CSci 4270 and 6270
Computational Vision
Spring Semester, 2025
Course Syllabus

Professor:	Chuck Stewart
Class:	Monday, Thursday, 2:00 - 3:50, Low 4050
Office Hours:	after class; Wednesday 3:00-5:00, Lally 307
Email:	`stewart@rpi.edu`
TA: F. Mohbat	`mohbaf@rpi.edu`
TA Hours:	Thursday 4:00 - 6:00, Friday 2:00 - 4:00
TA Location:	TBD
TA: Ian Rudy	`rudyl@rpi.edu`
TA Hours:	Monday 10-12, Tuesday 3-5
TA Location:	TBD
CA: Ilan Beyen	`beyeni@rpi.edu`
CA Hours:	Wednesday 1-3
CA Location:	TBD
Submitty URL:	`https://submitty.cs.rpi.edu/courses/s25/csci4270/`

Note: all course material will be posted on the Submitty site.

Overview

Cameras and digital images are everywhere, with billions of cameras being used to take trillions of images, producing far more data than humans can possibly absorb, interpret or act upon. The result is a wide-spread need for algorithms and software that can combine, summarize and interpret these images, a need for the fields of computer vision and image analysis. Applications of computer vision include face, object and scene recognition, security and biometrics, photomontaging and virtual tours, special effects in graphics, photography and the movies, autonomous robots, self-driving cars, human-computer interaction, and medical diagnosis and treatment. Given the exponential growth of digital devices, the potential for new applications seems unlimited.

Computer vision is challenging. Each image is a large, noisy, quantized, two-dimensional array of intensity values. Each intensity value is created by light reflected off a surface sitting in the three-dimensional world, focused by a lens, digitized and recorded by a camera. Information is lost and noise is added at each step. To address these challenges, computer vision research scientists and applications programmers are employing a wide variety of physical, engineering, mathematical, statistical, algorithmic and software techniques to develop computer vision systems.

Our goal in this course will be to learn about the challenges, the techniques, and the applications of computer vision. Since this is a computer science course, much of our focus will be on computational aspects of the computer vision problem. We will also consider potential impacts of advanced computer vision technologies on individuals and on society.

Learning Objectives

At the end of this course, each successful student will be able to

- Apply techniques of calculus and linear algebra to solve problems involved in building the components of a computer vision system.
- Develop efficient algorithms for solving problems in computer vision.
- Write small-sized and intermediate-sized programs and train neural networks to solve problems in computer vision.
- Map potential applications of computer vision into specific technical problems.
- Assess the difficulty of specific technical problems in computer vision and select potential solution techniques.
- Discuss thoughtfully some of the social implications of computer vision technology.
- (6270 only) Evaluate the significance of the ideas and the thoroughness of the experimental analysis of a current research paper in the computer vision field.

Prerequisites

Students should have had courses in programming, in data structures and in algorithms (e.g. CSCI 2300). Mathematical background should include a course in multivariable calculus and linear algebra (at least MATH 2100). This requirement is somewhat flexible since some students have done well in previous semesters without this background. In addition, we will be discussing the some of the necessary mathematical techniques as we proceed through the semester. Students may find my mathematical methods lectures notes helpful:

`http://www.cs.rpi.edu/~stewart/math_techniques`

Requirements

Student grades will be determined by the following simple formula:

- 60% — homework assignments
- 40% — in-class quizzes

Letter grades will be determined based on the rounded, combined averages. Final cut-offs will be at the instructor's discretion, but will be no higher than 92 for an A, 89 for an A-, 86 for a B+, 82 for a B, etc. The same cutoffs will be used for the 4270 and 6270 courses. The grading and the curves will be different for each course due to differences in assignments.

Homework and Programming

Homework will involve solving mathematical problems, developing algorithms, writing programs, training models, and analyzing results. The mixture of these will vary between assignments, but the programming aspect and resulting analysis will be the most important. Expect to write a lot of code.

Most homework assignments will be done individually. Occasionally, and only if explicitly allowed, some problems may be solved in teams of two.

Programming will be done using Python, NumPy, SciPy, PyTorch and OpenCV. Students will need at least the following Python packages:

- python, version 3.9
- numpy, version 1.26
- scipy, version 1.11
- matplotlib, version 3.8
- cv2, version 4.5, including the contributed software
- pytorch, version 1.11

The dependencies between these can sometimes be a little tricky, so I recommend that you use a managed installation such as `anaconda`, together with the virtual environment it can create. See `https://www.anaconda.com/distribution`. You will have install OpenCV separately. To do this, switch into the anaconda bin directory (`~/opt/anaconda3` on my Mac) and type

```
pip install opencv-contrib-python
```

Do **not** install just the package `opencv`.

Midway through the semester you will need `pytorch`. It may be installed using something like

```
conda install pytorch torchvision torchaudio -c pytorch
```

You may be able to use `pip`:

```
pip install torch torchvision torchaudio
```

If you have a GPU on your machine you will have to configure `pytorch` specially.

GPU Resources

The last three assignments will require you to train neural networks, and for this you will need access to a GPU. One possibility is to make use of a cloud resource such as Google Colab. Although there are free versions of these, it may be worthwhile to purchase a professional version for at least the month or two you need it. We will also give access to the NPL cluster at RPI's Center for Computational Innovation (CCI). Watch for a Submitty "Gradeable" to register for this resource.

