# High-Level Vision 4

Week 11

High-level Vision: Object Detection

High-level Vision: Image Segmentation

# Announcements

- **Drop-in Sessions:** Please bring your assignment/exam questions to the following locations
  - 14:00-16:00 Wednesday 15 and 22 May, 1.23 Hanna Neumann (Hoang)
  - 15:00-17:00 Friday 17 and 24 May, 1.23 Hanna Neumann (Yiran)

# Announcements

- **Assignment 3** due 11:59pm **this Friday** (17 May)
  - **Zero** marks if either report or code submitted late (unless extension)
    - Submit early; you can always resubmit an updated version later
    - Depending on your internet connection and load on the TurnItIn servers, uploading can sometimes be slow, so please factor this into your submission schedule
  - Submit your report (PDF) and code (ZIP file) **separately under the correct tab** in the submission box, or you will receive a mark of zero
  - Follow the instructions under Submission Requirements
  - Assignment figures/screenshots: must be sufficient resolution for the markers to read and interpret

# Weekly Study Plan: Overview

| Wk | Starting | Lecture | Lab | Assessment |
|---|---|---|---|---|
| 1 | 19 Feb | Introduction | ✗ | |
| 2 | 26 Feb | Low-level Vision 1 | 1 | |
| 3 | 4 Mar | Low-level Vision 2 | 1 | |
| | | Mid-level Vision 1 | | |
| 4 | 11 Mar | Mid-level Vision 2 | 1 | CLab1 report due Friday |
| | | High-level Vision 1 | | |
| 5 | 18 Mar | High-level Vision 2 | 2 | |
| 6 | 25 Mar | High-level Vision 3[1] | 2 | |
| | 1 Apr | Teaching break | ✗ | |
| | 8 Apr | Teaching break | ✗ | |
| 7 | 15 Apr | 3D Vision 1 | 2 | CLab2 report due Friday |
| 8 | 22 Apr | 3D Vision 2 | 3 | |
| 9 | 29 Apr | 3D Vision 3 | 3 | |
| 10 | 6 May | 3D Vision 4 | 3 | |
| | | Mid-level Vision 3 | | |
| 11 | 13 May | High-level Vision 4 | ✗ | CLab3 report due Friday |
| 12 | 20 May | Course Review | ✗ | |

# Weekly Study Plan: Part B

| Wk | Starting | Lecture | By |
|---|---|---|---|
| 7 | 15 Apr | 3D vision: introduction, camera model, single-view geometry | Dylan |
| 8 | 22 Apr | 3D vision: camera calibration, two-view geometry (homography) | Dylan |
| 9 | 29 Apr | 3D vision: two-view geometry (epipolar geometry, triangulation, stereo) | Dylan |
| 10 | 6 May | 3D vision: multiple-view geometry | Weijian |
| | | Mid-level vision: optical flow, shape-from-X | Dylan |
| 11 | 13 May | High-level vision: self-supervised learning, detection, segmentation | Dylan |
| 12 | 20 May | Course review | Dylan |

# Photometric Stereo: Student Question

$$\underbrace{[I_1 \quad \dots \quad I_n]}_{\substack{I \\ \text{mxn}}} = \underbrace{\rho N^\top}_{\substack{G \\ \text{mx3}}} \underbrace{[L_1 \quad \dots \quad L_n]}_{\substack{L \\ \text{3xn}}}$$

- More than 1 pixel? Stack into a system and solve as before

$$G = (IL^\top)(LL^\top)^{-1}$$

- Question: What happens if you have more pixels than lights?
  - This is entirely okay, so long as you have at least 3 (linearly independent) lights

# Assignment 3 Homographies

- Why can we compute a homography? Mountains aren't planar...
  - Camera is rotating but not translating (approx): translation is zero ($\boldsymbol{t} = \mathbf{0}$)
  - Points are on the "plane at infinity"

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \propto \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}_{\{3\times 3\}}} \underbrace{[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3 \quad \boldsymbol{t}]}_{[\mathbf{R}_{\{3\times 3\}} \quad \boldsymbol{t}_{\{3\times 1\}}]} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \underbrace{\mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3]}_{\mathbf{H}_{\{3\times 3\}}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{H} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

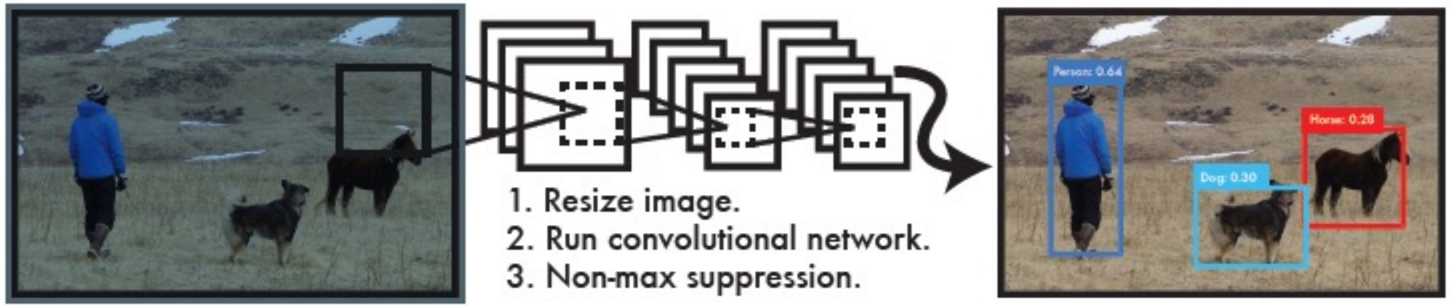
- While we are here:
  - $\gamma$: "skew" parameter – zero unless you have a strange camera

# Object Detection

High-level Vision

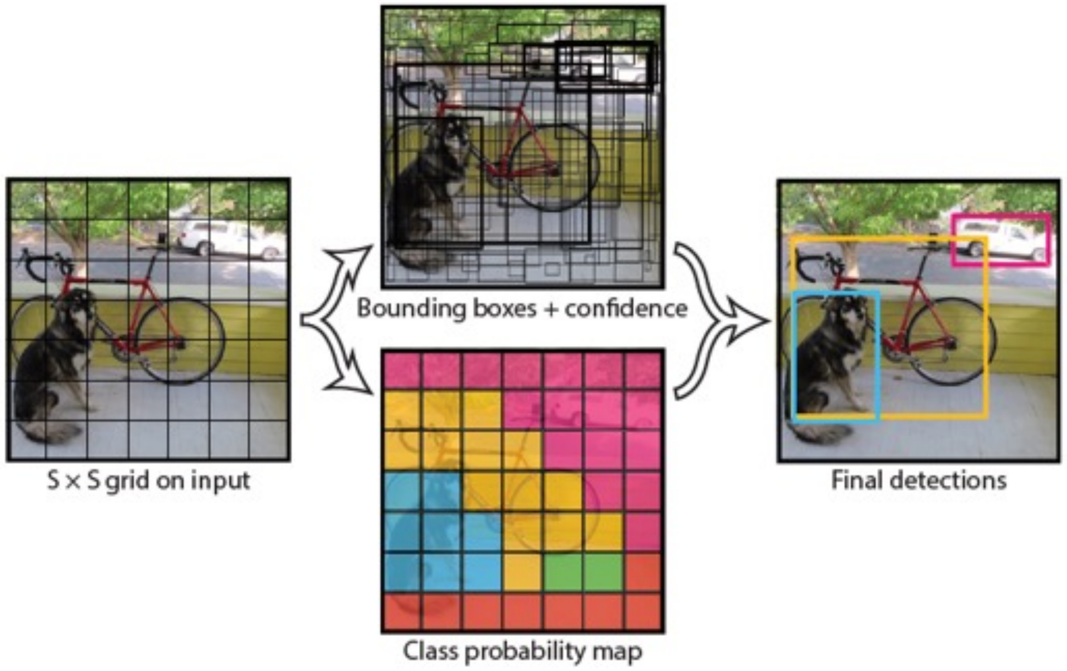
# Single-stage Detector: YOLO

- Redmon et al., You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016
- Defines detection as a regression problem
- Also, a general model for handling regression problems



