# CS 461/661 Computer Vision
# Final Project

Proposal Due: Oct. 28, 2019

Report Due: Dec. 6, 2019

The objective of the final project is to make use of what you have learned during this course to explore potential solutions to problems using computer vision. The final deliverables will be

- A written report of 2-4 pages.
- Source code of your implementation.
- A brief (<10 minute) oral discussion of your project and your results during class.

You will work in **groups of 3 or 4**. To facilitate forming teams, we have created a team search form on piazza. You could either look for a team or look for more students.

Only one member of your team will need to submit your work on Gradescope, but make sure that this member successfully adds all team members on Gradescope.

Below we will define some example projects that you can work on, but feel free to propose other projects that you find interesting. The goal of the project is to explore and to compare different approaches to whatever problem you choose to work on. Grading will reflect the thoroughness and quality of that exploration.

We will ask the following questions:

- How challenging is this problem?
- What approaches did you take to solve this problem, and why?
- How did you explore the space of solutions -- architectures, hyperparameters, training methods etc. -- your chosen approach(es) presented?
- How did you evaluate the performance of the final approach(es) you investigated?

The key point here is that we're not just interested in the absolute performance of what you produce, but also the thought process that led you there. Ideally, you will have done some comparative performance analysis to back up the choices you made.

Grading will be based on the completeness of the project, the clarity of the writeup, the level of complexity/difficulty of the project, and your ability to justify the choices you made.

# Schedule

**October 28:** By this date, we should know who is working together, and we should have a short (no more than 1 page) project proposal.

**November 19:** By this date, you should have done some initial exploration and you should report the progress you've managed to make and/or roadblocks you've hit.

**December 3/5 :** You make a short (< 10 minute) presentation on your project.

**December 6:** A final 2-4 page report is due, plus source code that can be tested on an example that is also submitted with the code, or is otherwise readily available.

# Grading

There are three levels of grading for these projects:

**Baseline**: The baseline project is to show reasonable (aka near state of the art) performance on a basic dataset. You should assess the performance of your solution and be able to meaningfully discuss the choices you made. This would be 80-90% credit.

**Preferred**: The preferred project will perform baseline operations, but explore a broader set of solutions and/or test on additional data sets and be able to discuss the results of those investigations. This would get 90-100% credit.

**Deluxe**: A deluxe project would include exploration of more approaches, the creation of a stand-alone app, a substantial data set to enhance training and/or test generalization, and so forth. Credit for deluxe items (up to an additional 10%) will be given on a case-by-case basis.

Grade breakdown:

- Initial Concept - 10%
- Progress Report - 10%
- Write up: Implementation and rationale - 30%; Testing and results - 25%
- Final Demonstration - 25%

# Project Proposal Template

You 1-page-long project proposal should include the following:

- Team member list
- Project description: 2-3 paragraphs including:
  - A definition of your problem.
  - Some details of how you propose to solve your chosen problem including a breakdown into approaches/components.
  - The datasets you've identified (to the extent needed for your project).
  - At least one (ideally 2-3) papers or similar expositions that form a starting point for your investigation.
- Team member assignments (who does what).

We are happy to entertain some "special" projects that do not fit this template if you already have something specific in mind where you've done some initial work, but in that case you need to give us a written proposal which outlines the scope of the project and your goals.

# Final Report Format

Your final report should document the following:

- Your initial project goals.
- The goals you achieved.
- The computer vision methods you used to achieve these goals.
- Known limitations and possible future extensions.
- The two or three things you learned during this project.
- Advice you would give to next year's CV students (and instructors).

# Additional Resources

We can request Google Cloud Credits for teams that need additional resources.

# Example Projects

## Image Sorting

The goal of this project is to take a pile of photos and sort and filter them. Ideally, the criteria by which the photos are sorted are user-definable -- e.g. by photos that are outdoor vs. indoor, beach, show, mountains, containing people, urban, and so forth. It would also be helpful to group similar photos (e.g. multiple shots of the same thing) so that its not necessary to look at all of them.

There are many ways to approach this problem. For example, you can use some texture operators, color histograms, and bag of words ideas to define categories. You can use feature-based methods and/or histograms to test for photo similarity, and you could use object detection methods using convolution neural networks to detect objects. Or, you could use pre-trained networks for semantic segmentation, or train your own networks to do certain types of categorization.

References:


- A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- V. Bettadapura, D. Castro, and I. Essa. Discovering picturesque highlights from egocentric vacation videos. In *WACV*, 2016.
- J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *CVPR*, 2006.
- S. O'Hara and B. A. Draper. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv:1101.3354*, 2011.


## Object Counting

Counting objects is a highly useful process. For example, logs loaded on a truck coming out of the forest provides an estimate of the amount of timber harvested, the number of pipes on a truck provides an estimate of weight, or the number of elephants in an image provides an estimate for population. In this project, you will build a counting app. There are two variations on this project:

1. Build an app to count specific types of objects -- people, cars, ships, planes, etc.
2. More challenging, but more useful -- build an app where the user can provide a small number of examples, and the app counts similar things.

In your final report, it will be important to identify some data sets that you can use to assess performance.

References:


- C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In *ECCV*, 2014.
- P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh. Counting everyday objects in everyday scenes. In *CVPR*, 2017.
- V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.
- T. Moranduzzo and F. Melgani. Automatic car counting method for unmanned aerial vehicle images. In *TGRS*, 2014.
- E. Walach and L. Wolf. Learning to count with CNN boosting. In *ECCV*, 2016.


## Object Tracking

The goal of this project is to track objects through video. The canonical example would be to acquire video from a fixed surveillance camera, and to e.g. track cars in that video. But, many other possibilities exist. For example, tracking people in a street scene, tracking animals as they move across the savanna, or tracking birds as they fly. There is also a question of how detailed the track is -- are you just providing location, or do you also compute change in orientation or configuration.

There are several different approaches you can take, in which the word "tracking" actually means very different tasks. For example, you can focus on building a tracker that tracks one target of arbitrary type, which is manually initialized at the beginning of the track. Since the type of the object is arbitrary, the tracker cannot rely on an object detector that works only on a limited set of object categories.

On the other hand, in many multi-object tracking tasks we only care about objects of known categories (e.g., cars). In this case, target objects can be detected using a separate detector, and the tracker should focus on how to link the detected objects together.

References:


- G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. In *TPAMI*, 1998.
- C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. In *TPAMI*, 2001.
- G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. In *CVPR*, 2004.
- J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. In *TPAMI*, 2015.
- L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.
- W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, 2015.

## 3D Reconstruction

Given a video sequence or similar collection of images, produce a 3D reconstruction of the environment. The tools to build a sparse reconstruction using bundle adjustment are widely available, so the objective here is to see how far you can go toward building up a dense reconstruction from "data in the wild." The references below provide some ideas for getting started.

References:


- Snavely, Noah, Steven M. Seitz, and Richard Szeliski. Photo Tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)*, 2006.
- H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. In *TPAMI*, 2012.
- C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.
- B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *CVPR*, 2016.