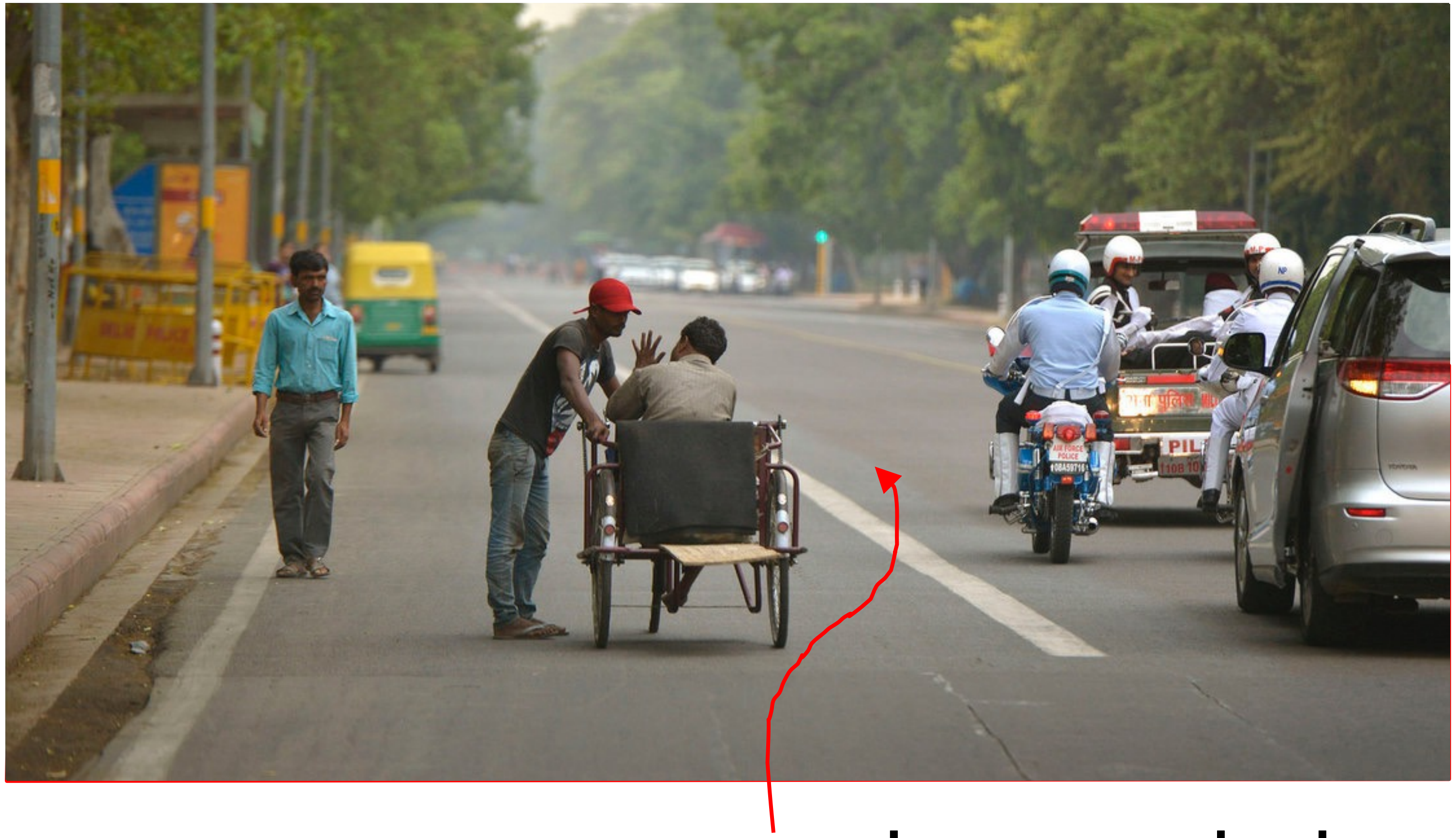
Introduction to Recognition

Saurabh Gupta

Many slides from Justin Johnson

Computer Vision

To extract “meaning” from pixels



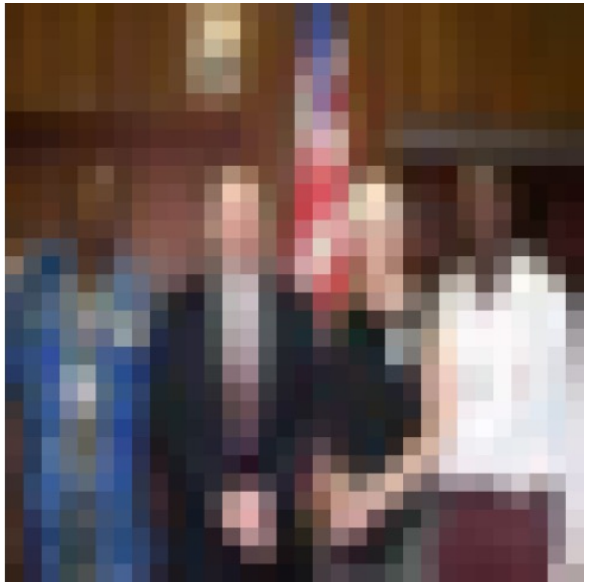
person, motorcycle, car, chair

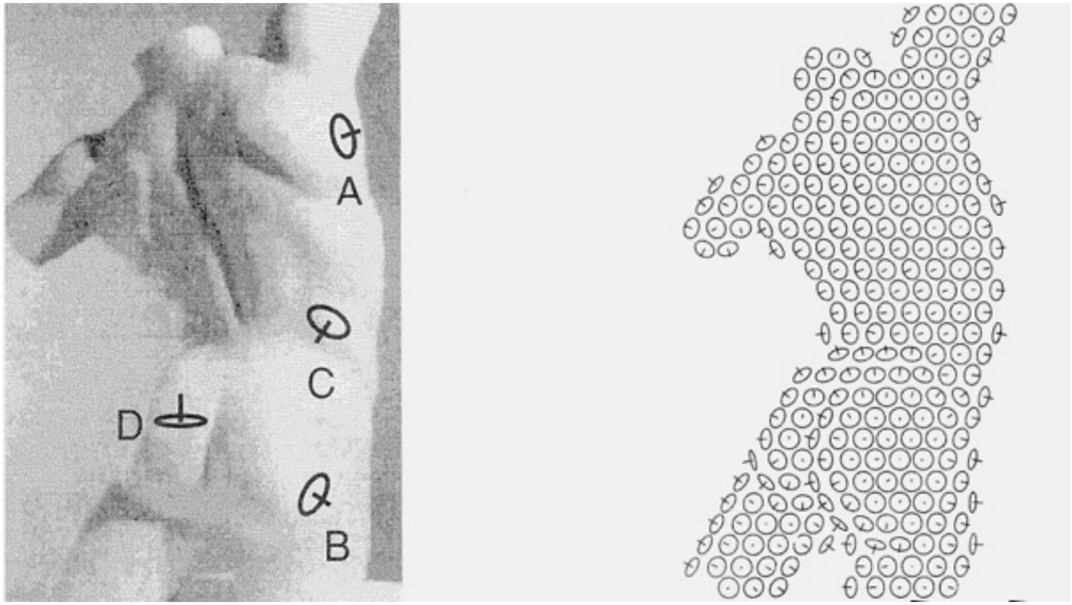
Meaning can take different forms:

- Geometric Inferences
- Semantic Inferences
- Inferences about actions
- …

Computer vision is easy for humans

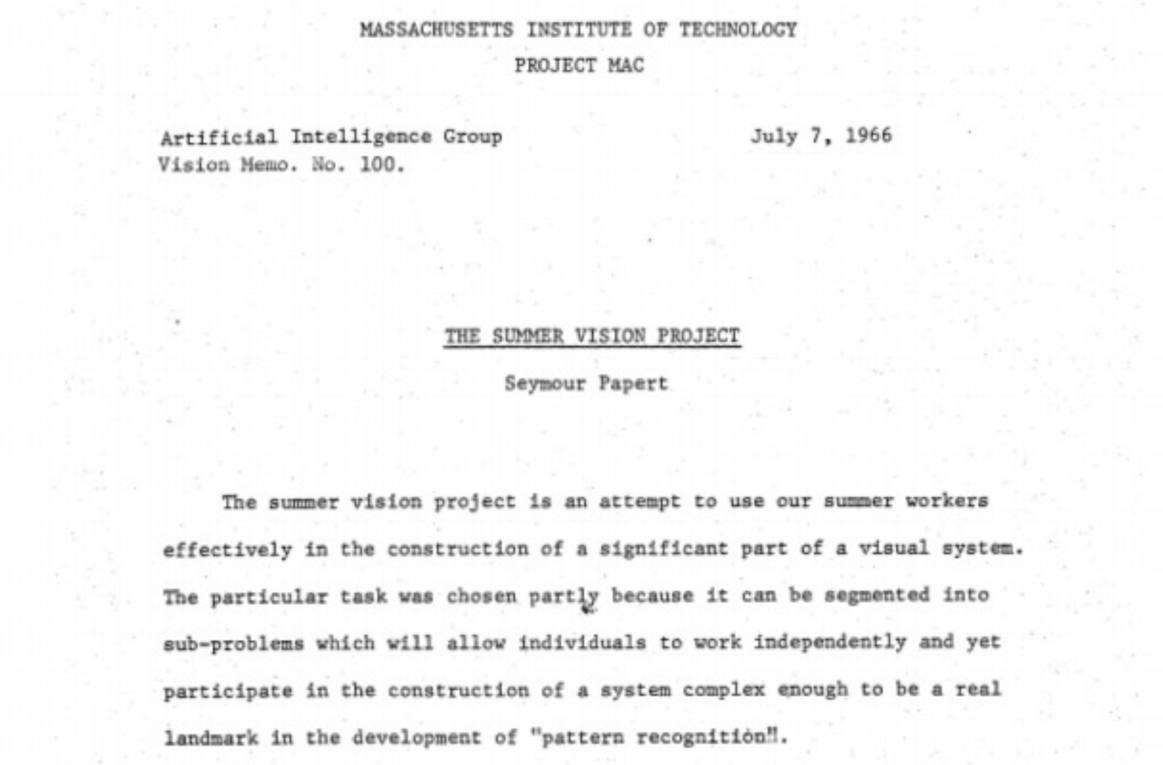
- Effortlessly analyze images for a variety of tasks
- Infer semantics even from severely ablated
- Can also make precise inference about certain geometric properties





