Animal behaviour video algorithm: concatenating videos followed by clipping into individual animal videos

Author: Yash Dixit
Date: 08/10/2024

```
#Import all the necessary libraries

import os

import glob

import numpy as np

import pandas as pd

from moviepy.editor import * #(https://pypi.org/project/moviepy/) MoviePy is a Python library for video editing: cutting, concatenations, title insertions, video compositing (a.k.a. non-linear editing), video processing, and creation of custom effects.


#Create a variable for the folder containing videos

path= "O:\\Te361 TA0573 Piglet Behaviour\\Original Video Footage\\"

#Change directory using the path variable

os.chdir(path)

#List all the files (in this case these were individual folders containing video files)

M= os.listdir()
```

There are 3 for loops:

Loop 1: runs for total number of folders containing video files.

a) Changes the directory to the first folder in the list
b) Creates a list of files with .avi extension.
c) If there are no files it skips the loop else create an empty list “clip”
    a. Loop 2: runs for the total number of video files in the folder
        i. Reads (VideoFileClip) the first video in the folder, checks for the size (to ensure all videos are of same size), if not resizes it to 1600 X 1200 px.
        ii. Once done, appends the video file in the “clip” list.
        iii. This is repeated for all videos in that folder.
        iv. The list of videos “clip” is then concatenated (concatenate_videoclips) and audio is removed (final.without_audio()). Note: Removing audio can be avoided, it is to remove unnecessary data.
        v. J^{th} excel sheet from the Excel workbook is read which contains timestamps: start and stop times for each animal as well as code for the animal, where each excel sheet in the workbook is related to a particular folder (list “M”).
            1. Loop 3: runs for the total number start and stop times which is equal to total number of animals in the concatenated video

Animal behaviour video algorithm: concatenating videos followed by clipping into individual animal videos

Author: Yash Dixit
Date: 08/10/2024

a. The algorithm uses the information (start time, stop time and animal code) to clip video for individual animal (final.subclip(start_seconds, end_seconds)).
b. The clipped video is saved with individual animal code in a folder (form the list M) as an .MP4 file (e.g., T161.MP4).
c. The whole process is repeated for all the folders (list "M") until all the videos for individual animals have been saved.

```
for j in range(len(M)):
    os.chdir(os.path.join(path,M[j]))# Changes the directory to the ith  folder in the list
    L = glob.glob('*avi')# Creates a list of files with .avi extension.
    if L == []: # if no .avi files are found skip the loop
        pass
    else:
        clip = [] # create an empty list
        for i in range(len(L)):
            tmp1= VideoFileClip(L[i]) # reads ith .avi file
            if tmp1.size == [1600, 1200]: # check the video frame size
                clip.append(tmp1)# if size is 1600X1200 append it to the empty list "clip"
            else:
                tmp2= tmp1.resize((1600,1200)) # if size is not 1600X1200 resize the file
                clip.append(tmp2)# append it to the empty list "clip"
        final = concatenate_videoclips(clip) #append it to the empty list "clip"
        final = final.without_audio()# remove audio from the file
        time_stamps= pd.read_excel("O:\\Te361 TA0573 Piglet Behaviour\\Behaviour Testing Recording sheets\\recording_time_new.xlsx",
                                   sheet_name= j) #read the jth sheet from excel work book read which contains timestamps: start and stop times for each animal as well as code for the animal, where each excel sheet in the workbook is related to a particular folder (list "M").
        time_stamps['animal_start_stop'] = time_stamps[['animal_no','Start','Stop']].apply(tuple, axis=1)# extract start and stop time from the work sheet
        segments= list(time_stamps['animal_start_stop'])#convert the timestamps to a list to be used in last loop
```

Animal behaviour video algorithm: concatenating videos followed by clipping into individual animal videos

Author: Yash Dixit
Date: 08/10/2024

```
    if segments == []: #if the sheet is empty skip the loop

        pass

    else:

        for anim, start_seconds, end_seconds in segments: #loop through start, stop times and animal code

            # crop a video clip and add it to list

            c = final.subclip(start_seconds, end_seconds)# clip the video using start and end times

            save_directory = os.path.join(path,M[j]) # create save directory for the video

            c.write_videofile(os.path.join(save_directory,str(anim)+".mp4"))#save the clipped video with the animal code from the excel sheet
```