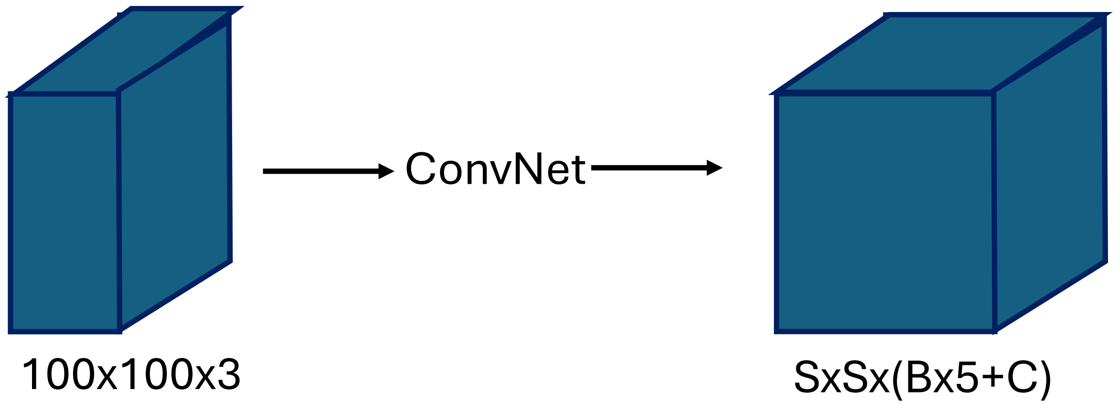
# Single-stage Detector: YOLO

- Divides image into an SxS grid
- Each grid cell predicts whether an object centre is in cell
  - Uses whole image to predict bounding box for each object
  - All classes predicted at once
- Each cell predicts:
  - B bounding boxes (x,y,w,h) and confidences (c) for all classes
    - Centre, width, height
  - C class probabilities
    - If no object, all classes predict 0



# Single-stage Detector: YOLO

- Output encoding
- Predicted tensor: SxSx(Bx5+C)
  - B bounding boxes: $p_b = p(\text{object})$
  - C class probabilities: $p_c = p(\text{class} \mid \text{object})$
  - $[(p_{c1}, p_{c2}, p_{c3}, \ldots, p_{cC}),$
    $(p_{b1}, x_1, y_1, w_1, h_1), \ldots,$
    $(p_{bB}, x_B, y_B, w_B, h_B)]^T$
- Example: empty cell
  - $[(?, ?, ?, \ldots ?),$
    $(0, ?, ?, ?, ?), \ldots,$
    $(0, ?, ?, ?, ?), \ldots]^T$



# Single-stage Detector: YOLO – Losses

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell $i$, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class $c$ in cell $i$.

$\hat{C}_i$ is the box confidence score of the box $j$ in cell $i$.

Square root: power normalisation

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

# Single-stage Detector: YOLO – Losses

- Sum squares
- Square root:
  - Partially normalise for box size
  - Don't want big bounding boxes to dominate
- Different weight for classification vs localization
  - $\lambda_{coord}$ is larger
- Different loss if objects present
  - Avoid training null coordinates, just probability
  - $\lambda_{noobj}$ is smaller

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

# Single-stage Detector: YOLO – NMS

- Non-maximal suppression (NMS) on output
- Each grid cell only gives best two bounding boxes
- Ignore all low probability bounding boxes
- For each class (e.g., pedestrian):
    - Use NMS on final outputs (limit per class overall)

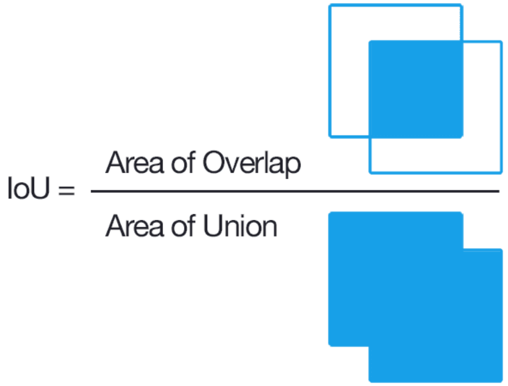
# Single-stage Detector: YOLO

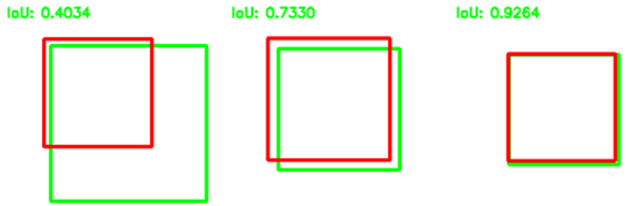
- General approach of bounding boxes is common to all detection architectures
- General approach of regression to estimate continuous numbers in networks
- A good baseline for understanding detection and regression tasks

# Evaluating Detectors

- Intersection over Union (IoU)

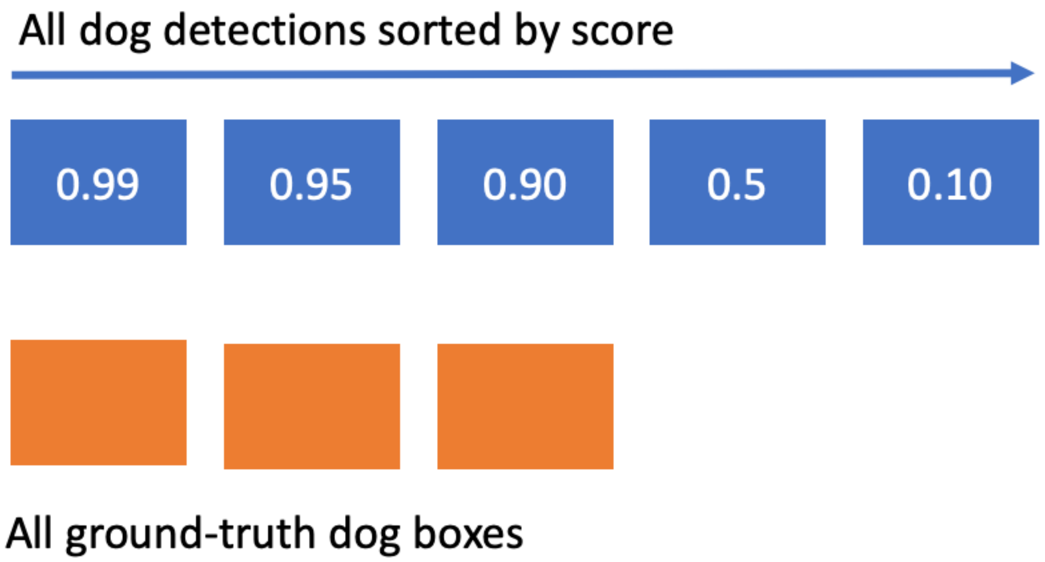
$$\mathrm{IoU} = \frac{|X \cap Y|}{|X \cup Y|}$$