Yet has proven very hard for computers

- Computer vision research easily goes back 60 years …



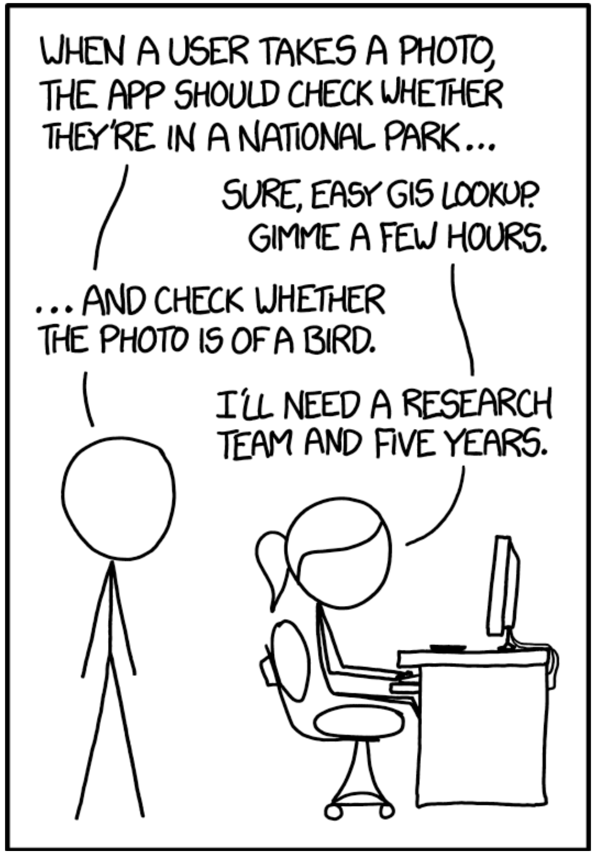
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

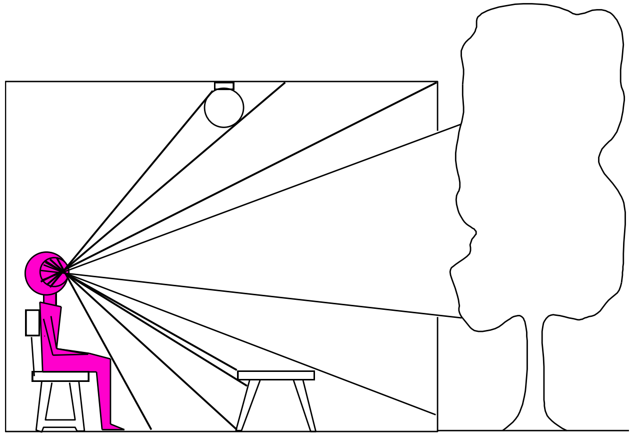

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

- Entirely true as of 2014 (or so) when this xkcd was published

Why is computer vision hard?

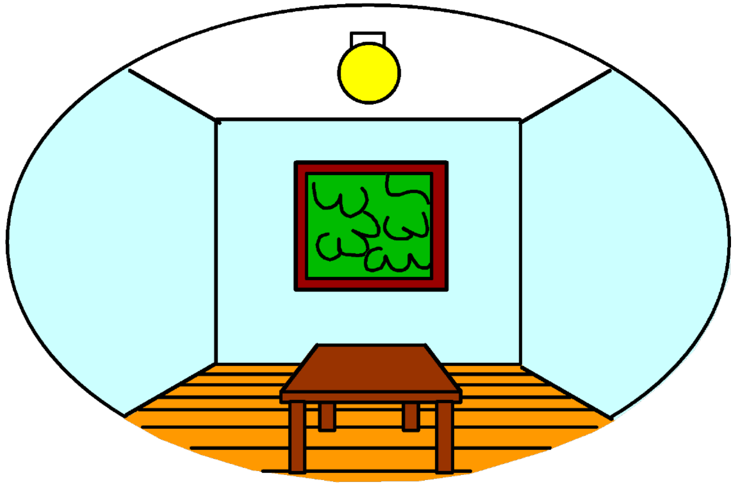
- Images are a lossy projection of the world

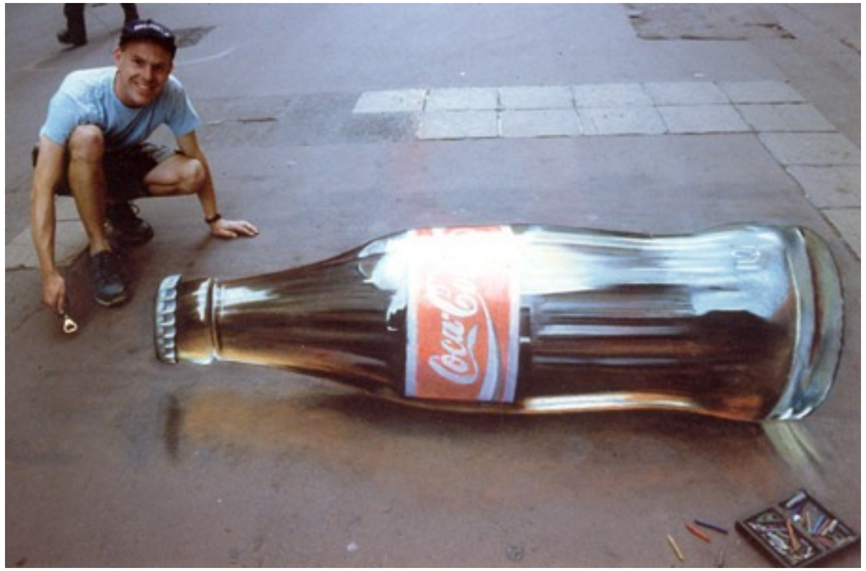
3D world



Point of observation

2D image





Geometry information is lost

Why is computer vision hard?

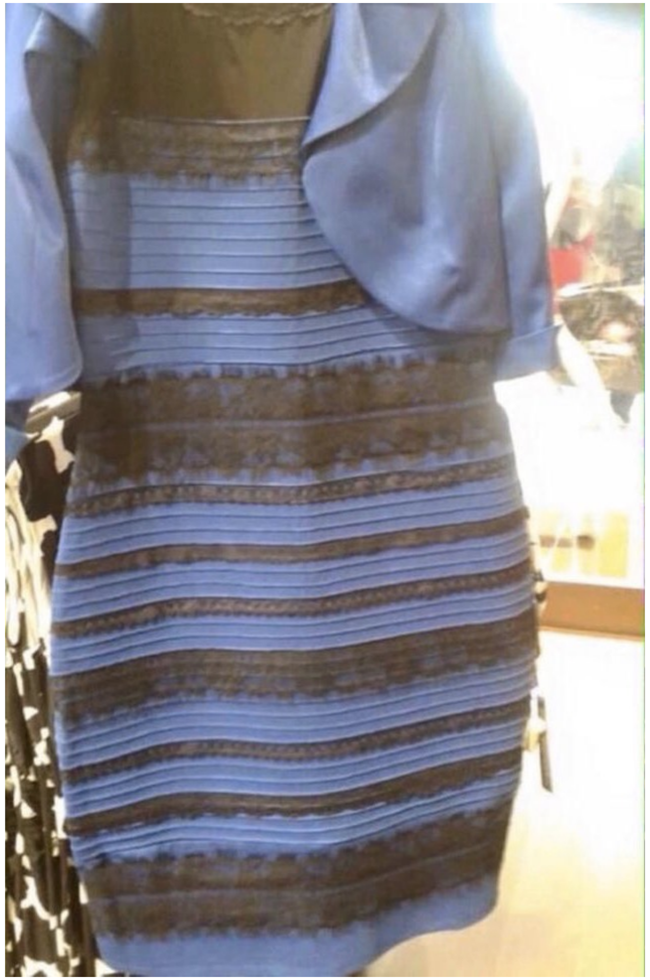
- Images are a lossy projection of the world

What color is the dress?

A) Black and blue

B) White and gold?

Appearance information is also lost

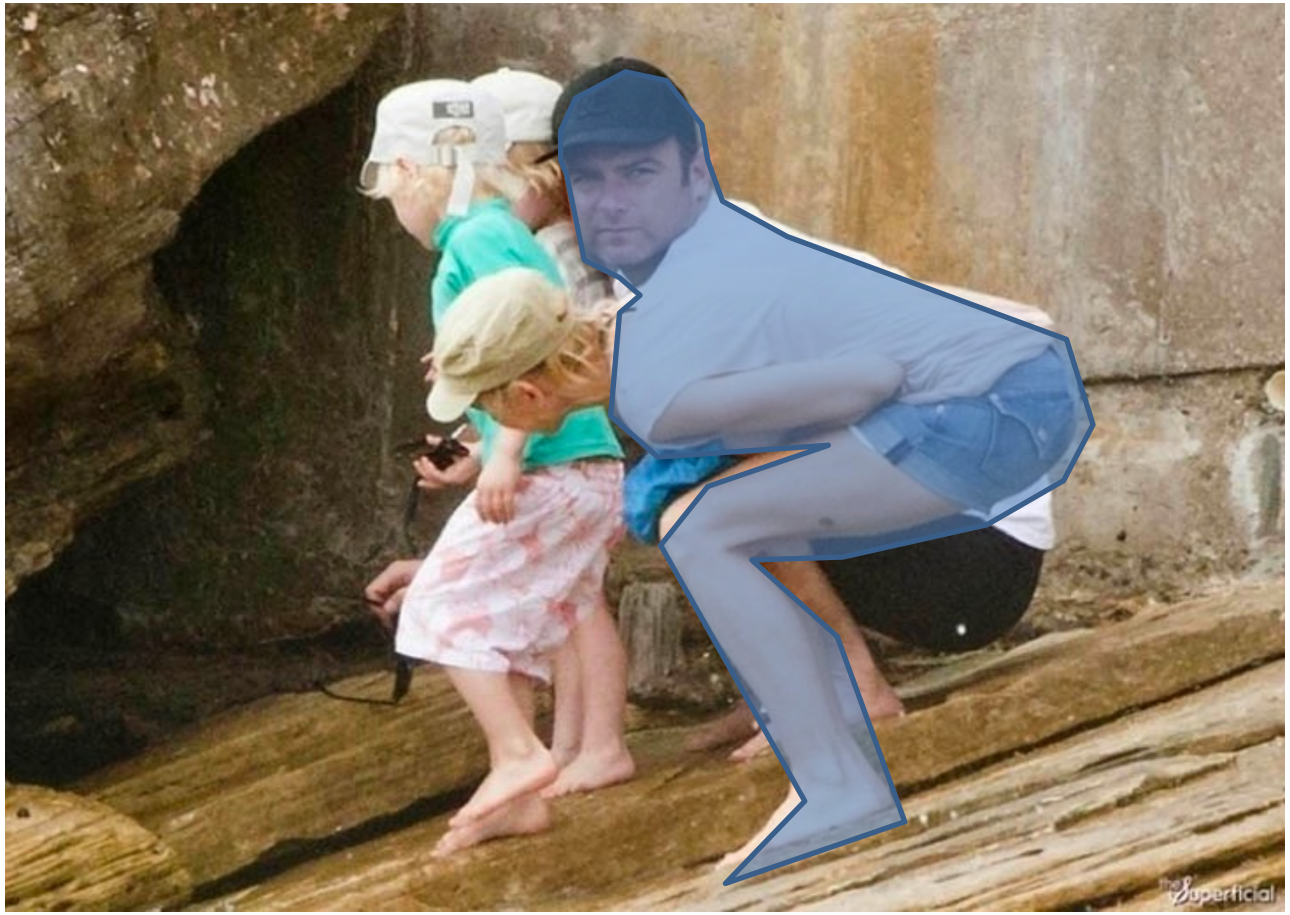


https://www.wired.com/2015/02/science-one-agrees-color-dress/

Slide by L. Lazebnik

Why is computer vision hard?

