Computer Vision I (CSE 40535 / 60535)

Syllabus, Fall 2020
(ver. 08/06/2020)

Adam Czajka
University of Notre Dame

1 Goals and scope

The aim of Computer Vision is to give computers the ability to *understand* what they *see* in images and videos taken by one or more sensors (usually visible-light cameras). The goal of this course is to introduce methods for interpreting the visual information captured by machines.

The course is divided into **four parts**. In the **first part** we define the aims of computer vision, current expectations and successful applications, as well as its limitations. We also discuss selected biological vision mechanisms as an inspiration to create better computer vision algorithms. The **second part** explains the basics of signal processing from the computer vision perspective. This part includes image formation, image acquisition, understanding and effective use of color information, and image processing (filtering and segmentation). The **third part** focuses on automatic recognition of patterns. It covers feature extraction and selection, texture descriptors, Bayesian inference, classification and decision making. In this part we will also discuss how these tasks can be solved by deep learning techniques, especially convolutional neural networks. Finally, the **fourth part** considers multiple-view and geometry topics in vision. It consists of motion analysis, including object tracking, projective geometry, camera geometric model, camera calibration and 3D reconstruction.

After completing this course students will be able to understand computer vision literature, recognize frontiers of state-of-the-art computer vision systems, and select appropriate mathematical and software tools to develop algorithms solving the most important computer vision problems. Practical classes will utilize high-level programming languages (Python and Matlab) and popular computer visions tools and machine learning packages, such as OpenCV, Keras, Tensorflow or Pytorch, to illustrate in practice selected topics discussed in class. The goal of the semester project is to exercise the **entire computer vision pipeline** on the selected vision problem.

2 Prerequisites

- **Required:** basic skills in Python and Matlab programming (syntax, math operations, working with Python packages / Matlab toolboxes).
- **Good to have:** basics in image and signal processing, linear algebra, geometry and optics, and basic experience with OpenCV; these will help in faster adoption of the topics discussed in class and will make assignments easier.

3 Course structure

- 36 lectures, 50 minutes each (see Sec. 4 for details);
- 5 practical classes, 50 minutes each, presenting selected topics discussed in class in practice (see Sec. 5 for details);
- 1 multi-part semester project (see Sec. 6 for details);
- 5 short quizzes asking for solving short programing tasks or answer questions related to the class material.

4 Lectures (synopsis)

4.1 PART I: Introduction (3 lectures)

Notion of computer vision: goals of computer vision (and why they are so difficult), computer vision pipeline, current challenges and applications.

Biological visual mechanisms: human vision system, image formation in the human eye, image understanding, optical illusions.

4.2 PART II: Digital image analysis (10 lectures)

Image acquisition and formation: electromagnetic spectrum, optical systems, pixels and pixel arrays, spatial and image intensity resolutions, image sensing, cameras, color representation and interpretation, image compression and coding.

Image processing: point operators (pixel transforms, color transforms, histogram manipulations), neighborhood operators (Fourier transform, filtering and noise reduction, edge detection, mathematical morphology), interpolation and decimation, image segmentation (parametric and non-parametric, active contours).

4.3 PART III: Pattern recognition (16 lectures)

Feature extraction: geometric features, image intensity features, texture features, wavelets, keypoint detectors and descriptors.

Feature selection: filters and wrappers, feature space dimensionality reduction, forward and backward selection mechanisms, use of mutual information.

Feature classification and evaluation: Bayesian inference in vision, linear and nonlinear classifiers (k-NN, QDA, LDA, Support Vector Machines, neural networks), performance analysis and evaluation (classification errors, application of statistical hypothesis testing, correct data usage, cross-validation techniques).

Deep learning in Computer Vision: convolutional and recurrent neural networks, applications to object detection, image segmentation and captioning.

4.4 PART IV: Geometry and multiple-view analysis (6 lectures)

Video analysis: object detection and tracking, optical flow, application of Kalman and particle filtering.

Camera geometric model: pinhole model, homogenous coordinates, projective geometry, 2D and 3D transformations, camera calibration.

Multiple-view analysis: image stitching, epipolar geometry, fundamental matrix, stereo correspondence, 3D reconstruction methods.

4.5 Guest lecture

A guest lectures is planned to be given by invited computer vision expert, either from academia or industry.

5 Practical classes

During practical sessions, students will develop short programs using specialized libraries (Python + OpenCV | Keras | Tensorflow | Pytorch and/or Matlab + Image Acquisition Tlbx | Image Processing Tlbx | Computer Vision Tlbx).

1. **Detection of color objects in live stream:** this practical session will demonstrate how to process live webcam stream, detect color objects and track them, or do a color-based pixel value replacement to implement an "invisible cloak" working in real time (Python + OpenCV).

