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

Nathan Kopp
Chamberlain Group
September 2020

# Intro

**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 (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

# 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

CHAMBERLAIN GROUP

# CNN Strengths and Weaknesses

**Strengths**

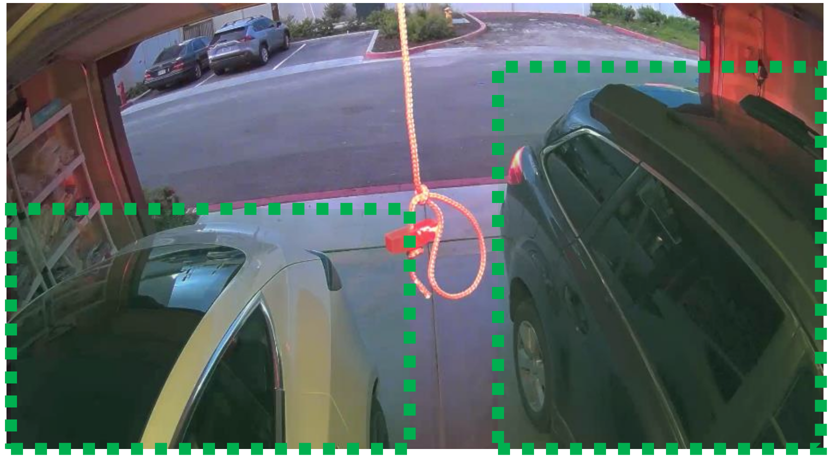
- Generalize to unseen circumstances
- Can be trained for various tasks
- State-of-the-art results

**Weaknesses**

- Requires large training dataset
- High compute requirements
- High memory requirements

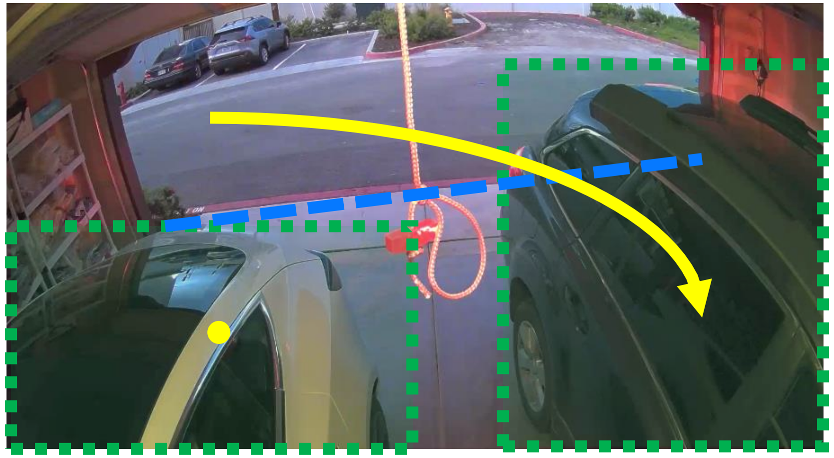
# CNN-Only Solutions: Vehicle Presence

Object Detection (YOLO, SSD)