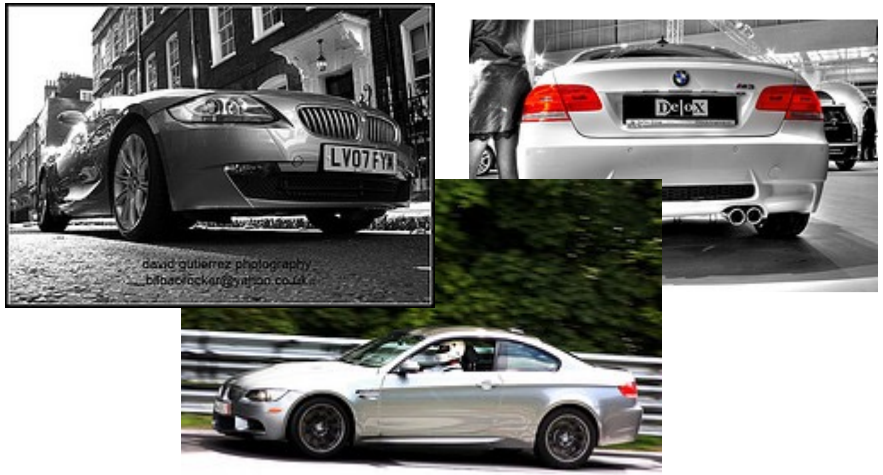
- Images are a lossy projection of the world



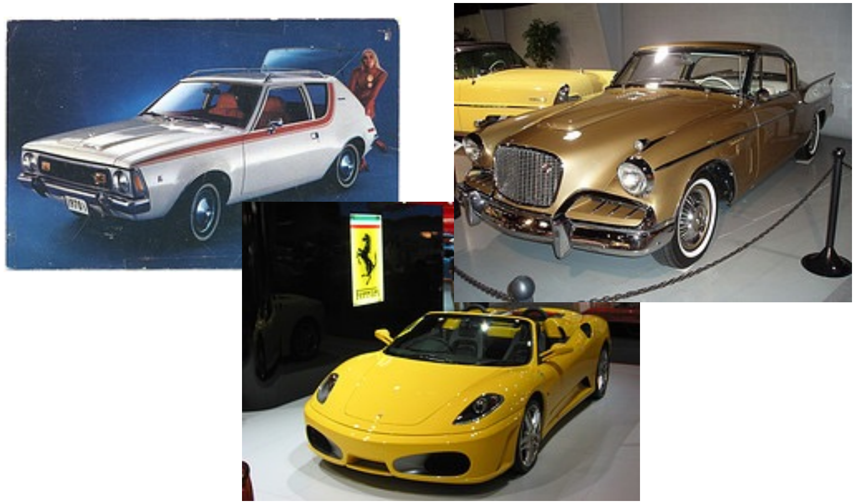
Might cause objects to blend

Why is computer vision hard?

- Images are a lossy projection of the world (geometry, appearance, … are lost)
- Visual world is diverse



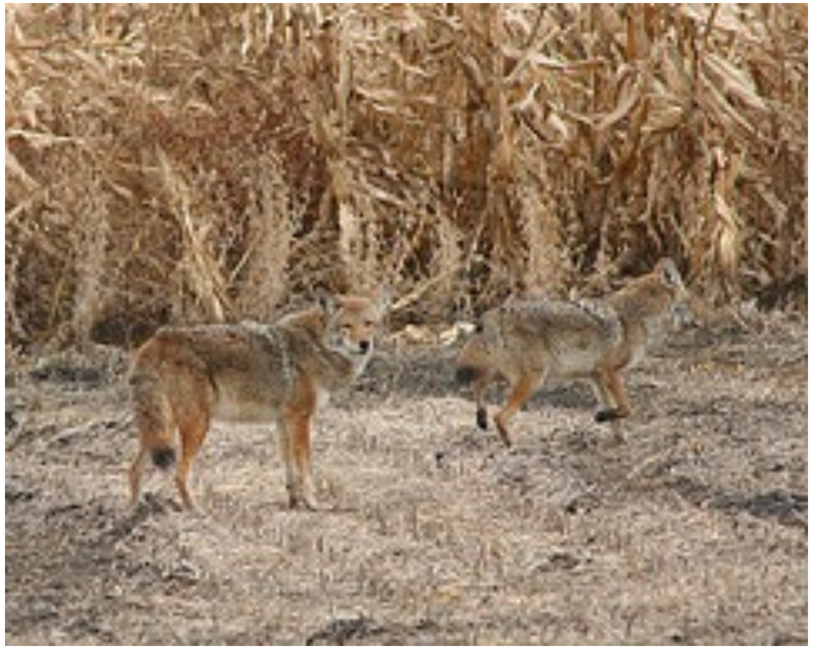
Viewpoint variation



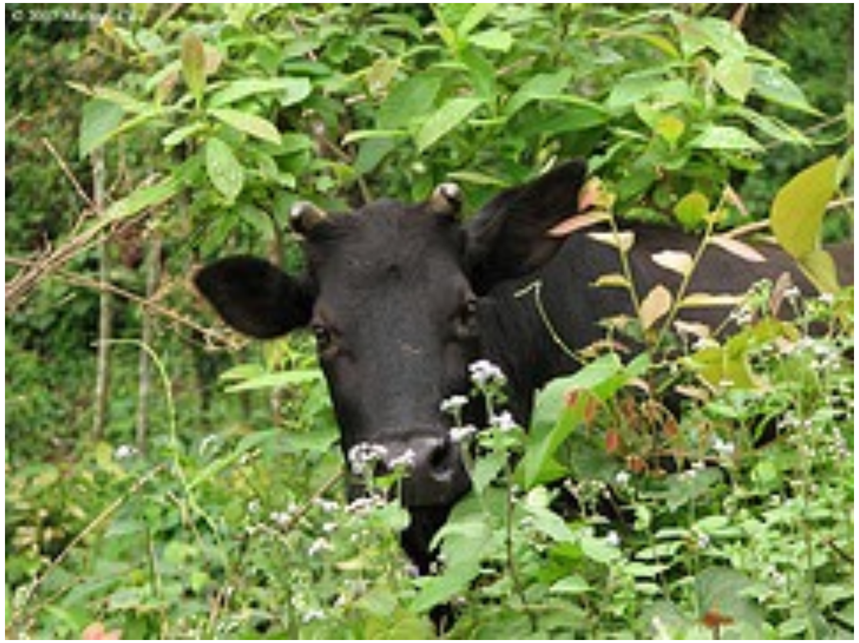
Shape variation

Why is computer vision hard?

- Images are a lossy projection of the world (geometry, appearance, … are lost)
- Visual world is diverse



Background clutter



Occlusion

Why is computer vision hard?

- Images are a lossy projection of the world (geometry, appearance, … are lost)
  - need some priors to interpret what you are seeing

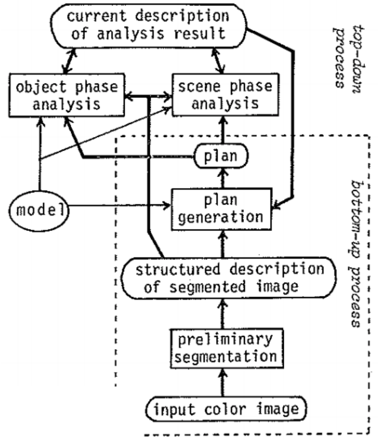
- Visual world is diverse
  - can’t write down these priors by hand

Enter machine learning

John's Diner with John's Chevelle, 2007

Why machine learning?

- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent



low of data in the
analysis process.

lock as a set of
s it must satisfy
The rules have

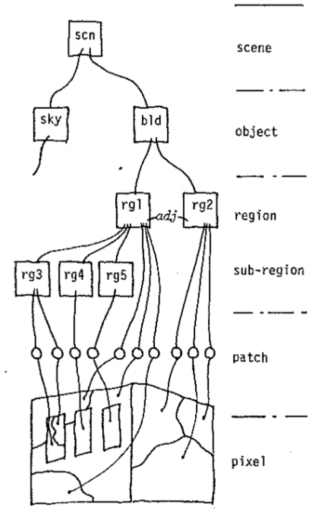


Figure 3. Structure of description built by the system.

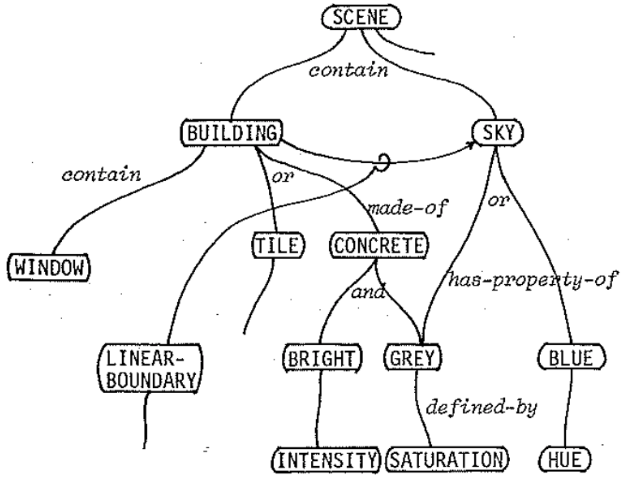


Figure 2. Semantic network for knowledge representation.

