Image Processing & Vision

Lecture 09: Object Detection

Hak Gu Kim

hakgukim@cau.ac.kr

Immersive Reality & Intelligent Systems Lab (IRIS LAB)

Graduate School of Advanced Imaging Science, Multimedia & Film (GSAIM)

Chung-Ang University (CAU)

15 May 2023

Topics

- Object Detection

***Note:** Many of these slides in this course were adapted from Convolutional Neural Networks for Visual Recognition (Stanford Univ.) and Deep Learning for Computer Vision (Univ. of Michigan)

Object Detection

- We assumed the image contained a single, central object, and so on

- The task of **object detection** is to detect and localize all instances of a target object class in an image

— Localization typically means putting a tight bounding box around the object

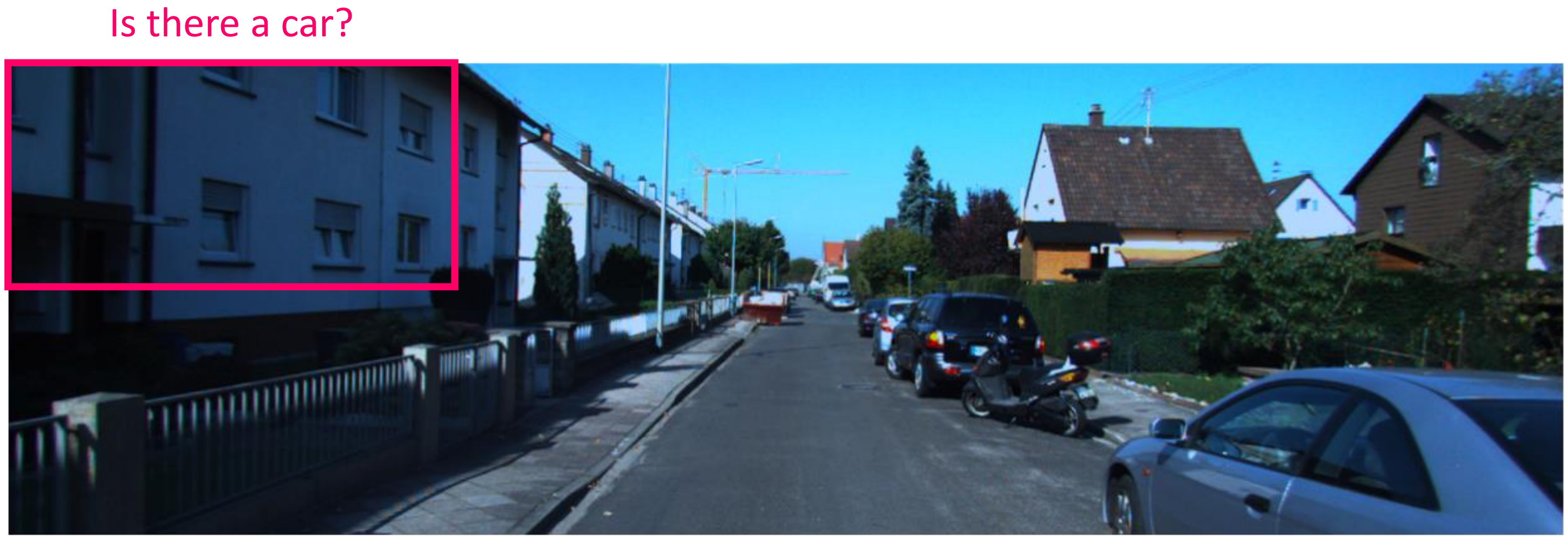
An example on KITTI dataset benchmark

Object Detection: **Sliding Window**

- **Slide** a fixed-sized detection window across the image and evaluate the classifier on each window

— We have to search over **scale** as well

— We may also have to search to search over **aspect ratios**



An example on KITTI dataset benchmark

Object Detection: **Sliding Window**

- **Slide** a fixed-sized detection window across the image and evaluate the classifier on each window

— We have to search over **scale** as well

— We may also have to search to search over **aspect ratios**

Is there a car?

An example on KITTI dataset benchmark

Object Detection: **Sliding Window**

- **Slide** a fixed-sized detection window across the image and evaluate the classifier on each window

— We have to search over **scale** as well

— We may also have to search to search over **aspect ratios**

Is there a car?

An example on KITTI dataset benchmark

Object Detection: **Sliding Window**

- **Slide** a fixed-sized detection window across the image and evaluate the classifier on each window

— We have to search over **scale** as well

— We may also have to search to search over **aspect ratios**

Is there a car?

An example on KITTI dataset benchmark

Face Detection: **Viola-Jones**

- The **Viola-Jones face detector** is a classic sliding window detector that learns both efficient features and a classifier

① Haar Feature Selection

② Creating an Integral Image

③ Adaboost Training

④ Cascading Classifiers

P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ① **Haar Feature**

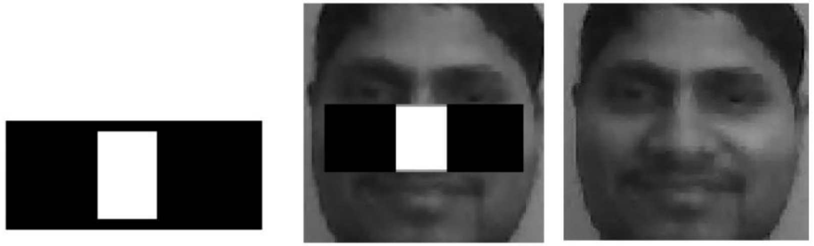
- A **rectangular feature** is computed by summing up pixel values within rectangular regions and then differencing those region sums

- **All human faces** share **similar** properties. These regularities may be matched using Haar features