- Accuracy issues
- Need custom dataset to improve accuracy

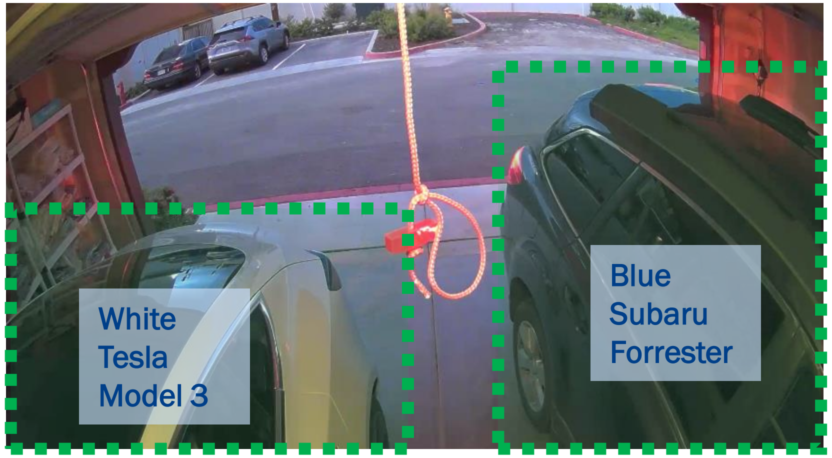
Object Tracking



- 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



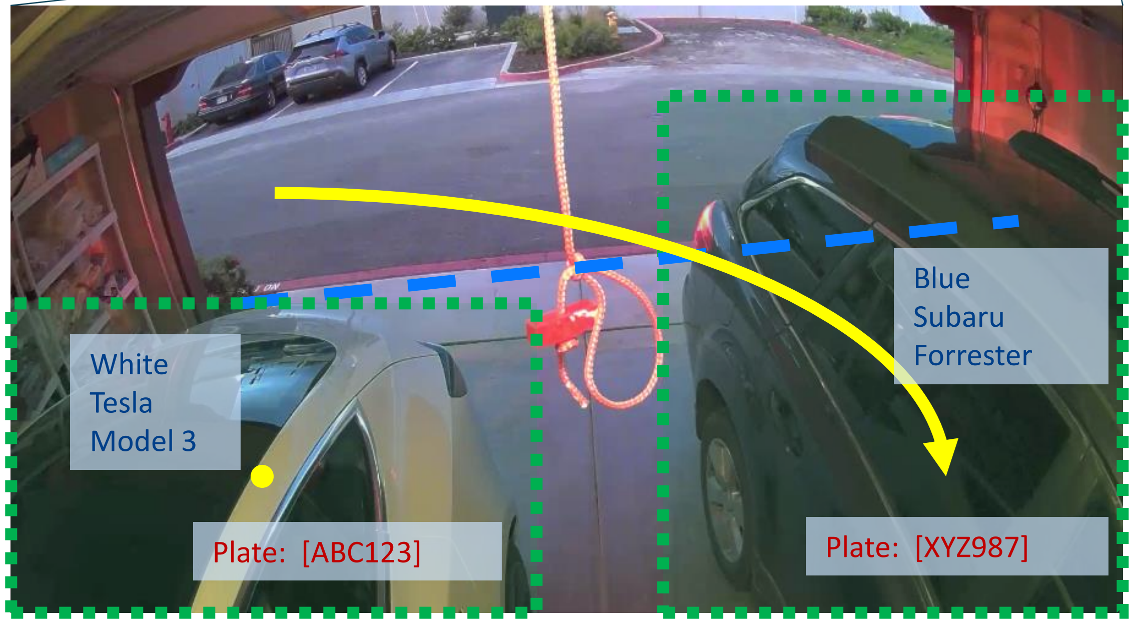
CHAMBERLAIN GROUP

