Projects and Review

"If I have seen further it is by standing on the shoulders of giants." - Isaac Newton

Project

Instructions for submission:

- Write a document and submit your proposal and report as **PDFs**.
- Include your code as a set of separate files
- Submit through **MarkUS**.

Max points: 40

- 5 points for the proposal. **Due October 27, 10:59pm**
- 20 points for the report. **Due Nob 22, 10:59pm**
- 15 points for the oral presentation. **Nov 27, 29, Dec 4**

Teams

- Projects can be done individually or in groups of up to three students. Most projects are quite big, so pair work is actually encouraged.
- You will be given time at the end of this class to find/discuss with your group.
- Projects are calibrated for two students, for groups of three, justify what work each member will carry out and how you will **expand** upon the project in comparison to a group of two.
- As a ballpark each student is expected to do roughly as much work as 2.5 of the regular term assignments on the project, and one assignment's worth of work on the report.

Grading

The grade will take into account the following factors:

- **Your demonstrable work:** what you've learned, and what knowledge you've applied.
  - Whether you implemented a technique yourself or used code available online.
  - What works and what doesn't.
- **Problem solving/Critical Thinking:** Your ideas to tackle a problem including how appropriate the techniques you chose are for the problem.
  - Coming up with novel ideas is a big plus.
- **Presentation/Analysis of results:** Show your results for each task by including a few pictures in your report.
  - Show good cases where your method works and also bad cases where it doesn't.
  - Providing some intuitive explanation of the failure cases is a plus.

Grading

Continued from previous slide:

- **Thoroughness of your report:**
  - How well you describe the problem and the techniques you chose.
  - How well you analyzed your results.
  - Whether your citations to the techniques you used are appropriate.
- **Your implementation:** The accuracy of the solution, run-time, and partly also how organized the code is (scripts that run the full pipeline or specific subtasks, documentation).
  - The grade will also take into account the accuracy of your solution in relative comparison to other students' results (i.e. taking into account problem/data difficulty).

Project Proposal: Due Oct 27, 10:59pm

- A project proposal is your commitment to a particular project.
- Include a header:
  - Project title
  - Names of project partner(s) (if a group project).
- For groups of three include:
  - Justification for a larger group.
  - What work each member will do.
  - How you will expand on the two-person project
    - E.g. implementing instead of downloading additional modules.
- Main body:
  - Outline your general ideas and difficulties you think the project has
  - Reference/justify the scope of work (in similar steps taken) to one of the included proposals.
  - It should be clear that work is proportional, and we reserve the right to refuse a proposal (refusal will not affect your proposal mark).
  - Cite any key works you will make use of.
- Max length is 1 page (font size 11, 1 inch margins, single space).

Project Report: Due Nov 22, 10:59pm

Your project report is the basis for showing your work and demonstrating what you've learned in this course. It should:

- Be 4-6 pages in length (if single spaced) plus figures and references.
- Reference your code, but do not include it in the report.
- Build it as you go, be clear and concise.
- Use the noted section breakdown (to follow). Notice that it **approximately** follows the structure of an academic paper.

There are many great guides online to writing scientific papers (e.g. link).

Project Report: Due Nov 22, 10:59pm

- Whenever you use ideas, code or data from a paper, forum, webpage, etc, you need to:
  - Cite it in your report.
  - Include a short description of the method showing your understanding of the technique.
  - If there were multiple options for techniques or code, please also explain why you chose a particular one (See below for report outline and details).
- It may be that the project has too many questions and you might not be able to solve them all. Don't worry. The goal is to teach you how to solve cool problems and hopefully you have fun doing it. We will evaluate relative to how everyone does. Think of it as a competition.

Project Report: Due Nov 22, 10:59pm

- **Each student, in each group, prepares and submits his/her own individual report.**
- You may share figures, but properly credit them to the student who created them.
- Each student must be able to defend the project on his/her own.
- From the report it should be clear what each student contributed to the project.
- By November 22nd you will need to hand in the project report including code.
- Make the code organized and documented, ideally including scripts that run your pipeline.

Introduction and related work

Approximately 0.5-1.5 pages.