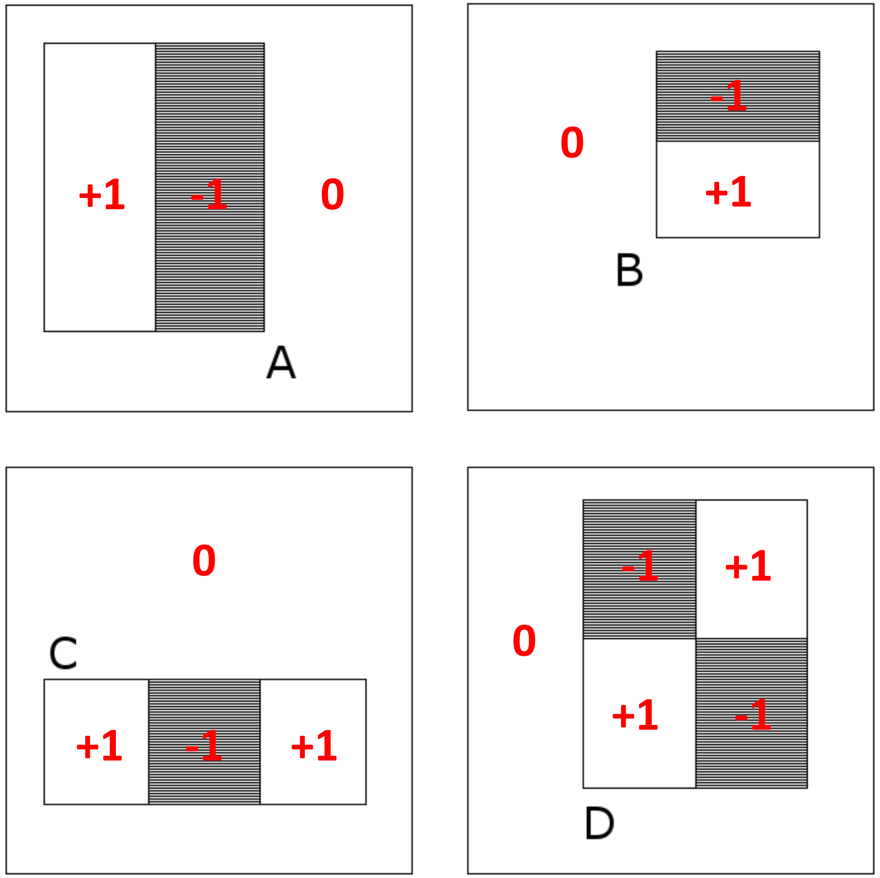
— The eye region is darker than the upper-cheeks

— The nose bridge region is brighter than the eyes

Haar Feature that looks similar to the eye region which is darker than the upper cheeks is applied onto a face



Haar Feature that looks similar to the bridge of the nose is applied onto the face



P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ② **Integral Image**

- Given an integral image, the sum within a rectangular region can be computed with just 3 additions/subtractions

— Does not depend on the size of the region

1.

31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

15+16+14+28
+27+11 = **111**

Original image

2.

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

450-254-186
+101 = **111**

Integral image

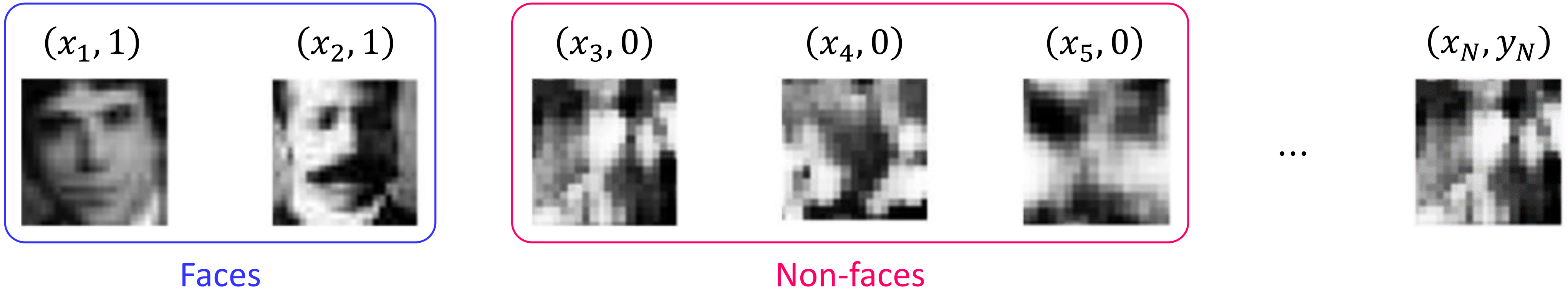
P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ③ **AdaBoost**

- Object detection framework employs AdaBoost to both select the best features and to train classifiers that use them

— **AdaBoost:** It constructs a strong classifier as a linear combination of weighted simple weak classifiers

Training dataset

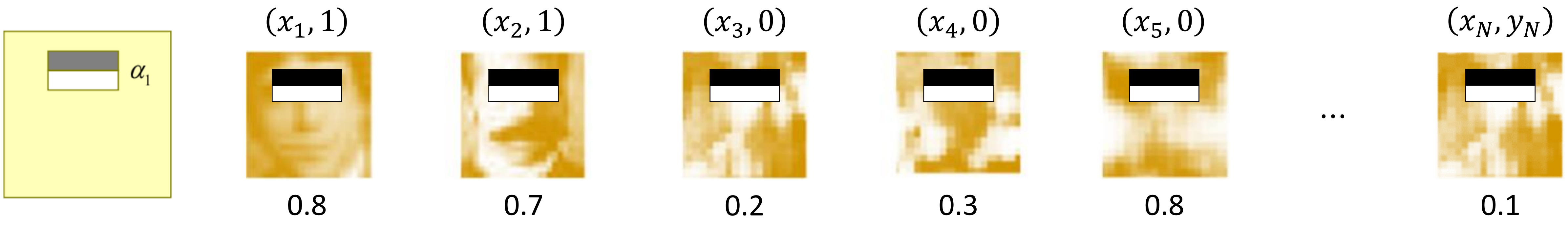


P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ③ **AdaBoost**

- Object detection framework employs AdaBoost to both select the best features and to train classifiers that use them

— **AdaBoost:** It constructs a strong classifier as a linear combination of weighted simple weak classifiers



Weak classifier: $h_j = \begin{cases} 1, & if\ f_j(x) > \theta_j \\ 0, & otherwise \end{cases}$

P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ④ **Cascading Classifiers**

- **Observations:**

— On average only 0.01% of all sub-windows are positive (faces)

— Equal computation time is spent on all sub-window

— Shouldn't we spend most time only on potentially positive sub-windows?

