The Question:

Are CNNs always the best choice for every computer vision problem?

The Idea:

Sometimes simpler algorithms are the better choice.

Simpler algorithms can be guided by CNNs to overcome limitations.

Simplify the problem and leverage constraints to utilize simpler algorithms.

The Story:

Creating the Smart Garage (Chapter 1)
Company Background

CHAMBERLAIN GROUP

Chamberlain Group (CGI) is a global leader in access solutions and products.

Over 6,000+ Employees Worldwide
CGI is a global team with solutions and operations designed to serve customers in a variety of markets worldwide.

VISION
Giving The Power Of Access And Knowledge

MISSION
People Everywhere Rely On CGI To Move Safely Through Their World, Confident That What They Value Most Is Secure Within Reach

END-MARKETS SERVED
Residential Commercial Automotive

© 2020 Chamberlain Group
Creating the Smart Garage

New flagship garage door opener with camera

How do we make this product smarter?

• Know when vehicles leave and arrive.
• Know whether a specific vehicle is in the garage.

Vehicle Presence & Vehicle Re-Identification
CNN Strengths and Weaknesses
CNN Strengths and Weaknesses

Strengths

• Generalize to unseen circumstances
• Can be trained for various tasks
• State-of-the-art results

Weaknesses

• Requires large training dataset
• High compute requirements
• High memory requirements
CNN-Only Solutions: Vehicle Presence

Object Detection (YOLO, SSD)

- Accuracy issues
- Need custom dataset to improve accuracy

Object Tracking

- Analyze video = high cost
CNN-Only Solutions: Vehicle Identification

Make / Model Classification

- Accuracy Issues
- Mix-up similar-looking cars
- Datasets trained for fully-visible vehicles

License Plate Recognition

- Plate not visible when parked
- Analyze video = high cost
- Blur, lighting, compression

Pairwise Similarity

- No truly-free pre-trained models
- No truly free datasets
- Need a large training dataset
More Problems: Cost Constraints

• Existing cameras have relatively low compute:
  • Single core low-cost ARM
  • Low memory

• Minimal cloud expenses allowed
Simplifying the Problem
Rich metadata is fun, but we still must **SIMPPLY** to solve the real problem.

Vehicle Presence & Vehicle Re-Identification
Simplifying the Question

Simplify the question, instead of simplifying the answer.

Vehicle Presence & Vehicle Re-Identification
Find Useful Constraints

Stuck with “bad” constraints? Look for “useful” constraints!

- Fixed camera
- Limited number of vehicles
- Fixed vehicle location (region of interest)
- Personal habits of parking location
- Temporal consistency of the background (When did you last clean your garage?)

Less generalization is needed!
Our Hybrid Solution
Find an Appropriate Simple Algorithm

Selected Algorithm: Modified Histogram of Oriented Gradients (HOG)

- HOG lowers the spatial resolution: built in flexibility
- Features can be translated, to some degree
- Add a custom “attention mask”
- Easy to compare features

- Implement “training mode” to automatically gather images representing known conditions
The “Fast Scene Matcher”

**Train Mode**
1. Gather images of known conditions
2. Extract features
3. Build attention masks
4. Process features into library

**Run Mode**
1. Gather image of unknown condition
2. Extract features
3. Compare to library
4. Find best match, or no match
Combined Meta-Algorithm

Train Mode
- Multiple CNNs (PoseNet, YOLO, etc.)
- Object tracking, ROIs, line crossing
- License plate recognition
- Leverage user input to fix mistakes

Output:
- Processed “known-condition” images

Run Mode
- HOG-based “Fast Scene Matcher”

Output:
(for each parking spot)
- Specific vehicle present
- OR empty
- OR indeterminate
"Guide-Follower" Design Pattern

Guide

Informs

Ask for Help

Follower

• Expensive
• Slow
• Generalized
• Infrequent

Input

Output

• Specialized
• Frequent
• Inexpensive
• Fast
Results
# Results – Single-Frame Run Mode

<table>
<thead>
<tr>
<th>Metric</th>
<th>YOLO-COCO 256x256 (^1)</th>
<th>YOLO-VEH 256x256 (^1)</th>
<th>YOLO-VEH 416x416 (^2)</th>
<th>FSM-HOG-4 (^4)</th>
<th>FSM-HOG-7 (^5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect Any Vehicle Present</td>
<td>41.5%</td>
<td>45.5%</td>
<td>57.8%</td>
<td>86.1%</td>
<td>93.0%</td>
</tr>
<tr>
<td>Detect Vehicle Absent</td>
<td>99.4%</td>
<td>98.7%</td>
<td>99.4%</td>
<td>99.7%</td>
<td>99.4%</td>
</tr>
<tr>
<td>Re-Identify Known Vehicle</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>89.2%</td>
<td>95.9%</td>
</tr>
<tr>
<td>Recognize Unknown Vehicle</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>63.6%</td>
<td>80.4%</td>
</tr>
<tr>
<td>Recognize Moved Camera Position</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>37.5%</td>
<td>29.2%</td>
</tr>
<tr>
<td>Cloud cost per 1,000,000 frames</td>
<td>$7.38</td>
<td>$7.38</td>
<td>$19.50</td>
<td>$0.39</td>
<td>$0.39</td>
</tr>
<tr>
<td>Edge compute on existing hardware</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>2 FPS</td>
<td>2 FPS</td>
</tr>
</tbody>
</table>

1) 256x256 YOLOv2; trained on MS COCO + VOC datasets; threshold 0.3; cost based on AWS g3s.xlarge instance size
2) 256x256 YOLOv2; trained on custom vehicle dataset; threshold 0.35; cost based on AWS g3s.xlarge instance size
3) 416x416 YOLOv2; trained on custom vehicle dataset; threshold 0.35; cost based on AWS g3s.xlarge instance size
4) Unoptimized OpenCV C++ code; 4 example images per known condition; cost based on AWS m4.xlarge instance size
5) Same as (4), but with up to 7 example images per known-vehicle.
Hybrid Solution: Combined Meta-Algorithm

“Train” in the Cloud

Guide

“Inference” on the Edge

on already-deployed devices!

Follower
Take-Aways

1. Simplify the problem first.
   • Identify your core goal.
   • Simplify the question, instead of simplifying the answer.
   • Leverage useful constraints.

2. Sometimes simpler algorithms are the better choice.
   • Simplifying the problem allow use of a less generalized algorithm.
   • Simpler algorithms can be guided by CNNs to overcome limitations.
   • It is possible to preserve most of the “magic” of deep learning, at a lower cost.
Q & A
Resources

Chamberlain

https://www.chamberlain.com/

YOLOv2


Histogram of Oriented Gradients


2020 Embedded Vision Summit

“Combining CNNs and Conventional Algorithms for Low-Compute Vision: A Case Study in the Garage”

Nathan Kopp

1:00pm, September 17, 2020