- Give an introduction to your problem and why you picked it
- Summarize what previous work you found and read (at a high level, referencing back to concepts/algorithms we learned in the course).
- You may also include your hypothesis:
  - What did you expect to work/not work.
  - What did you expect the main challenges to be, the hardest data, etc.
  - What limitation with existing methods (or lack thereof) is your novel approach to address

Methods

Approximately 1.5-2 pages.

- Describe each step of your method.
- If you use an algorithm, e.g. SIFT, summarize in your own words how it works.
- I suggest using pictures and diagrams to show the modular/algorithmic flow, and results of intermediary steps of your algorithm to show your processing pipeline.
- Make clear references/associations to your code so we can follow along.
- **Make it clear which parts you did, which work your group did, and what was obtained online.**

Results and discussions

Approximately 1-2 pages.

Describe and demonstrate your results.

- Compare various steps of your process, e.g. how much additional accuracy is gained by each step.
- Show example images and/or videos.
- Explain why your method works when it does, and why it fails.
- Show examples and justify your results (i.e. if it a is problem with the algorithm or a problem with your implementation).

Main challenges

Approximately 0.5-1 pages.

Describe the main challenges you faced, and how you solved them.

- Parameter tuning
- Thought X would work, but needed to implement Y
- Lost hours debugging, but solved using controlled example
- Etc.

Conclusion and future work

Approximately 0.25 pages

- Summarize what you did, and how it worked.
- What you would do in the future to improve your results?

Oral Presentation Nov 27, 29, Dec 4

- You will book a 10 minute slot.
- The slot schedule will be posted in a few weeks.
- In the oral defense you'll need to run some of your code and be able to defend it.
- Prepare a short introduction and demo (6-7 minutes).
- Grade based on clarity of demonstration and question responses.

Sample Project 1

(do not pick this. just an example to give you a sense about size/scope)

This project is about analysis of news broadcast. Here are the tasks to solve:

1. Download three sets of news broadcasts: Lego Space man;Clowns.. Parliament
2. Download a set of male and female faces with gender labels: link, e.g. link.
3. Detect shots in the videos. A shot is a set of consecutive frames with a smooth camera motion.
4. (Manually) Annotate shot boundaries in the video. How would you evaluate how well you are detecting the shots? Compute your performance.
5. Detect the news company's logo (without prior knowledge of location).
6. Detect faces in the video.

Sample Project 1

(do not pick this. just an example to give you a sense about size/scope)

This project is about analysis of news broadcast. Here are the tasks to solve:

1. Perform face tracking by correctly associating a face detection in the previous frame to a face detection in the current frame.
2. Train a classifier that can predict whether a face is female or male.
3. Visualize your results: produce a video in which you show a bounding box around the detected company logo, and bounding boxes around the detected faces. Each face bounding box should have text indicated whether the face is male or female.
4. **Bonus:** Can you detect the clowns as a 3rd or class?

Example results for previous year's data (Source: Marisol Miel Aguila and Ilia Samsonov):

1. Clip A
2. Clip B

Sample Project 2

(do not pick this. just an example to give you a sense about size/scope)

This is a project for a single student (no group work allowed). Imagine a

phone app where you take a picture of a DVD and the app tells you which movie it is. In this project, the goal is to localize and recognize a DVD cover in an input image given a database of a large set of DVD covers.
You will need to implement the following paper: Nister, Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR2006,link.
You will download a set of images of DVD covers containing training images of DVD cover templates, and testing images each containing one DVD in a non-frontal viewpoint taken from a phone camera.

Sample Project 2

(do not pick this. just an example to give you a sense about size/scope)

This is a project for a single student (no group work allowed).

1. Download the Stanford Mobile Visual Search Dataset database: link. The **Reference** covers are your training data, and the rest is for testing.
2. Perform homography estimation with RANSAC to match one DVD cover image to a test image.
3. Implement the efficient retrieval approach by Nister and Stewenius including:
    1. Keypoint detection
    2. Keypoint description with rotation and scale in-variance
    3. Building and using the vocabulary tree (i.e. quantization)
    4. Hierarchical scoring
    5. Retrieval

Sample Project 2

(do not pick this. just an example to give you a sense about size/scope)

This is a project for a single student (no group work allowed).