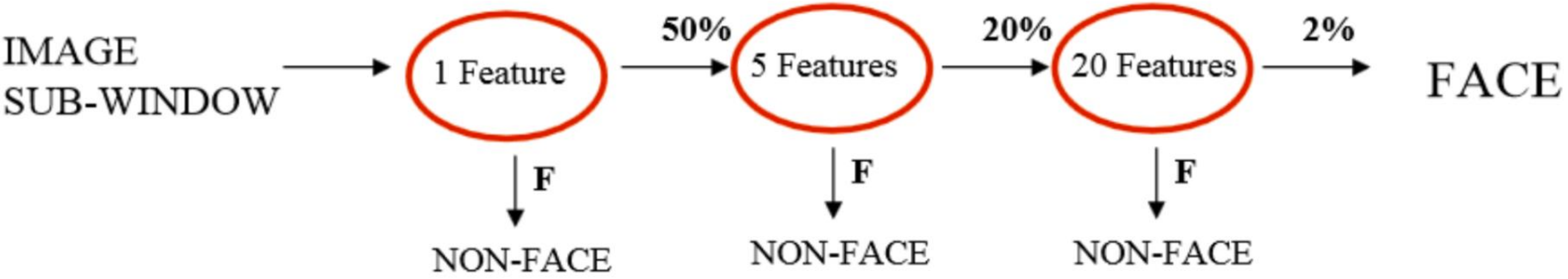
- **Solution:**

— A simple 2-feature classifier can act as
  - 1st layer of a series to filter out most negative (clearly non-face) windows
  - 2nd layer with 10 features can tackle "harder" negative-windows which survived the 1st layer, and so on…

P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ④ **Cascading Classifiers**

- To make detection faster, features can be reordered by increasing complexity of evaluation and the thresholds adjusted so that the early (simpler) tests have few or no false negatives
- Any window that is rejected by early tests can be discarded quickly without computing the other features



P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Summary

- Detection scores in the deformable part model are based on both appearance and location

- The deformable part model is trained iteratively by alternating the steps

— Assume components and part locations given; compute appearance and offset models

— Assume appearance and offset models given; compute components and part locations

