Computer Vision 1

HC5a

Retrieval, Detection, Segmentation

Dr. Martin Oswald, Dr. Dimitris Tzionas, Dr. Arun Mukundan,

[m.r.oswald, d.tzionas, a.mukundan]@uva.nl

University of Amsterdam

computer vision lab

Last Lecture

Recap

How many features are needed?

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

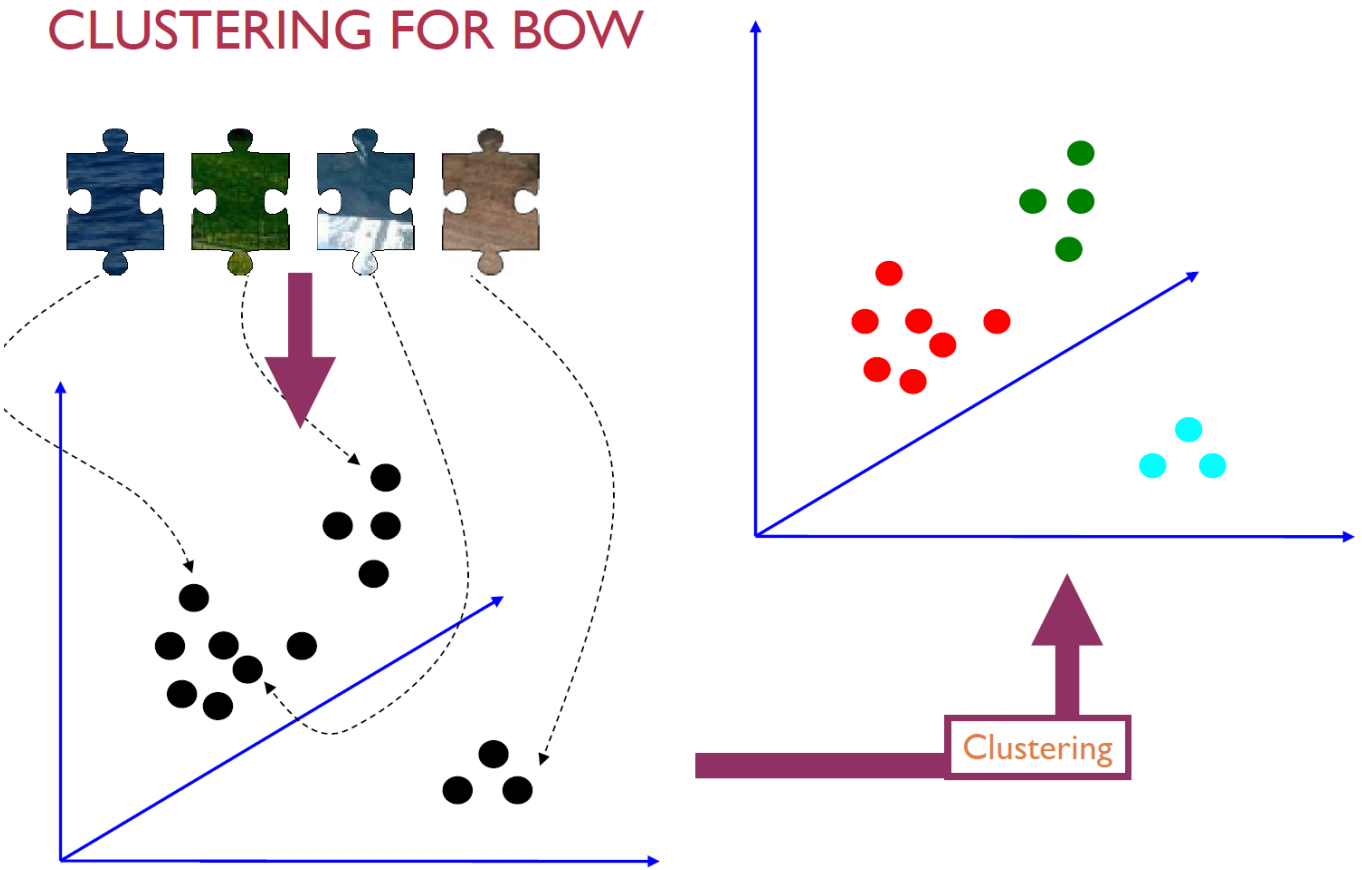
15

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

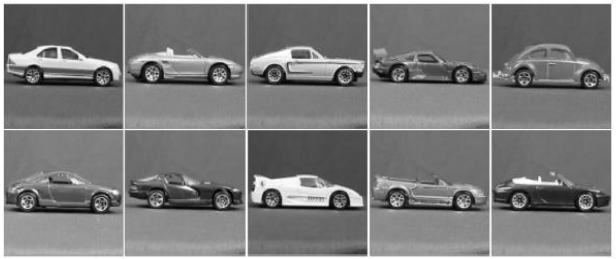
Bag of Visual Words

Slide credits: Jan van Gemert

Feature Clustering

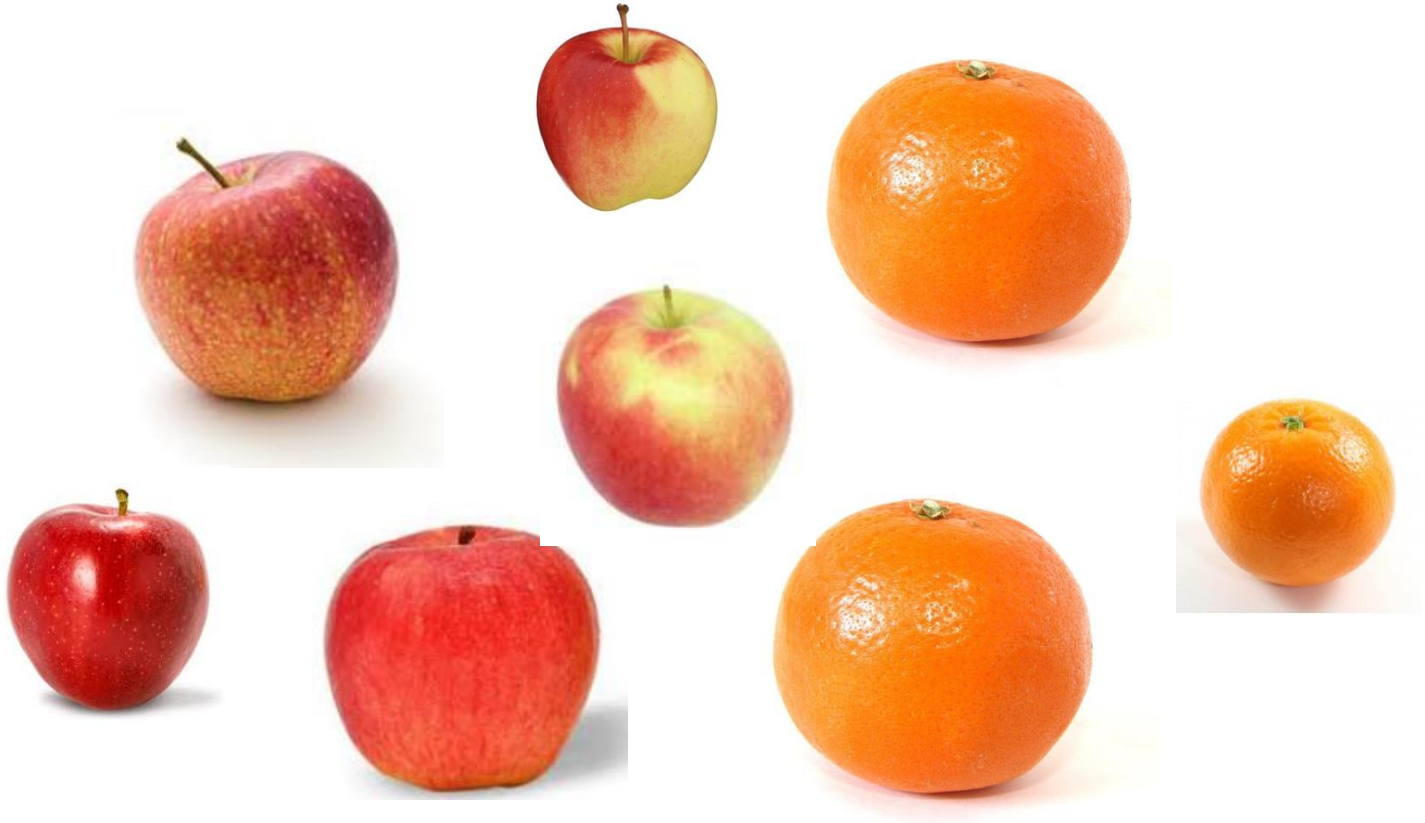


Codebook Example

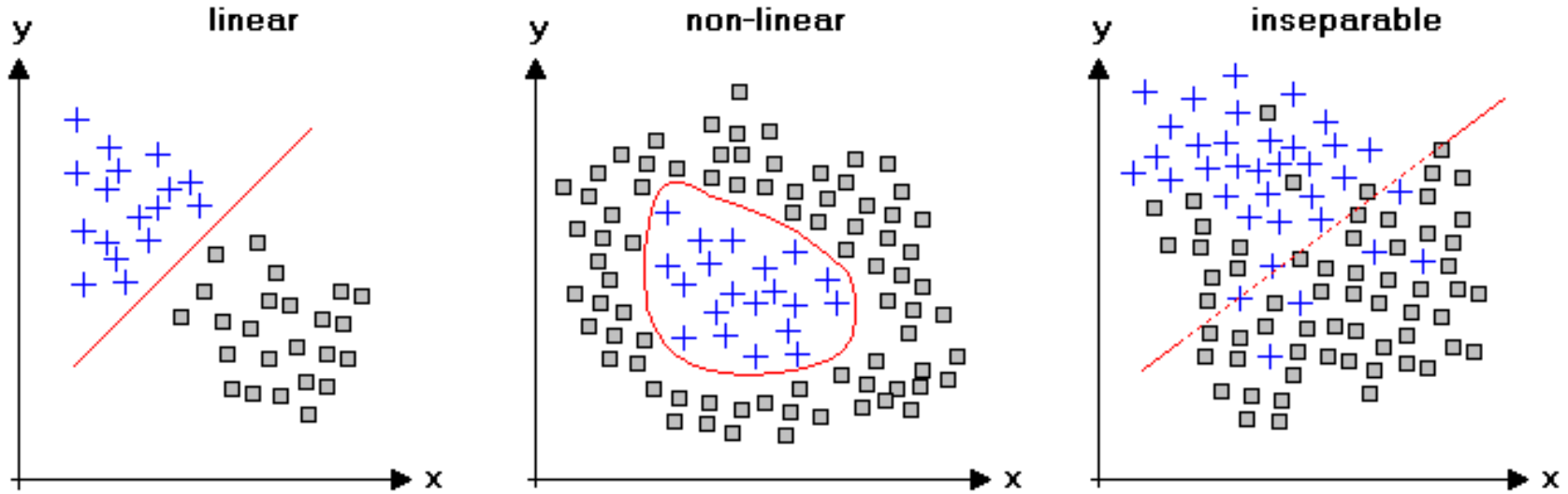


[Slide credit: B. Leibe]

Apple – Orange Classification



Feature Spaces - Separability



Classification - Performance Evaluation

- Accuracy:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}}$$

- Precision and Recall:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- F1 score: geometric mean of precision and recall

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

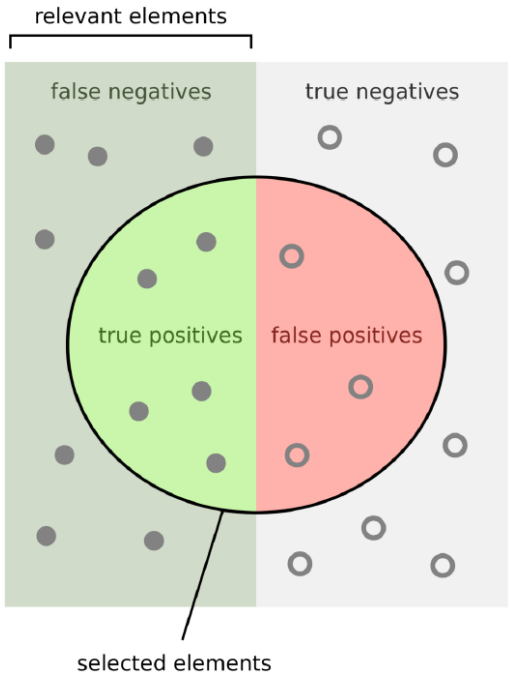
Precision and Recall

How many selected items are relevant?

Precision = 

How many relevant items are selected?

Recall = 



Combine P+R into: F1-Measure

Textbook

- Chapter 6.3
- Chapter 6.4 (introduction)
- Chapter 6.4.2

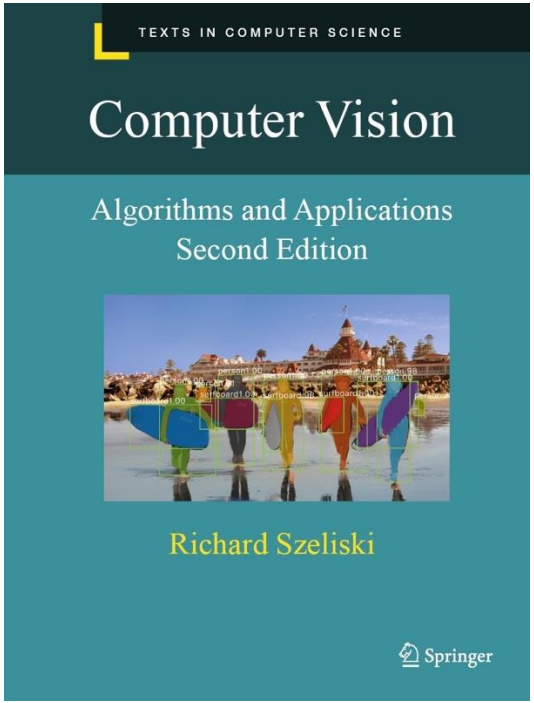


Image Understanding

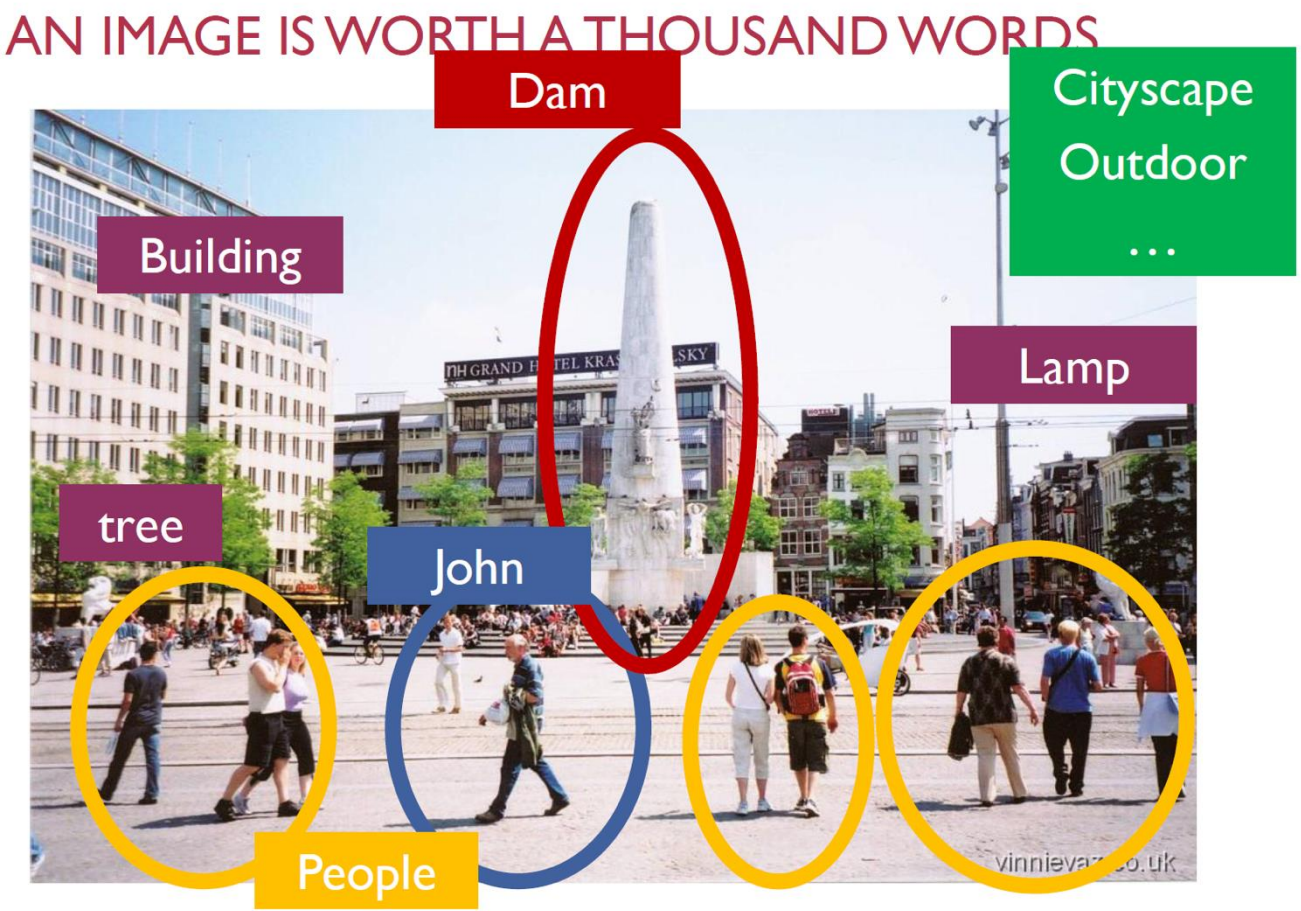
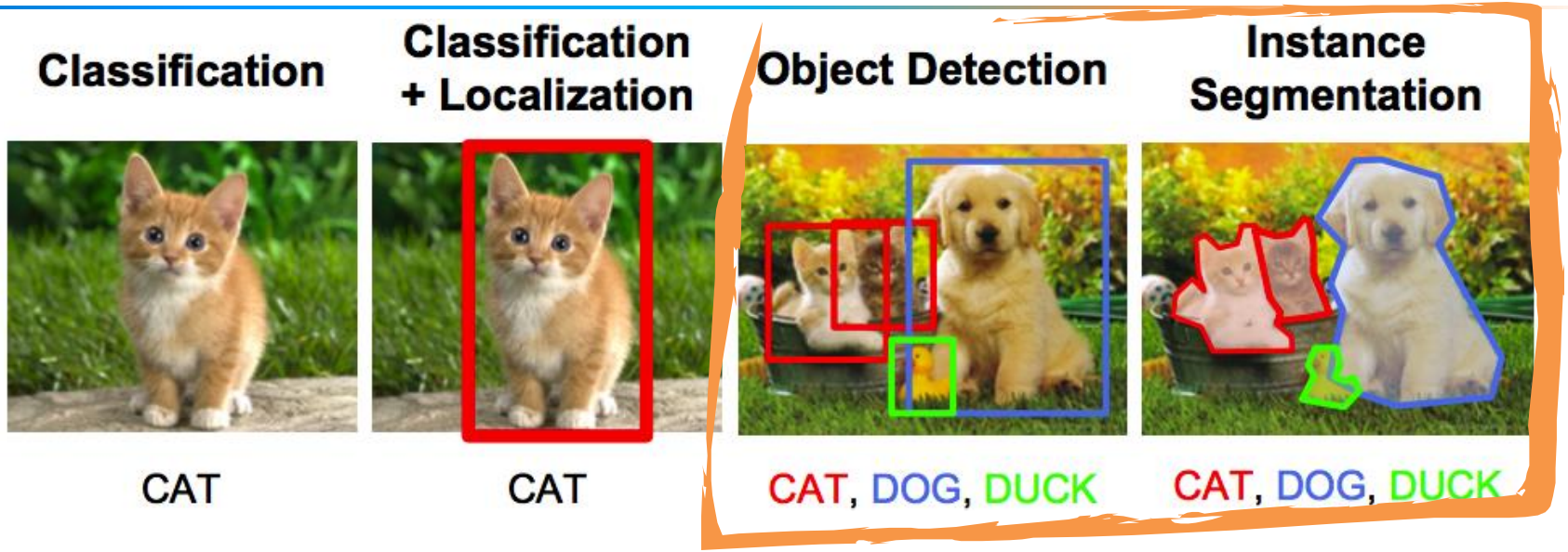


Image Understanding



- Object Classification: classify the main object in an image if the image has one main object.
- Visual Image Retrieval: retrieve images from the library through a query image.
- Object Detection: identify whether there is an object in an image and recognize the object.
- Image Segmentation: segment an image into regions according the pixels the object belong to.

Today: Object Detection + Segmentation

[Slide credit: Justin Johnson]

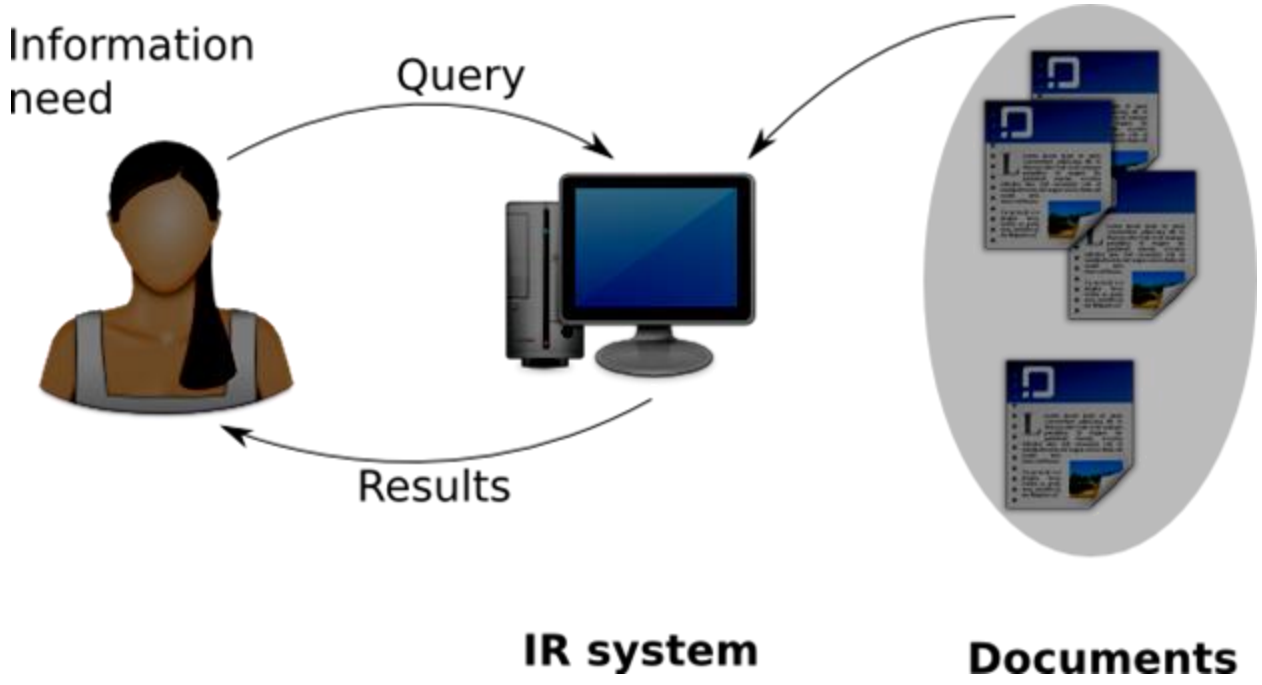
Today's Agenda

- Image Retrieval
- Object Detection
- Image Segmentation

Today's Agenda

- **Image Retrieval**
- Object Detection
- Image Segmentation

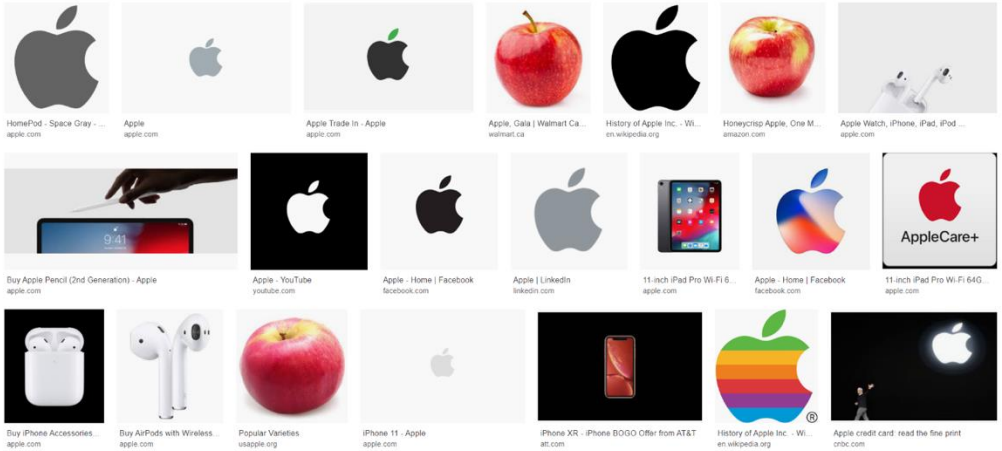
What is Image Retrieval?



What is Image Retrieval?