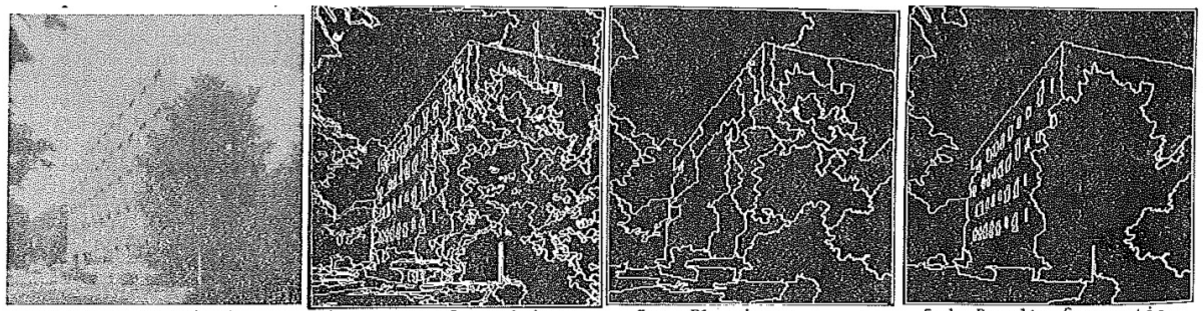


Figure 5-a. Digitized color scene.

5-b. Result of preliminary segmentation.

5-c. Plan image.

5-d. Result of semantic segmentation.

Y. Ohta, T. Kanade and T. Sakai. An Analysis System for Scenes Containing objects with Substructures. Proc. of the Fourth International Joint Conference on Pattern Recognition, pp. 752-754, 1978

Why machine learning?

- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent

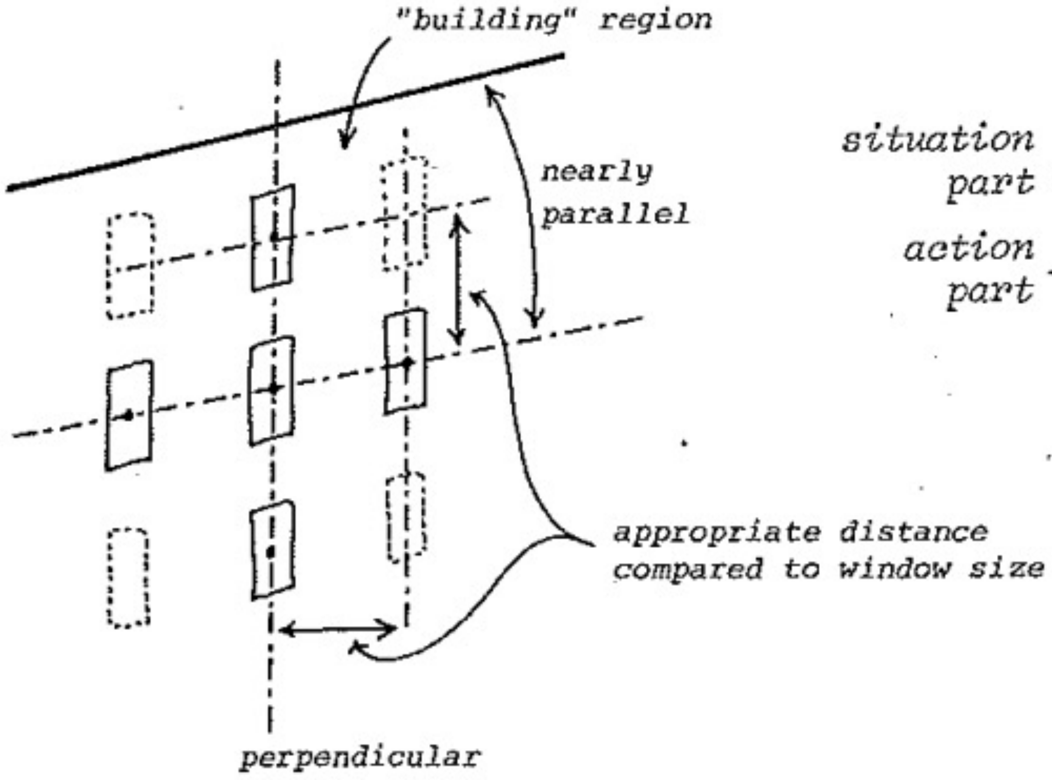
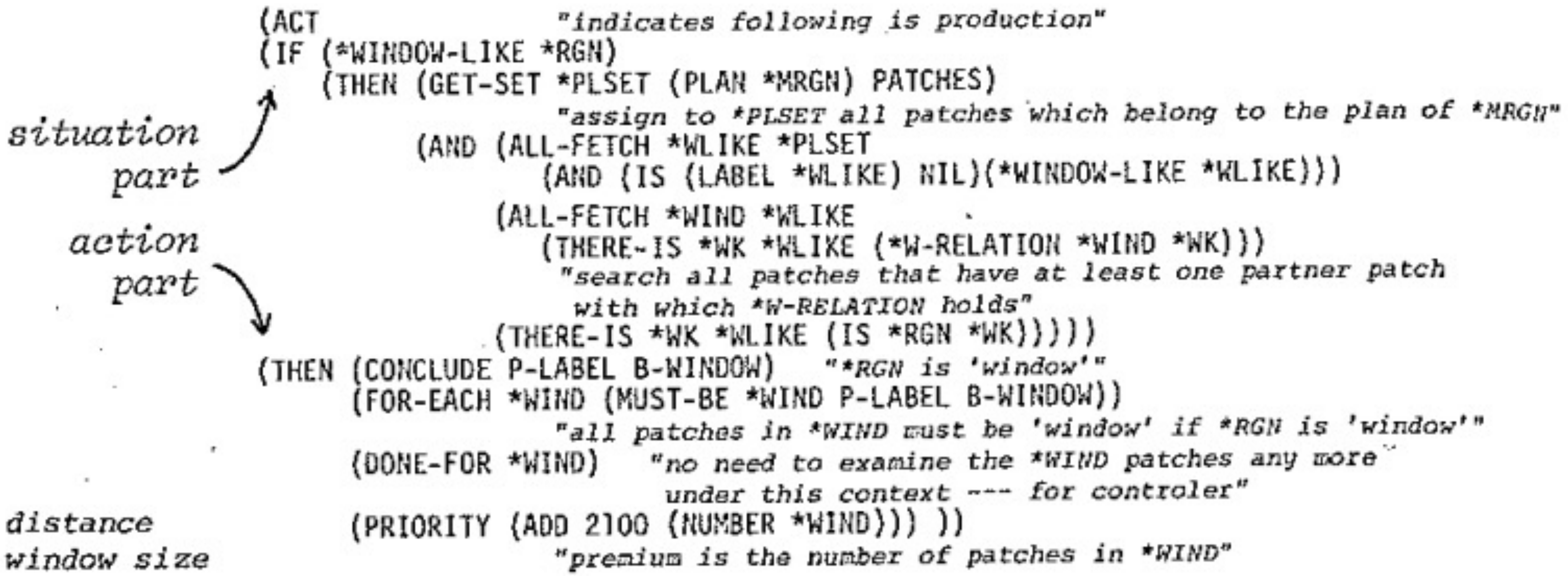


Figure 4-a. "Building" region and "windows".