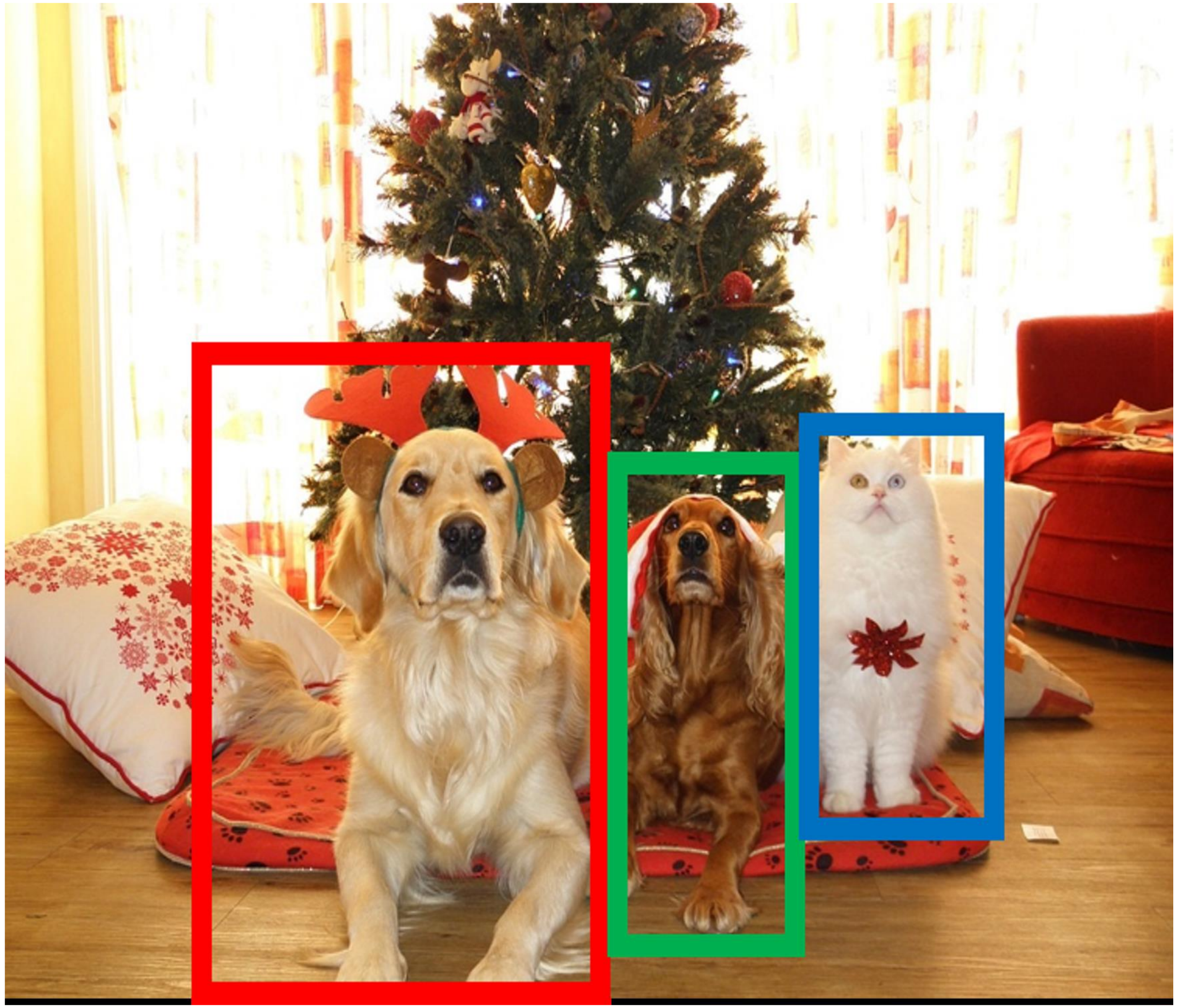
Recent Object Detection

- **Input:** Single RGB Image
- **Output:** A set of detected objects

- **For each object prediction:**

— Category label
  - From fixed, known set of categories

— Bounding box
  - Four numbers: x, y, width, height



Detecting **A Single Object**

- Treat the localization as a regression problem

What

Feature Extraction

4096-d
feature vector

Classification

Class Scores

- *Cat*: 0.9
- *Dog*: 0.05
- *Car*: 0.01
- ⋮

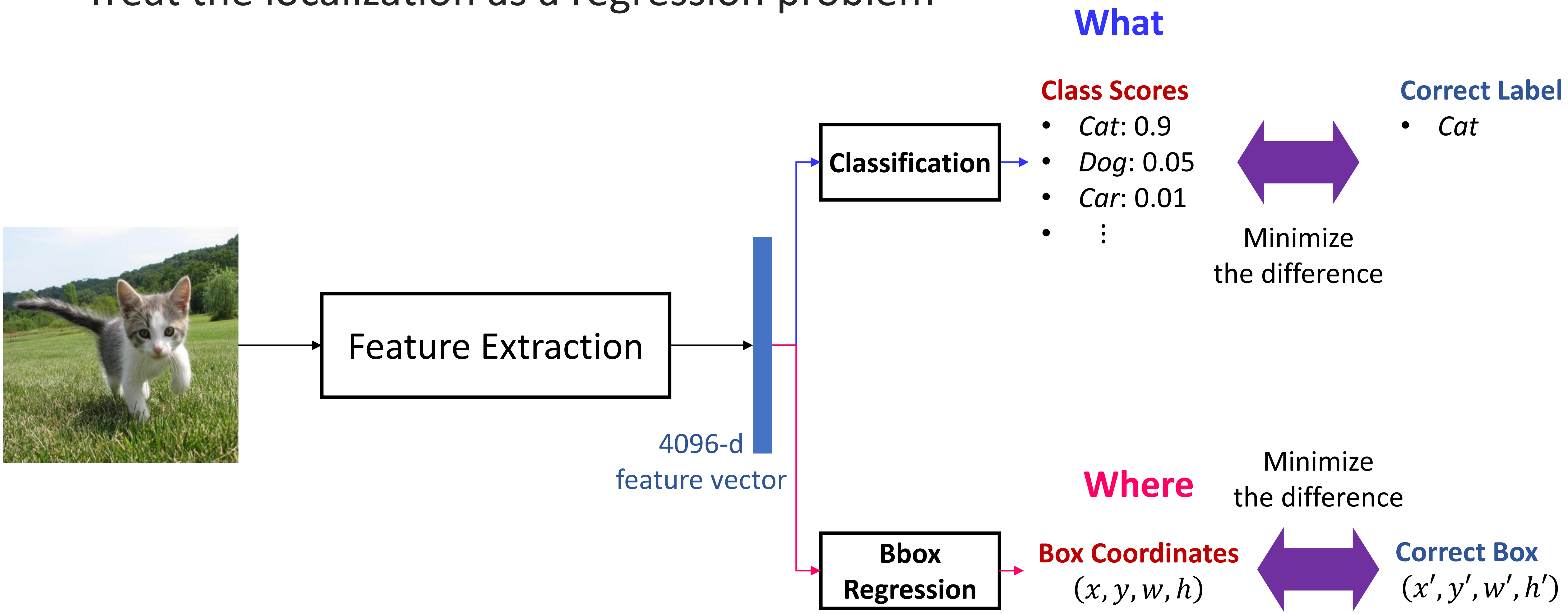
Minimize
the difference

Correct Label

- *Cat*

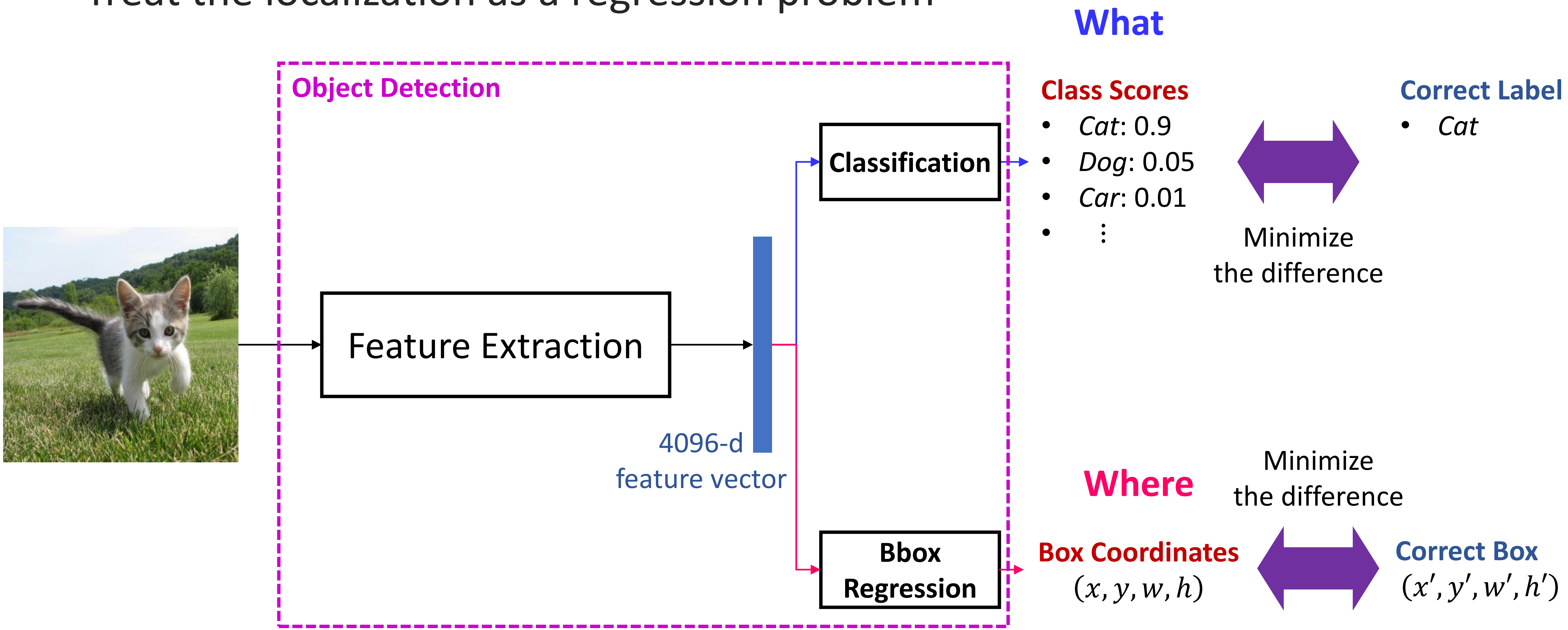
Detecting **A Single Object**

- Treat the localization as a regression problem