Image Retrieval

Search term: “apple”



Google Search

Bing Search

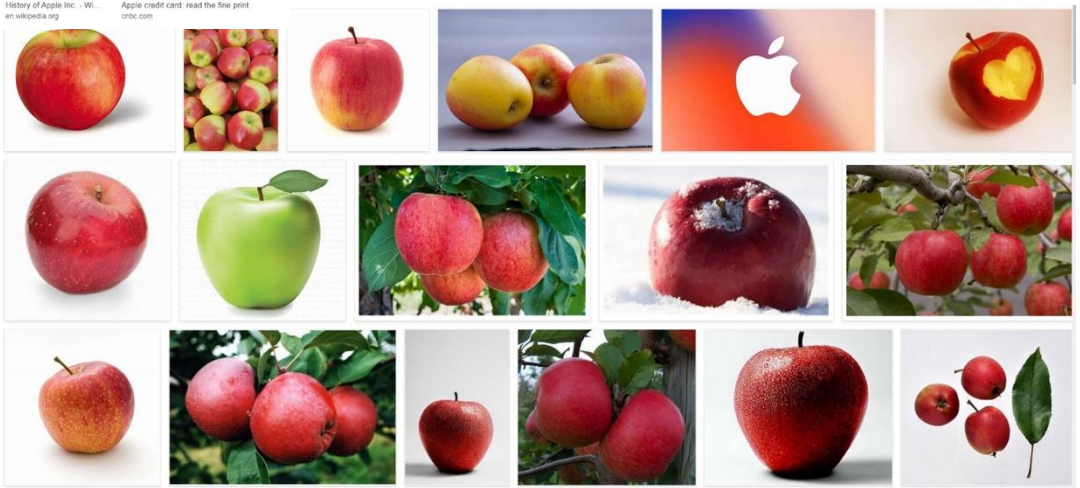
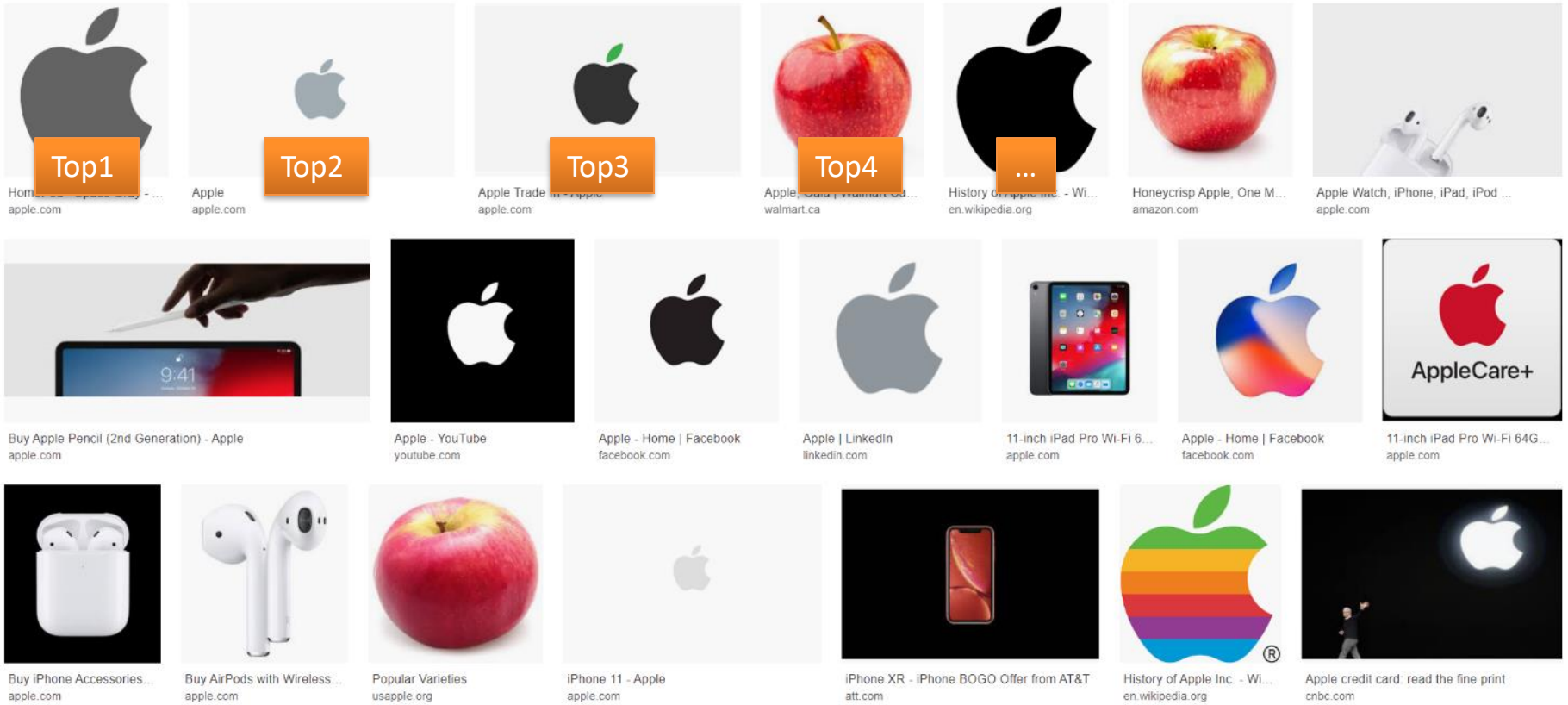


Image Retrieval



Google Search

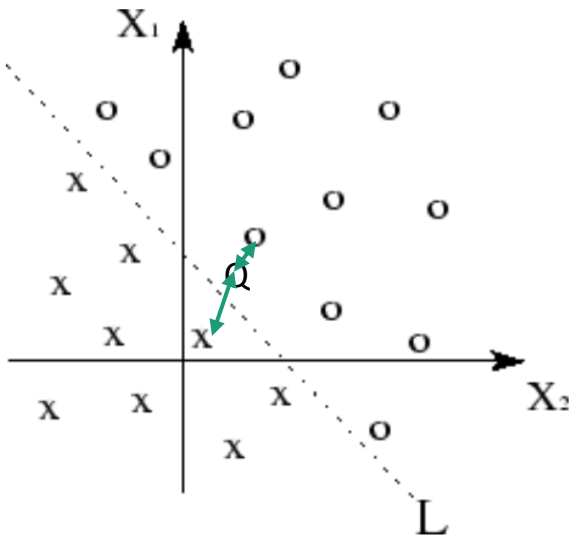
Bag of Words for Image Retrieval

- Offline: construct BoW Vocabulary
- Offline: All images in the dataset -> BoWs

- Create description from Query image
- Compare to all images in the dataset
- Rank according to similarity

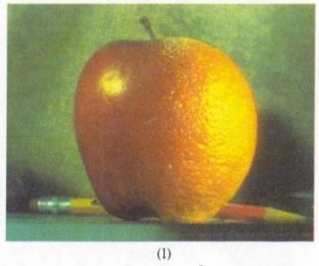
Slide credits: Jan van Gemert

Retrieval Example



Data base

1. A Query: Q



2. Extract feature

3. Get similarity

Top 1: o
Top 2: x
…

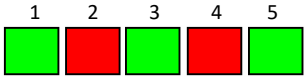
Retrieval Evaluation

Precision@K

- Set a rank threshold K
- Compute % relevant in top K
- Ignores images ranked lower than K

$$\text{P@K} = \underbrace{\frac{TP}{TP + FP}}_{\text{Precision}} = \underbrace{\frac{TP}{K}}_{\text{K}}$$

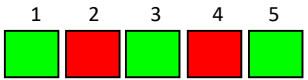
- Example:



  - Prec@3 of 2/3
  - Prec@4 of 2/4
  - Prec@5 of 3/5

Retrieval Evaluation

Average Precision@K

- Consider rank position of each relevant images
  - $K_1, K_2, \ldots, K_R$
- Compute Precision@K for each $K_1, K_2, \ldots, K_R$
- Average precision = average of P@K
- Example: 

has average precision:

rel: 1 0 1 0 1

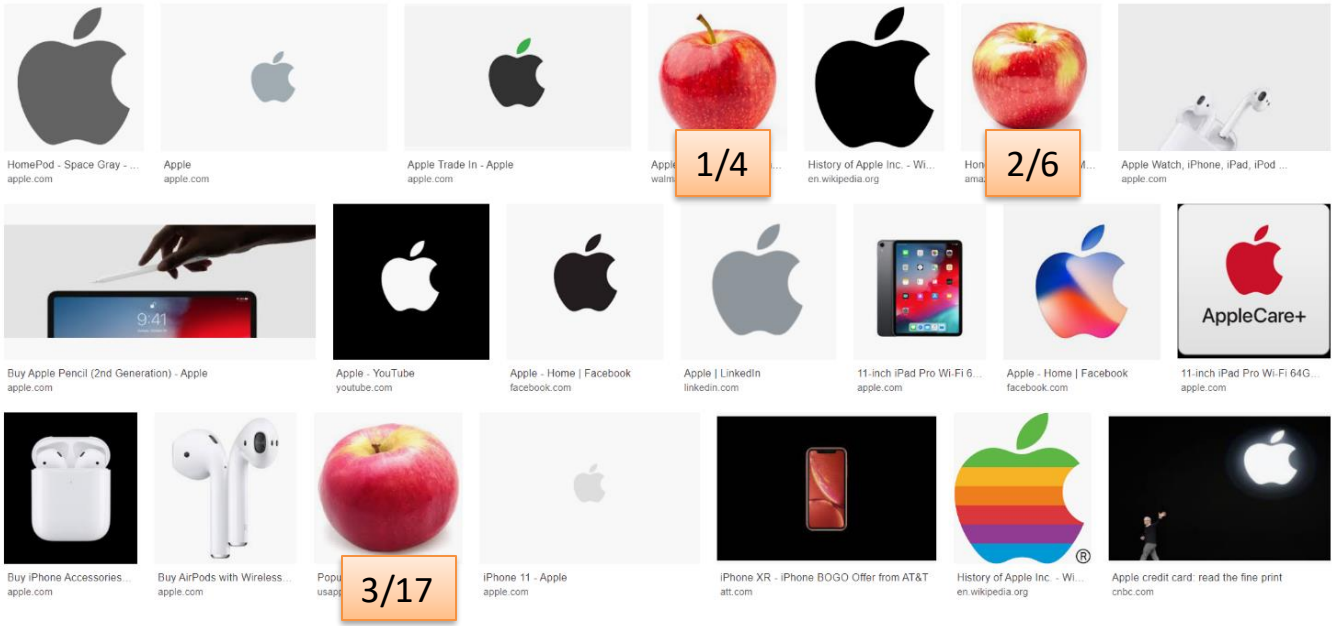
relevance: {0,1}

$$\textbf{AP@K} = \frac{\sum_{k=1}^{K}(P@k * rel(k))}{\#\ relevant\ results}$$

$$AP@K = \frac{1}{3} * \left({}^{1}/_{1} * 1 + {}^{1}/_{2} * 0 + {}^{2}/_{3} * 1 + {}^{2}/_{4} * 0 + {}^{3}/_{5} * 1\right) \approx 0.76$$

Retrieval Evaluation

P@K and AP@K for “apple” (relevant: food)



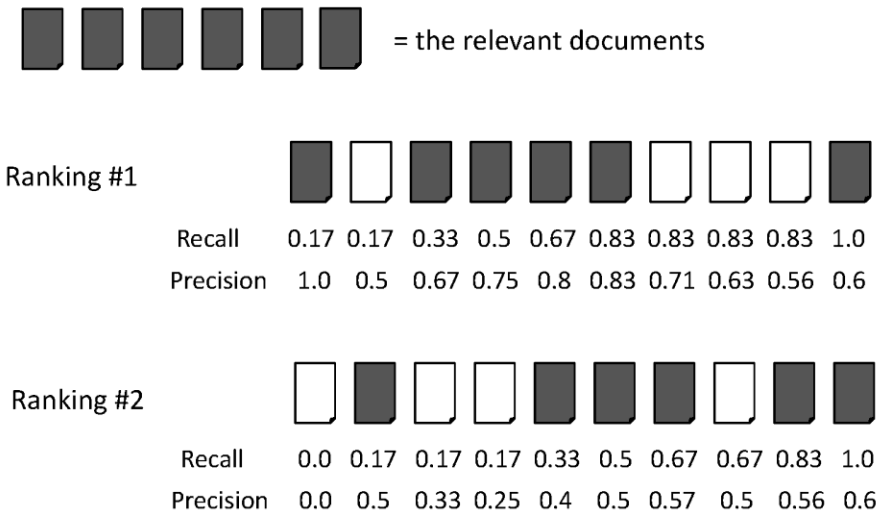
Average Precision @ 17 = 1/3 * (1/4 + 2/6 + 3/17)

Retrieval Evaluation

Mean Average Precision (mAP)

- Take the mean over multiple quieries/rankings

$$mAP@K = \frac{1}{Q}\sum_{q=1}^{Q} AP@kq$$



- Example:

$$\text{Ranking \#1: } (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$

$$\text{Ranking \#2: } (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6)/6 = 0.52$$

$$mAP@K = \frac{1}{2} * (0.78 + 0.52)$$

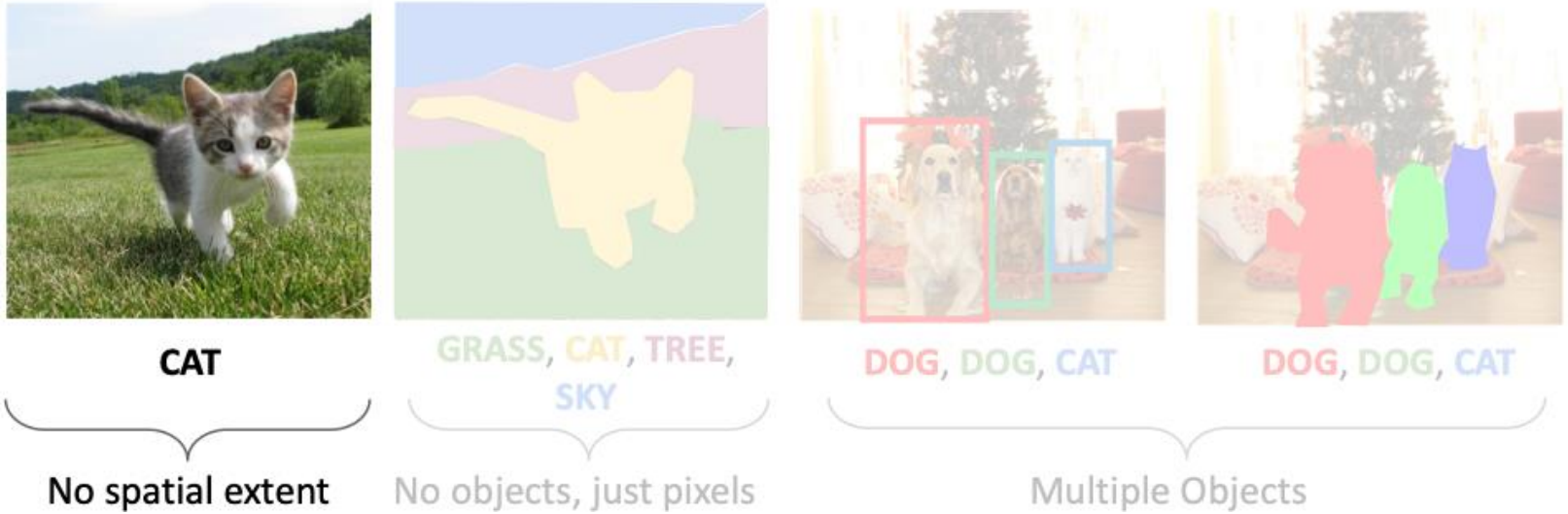
Today's Agenda

- Image Retrieval
- **Object Detection**
- Image Segmentation

Object Detection

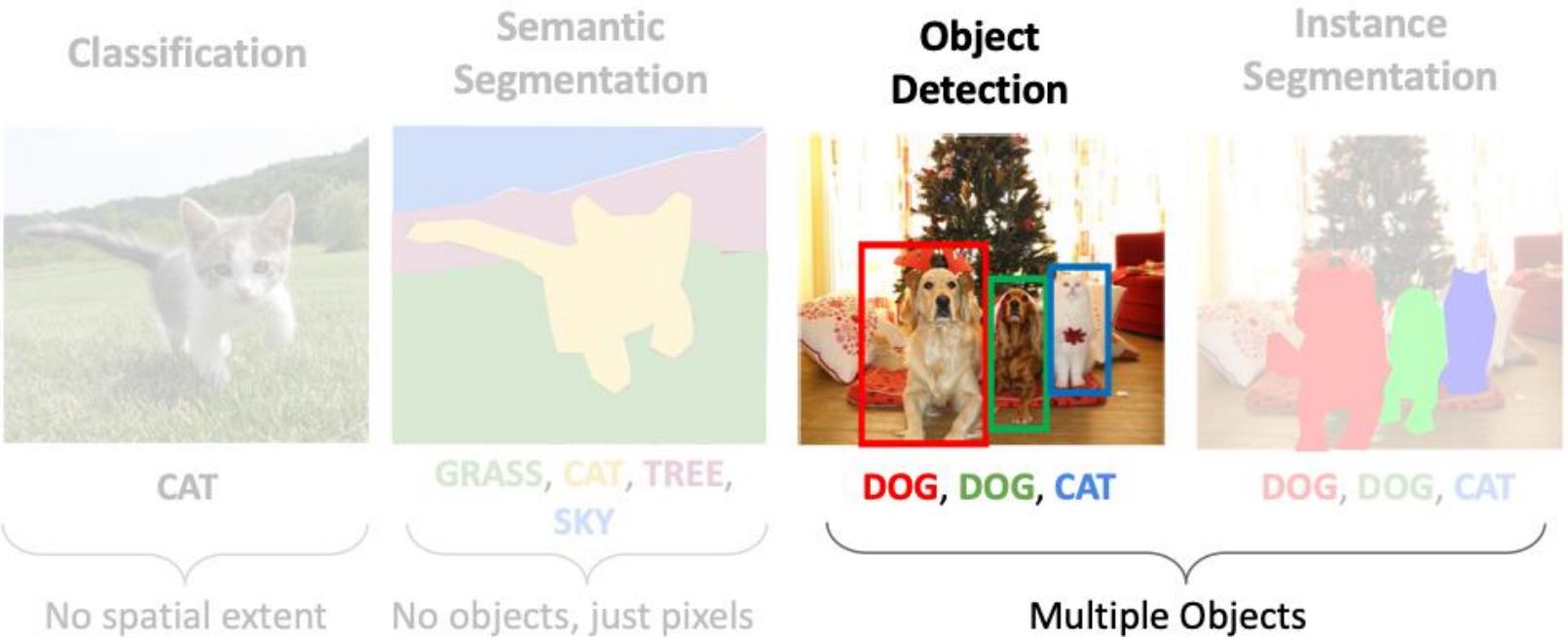
[https://youtu.be/7BgiFrPs-x8?si]

Object Detection



[Slide credit: Justin Johnson]

Object Detection



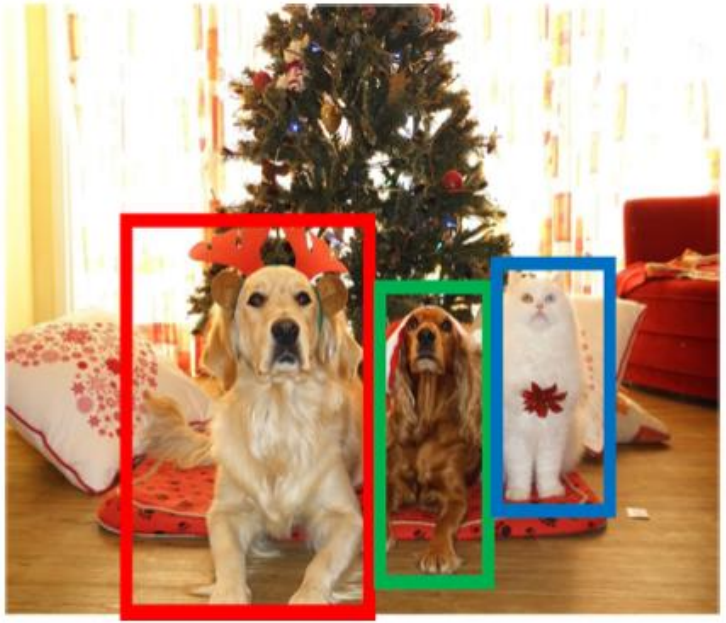
[Slide credit: Justin Johnson]

Object Detection: Task Definition

Input: Single RGB Image

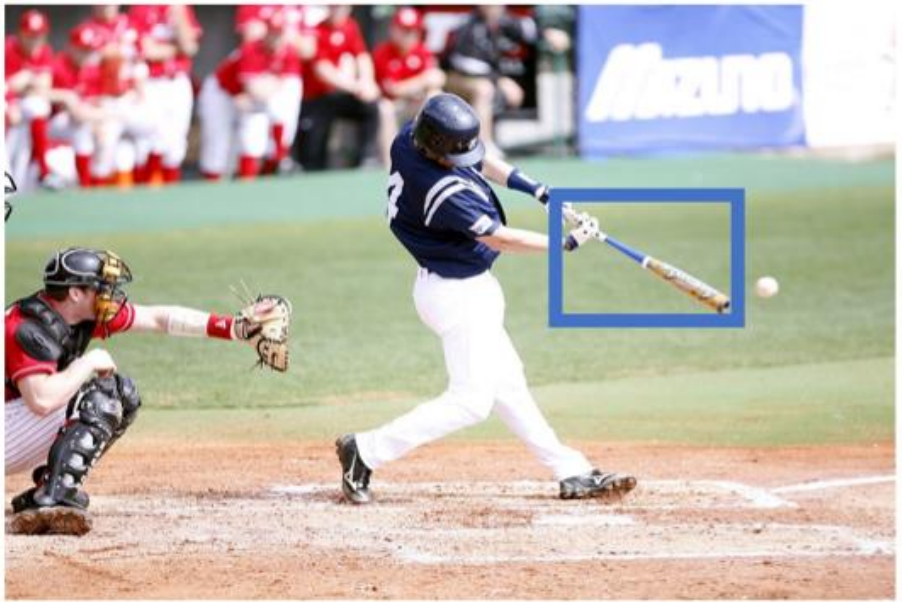
Output: A set of detected objects;
For each object predict:

1. Category label (from fixed, known set of categories)
2. Bounding box (four numbers: x, y, width, height)



Bounding Boxes

Bounding boxes are typically *axis-aligned*

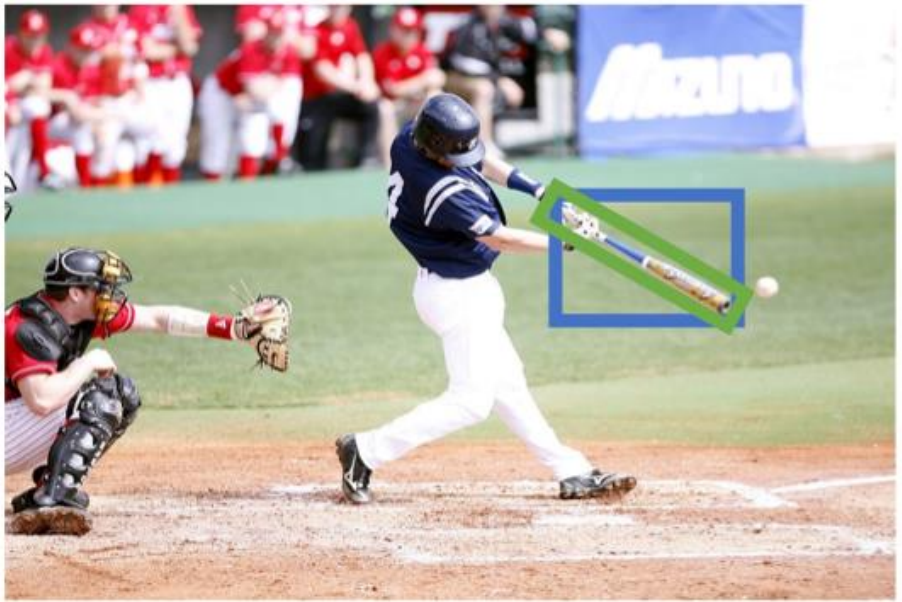


[Slide credit: Justin Johnson]

Bounding Boxes

Bounding boxes are typically *axis-aligned*

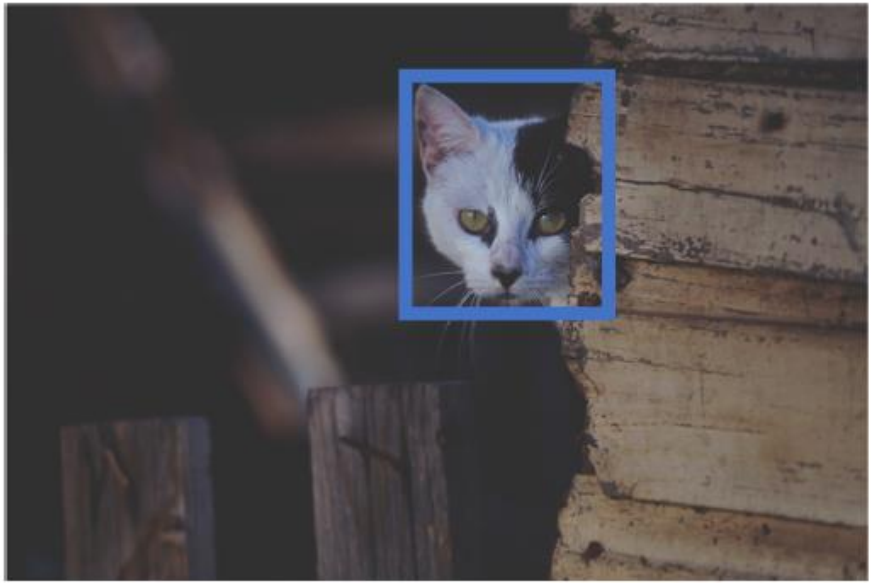
Oriented boxes are much less common



[Slide credit: Justin Johnson]

Bounding Boxes

Bounding boxes (usually) cover only the visible portion of the object

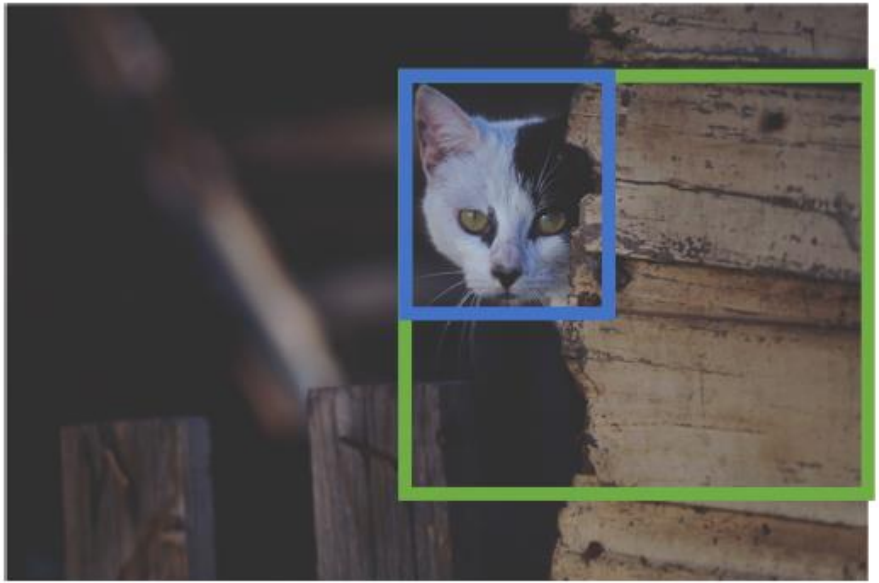


[Slide credit: Justin Johnson]

