CSS 484 – Multimedia Data Processing

Name: Anne Ha

**Assignment 4**

**VIDEO SHOT BOUNDARY DETECTION SYSTEM**

# RUN THE PROGRAM

## Directory Structure

**SampleCodePython**

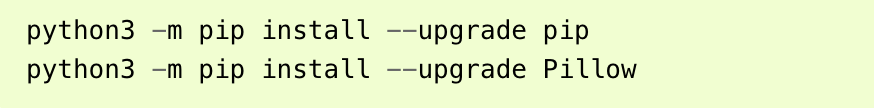
* **Frames**: store frames (1000 – 4999) extracted from the video as ***.jpg*** files for inspection
* **Videos**: store the video for analysis (***.mpg*** file was used)
* **PixInfo.py**: evaluate one frame at a time and add it into ***imageList***. For each image, ***encode()*** iterates through its pixel vector, calculate intensity level of each pixel based on Intensity method and store in ***InBins*** array.
* **ImageViewer.py**: extract frames from video file which will be used to detect video shots by determining cuts and gradual transitions using Twin-Comparison Algorithm. The file also creates layouts for the Video Analysis Tool.
* **.vscode**: set up debugging tool for validation.

## Install the Program

To run the program, your computer needs to have Python and required libraries installed.

* Tkinter and Pillow for image processing
* Cv2 for video processing
* Numpy for math calculation

Since Tkinter, cv2, numpy are built-in GUI frameworks in Python3, you may need only install Pillow with commands as below:



## Execute the Program

Once everything is set up, run the program with the command below:



# USE THE SYSTEM

## Read Video and Extract Frames

With the support of cv2 library, ***read\_video()*** reads the given video “videos/20020924\_juve\_dk\_02a.mpg” and extract frames from 1000 – 4999. These 4000 frames will be shown on the Video Analysis Tool window when the program is executed.

A screenshot of a video analysis tool

Description automatically generated

You can find these frames in the ***frames*** folder with the format “{i}.jpg” where i: 1000 ~ 4999.

A screenshot of a computer

Description automatically generated

## Detect Video Shots Using Twin-Comparison Algorithm

By clicking Intensity button, you can see all the first frames of the shots detected from the video in the sequence order. These shots are determined using Twin-Comparison Algorithm which is implemented in ***find\_distance()*** and ***check\_thresholds ()****.*

* The ***find\_distance()*** gets frame-to-frame difference by calculating Manhattan distance between an image and the next one in the ***imageList*** based on their Intensity feature vector ***InBins***, building SDs matrix.
* The ***check\_thresholds()*** calculates Tb, Ts based on the derived SDs using Numpy library and then uses these values to detect the cuts (Cs, Ce) and gradual transitions (Fs, Fe) in the video.

The frames presenting the cuts ***(Cs, Ce)*** and gradual transitions ***(Fs, Fe)*** will be printed in the terminal, along with the shot information when considering ***Ce, Fs + 1*** as the first frame of each shot and ***Cs, Fs*** as the end of its previous shot.

A black screen with numbers

Description automatically generated

### View the Shots

There are 27 shots detected whose first frames will be displayed in the Result Viewer window in a sequence order.

A screenshot of a video game

Description automatically generated

Now, you can view the shots by clicking their corresponding first frame. The shot will be played in a new window.

A screenshot of a video of a football game

Description automatically generated

# DISCUSSION

In terms of first frame results, while there is a match in ***Ce***, there are 4 differences in ***Fs + 1*** when comparing my program with the provided values. This is acceptable because the two programs may use different libraries, programing languages or video input.

Debugging tool is already set up to check these results.

Tb = 97995.73707960772

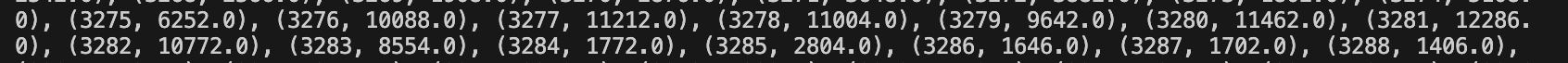
Ts = 7429.240310077519

Tor = 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Given results** | | **My results** | | **Match** | **Evidence** |
| **Ce** | **Fs + 1** | **Ce** | **Fs + 1** |
| 1091 |  | 1091 |  | Y |  |
| 1112 |  | 1112 |  | Y |  |
| 1575 |  | 1575 |  | Y |  |
| 1618 |  | 1618 |  | Y |  |
|  | 1865 |  | 1865 | Y |  |
| 1926 |  | 1926 |  | Y |  |
| 2332 |  | 2332 |  | Y |  |
|  | 2406 |  | 2406 | Y |  |
| 2584 |  | 2584 |  | Y |  |
|  | 2676 |  | 2676 | Y |  |
| 3008 |  | 3008 |  | Y |  |
| 3050 |  | 3050 |  | Y |  |
|  | 3200 |  | 3200 | Y |  |
|  | ***3276*** |  | ***Missing*** | **N** | SDs [3276-3283] = 85020 < Tb [1] |
| 3532 |  | 3532 |  | Y |  |
|  | 3551 |  | 3551 | Y |  |
|  | 3624 |  | 3624 | Y |  |
|  | 3765 |  | 3765 | Y |  |
|  | 3838 |  | 3838 | Y |  |
|  | 3928 |  | 3928 | Y |  |
|  | 4042 |  | 4042 | Y |  |
|  | ***Missing*** |  | ***4299*** | **N** | SDs [4298-4299] = 193878 > Tb [2] |
| 4358 |  | 4358 |  | Y |  |
| 4484 |  | 4484 |  | Y |  |
|  | 4561 |  | 4561 | Y |  |
|  | ***4604*** |  | ***4608*** | **N** | SDs [4603] = 8328 > Ts  SDs [4604] = 6948 < Ts  SDs [4605] = 3324 < Ts  SDs [4607-4619] = 363608 > Tb [3] |
|  | 4776 |  | 4776 | Y |  |
|  | ***4892*** |  | ***Missing*** | **N** | SDs [4891] = 7218 < Ts [4]  SDs [4893-4896] = 49472 < Tb |
|  | 4986 |  | 4986 | Y |  |

[1]

A screenshot of a computer

Description automatically generated

[2]

A screenshot of a computer

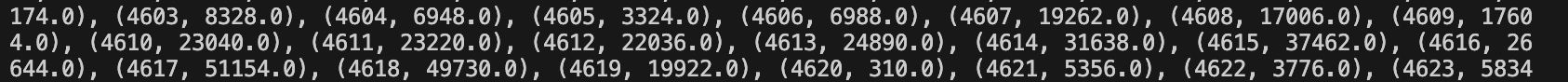
Description automatically generated



[3]

A screenshot of a computer

Description automatically generated



[4]

A screenshot of a computer program

Description automatically generated

