# Image Processing - Exercise 1

Achsaf Atzmon, achsaf, 316129501

## Introduction

(a) In your own words, state the goal of the exercise and what was the main technique (i.e. an idea or concept you've learned in class, not a technical tool like numpy) you've used to solve it.

In this exercise we were given short Videos with the goal to find where a scene change occurs. I used Cumulative sum to find the "signature" of each frame, and then found the frames with the biggest difference.

(b) Briefly specify the differences between the two categories of videos (category1 and category2) and how these differences may affect the approach.

Category 1 was much cleaner and easier to identify the cut. There was only one major spike. Category two had some distortion/noise mixed in, leading to multiple spikes in the cumsum-diff. The approach was the same, since these spike were still not as significant as a complete scene change

## Algorithm

(a) Clearly describe the steps involved in your scene cut detection algorithm (i.e. describe the conceptual steps and building blocks, e.g., if the algorithm is how to make coffee, the steps could be 1) Heat water, 2) Put coffee in glass, 3) Pour water, 4) Add sugar, 5) Add milk).

1. Convert each frame to greyscale
2. Calculate the histogram of each frame
3. Calculate the cumulative sum of each histogram
4. Normalize to avoid massive differences in values for similar images
5. Calculate the area under the cumulative sum
6. Find the two frames with the biggest difference in their cumulative sum area

(b) Clearly describe any modifications or adjustments made between the first video category and the second one.

No adjustments were made between the two categories.

# Implementation Details

(a) Describe your implementation of the algorithm (i.e. describe the actual implementation of the steps from section 2.a, using the same coffee example, here you would describe how much coffee, how much sugar, water to milk ratio, etc.).

1. Convert the video to a tensor
2. Multiply each color channel by the tenacity vector ([0.2989, 0.5870, 0.1140]) and taking the mean to find the greyscale value
3. In a loop for each frame:
   a) Calculate the histogram using np.histogram()
   b) Calculate the cumulative sum of the histogram using np.cumsum()
   c) Normalize the value by dividing by the final value
   d) Calculate the area under the normalized cumsum
   e) Save into vector
4. Find the biggest difference of area of the normalized cumsum of two frames using np.diff
5. Return found consecutive frames

(b) Specify the parts that you implemented from scratch and those that you've used functionality from an existing library. What libraries did you use and why did you choose these?

Scratch:

Color to greyscale conversion
Normalized cumsum

Exisiting libraries:

Numpy, mediapy
I used these libraries to open and manipulate the videos, they have many easy to use and efficent funtions that help me do the basic operations learned in class

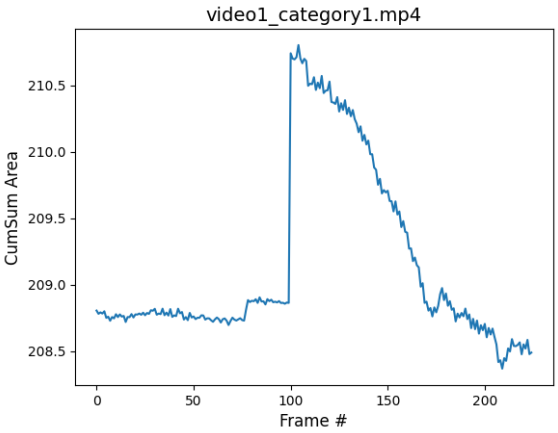
(c) Describe any necessary hyper-parameters, thresholds, or other choices used in your algorithm.

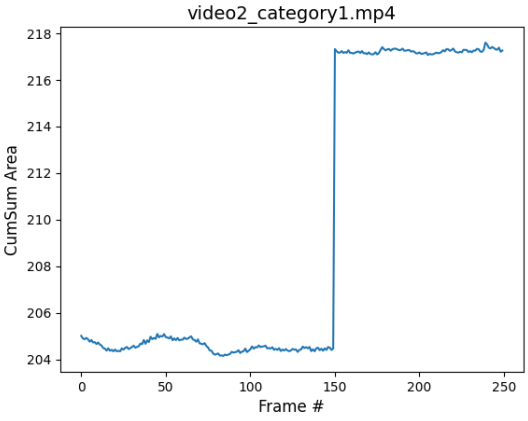
There are none

(d) Discuss any challenges faced during implementation and how they were addressed.

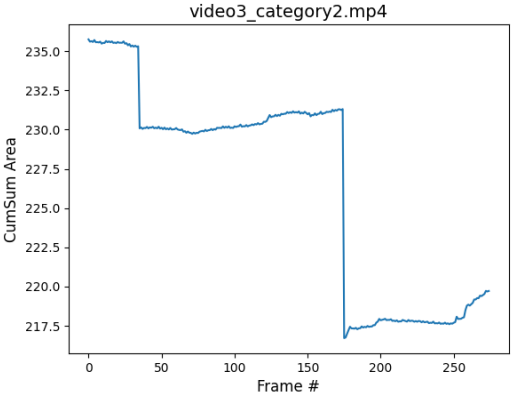
Normalizing the cumsum was not immediately obvious and threw off the data. Realizing that the actual values are less important than the overall shape meant we can normalize without losing data but still being able to compare images.

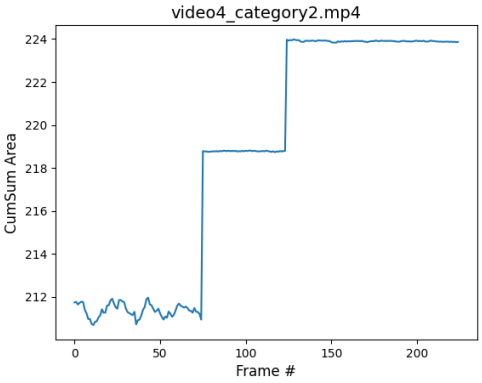
# Category 1 Results



# Category 2 Results




It is clear to see that there are two major jumps in category 2 videos by how the graph jumps/drops dramatically in value.

# Conclusion

Cumulative Sum is a very simple method for describing how similar two pictures are. We were shown in the recitation that you can “recolor” one image to have the same CumSum as another, so this is not an end all be all solution, but considering it’s simplicity it is remarkable effective.

We can use CumSum as a quick and dirty method for finding a “signature” for images.