General

1. Prepare your data
   a. Either, a folder contains only files of one channel, or
   b. In each folder weh ave the structure channel1/, channel2/... containing the corresponding data.
2. Open convertVideos.py (e.g. with spyder)
3. Adjust parameters (see below) and run the file.
4. Type convertVideos() into shell / console in order to start the process.

Variables and Parameters

We make an example:

- Zoo: **FancyZoo**
- Species: **Wildebeest**
- [videonumber **1** (only if the folder contains only the files of one channel)]
- Path to the .asf files: **Server/Wildebeest/FancyZoo/Box_1/, …, Server/Wildebeest/FancyZoo/Box_3/** (3 channels)
- Recording per night: 17:00 to 06:59:59 and per day 07:00 to 16:59:59
- Resulting video files (output)**: YYYY-MM-DD_Wildebeest_FancyZoo_1.avi, …, YYYY-MM-DD_Wildebeest_FancyZoo_3.avi**

Parameter	Type	Explanation	Example
PATH_TO_VID	String	• Path to the videofiles created by lupus (one folder, one channel) or various subfolders each containing the files of one channel.	PATH_TO_VID = ,Server/Wildebeest/FancyZoo/'
CONVERT_WHOLE_FOLDER	Bool	• True: Multiple subfolders (for each channel one) exist. • False: Path directly tot he .asf files. • Beware: The subfolders need to be named ,Box_i' for any integer i.	CONVERT_WHOLE_FOLDER = True
ALREADY_SORTED	Bool	• True: the video files per night were already moved to a subfolder (not standard). • False: all .asf files per channel are contained in one folder without subfolders, the script creates different files per night and per day automatically.	ALREADY_SORTED = False
ANIMAL_NAME	String	• Species	ANIMAL_NAME = ,Wildebeest'
VIDEO_NUMBER	String	• Videonumber, as a string! • Only used if CONVERT_WHOLE_FOLDER = False	VIDEO_NUMBER = ,1'
ZOO_NAME	String	• Name of Zoo, may not contain special characters beside -.	ZOO_NAME = ,FancyZoo'

ORIGINAL_RES	Bool	• True: original resolution of .asf files is kept. • False: rescaling to a width of MAXIMAL_WIDTH.	ORIGINAL_RES = False
MAXIMAL_WIDTH	Int	• Upscaling is not possible, only downscaling!	MAXIMAL_WIDTH = 1280
ORIGINAL_FPS	Bool	• True: Keeps the fps of the .asf files. • False: Output has OUTPUT_FPS many frames per second.	ORIGINAL_FPS = False
OUTPUT_FPS	Int	• How many fps should the output video have? Needs to be smaller or equal to the original fps of the .asf files and a divisor.	OUTPUT_FPS = 1
NIGHT_TIME_BEGIN	String	• Time to which a night recording starts, will be used to detect the time in LUPUS structure, e.g.: • HVCR_ch3_main_20181128**17**0000_20181128180000.asf	NIGHT_TIME_BEGIN = ‚17'
DAY_TIME_BEGIN	String	• Time to which a day recording starts, will be used to detect the time in LUPUS structure, e.g.: • HVCR_ch3_main_20181128**07**0000_20181128080000.asf • If no day recordings are conducted, this can be set to the empty string ''.	DAY_TIME_BEGIN = ‚07'
		Testing variables	
KI_EXPORT	Bool	• True: Additionally, a grayscale variant is created.	KI_EXPORT = False
TEST_PROC	Bool	• True: If set to true, only the first 10 seconds of each videofile (.asf) will be used, this allows to test the program / structure in short time.	TEST_PROC = False

Requirements

Python

- Python >= 3.5
- numpy, opencv

Structure

- .asf-files as given through some variants of LUPUS or Technaxx:
  - PREFIX_CHANNEL_FILLWORD_YYYYMMDDHHMMSS_YYYYMMDDHHMMSS.asf
  - E.g.: HVCR_ch3_main_20181128100636_20181128110001
  - E.g.: LUPUS_ ch3_main_20181128100636_20181128110001