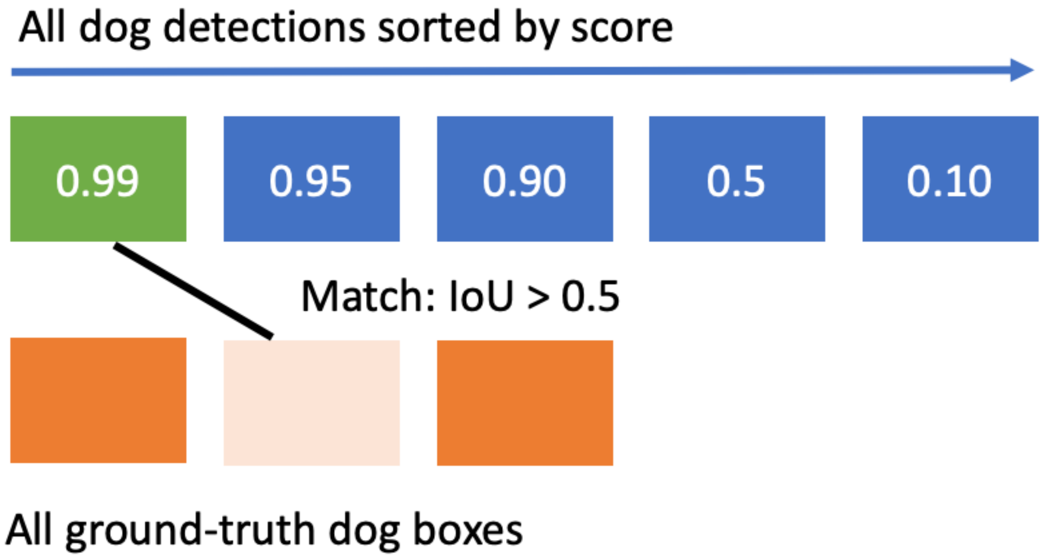

poor good very good

# Evaluating Detectors



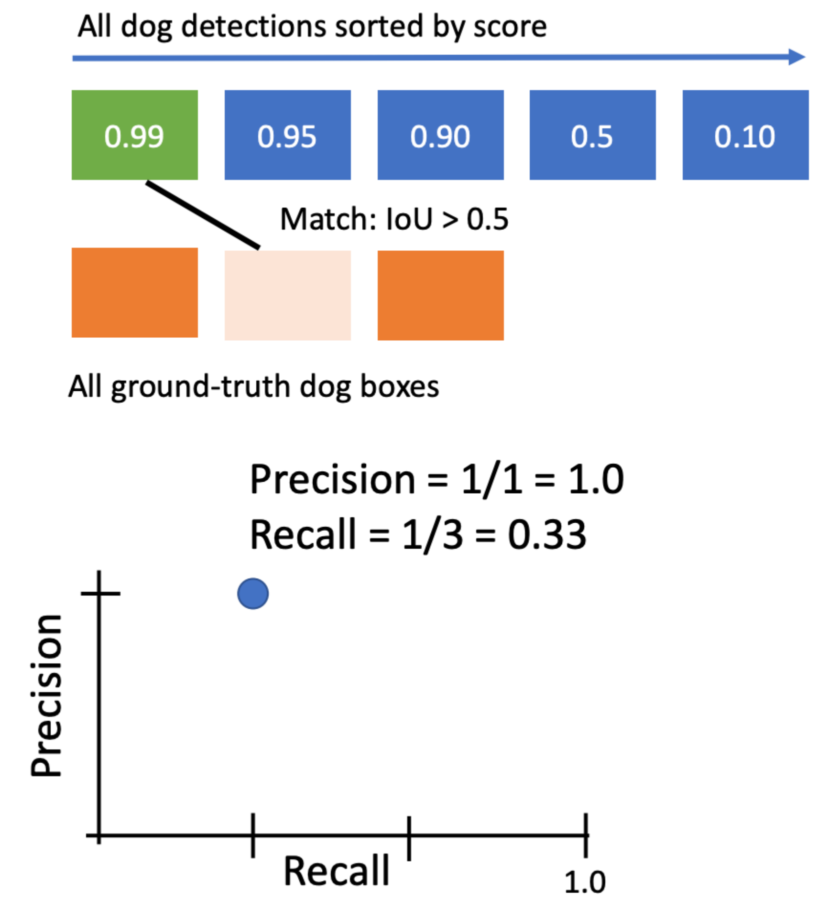

Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors



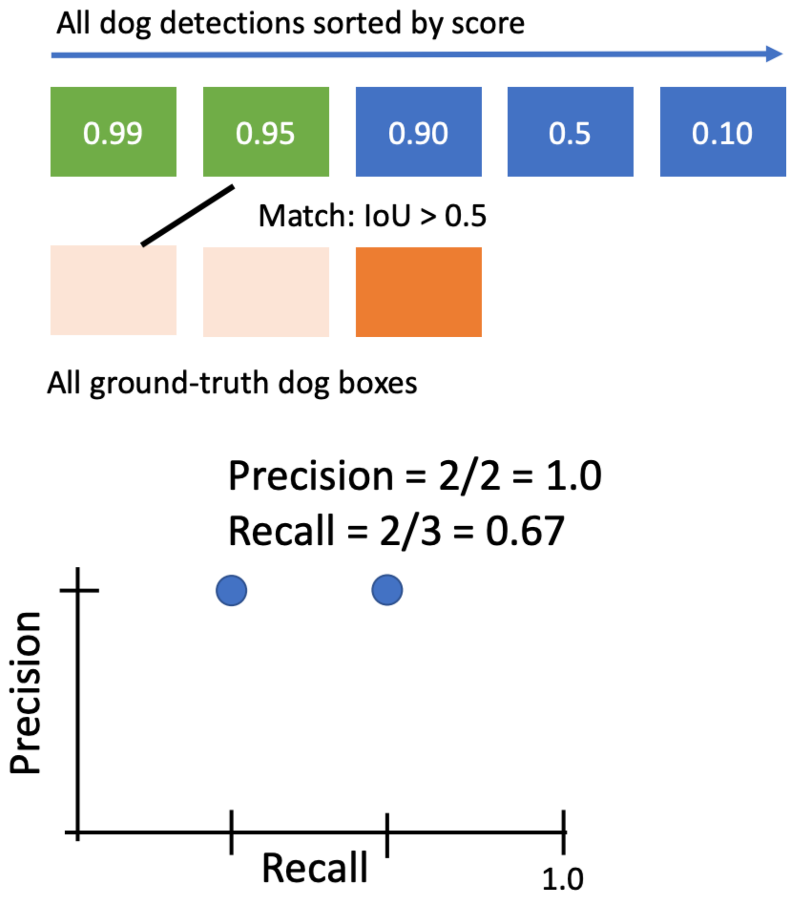

Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors



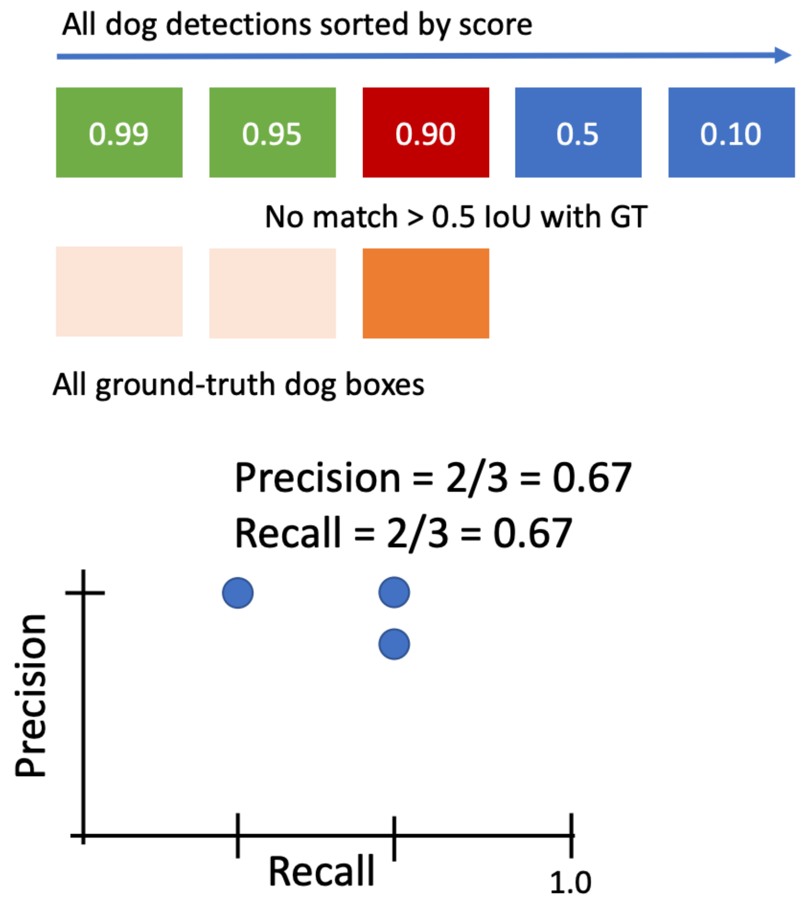

Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors

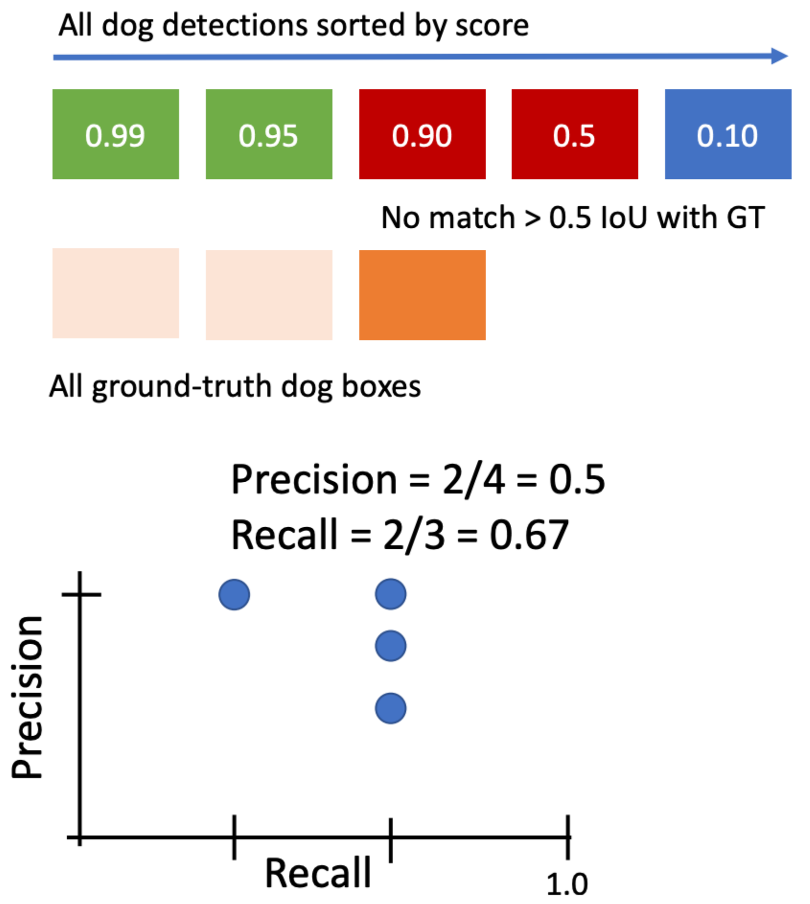



Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors

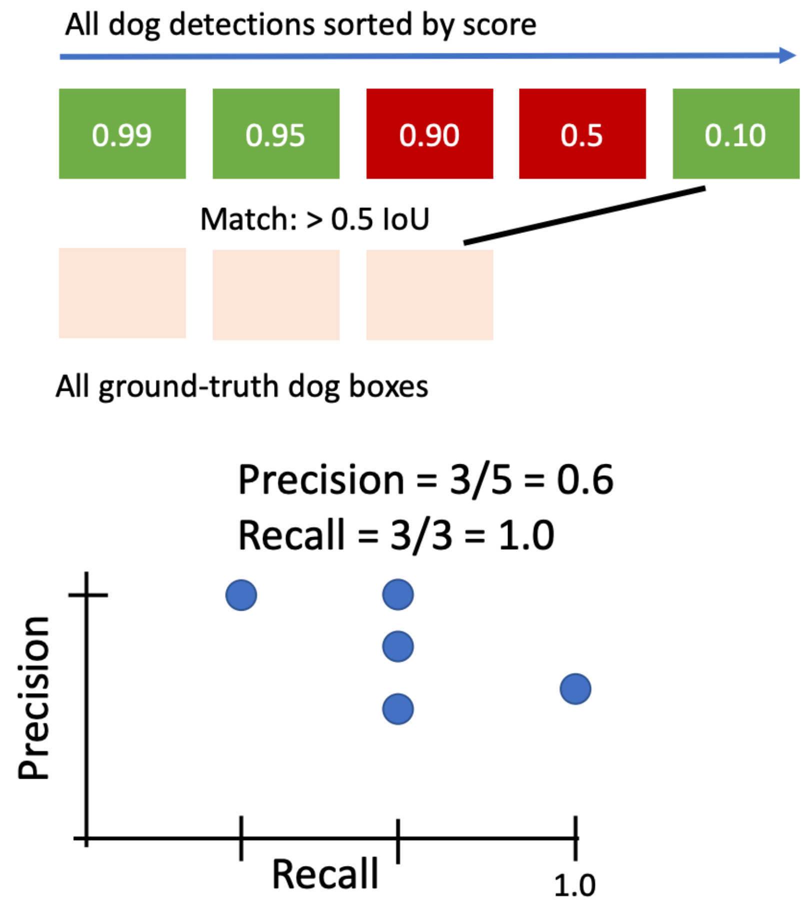


# Evaluating Detectors



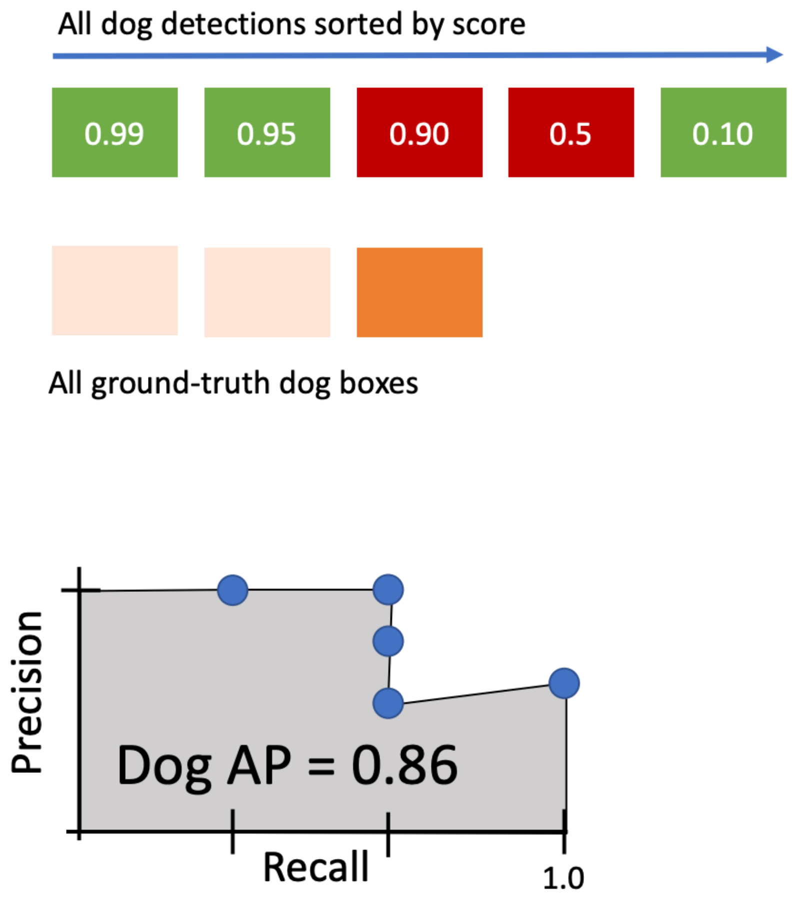

Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors




Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Evaluating Detectors




Car AP = 0.65

Cat AP = 0.80

Dog AP = 0.86

mAP@0.5 = 0.77

Credit: https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture15.pdf

# Image Segmentation

High-level Vision

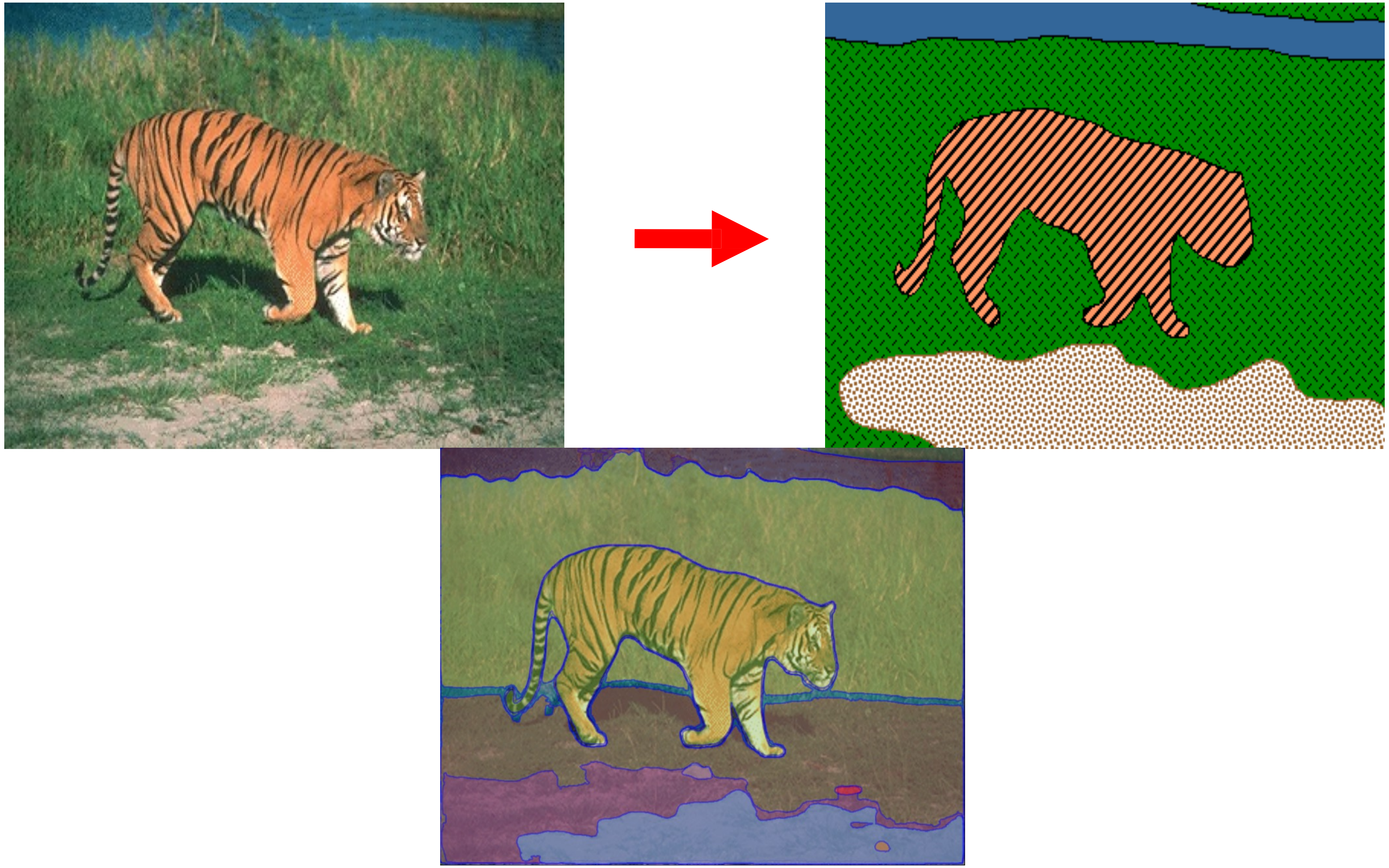
# Image Segmentation: Outline

- Motivation and applications