2. **Object classification:** this practical session will demonstrate how to automatically estimate simple geometrical and intensity features of objects and use then in object classification (Python + scikit-learn).

3. **CNN-based object recognition:** this practical session will demonstrate how to use pre-trained convolutional neural networks in feature extraction and image recognition (Python + Keras).

4. **Object tracking:** this practical session will present how to use Kalman filters and particle filtering in object tracking (Python).

5. **Image stitching:** this practical session will demonstrate sparse keypoint detection, geometrical transformations, inverse and forward warping and their application to image stitching (Matlab + Computer Vision Toolbox).

6 Semester project

Semester project will solve a complete visual object recognition task, from image sensing to a decision. Each project has three milestones, graded independently:

- data acquisition, pre-processing and description of the proposed solution (5 points),

- feature extraction, selection and classification: preliminary solution with source codes, interim results and ways to go forward (15 points),

- final solution with report and source codes, performance evaluation on unknown data, and short presentation (30 points).

Students will be offered several different project ideas and will select one of them. It is also possible to discuss individual project idea with the instructor. Students can form a team, but each teammate is graded independently. Appendix A presents a list of projects realized in past semesters.

7 Differences between 40535 and 60535 sections

The key distinction between 40535 (undergraduate) and 60535 (graduate) sections is an amount of research-oriented effort in solving practicals and proposing algorithms for the semester project solutions. More specifically:

1. **Semester project:**
   - Students attending the 40535 section will use computer vision methods fully covered and discussed during class meetings and databases suggested by the instructor. Students will be *encouraged* to use other methods and data found in additional sources suggested in class, but these will not be required to get a maximum grade for a given assignment.
   - Students attending the 60535 section will be *expected* to study additional reading materials suggested by the instructor, and demonstrate in their solutions a *research component*. For instance, (a) investigating which modern deep learning-based models are suitable for a given problem, (b) fusing information of different nature (data-driven, hand-crafted, human expert-driven) to make a final classification more accurate, (c) curating available computer vision databases and applying them in researching how the proposed models generalize on unseen data, (d) using visualization tools to explain developed models ("explainable AI" techniques), etc. This additional effort is estimated as about extra 25% when compared to semester project requirements in the 40535 section.
2. **Practicals:**
   - Students attending the 40535 section will be given short programming assignments (usually to be completed in 50 minutes during the class meeting) illustrating selected aspects discussed in class.
   - Students attending the 60535 section will be given the same source codes as students in the 40535 cohort, but with one or two additional questions that may need a quick additional reading, thinking or coding. This additional effort is estimated as about extra 25% when compared to practicals in the 40535 section.

8 Addressing qualifier course objectives

The 60535 section meets the following requirements for qualifier course objectives:

1. **Increased knowledge of fundamentals.** This course is cross-listed with undergraduate section (40535) and thus will cover some of the fundamentals of computer

vision. In particular, human vision system, image formation, point and neighborhood operators, Bayesian inference in vision, feature extraction and classification. Four (out of five) practical assignments will be devoted to fundamental aspects of vision, and will extend to more complex tasks for 60535 section attendees.

2. **Exposure to advanced techniques and skills.** At least five class meetings, and one practical assignment are devoted to deep learning methods, which are considered to be used in modern computer vision applications. Also, a guest lecture will essentially cover the most recent, selected research results and applications in computer vision. Attendees of the 60535 section will be also encouraged to include deep learning-based elements in their solutions to semester projects, if appropriate. Modern computer vision software libraries (such as PyTorch / Keras, and the most recent version of OpenCV) are planned to be used, especially in the 60535 section of the course.

3. **Evidence of research synthesis.** This objective is directly addressed by research components introduced to the semester project and practicals for students attending the 60535 section, as explained in Sec. 7.

9 Learning objectives

At the end of the course students should be able to:

- describe principles of image formation and understand biological inspirations in computer vision,
- understand how images are coded and compressed,
- implement selected algorithms of image enhancement, based on point and neighborhood operations,
- design and implement selected image segmentation methods to separate a useful part of the image from background information,
- build algorithms to extract and select appropriate image features suitable for a computer vision task being solved,
- understand pattern recognition principles and select appropriate classification techniques,
- deploy statistical methods for performance evaluation,
- understand geometrical transformations suitable for 2D and 3D images,
- successfully use selected theoretical and programming tools, along with computer vision libraries, in developing own computer vision systems.

