This Lecture

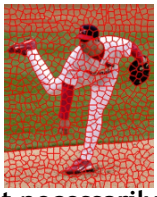
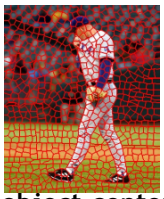
- Introduction
  - Mid-Level Vision
  - Gestalt Theory & Grouping
- Clustering & Segmentation
  - K-means & EM
  - EM algorithm

Slide Credits:
A. Efros, S. Palmer, B. Leibe, S. Lazebnik, K. Grauman, S. Seitz, C.Bishop, . Kokkinos

1

Mid-level vision

- Half-way between the image and the objects
  - Something more informative than pixels

Superpixels,
Ren & Malik

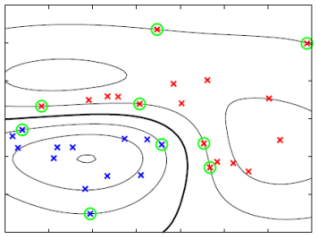
  - But not necessarily object-centered : Generic

`Blue Segment'V. Kandinsky

2

Why not go directly from image to objects?

Pattern recognition task

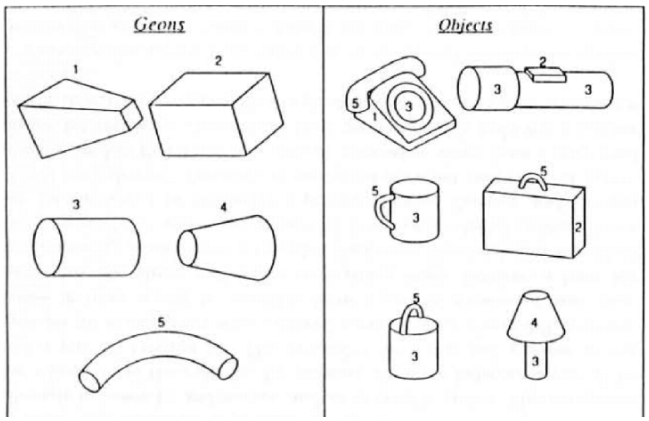


Mid-level representations suffice for recognition

What we think we see

What we really see

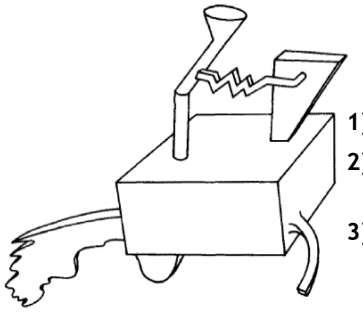
Scalability: objects and their geons



Hypothesis: there is a small number of geometric components that constitute the primitive elements of the object recognition system

Analogy: using letters to form words (compare with Ideograms)

Scalability: Recognition-by-components



1) We know that this object is nothing we know
2) We can split this objects into parts that everybody will agree
3) We can see how it resembles something familiar: "a hot dog cart"

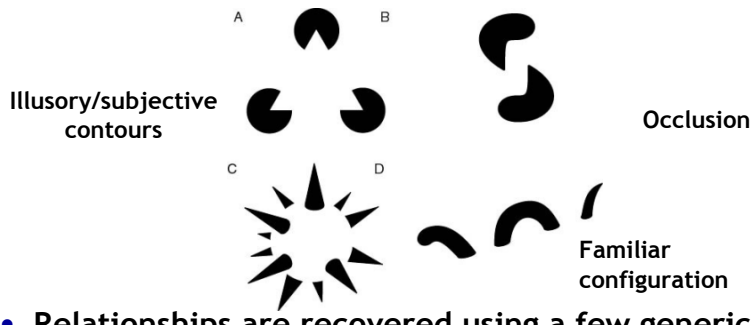
Mid-level vision

- How can we abstract from the image observations?
  - Too many pixels, edgels, blobs, junctions
  - Replace with representative, higher-level structures
  - Fewer and amenable to subsequent processing
- Core problem: Grouping
  - Region grouping (Segmentation)

7

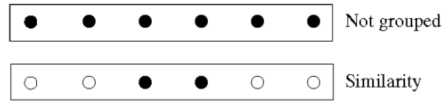
The Gestalt School

- "The whole is greater than the sum of its parts"
  - Properties result from relationships



- Relationships are recovered using a few generic cues

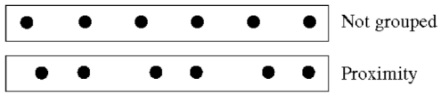
Similarity



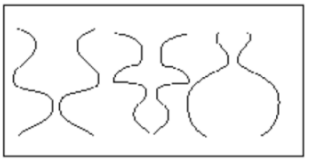
Common Fate

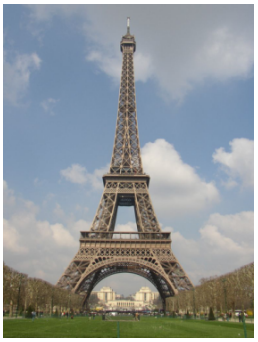
10

Proximity



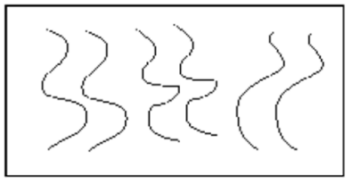
Symmetry





12

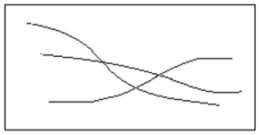
Parallelism



Parallelism

13

Continuity



Continuity

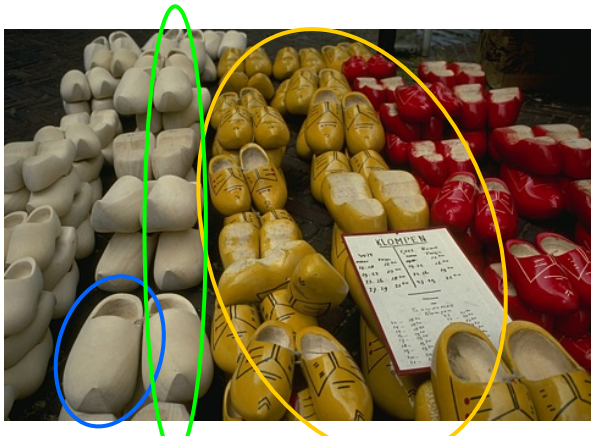
Gestalt theory and computer vision

- Gestalt heritage: mostly conceptual
- Turning Gestalt cues into numerical quantities:
  - Common fate: motion estimation
  - Parallelism: texture analysis
  - Symmetry: ridge detection
  - Similarity: *region-based segmentation*
  - Closure, continuity: *boundary-based segmentation*
- Main problem: how do we choose when to rely on each of these cues?