- Segmentation as a clustering problem
  - K-means
  - Mean shift
- Segmentation as a graph partitioning problem [not covered]
  - Graph-cut (see e.g., Szeliski 5.4-5.5)

# Image Segmentation

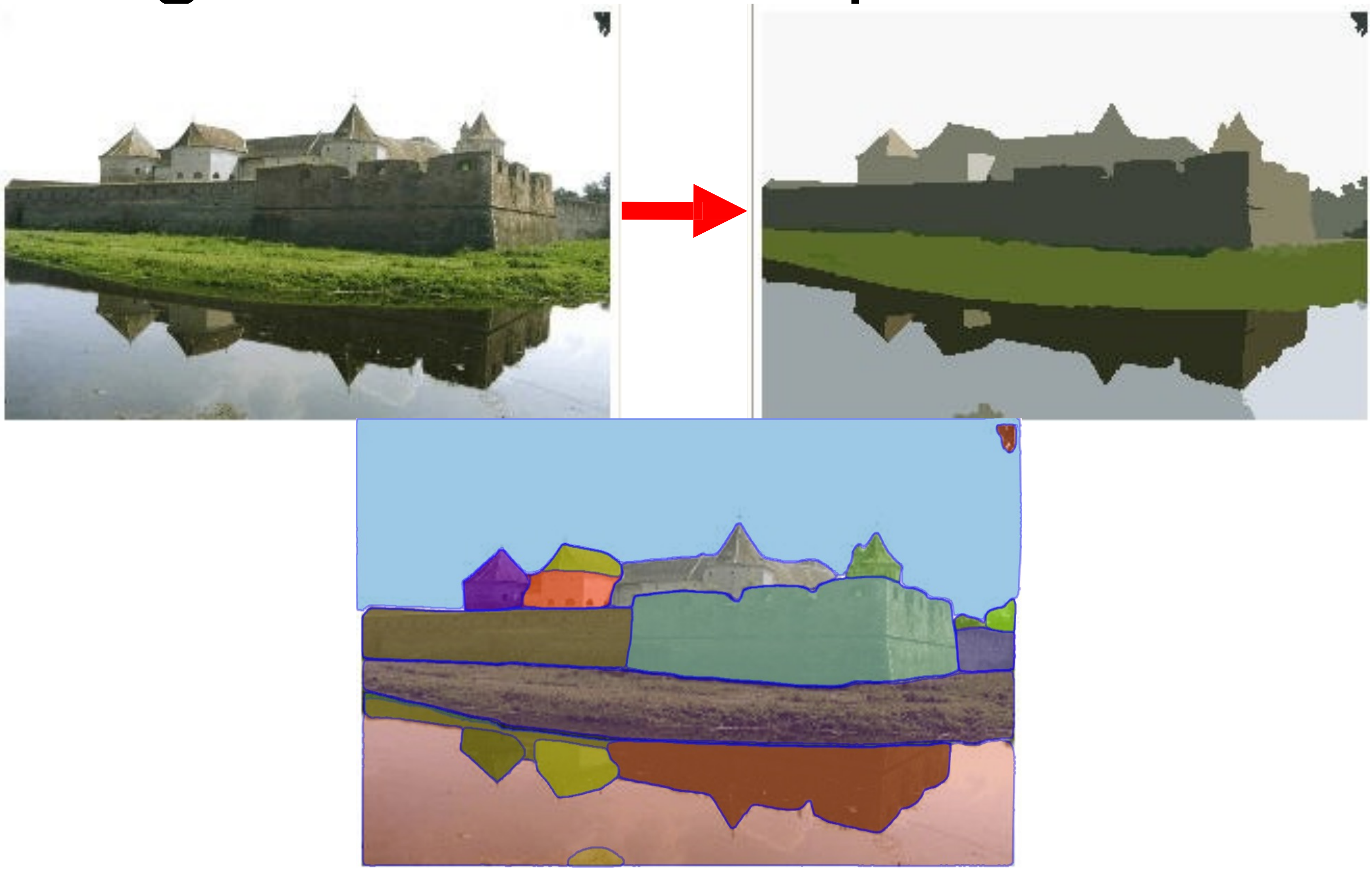
- Goal: break up the image into semantically-meaningful or perceptually-similar regions
- “Semantic segmentation”

# Image Segmentation: Examples



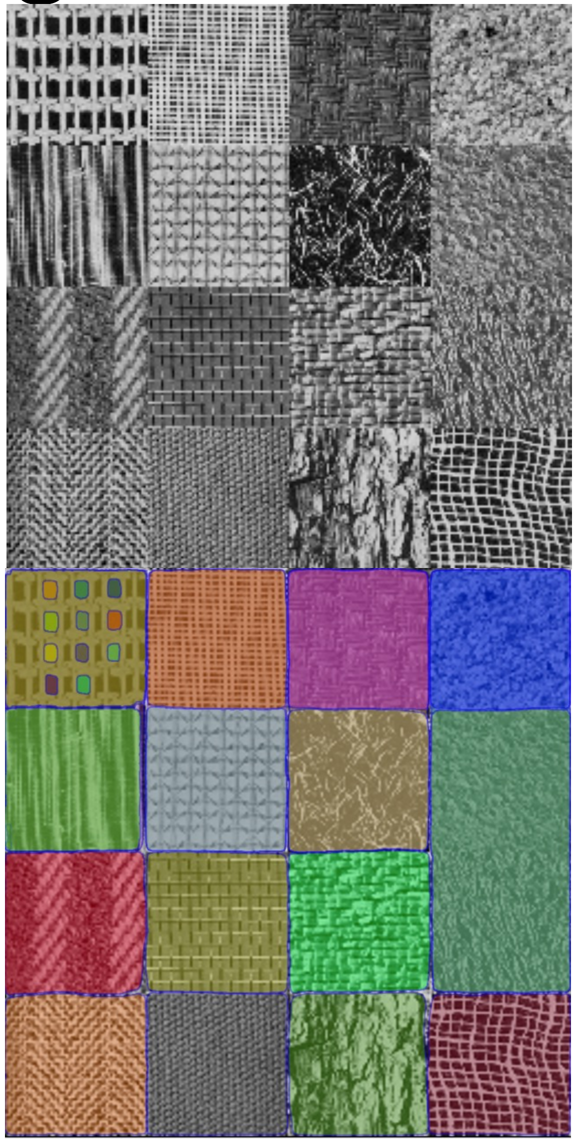
Kirillov et al., “Segment Anything”, ICCV 2023

