# Project Module 2:
# Visual Object Tracking

## Project tasks

**MANDATORY tasks:**

1. Implement and test the MOSSE tracker (for greyscale input).
2. Extend the standard MOSSE tracker to a multi-channel MOSSE tracker to incorporate multi-dimensional features. Test it on hand-crafted features.
3. Extend the feature extraction code to use deep features in the multi-channel MOSSE tracker. Evaluate different layers in the deep network.
4. Summarize your overall findings and results in a short technical report (Deadline: by the end of the year).

**Further optional tasks:**

1. Evaluate different hand-crafted features in the multi-channel MOSSE tracker.
   E.g. HOG, colour, colour names, greyscale, gradient filter bank.
2. Implement a scale estimation component to handle scale variations in the multi-channel MOSSE tracker.
3. Improve the speed of your tracker. One option is to employ feature dimensionality reduction methods. You could also try switching to a more efficient FFT than the one in numpy/scipy.
4. Improve the accuracy of your tracker. See reference papers for ideas.
5. Evaluate your tracker using the VOT2020 Toolkit, see `https://www.votchallenge.net/howto/tutorial_python.html`.

## Organization

The projects are conducted in groups of 4 or 3 students (in order of preference).
Each member should contribute to the overall project.

## Project Examination

In order to pass, each group should:

- Complete *at least the mandatory* tasks in the project.
- Write the Technical Report of the project and submit it on time.

## Utility Code

- Skeleton code is available in OneDrive in the folder `TrackingProject`.
- The Skeleton code is written in Python and uses numpy.
- Provided Items:
  - A baseline visual tracker: NCC + Code
  - Object tracking dataset (Mini-OTB)
  - Code to evaluate your own tracker
  - Feature extraction code for e.g. Colour names and AlexNet.

## Dataset: Mini-OTB

The Mini-OTB dataset is a subset of the OTB benchmark. The dataset is available in the course OneDrive folder.

## Technical Report

Length: approximately 4 pages including figures and references.

Contents:

1. Introduction and description of the problem that is solved.
2. Description of experimental methods and how the problem is solved.
3. Description of dataset and evaluation protocol.
4. Experimental results: both quantitative and qualitative results.
5. Analysis and discussion of the experimental results.
6. Conclusion and References.

Format: The report should be submitted as a PDF.

Template: Use the following report template:
`http://cvpr2017.thecvf.com/files/cvpr2017AuthorKit.zip`

## Resources

**Supervision**:

Due to the open schedule of the course, systematic supervision during the project is difficult to organize. Please try to help each other. You may also want to ask WASP students at CVL. If you get stuck entirely, please contact me.

**Computational resources**:

Please check with your supervisor if your ordinary desktop machine is not sufficient.

## Reference Papers


1. David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui: *Visual object tracking using adaptive correlation filters.* CVPR 2010.
2. Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, Joost van de Weijer: *Adaptive Color Attributes for Real-Time Visual Tracking.* CVPR 2014.
3. Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, Michael Felsberg: *Accurate Scale Estimation for Robust Visual Tracking.* BMVC 2014.
4. Y. Li and J. Zhu: *A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration.* ECCV 2014.
5. Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, Michael Felsberg: *Convolutional Features for Correlation Filter Based Visual Tracking.* ICCV Workshops 2015.
6. J. F. Henriques et al.: *High-Speed Tracking with Kernelized Correlation Filters*, IEEE TPAMI 2015.


**PDFs of all reference papers are available in the course OneDrive folder.**