15

Cue combination problem

- Different cues lead to different segmentations



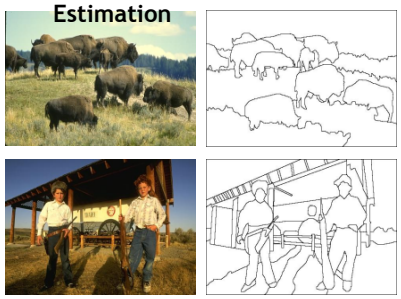
16

Image segmentation is an ill-posed problem

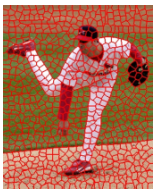
- No `optimal' segmentation exists

- `Good' segmentation: highly task-dependent

Depth Ordering Estimation

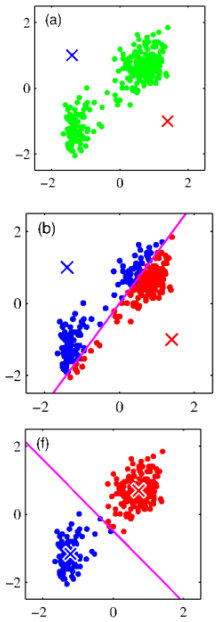


Motion



This Lecture

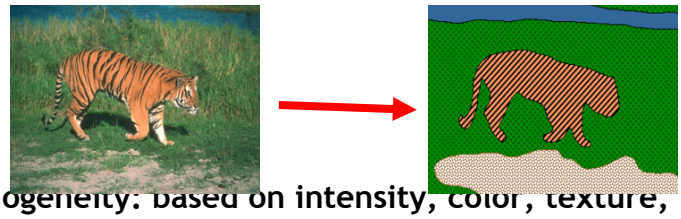
- Introduction
  - Mid-Level Vision
  - Gestalt Theory & Grouping
- Clustering & Segmentation
  - K-means
  - EM algorithm



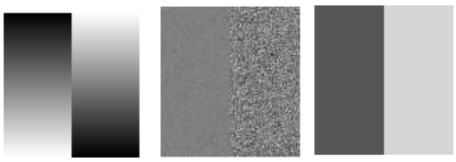
18

Segmentation Problem

- Task: Partition image into homogeneous regions

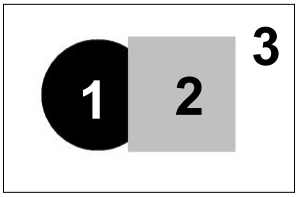


- Homogeneity: based on intensity, color, texture, motion, depth, shading ….
  - Intensity:

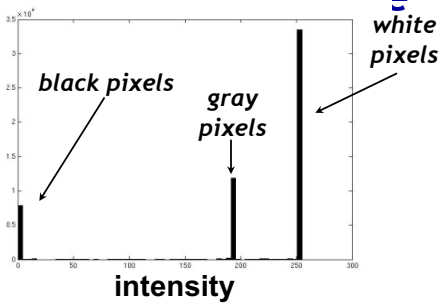


19

Intensity-based segmentation: toy example



input image

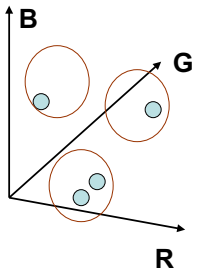


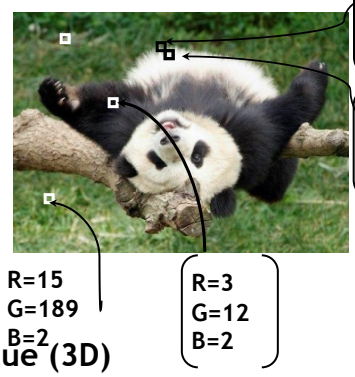
- 1D feature vector: intensity measurement
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.

20

Feature Space

- At each pixel, form a vector of measurements describing image properties: image features
- Map observations into feature space
- Group pixels based on color similarity



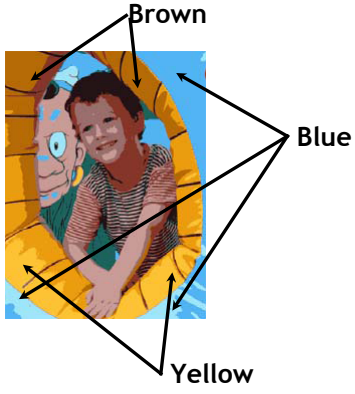


- Feature space: color value (3D)

21

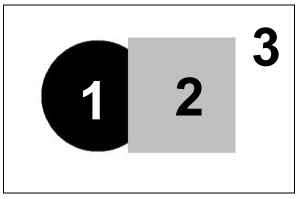
Segmentation Problem

- Modeling the features separately within each segment: substantially easier than modeling the image.

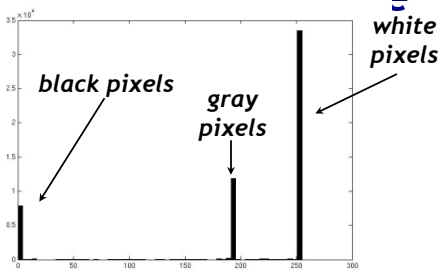


22

Intensity-based segmentation: toy example

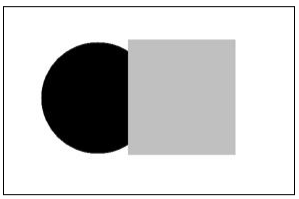


input image

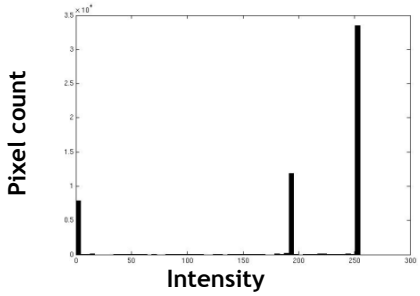


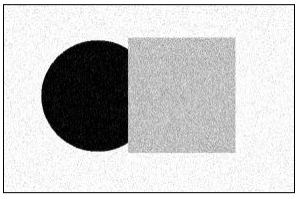
- 1D feature vector: intensity measurement
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.

23

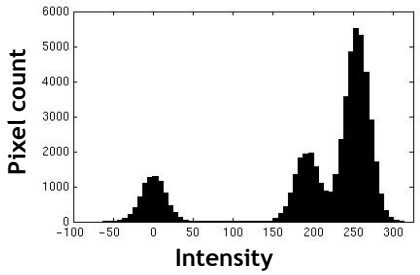


Input image

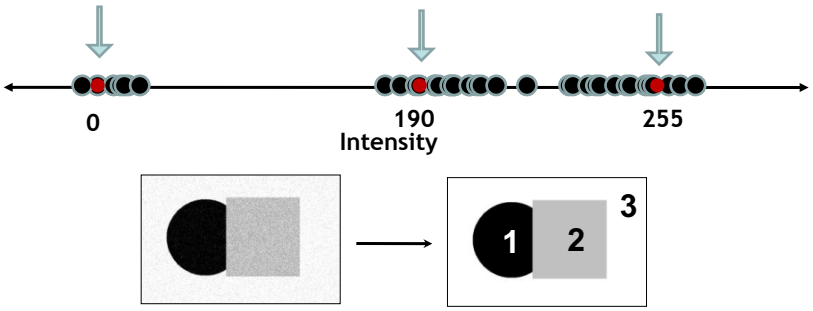




Input image



24



- Goal: choose three "centers"
- Label every pixel according to closest center
- But how can we find the centers?

25

Clustering

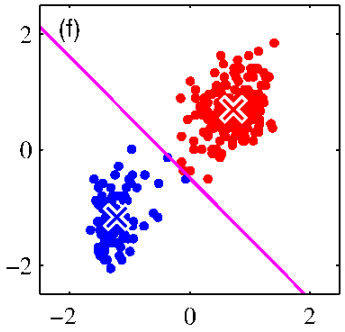
- "chicken and egg" problem:
  - Known *centers*: group points by allocating each to its closest center.