Detecting **A Single Object**

- Treat the localization as a regression problem



Detecting **Multiple Objects**

- **Problem:** Images can have more than one object
- **Solution:** Different numbers of outputs per image

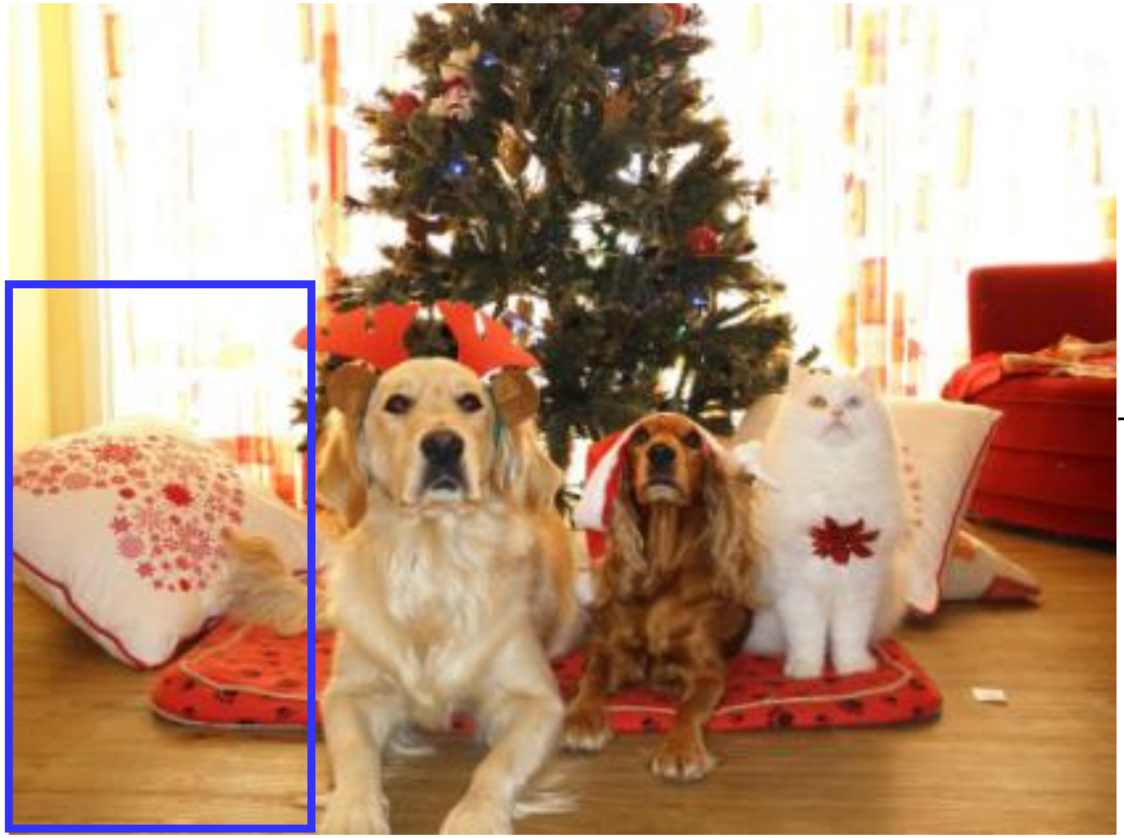
Object Detection → *Cat*: (x, y, w, h) : 4 numbers

Object Detection →
Dog: (x, y, w, h)
Dog: (x, y, w, h) : 12 numbers
Cat: (x, y, w, h)

Object Detection →
Duck: (x, y, w, h)
Duck: (x, y, w, h) : Many numbers
$\vdots$

Detecting Multiple Objects: **Sliding Window**

- Apply an object detection to **many different crops** of the image, the classifier classifies each crop as object or background