Bounding Boxes

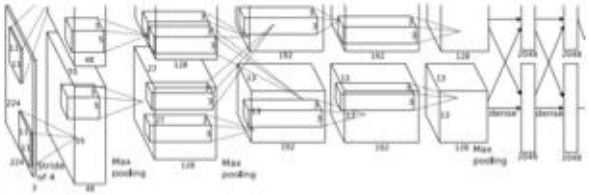
Bounding boxes (usually) cover only the visible portion of the object

Amodal detection: box covers the entire extent of the object, even occluded parts



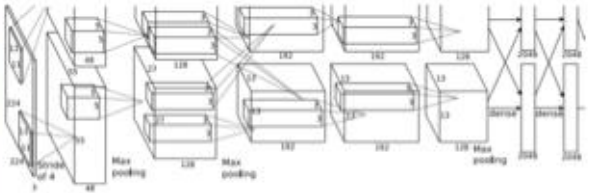
[Slide credit: Justin Johnson]

Detecting Multiple Objects



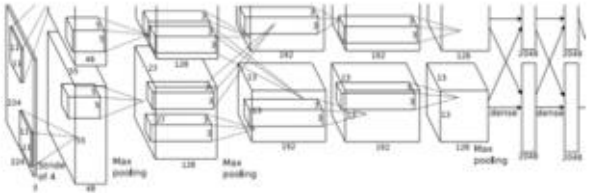
CAT: (x, y, w, h)

4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 numbers



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

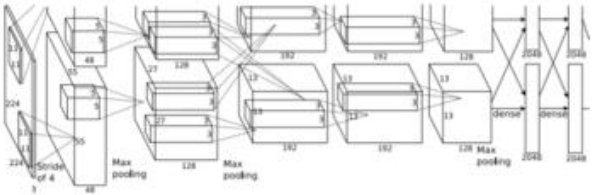
[Slide credit: Justin Johnson]

Object Detection

- Sliding window
- Selective Search
- Region proposals (next lecture)

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

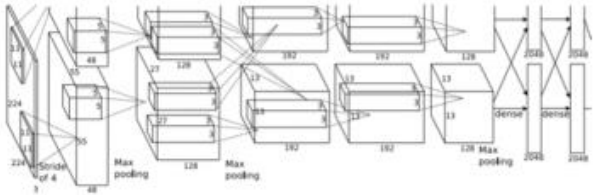


Dog? NO
Cat? NO
Background? YES

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

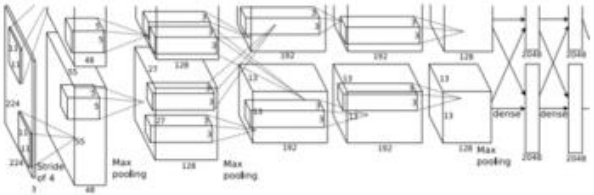


Dog? YES
Cat? NO
Background? NO

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

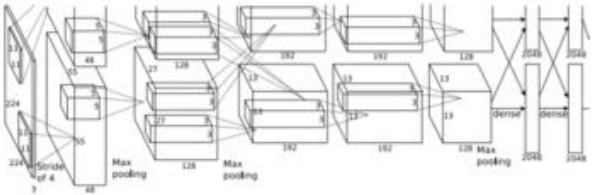


Dog? YES
Cat? NO
Background? NO

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

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

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

[Slide credit: Justin Johnson]

Detecting Multiple Objects: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

800 x 600 image has ~58M boxes! No way we can evaluate them all

Question: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

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

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

[Slide credit: Justin Johnson]

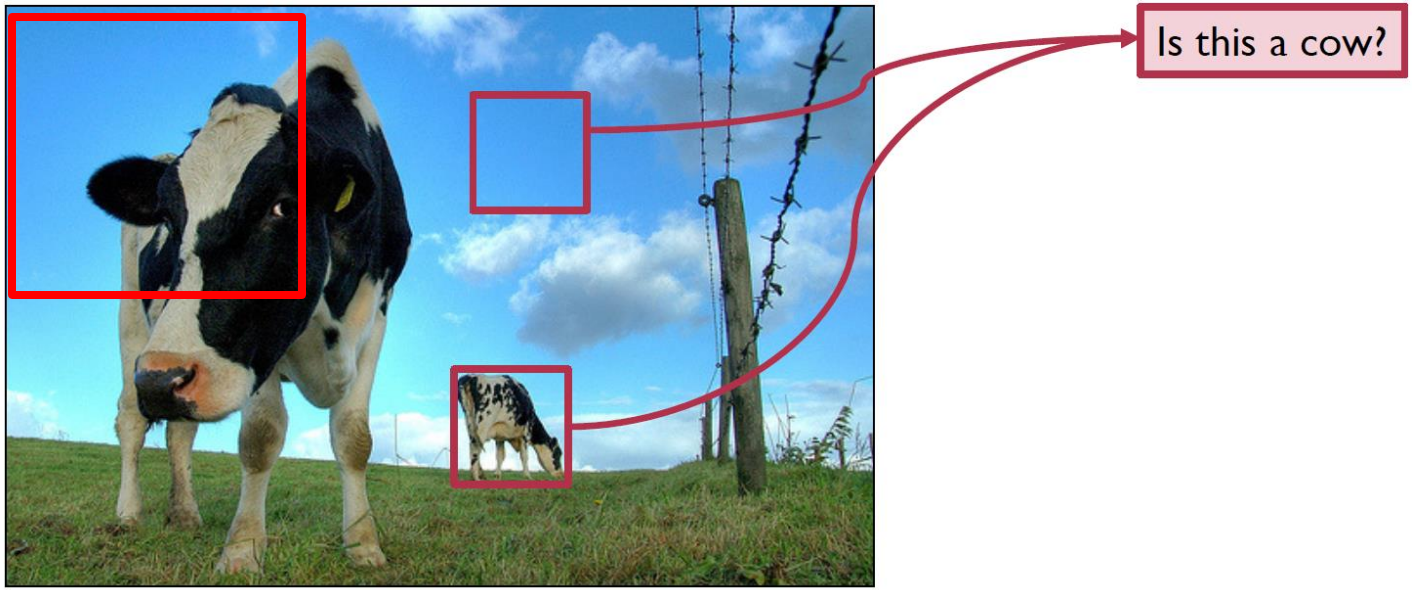
Region Proposals

- Find a small set of boxes that are likely to cover all objects
- Often based on heuristics: e.g. look for “blob-like” image regions
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU

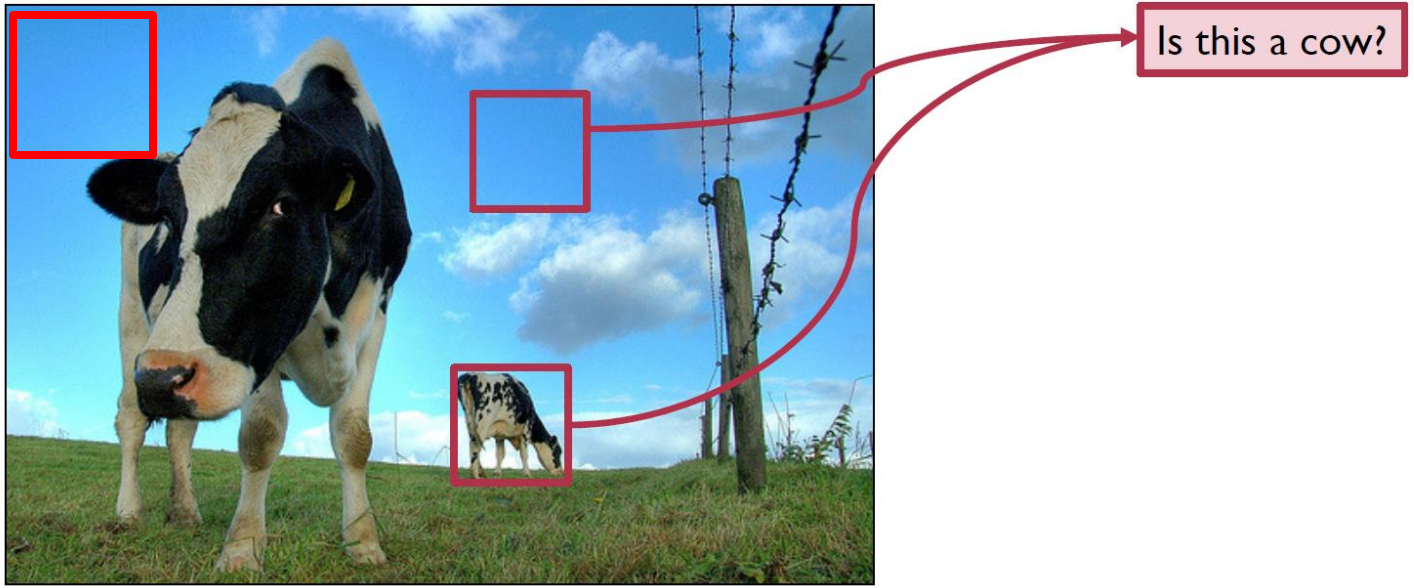
[Slide credit: Justin Johnson]

How to Find The Bounding Box?

Sliding Window



Sliding Window



Sliding Window

+ Many boxes
+ Many scales
+ Many box ratios
+ Many classes
= **Computational Expensive**

Selective Search

- **Idea: Images are hierarchical**

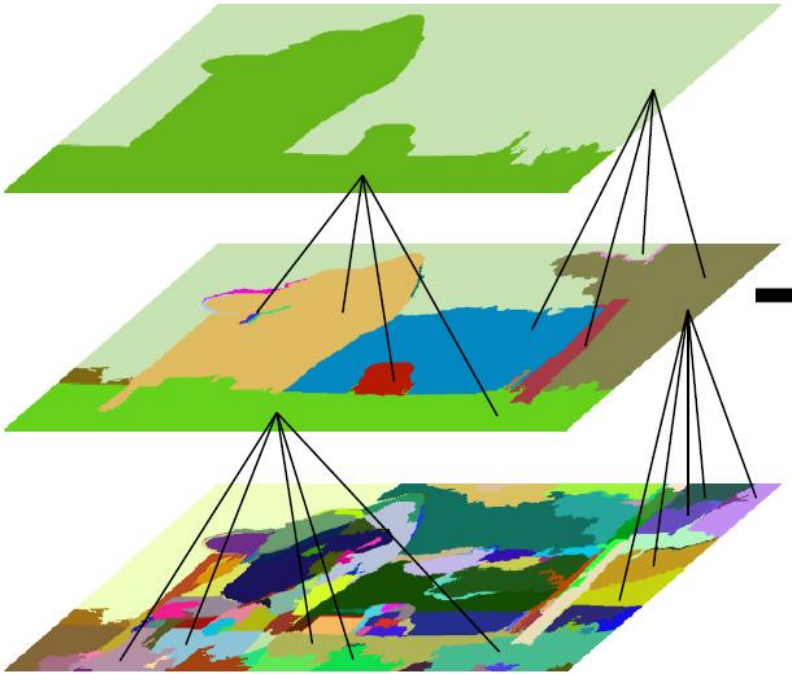
[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

Selective Search

- **Initial segments from over-segmentation**

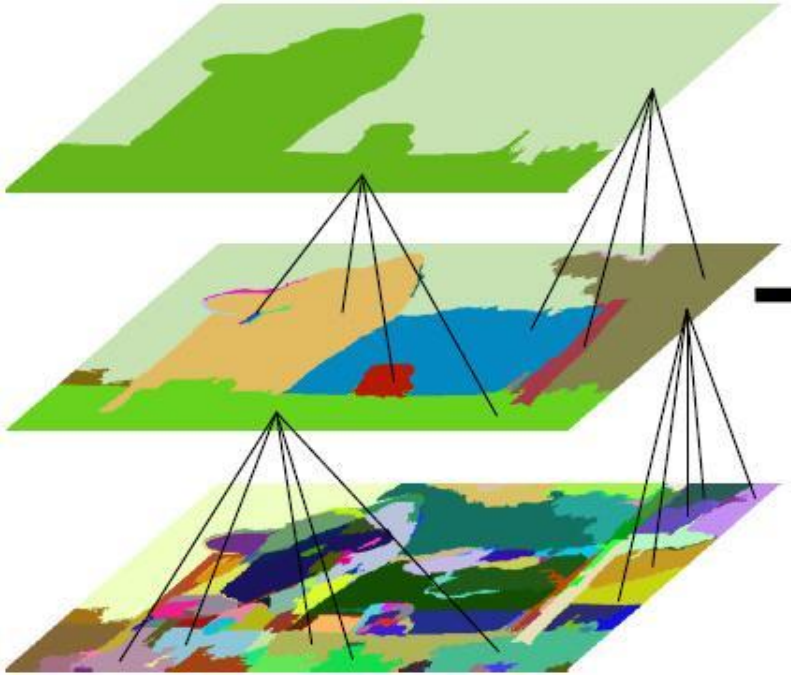
[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

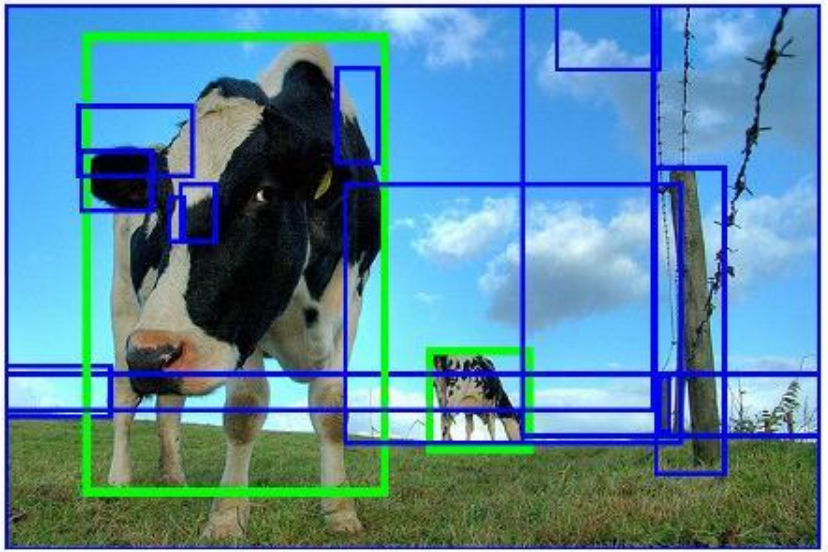
Selective Search



[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

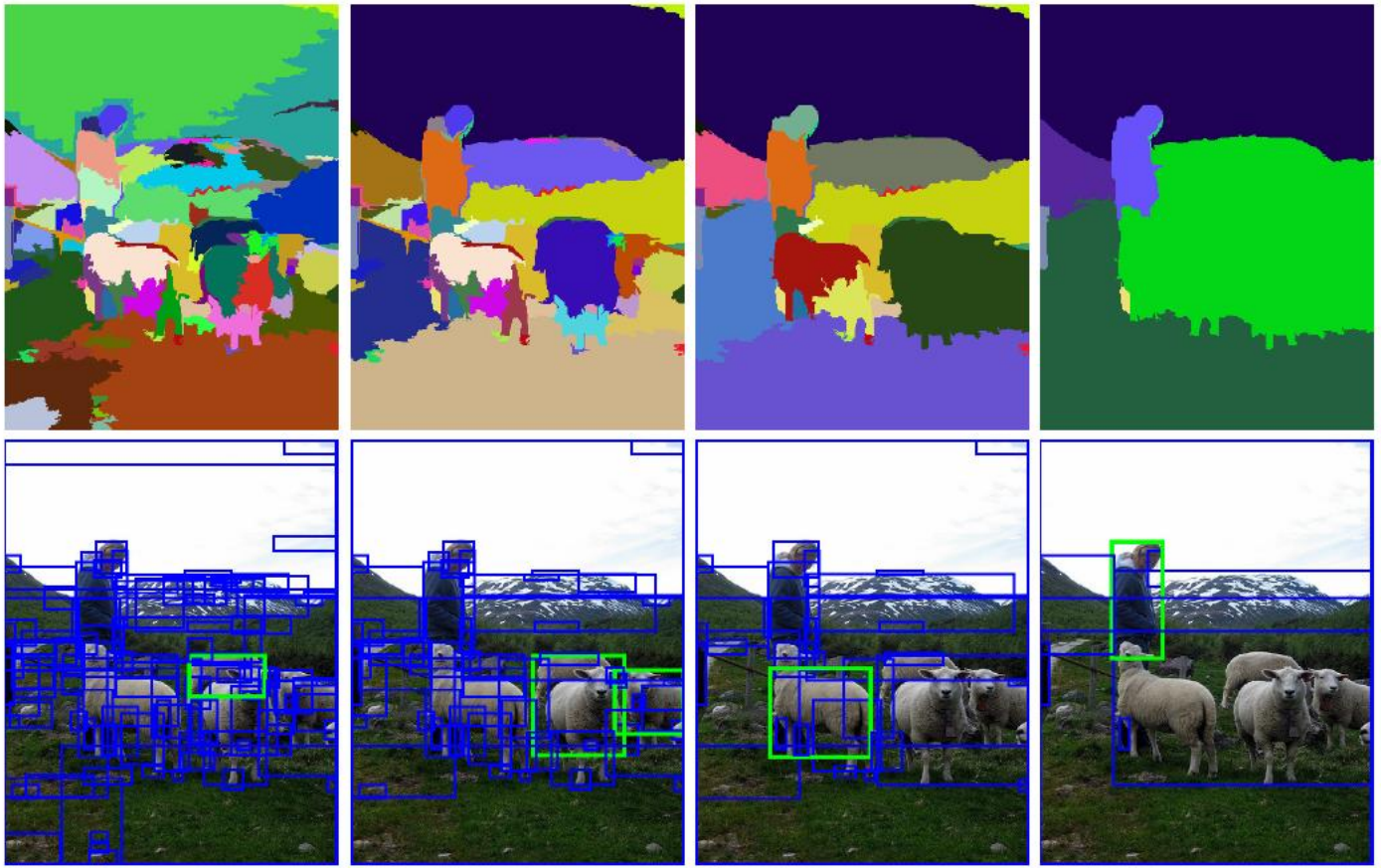
Selective Search





[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

Selective Search - Examples

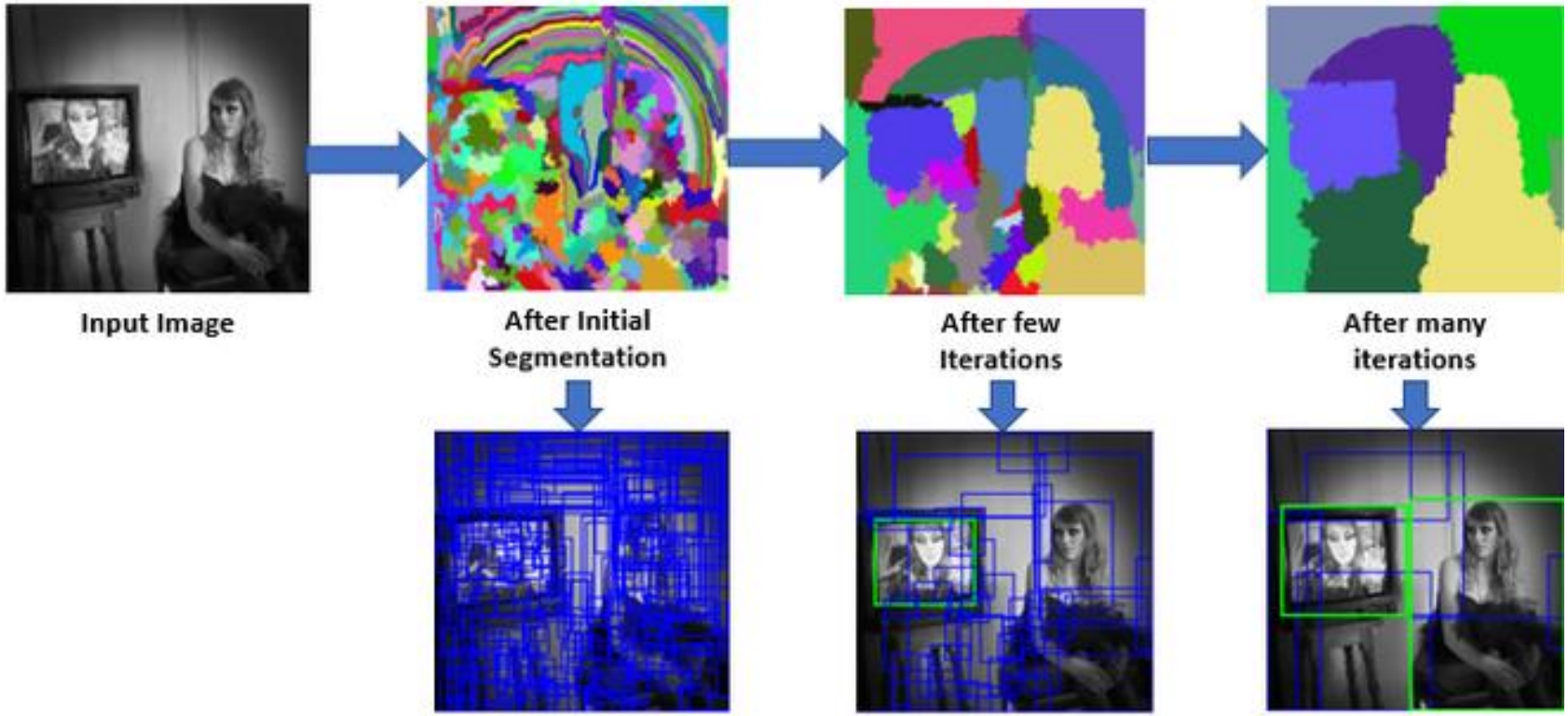


[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

Selective Search - Examples

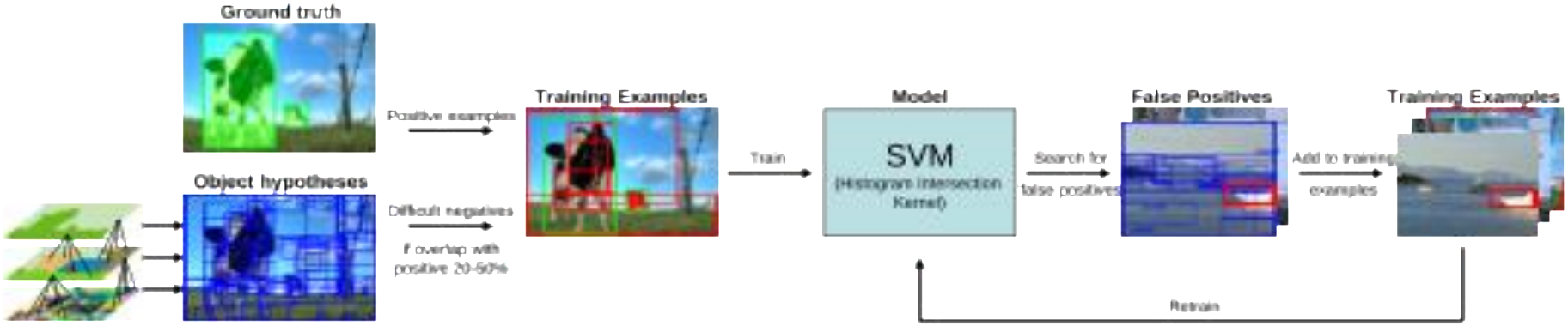
[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

Selective Search - Examples



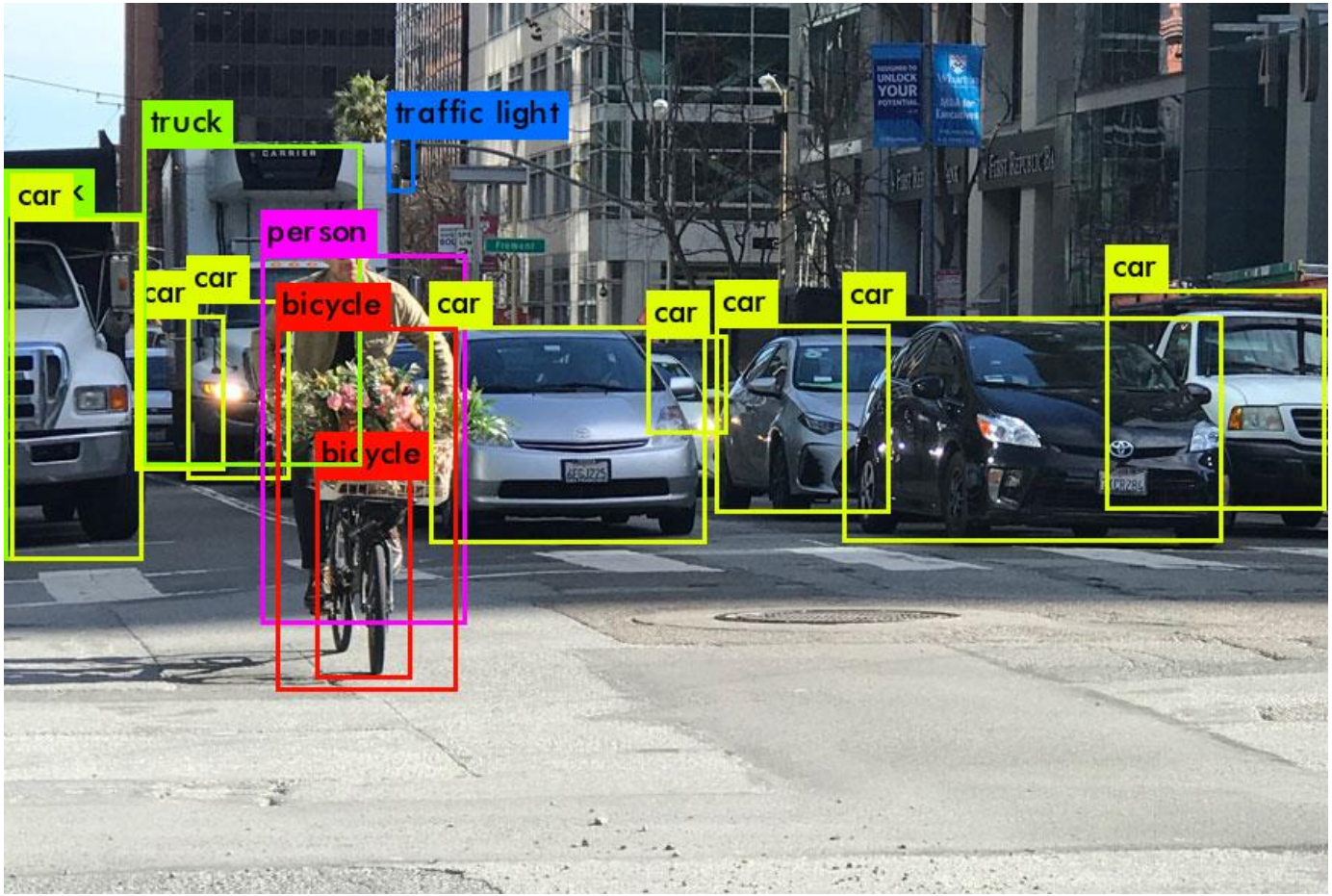
[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013] [image souce]

Selective Search in Object Recognition



[Uijlings, van de Sande, Gevers, Smeulders: Selective Search for Object Recognition, IJCV 2013]

You only look once (YOLO)



You only look once (YOLO)

Autonomous Driving

Object Detection Performance Evaluation

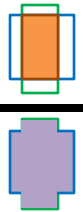
- How can we evaluate object detection?