  - Known *group memberships*: get centers by computing group mean.

26

K-Means algorithm

- Input: features
  - d: feature vector dimensionality
  - N: number of pixels
- Output: centers & assignments



K-Means Clustering

- Randomly initialize *k* cluster centers.
- Iterate:
  1. Given cluster centers, determine points in each cluster
     - For each point i, find the closest center. Put i into cluster j
  2. Given points in $m(i) = \text{argmin}_j |x^i - c^j|$ enters.
     - Set center to be the mean of points in cluster j
  3. If c_i have change $c^j = \frac{\sum_{i=1}^{N}(m(i)=j)x^i}{\sum_{i=1}^{N}(m(i)=j)}$
- Guaranteed convergence to local minimum of

$$F(m,c) = \sum_{i=1}^{N}\sum_{k=1}^{K} |x^i - c^{m(i)}|^2$$

28

K-Means Clustering Results

- K-means clustering based on intensity or color

Image | Intensity-based clusters | Color-based clusters

- Clustering (r,g,b,x,y) values enforces spatial coherence

29

SLIC Superpixels

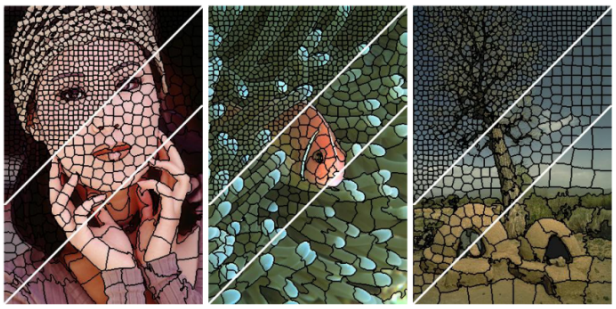


Fig. 1. Image segmented using our algorithm into superpixels of (approximate) size 64, 256, and 1024 pixels. The superpixels are compact, uniform in size, and adhere well to region boundaries.

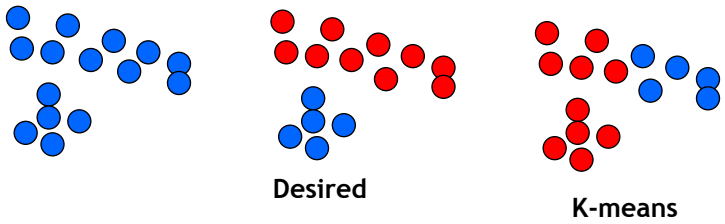
putational overhead. We introduce a novel algorithm that clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels. The simplicity of

Achanta et al. "SLIC Superpixels"

30

Limitations of k-means

- Euclidean distance-based criterion
  - No justification for Euclidean metric in arbitrary feature space



Desired

K-means

- Remedy: introduce more flexible models for observations within each group
  - K-means allows only spherical clusters
  - Consider ellipsoidal

31

d-dimensional Gaussian distribution

- Determined by mean and covariance matrix

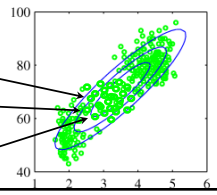
$$P(x|\mu,\Sigma) = \frac{1}{\sqrt{2\pi^d|\Sigma|}} \exp\left(-(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- Maximum likelihood parameter estimates:

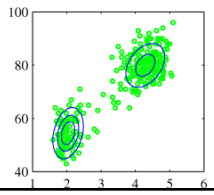
$$\mu^* = \frac{\sum_i x^i}{N}$$

$$\Sigma_k^* = \frac{\sum_i (x^i-\mu^*)^T(x^i-\mu^*)}{N}$$

P(x) = .1
P(x) = .2
P(x) = .5

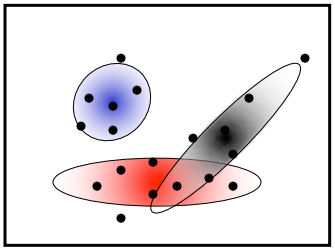


?



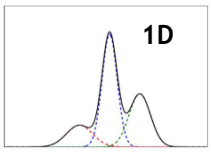
32

Mixture of Gaussians



- K Gaussian blobs with parameters: μ_k, Σ_k
- Blob *k* is selected with probability π_k
- The likelihood of observing x is a weighted mixture of Gaussians

$$P(x, \theta) = \sum_{k=1}^{K} \pi_k P(x, \mu_k, \Sigma_k)$$



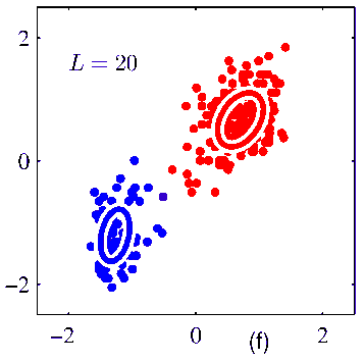
- Parameter Estimation: maximize

$$P(x|\theta) = \prod_{i=1}^{N} P(x^i|\theta)$$

33

Expectation Maximization algorithm

Start with two random Gaussians!



E-step (Bayes' Rule)

$$\begin{aligned} R_{i,j} &= P(z^i = j|\theta) \\ &= \frac{\pi_j P(x^i|\mu_j, \Sigma_j)}{\sum_{k=1}^{K} P(x^i|\mu_k, \Sigma_k)\pi_k} \end{aligned}$$

M-step

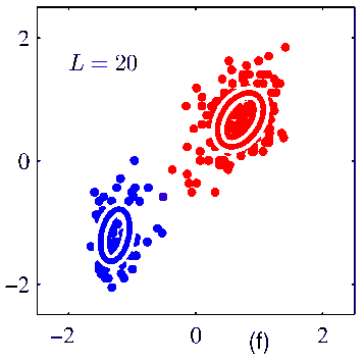
$$\mu^j = \frac{\sum_{i=1}^{N} R_{i,j} x^i}{\sum_{i=1}^{N} R_{i,j}}$$

$$\Sigma^j = \frac{\sum_{i=1}^{N} R_{i,j}(x^i - \mu^j)^T(x^i - \mu^j)}{\sum_{i=1}^{N} R_{i,j}}$$

$$\pi^j = \frac{\sum_{i=1}^{N} R_{i,j}}{N}$$

Expectation Maximization algorithm

Start with two random Gaussians!



E-step (Bayes' Rule)

For each point:

- Does it look like it came from G1?
- Does it look like it came from G2?
- P(G|point)? -> posterior!

M-step

- Adjust the mean of the Gaussians to fit points assigned to them
- Adjust the variance of the Gaussians to fit points assigned to them

Keep iterating until convergence!