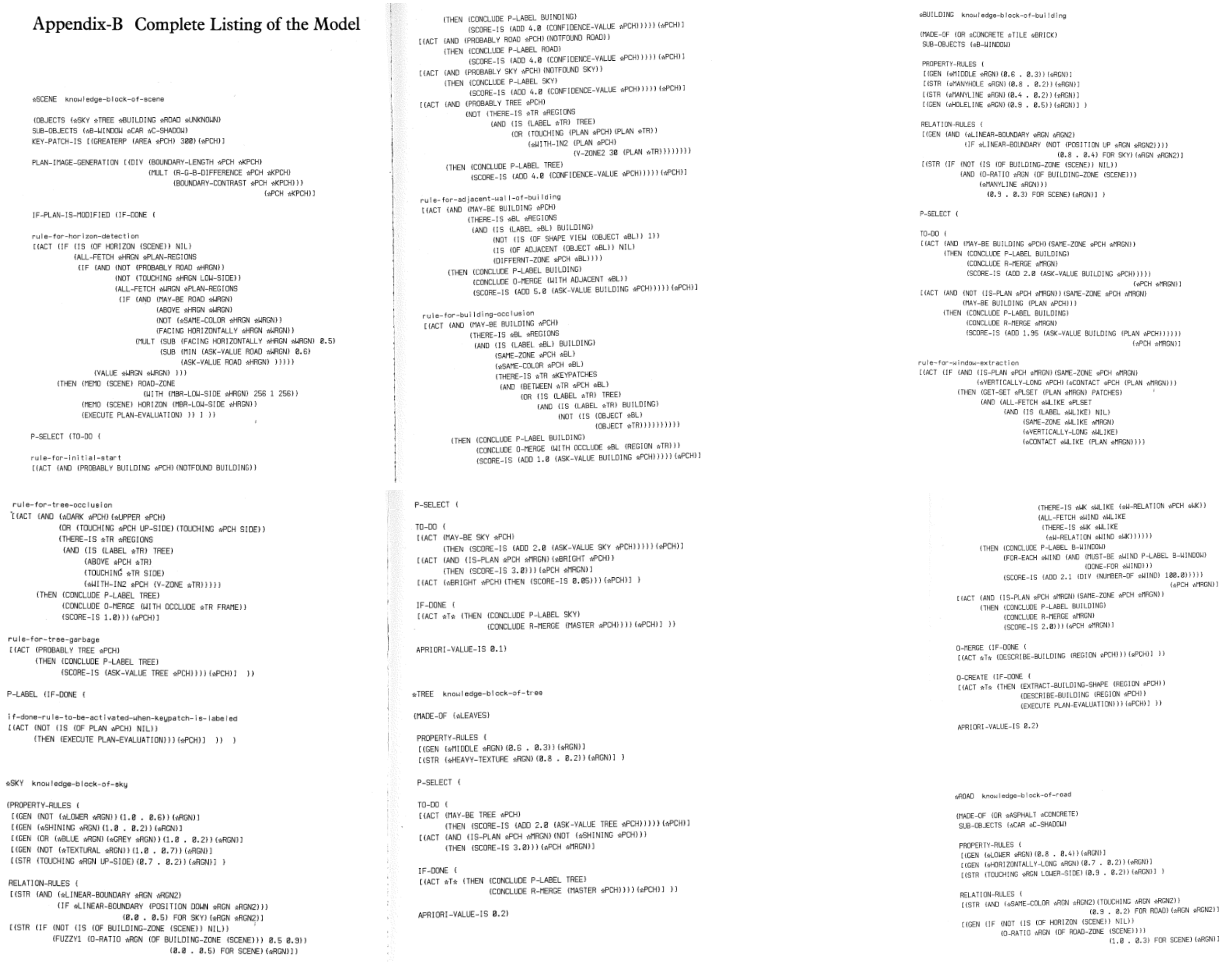

situation
part
action
part

```
(ACT                  "indicates following is production"
(IF (*WINDOW-LIKE *RGN)
    (THEN (GET-SET *PLSET (PLAN *MRGN) PATCHES)
                      "assign to *PLSET all patches which belong to the plan of *MRGN"
         (AND (ALL-FETCH *WLIKE *PLSET
                    (AND (IS (LABEL *WLIKE) NIL)(*WINDOW-LIKE *WLIKE)))
                 (ALL-FETCH *WIND *WLIKE
                    (THERE-IS *WK *WLIKE (*W-RELATION *WIND *WK)))
                     "search all patches that have at least one partner patch
                      with which *W-RELATION holds"
                 (THERE-IS *WK *WLIKE (IS *RGN *WK)))))
(THEN (CONCLUDE P-LABEL B-WINDOW)    "*RGN is 'window'"
      (FOR-EACH *WIND (MUST-BE *WIND P-LABEL B-WINDOW))
                      "all patches in *WIND must be 'window' if *RGN is 'window'"
      (DONE-FOR *WIND)   "no need to examine the *WIND patches any more
                          under this context --- for controler"
      (PRIORITY (ADD 2100 (NUMBER *WIND))) ))
                      "premium is the number of patches in *WIND"
```

Figure 4-b. The production for analyzing "windows".

Why machine learning?

- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent



Appendix-B Complete Listing of the Model

```
☆SCENE  knowledge-block-of-scene

(OBJECTS (☆SKY ☆TREE ☆BUILDING ☆ROAD ☆UNKNOWN)
SUB-OBJECTS (☆B-WINDOW ☆CAR ☆C-SHADOW)
KEY-PATCH-IS [(GREATERP (AREA ☆PCH) 300)(☆PCH)]

PLAN-IMAGE-GENERATION [(DIV (BOUNDARY-LENGTH ☆PCH ☆KPCH)
                            (MULT (R-G-B-DIFFERENCE ☆PCH ☆KPCH)
                                  (BOUNDARY-CONTRAST ☆PCH ☆KPCH)))
                                                        (☆PCH ☆KPCH)]

IF-PLAN-IS-MODIFIED (IF-DONE (

rule-for-horizon-detection
[(ACT (IF (IS (OF HORIZON (SCENE)) NIL)
          (ALL-FETCH ☆HRGN ☆PLAN-REGIONS
           (IF (AND (NOT (PROBABLY ROAD ☆HRGN))
                    (NOT (TOUCHING ☆HRGN LOW-SIDE))
                    (ALL-FETCH ☆WRGN ☆PLAN-REGIONS
                     (IF (AND (MAY-BE ROAD ☆WRGN)
                              (ABOVE ☆HRGN ☆WRGN)
                              (NOT (☆SAME-COLOR ☆HRGN ☆WRGN))
                              (FACING HORIZONTALLY ☆HRGN ☆WRGN))
                         (MULT (SUB (FACING HORIZONTALLY ☆HRGN ☆WRGN) 0.5)
                               (SUB (MIN (ASK-VALUE ROAD ☆WRGN) 0.6)
                                    (ASK-VALUE ROAD ☆HRGN) )))))
                (VALUE ☆WRGN ☆WRGN) )))
        (THEN (MEMO (SCENE) ROAD-ZONE
                        (WITH (MBR-LOW-SIDE ☆HRGN) 256 1 256))
              (MEMO (SCENE) HORIZON (MBR-LOW-SIDE ☆HRGN))
              (EXECUTE PLAN-EVALUATION) )) ] ))

P-SELECT (TO-DO (

rule-for-initial-start
[(ACT (AND (PROBABLY BUILDING ☆PCH)(NOTFOUND BUILDING))
```

```
      (THEN (CONCLUDE P-LABEL BUINDING)
            (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE ☆PCH)))))(☆PCH)]
[(ACT (AND (PROBABLY ROAD ☆PCH)(NOTFOUND ROAD))
      (THEN (CONCLUDE P-LABEL ROAD)
            (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE ☆PCH)))))(☆PCH)]
[(ACT (AND (PROBABLY SKY ☆PCH)(NOTFOUND SKY))
      (THEN (CONCLUDE P-LABEL SKY)
            (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE ☆PCH)))))(☆PCH)]
[(ACT (AND (PROBABLY TREE ☆PCH)
           (NOT (THERE-IS ☆TR ☆REGIONS
                 (AND (IS (LABEL ☆TR) TREE)
                      (OR (TOUCHING (PLAN ☆PCH)(PLAN ☆TR))
                          (☆WITH-IN2 (PLAN ☆PCH)
                                     (V-ZONE2 30 (PLAN ☆TR))))))))
      (THEN (CONCLUDE P-LABEL TREE)
            (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE ☆PCH)))))(☆PCH)]

rule-for-adjacent-wall-of-building
[(ACT (AND (MAY-BE BUILDING ☆PCH)
           (THERE-IS ☆BL ☆REGIONS
            (AND (IS (LABEL ☆BL) BUILDING)
                 (NOT (IS (OF SHAPE VIEW (OBJECT ☆BL)) 1))
                 (IS (OF ADJACENT (OBJECT ☆BL)) NIL)
                 (DIFFERNT-ZONE ☆PCH ☆BL))))
      (THEN (CONCLUDE P-LABEL BUILDING)
            (CONCLUDE O-MERGE (WITH ADJACENT ☆BL))
            (SCORE-IS (ADD 5.0 (ASK-VALUE BUILDING ☆PCH)))))(☆PCH)]

rule-for-building-occlusion
[(ACT (AND (MAY-BE BUILDING ☆PCH)
           (THERE-IS ☆BL ☆REGIONS
            (AND (IS (LABEL ☆BL) BUILDING)
                 (SAME-ZONE ☆PCH ☆BL)
                 (☆SAME-COLOR ☆PCH ☆BL)
                 (THERE-IS ☆TR ☆KEYPATCHES
                  (AND (BETWEEN ☆TR ☆PCH ☆BL)
                       (OR (IS (LABEL ☆TR) TREE)
                           (AND (IS (LABEL ☆TR) BUILDING)
                                (NOT (IS (OBJECT ☆BL)
                                         (OBJECT ☆TR))))))))))
      (THEN (CONCLUDE P-LABEL BUILDING)
            (CONCLUDE O-MERGE (WITH OCCLUDE ☆BL (REGION ☆TR)))
            (SCORE-IS (ADD 1.0 (ASK-VALUE BUILDING ☆PCH)))))(☆PCH)]
```

```
rule-for-tree-occlusion
[(ACT (AND (☆DARK ☆PCH)(☆UPPER ☆PCH)
           (OR (TOUCHING ☆PCH UP-SIDE)(TOUCHING ☆PCH SIDE))
           (THERE-IS ☆TR ☆REGIONS
            (AND (IS (LABEL ☆TR) TREE)
                 (ABOVE ☆PCH ☆TR)
                 (TOUCHING ☆TR SIDE)
                 (☆WITH-IN2 ☆PCH (V-ZONE ☆TR)))))
      (THEN (CONCLUDE P-LABEL TREE)
            (CONCLUDE O-MERGE (WITH OCCLUDE ☆TR FRAME))
            (SCORE-IS 1.0)))(☆PCH)]

rule-for-tree-garbage
[(ACT (PROBABLY TREE ☆PCH)
      (THEN (CONCLUDE P-LABEL TREE)
            (SCORE-IS (ASK-VALUE TREE ☆PCH))))(☆PCH)]  ))

P-LABEL (IF-DONE (

if-done-rule-to-be-activated-when-keypatch-is-labeled
[(ACT (NOT (IS (OF PLAN ☆PCH) NIL))
      (THEN (EXECUTE PLAN-EVALUATION)))(☆PCH)]  ))  )


☆SKY  knowledge-block-of-sky

(PROPERTY-RULES (
 [(GEN (NOT (☆LOWER ☆RGN))(1.0 . 0.6))(☆RGN)]
 [(GEN (☆SHINING ☆RGN)(1.0 . 0.2))(☆RGN)]
 [(GEN (OR (☆BLUE ☆RGN)(☆GREY ☆RGN))(1.0 . 0.2))(☆RGN)]
 [(GEN (NOT (☆TEXTURAL ☆RGN))(1.0 . 0.7))(☆RGN)]
 [(STR (TOUCHING ☆RGN UP-SIDE)(0.7 . 0.2))(☆RGN)] )

RELATION-RULES (
[(STR (AND (☆LINEAR-BOUNDARY ☆RGN ☆RGN2)
           (IF ☆LINEAR-BOUNDARY (POSITION DOWN ☆RGN ☆RGN2)))
                           (0.0 . 0.5) FOR SKY)(☆RGN ☆RGN2)]
[(STR (IF (NOT (IS (OF BUILDING-ZONE (SCENE)) NIL))
          (FUZZY1 (O-RATIO ☆RGN (OF BUILDING-ZONE (SCENE))) 0.5 0.9))
                                     (0.0 . 0.5) FOR SCENE)(☆RGN)])
```

```
P-SELECT (

TO-DO (
[(ACT (MAY-BE SKY ☆PCH)
      (THEN (SCORE-IS (ADD 2.0 (ASK-VALUE SKY ☆PCH)))))(☆PCH)]
[(ACT (AND (IS-PLAN ☆PCH ☆MRGN)(☆BRIGHT ☆PCH))
      (THEN (SCORE-IS 3.0)))(☆PCH ☆MRGN)]
[(ACT (☆BRIGHT ☆PCH)(THEN (SCORE-IS 0.05)))(☆PCH)] )

IF-DONE (
[(ACT ☆T☆ (THEN (CONCLUDE P-LABEL SKY)
                (CONCLUDE R-MERGE (MASTER ☆PCH))))(☆PCH)] ))

APRIORI-VALUE-IS 0.1)



☆TREE  knowledge-block-of-tree

(MADE-OF (☆LEAVES)

 PROPERTY-RULES (
 [(GEN (☆MIDDLE ☆RGN)(0.6 . 0.3))(☆RGN)]
 [(STR (☆HEAVY-TEXTURE ☆RGN)(0.8 . 0.2))(☆RGN)] )

 P-SELECT (

 TO-DO (
 [(ACT (MAY-BE TREE ☆PCH)
       (THEN (SCORE-IS (ADD 2.0 (ASK-VALUE TREE ☆PCH)))))(☆PCH)]
 [(ACT (AND (IS-PLAN ☆PCH ☆MRGN)(NOT (☆SHINING ☆PCH)))
       (THEN (SCORE-IS 3.0)))(☆PCH ☆MRGN)]

 IF-DONE (
 [(ACT ☆T☆ (THEN (CONCLUDE P-LABEL TREE)
                 (CONCLUDE R-MERGE (MASTER ☆PCH))))(☆PCH)] ))

 APRIORI-VALUE-IS 0.2)
```

```
☆BUILDING  knowledge-block-of-building

(MADE-OF (OR ☆CONCRETE ☆TILE ☆BRICK)
 SUB-OBJECTS (☆B-WINDOW)

 PROPERTY-RULES (
 [(GEN (☆MIDDLE ☆RGN)(0.6 . 0.3))(☆RGN)]
 [(STR (☆MANYHOLE ☆RGN)(0.8 . 0.2))(☆RGN)]
 [(STR (☆MANYLINE ☆RGN)(0.4 . 0.2))(☆RGN)]
 [(GEN (☆HOLELINE ☆RGN)(0.9 . 0.5))(☆RGN)] )

 RELATION-RULES (
 [(GEN (AND (☆LINEAR-BOUNDARY ☆RGN ☆RGN2)
            (IF ☆LINEAR-BOUNDARY (NOT (POSITION UP ☆RGN ☆RGN2))))
                                  (0.8 . 0.4) FOR SKY)(☆RGN ☆RGN2)]
 [(STR (IF (NOT (IS (OF BUILDING-ZONE (SCENE)) NIL))
           (AND (O-RATIO ☆RGN (OF BUILDING-ZONE (SCENE)))
                (☆MANYLINE ☆RGN)))
                 (0.9 . 0.3) FOR SCENE)(☆RGN)] )

P-SELECT (

TO-DO (
[(ACT (AND (MAY-BE BUILDING ☆PCH)(SAME-ZONE ☆PCH ☆MRGN))
      (THEN (CONCLUDE P-LABEL BUILDING)
            (CONCLUDE R-MERGE ☆MRGN)
            (SCORE-IS (ADD 2.0 (ASK-VALUE BUILDING ☆PCH)))))
                                                  (☆PCH ☆MRGN)]
[(ACT (AND (NOT (IS-PLAN ☆PCH ☆MRGN))(SAME-ZONE ☆PCH ☆MRGN)
           (MAY-BE BUILDING (PLAN ☆PCH)))
      (THEN (CONCLUDE P-LABEL BUILDING)
            (CONCLUDE R-MERGE ☆MRGN)
            (SCORE-IS (ADD 1.95 (ASK-VALUE BUILDING (PLAN ☆PCH))))))
                                                  (☆PCH ☆MRGN)]

rule-for-window-extraction
[(ACT (IF (AND (IS-PLAN ☆PCH ☆MRGN)(SAME-ZONE ☆PCH ☆MRGN)
               (☆VERTICALLY-LONG ☆PCH)(☆CONTACT ☆PCH (PLAN ☆MRGN)))
          (THEN (GET-SET ☆PLSET (PLAN ☆MRGN) PATCHES)
                (AND (ALL-FETCH ☆WLIKE ☆PLSET
                      (AND (IS (LABEL ☆WLIKE) NIL)
                           (SAME-ZONE ☆WLIKE ☆MRGN)
                           (☆VERTICALLY-LONG ☆WLIKE)
                           (☆CONTACT ☆WLIKE (PLAN ☆MRGN))))
```

```
                      (THERE-IS ☆WK ☆WLIKE (☆W-RELATION ☆PCH ☆WK))
                      (ALL-FETCH ☆WIND ☆WLIKE
                       (THERE-IS ☆WK ☆WLIKE
                        (☆W-RELATION ☆WIND ☆WK))))))
      (THEN (CONCLUDE P-LABEL B-WINDOW)
            (FOR-EACH ☆WIND (AND (MUST-BE ☆WIND P-LABEL B-WINDOW)
                                 (DONE-FOR ☆WIND)))
            (SCORE-IS (ADD 2.1 (DIV (NUMBER-OF ☆WIND) 100.0)))))
                                                     (☆PCH ☆MRGN)]
[(ACT (AND (IS-PLAN ☆PCH ☆MRGN)(SAME-ZONE ☆PCH ☆MRGN))
      (THEN (CONCLUDE P-LABEL BUILDING)
            (CONCLUDE R-MERGE ☆MRGN)
            (SCORE-IS 2.0)))(☆PCH ☆MRGN)]

O-MERGE (IF-DONE (
[(ACT ☆T☆ (DESCRIBE-BUILDING (REGION ☆PCH)))(☆PCH)] ))

O-CREATE (IF-DONE (
[(ACT ☆T☆ (THEN (EXTRACT-BUILDING-SHAPE (REGION ☆PCH))
                (DESCRIBE-BUILDING (REGION ☆PCH))
                (EXECUTE PLAN-EVALUATION)))(☆PCH)] ))

APRIORI-VALUE-IS 0.2)



☆ROAD  knowledge-block-of-road

(MADE-OF (OR ☆ASPHALT ☆CONCRETE)
 SUB-OBJECTS (☆CAR ☆C-SHADOW)

 PROPERTY-RULES (
 [(GEN (☆LOWER ☆RGN)(0.8 . 0.4))(☆RGN)]
 [(GEN (☆HORIZONTALLY-LONG ☆RGN)(0.7 . 0.2))(☆RGN)]
 [(STR (TOUCHING ☆RGN LOWER-SIDE)(0.9 . 0.2))(☆RGN)] )

 RELATION-RULES (
 [(STR (AND (☆SAME-COLOR ☆RGN ☆RGN2)(TOUCHING ☆RGN ☆RGN2))
                                 (0.9 . 0.2) FOR ROAD)(☆RGN ☆RGN2)]
 [(GEN (IF (NOT (IS (OF HORIZON (SCENE)) NIL))
           (O-RATIO ☆RGN (OF ROAD-ZONE (SCENE))))
                                      (1.0 . 0.3) FOR SCENE)(☆RGN)]
```