In-Class Quizzes

There will be four 60-minute, in-class quizzes given on the dates shown below. Each student's three best quiz scores will each be worth 12% of their final grade and the worst one will be worth 4%. All quizzes will be closed-book and closed-notes. Practice problems for the quizzes will be distributed with the lecture notes. Most quiz questions will be taken from these practice problems, often with minor modifications. Other quiz questions will be closely related to homework problems.

Late Policy

Students have five free "late" days they can use on homework throughout the semester, with *at most two used for any one homework*. A late day is defined as any whole or partial day after the submission deadline. These free late days are to be used for minor illnesses, balancing other course work, problems with your computer, etc. Students do not need to use late days for substantial personal emergencies. Just get an excuse from your class dean and then together we can arrange a suitable time to complete any missed work.

After all free late days have been used, any further late days will involve a 20% penalty off of the total available points on that assignment for each day or partial day.

4270 vs. 6270

There are two versions of this course, one at the 4000 level and one at the 6000 level, meeting together. The lecture material will be the same, but students in 6270 will need to complete more advanced work. This will be in the form of differences between problems in homework assignments, with more analytical questions assigned to students in 6270. There will be at least one problem — typically out of four to six — on each homework assignment that is required of students in 6270 but not students in 4270. In addition, students in 6270 will have one extra homework assignment requiring them to read and critically review a paper from the current research literature. Details of this assignment will be provided in the first month of the semester.

Lectures, Lecture Notes and Resources

Each lecture will be broken up into smaller segments with a break and Q/A discussion. I will try to post notes for each lecture on Submitty at least two days in advance of the class meeting.

While there is no textbook for the course, some references will be made to Rick Szeliski's book (`http://szeliski.org/Book/` — download the 2nd edition). General resources will be posted on the course Submitty site and pointers to reading material about each lecture will be embedded in the notes. Students are strongly encouraged to share references that they find helpful. The use of outside materials for this course is strongly encouraged and often a key to success.

Academic Integrity

The Rensselaer Handbook of Student Rights and Responsibilities and The Rensselaer Graduate Student Supplement define various forms of Academic Dishonesty and procedures for responding to them. All forms are violations of the trust between students and teachers. Student-teacher relationships are built on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own performance. Acts that violate this trust undermine the educational process.

The Rensselaer Handbook of Student Rights and Responsibilities and The Rensselaer Graduate Student Supplement define various forms of Academic Dishonesty and you should make yourself familiar with these.

In this class, all assignments that are turned in for a grade must represent the student's own work. In cases where help was received, or teamwork was allowed, a notation on the assignment should indicate who you collaborated with. Submission of any assignment that is in violation of this policy will result in a penalty. If found in violation of the academic honesty policy, students may be subject to two types of penalty. The first violation will result in 0 grade for that assignment and up to an additional 5% overall grade penalty. The second violation will result in failure of the course. Electronic comparison tools will be used to find potential integrity violations. If you have any questions concerning this policy before submitting an assignment, please ask for clarification.

More specifically: Unless otherwise specified, students are expected to submit their own solutions and write-ups. Students may work together to understand problem requirements, to sketch solution ideas, and to discuss results. Implementations and write-ups must be done individually. Automatic comparison tools such as MOSS will be used to find and compare submissions that are overly similar. Students who find outside resources to help in solving homework problems must reference these resources in their write-ups and make clear what they did individually in addition to what the resource did.

Students may use AI-based resources such as LLMs in much the same way you interact with fellow students in the course — to explain background material, to understand problem requirements, and to aid in debugging. Your solutions should be your own, however. If you use an AI-based resource (actually, any resource), be sure to explain exactly how in your homework write-ups. Submission of an AI-generated solution for a homework assignment will be penalized as an academic integrity violation.

Finally, copying or any form of collaboration on in-class quizzes will result in an automatic F in the course.

Topic Schedule

Here is a summary of the topics we will be covering during the semester, along with tentative due dates for assignments. These will be updated throughout the semester.

Lec.	Day	Date	Topic	Due
1	Mon	1/06	Introduction; images	
2	Thu	1/09	Numpy and OpenCV	
3	Mon	1/13	Linear algebra*	
4	Thu	1/16	Lines and estimation*	HW 1
	Mon	1/20	No Class; MLK Jr Day	
5	Thu	1/23	Transformations	
6	Mon	1/27	Image processing	
7	Thu	1/30	Features	HW 2
8	Mon	2/03	Descriptors	Quiz 1
9	Thu	2/06	Camera geometry	
10	Mon	2/10	Two image matching	
11	Thu	2/13	Intro. to machine learning	
12	**Tue**	2/18	Intro. neural networks	HW 3
13	Thu	2/20	Convolutional neural nets	
14	Mon	2/24	Pytorch	Quiz 2
15	Thu	2/27	Catch up and review*	
16	Mon	3/10	Data and classification	
17	Thu	3/13	Detection, part 1	
18	Mon	3/17	Detection, part 2	HW 4
19	Thu	3/20	Transformers	
20	Mpn	3/24	Segmentation	
21	Thu	3/27	Faces	Quiz 3
22	Mon	3/31	Diffusion	
23	Thu	4/03	Motion	HW 5
24	Mon	4/07	Tracking	
25	Thu	4/10	Stereo	
26	Mon	4/14	LiDAR and 3d	HW 6
27	Thu	4/17	Catch up and review	
28	Mon	4/21	Conclusion	Quiz 4

Lectures labeled * will be prerecorded since I will be out of town.