K-means vs. EM

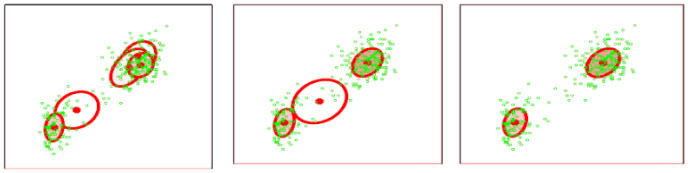
k-means	EM
Closest center's index assignment	Soft
Isotropic Distance (Euclidean)	Anisotropic Likelihood (Covariance-based)
Fast (e.g. kd-trees)	Accurate & more flexible
More robust to initialization	Prone to local minima

Typical usage: initialize EM with k-means results

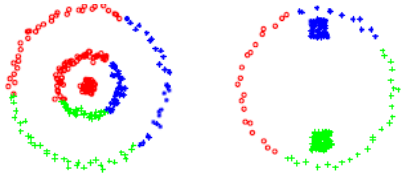
36

Problems of K-Means/EM

- Number of clusters



- Initialization/local minima
- Mismatch with data distribution

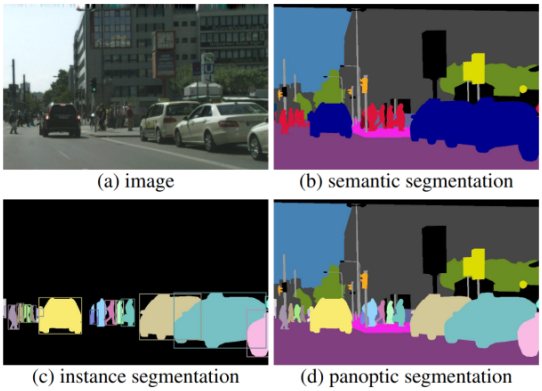


Object Detection and Segmentation

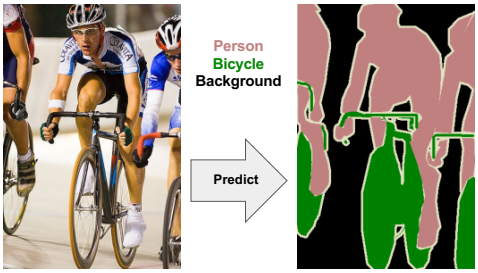
Recent approaches

Slides originally done by Vincent Chen & Edward Chou

Types of Image Segmentation



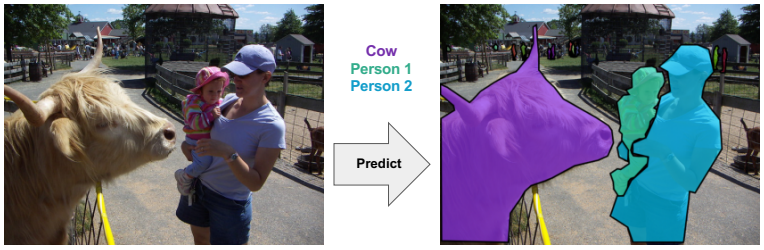
Types of Image Segmentation: Semantic segmentation



1 semantic label per pixel

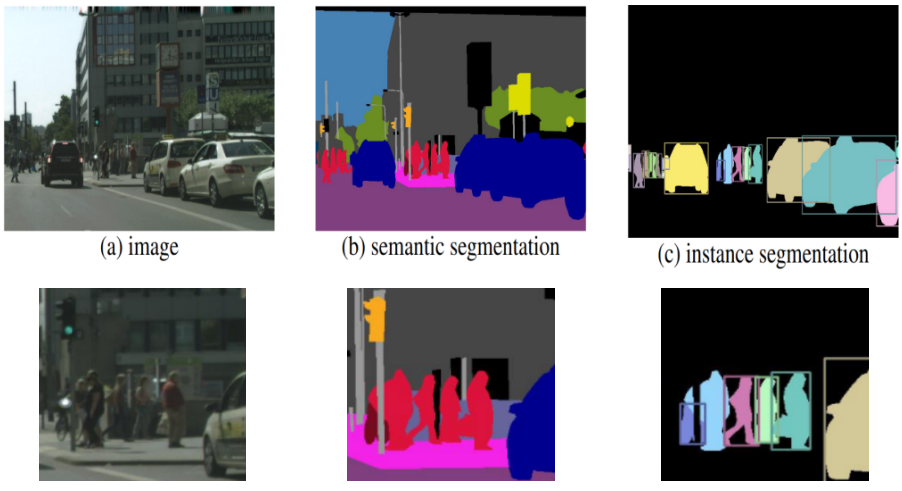
Types of Image Segmentation: Instance segmentation

~ Object detection + outlining objects of interest.

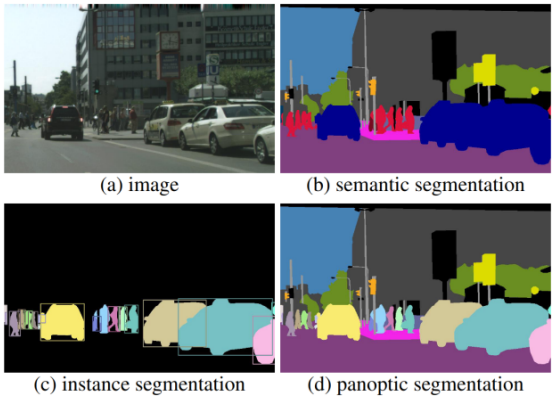


Each instance has a mask.

Types of Image Segmentation: Panoptic segmentation



Types of Image Segmentation: Panoptic segmentation



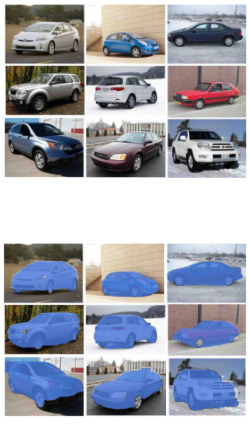
Stuff: road, grass, sky, etc
Things: people, cars, animals, etc
Each pixel has two labels:
Semantic label + instance label

Types of Image Segmentation: Others

- Scene parsing

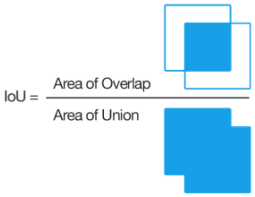
Zhou et al. "Scene Parsing through ADE20K Dataset". CVPR'17

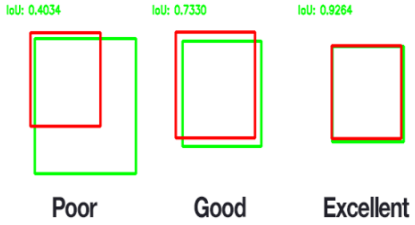
- Object co-segmentation



Chen et al. "Show, Match and Segment: Joint Weakly Supervised Learning of Semantic Matching and Object Co-segmentation". PAMI'20

Evaluation of semantic/instance segmentation





For a given threshold:

TP: IoU between the predicted and the ground truth exceeds the threshold.
FP: the predicted cannot be associated to the ground truth because their IoU is lower than threshold.
FN: the ground truth cannot be associated to the predicted because their IoU is lower than threshold.