Why machine learning?

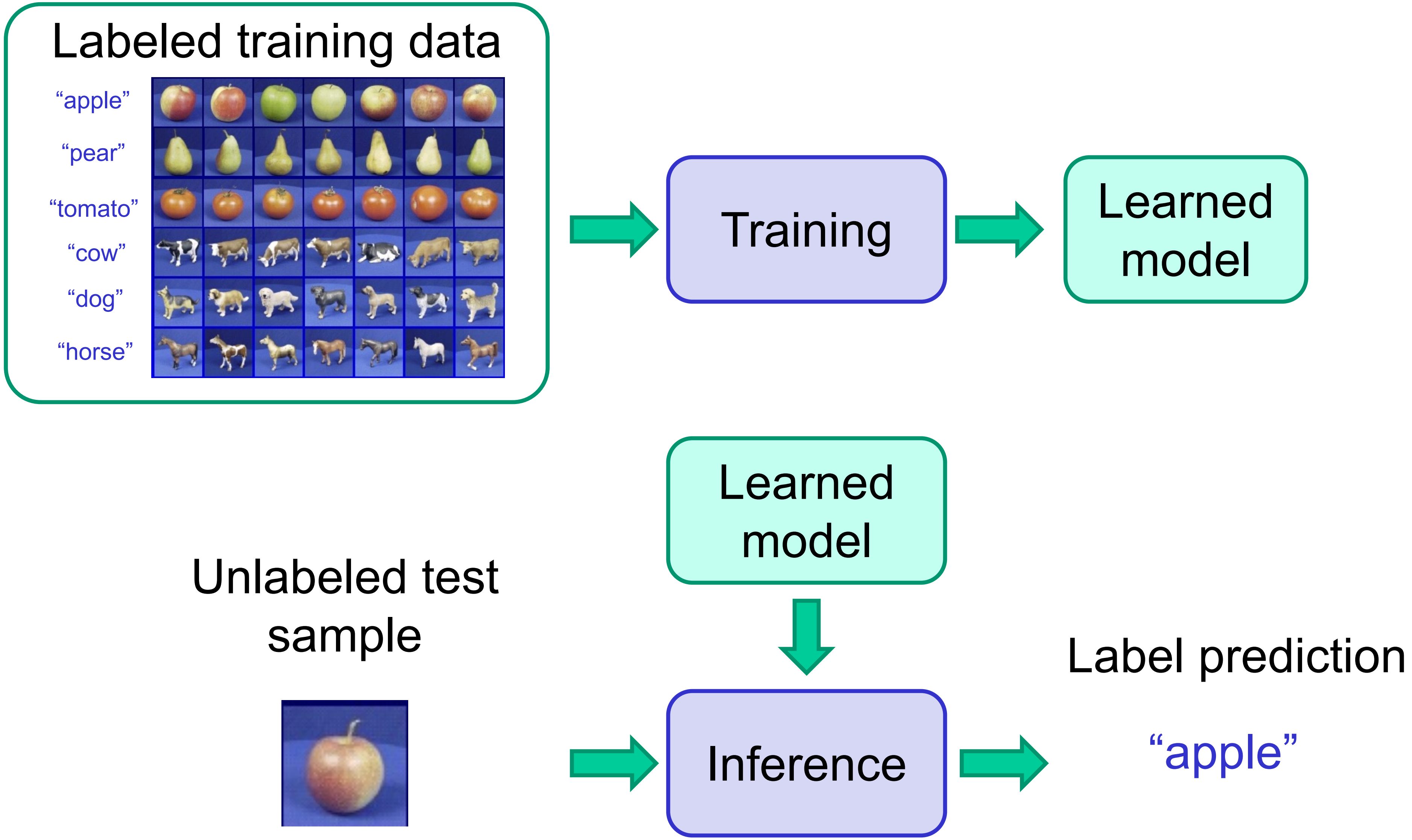
- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent
  - Never worked (in general)

Why machine learning?

- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent

- Modern answer: Program into the agent the *ability to improve performance based on experience*
  - Experience should come from *training data* or *demonstrations*
  - We want to optimize the performance of the agent on the training data, with the hope that it will *generalize* to unseen inputs
  - This is the *statistical learning* viewpoint

The basic ML framework (for supervised learning)



Training time

Labeled training data

"apple"

"pear"

"tomato"

"cow"

"dog"

"horse"

Training

Learned model

Test time

Unlabeled test sample

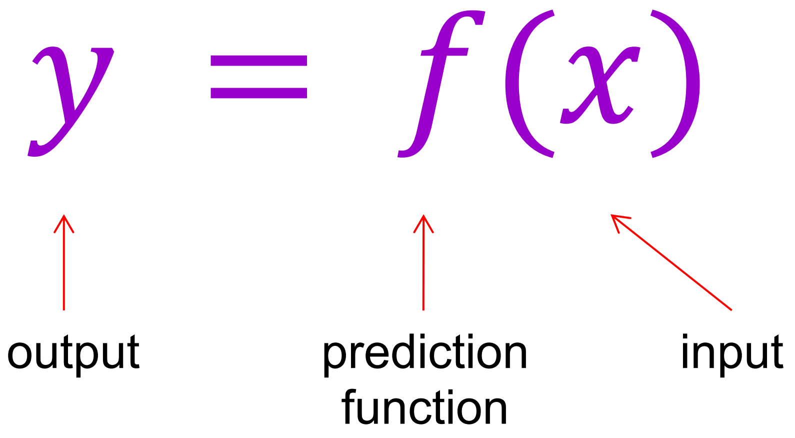
Learned model

Inference

Label prediction

"apple"

The basic ML framework (for supervised learning)



$$y = f(x)$$

output

prediction function

input

- **Training** (or **learning**): given a *training set* of labeled examples $\{(x_1, y_1), \dots, (x_N, y_N)\}$, instantiate a predictor f
- **Testing** (or **inference**): apply f to a new *test example* x and output the predicted value $y = f(x)$

- Rather than hand-defining how 2D projections of apples are different from pears, f will learn this from the data.

Deep Learning

- A general way to model function f as composition (layers) of simple functions, very loosely inspired by the brain.

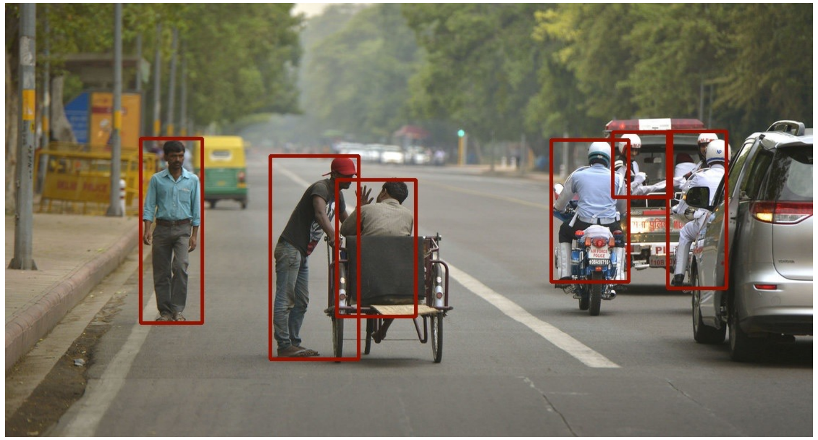
Lecture overview

- Different recognition problems in computer vision
- Supervised classification
- Taxonomy of learning problems

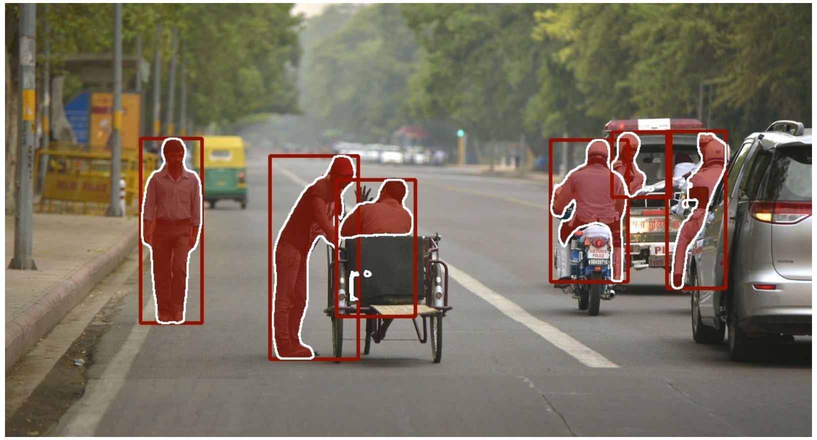
Different Recognition Problems

This image by Nikita is licensed under CC-BY 2.0

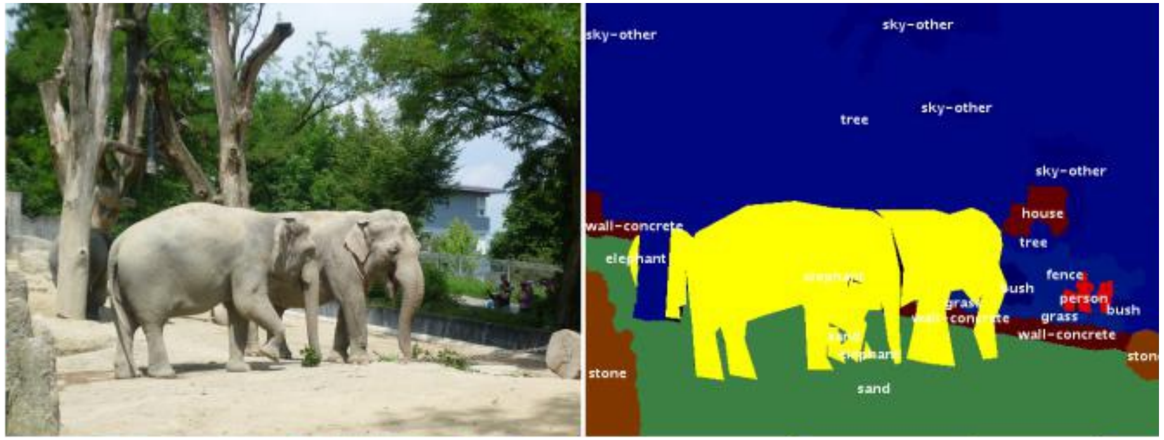
Classification: Assign image to one of a fixed set of categories



Object Detection: Put a bounding box around each instance of a class



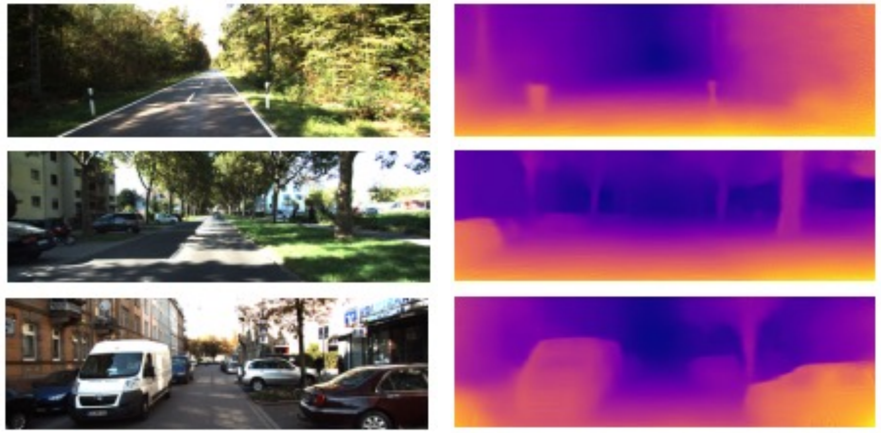
Instance Segmentation: Mark pixels for each instance of a class



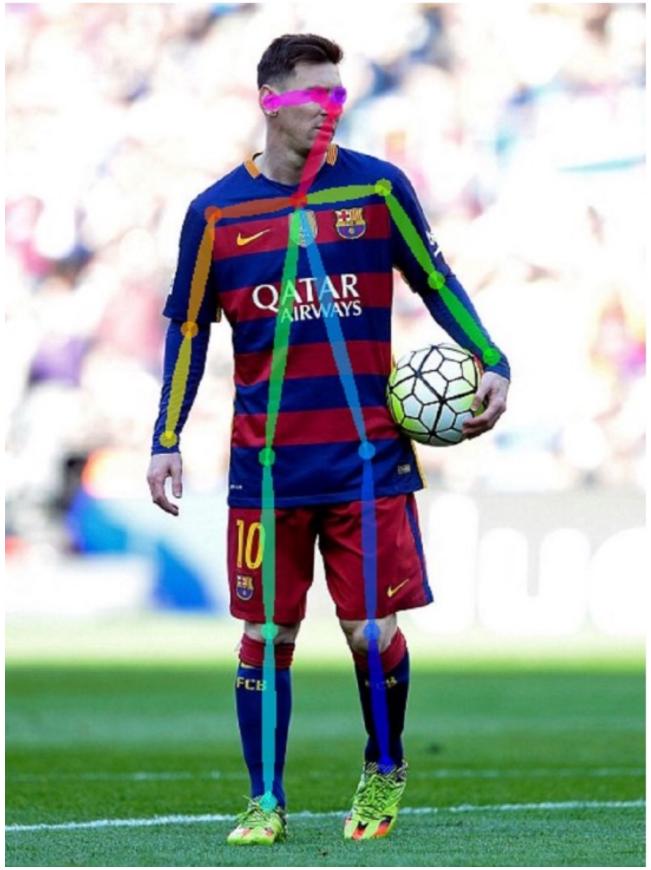
Semantic Segmentation: Label each pixels with its category

Different Recognition Problems

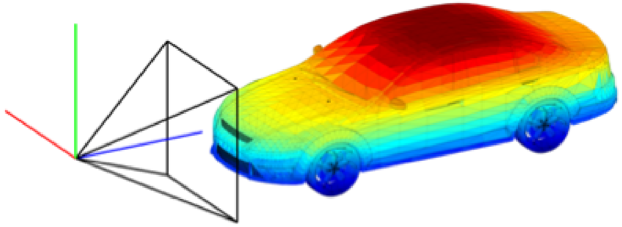
Image Captioning: Man riding a horse on a beach