10 Instructors

Dr. Adam Czajka
Office: 180 Fitzpatrick Hall
Office hours: Zoom link and time in SAKAI
Email: `aczajka@nd.edu`

Damien Dablain
Teaching Assistant
Office hours: Zoom link and time in SAKAI
Email: ddablain@nd.edu

11 Grading

- 50 points for semester, multi-part project
- 20 points for practical classes (4 × 5 points)
- 10 points for quizzes (5 quizzes × 2 points)
- 20 points for final exam

Grade scale:

		Total		
A	4.000	[100	->	91)
A-	3.667	[91	->	83)
B+	3.333	[83	->	75)
B	3.000	[75	->	66)
B-	2.667	[66	->	58)
C+	2.333	[58	->	50)
C	2.000	[50	->	41)
C-	1.667	[41	->	25)
D	1.000	[25	->	25]
F	0.000	<25		

Late grading policy:

- -1 point / day of late delivery, but ...
- students are allowed to be late **once** during the semester (*i.e.*, there is no penalty for the first late submission).

12 Code of honor

This class follows the binding Code of Honor at Notre Dame (http://honorcode.nd.edu). The graded work you do in this class must be your own. In the case where you collaborate with other students make sure to fairly attribute their contribution to your project.

13 Suggested reading

There are no required textbooks, however the following materials are suggested:

- R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011 (draft available on-line: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf)

- Computer Vision Foundation open access repo (including CVPR papers): `http://openaccess.thecvf.com/menu.py`

- J. Daugman, *Computer Vision – Lecture Notes*, available on-line: `https://www.cl.cam.ac.uk/~jgd1000/`

- CVonline: *Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision*: `http://homepages.inf.ed.ac.uk/rbf/CVonline`

- *Matlab Functions for Computer Vision and Image Processing:* `http://www.peterkovesi.com/matlabfns/index.html`

- R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Edition, Cambridge University Press, 2004 (available on-line: `http://www.robots.ox.ac.uk/~vgg/hzbook`)

- Reinhard Klette, *Concise Computer Vision*, Springer, 2014

14 COVID-19-related announcements

14.1 Health and safety protocols

In this class, as elsewhere on campus, students must comply with all University health and safety protocols, including:

1. Face masks that completely cover the nose and mouth will be worn by all students, instructor and teaching assistant.

2. Physical distancing will be maintained in all instructional spaces.

3. Students will sit in assigned seats throughout the semester, which will be documented by faculty for purposes of any needed contact tracing.

4. Protocols for staged entry to and exit from classrooms and instructional spaces will be followed.

14.2 Health checks and attendance

Every morning, members of the Notre Dame Community will be asked to complete a daily health check and submit their information via the Return to Campus Advisor application. The health check application will indicate one of the following:

a) Student is cleared for class and should attend class in person; or

b) Student is advised to stay home to monitor symptoms and should participate in class virtually and complete all assignments and assessments; or

c) Student must consult a healthcare provider and should contact University Health Services (UHS) for an assessment. In the meantime, the student should participate in class virtually and complete all assignments and assessments. Depending on the medical assessment, UHS will follow the University's standard protocol for obtaining an excused absence for medical reasons.

A Semester project topics

The main project ideas include:

- **Driving coach:** Create an application that analyzes dashcam videos and identifies and tracks objects relevant to the driver, like cars, lanes, signs, etc. The software should be able to detect/track multiple objects at a time, on a video stream.

- **Where am I? Identification of ND campus buildings:** Create an application that can recognize and identify selected ND campus buildings. The application should be able to analyze pictures of ND buildings taken at different angles, distance and weather conditions, and identify the name of the building. Optionally, the application can be implemented on a mobile device and show the location of the building on a map.

- **Hand gestures recognition:** In this project you will write a program that recognizes a hand gesture (one of a few pre-defined). You can assume a uniform background and a vertical placement of a hand. You will need to implement a segmentation algorithm, propose and select features and do the classification. The final solution should work in real-time for a video stream captured by the camera in your laptop ("real-time" means 1 frame per second, or faster).

- **Face recognition:** In this project you will write a program that uses a face detection method (e.g., Viola-Jones) and one of the image feature descriptors (e.g., LBP) to recognize your face. Your solution should correctly recognize you when tested on a small set of people, for instance your classmates.

- **Object tracking in outdoor environment:** In this project you will write a program that detects and tracks selected object observed in outdoor environment. One example are the planes that approach SBN and can be easily observed on the sky from ND campus. Assume that you are filming an object with a camera that shakes and then you need to detect the object's position in each frame. If you don't like planes, pick an object you like, for instance a car. You can use some pre-trained deep learning models to extract useful features.

- **Recognition of handwritten digits:** In this project you will write a program that first learns to recognize handwritten digits (0,...,9) on a database, and then can recognize handwritten digits written by you. You can't use your writing samples in the training.

