Overview

This Project is an opportunity for you to apply what you have learned in class to a problem of your interest. Potential projects usually fall into these two tracks:

- **Applications.** If you're coming to the class with a specific background and interests (e.g. biology, engineering, physics), we'd love to see you apply ConvNets to problems related to your particular domain of interest. Pick a real-world problem and apply ConvNets to solve it.
- **Models.** You can build a new model (algorithm) with ConvNets, or a new variant of existing models, and apply it to tackle vision tasks. This track might be more challenging, and sometimes leads to a piece of publishable work.

One **restriction** to note is that this is a project on Computer Vision, so your project should involve pixels of visual data in some form somewhere. E.g. a pure NLP project is not a good choice, even if your approach involves ConvNets.

To get a better feeling for what I expect from projects, I encourage you to take a look at the project reports from Stanford CS231n class:

- Spring 2017
- Winter 2016
- Winter 2015

To inspire ideas, you might also look at recent deep learning publications from top-tier conferences, as well as other resources below.

- CVPR: IEEE Conference on Computer Vision and Pattern Recognition
- ICCV: International Conference on Computer Vision
- ECCV: European Conference on Computer Vision
- NIPS: Neural Information Processing Systems
- ICLR: International Conference on Learning Representations
- ICML: International Conference on Machine Learning
- Publications from the Stanford Vision Lab
- Awesome Deep Vision
- Past CS229 Projects: Example projects from Stanford's machine learning class
- Kaggle challenges: An online machine learning competition website. For example, a Yelp classification challenge.

For applications, this type of projects would involve careful data preparation, an appropriate loss function, details of training and cross-validation and good test set evaluations and model comparisons. Don’t be afraid to think outside of the box. Some successful examples can be found below:

- Teaching Deep Convolutional Neural Networks to Play Go
- Playing Atari with Deep Reinforcement Learning

- Winning the Galaxy Challenge with convnets

ConvNets also run in real time on mobile phones and Raspberry Pi's - building an interesting mobile application could be a good project. If you want to go this route you might want to check out TensorFlow Mobile / Lite or Caffe2 iOS/Android integration.

For models, ConvNets have been successfully used in a variety of computer vision tasks. This type of projects would involve understanding the state-of-the-art vision models, and building new models or improving existing models for a vision task. The list below presents some papers on recent advances of ConvNets in the computer vision community.

- **Image Classification**: [Krizhevsky et al.], [Russakovsky et al.], [Szegedy et al.], [Simonyan et al.], [He et al.], [Huang et al.], [Hu et al.] [Zoph et al.]
- **Object detection**: [Girshick et al.], [Ren et al.], [He et al.]
- **Image segmentation**: [Long et al.] [Noh et al.] [Chen et al.]
- **Video classification**: [Karpathy et al.], [Simonyan and Zisserman] [Tran et al.] [Carreira et al.] [Wang et al.]
- **Scene classification**: [Zhou et al.]
- **Face recognition**: [Taigman et al.] [Schroff et al.] [Parkhi et al.]
- **Depth estimation**: [Eigen et al.]
- **Image-to-sentence generation**: [Karpathy and Fei-Fei], [Donahue et al.], [Vinyals et al.] [Xu et al.] [Johnson et al.]
- **Visualization and optimization**: [Szegedy et al.], [Nguyen et al.], [Zeiler and Fergus], [Goodfellow et al.], [Schaul et al.]

You might also gain inspiration by taking a look at some popular computer vision datasets:

- Meta Pointer: A large collection organized by CV Datasets.
- Yet another Meta pointer
- ImageNet: a large-scale image dataset for visual recognition organized by WordNet hierarchy
- SUN Database: a benchmark for scene recognition and object detection with annotated scene categories and segmented objects
- Places Database: a scene-centric database with 205 scene categories and 2.5 millions of labelled images
- NYU Depth Dataset v2: a RGB-D dataset of segmented indoor scenes
- Microsoft COCO: a new benchmark for image recognition, segmentation and captioning
- Flickr100M: 100 million creative commons Flickr images
- Labeled Faces in the Wild: a dataset of 13,000 labeled face photographs
- Human Pose Dataset: a benchmark for articulated human pose estimation
- YouTube Faces DB: a face video dataset for unconstrained face recognition in videos

- UCF101: an action recognition data set of realistic action videos with 101 action categories
- HMDB-51: a large human motion dataset of 51 action classes
- ActivityNet: A large-scale video dataset for human activity understanding
- Moments in Time: A dataset of one million 3-second videos

Collaboration Policy

You can work in teams of up to **2** people. I do expect that projects done with 2 people have more impressive writeup and results than projects done by 1 person.

Honor Code

You may consult any papers, books, online references, or publicly available implementations for ideas and code that you may want to incorporate into your strategy or algorithm, so long as you clearly cite your sources in your code and your writeup. However, under no circumstances may you look at another group's code or incorporate their code into your project.

Final Report

Your final write-up is required to be between **6 - 8** pages using the provided template, structured like a paper from a computer vision conference (CVPR, ECCV, ICCV, etc.). Please use this template so we can fairly judge all student projects without worrying about altered font sizes, margins, etc.

The following is a suggested structure for your report, as well as the rubric that we will follow when evaluating reports. You don't necessarily have to organize your report using these sections in this order, but that would likely be a good starting point for most projects.

- **Title, Author(s)**
- **Abstract**: Briefly describe your problem, approach, and key results. Should be no more than 300 words.
- **Introduction (10%)**: Describe the problem you are working on, why it's important, and an overview of your results
- **Related Work (10%)**: Discuss published work that relates to your project. How is your approach similar or different from others?
- **Data (10%)**: Describe the data you are working with for your project. What type of data is it? Where did it come from? How much data are you working with? Did you have to do any preprocessing, filtering, or other special treatment to use this data in your project?
- **Methods (30%)**: Discuss your approach for solving the problems that you set up in the introduction. Why is your approach the right thing to do? Did you consider alternative approaches? You should demonstrate that you have applied ideas

and skills built up during the quarter to tackling your problem of choice. It may be helpful to include figures, diagrams, or tables to describe your method or compare it with other methods.

- **Experiments (30%)**: Discuss the experiments that you performed to demonstrate that your approach solves the problem. The exact experiments will vary depending on the project, but you might compare with previously published methods, perform an ablation study to determine the impact of various components of your system, experiment with different hyperparameters or architectural choices, use visualization techniques to gain insight into how your model works, discuss common failure modes of your model, etc. You should include graphs, tables, or other figures to illustrate your experimental results.
- **Conclusion (5%)** Summarize your key results - what have you learned? Suggest ideas for future extensions or new applications of your ideas.
- **Writing / Formatting (5%)** Is your paper clearly written and nicely formatted?
- **Supplementary Material**, not counted toward your 6-8 page limit and submitted as a separate file. Your supplementary material might include:
    - Source code (if your project proposed an algorithm, or code that is relevant and important for your project.).
    - Cool videos, interactive visualizations, demos, etc.
- Examples of things to not put in your supplementary material:
    - The entire TensorFlow Github source code.
    - Any code that is larger than 10 MB.
    - Model checkpoints.
    - A computer virus.

Submission: You will submit your final report as a PDF and your supplementary material as a separate PDF or ZIP file. We will provide detailed submission instructions as the deadline nears.