- If we binary select inliniers and outliers of predicted bounding boxes, we can use retrieval evaluation measures:
  Precision and Recall, AP, mAP, etc.

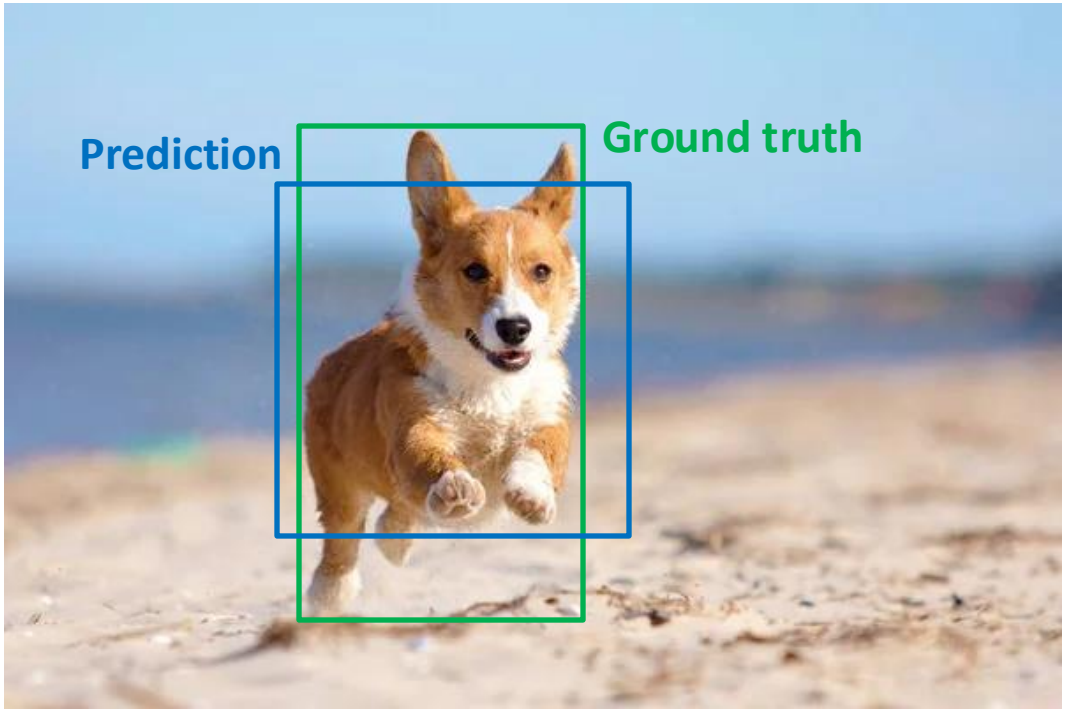
Comparing Bounding Boxes

- Intersection over Union (IoU)
  (a.k.a. “Jaccard similarity/index”)

$$IoU = \frac{\text{Intersection}}{\text{Union}}$$



- Typically choose a threshold e.g. IoU >= 0.5, or 0.7 between prediction and GT to decide about correct or incorrect predictions



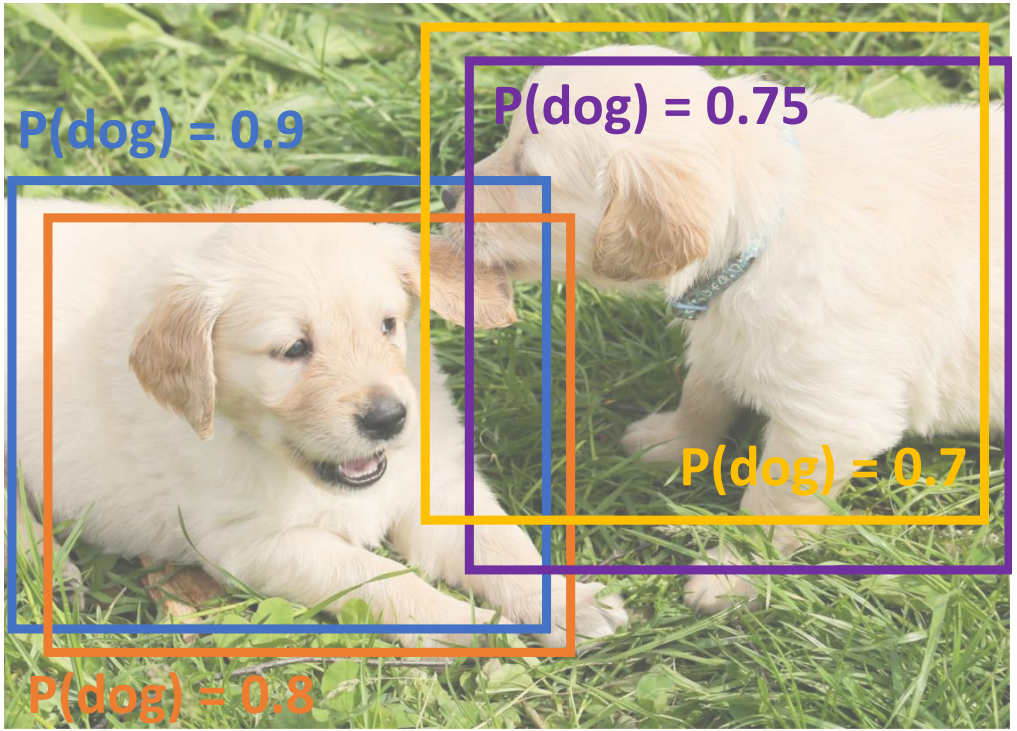
[https://depositphotos.com/photos/corgi-running.html]

Duplicate Filter: Non-Maximum Suppression (NMS)

Problem: Object detectors often output many overlapping detections

Solution: Post-process raw detections using **Non-Max Suppression (NMS):**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1



[Puppy image is CC0 Public Domain]

[Slide credit: Justin Johnson]

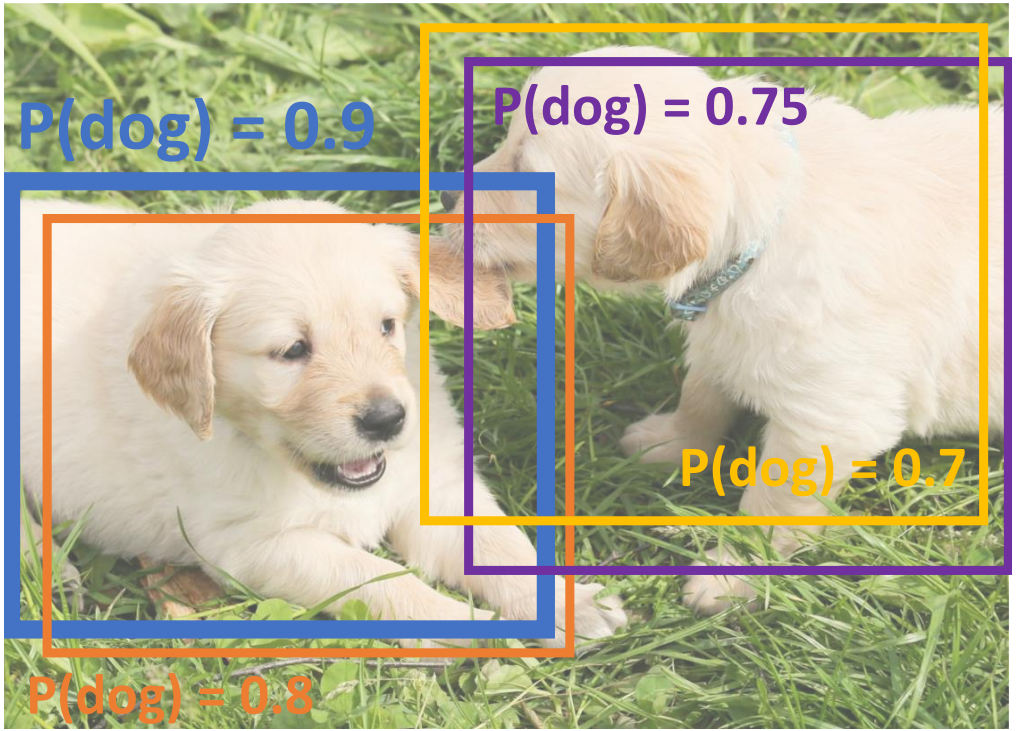
Duplicate Filter: Non-Maximum Suppression (NMS)

Problem: Object detectors often output many overlapping detections

Solution: Post-process raw detections using **Non-Max Suppression (NMS):**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.78**
IoU(■, ■) = 0.05
IoU(■, ■) = 0.07



[Puppy image is CC0 Public Domain]

[Slide credit: Justin Johnson]

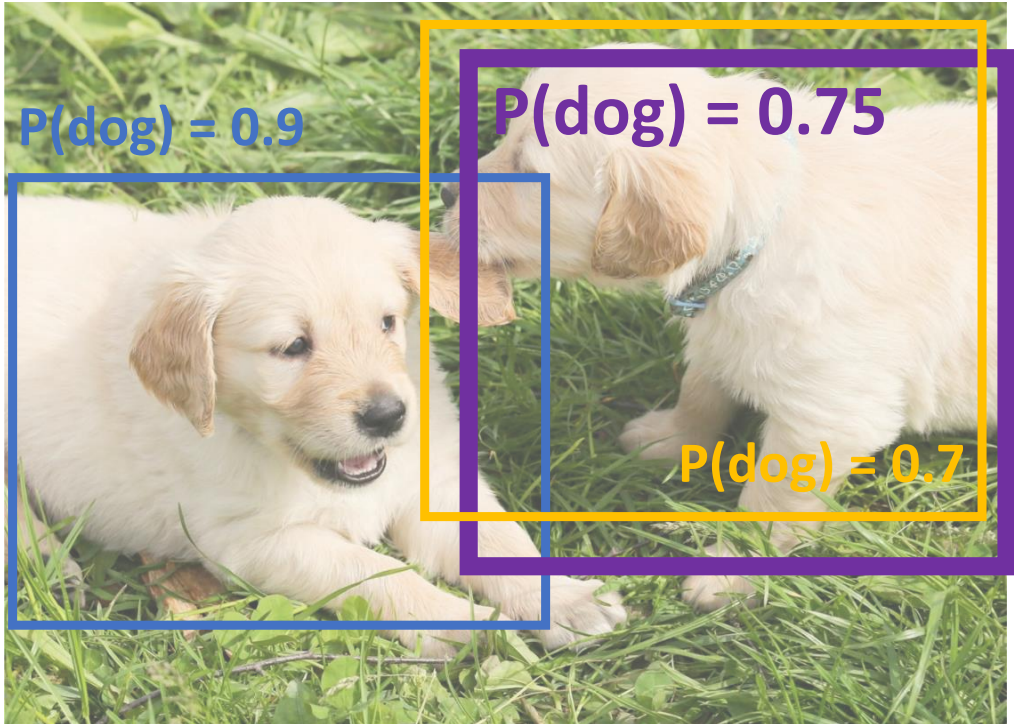
Duplicate Filter: Non-Maximum Suppression (NMS)

Problem: Object detectors often output many overlapping detections

Solution: Post-process raw detections using **Non-Max Suppression (NMS):**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.74**



[Puppy image is CC0 Public Domain]

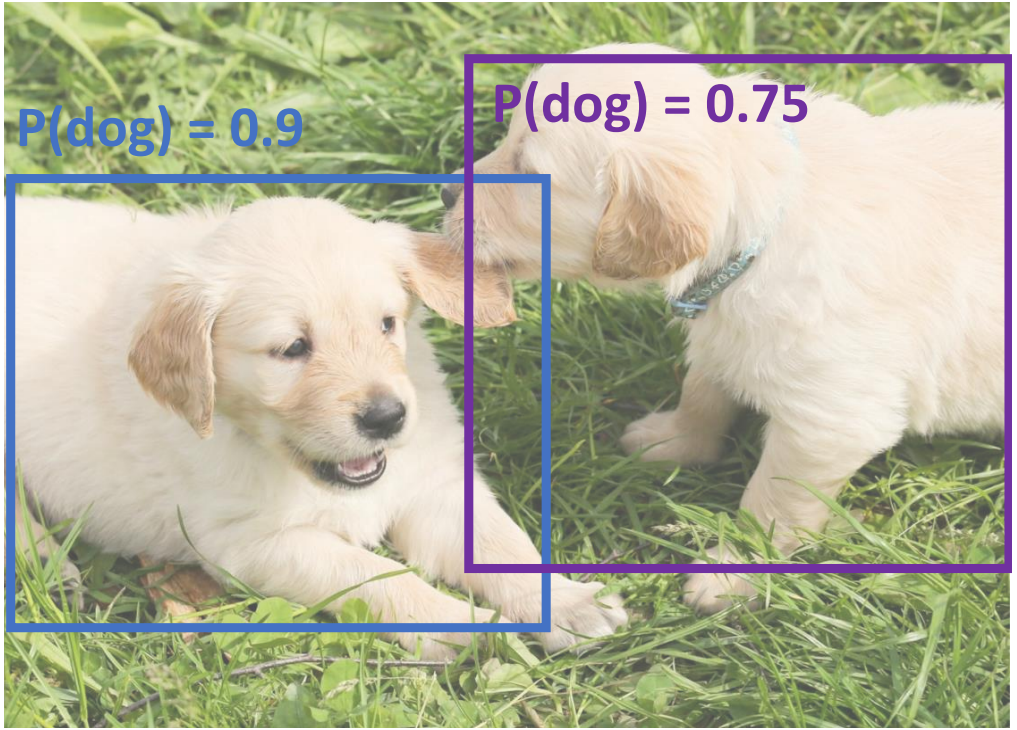
[Slide credit: Justin Johnson]

Duplicate Filter: Non-Maximum Suppression (NMS)

Problem: Object detectors often output many overlapping detections

Solution: Post-process raw detections using **Non-Max Suppression (NMS):**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1



[Puppy image is CC0 Public Domain]

[Slide credit: Justin Johnson]

Duplicate Filter: Non-Maximum Suppression (NMS)

Problem: Object detectors often output many overlapping detections

Solution: Post-process raw detections using **Non-Max Suppression (NMS):**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

[Crowd image is for commercial use under the Pixabay license]

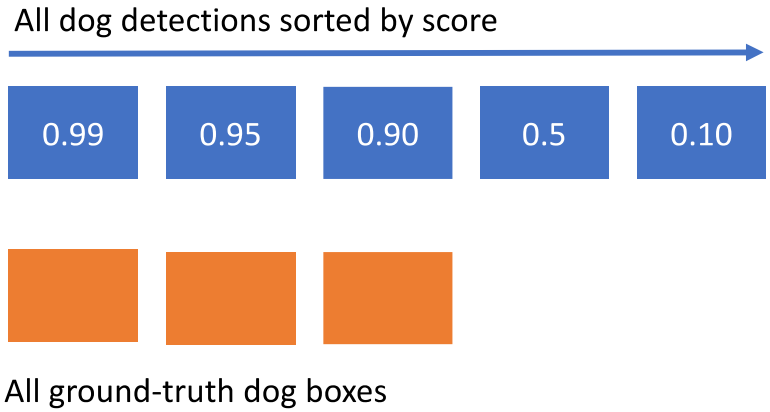
Problem: NMS may eliminate ”good” boxes when objects are highly overlapping. There is not yet a good solution.

[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)

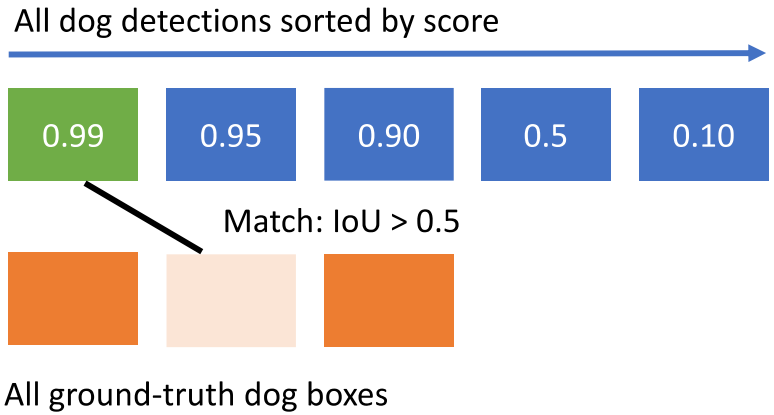


[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative

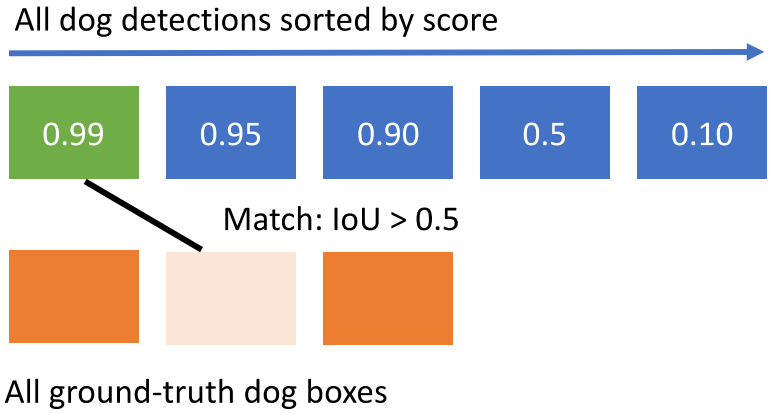


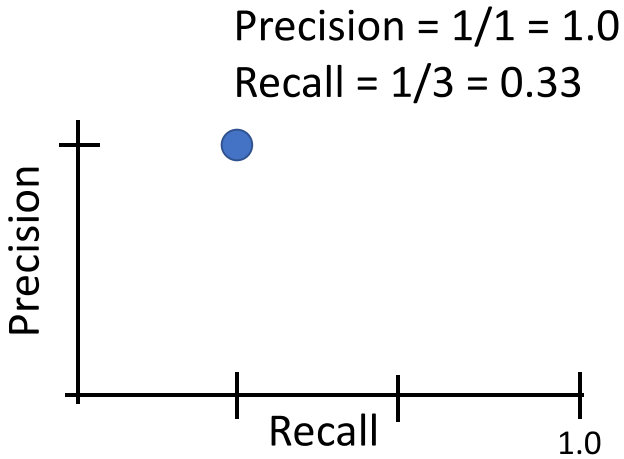
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve



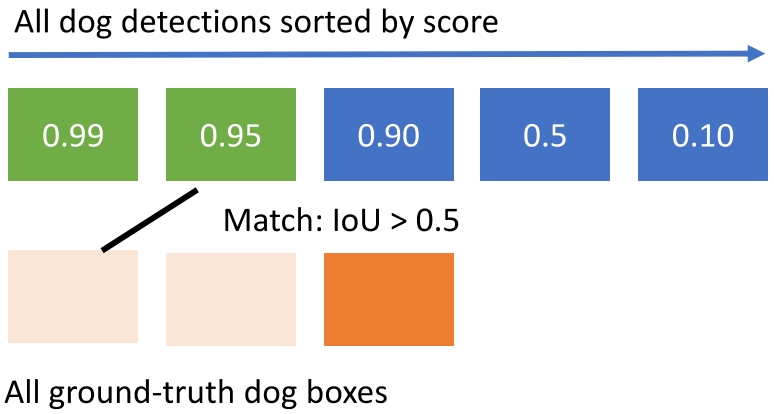


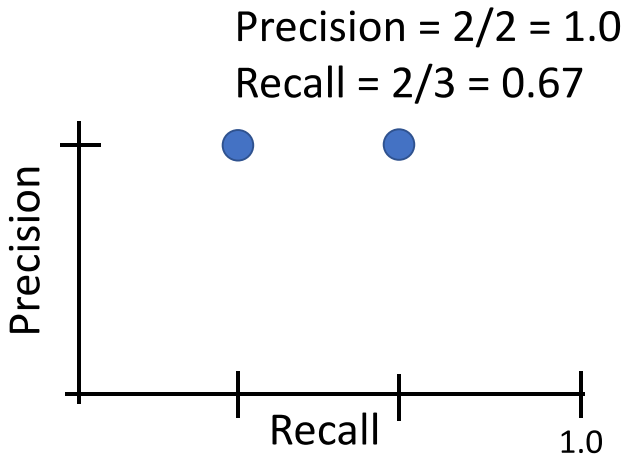
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve



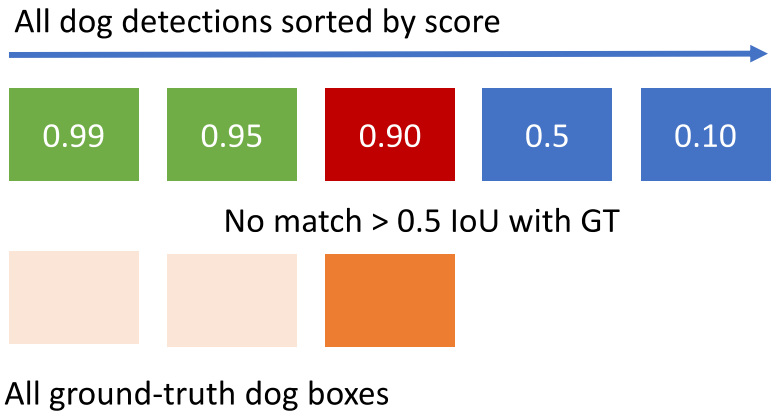


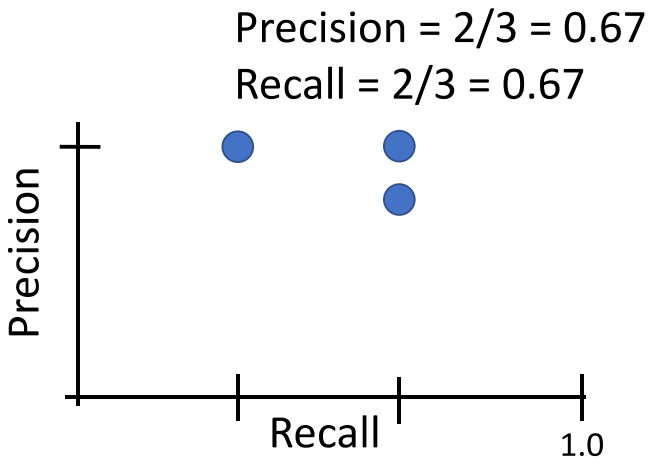
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve



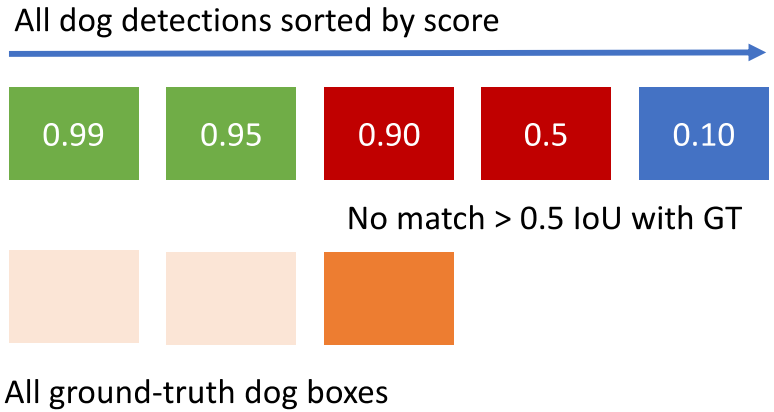


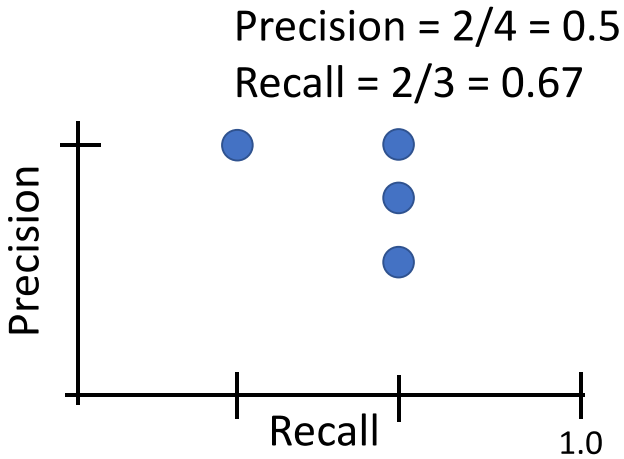
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve



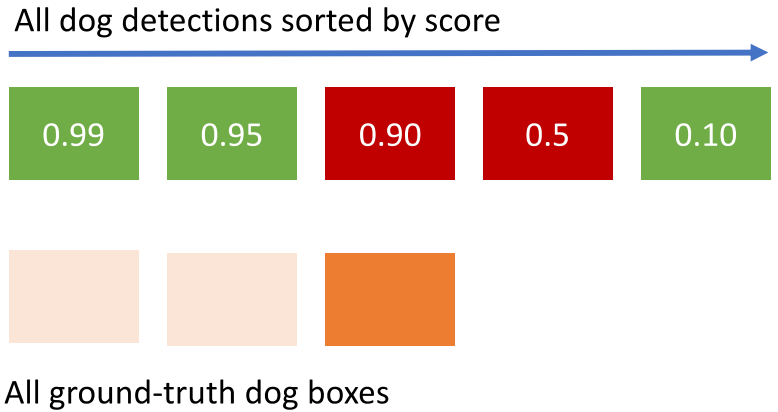


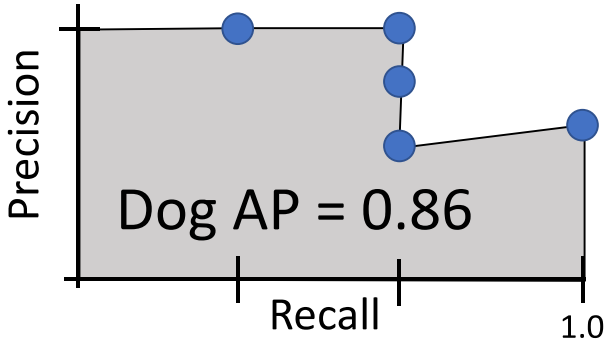
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve



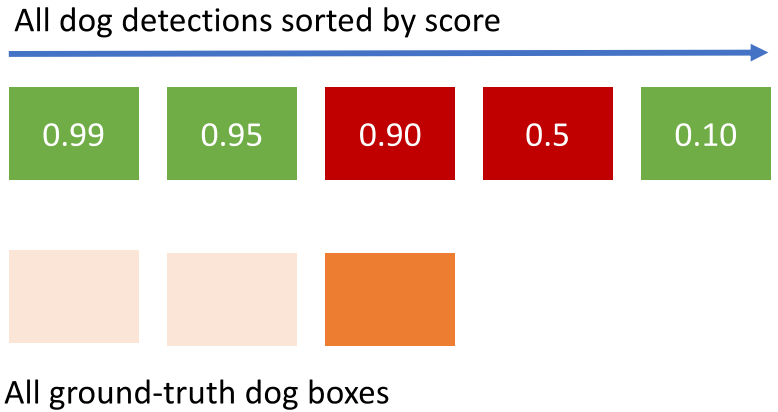


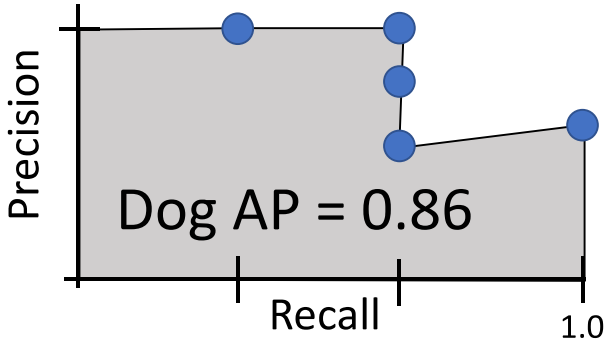
[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve





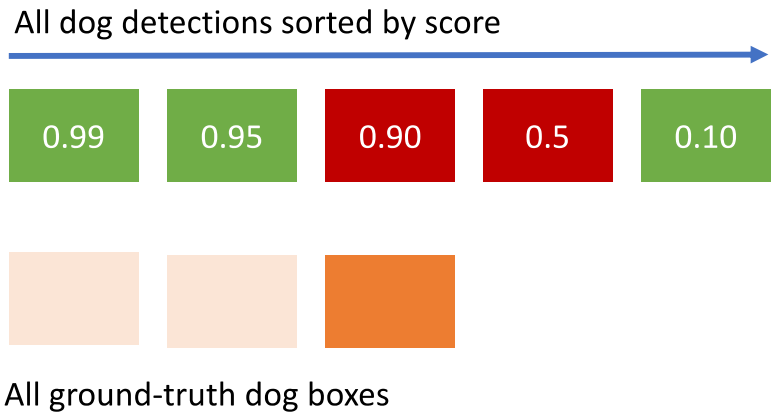
[Slide credit: Justin Johnson]

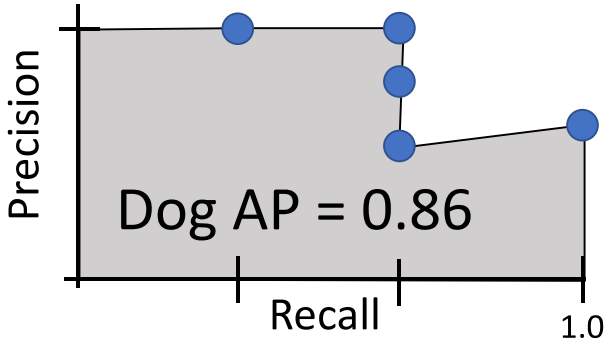
Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve

How to get AP = 1.0: Hit all GT boxes with IoU > 0.5, and have no "false positive" detections ranked above any "true positives"





[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve
3. Mean Average Precision (mAP) = average of AP for each category

Car AP = 0.65
Cat AP = 0.80
Dog AP = 0.86

mAP@0.5 = 0.77

[Slide credit: Justin Johnson]

Evaluating Object Detectors

Mean Average Precision (mAP):

1. Run object detector on all test images (with NMS)
2. For each category, compute Average Precision (AP) = area under Precision vs Recall Curve
    1. For each detection (highest score to lowest score)
        1. If it matches some GT box with IoU > 0.5, mark it as positive and eliminate the GT
        2. Otherwise mark it as negative
        3. Plot a point on PR Curve
3. Mean Average Precision (mAP) = average of AP for each category
4. For "COCO mAP": Compute mAP@thresh for each IoU threshold (0.5, 0.55, 0.6, …, 0.95) and take average

mAP@0.5 = 0.77
mAP@0.55 = 0.71
mAP@0.60 = 0.65
…
mAP@0.95 = 0.2

COCO mAP = 0.4

[Slide credit: Justin Johnson]

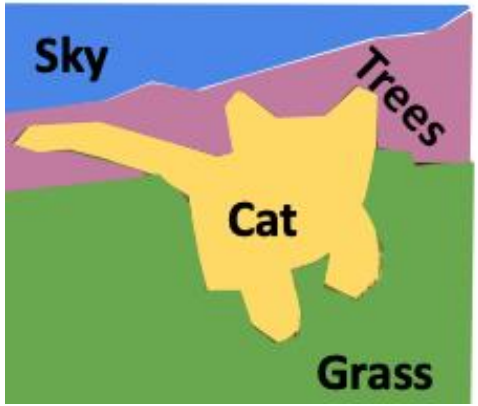
Today's Agenda

- Image Retrieval
- Object Detection
- **Image Segmentation**

Image Segmentation

Goal: Partition the image into semantically-meaningful or perceptually-similar regions.
(In contrast to object detection: all pixels are labeled!)

That is: Label each pixel in the image with a category label



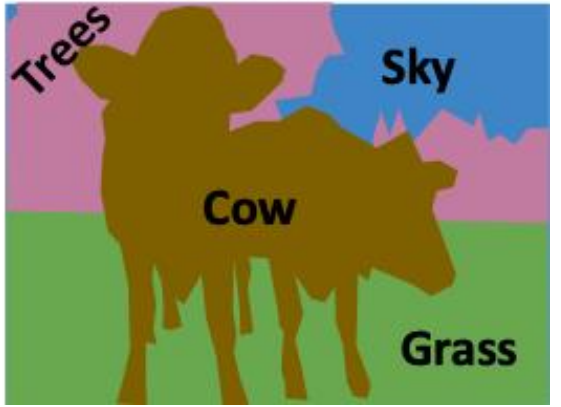
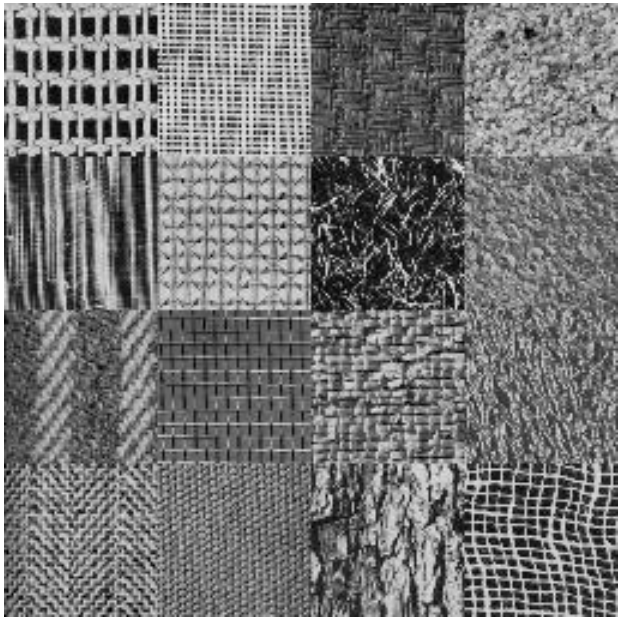
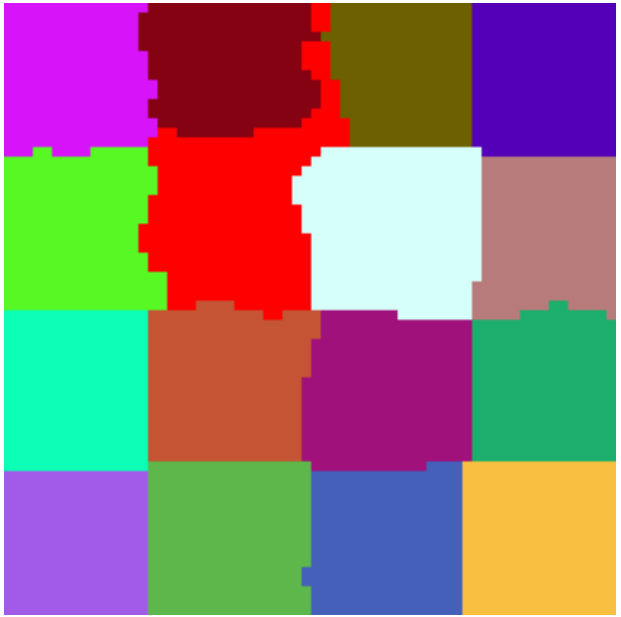


Image Segmentation

Example: Color space clustering to reduce color palette.

Texture Segmentation

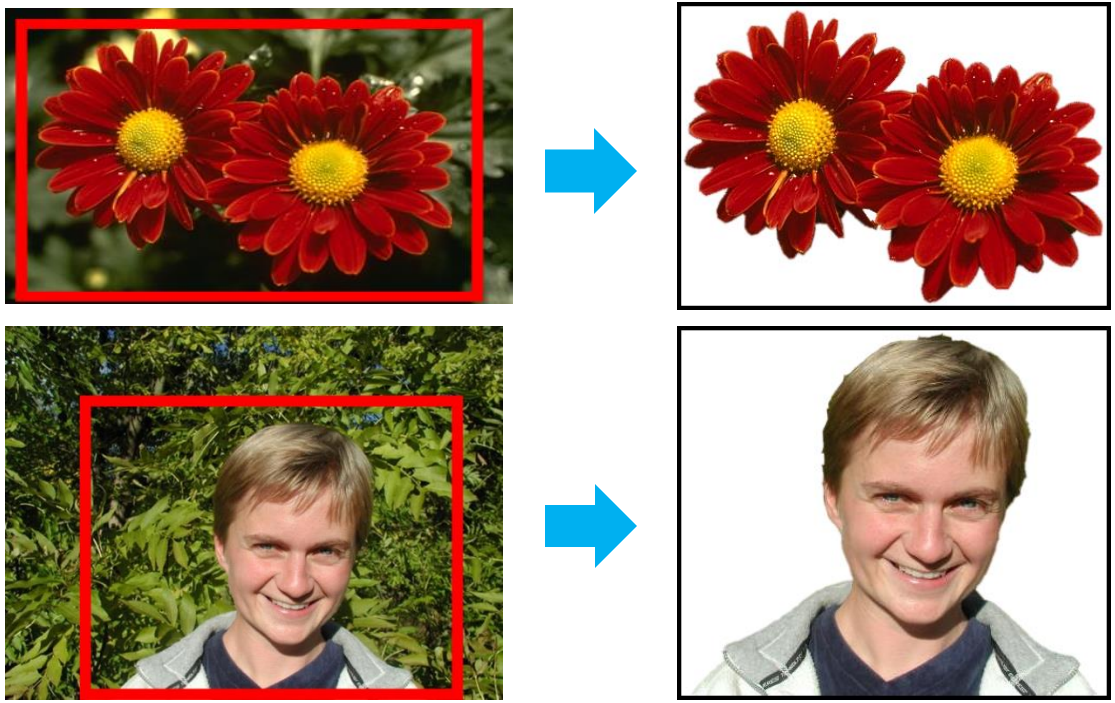




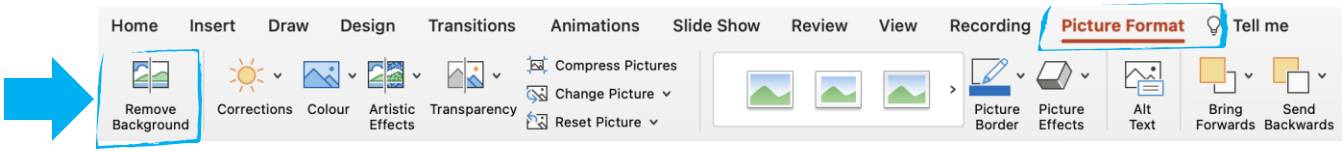
Courtesy Uni Bonn

Image Segmentation

- “Grabcut”
  Graphcut-based
  Image segmentation



- Integrated into MS Office
  e.g. Powerpoint

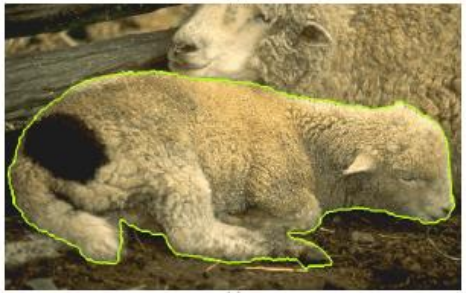


[Rother et al. “Graphcut” TOG 2004]

Image Segmentation

(a)

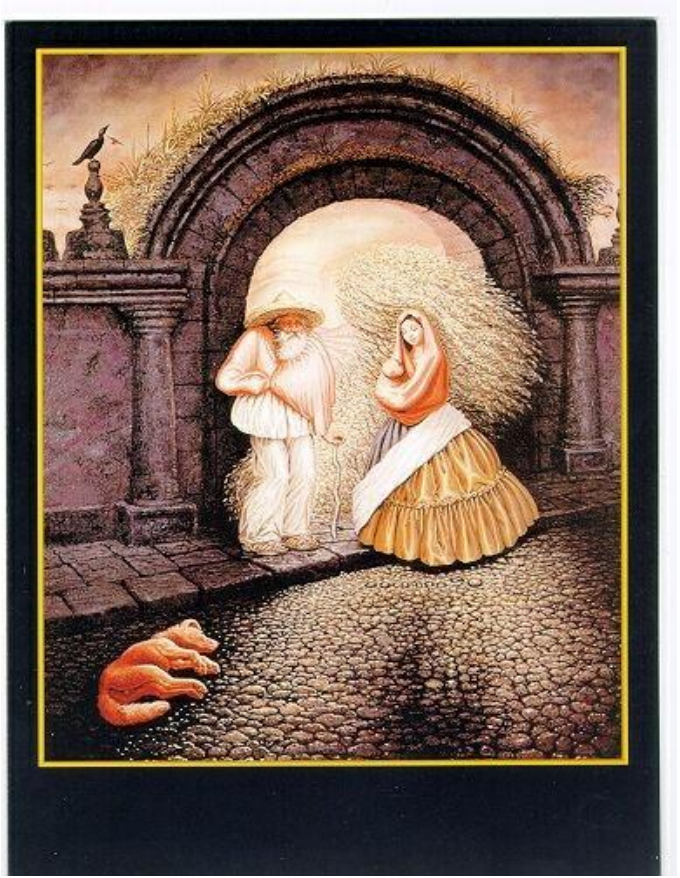
(b)



(c)

Segment first, or recognise first? (what is the context?)

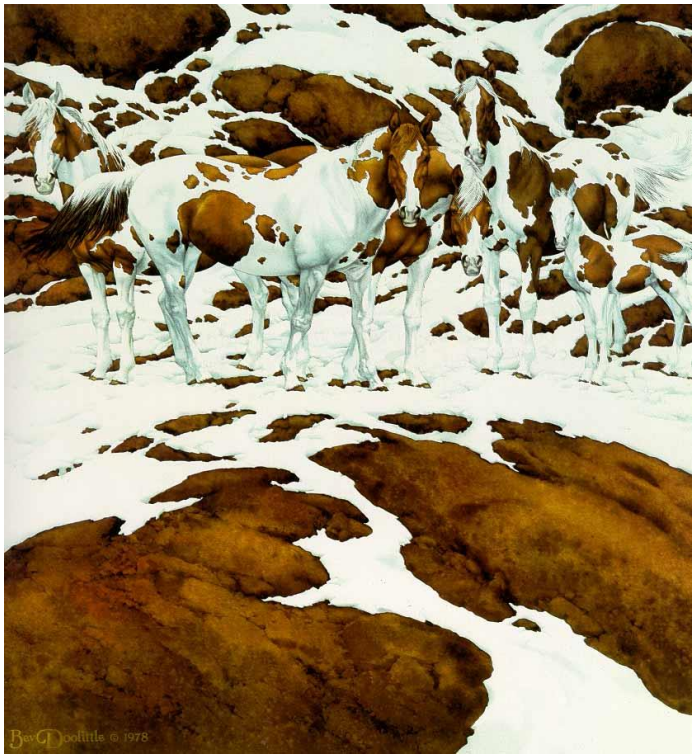
A portrait of
an old man, or



A couple with
a dog on the street

Segment first, or recognise first? (what is the context?)

Snow and rocks,
or



A group of horses
In snow and rocks

Segmentation for Recognition

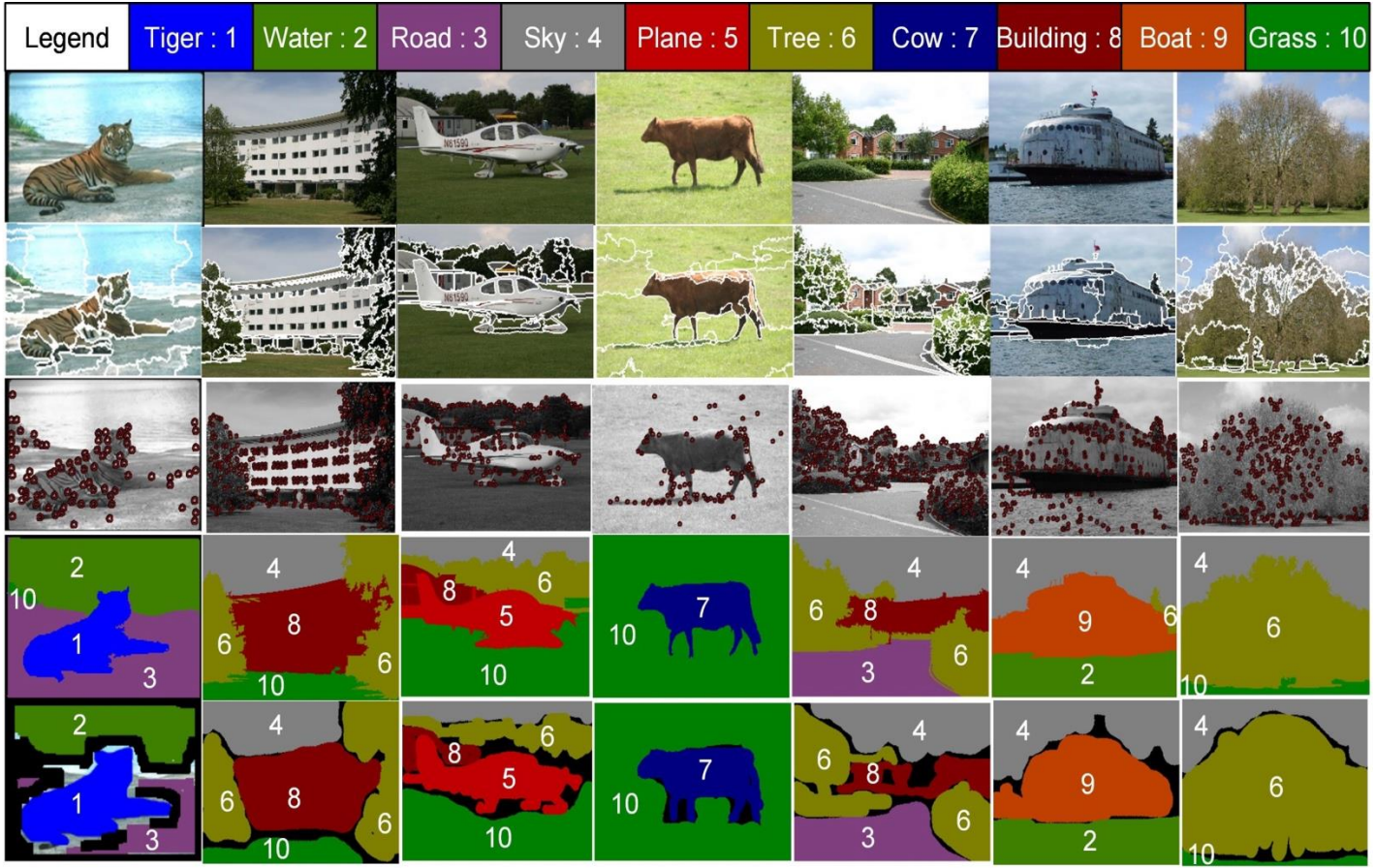
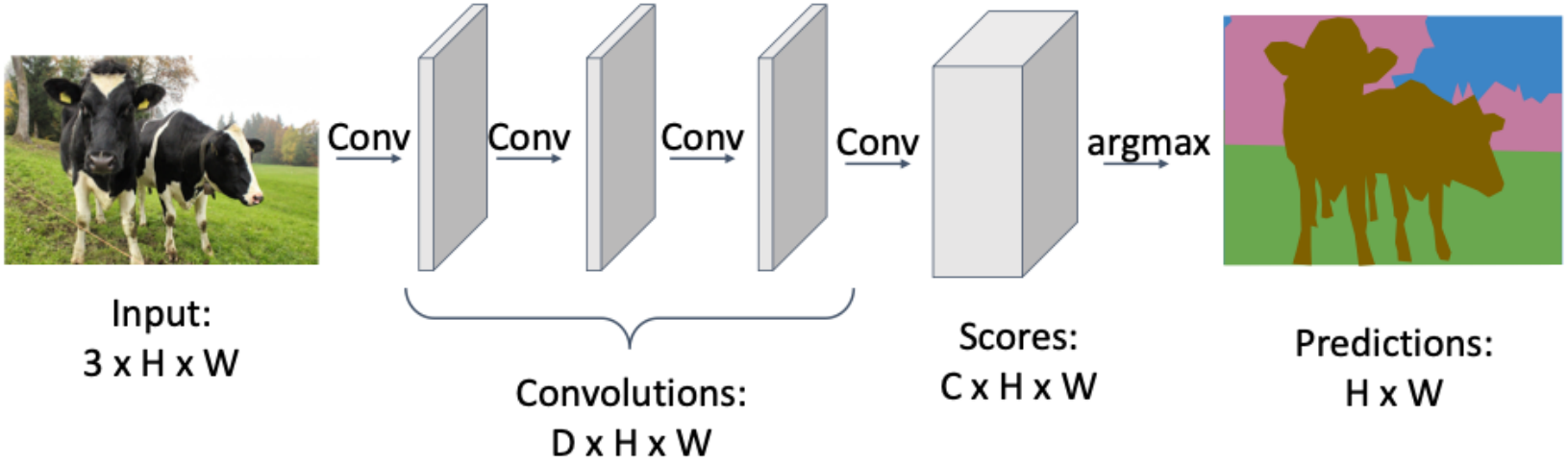


Image Segmentation

Design a network with multiple convolutional layers to make predictions for pixels all at once!



Loss function: Per-Pixel cross-entropy

[Slide credit: Justin Johnson] [Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015]

Today's Agenda

- Image Retrieval
- Object Detection
- **Image Segmentation**
  - **K-Means**

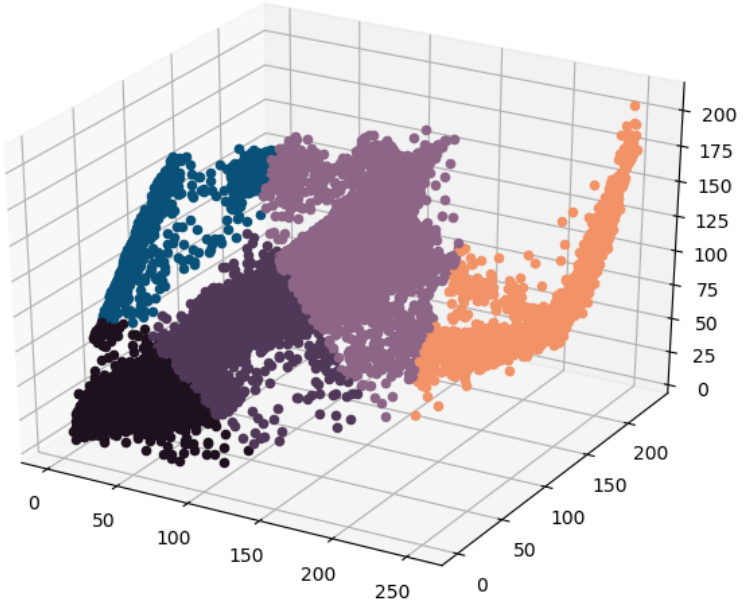
Segmentation as Clustering

- Low- and mid-level visual cues for segmentation.
- Partition the image pixels into groups with perceptually homogeneous or similar pixel properties (e.g., intensity, colour, texture, or similar spatial location).
- Grouping pixels into perceptually homogeneous regions.

