UNIVERSITEIT TWENTE.

# Data Science [201400174]

Course year 2022/2023, Quarter 1B

DATE
November 11, 2022

EXCERPT

## Computer vision and image classification [CV&IC]


TEACHERS
Faizan Ahmed
Ellen-Wien Augustijn
Nacir Bouali
Faiza Bukhsh
Rolf de By
Karin Groothuis-Oudshoorn
Maurice van Keulen
Mahdi Khodadadzadeh
Elena Mocanu
Estefania Talavera
Brenda Voorthuis
Shenghui Wang

COURSE COORDINATOR
Karin Groothuis-Oudshoorn (quartile 1A)
Maurice van Keulen (quartile 1B)
Faizan Ahmed (quartile 2A)

PROJECT OWNERS
Faiza Bukhsh
Karin Groothuis-Oudshoorn
Maurice van Keulen
Elena Mocanu
Mannes Poel
Michel van Putten
Mohsen Jafari Songhori
Luc Wismans

Topic 9

# Computer vision and image classification [CV&IC]

## 9.1 Introduction

Topic teachers: Estefania Talavera

In this course, we discuss concepts such as computer vision, image analysis, feature extraction, supervised learning, deep learning and image classification. In the assignments, you will learn about some basic tasks in image processing, computer vision and machine learning. The assignments will also introduce you to some helpful tools in the field.

### 9.1.1 Study material and tools

A very comprehensive resource is the online Pyhton and computer vision book 'Programming Computer Vision with Python'
https://github.com/Ricky-Wilson/Programming-books/blob/master/PDF/OReilly.Programming.Computer.Vision.with.Python.Jun.2012.RETAIL.eBook-ELOHiM.pdf.
Note: Links to those resources are given in the details of each assignment.

Material for beginners and more;

- Brief introduction to the **Python** programming environment at
  https://www.python.org/about/gettingstarted/
- **Scikit-learn** is a library of machine learning packages for Python. Check more information at
  https://scikit-learn.org/stable/
- **Pytorch** is an open source machine learning library based on the Torch library, used for applications such as computer vision. You will be expected tu use deep learning packages from this library. Check more information at
  https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
- Deep learning book
  https://www.deeplearningbook.org/

### 9.1.2 Deliverables and obligatory items

The detailed descriptions of the deliverables are shown in each assignment. Upload an ipython notebook (.ipynb) with your solutions and also a PDF printout (File ! Print Preview in the ipython Notebooks) for each assignment separately.

## 9.2 Description of the practical assignments

### 9.2.1 Data set

You will need the following data sets. All of them are available in PyTorch in Python, which means that there is no need to download them manually
(see `https://pytorch.org/vision/stable/datasets.html`).

- COCO
  (see `https://cocodataset.org/#home`)
- LSUN
  (see `https://www.yf.io/p/lsun`)
- CIFAR-100
  (see `https://www.cs.toronto.edu/~kriz/cifar.html`,
  Github code `https://github.com/kuangliu/pytorch-cifar`)

Other datasets that you might find interesting and that you can use to train and test your models are:

- Places 365-Extra69 (see
  Paper: `http://places2.csail.mit.edu/index.html`,
  Dataset: `http://places2.csail.mit.edu/download.html`).
- Tiny ImageNet (see
  Paper: `http://cs231n.stanford.edu/reports/2017/pdfs/930.pdf`,
  Dataset: `http://cs231n.stanford.edu/tiny-imagenet-200.zip`).

It is recommended to follow the introductory lecture to Python to learn and refresh basic Python programming concepts.

### 9.2.2 Prerequisites

For the code examples to work we need python, its scientific libraries and a jupyter notebook server.

- Basic knowledge on data mining.
- Working on your pc: There are many possibilities to install python, SciPy, Scikit-learn and the Jupyter notebook server. We recommend using the 'all-in-one'-solution Anaconda, which is available for all (major) platforms, including Linux, OS X and Windows. Anaconda can be downloaded from https://www.anaconda.com/download. Once installed you can run Jupyter notebook from your terminal. The notebook opens a browser window in which you could load an existing notebook or write your code from scratch.
- Working on the jupyter server of the UT: An option is to run your experiments in the . You can find information about the server in its wiki page at `https://www.utwente.nl/en/service-portal/research-support/research-support-topics/it-facilities-for-research/jupyter-jupyterlab`

### 9.2.3 Assignments overview

The assignments will be provided in a jupyter notebook format. Please, check the notebooks for the instruction of the different assignments.

#### Assignment 0

Set of Jupyter notebooks with examples on how to read images and work with them {v1: Numpy Intro, v2: Skimage Intro, v3:Basic Image manipulation}.

Formative assignment, i.e. assignment 0 will not be assessed with fail or pass by the teaching staff. Assignment 0 is not connected to assignments 1 and 2. Assignments 1 and 2 are meant to be solved sequentially and therefore are related. You will be selecting the same dataset(s) to solve both assignments.

#### Assignment 1: Understanding the dataset

**Introduction:**
Large datasets can be overwhelming at the beginning. Working with a dataset is not only feeding it to a model. It is important to understand the properties of the samples and classes. Tables, plots and other visualization help understand the data you are working with and will help explain the challenges you may face. For instance, present an unbalanced set of classes, the features are not descriptive enough or identify mislabelled data.

**Task:**
Given a set of datasets, you need to describe them with tables and figures. To this end, you will extract simple features such as RGB descriptors. They will be visualized in a 2D and 3D space plot that helps observe the data samples distribution.

#### Assignment 2: Feature extraction and image classification

**Introduction:**
Supervised machine learning methods allow us to learn functions that map an input to an output based on example input-output pairs. Input samples are represented by a feature vector. It became common practise to rely on pre-trained networks to extract global descriptors from the images. These descriptors are later used in classification frameworks as baseline results to which more complex architectures are compared.

**Task:**
Given a dataset, you are asked to implement a train test dataset split. Later, you are asked to rely on traditional machine learning methods (e.g. k-Nearest Neighbour) for the classification of the test set. As images descriptors, you are asked to use a pre-trained network for the extraction of global features. Finally, you are asked to implement k-fold cross-validation to assess the performance of your chosen framework (selected model and extracted features).

#### [Optional] Assignment 3: Transfer learning by fine tuning pre-trained networks

**Introduction:**
Fine-tuning is the process in which parameters of a model must be adjusted very precisely in order to fit with certain observations. When working with deep networks, which are data-hungry, it is common to re-use gained knowledge in similar classification tasks, i.e. re-use pre-trained weights. This process is known as transfer learning, a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

**Task:**
You need to perform transfer learning by fine-tuning a pre-trained model on a new dataset. You need to create a train test dataset split, define the data augmentation techniques, the optimization parameters and finally, train your network. You will assess your model with k-fold cross-validation.

### 9.2.4 Some hints:

If you are not yet familiar with python, the following concepts should be mastered before diving into image analysis with python.

- Basic pyhton concepts: numbers, strings, loops
  (`https://docs.python.org/3/tutorial/introduction.html`)

and more complex control flow including basic knowledge about functions (https://docs.python.org/3/tutorial/controlflow.html).

- Data structures: especially lists, sets and dictionaries. Also lists of lists come in handy (tutorial here https://docs.python.org/3/tutorial/datastructures.html).
- Feature extraction:
  - Dimensionality reduction https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be
  - Colour information https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/
  - Object detection https://machinelearningmastery.com/object-recognition-with-deep-learning/
- You might also want to know some basics about exceptions to be able to figure out what went wrong when something went wrong (https://docs.python.org/3/tutorial/errors.html).