

Exercise2b - Change detection in videos

The goal of this exercise is to create a small program for real-time change detection using OpenCV.

Learning Objectives

After completing this exercise, the student should be able to do the following:

1. Use OpenCV to access a web-camera or the camera or a mobile phone.
2. Use the OpenCV function `cvtColor` to convert from color to gray scale,
3. Convert images from integer to floating point using the `img_as_float` function.
4. Convert image from floating point to uint8 using the `img_as_ubyte` function.
5. Compute a floating point absolute difference image between a new and a previous image.
6. Compute the frames-per-second of an image analysis system.
7. Show text on an image using the OpenCV function `putText`.
8. Display an image and zoom on pixel values using the OpenCV function `imshow`.
9. Implement and test a change detection program.
10. Update a background image using a linear combination of the previous background image and a new frame.
11. Compute a binary image by thresholding an absolute difference image.
12. Compute the total number of changed pixels in a binary image.
13. Implement a simple decision algorithm that is based on counting the amount of changed pixels in an image.

Installing Python packages

In this exercise, we will be using the popular *OpenCV* library to perform real-time image analysis.

We will use the virtual environment from the previous exercise (`course02502`). Start an **Anaconda prompt** and do:

```
activate course02502
conda install -c conda-forge opencv
```

You might also need to install **Numpy**:

```
conda install -c anaconda numpy
```

Exercise data and material

The data and material needed for this exercise can be found here: (<https://github.com/RasmusRPaulsen/DTUImageAnalysis/blob/main/exercises/ex2b-ChangeDetectionInVideos/data/>)

Start by creating an exercise folder where you keep your data, Python scripts or Notebooks. Download the data and material and place them in this folder.

OpenCV program for image differencing

In the exercise material, there is a Python script using OpenCV that:

1. Connects to a camera
2. Acquire images, converts them to gray-scale and after that to floating point images
3. Computes a difference image between a current image and the previous image.
4. Computes the frames per second (fps) and shows it on an image.
5. Shows images in windows.
6. Checks if the key `q` has been pressed and stops the program if it is pressed.

It is possible to use a mobile phone as a remote camera by following the instructions in Using a mobile phone.

Note that we sometimes refers to an image as a *frame*.

Exercise 1: *Run the program from the exercise material and see if shows the expected results? Try to move your hands in front of the camera and try to move the camera and see the effects on the difference image.*

Exercise 2: *Identify the important steps above in the program. What function is used to convert a color image to a gray-scale image?*

Change detection by background subtraction

The goal of this exercise, is to modify the program in the exercise material, so it will be able to raise an alarm if significant changes are detected in a video stream.

The overall structure of the program should be:

- Connect to camera
- Acquire a background image, convert it to grayscale and then to floating point
- Start a loop:
 1. Acquire a new image, convert it to grayscale and then to floating point: I_{new} .

2. Computes an absolute difference image between the new image and the background image.
3. Creates a binary image by applying a threshold, T , to the difference image.
4. Computes the total number of foreground, F , pixels in the foreground image.
5. Compute the percentage of foreground pixels compared to the total number of pixels in the image (F).
6. Decides if an alarm should be raised if F is larger than an alert value, A .
7. If an alarm is raised, show a text on the input image. For example **Change Detected!**.
8. Shows the input image, the background image, the difference image, and the binary image. The binary image should be converted to uint8 using `img_as_ubyte`.
9. Updates the background image, $I_{\text{background}}$, using:

$$I_{\text{background}} = \alpha * I_{\text{background}} + (1 - \alpha) * I_{\text{new}}$$

10. Stop the loop if the key `q` is pressed.

You can start by trying with $\alpha = 0.95$, $T = 0.1$, and $A = 0.05$.

Exercise 3: *Implement and test the above program.*

Exercise 4: *Try to change α , T and A . What effects do it have?*

The images are displayed using the OpenCV function `imshow`. The display window has several ways of zooming in the displayed image. One function is to zoom x30 that shows the pixel values as numbers.

Exercise 5: *Try to play around with the zoom window.*

Exercise 6: *Use `putText` to write some important information on the image. For example the number of changed pixel, the average, minimum and maximum value in the difference image. These values can then be used to find even better values for α , T and A .*

Also try to find out how to put a colored text on a color image. Here you need to know that OpenCV stores color as BGR instead of RGB.

Using a mobile phone camera (optional)

It is possible to use a mobile phone as a remote camera in OpenCV.

You need to install a web cam app on your phone. One option is **DroidCam** that can be installed from Google Play or from Apple App Store.

The computer and your phone should be on the same wireless network. For example one of the DTU wireless networks.

Now start the DroidCam application on your phone. It should now show an web-address, for example `http://192.168.1.120:4747/video`

Use this address, in the program:

```
use_droid_cam = True
if use_droid_cam:
    url = "http://192.168.1.120:4747/video"
cap = cv2.VideoCapture(url)
```

You should now see the video from your mobile phone on your computer screen. Remember you phone should be unlocked when streaming video.

Notes on OpenCV and MacOS

Viktor Holmenlund Larsen had problems using MacOS (Big Sur 11.2.3) and fixed it by changing `cv.VideoCapture(0)` to `cv.VideoCapture(1)`. Also by updating `Gstreamer` and its dependencies and deinstalling and reinstalling OpenCV after.