Introduction to Computer Vision, CSC 589

Fall, 2017 Syllabus
Monday/Thursday 11:20-12:35pm
Room: Meyers Building (DMTI) 109

Instructor:
Prof. Bei Xiao, Department of Computer Science
bxiao@american.edu
Office: Myers Building Room (DMTI) 204

Course Overview:
This course is an introduction to current algorithms used in computer vision and computational photography (automatic image editing and manipulations). We will start from low-level image processing (edges), and then move to mid-level feature analysis (texture, color, motion), and eventually to high-level image and video understanding (objects, faces, scene, human activity). The topics include basic image processing and image analysis, camera models, texture synthesis, motion analysis, automatic image editing, object and scene recognition, face and pose recognition and a gentle survey of deep learning methods for computer vision. Students will learn the state-of-the art tools in computer vision and also hands-on experiences on image manipulations.

We will use Python as our primary programming tool in this course. Codes written in C and MATLAB might be demoed if needed. Working models of computer vision will be demoed in class and homework will be programming exercises with tools such as Numpy, SciPy, and OpenCV.

Delivery methods:
The lectures are given through coding demos, power-point presentations, white boarding, and discussions. Students are expected to engage in active coding throughout lectures. There will be a quiz at the end of lecture in a random day each week. Readings from the textbook are assigned at the end of each lecture. There will be one written mid-term exam and a take-home final project. Homework includes 6 programming projects.

Expected Learning Outcomes:
Upon completion of this course, students should be able to:

1. Recognize and describe both the theoretical and practical aspects of computing with images. Connect issues from Computer Vision to Human Vision

2. Knowing how to use programming tools (Python) to process, manipulate and make simple inferences from images.

3. Describe the foundation of image formation and image analysis. Understand the basics of 2D Computer Vision.

4. Become familiar with the major technical approaches involved in computer vision.

5. Get an exposure to advanced concepts leading to object and scene categorization from images.

6. Build simple computer vision applications with existing tools and frameworks.

Perquisites:

You should have taken CSC 280 and ideally, CSC 281 before taking this class. If you haven't taken any programming classes before, you might consider take this class **after** you obtain some programming skills. No prior experience with computer vision is assumed, although previous knowledge of visual computing or signal processing will be helpful. The following skills are necessary for this class:

- Data structures: You'll be writing code that builds representations of images, features, and geometric constructions.
- Programming: A good working knowledge of programming environments that support image and video analysis. All lecture code and project starter code will be in Python.
- Math: Linear algebra, vector calculus, and probability. Linear algebra is the most important and students who have not taken a linear algebra course is encouraged to take some simple tutorials.

Expected work outside class:

- This is an advanced technical class. It is expected that you work at least 6 hours outside the class room every week (2 hours per credit) to be order to receive a grade of B and above. These hours might increase as the course becomes harder or if you are finding yourself catching up.

- The most useful way to learn is to do programming exercises. This includes your homework, your in-class demos, your weekly labs and online exercises you can find.

- Computer vision is an applied and also theoretical discipline. You should read the assigned readings and catch up with the assigned tutorials. The projects will be increasing demanding as the semester moves on. Please make sure you assign enough time for the projects.

Office hours:

Location: Meyers Building Room 204
Monday: 5:30pm-6:30pm
Thursday: 4pm-5:30pm

Online discussions:

Not all questions can be answered in office hours. We highly recommend you join online discussions using Pizza. Sign up for Pizza here to join online discussions about lectures and homework:

piazza.com/american/fall2017/csc589

You can ask any coding/software set up/homework/lecture content related questions on Piazza. In this way, when I answer your post, other students can also see the responses.

Textbooks and references: (all books are in library reserve and in the bookstore)

1. (Required) R. Szeliski, Computer Vision: Algorithms and Applications available at http://research.microsoft.com/en-us/um/people/szeliski/Book/
2. (Recommended) Computer vision with Python. O'Reilly.
3. (Recommended) Modern Approach to computer vision. http://cmuems.com/excap/readings/forsyth-ponce-computer-vision-a-modern-approach.pdf https://www.amazon.com/Computer-Vision-Modern-Approach-2nd/dp/013608592X

4. OpenCV -Python tutorial
   http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html

Computer and software:
Students are encouraged to bring their laptop to class. Python 3.5 and Numpy, Scipy, OpenCV should be downloaded. All of these libraries must be downloaded and tested to run during the first two weeks of classes. This is a hands-on and technical class. You are extremely encouraged to bring your own laptop to class to follow in-class demos. The best way to obtain libraries of Python is to download Anaconda. Detailed instructions will be included in the course materials on blackboard.

Attendance policy:
It is important to attend and be on time for all lectures. Announcements made at the beginning of class often are vital and arriving late may also result in missing an assignment. While in class, you should be prepared to engage in discussions and classroom activities. To this end, please have completed, understood, and thought about the assignments and readings prior to class. In-class quiz, which will be randomly assigned every week, cannot be made up if missed. If you missed quiz because of (religious holiday, medical emergency or sports events) please arrange with me to make them up.

Missing one class without written request will result in 2% reduction in attendance score.
However, students can be absent for class and arrange for make-up exams if they have written proof of religious holidays and documented disabilities. Athlete who would miss class due to sports events must send written form to instructor at least 24 hours before the class.