# Image Segmentation: Examples



Kirillov et al., “Segment Anything”, ICCV 2023

# Image Segmentation: Examples



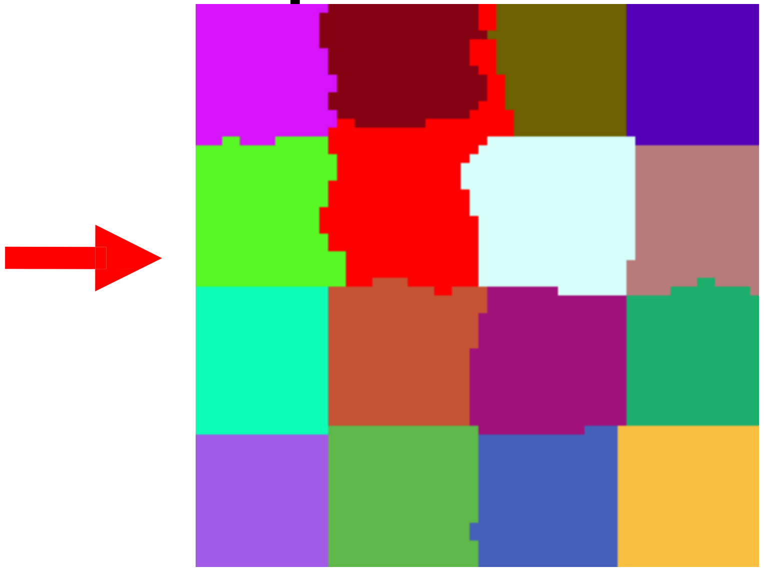
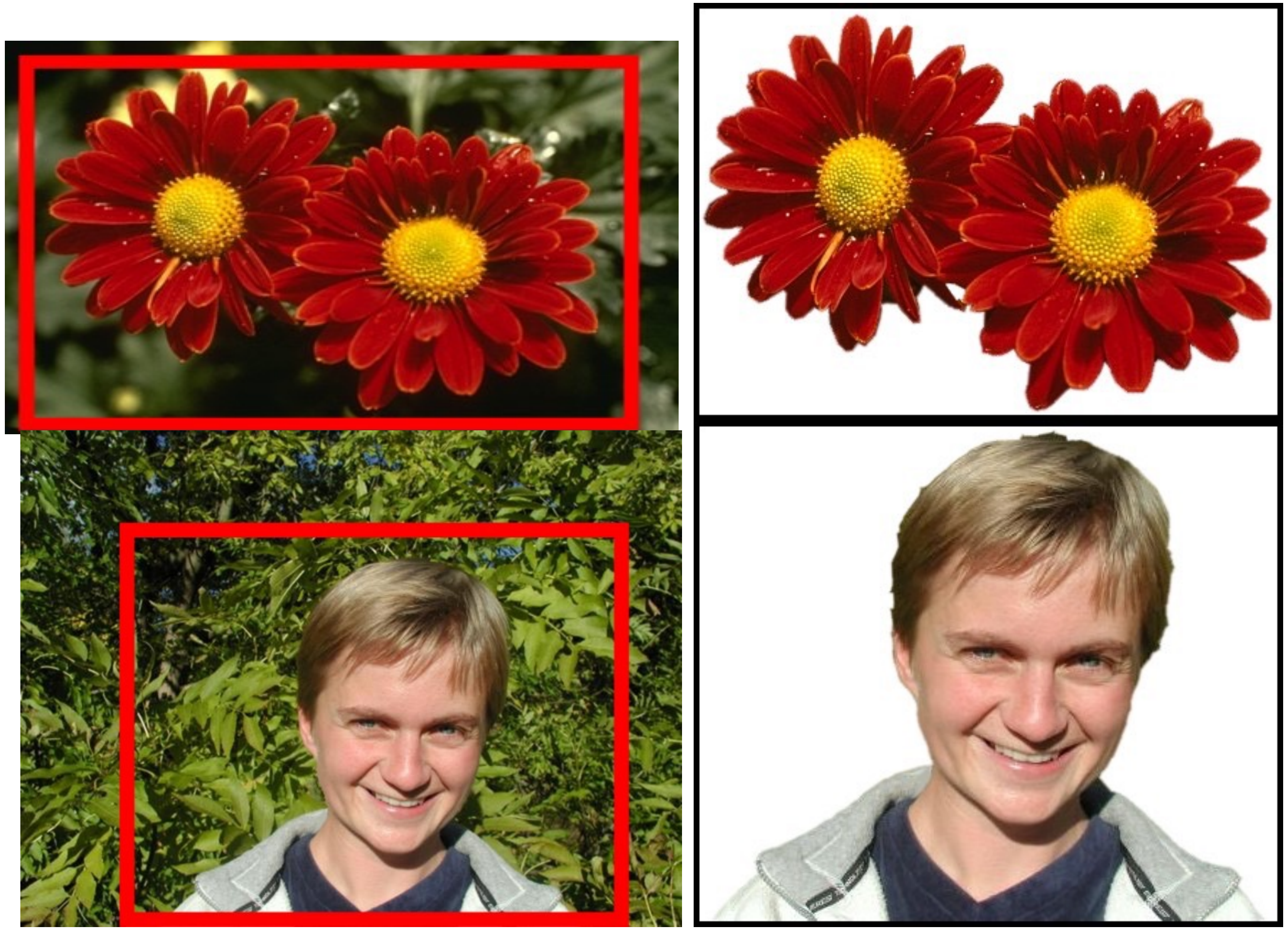


Image Credit: Uni Bonn

Kirillov et al., “Segment Anything”, ICCV 2023

# Applications: Photo Editing

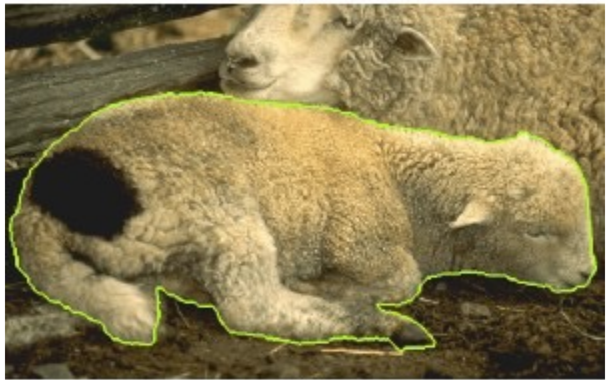


Credit: Rother et al. 2004

# Applications: Photo Montage

(a)

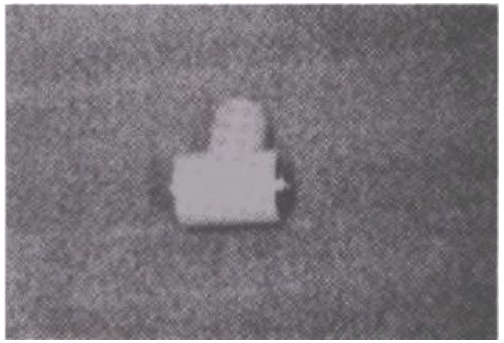
(b)

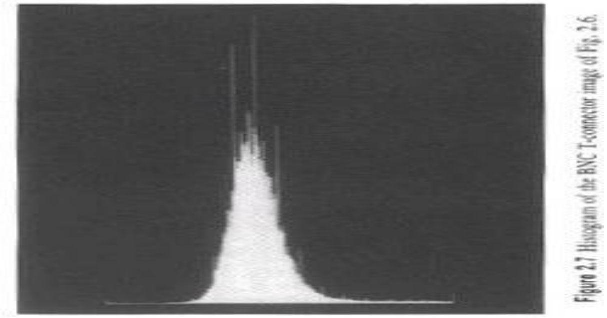


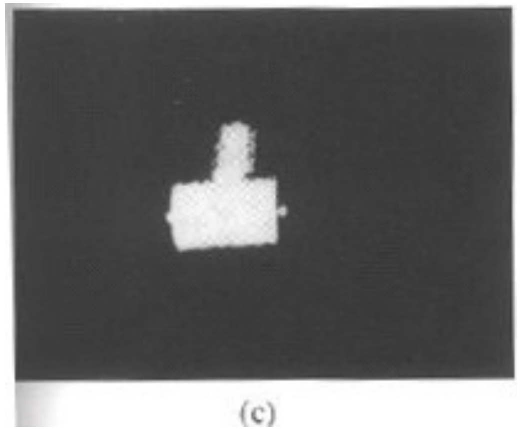
# Image Segmentation

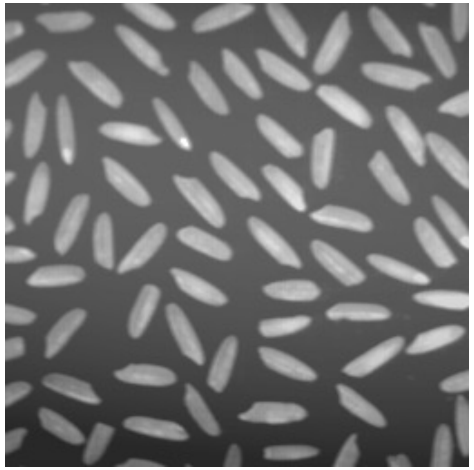
- Uses low- and mid-level visual cues for segmentation
- Partitions the image pixels into groups with perceptually homogeneous or similar pixel properties (e.g., intensity, colour, texture, or similar spatial location)
  - Grouping pixels into perceptually homogeneous regions