Images credit: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Evaluation of semantic/instance segmentation

For a given threshold:

TP: IoU between the predicted and the ground truth exceeds the threshold.
FP: the predicted cannot be associated to the ground truth because their IoU is lower than threshold.
FN: the ground truth cannot be associated to the predicted because their IoU is lower than threshold.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

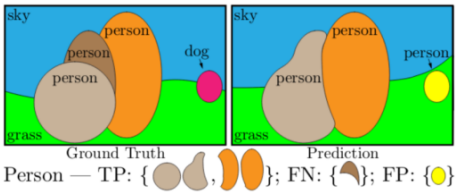
Area under the Precision-Recall curve is called **Average Precision (AP)**

```
Average Precision (AP):
  AP               % AP at IoU=.50:.05:.95 (primary challenge metric)
  AP^IoU=.50       % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75       % AP at IoU=.75 (strict metric)
```

← MS-COCO

Evaluation of panoptic segmentation

For a predicted segment "p" and a ground truth segment "g":

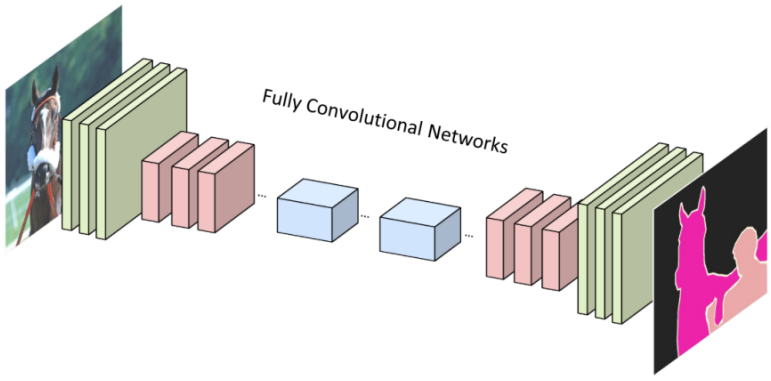


$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$$

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

Image segmentation methods

Modern methods



Slide credit: VALSE 2019 Tutorial

Semantic segmentation

- FCN
- SegNet/DeconvNet
- U-net
- DeepLab v1, v2,v3, v3+

SegNet/DeconvNet

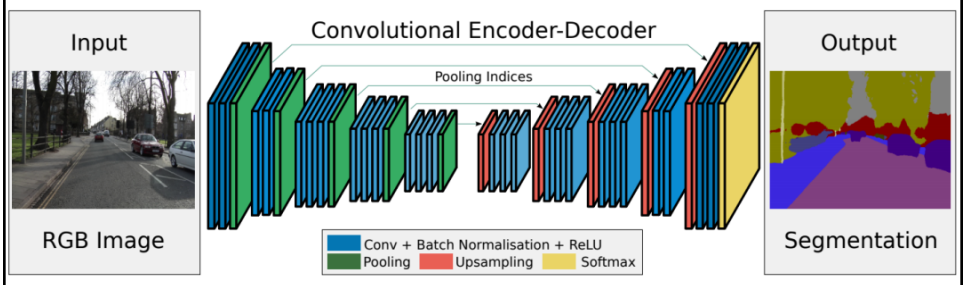


Figure taken from the SegNet paper

Inverse max-pooling in SegNet and DeconvNet

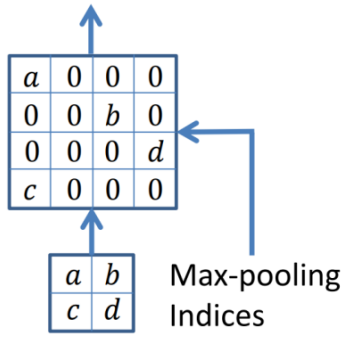


Figure taken from the SegNet paper

SegNet

1. Applications include autonomous driving, scene understanding, etc.
2. Yields poor results mainly because max-pooling and subsampling reduce feature map resolution and hence output resolution is reduced.
3. Even if extrapolated to original resolution, lossy image is generated.

U-net

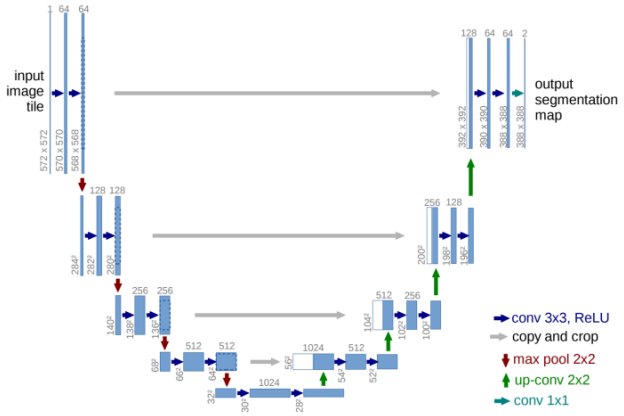
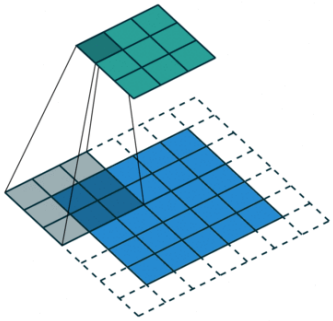
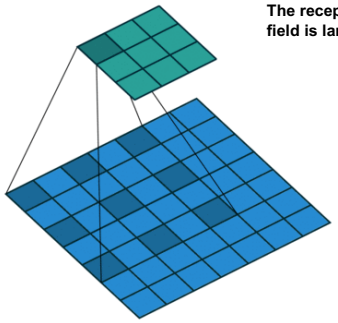


Figure taken from the U-net paper

DeepLab Family - dilated (atrous) convolution



Standard convolution

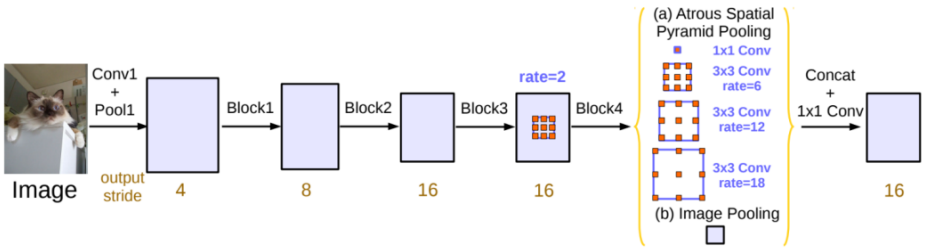


The receptive field is larger!!

