Individual assignment 1:
From basic image processing towards object detection

Computer Vision (H02A5a)

The goal of this individual assignment is to apply some basic image processing techniques to detect an object and produce a one minute video for submission. You will have to submit this video output and your code by **wednesday 20 March 23:59**.

1 Overview

In this session, you are the director, scenario writer, producer and at the same time actor of a new **short film of one minute**. The jury (your TAs) will analyse your piece of art and give you a score that contributes to your final grade (score is dependent on technical performance and following the assignment guidelines). Your first step as a producer is the pre-production, we advise you to first read through this assignment carefully, think about the story line, the props you'll be using and which effects you'll be applying. In the production step you do the recording (use your phone/webcam), note that the footage should contain at least some object(s). In the post-production you add effects, show off some techniques and finally stitch all scenes together into a one minute blockbuster. Provide subtitles to describe which techniques you apply where, this helps to draw the attention of the jury members to the important bits.

Your final film should be structured as followed:

- 0s-20s: Use basic image processing techniques to provide special effects (Sect. 2).

- 20s-40s: Use a round colored object as the key figure and perform some object detection (Sect. 3).

- 40s-60s: Do whatever it takes to end with a bang (Sect. 4).

If you didn't have a chance to look at Individual Assignment 0, do it now! You need to use Python and the OpenCV library for the implementation. The actual video capture can be done using standard video recording tools (e.g. your smartphone or PC video recording software) and the video processing should be done in OpenCV[1]. You can process the video *offline* and simply process it frame by frame or experiment using temporal information.

---

[1]You can find some skeleton python code for reading and writing video files using OpenCV in this repository: https://github.com/gourie/opencv_video.

Lastly, while most of the computation will be rather efficient, it can be a good idea to downsample your video right from the start. You will anyway have to make sure your video is smaller than 30 MB before you upload it on Toledo (see Sect. 5). It's up to you now, good luck!

2   Basic image processing: 0 - 20s

Apply different techniques throughout your video and mention which technique is being used and what it does in the subtitles:

1. Switch the movie beween color and grayscale a few times (±4s).

2. Smoothing or blurring is a simple image processing technique often used to reduce noise. Experiment with Gaussian and bi-lateral filters and increase the effect of the blurring by widening your filter kernel. Clearly explain the difference between the two in the subtitles (±8s).

3. *Grab* your object in RGB and HSV color space. Show binary frames with the foreground object in white and background in black. If you carefully choose your object (i.e. with a distinct color compared to the rest of the scene), this can be a simple thresholding operation. Choose a color space and try to improve your *grabbing* (e.g. fill holes, undetected edges) by using binary morphological operations. Put the improvements in a different color (±8s).

3   Object detection: 20 - 40s

At this point the audience should already be in ecstasy, however, you still have more in store. Based on your previous experience with grabbing the object, you might have noticed that in most cases intensity/color values are not distinctive. Now, improve performance for grabbing an object by first building features that capture properties of the object. Then, use these features to detect the object. You will explore edge detection, detection of particular shapes (here, circles) and finally use feature descriptors to detect an object of interest via a technique called template matching. Here are some things the audience is a fan of:

1. Make use of the Sobel edge detector to detect horizontal edges. Do the same for vertical edges. Visualize the edges for 5s, it is up to you to find a nice way to visualize the detected edges (try to use some color). During these 5s, stick to your vizualization method but tweak the parameters of the Sobel detector to show how the edge detection changes (±5s).
   While Sobel is the easiest and most natural edge detector, more advanced versions exist, an example is the Canny edge detector. Feel free to try out, in a similar way as the Sobel edge detector, however, it should not be part of the video.

2. Make sure you understand how the Hough transform works, use it to detect the shapes in your scene which are close to circular and visualize the detected circles by flashy

contours overlayed on your original color video (not on the detected edges). There are many parameters for the Hough tranform. You have 10s to show how the choice of parameters influences the detection results (±10s).

3. Now it's time to introduce your object and put it at a certain position in your scene. For the first 2s of this interval, draw a flashy rectangle around the object, drawing the viewer's attention to its location. The next 3s of the video must be gray scale, with the intensity values proportional to the likelihood of the object of interest being at that location (±2s+3s). Thus white means that (the center of) the object is at that particular location with 100 % certainty, while black means the opposite.
In order to come up with such a gray scale map, you can compute features in the same way as you did for your object of interest, but now at each location. Computing, e.g. the mean squared error between the two feature representations will give you an inversely proportional likelihood.

4   Carte blanche: 40 - 60s

Now it is your time to shine, a complete 20s freestyle, attempt to capture the attention of the audience using advanced techniques. You can of course invent your own *tricks* but for the first part focus on trying to demonstrate how you are able to grab an object as accurately as possible and influence the video scene using this object. An example could be to detect and follow a coloured ball in the video. You can make the ball *invisible* or replace it automatically/digitally with another object.
Or you could e.g. show off by adding extra magic to your scene. Are you able to detect other object(s) of interests (e.g. your eyes)? Can you manipulate objects of interest (e.g. change color, move them to other places in your video, draw boxes around them). Are you able to sharpen an object that was captured out-of-focus?
You could also come up with something nice, demonstrate the robustness of your technique. For example you can demonstrate how your method is invariant to illumination or rotation, put the object at a different spot in the scene (further away so its scale changes), walk around with the object of interest. Does your algorithm still work?
It is not the goal to spend hours on this carte blanche section; just pick one (or a few) and show us what you found out about it. Make sure to provide subtitles for clarity!

5   Submission

Submit the video and your code separately on Toledo before the deadline. To do so, first make sure to compress and upload the video under MPEG-4 (MP4) format. Then, also upload all your code as a single compressed ZIP-file. The video you upload must be smaller than 30 MB in size. Remember that you can choose to downsample your video right from the start, or work on full resolution and downsample it just before compressing and uploading it on Toledo. **Submissions with other compression than ZIP for the code and/or with videos larger than 30 MB and/or in a different format than MP4**

will not be reviewed!. Make use of the Individual assignment 1 forum/discussion board on Toledo if you have any questions.