→ Object Detection →

- *Dog*? NO
- *Cat*? NO
- *Background*? YES

Detecting Multiple Objects: **Sliding Window**

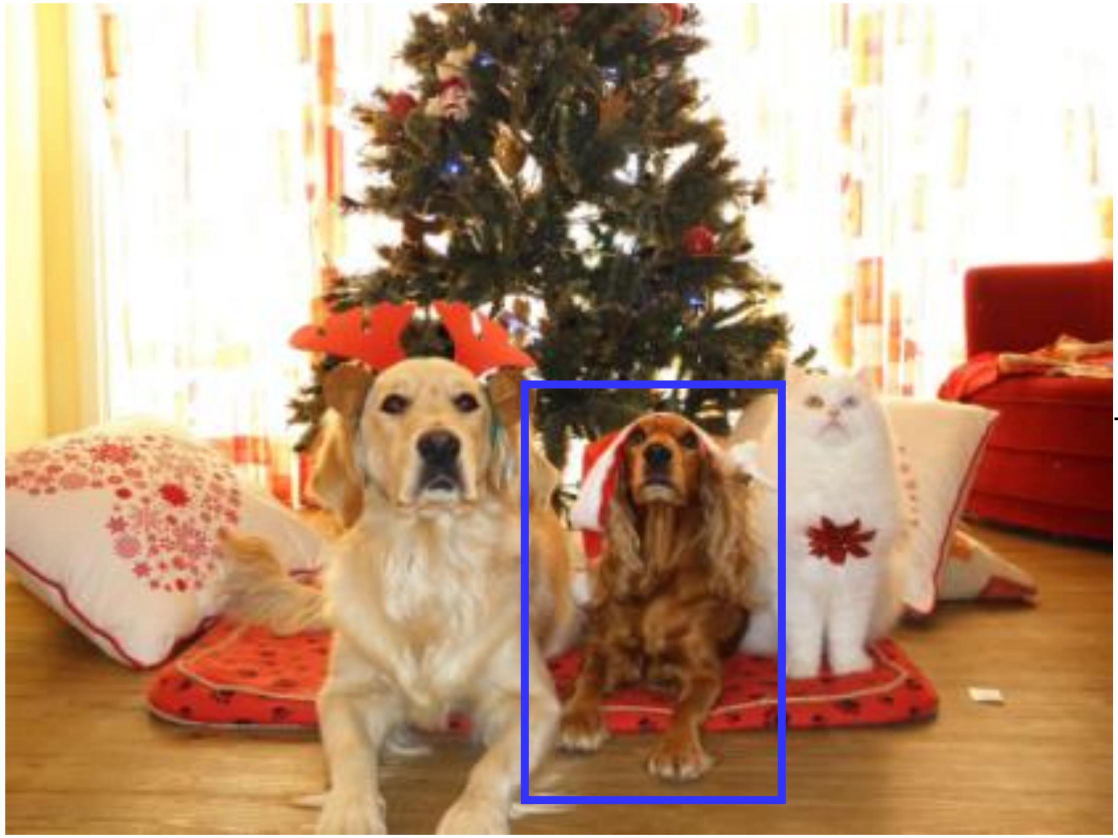
- Apply an object detection to **many different crops** of the image, the classifier classifies each crop as object or background

Object Detection

- *Dog*? YES
- *Cat*? NO
- *Background*? NO

Detecting Multiple Objects: **Sliding Window**

- Apply an object detection to **many different crops** of the image, the classifier classifies each crop as object or background



Object Detection

- *Dog*? YES
- *Cat*? NO
- *Background*? NO

Detecting Multiple Objects: **Sliding Window**

- Apply an object detection to **many different crops** of the image, the classifier classifies each crop as object or background

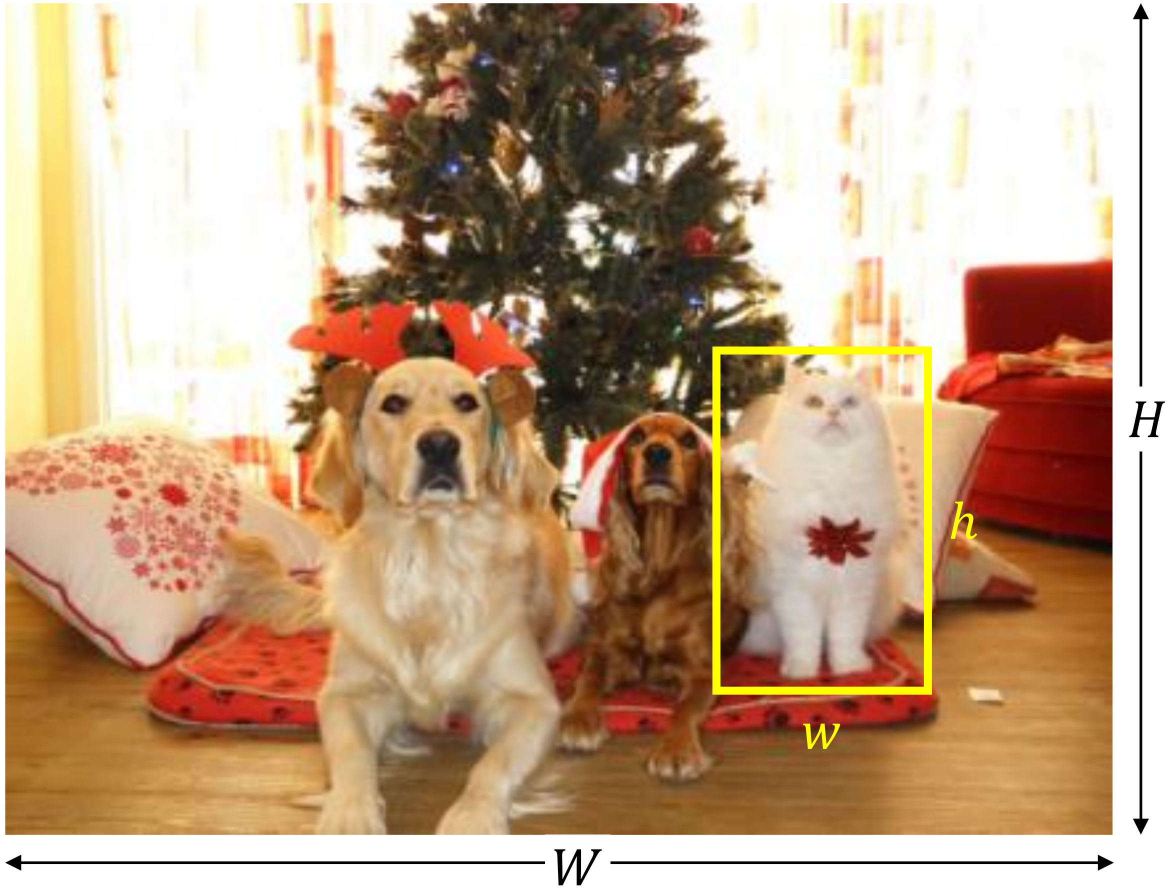
Object Detection

- *Dog*? NO
- *Cat*? YES
- *Background*? NO

Detecting Multiple Objects: **Sliding Window**

- How many possible boxes are there in an image of size $H \times W$?

