INF6804 Computer Vision
H2020 – Practical Assignment 1
Video Segmentation

Objectives:

- Introduce the processing of video and image data using computer vision libraries.
- Learn and apply video segmentation based on optical flow.
- Learn and apply video segmentation based on object detection by classification techniques.
- Explain the advantages and shortcomings of both approaches.

Submission:

- *All your source code* (we should be able to run your tests)
- A report (*.pdf format* of 8 to 15 pages with font size of 10)
- Submit before Feb. 7th, 5:00 PM, on Moodle –*late submissions will not be accepted*
- *You must also submit your report on TurnItIn*
  - Register at www.turnitin.com using the info available on Moodle!

References:

- See course notes on Moodle (Chapter 2)

Other directives:

- The assignments must be made in teams of two, submit only one version of your work!

Presentation

In this assignment, you will have to characterize two methods used for the identification of regions of interest in video sequences, and determine which method is better, and in which circumstances. A description of your work, your experiments, and answers to the questions outlined in this document must be included in your report.

In this assignment, you will have to compare two foreground/background video segmentation methods, namely an optical flow method and an object detection by classification method. These methods should be briefly presented in class — you can use your course notes as a reference to understand their basic working principles. For more details, go look online!

In your report, you have to include the following elements (marked on 20 pts):

1. Presentation of the two methods (4 pts) :

   In your own words, give the general description and principles of your two methods.

2. Performance hypotheses in specific use cases (3 pts) :

   Identify, based on your theoretical understanding of the two methods, which one should be the best of the two in at least THREE specific use cases. For example, which is the best method to use if the camera is not fixed? Why? And if the contrast is low?

3. Description of experiments, datasets and evaluation criteria (2 pts) :

   Describe in detail the experiments realized to test the hypotheses of the previous point. Which dataset did you use? What are the difficulties in this dataset's videos? Which evaluation criteria did you use?

4. Description of the implementations used (2 pts) :

   Describe the implementation of the two studied methods. If you did not write all the code yourself, where does it come from? Did it require modifications? Otherwise, from which papers or websites did you inspire yourself to write it? In all cases, what are the primary parameters of your methods? How did you set their values?

5. Experimentation results (3 pts) :

   Provide the evaluation results from your experiments related to the hypotheses of the first point. Use a proper format for their presentation — tables, figures, ...

6. Discussion on results and prior hypotheses (3 pts) :

   Discuss the results of the fourth point in relation with the hypotheses of the first point. Which hypotheses are supported by these results? Which are not? Which test resulted in a lack of conclusion? How could you improve these tests?

7. Readability and completeness (3 pts) :

   In addition to the content, the format must be clean and complete.

You can use any dataset you want to compare both methods, but if you do not know any, the CDNET dataset is a good one for this assignment.

During the lab periods, do not hesitate to ask questions to the TAs — they can help you with any technical issue if you are working on Windows/Linux, or if you are coding in C/C++, Python or Matlab.

You will be penalized by 50% of the total grade if you do not hand in your code. Also, if your report is not submitted to TurnItIn, it will not be graded. The order of presentation for the topics listed above does not matter, as long as they are all present.

Resources

Datasets:

- CDNET dataset (http://changedetection.net/)

Vision libraries:

- OpenCV (https://docs.opencv.org/4.0.0/d9/df8/tutorial_root.html)
- scikit-image (https://scikit-image.org/docs/stable/auto_examples/index.html)

Deep learning frameworks:

- PyTorch (https://pytorch.org/tutorials/)
- Tensorflow (https://www.tensorflow.org/tutorials)

Python:

- Guide (https://wiki.python.org/moin/BeginnersGuide/Programmers)
- NumPy (https://docs.scipy.org/doc/numpy/user/quickstart.html)

Matlab:

- Guide (http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf)
- Cheatsheet (http://web.mit.edu/18.06/www/Spring09/matlab-cheatsheet.pdf)