# Simplifying the Problem

CHAMBERLAIN GROUP

# Simplifying the Question

Raw Video → Process with CNN → Rich Metadata → Interpret Metadata → Goal

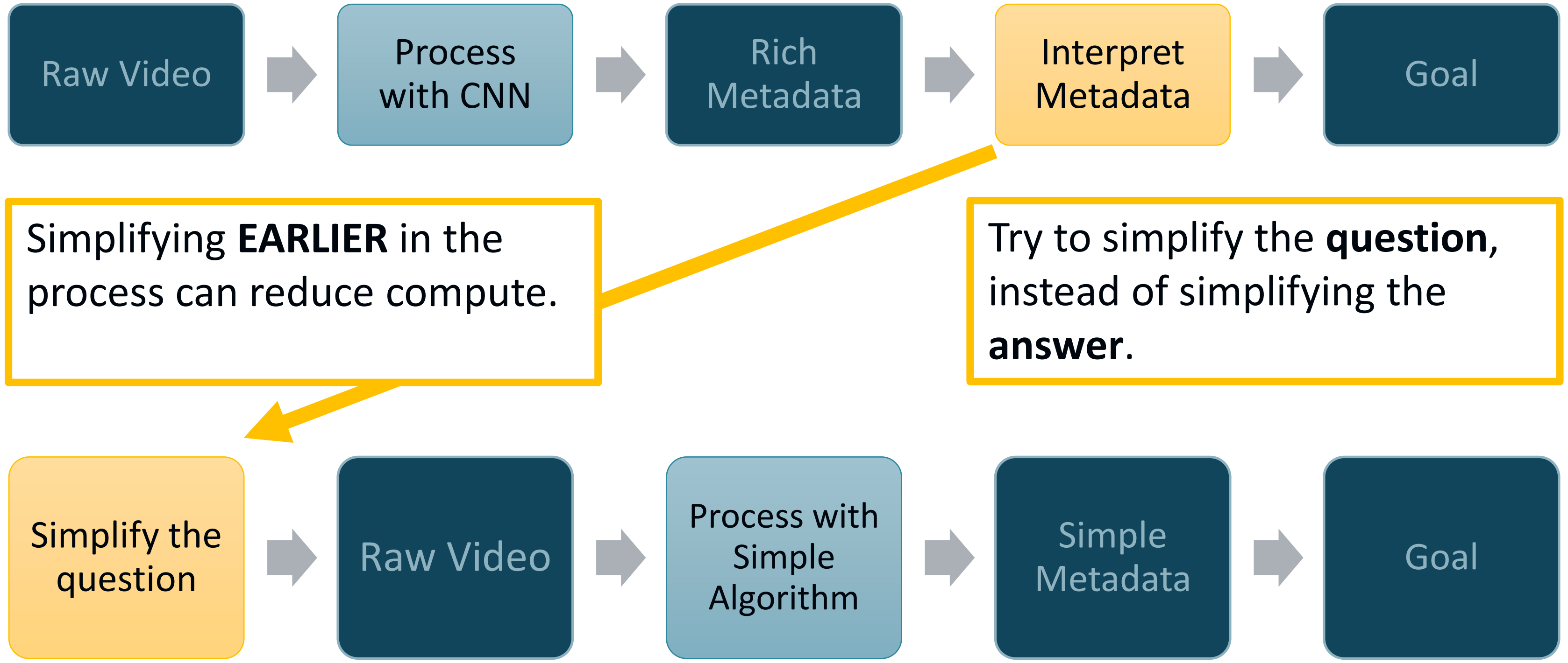



Rich metadata is fun, but we still must SIMPLIFY to solve the real problem.