1. For a test image, retrieve top 10 matches (DVD cover images from the database) returned via the implemented approach. Compute a homography with your implementation from (2) for each retrieved DVD cover image and the test image.
2. Find the DVD cover image from (4) with the highest number of inliers. Plot the test image with the localized DVD cover as well as the best retrieved DVD cover.
3. This is a 2006 paper, come up with and implement your own improvements to address the miss-classifications results based on what we've learned or other resources (see bottom of this document).

Sample Project 3

(do not pick this. just an example to give you a sense about size/scope)

- This is a project on segmentation and object recognition in 2D medical images using deep learning. You will build an automated system to help characterize histopathology images of cancerous cells.
- There are two main parts to this project, detecting the cells and then classifying them.
- Sirinukunwattana, Korsuk, et al. "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images." IEEE transactions on medical imaging 35.5 (2016): 1196-1206, link.

Sample Project 3

(do not pick this. just an example to give you a sense about size/scope)

1. Download the data set: link.
2. Detect patches with a nucleus at the centre:
    1. Use techniques learned in the course to build an approach to detect nucleus centres; much like you would faces, cars, etc.
    2. Experiment with different approaches using both grayscale and colour information; Compare and contrast.
    3. Reproduce (approximately) their baseline CP-CNN result. Take a look at MatConvNet (link).
    4. Setup cross-validation and evaluate your results. Compare to those in the paper.

Sample Project 3

(do not pick this. just an example to give you a sense about size/scope)

1. Classify the detected nuclei patches in the 4 classes (epithelial, inflammatory, fibroblast, and miscellaneous):
    1. Build your own patch-feature descriptor
    2. Classify your detected patches into one of the 4 classes, and visualize your results.
    3. Experiment with different approaches using both grayscale and colour information; Compare and contrast.
    4. Implement their NEP classification step (this can be combined with any type of classifier in practice) and try in conjunction with your own approach

Sample Project 4

(do not pick this. just an example to give you a sense about size/scope)

In this project you'll familiarize yourself with some of the most important problems that arise in the field of autonomous driving.

You will use data from the KITTI Vision Benchmark Suite: link

Sample Project 4

(do not pick this. just an example to give you a sense about size/scope)

Part 1 road detection:

1. Download the left and right colour images and calibration data for the road recognition dataset: Info and link. You will find training and testing data, do not mix them up.
2. Compute disparity between the two stereo images. We do not mind if you use existing code as long as you include a description of the algorithm you used, showing you understand what it is doing.
3. Compute depth of each pixel. Compute 3D location of each pixel.
4. Train a road classifier on a set of annotated images, and compute road pixels in your image. Which features would you use? Try to use both 2D and 3D features.
5. Train your algorithm on the training data, and show example outputs on the testing images.
6. Fit a plane in 3D to the road pixels by using the depth of the pixels. Make sure your algorithm is robust to outliers.
7. Plot each pixel in 3D (we call this a 3D point cloud). On the same

Sample Project 4

(do not pick this. just an example to give you a sense about size/scope)

Part 2 object detection:

1. Download the left and right colour images for the object recognition dataset and the camera calibration matrices: Info and link. You will find training and testing data, do not mix them up.
2. Detect cars in the image. You can use the pre-trained models available here: link, and detection code available here: link
3. Train a classifier that predicts viewpoint for each car. The viewpoint labels are in 30. increments, thus train 12 classifiers. Which features would you use?
4. Show a test image with the detected car bounding boxes and show the estimated viewpoints by plotting an arrow in the appropriate direction.
5. Given the ground plane, estimated depth, and the location of the car?s bounding box, how would you compute a 3D bounding box around each detected car? Add the 3D bounding boxes to your plot from 1.g.

Sample Project 4

(do not pick this. just an example to give you a sense about size/scope)

Part 3: Make a result video of your algorithm running on the road recognition test set, see **getFrame.m** and **videowriter.m**.
Interesting result videos from different KITTI datasets and projects:

- Road and object segmentation: Video.
- Lane detection: Video.
- Card, pedestrian and Cyclist detection: Info Video.

Building Your Own and Review!

(do this!)

1. Prepare a more detailed proposal.
2. Download data (e.g. use Google dataset search, or Kaggle), or build your own.
3. Your project must integrate at least 3 of the courses main subjects: Image processing, Features and Matching, Geometry, and Recognition
    1. The fewer the number of subject categories used, the more depth, detail, experimentation, and implementation expected in each.