a given login is already in database, then it loads the data (answers, indexes to be used) to variables.
If a given login is not in the database, it creates an empty data dictionary and fills indexes to be used with every index.
After accepting the login, it turns on the main window of the program.

def mainWindow(self):

Creates the whole structure of the program with linking functions to certain buttons and loading history of answers from database.
Left side of the screen is meant for the interaction with the user (opening camera, drawing a question, saving a photo, showing photos).
Right side of the screen is meant for displaying the history of user's answers with a button allowing for signing out.

def openCamera(self):

Function that open a camera. When user types q, then last frame is saved (changing colors, dimenentions), predictions are made. Predictions are added to history log and answers. Photo to be shown in main window gui is changed.

Anna Kondrat, s32252

def addHisRow(self, answer):

Function to add an answer to the history frame. Used both with loading all answers from history,
but also new answers predicted by gesture recognition model.
Args:
    answer (str): The answer to be displayed.

def getAnswers(self):

Function to load all user answers from the JSON database file into memory.
Reads the 'answers.json' file and stores its content in a dictionary, which maps usernames to their saved answers and remaining question indexes.

def drawQuestion(self):

Function to draw a random question from the question base that the user hasn't answered yet.
If there is no questions left for a user, disables the ability to open the camera.
Returns:
    str: question text or "brak pytań"

def saveAnswers(self):

Function to save all user answers from memory to a JSON file.

def saveLastPhoto(self):

Function to save the last photo (in the grayscale) from the history in the photoAnswers directory with the name: {nameOfUser}_{numerOfQuestion}.jpg. Returns an info box with either information about successful saving or a warning that there was no photo to be saved.

def updatePhoto(self):

Function to display (or not) the last image in greyscale, based on the option chosen by the user.