Dilated convolution

Figure taken from: https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

DeepLab Family - atrous spatial pyramid pooling



Instance segmentation: Mask R-CNN

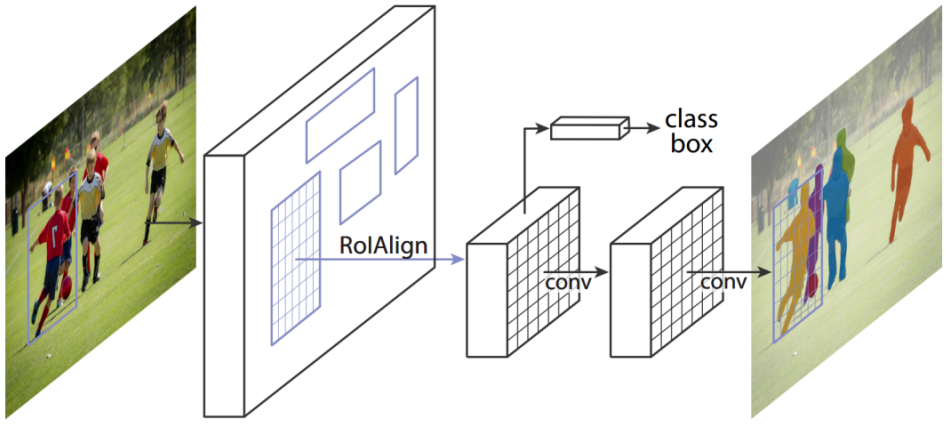


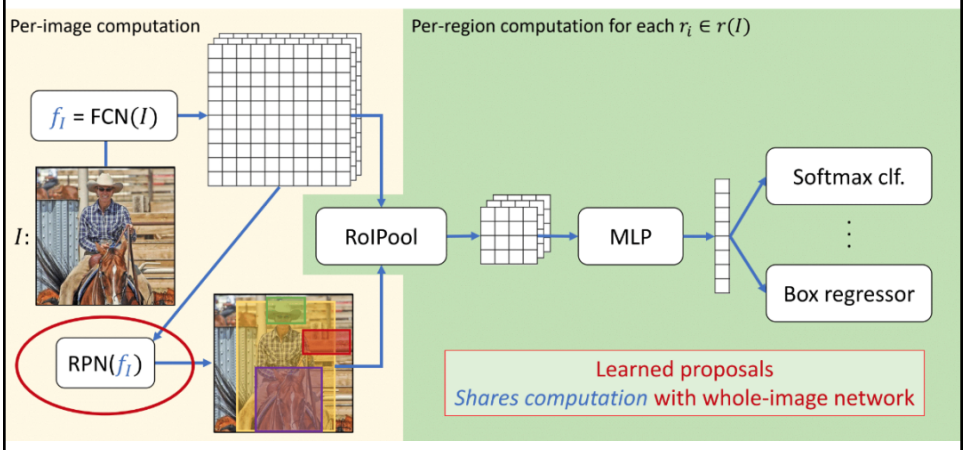
Figure taken from the Mask R-CNN paper

Mask R-CNN

1. Backbone Architecture
2. Scale Invariance (e.g. Feature Pyramid Network (FPN))
3. Region Proposal Network (RPN)
4. Region of interest feature alignment (RoIAlign)
5. Multi-task network head
   a. Box classifier
   b. Box regressor
   c. Mask predictor
   d. Keypoint predictor

Slide credit: ICCV 2019 tutorial

Faster R-CNN



RIO align

Bilinear interpolation to obtain sub-pixel features

Transform arbitrary size proposal into a fixed-dimensional!

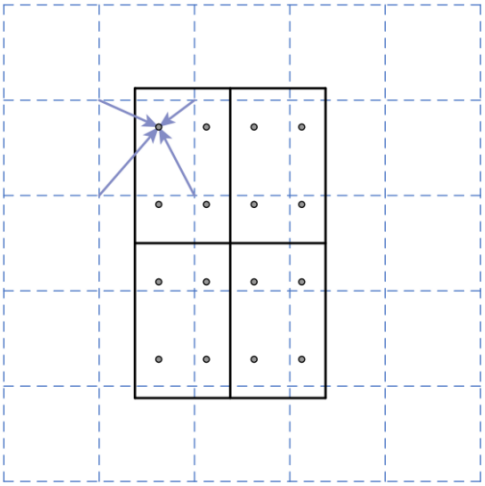
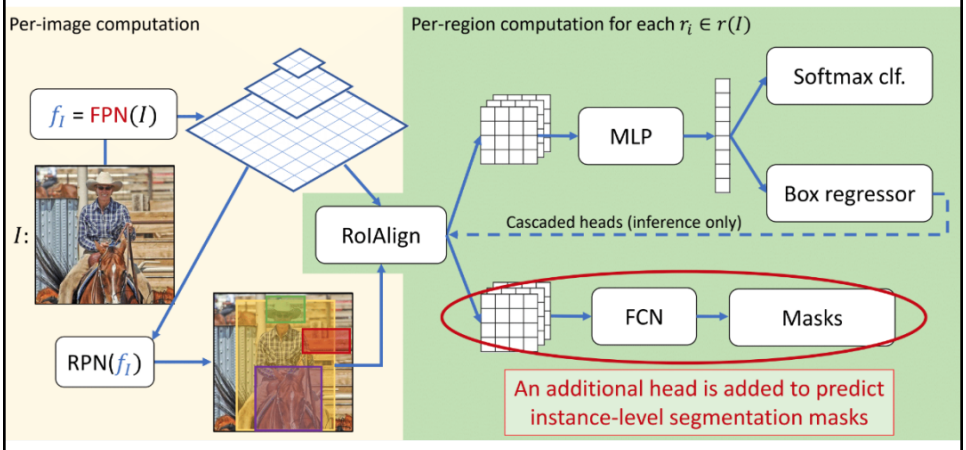


Figure taken from the Mask R-CNN paper

Slide credit: ICCV 2019 tutorial

Mask R-CNN



Mask R-CNN results

Detection vs. segmentation

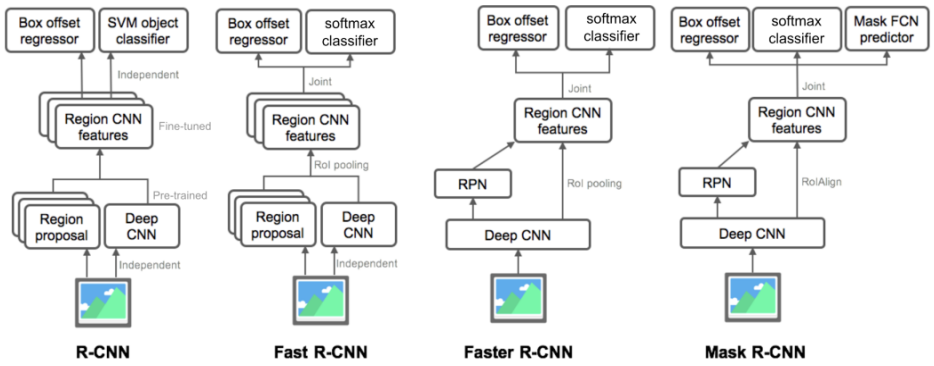
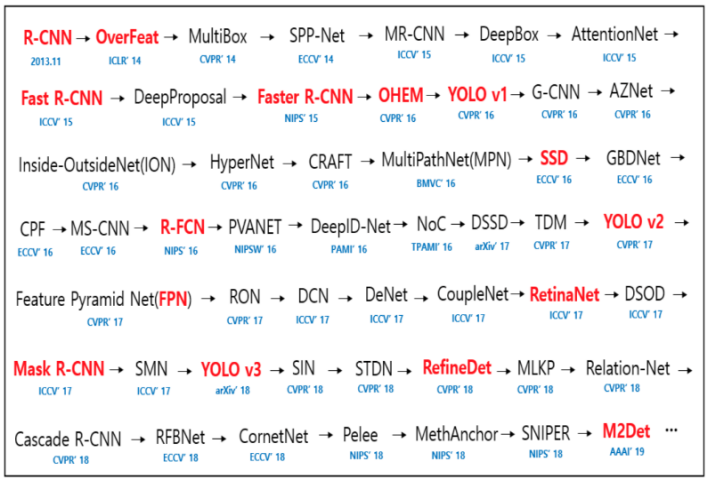