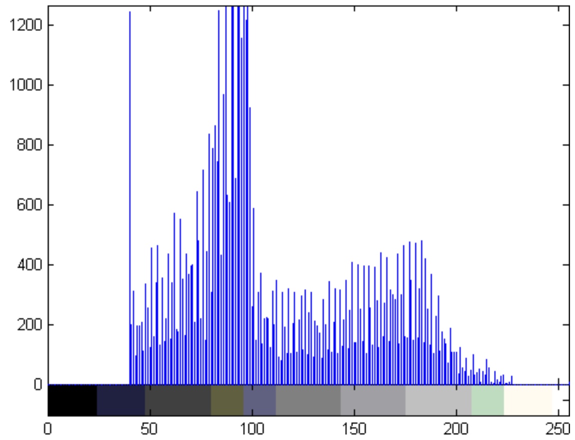
# Approaches: Histogram – Image Binarisation





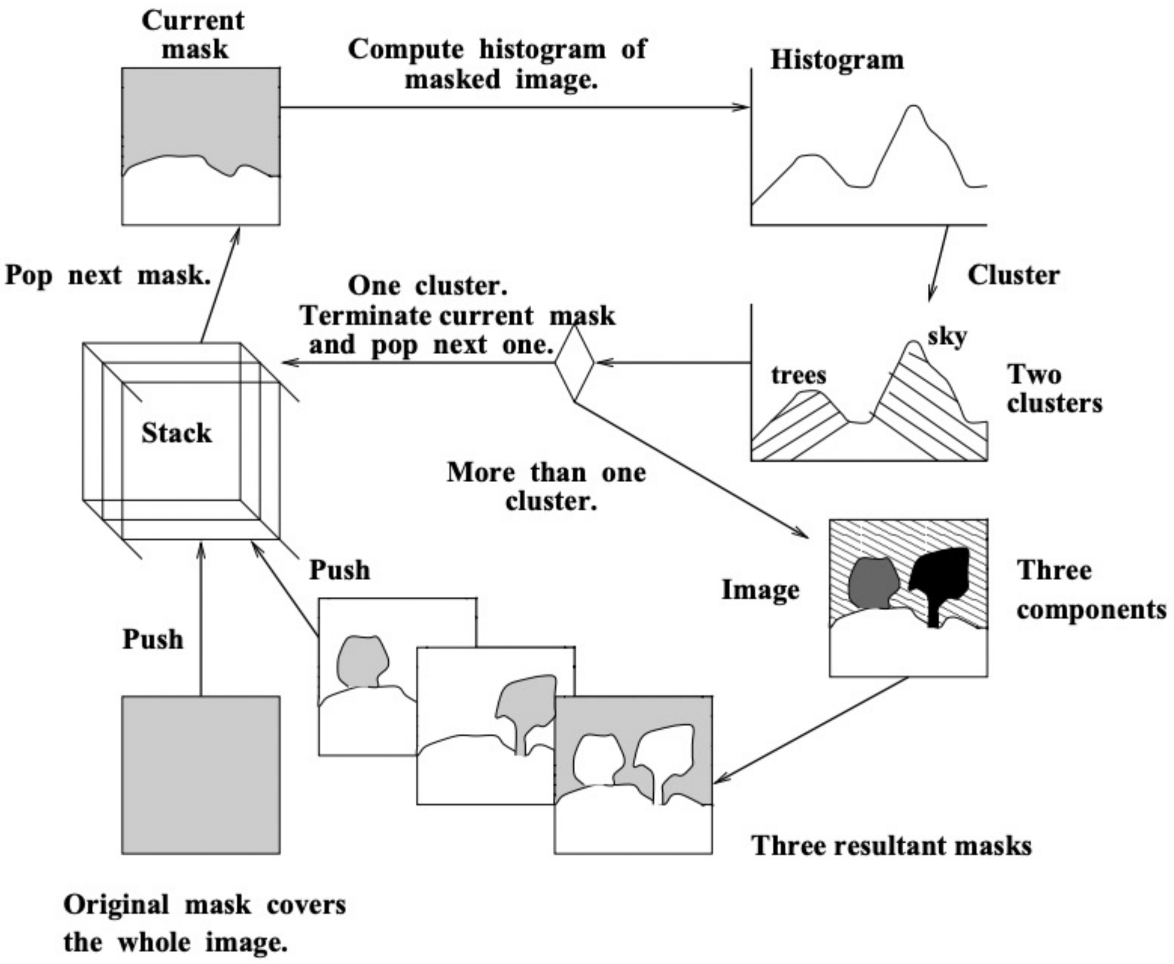



# Approaches: Histogram – Ohlander Algorithm

1. Input a colour image of a scene
2. Start with the whole image
3. Select the R, G, or B histogram with the largest peak and find clusters from that histogram
4. Convert to regions on the image and create masks for each
5. Push each mask onto a stack for further histogram clustering

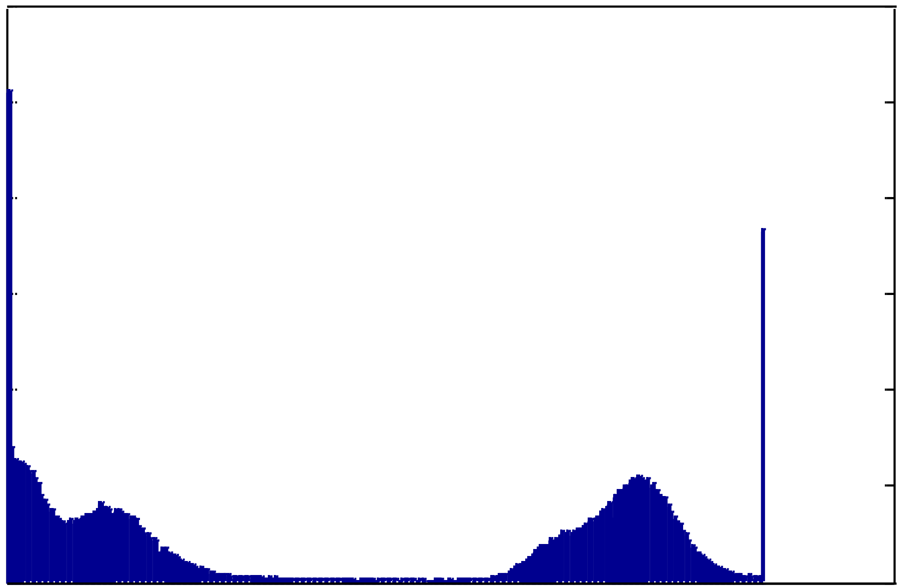
# Approaches: Histogram – Ohlander Algorithm

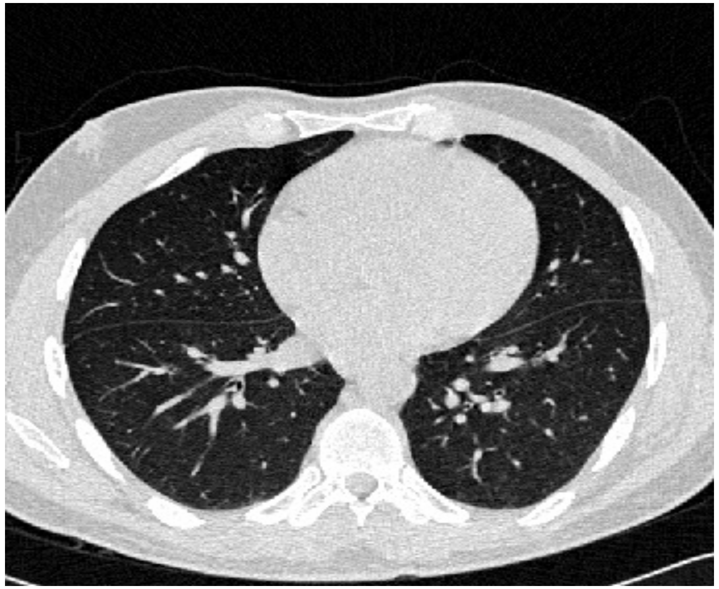
- Recursive histogram-directed spatial-clustering scheme
- The original image has four regions: grass, sky and two trees
- The current mask (upper left) identifies the region containing the sky and the trees
- Clustering its histogram leads to two clusters in color space, one for the sky and one for the trees
- The sky cluster yields one connected component, while the three cluster yields two
- Each of the three connected components become masks that are pushed onto the mask stack for possible further segmentation