- **Object-following Raspberry Pi robot:** This project is for students who love mixing computer vision, programing and electronics. The aim is to build an autonomous robot equipped with a camera and following the selected object. Students will be given basic components: Raspberry Pi 3+, camera, robot car chassis, motors, motor controllers and battery. Other minor elements, such as wiring, connectors, etc. should be secured by students. The solution should go beyond just color-based detection: assume the objects are more complicated are require both color- and shape-based features. Use of deep learning-based feature extraction is permitted.

- **Scavenger Hunt:** Create a game of scavenger hunt with pictures of objects inside ND campus (or inside a building). The game should be played in pairs. The person "hiding" the objects will take pictures of specific objects (statues, doors, paintings) on campus. The hunter should look for these places or objects and take pictures of them. Then the pictures should be matched to the original images, to verify if the object is the same that was "hidden." Ideally, the game should be implemented to run on mobile devices.

Below is an example list of project ideas proposed by students, adapted by the instructor to the purpose of this class, and realized in past editions:

- **Credit card autofill:** The aim of this project is to develop an algorithm to detect a credit card in a picture and to classify its main components, such as credit card number, expiration date and the cardholder's name. While the solution can be developed on a database of credit card images acquired on a uniform background, the final solution should be tested in unconstrained environment. The reasons of (possibly) worse performance should be explained and appropriate updates to the designed method should be discussed.
- **Brand recognition from labels:** The aim of this project is to automatically detect and classify brands of food manufacturers (limited to 4-5 different brands). This will be done after taking pictures of different packages showing labels with brand logos (at different angles, in different scale, etc.).
- **Tracking of a Rubik's cube:** The aim of this project is rotational tracking of a Rubik's cube (due to its uniform shape and distinctly colored sides). Ideally the program will be able to find the cube against most backgrounds and then be able to track how much the cube has been rotated from its original position on each of the three axes.
- **Automatic recognition of workout data:** The goal of this project is to capture workout data from a Concept2 erg. This data can then be automatically uploaded to a workout log.
- **Active contours for refining deep learning-based segmentation:** In this project a deep learning-based method will be used to roughly segment the objects from the background. Then the active contours will be applied to refine the segmentation results. This project will require finding a suitable energy function for the problem.
- **Deep learning-based OCR:** A "Recurrent U-Net" will be developed in this project that passes letter predictions from U-Net into a bidirectional LSTM to pull word predictions from there.
- **Presentation Attack Detection in iris recognition:** The result of this project will be a binary classifier that states whether an iris is live or a spoof based solely on the textures in the image.
- **Lip-reading service:** The aim of this project is to develop a solution that recognizes words by observing lips in videos. This solution may later augment speech-to-text services by adding visual information to the audio channel. This "lip-reading" service will assume that the speaker's frontal face is fully visible.

- **Transformation-aware distance metric learning:** This project aims to develop an approach to improve descriptor matching for applications reliant on image transformation analysis. It will highlight the drawbacks of transformation-invariant deep networks for specific applications such as image forensics and try to improve upon them.

- **Deep learning in semantic segmentation of iris images:** The aim of this project is to use CNN-based models (including a CNN that performs dilated convolutions) to solve a semantic segmentation task. The use case is iris recognition and its first step: image segmentation.

- **Automatic learning-free neural image sparse segmentation and reconstruction:** Microscopic image data (microCT X-ray, sSEM, SCoRe, etc.) is difficult to properly segment due to a lack of ground truth annotations for training learning models. This project aims to improve upon the quality of reconstructions generated by classical (learning-free) image processing techniques by developing an optimization technique that adapts to local image conditions and iteratively selects the highest-quality segmentation.

- **CAPTCHA solver:** The aim of this project is to write an algorithm that recognizes CAPTCHA letters/digits from a few different CAPTCHA sources.

- **Recognition of face expressions:** The aim of this project is to recognize a facial expression (out of a set of pre-defined expressions).

- **Recognition of cities based on skyline pictures:** The aim of this project is to write a program that accepts a picture with a skyline and outputs which city the picture was taken in. A small set of cities, say 10, will be selected to reduce the difficulty of the problem.

- **Recognition of numbers in lottery tickets:** This project will focus on automatic detection and recognition of numbers on various lottery tickets. Tickets from at least three different types of lotteries will be used. A nice complication for this semester project is that the program will not be looking at a particular region for numbers.

- **Seam carving:** Seam carving, or content-aware image resizing, is a technique that allows image retargeting (displaying them on media of various sizes) by removing "unimportant" image areas and using only the "important" ones. This is done by introducing or removing seams. In this project student will apply object segmentation and recognition to generate a new kind of content-aware energy function that is used in content-aware image scaling.

For each project a few appropriate databases will be suggested. Additionally, it is good to analyze this well-maintained repository of links to various Computer Vision databases: `http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm`.