Context for vehicles and sensing

- **Environments:**
  - **Case 1:**
    - work anywhere
    - likely: various gadgets improve safety and experience
    - implausible: full autonomy
  - **Case 1.5:**
    - work some places
    - some are specialized to freeways, etc.
  - **Case 2:**
    - work only in tightly controlled environment (eg smart city)
    - there are models of full autonomy (eg transporters at airports)
Case 1

- Various gadgets improve safety and experience
- case-by-case reasoning about representation and sensing
- Issues:
  - what’s worth doing?
  - what can be done easily?
  - how much sensing?
Case 1 examples

• Reversing cameras
• Reversing sonar
• Forward sonar for collision avoidance
• Active collision management
• Pedestrian detection
• Various safety cameras
  • driver attention
  • record events for dispute resolution
  • driver sobriety
• Smarter links to maps
Case 1.5:

• Mostly, more specialized gadgets, mostly for highways
  • lane following for highways
  • predicting highway turnoffs
  • speed control that’s aware of cars in front
  • neat tricks to reduce traffic jams

• Issues
  • what’s worth doing?
  • what can be done easily?
  • how much sensing?
Case 2: Strongly controlled environments

- Full autonomy quite plausible
  - depending on regulatory and environmental control
    - there are models of full autonomy (eg transporters at airports)
  - This case is valuable, and may be important
    - public transport -> apartment in high density living areas

- Issues:
  - how much control do you need?
  - what density of traffic can be sustained?
  - how do you ensure safe behavior if weird stuff happens?
The questions that will plague us

- What representation do we need?
- How much data do we need to make it?
  - and where do we get it?
- How do we know if it works
Representation

• A1: No representation required
  • link control inputs to sensing with multiple network layers
  • train on simulation with reinforcement learning
  • dubious position, but…
    • notice that, IN PRINCIPLE, this deals with full autonomy
  • Q:
    • how do you know it will do the right thing in a given situation?
  • A (dubious)
    • watch what it does on training data
Representations

- **A2: 3D reconstruction**
  - build complete 3D model of world around you
    - LIDAR, SFM, etc.
    - label it with appropriate labels (next slides)
  - use a planner, etc to make paths in that environment
  - follow paths
  - Q:
    - how do you know it will do the right thing in a given situation?
  - A (dubious):
    - prove that environment is right and software is correct
  - Q:
    - do you really need a 3D representation?
  - A:
    - who knows?
Optic flow as a theory of perception
Fun fact about vision

Fun fact: time to contact = \( \frac{x}{dx/dt} \)
TTC - Long
TTC - AAARGH!
Likely truth about 3D vs 2D

- Straightforward to convert from 2D to 3D reps and back
- This means anything you can do w/3D, you can do w/2D
- But: convenience is important
  - some planners want 3D
  - sensing 3D as 3D might be a good idea (LIDAR)
  - detection is generally faster in 2D, might be easier
Representation

• A3: Label images (or 3D reconstruction)
  • with what?
    • label all possible objects with all names
    • label some classes, ignore others
  • what taxonomy?
    • likely a derived taxonomy from actions
  • Q:
    • how do you know it will do the right thing in a given situation?
  • A (dubious):
    • prove that environment is right and software is correct
  • Q:
    • what should be labelled and what should be ignored?
  • A:
    • who knows? likely the things that most affect performance?
Labelling
Labelling

MS-CoCo
Labelling
The questions that will plague us

- What representation do we need?
- How much data do we need to make it?
  - and where do we get it?
- How do we know if it works?
XXXXX Autonomy data
Special features: rich appearance variation
Special features: rich appearance variation
Special features: rich appearance variation
Standard semantic segmenter
XXX data consequences
XXXX data consequences
GPS is quite good, but not perfect
Another curious data problem
The questions that will plague us

- What representation do we need?

- How much data do we need to make it?
  - and where do we get it?

- How do we know if it works?
Image classification

Image

Some neural stuff; differentiable wrt parameters, input

Cat
Dog

.
.
.

Car
Adversarial example

- Search for
  - small update to image
  - such that
    - output for true class is low
    - output for some other class is high

- Surprising fact:
  - such updates can be VERY small
Correctly classified

“Ostrich”

Szegedy et al, 13
Fast gradient sign search

- Search $\text{sign}(\text{gradient})$
  - $x$ is image
  - $J$ is some cost
    - eg $J = (\text{true class}) - (\text{best false class})$

- Iterate

$$\alpha = \epsilon \text{sign}(\nabla_x J(\theta, x, y)).$$

$$X_{\text{adv}} = X - \epsilon \cdot \text{sign}(\nabla_x J(X, y_{\text{foot}})).$$

Goodfellow et al 15
Fast gradient sign

\[ x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \]

\( x \)

“panda”
57.7% confidence

sign(\(\nabla_x J(\theta, x, y)\))

“nematode”
8.2% confidence

\( x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \)

“gibbon”
99.3% confidence
Deepfool

- Find $r$ by
  - linearize $k$
    - update
  - (possibly) repeat

\[
\Delta(x; \hat{k}) := \min_{r} \|r\|_2 \text{ subject to } \hat{k}(x + r) \neq \hat{k}(x)
\]
Deepfool

Whale

Turtle

Difference image

Moosavi-Desfooli et al 16
Compare fast gradient sign

Turtle

Difference image

Moosavi-Desfooli et al 16
Flow based methods

- New image obtained by

Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

Xiao et al
2018
Flow based methods

- New image obtained by moving pixels
  \[ \chi_{adv}(u, v) = \chi(u + f_u, v + f_v) \]

\[ f^* = \arg\min_f \quad \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f) \]

Xiao et al. 2018
Flow methods

Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

Xiao et al 2018
Flow methods

(a) mountain bike (b) goldfish (c) Maltese dog (d) tabby cat

Xiao et al 2018
Yolo attack

- Yolo uses a large image area to
  - predict boxes
  - predict classes
- This means that a detection is
  - affected by pixels OUTSIDE box