Late Policy:
The course doesn't grant extension on homework no matter what the excuses. Instead, we offer late days and the grades will be automatically adjusted based on these policies.

The submission deadline of homework and take-home exam is **11:59 pm** on the due date**.** During the entire semester, the students may accumulate maximum of **four late days** that they can use on problem sets starting with Homework 1 (**excluding Final project**). Late days are discrete (a student cannot use half a late day, e.g. 12 hours after the deadline is counted as one day). If your assignment is late, your TA or instructor will let you know how many late days are left. **Any additional late work beyond these late days will not be accepted.** To avoid surprises, we suggest that after you submit your problem set, you double check to make sure the submission was uploaded correctly.

Exam Policy:
Mid-term exam will be announced at least one week ahead of time. If you have special needs, you need to notify me at least **7 days** before to arrange the test be performed off-class in the exam center. **Missed exams and quizzes cannot be made up.**

Email Policy:
You can email me if you have questions regarding lectures and homeworks. But you must write to me at least 48 hours to expect an answer. No homework is accepted via Email. Everything must be uploaded onto Blackboard.

Learning accommodations:

Students wishing to receive accommodations for a disability, are to bring their documentation directly to the Academic Support and Access Center (ASAC- http://www.american.edu/ocl/asac/index.cfm), in MGC 243, x3360. ASAC, in turn, will notify the professor of the accommodation required. Keep in mind that accommodations can only begin when the professor is notified. This means that students should take care of this at the start of the semester, *before* the work for which they require accommodation is due.

Academic Integrity:
Plagiarism and academic misconduct are defined in the University Academic Integrity Code. You should be familiar with what constitutes academic dishonesty. In particular, you should observe the following rules: Collaboration on projects is restricted (if you have high-level discussion with another person, please write down the name of the person). Your code must be entirely your own work. All exams will be close-book, close-note, no Internet, no smart phones. Instances of plagiarism may be reported and could result in disciplinary action. Please see above (**Policy of collaborative work)** for details. All the violations will be reported immediately to the Dean of College of Arts and Sciences.

There is absolutely no copy of other's code during the exams. If caught, students will be immediately reported to the Dean of College of Arts and Sciences.

AU 's academic integrity code is described here:
http://www.american.edu/academics/integrity/code.cfm

Grading: 60% homework projects, 10% mid-term written exam, 10% Final project, 5% in-class quiz, 5% attendances. We sometimes offer extra credits for additional features in homework and projects.
Programming projects will be typically graded as follows:

10% - *Did you make a reasonable effort and submit something?*
40% - *Does it compile without errors?*
10% - *Does your code have sufficient and meaningful comments?*
20% - *Did you follow the appropriate structure?*
20% - *Does your program work correctly and generate desirable results and did you present your results with efficient graphics and plots?*

A final letter grade will be converted from the percentage you receive through out the course.

Grading Scale listed below:

93-100%	A Excellent
90-92%	A-
87-89%	B+
83-86%	B Good
80-82%	B-
77-79%	C+
73-76%	C Acceptable
70-72%	C- (Cut off for receiving credit for CS major)
60-69%	D
0-59%	F

If your percentage is 85.6%, for example, it will be round up to 86%, but if it is 85.4%, it will be round up to 85%. I will try my best to upload your up-to-date grade percentage as the semester goes along. But the most important thing is submitting all of homework on time, showing up in class, and catch up content you are lost. It is usually too late to argue for grades during the final week.

You can ask grading related questions to me personally in office hours (no email correspondences please). Only numerical errors and data entry errors can be argued about grading. But I am happy to provide feedback on your homework.
Please refer to university academic rules for converting grading scale and the corresponding honors degree requirements:

http://www.american.edu/provost/undergrad/undergrad-rules-and-regulations.cfm#5.1

Syllabus:

Class Date	Topics	Projects
August 28th	**No lecture, Instructor travels**	**Set up Anaconda, Python,**
August 31	**No Lecture, Instructor travels**	**Numpy tutorial**
September 4	**No lecture, Labor day**	**Linear Algebra review**
September 7	**Introduction to Computer Vision**	
September 11	**Image Filtering**	**Project 1 out**
September 14	**Image Filtering**	
September 18	**Introduction to Linear Algebra**	
September 21	**Image derivatives**	**Project 2out**
September 25	**Thinking in Frequency**	
September 28	**Thinking in Frequency 2**	
October 2	**Sampling, Gaussian Pyramids**	
October 5	**More on Gaussian Pyramids**	
October 9	**Image Blending**	
October 12	**Edge detection, Mid-term review**	**Project 3 out**
October 16	**Mid-term exam**	
October 19	**Edge detection, Boundary Detection**	
October 23	**Interests points and Corners**	**Project 4 out**
October 26	**Local Image Features**	
October 30	**SFIT, Feature matching**	
November 2	**Alignment**	
November 6	**Panoramas**	
November 9	**Camera models, Lighting and color**	**Project 5 out**
November 13	**Stereo**	
November 16	**Two view geometry**	
November 20	**Introduction to Machine learning: unsupervised learning**	**Project 6 (tentative)**
November 23	**No lecture, Thanks giving**	
November 27	**Machine learning: supervised learning**	
November 30	**Introduction to recognition**	**Final Project out**

December 4	**Deep learning**
December 7	**Deep learning 2**

December 11

Tentative topics if time allows:

Camera models

Structure from motion

Motion and tracking