# **Video Processing Script Documentation**

## **Overview**

The Python script is designed to process input video files, extract frames from specified intervals, and save the frames as images. Additionally, it generates a CSV file to store information about each processed frame, including details such as frame name, video name, interval, frame number, class, type, and file path.

## **Dependencies**

The script relies on the following libraries:

* cv2 (OpenCV): For video capture, frame extraction, and image processing.
* os: For handling file and directory operations.
* csv: For working with CSV files.
* numpy: For numerical operations.
* keras.preprocessing.image.ImageDataGenerator: Currently commented out, but can be used for image data augmentation.

Ensure that these dependencies are installed before running the script. You can install them using the following:

pip install python-csv opencv-python numpy keras

## **Functions**

### **1. normalize\_image(image)**

Input: image: A NumPy array representing an image.

Output: A normalized version of the input image with pixel values in the range [0, 1].

Description:

This function normalizes the pixel values of an image to the range [0, 1]. It divides each pixel value by 255.0 to achieve normalization.

### **2. standardize\_image(image)**

Input: image: A NumPy array representing an image.

Output: A standardized version of the input image using mean and standard deviation.

Description:

This function standardizes the pixel values of an image using the mean and standard deviation calculated with the cv2.meanStdDev function. It subtracts the mean and divides it by the standard deviation.

### **3. video\_to\_frames(video\_path, output\_root\_folder, start\_time, end\_time, class\_name, type, interval\_length=3, output\_fps=14, size=(640, 480))**

Inputs:

* video\_path: Path to the input video file.
* output\_root\_folder: Root folder where processed frames will be saved.
* start\_time: Start time for frame extraction (in seconds).
* end\_time: End time for frame extraction (in seconds).
* class\_name: Class label for the video.
* type: Type label for the video.
* interval\_length: Length of each interval for frame extraction (default is 3 seconds).
* output\_fps: Frames per second for the output video file (default is 14 fps).
* size: Tuple representing the desired size of each frame (default is (640, 480)).

Description:

This function processes a video

extracts frames within specified intervals, and saves them as images. It also records frame information in a CSV file. The function performs the following steps:

Opens the input video file using OpenCV's VideoCapture.

Calculates the total number of frames, frames per second (fps), and the number of intervals based on the specified parameters.

Iterates through each interval, extracts frames, and saves them as images in the output folder.

Writes frame information, file names, and paths, to a CSV file.

Optionally performs normalization, resizing, and image augmentation (commented out) on each frame.

Outputs:

* Processed frames are saved in the specified output folder.
* Frame information is recorded in a CSV file.

### **4. process\_videos\_in\_folder(input\_folder, output\_root\_folder, video\_info)**

Inputs:

* input\_folder: Path to the folder containing input video files.
* output\_root\_folder: Root folder where processed frames and CSV files will be saved.
* video\_info: A list of tuples containing video information (filename, start time, end time, class name, type).

Description:

This function processes multiple videos by calling the video\_to\_frames function for each video based on the provided information. It iterates through the list of videos and their corresponding intervals, checking for the minimum interval requirement, and saves frames within each 3-second interval.

Outputs:

* Processed frames and CSV files are saved in the specified output folder.