Depth Prediction: how far is each pixel in the image

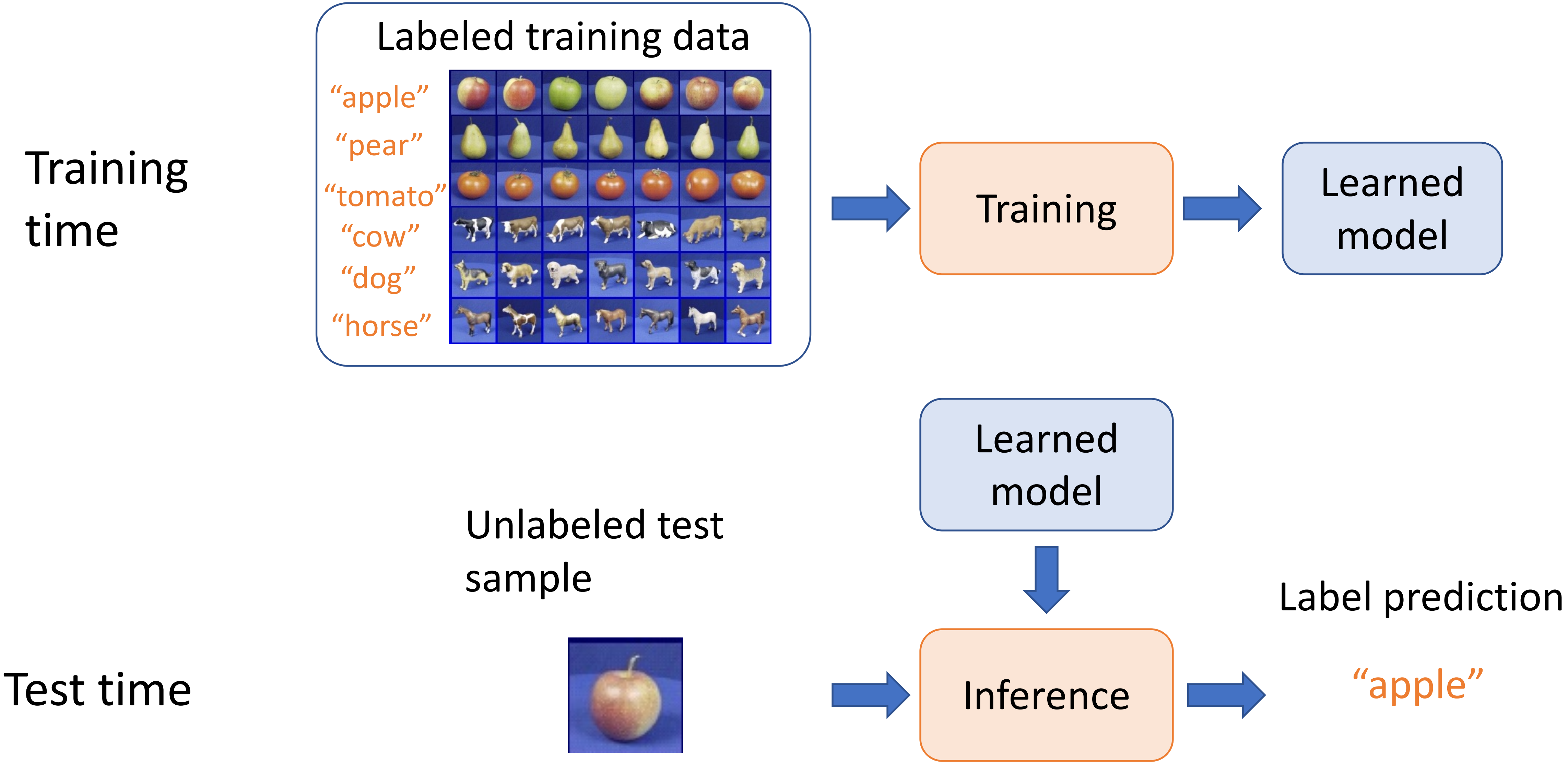


Keypoint prediction



Pose Prediction: Rotation that aligns object to a canonical pose

The basic ML framework (for supervised learning)



Training time

Labeled training data

“apple”

“pear”

“tomato”

“cow”

“dog”

“horse”

Training

Learned model

Test time

Unlabeled test sample

Learned model

Inference

Label prediction

“apple”

The basic ML framework (for supervised learning)

$$y = f(x)$$

output

prediction function

input

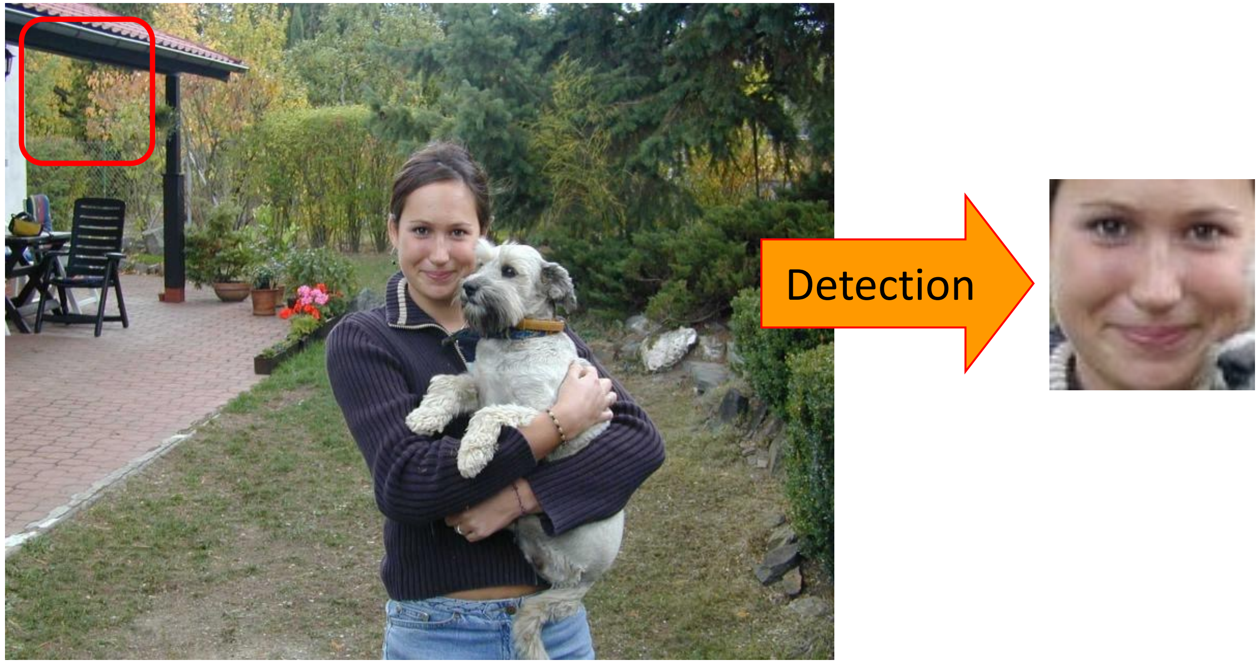
- **Training** (or **learning**): given a *training set* of labeled examples $\{(x_1, y_1), \dots, (x_N, y_N)\}$, instantiate a predictor f

- **Testing** (or **inference**): apply f to a new *test example* x and output the predicted value $y = f(x)$

- Rather than hand-defining how 2D projections of apples are different from pears, f will learn this from the data.

Is an image classifier all you need?

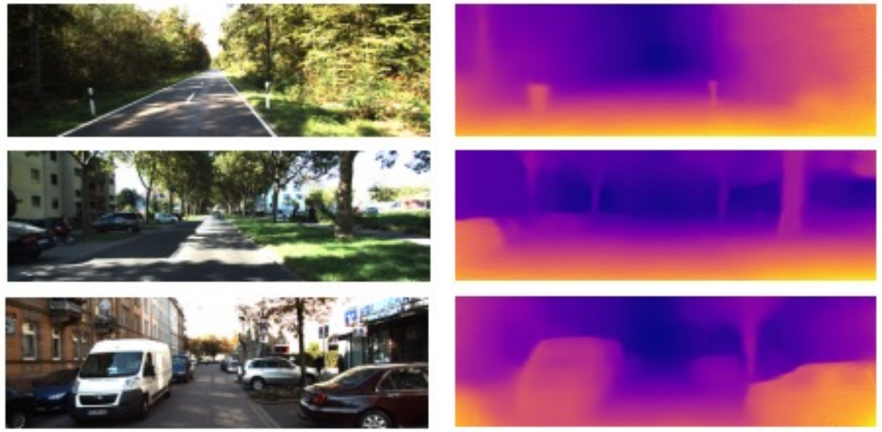
- Image Classification
- Object Detection
- Instance Segmentation
- Semantic Segmentation
- Image Captioning
- Depth Prediction
- Keypoint Prediction
- Pose Prediction
- …



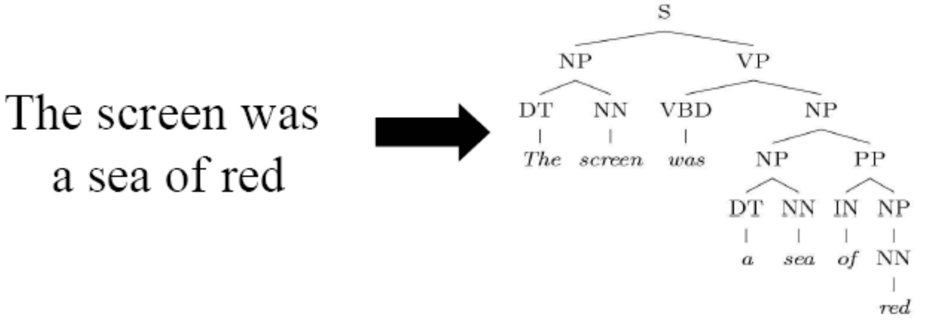
Detection

Taxonomy of learning problems

- **Type of output**
  - Classification
  - Regression
    - $y = f(x)$. y is an arbitrary scalar and not a class label.
  - Structured prediction
    - $y = f(x)$. y is a structured object.



Depth Prediction: how far is each pixel in the image



Sentence

Parse tree

Several computer vision problems have structure in the output space, but often solving a classification problem with some simple post-processing (or even without) ends up being sufficient.