— Consider a box of size $h \times w$



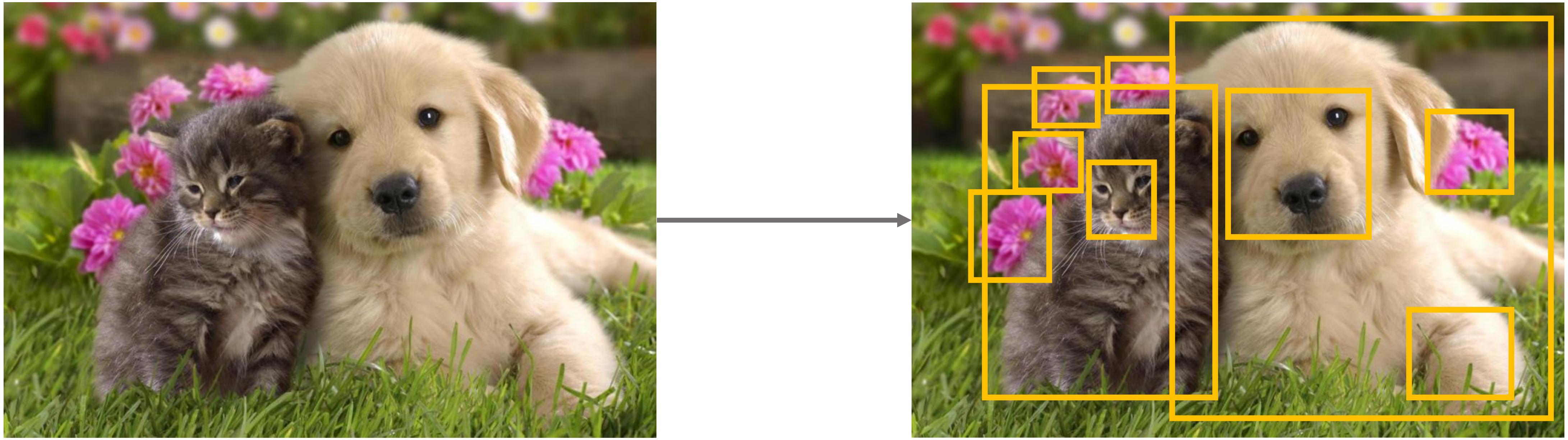
- Possible x positions: $W - w + 1$
- Possible y positions: $H - h + 1$
- Possible positions: $(W - w + 1) \times (H - h + 1)$
- **Total possible boxes:**

$$\sum_{h=1}^{H} \sum_{w=1}^{W} (W - w + 1) \times (H - h + 1)$$

$$= \frac{H(H+1)}{2} \frac{W(W+1)}{2}$$

Region Proposals

- Object region proposal algorithms generate **a short list of regions that have generic object-like properties**
- The object detector then considers **a small set of candidate regions only,** instead of exhaustive sliding window search