Segmentation as Clustering

- Cluster similar pixels using color features

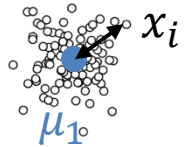
Cluster in color space (3D) or in color + position space (5D)



K-Means

Goal: Minimize the distance between N given data points and K assigned cluster centers:

$$\text{minimize} \quad \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$



K-Means Algorithm

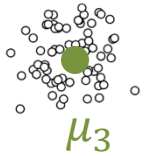
1. **Initialize** K cluster means $\mu_1, \dots, \mu_K$
2. **Reassign Points:** For i $= 1, \dots, N$ assign each point x_n to the closest cluster

   $$z_n = \text{argmin}_{k \in \{1,\dots,K\}} \|x_i - \mu_k\|^2$$

3. **Update Centroids:** Suppose $C_k = \{x_i : z_i = k\}$. Recompute the means as

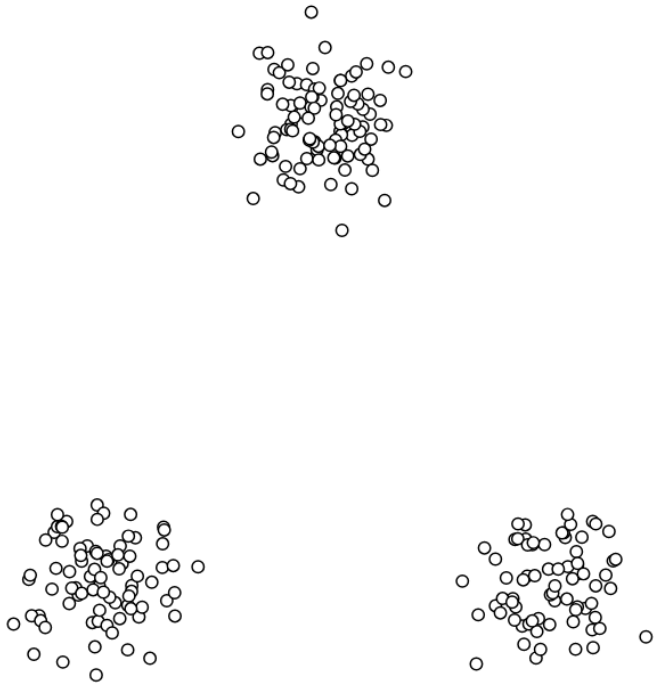
   $$\mu_k = \text{mean}(C_k), \quad k = 1, \dots, K$$

4. **Repeat:** Go to step 2 if not yet converged



K-Means

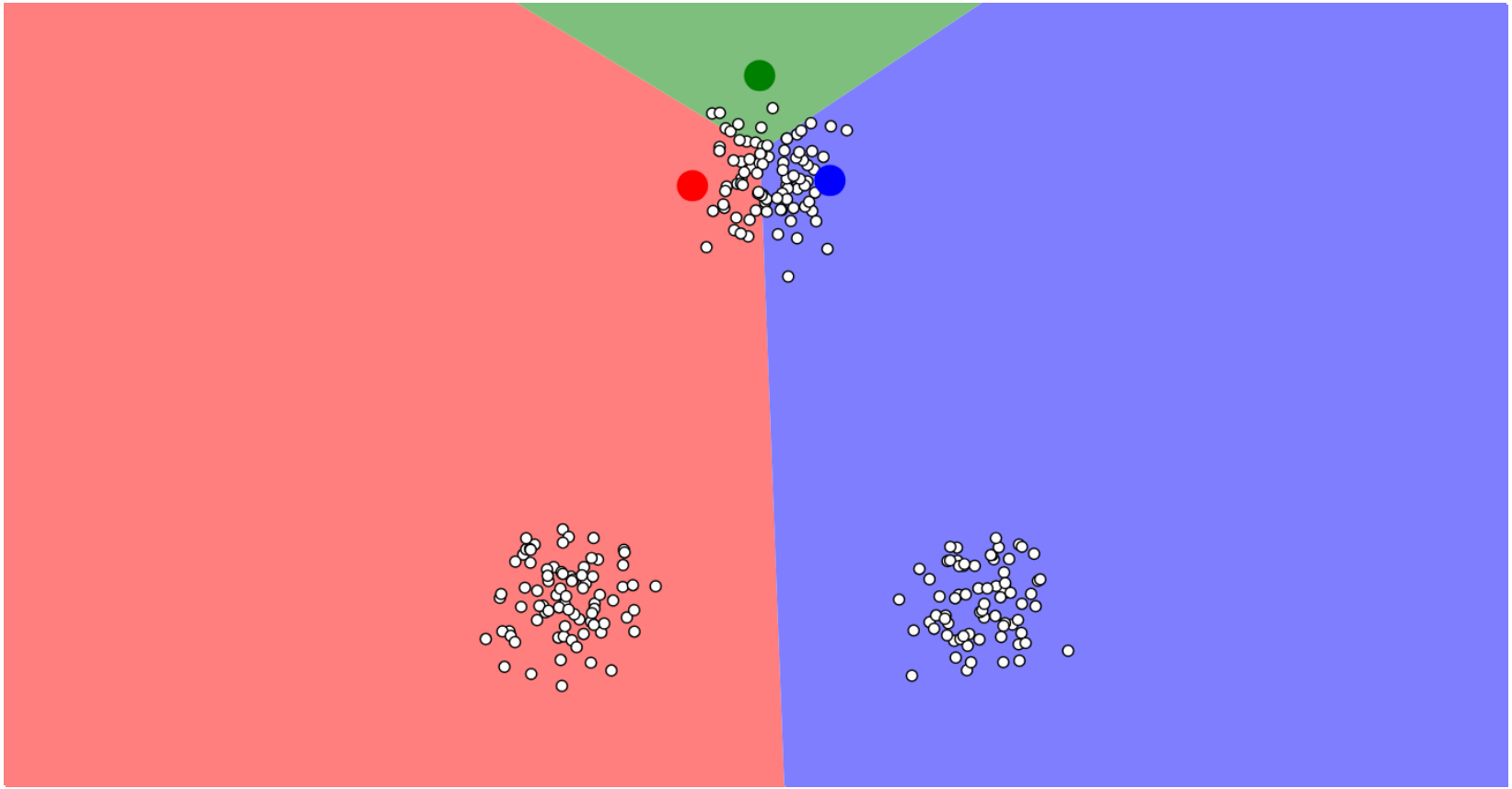
Next step:
Initialize
Means



[Source: Online Demo]

K-Means

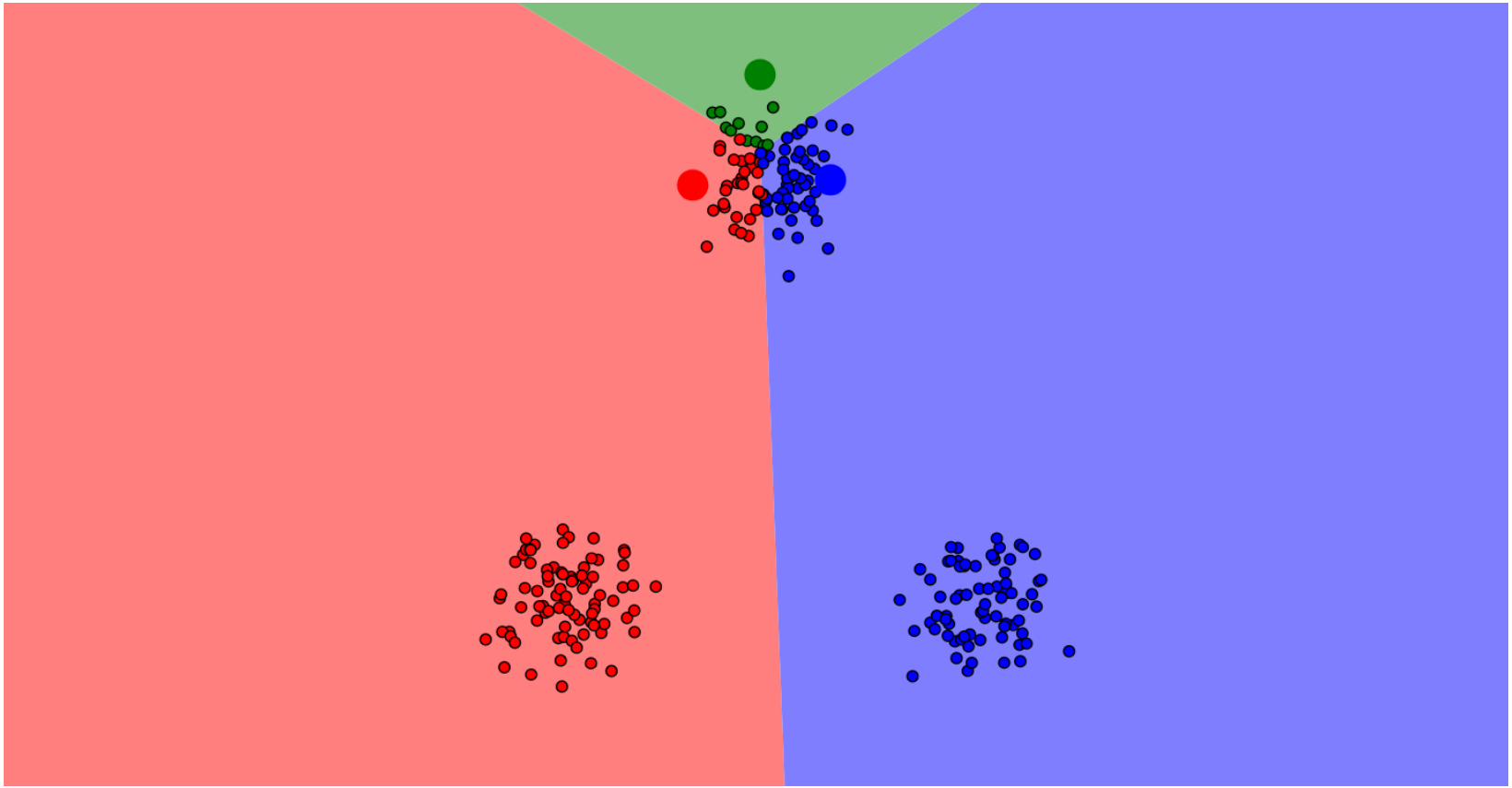
Next step:
Reassign Points



[Source: Online Demo]

K-Means

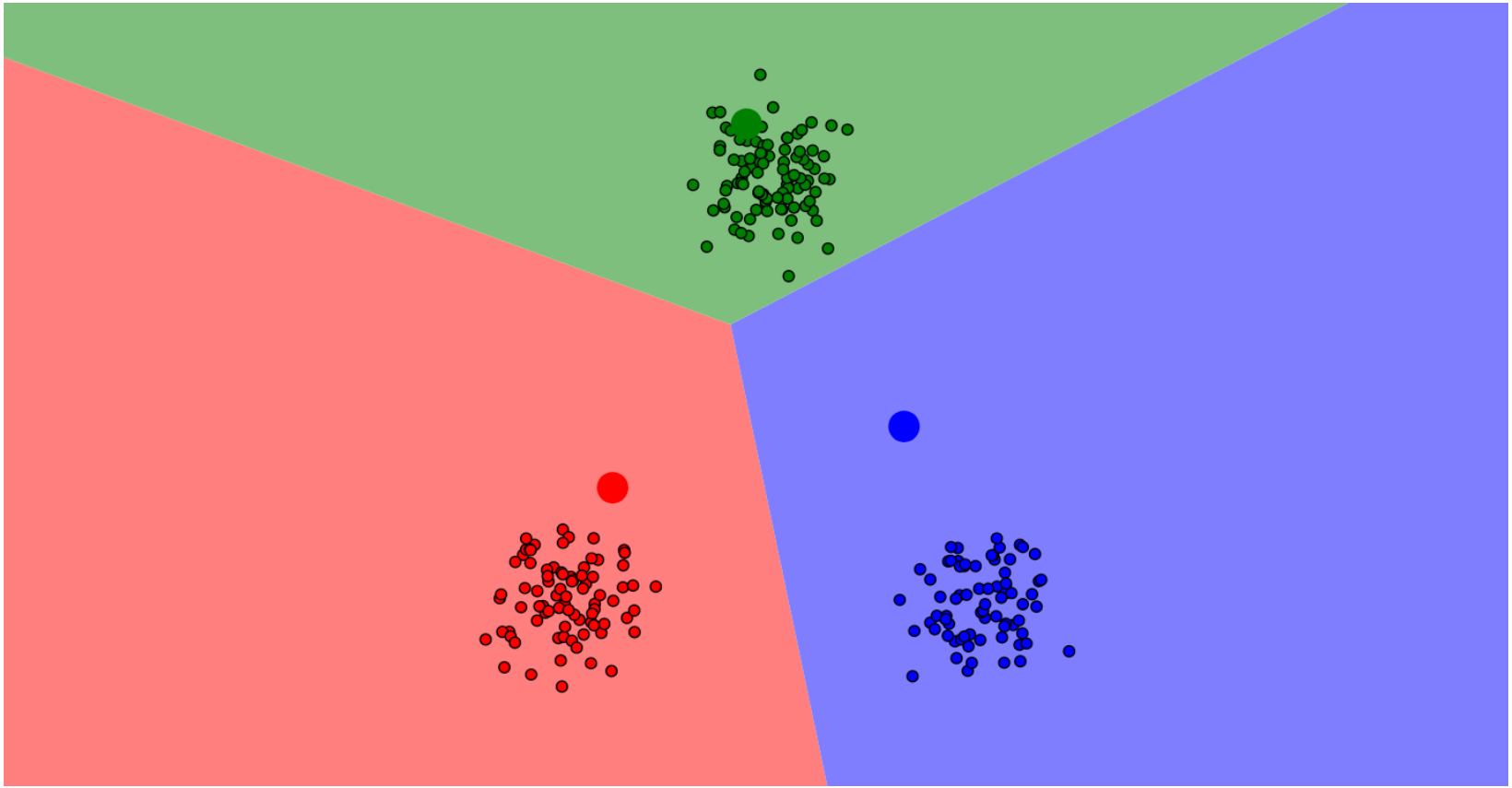
Next step:
Update Centroids



[Source: Online Demo]

K-Means

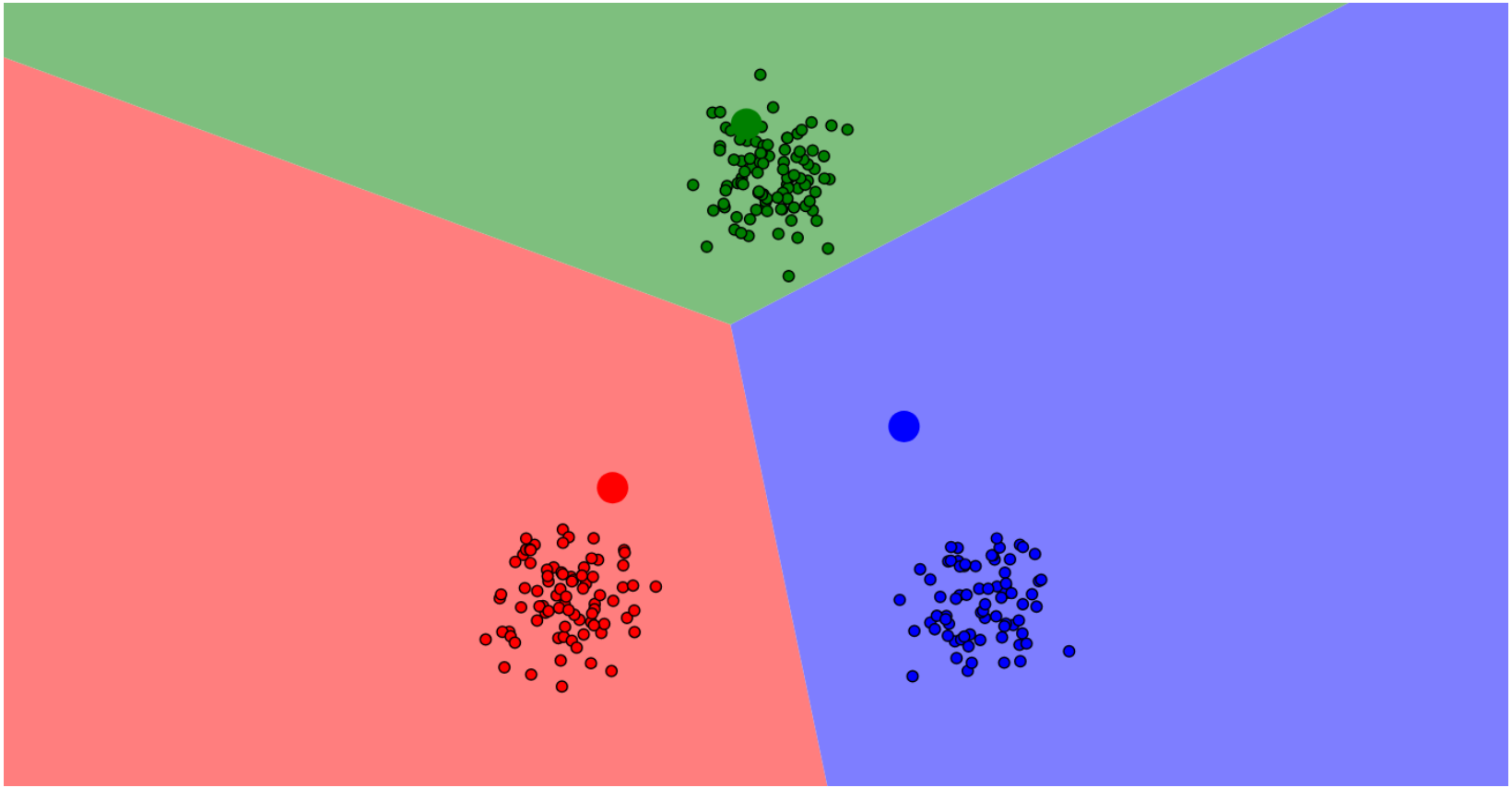
Next step:
Reassign Points



[Source: Online Demo]

K-Means

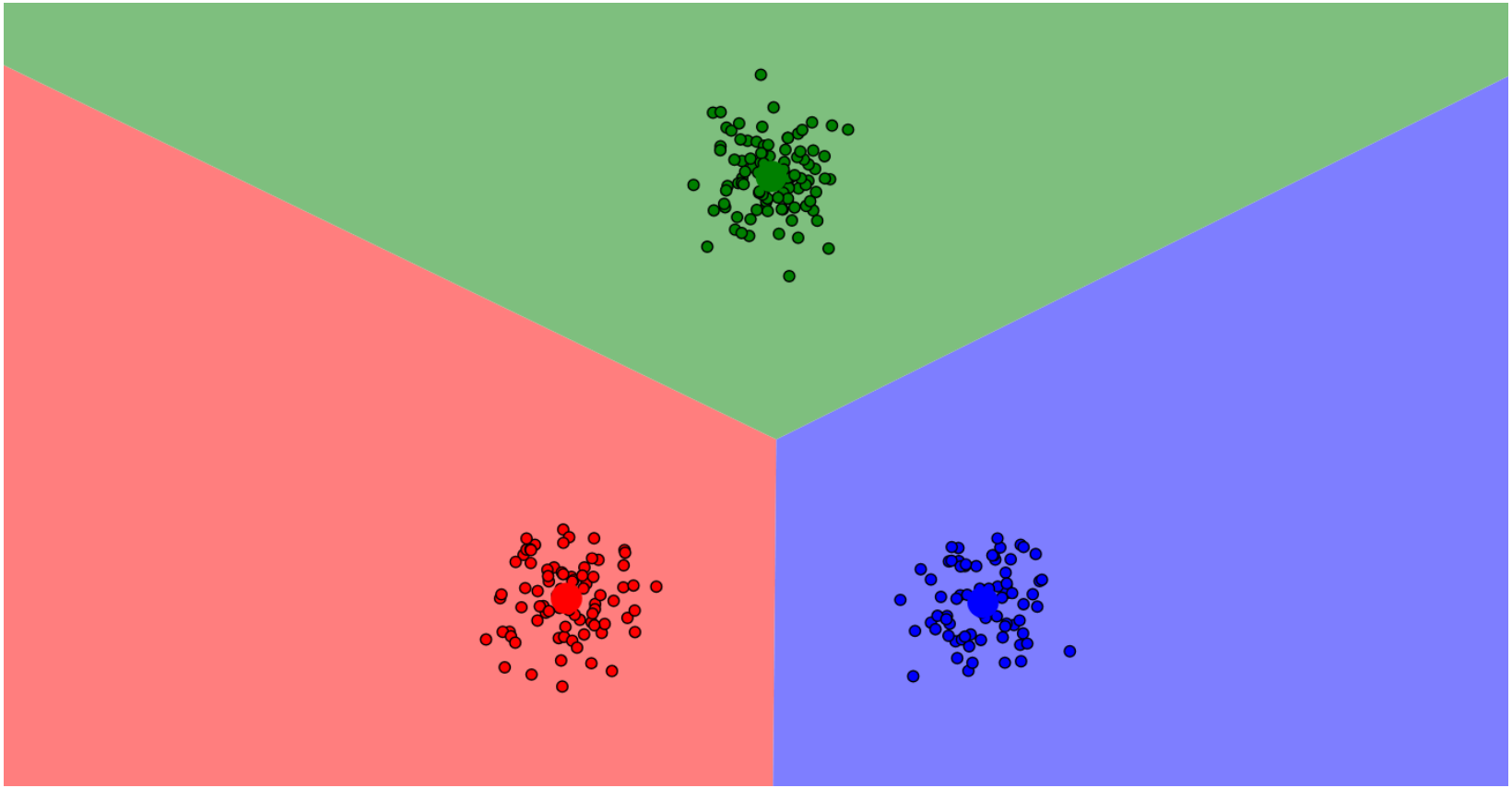
Next step:
Update Centroids



[Source: Online Demo]

K-Means

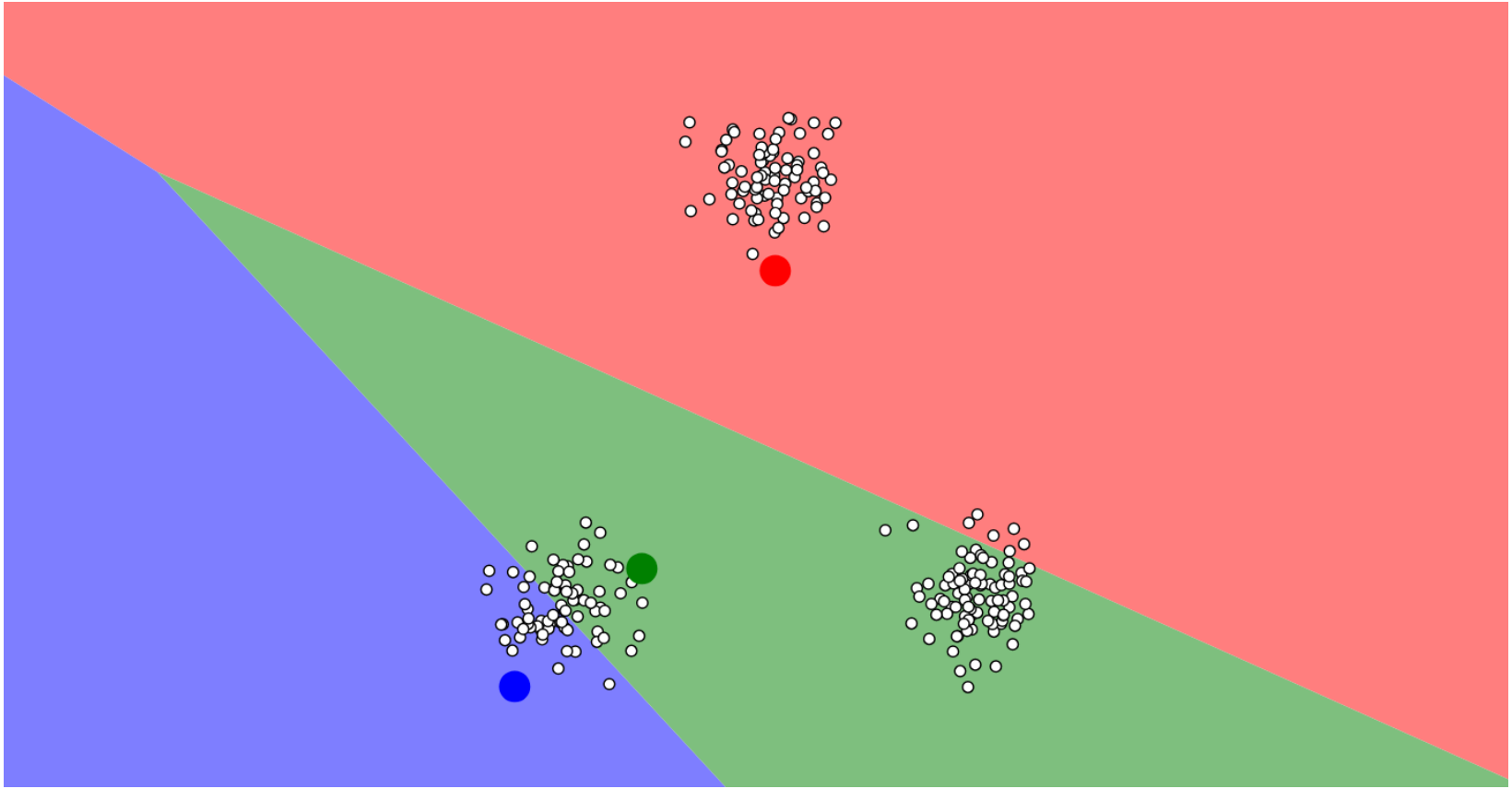
Next step:
End



[Source: Online Demo]

K-Means

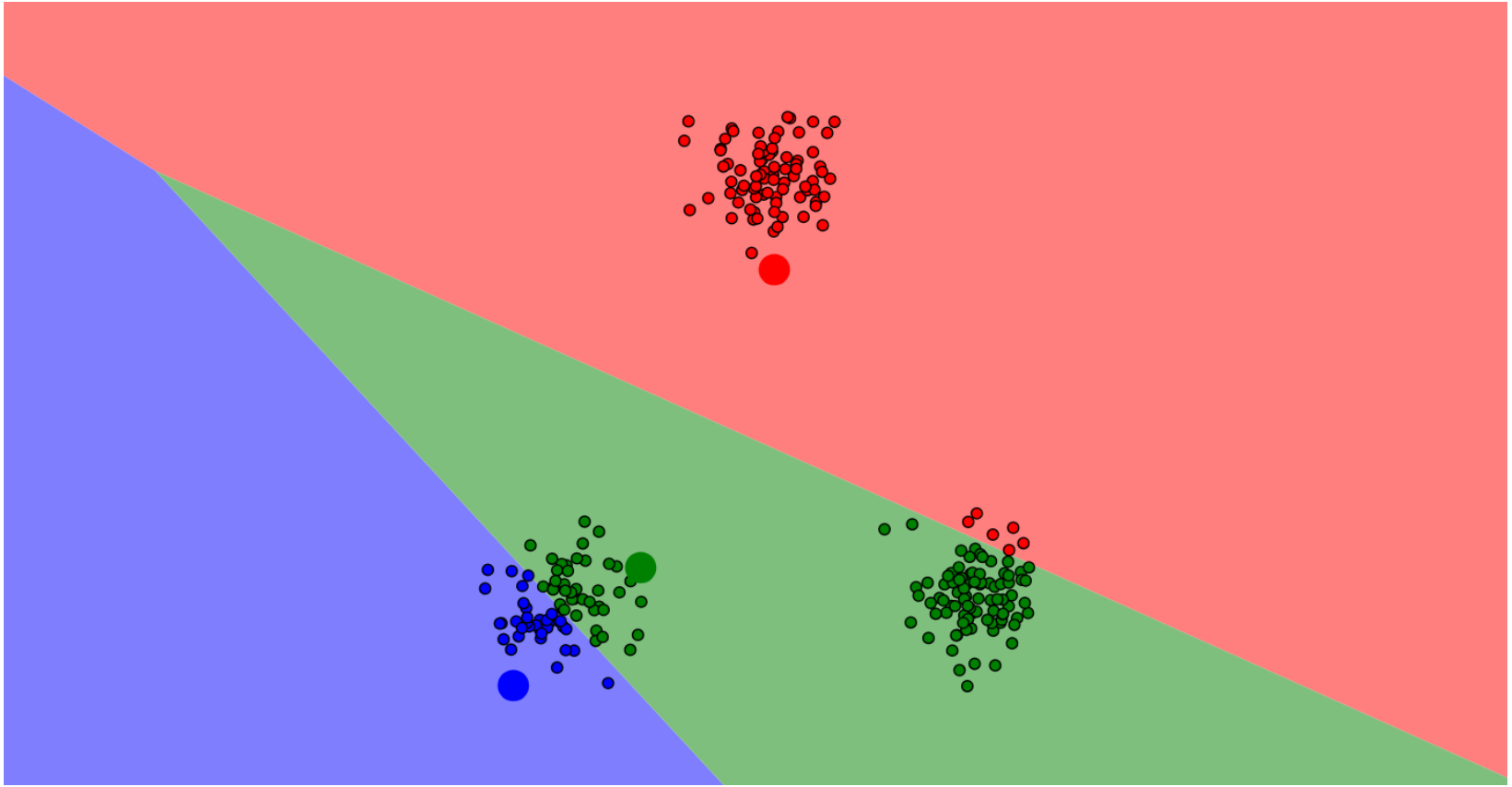
Next step:
Reassign Points



[Source: Online Demo]