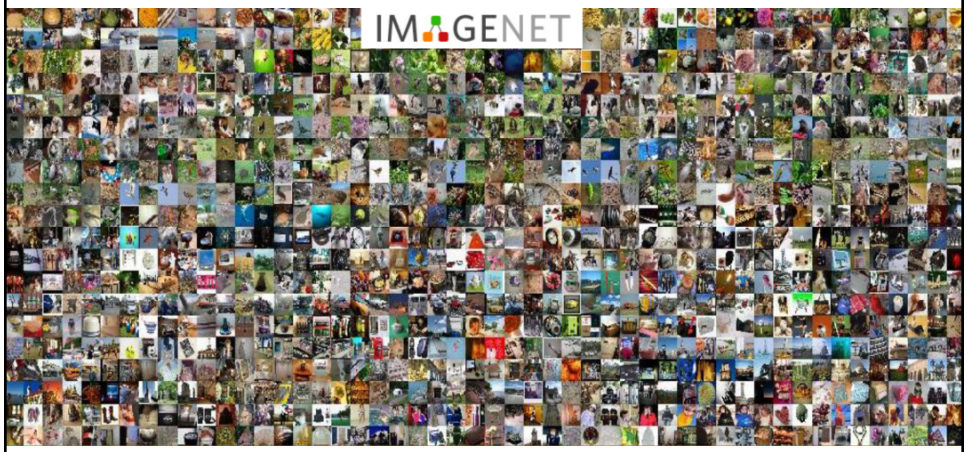
Figure from https://lilianweng.github.io/

Object Detection: State of the Art Progress



Slide credit: Lex Fridman

Weakly-Supervised Semantic Segmentation



Slide credit: VALSE 2019 Tutorial

Weakly-Supervised Semantic Segmentation

Slide credit: VALSE 2019 Tutorial

Weakly-Supervised Semantic Segmentation

Manual annotations for: **Motorbike**, **Person**

Image-level labels | Points | Bounding boxes | Scribbles | Pixel-level labels

Annotation time (s)

1s / class | 2.4s / instance | 10.2s / instance | 10.9s / instance | 78s / instance

Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

Weakly-Supervised Semantic Segmentation

Manual annotations for: **Motorbike**, **Person**

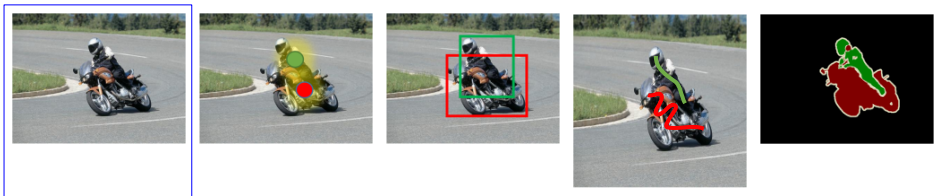


Image-level labels | Points | Bounding boxes | Scribbles | Pixel-level labels

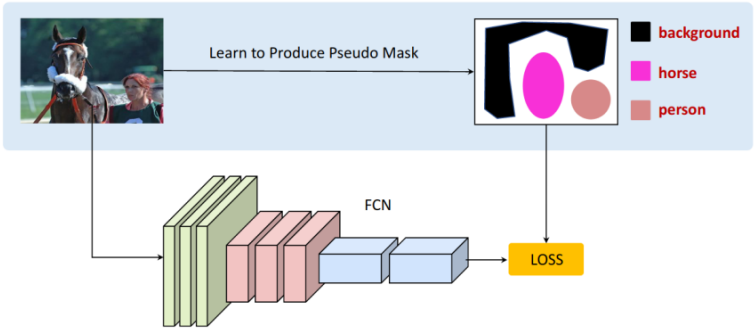
Annotation time (s)

1s / class | 2.4s / instance | 10.2s / instance | 10.9s / instance | 78s / instance

Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

Image-level labels: Pseudo Pixel-level Masks

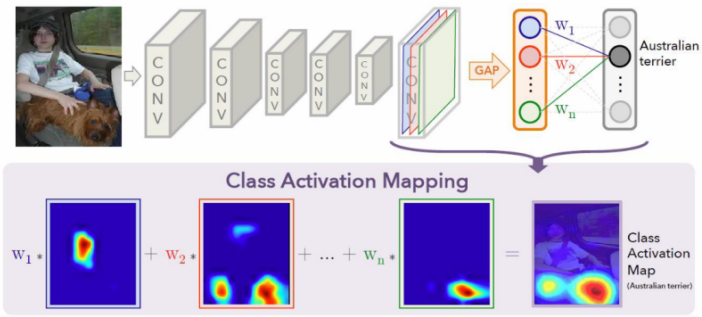
Standard Pipeline



Slide credit: VALSE 2019 Tutorial

Image-level labels: Pseudo Pixel-level Masks

Class Activation Mapping



Zhou et al. "Learning deep features for discriminative localization." CVPR 2016

Weakly-Supervised Semantic Segmentation

Manual annotations for: **Motorbike**, **Person**

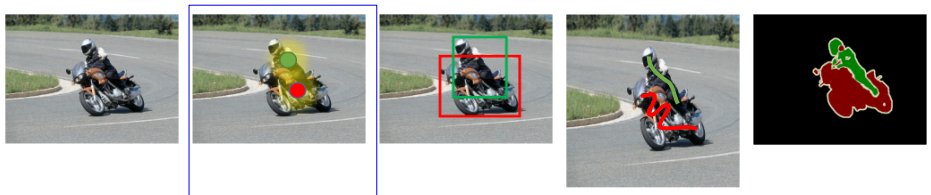


Image-level labels | Points | Bounding boxes | Scribbles | Pixel-level labels

Annotation time (s)

1s / class | 2.4s / instance | 10.2s / instance | 10.9s / instance | 78s / instance

Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

Point-level supervision:

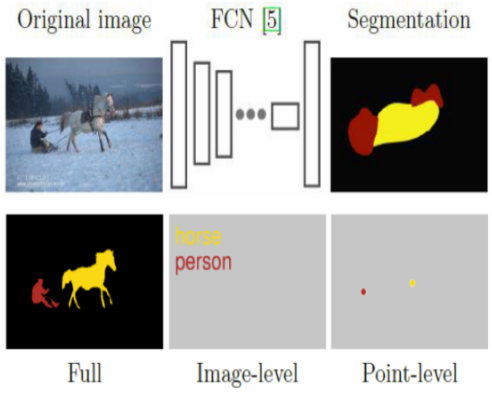


Image-Level Supervision:

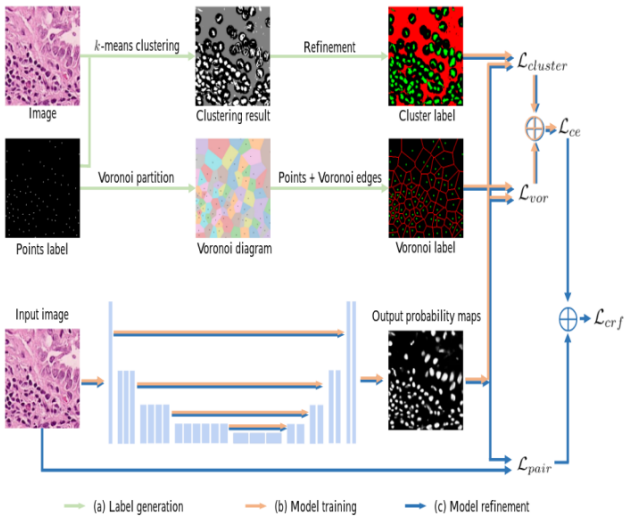
$$\mathcal{L}_{img}(S, L, L') = -\frac{1}{|L|}\sum_{c\in L}\log(S_{t_c c}) - \frac{1}{|L'|}\sum_{c\in L'}\log(1 - S_{t_c c})$$

Point-Level Supervision:

$$\mathcal{L}_{point}(S, G, L, L') = \mathcal{L}_{img}(S, L, L') - \sum_{i\in\mathcal{I}_s}\alpha_i \log(S_{iG_i})$$

Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

Point-level supervision:



Qu et al. "Weakly supervised deep nuclei segmentation using points annotation in histopathology images." MIDL 2019

Weakly-Supervised Semantic Segmentation

Manual annotations for: **Motorbike**, **Person**

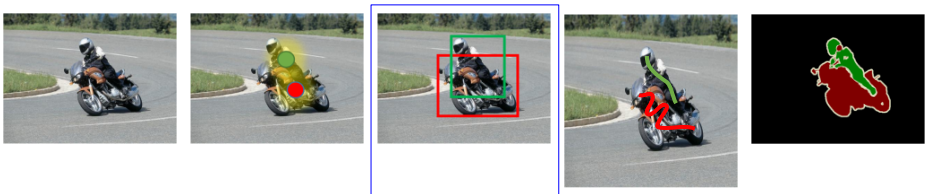


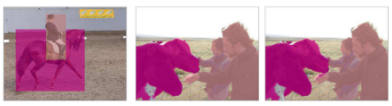
Image-level labels | Points | Bounding boxes | Scribbles | Pixel-level labels

Annotation time (s)

1s / class | 2.4s / instance | 10.2s / instance | 10.9s / instance | 78s / instance

Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

Object bounding boxes: Pseudo Pixel-level Masks

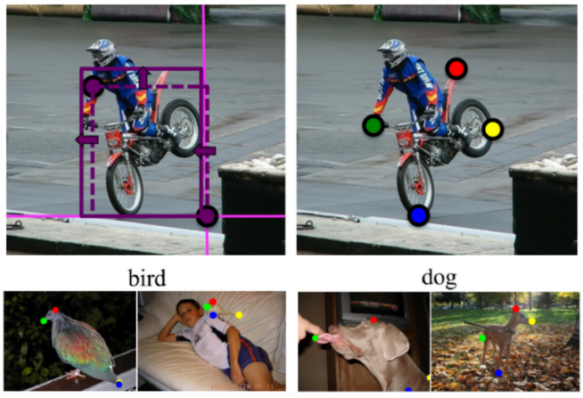


Training sample, with box annotations | Test image, fully supervised result | Test image, weakly supervised result

Example input rectangles | Output after 1 training round | After 5 rounds | After 10 rounds | Ground truth

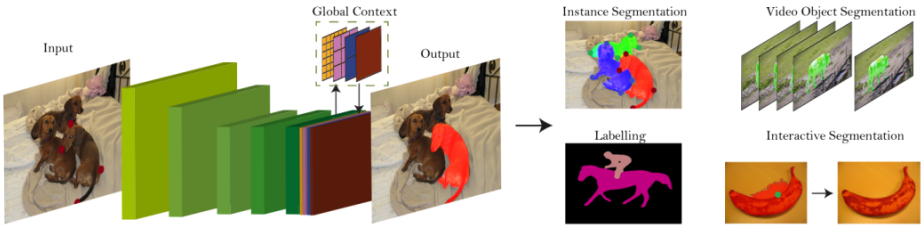
Khoreva et al. "Simple Does It: Weakly Supervised Instance and Semantic Segmentation." CVPR 2017

Object extreme points:



Maninis et al. "Deep extreme cut: From extreme points to object segmentation." CVPR 2018

Object extreme points:



Maninis et al. "Deep extreme cut: From extreme points to object segmentation." CVPR 2018

Weakly-Supervised Semantic Segmentation

Manual annotations for: **Motorbike**, **Person**

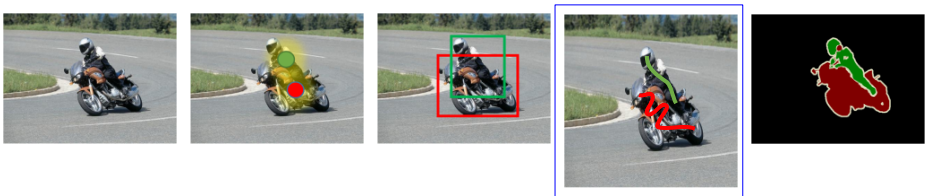


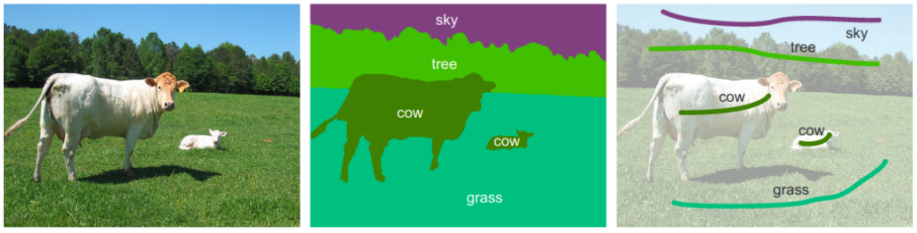
Image-level labels | Points | Bounding boxes | Scribbles | Pixel-level labels

Annotation time (s)

1s / class | 2.4s / instance | 10.2s / instance | 10.9s / instance | 78s / instance

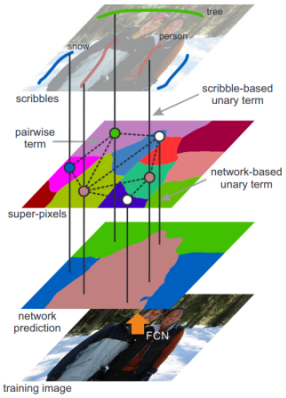
Bearman et al. "What's the point: Semantic segmentation with point supervision." ECCV 2016

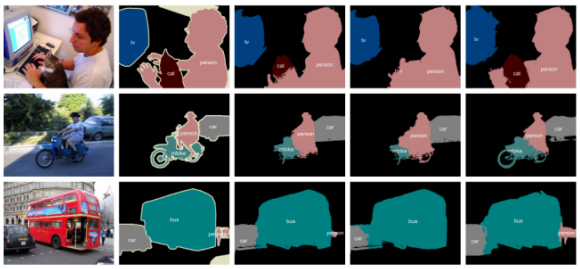
Scribbles:



(a) image (b) mask annotation (c) scribble annotation

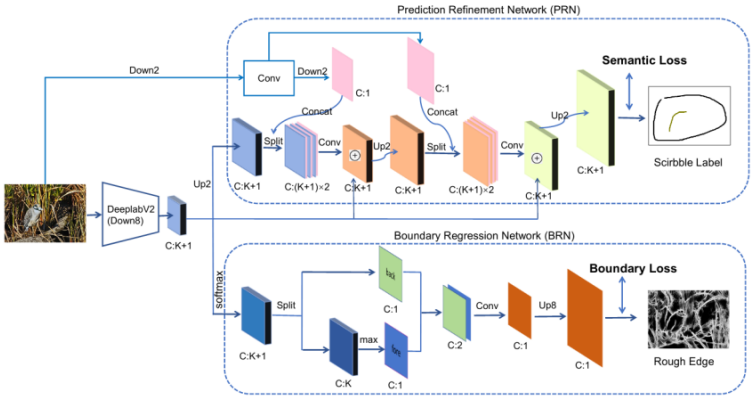
Scribbles:





Lin et al. "ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation." CVPR 2016

Scribbles:



Wang et al. "A scribble-supervised semantic segmentation approach." IJCAI 2019

Any questions?