Vehicle Presence
&
Vehicle Re-Identification

CHAMBERLAIN GROUP

# Simplifying the Question




Vehicle Presence
&
Vehicle Re-Identification

CHAMBERLAIN GROUP

# 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

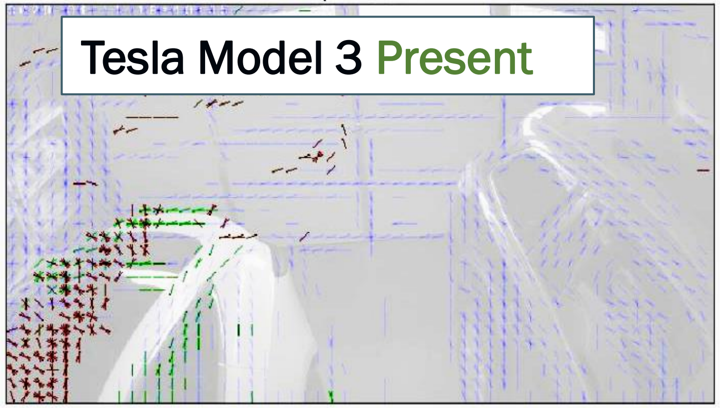
CHAMBERLAIN GROUP

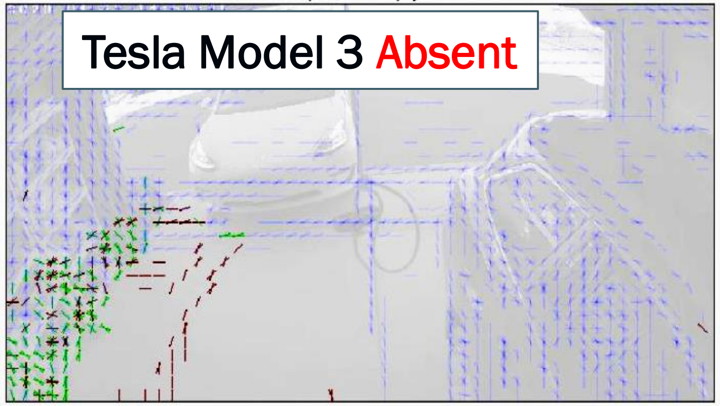
# Find an Appropriate Simple Algorithm

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

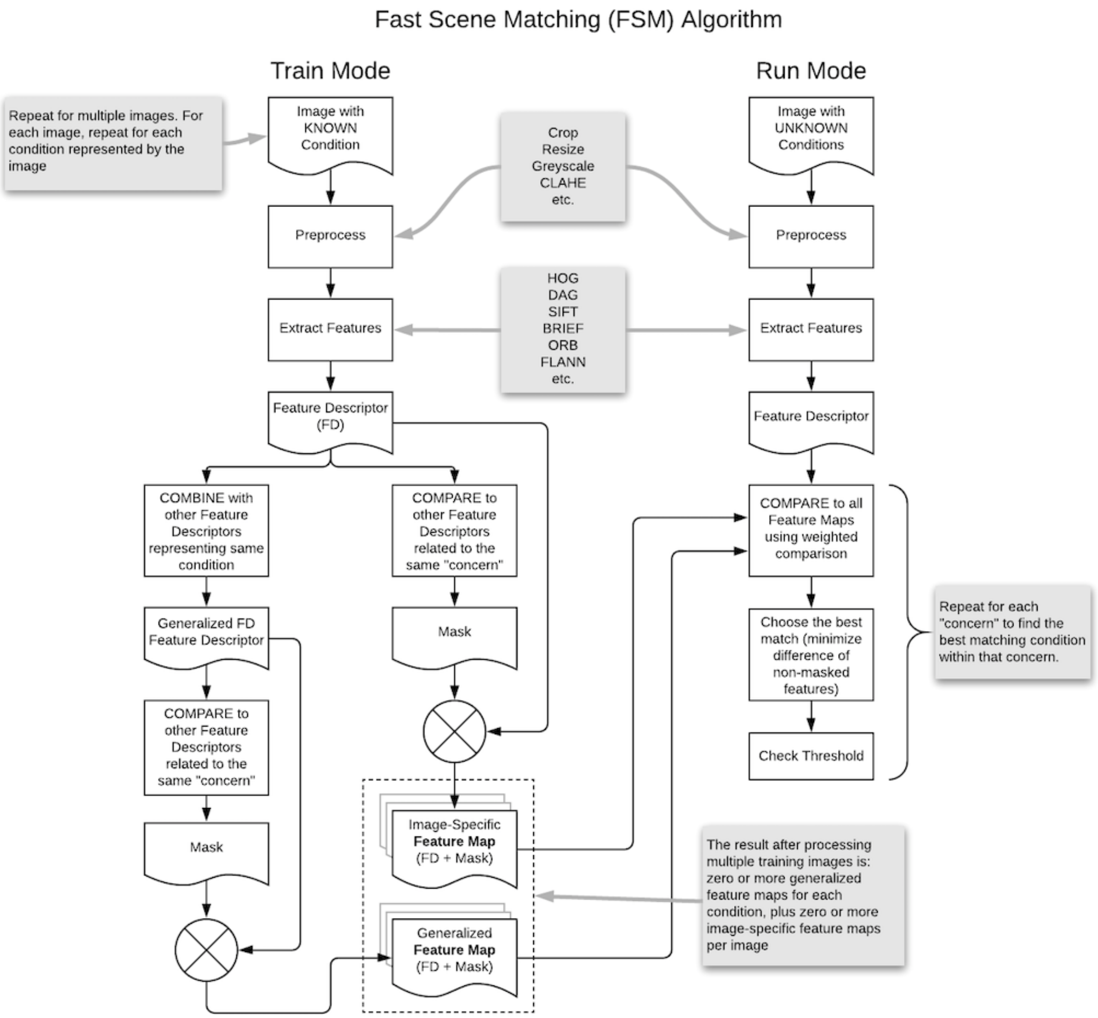
- Implement ”training mode” to *automatically* gather images representing **known conditions**