K-Means

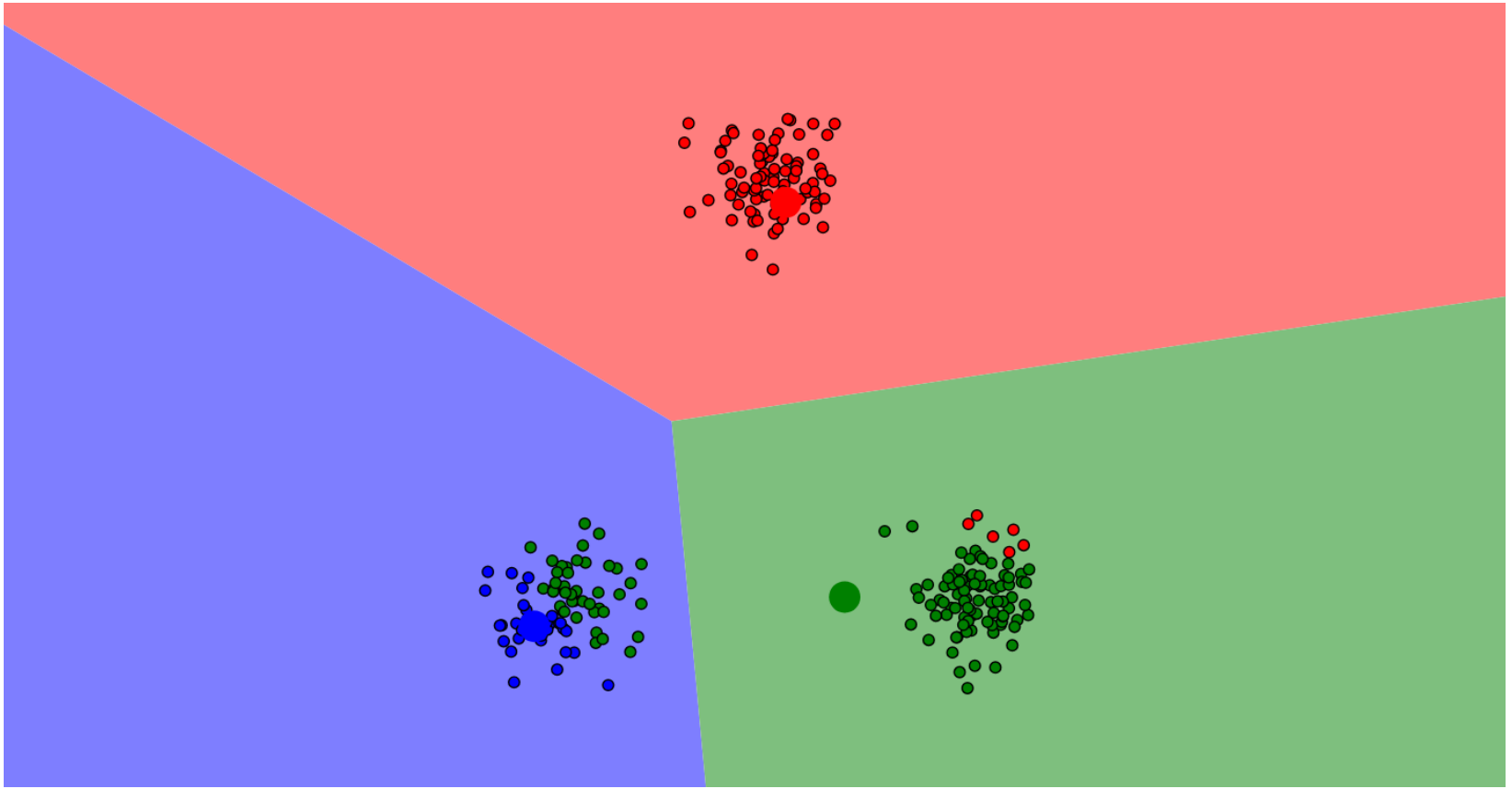
Next step:
Update Centroids



[Source: Online Demo]

K-Means

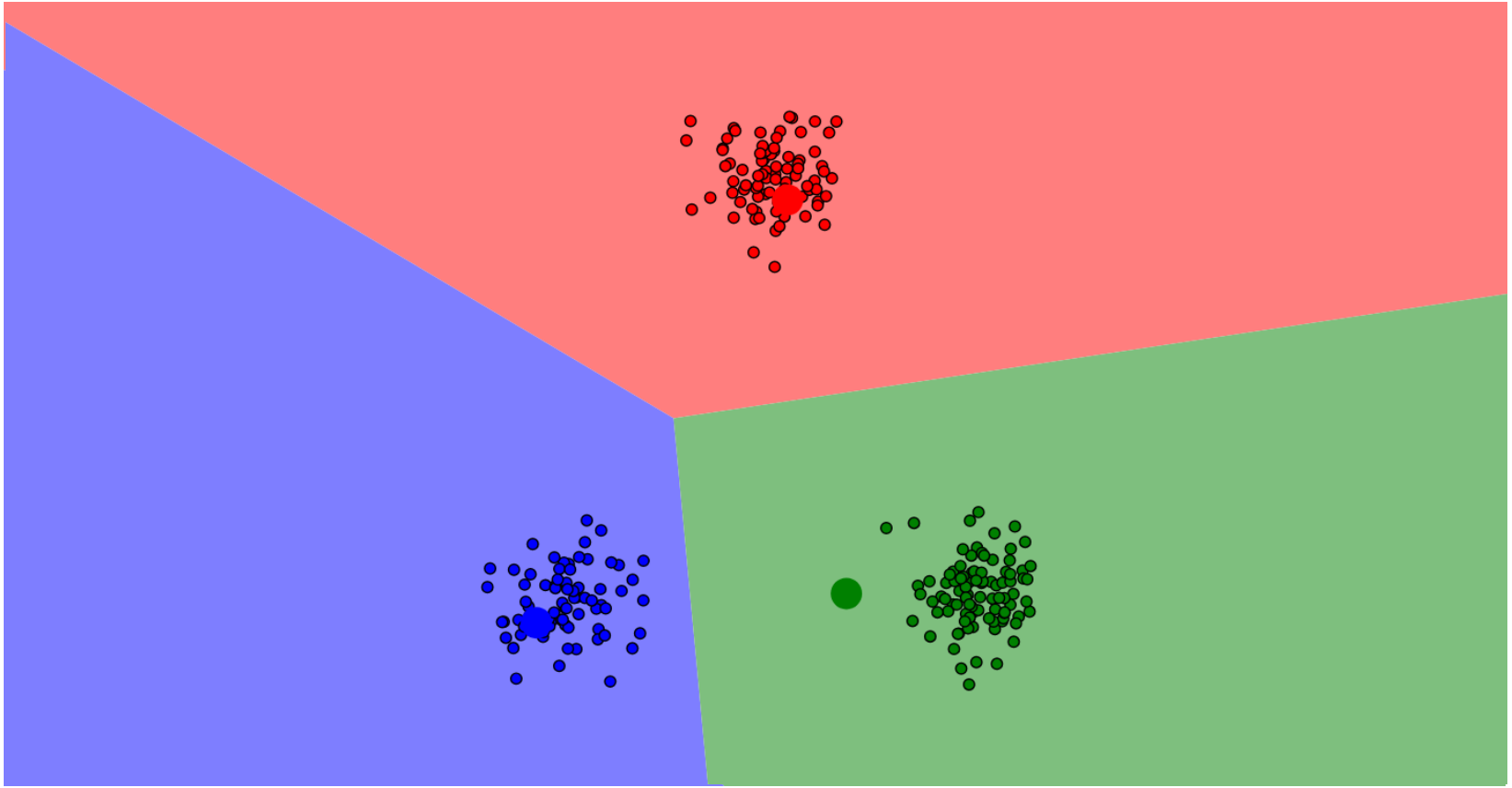
Next step:
Reassign Points



[Source: Online Demo]

K-Means

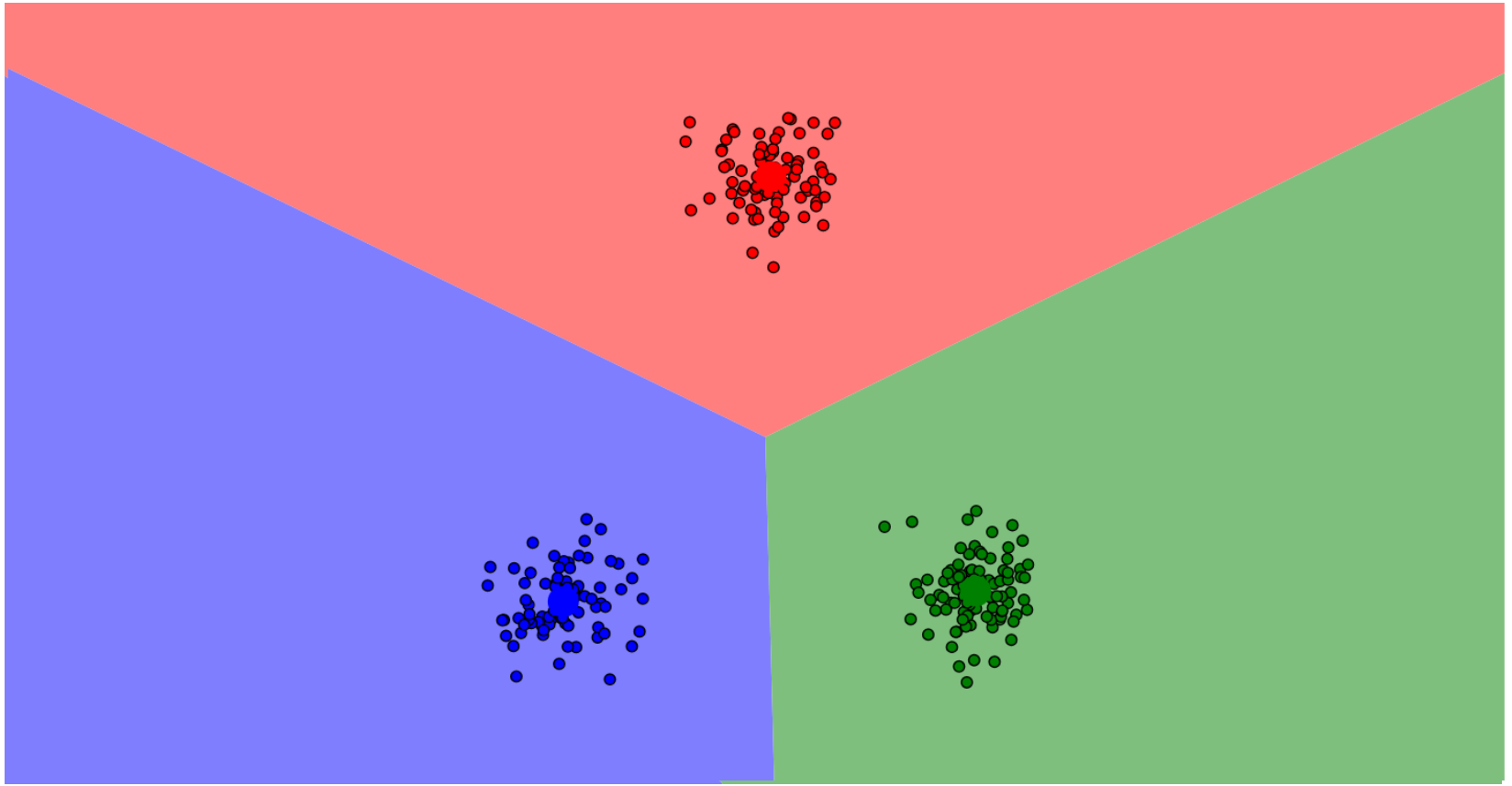
Next step:
Update Centroids



[Source: Online Demo]

K-Means

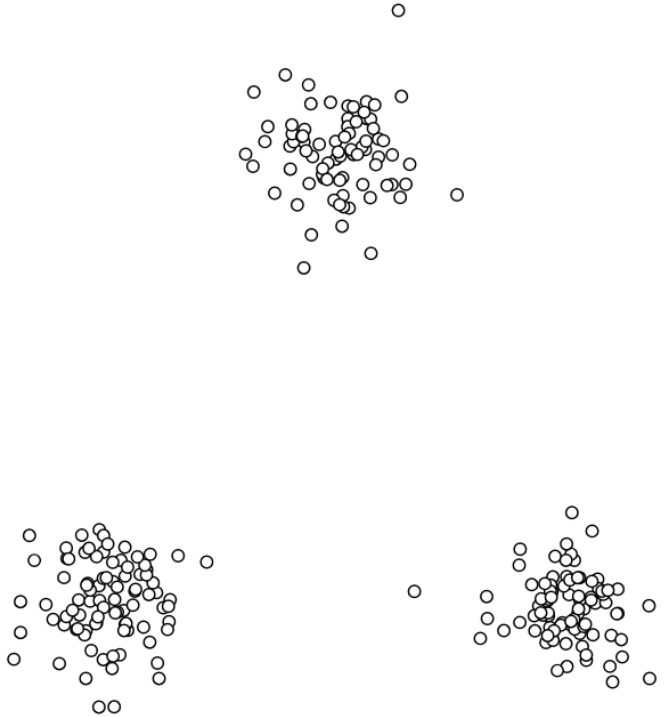
Next step:
End



[Source: Online Demo]

K-Means

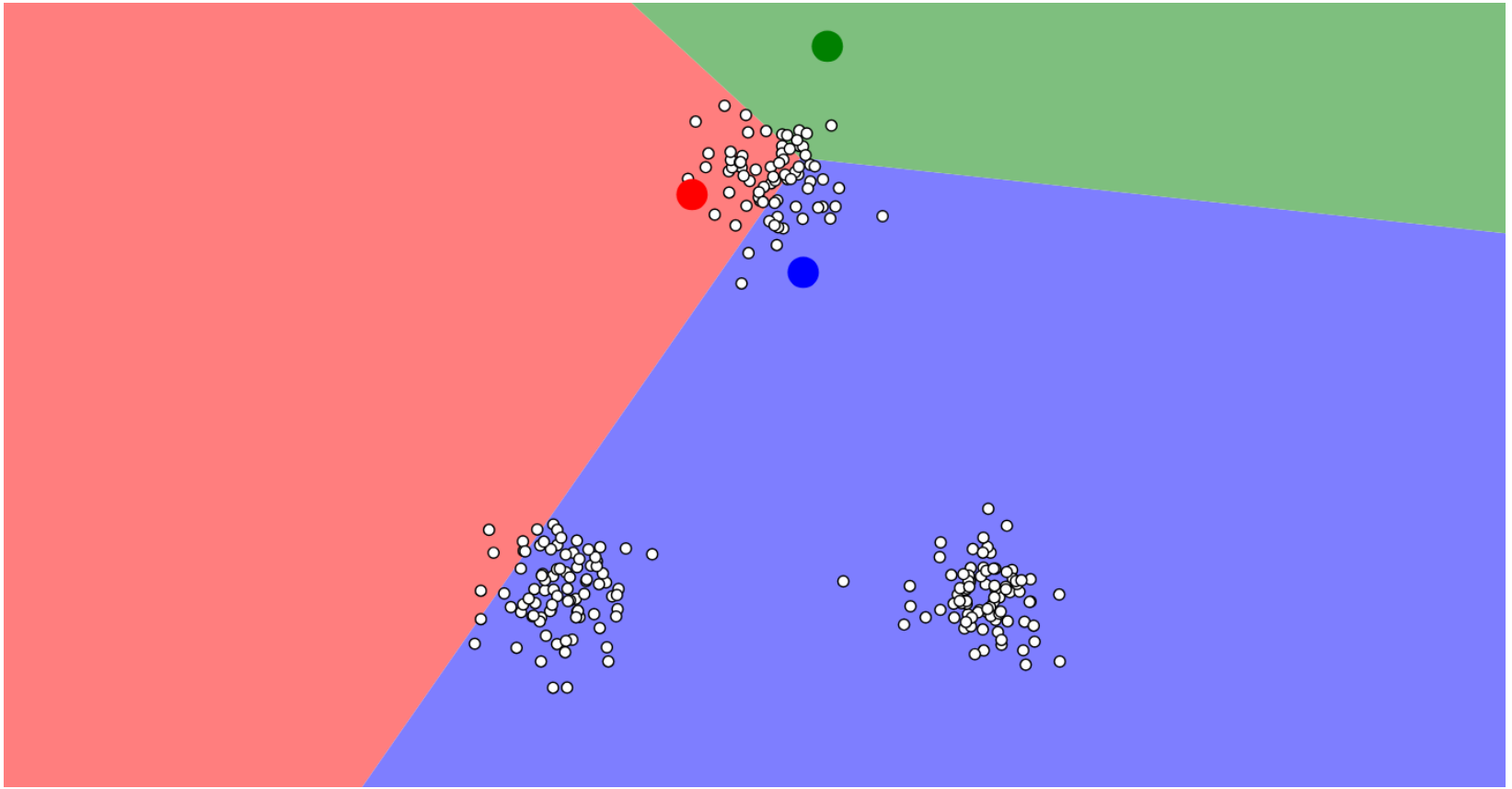
Next step:
Initialize
Means



[Source: Online Demo]

K-Means

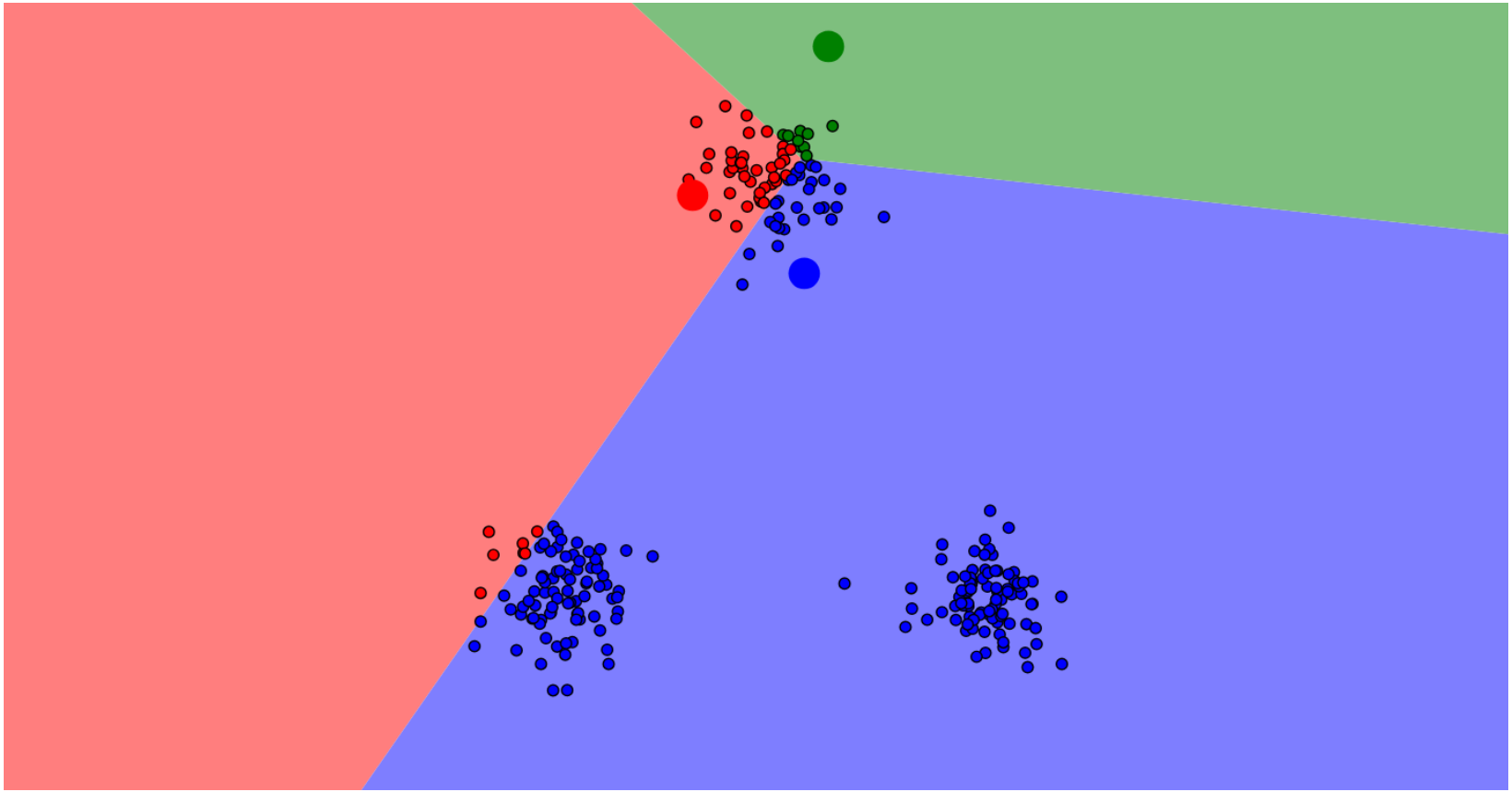
Next step:
Reassign Points



[Source: Online Demo]

K-Means

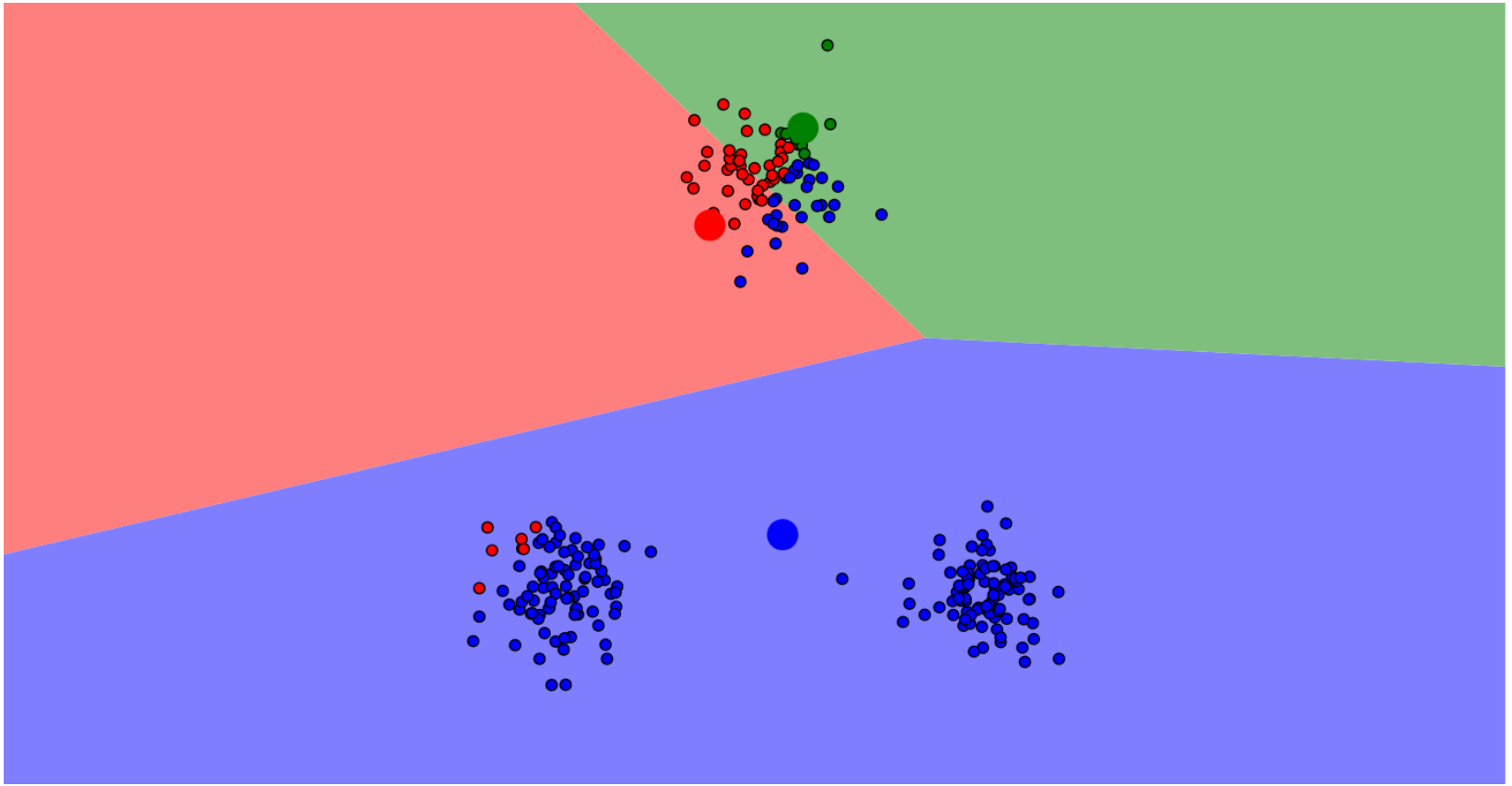
Next step:
Update Centroids



[Source: Online Demo]

