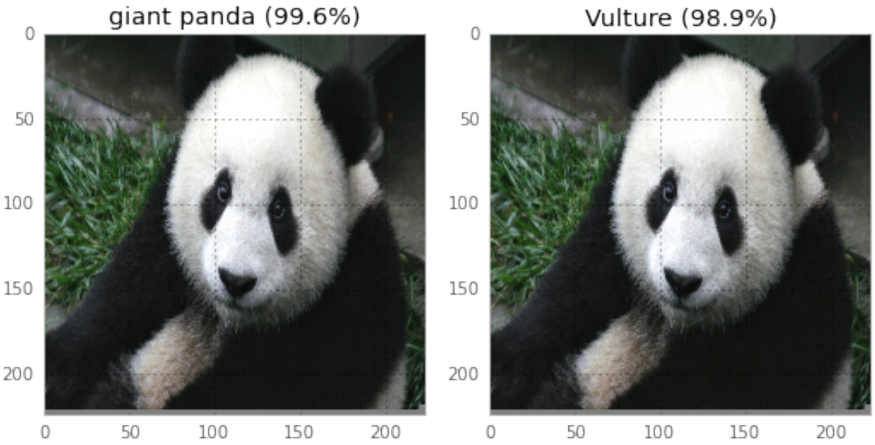
CSC 2547: Machine Learning for Vision as Inverse Graphics

Anthony Bonner
www.cs.toronto.edu/~bonner

Convolutional Neural Nets

- Achieved astounding breakthroughs in machine vision (and other areas).
- Require vast amounts of data for learning.
- Make silly mistakes.
- Do not understand what they see.
- Limitations have been extensively studied.
- Not full Artificial Intelligence.

Tricking a Neural Net



Read about it here (and try it!): https://codewords.recurse.com/issues/five/why-do-neural-networks-think-a-panda-is-a-vulture

Watch: https://www.youtube.com/watch?v=M2IebCN9Ht4

Scene Understanding

- Much more than just classification.
- Needs a rich 3-dimensional representation of the world.
- Objects, shape, position, orientation, appearance, category, composition, …
- Relationships between objects.
  (part-of, next-to, on-top-of, …)
- Illumination, camera angle, …

Inverse Graphics

- Computer graphics represents the world this way internally.
- Inverse problems:
  - Graphics generates a 2D image from a 3D representation.
  - Scene understanding generates a 3D representation from a 2D image.

After inferring a 3D representation of an image:

- Could answer many common-sense questions about the image.
- Could (approximately) regenerate the image with a graphics program.
- Deviations from the original image show the accuracy of the representation (useful for learning).
- Could modify the representation:
  - move, rotate, recolor objects
  - change illumination, camera position
- Generate modified images.

Inverting the Graphics process

- A difficult, non-deterministic problem.
- Loss of information in the graphics process.
- Many 3D representations have the same image (due to occlusion, shadows, etc,)
- Which one is right?

This Course

- Explores machine-learning approaches to inverse graphics
  - inferring internal representations of an image
    - distributions of representations
    - structured/interpreted representations
    - spatial/geometric properties
  - using these internal representations
    - to generate images from internal representations (generative models)
    - to test the accuracy of inference (feedback for learning)
    - to answer questions about an image
- Does not solve these problems.
- Does not do graphics.

Course Structure

- Seminar course with a major project.
- Study papers from the literature.
- First 3 classes:
  lectures on background material.
- Next 7 classes:
  student presentations of papers.
- Last 2 classes:
  project presentations

Paper Presentations

- Each week will focus on one topic, as listed on the course web page (soon).
- You can vote for your choice of topic/week (soon).
- I will assign you to a week (soon).
- Papers on each topic will be listed on the course web page.
- If you have a particular paper you would like to add to the list, please let me know.

Paper Presentations

- Goal: high quality, accessible tutorials.
- 7 weeks and 40 students = 6 students per week and 20 minutes per student (including questions).
- 2-week planning cycle:
  - 2 weeks before your presentation, meet me after class to discuss and assign papers.
  - The following week, meet the TA for a practice presentation (required).
  - Present in class under strict time constraints.

Team Presentatations

- Papers may be presented in teams of two or more with longer presentations (20 minutes per team member).
- Unless a paper is particularly difficult or long, a team will be expected to cover more than one paper (one paper per team member).
- A team may cover one of the listed papers and one or more of its references (but see me first).

Tentative Topics

- Variational inference and autoencoders
- Capsule networks
- Point Nets
- Group symmetries and equivariance
- Visual attention mechanisms
- CNNs for 3D
- Part-whole relationships
- Contrastive Learning
- Adversarial Learning

Marking Scheme

- Paper presentation: 20%
- Course project Proposal: 20%
- Project presentation: 20%
- Project report and code: 40%

Prerequisites

- Solid introduction to machine learning (eg, grad or senior undergrad course)
- Familiarity with neural nets and CNNs
- Solid background in linear algebra
- Basics of multivariate calculus and probability
- Programming skills (eg, Tensorflow or Pytorch if you plan an implementation project)
- Mathematical maturity will be assumed.

More information

- See the course website.
- Accessible through my home page:
  www.cs.toronto.edu/~bonner
- Announcements will be made through Quercus and Piazza.