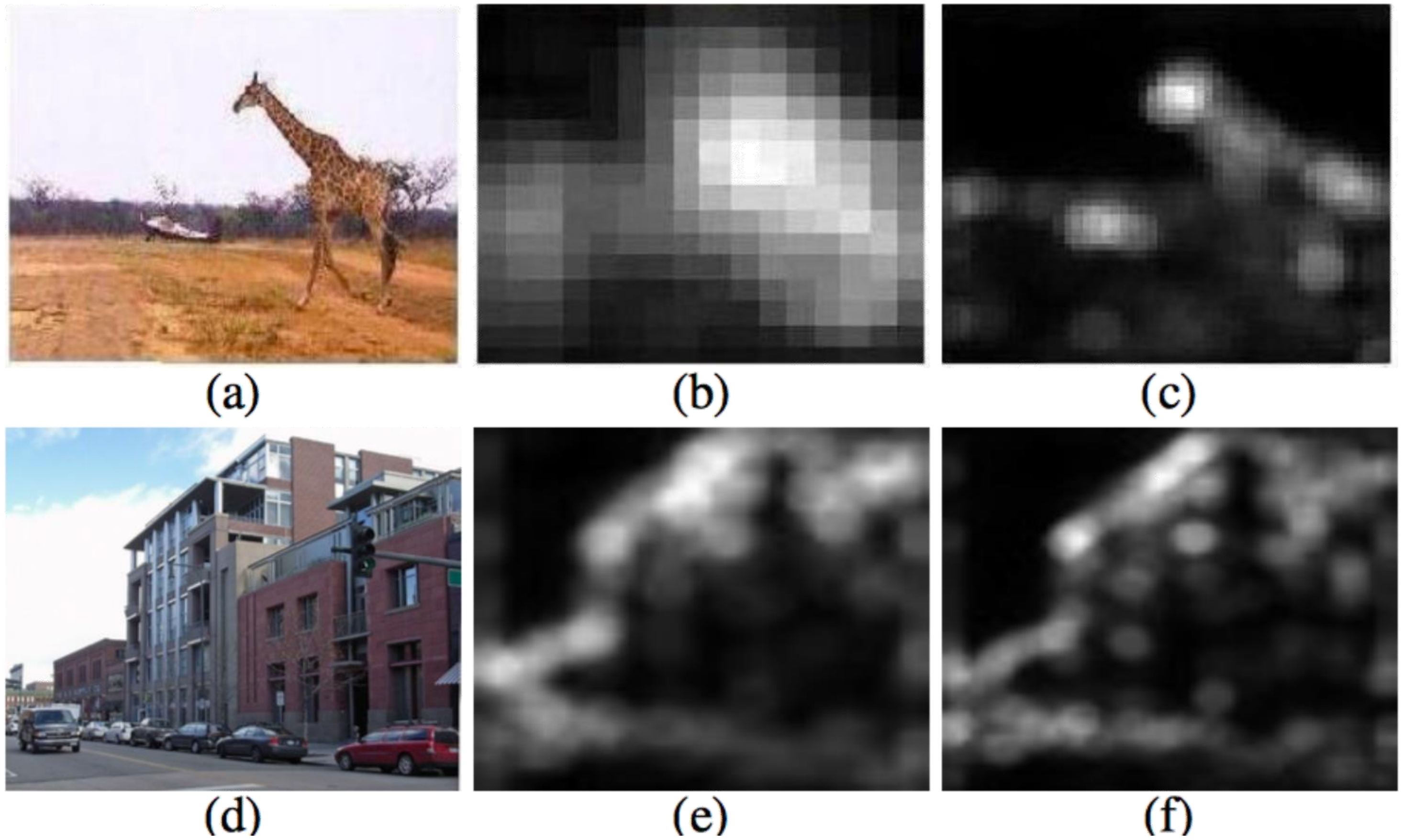


B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**
J. R. R. Uijlings et al., Selective Search for Object Recognition**, IJCV 2013**
M. M. Cheng et al., BING: Binarized Normed Gradients for Objectness Estimation at 300fps, **CVPR 2014**
C. L. Zitnick and P. Dollar, Edge Boxes: Locating Object Proposals from Edges, **ECCV 2014**

Region Proposals: **Multiscale Saliency**

- Favors regions with a unique appearance within the image

High Scale | **Low** Scale



Successful Case

Failure Case

B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**

Region Proposals: **Color Contrast**

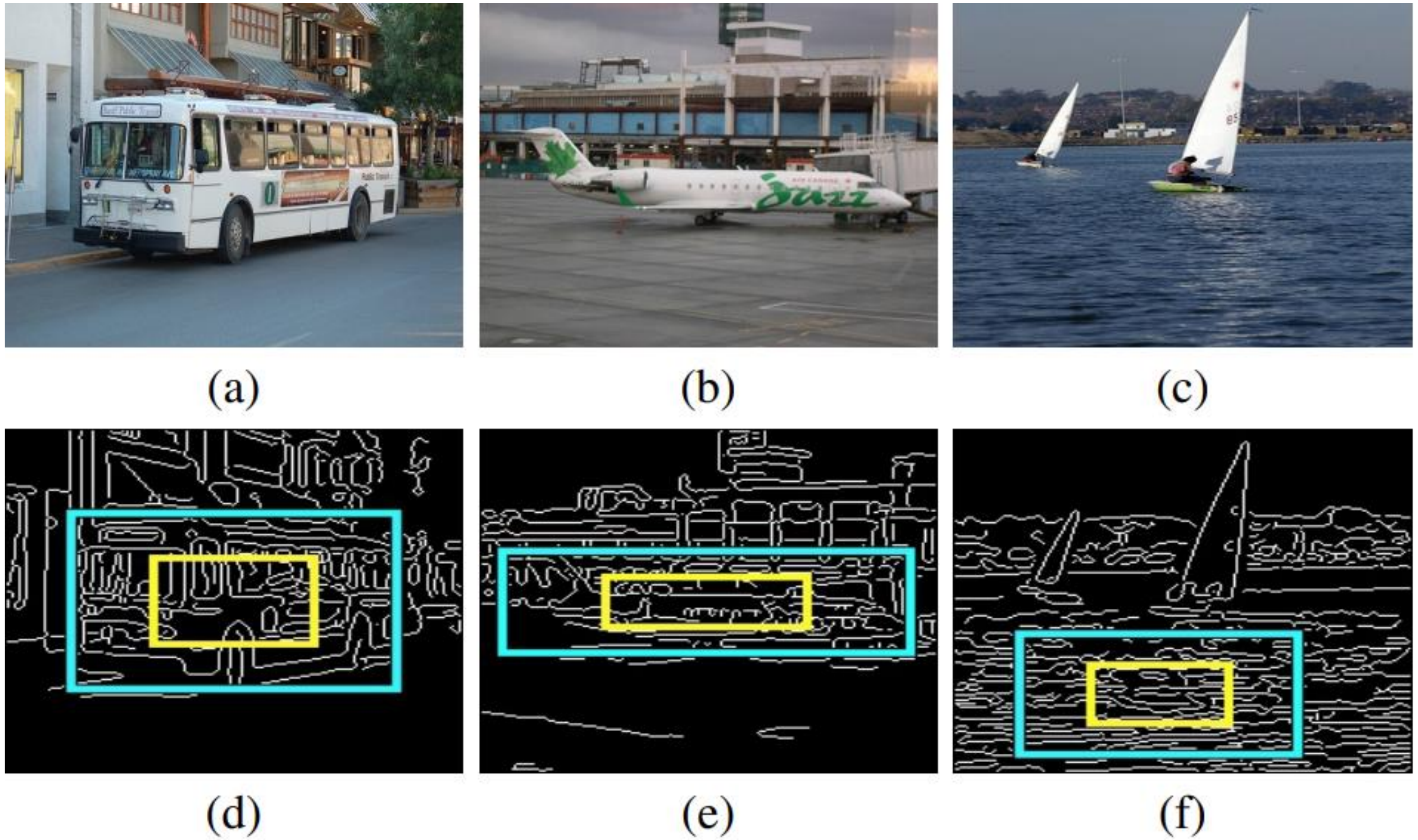
- Favors regions with a contrasting color appearance from immediate surroundings

Successful Cases　　　　**Failure** Case

B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**

Region Proposals: **Edge Density**

- Favors regions with the density of edges near the window borders



Successful Cases **Failure** Case

B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**

Region Proposals: **Performance Comparison**

TABLE 2: For each detector [11, 18, 33] we report its performance (left column) and that of our algorithm 1 using the same window scoring function (right column). We show the average number of windows evaluated per image #win and the detection performance as the mean average precision (mAP) over all 20 classes.

	[11]	OBJ- [11]	[18]	OBJ- [18]	ESS-BOW[33]	OBJ-BOW
mAP	0.186	0.162	0.268	0.225	0.127	0.125
#win	79945	1349	18562	1358	183501	2997

B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**

Summary: Region Proposals in Object Detection

- An object region proposal algorithm generates **a short list of regions** with generic object-like properties that can be evaluated by an object detector in place of an exhaustive sliding window search