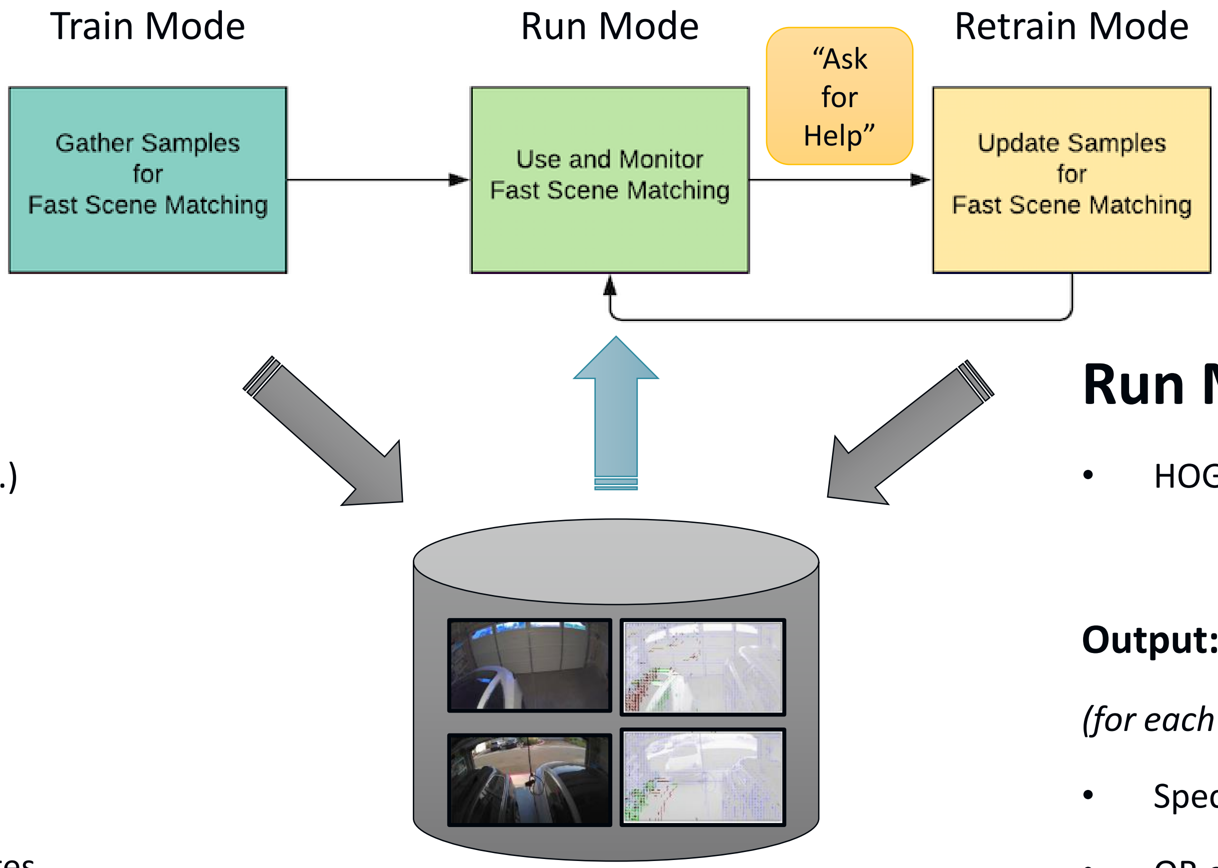
# The "Fast Scene Matcher"