K-Means

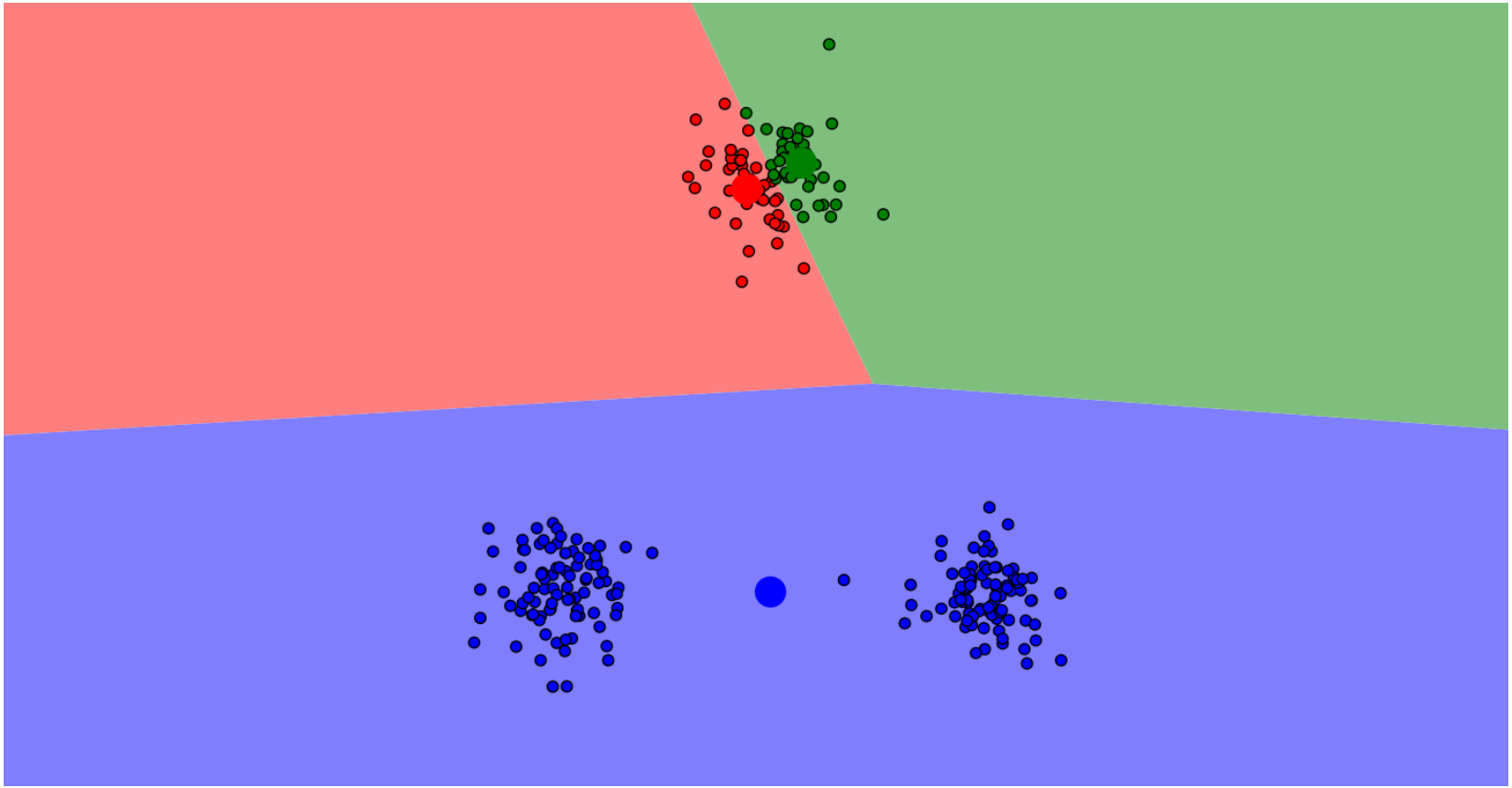
Next step:
Reassign Points



[Source: Online Demo]

K-Means

Next step:
End

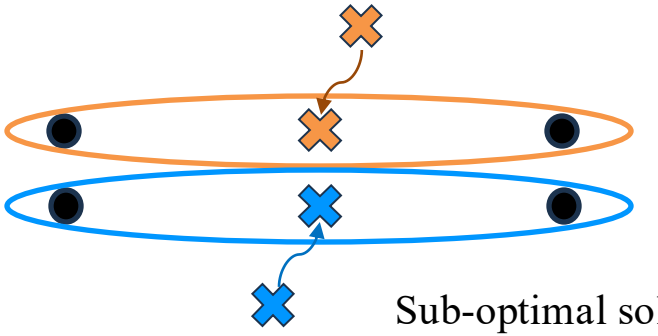


[Source: Online Demo]

K-Means

Local Minima

- Sensitive to initialization
  → sub-optimal solution



Sub-optimal solution

Optimal solution

K-Means

Pros:

- Finds cluster centers that minimize variance (good representation of data)
- Simple to implement, widespread applications.

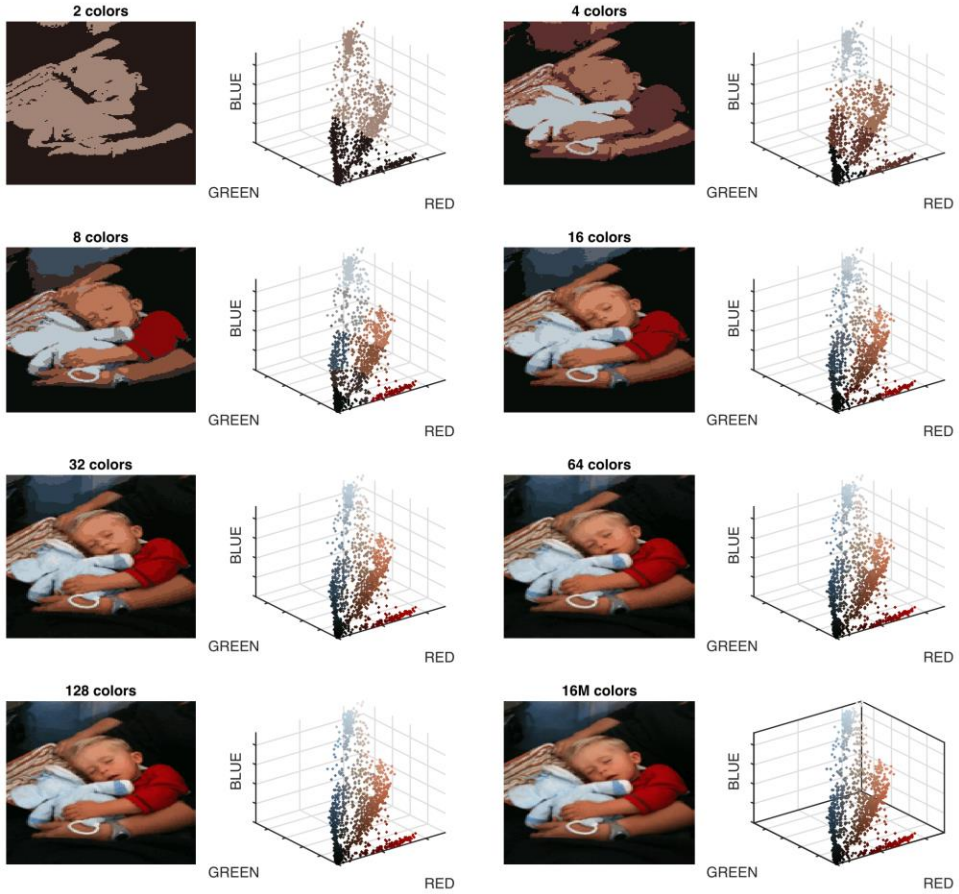
Cons:

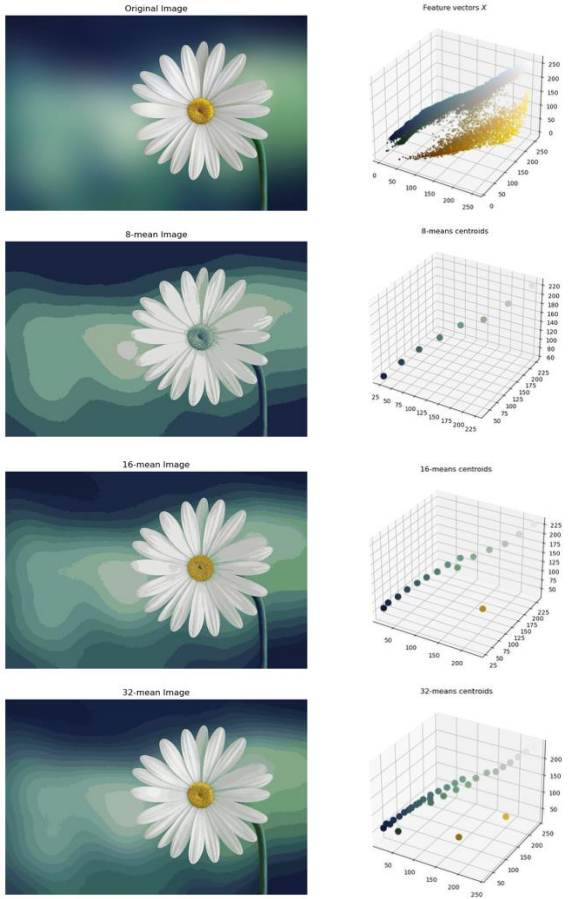
- All clusters have spherical distribution (same to all directions or isotropic)
- Prone to local minima
- Need to choose K
- Can be very slow: each iteration is O(KN) for N-dimensional points

K-Means Quantization

1. Select a value of K
2. Select a feature vector for every pixel (color, texture, position, or combination of these etc.)
3. Define a similarity measure between feature vectors (Usually Euclidean distance)
4. Apply the K-means algorithm to all the feature vectors

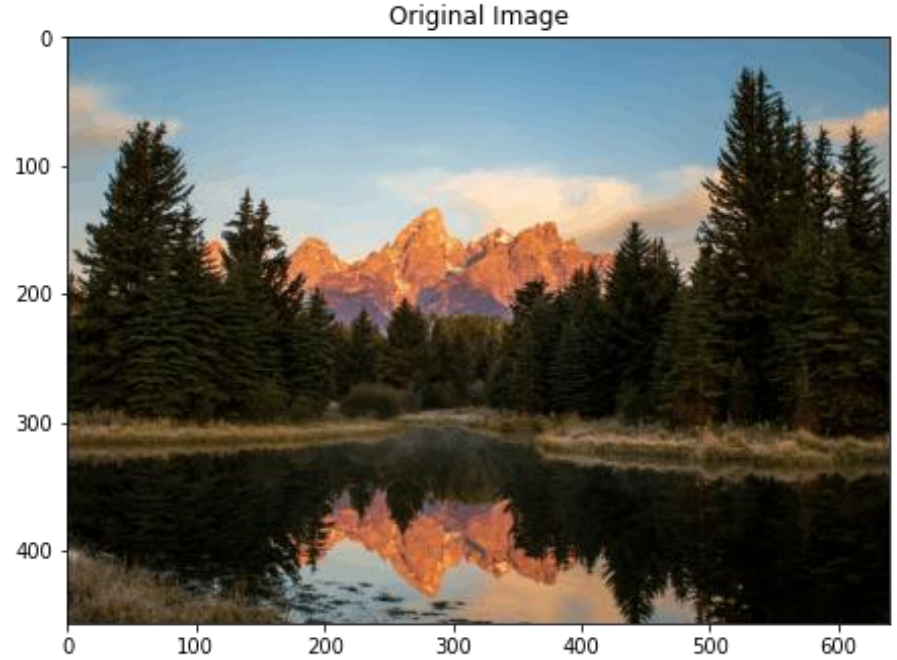
K-Means Quantization Examples

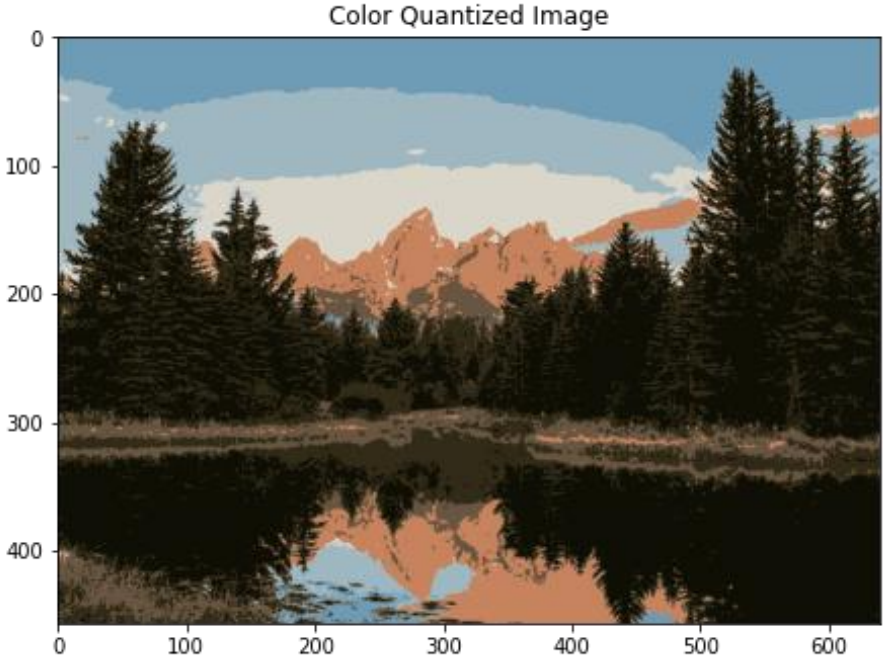




[Source: 1, 2]

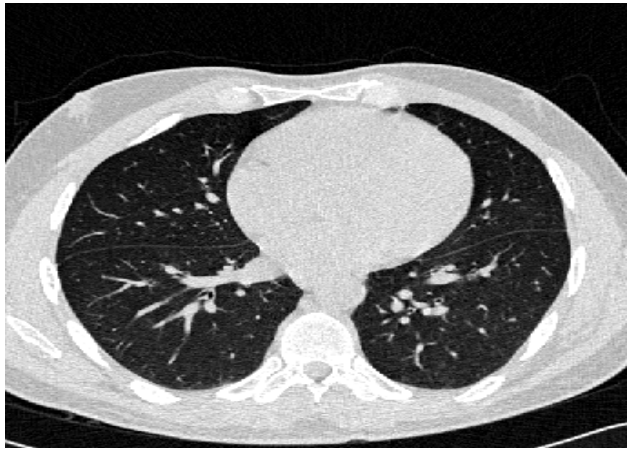
K-Means Quantization Examples



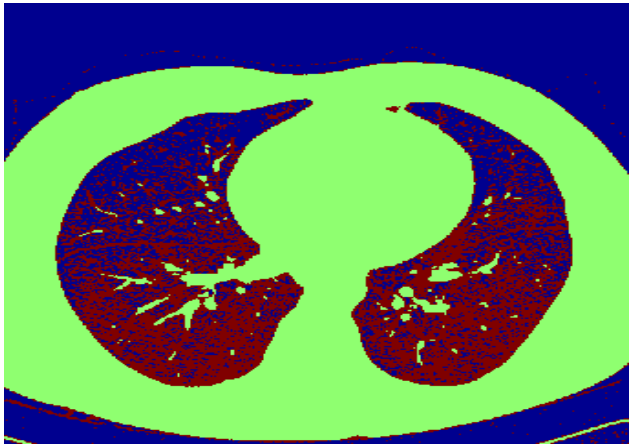


[Source: 1, 2]

K-Means Quantization Examples



Input image (*I*)



Three-cluster image using the gray levels of *input image*

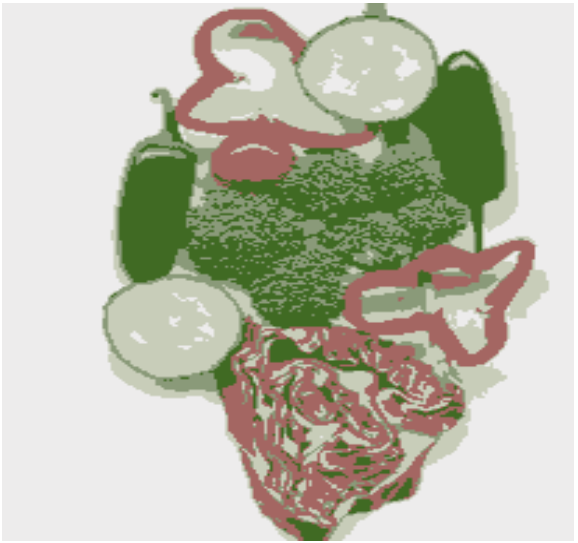
Matlab code:

```
I = double(imread('…'));

J = reshape(kmeans(I(:),3),size(I)); % need statistics toolbox
```

K-Means Quantization

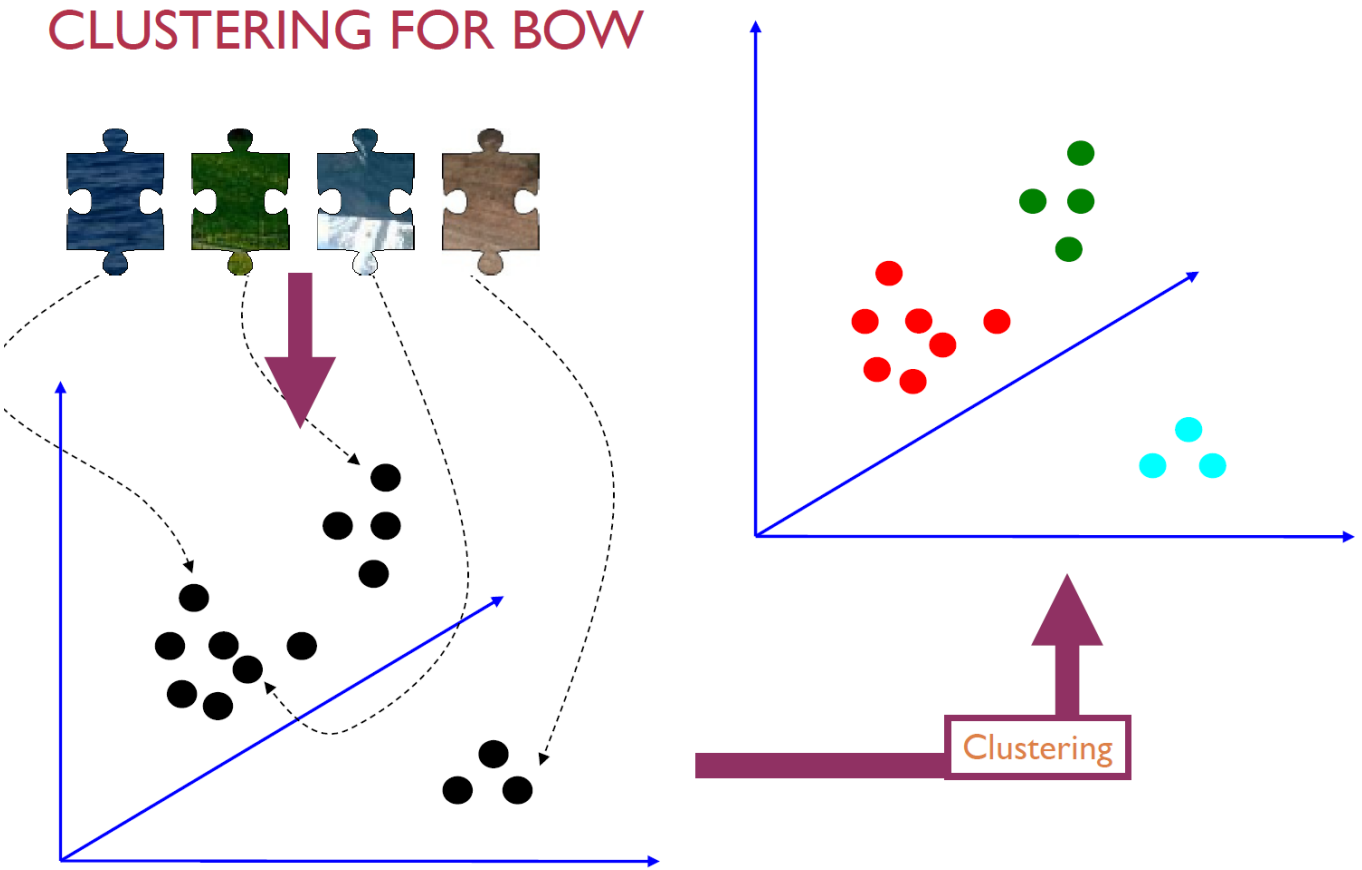
Color Image



Segmented on color

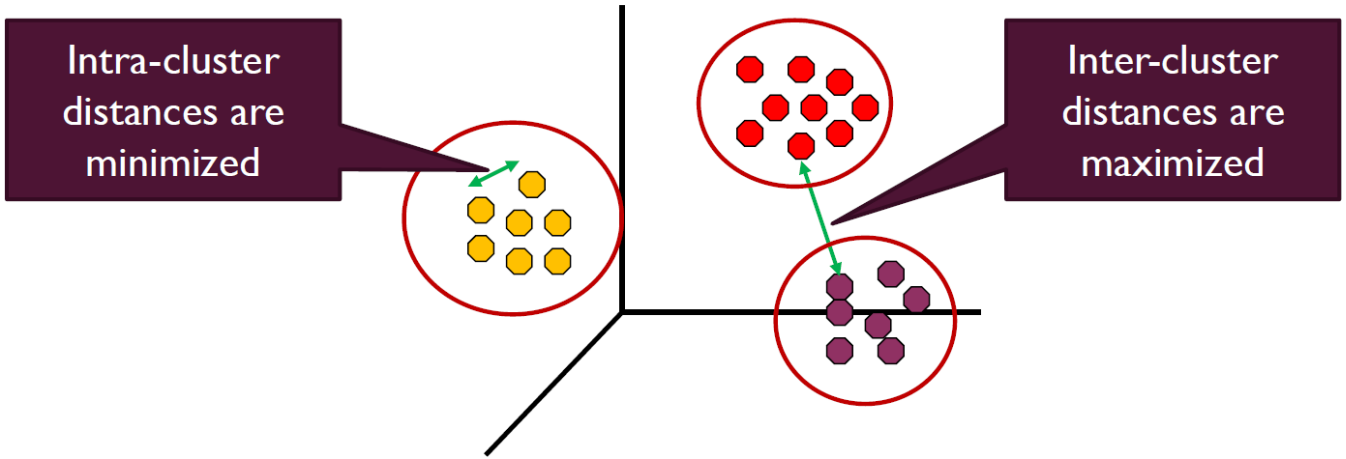
K-means clustering used for reducing the number of colors (e.g. k=4)

Reminder: Feature Clustering for BoW



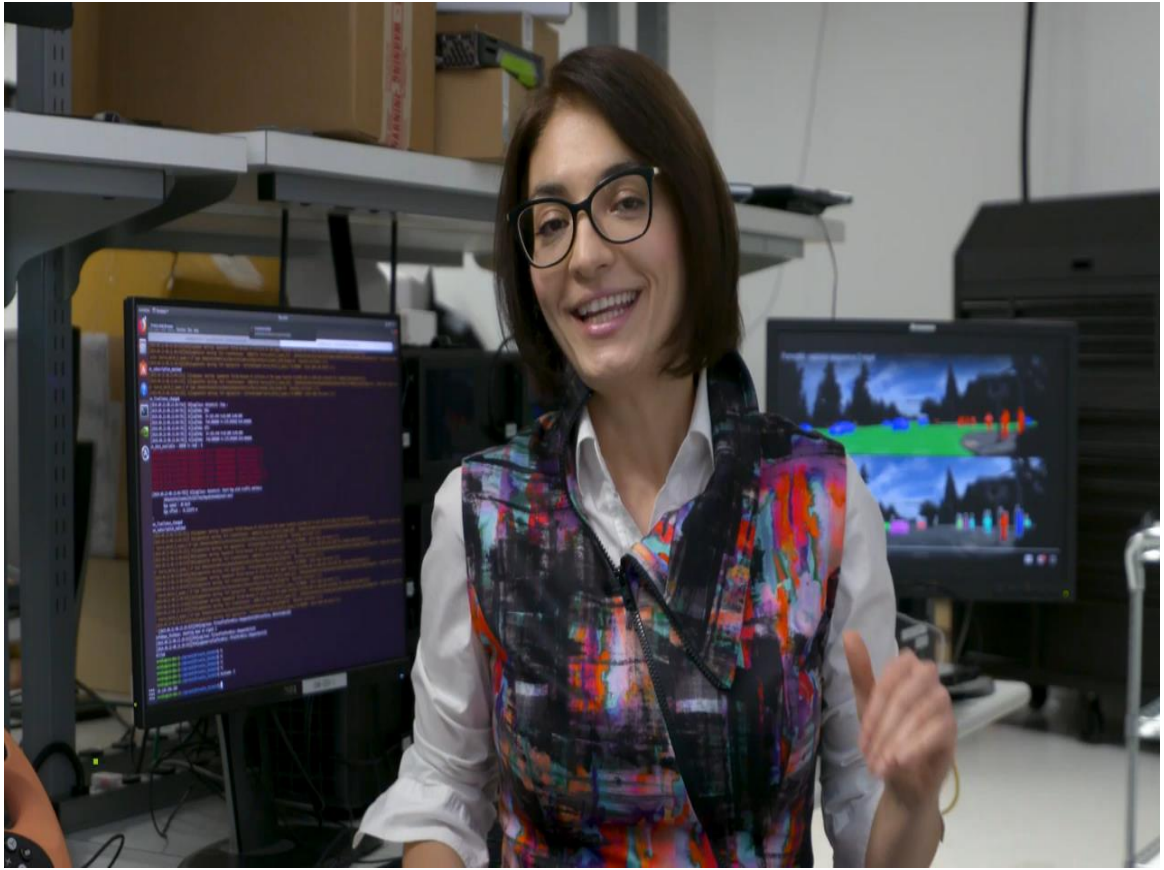
Reminder: Feature Clustering for BoW

Find clusters of similar descriptors



DeepLab v3+ Semantic Segmentation on Cityscape Dataset

Panoptic Segmentation (Semantic+Instance)



Questions?

Disclaimer

Many of the slides used here are obtained from online resources (including many open lecture materials) without appropriate acknowledgement. They are used here for the sole purpose of classroom teaching. All the credit and all the copyrights belong to the original authors. You should not copy it, redistribute it, put it online, or use it for any other purposes than for this course.