- 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




CHAMBERLAIN GROUP

# 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"

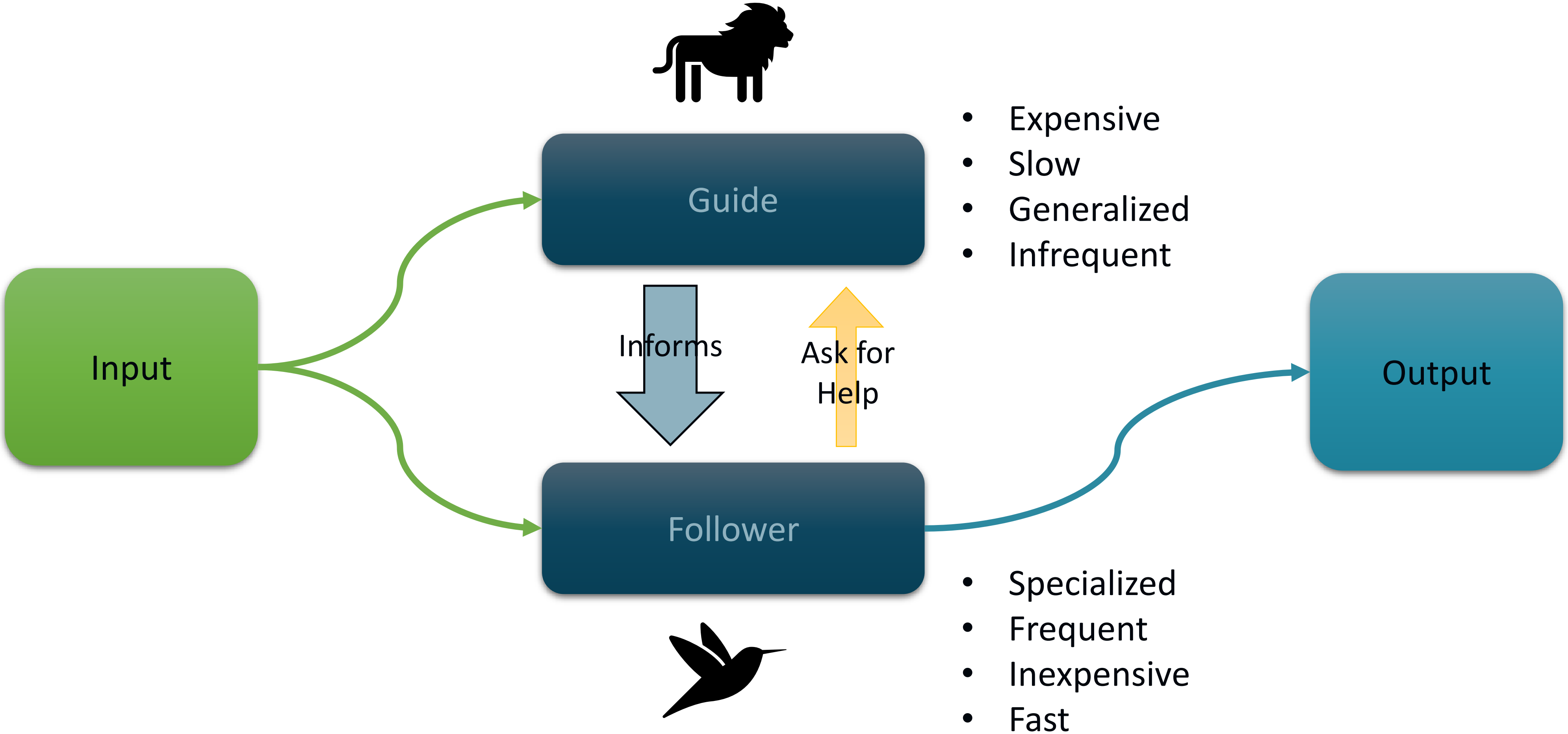
**Output:**

*(for each parking spot)*

- Specific vehicle present
- OR empty
- OR indeterminate

CHAMBERLAIN GROUP

# "Guide-Follower" Design Pattern




- Expensive
- Slow
- Generalized
- Infrequent

- Specialized
- Frequent
- Inexpensive
- Fast

# Results

# Results – Single-Frame Run Mode

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

*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.*

CHAMBERLAIN GROUP

# Hybrid Solution: Combined Meta-Algorithm

2020 embedded vision summit

## "Train" in the Cloud


## "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

CHAMBERLAIN GROUP

# Resources

Chamberlain

https://www.chamberlain.com/

YOLOv2

https://pjreddie.com/darknet/yolo/

Histogram of Oriented Gradients

https://en.wikipedia.org/wiki/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*