# Approaches: Histogram – Ohlander Algorithm

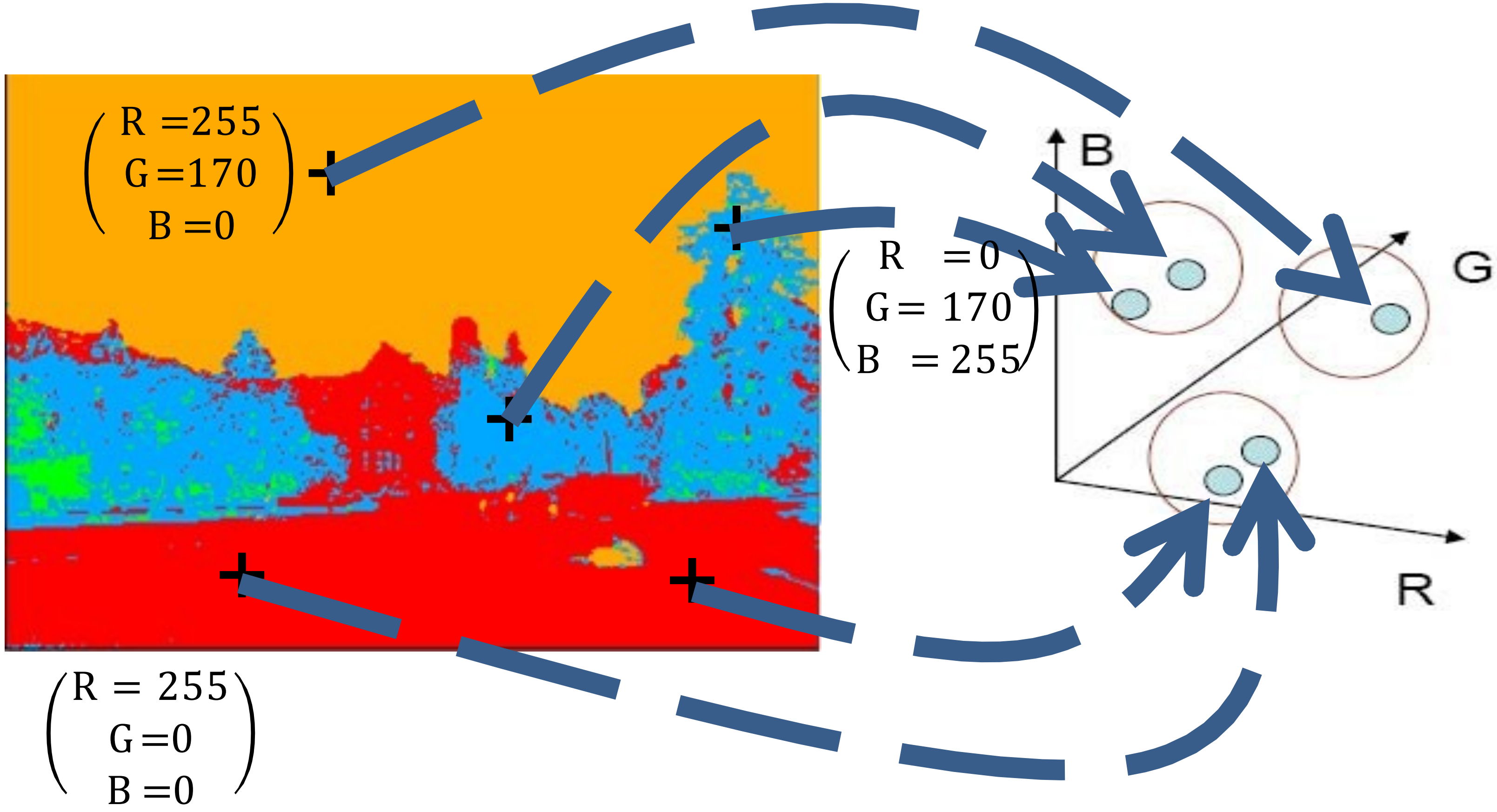
- Histogram for 3 cluster





# Segmentation as Clustering

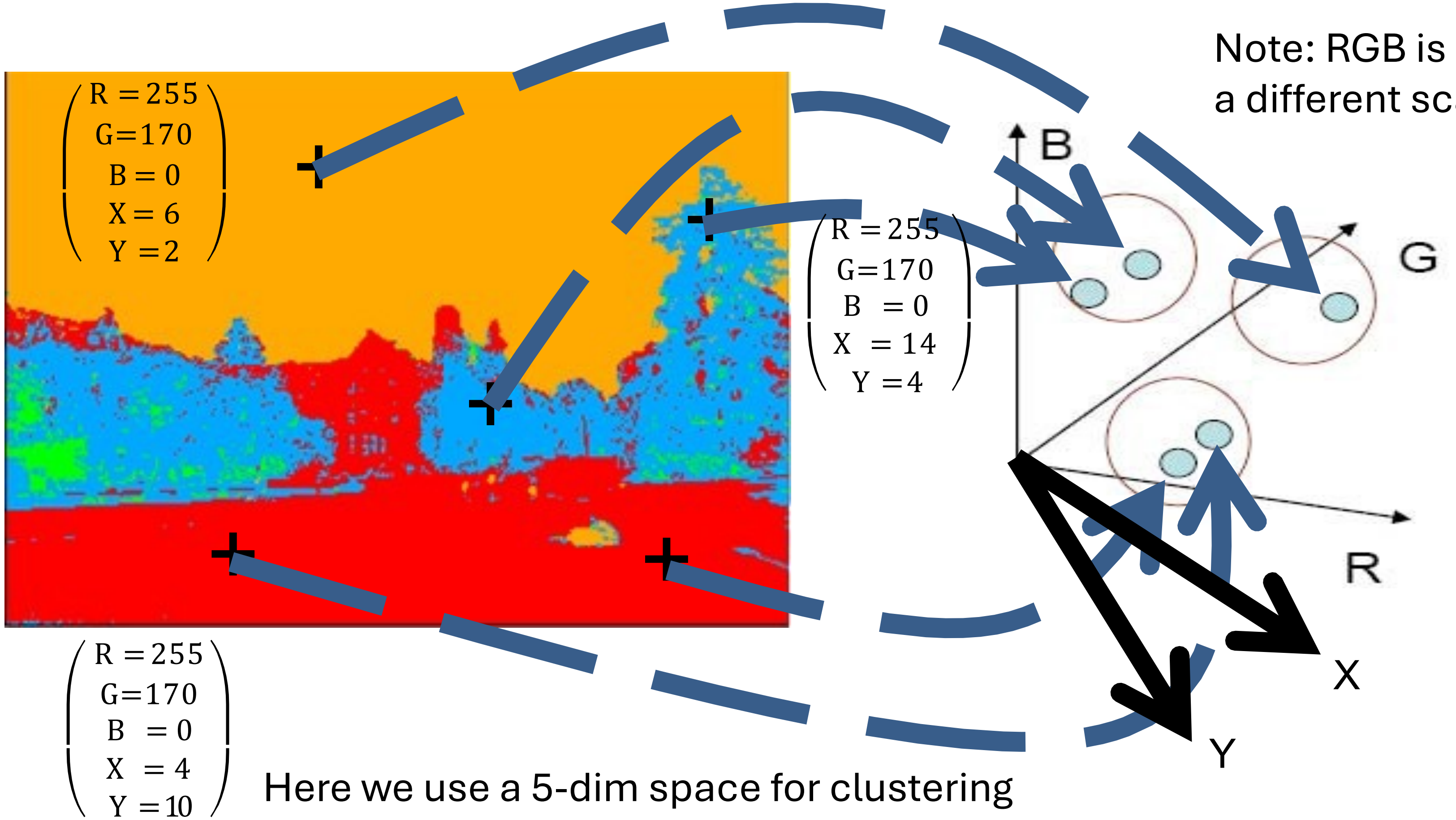
- Cluster similar pixels using colour features only



# Segmentation as Clustering

- Cluster similar pixels using colour + position features (RGB+XY)

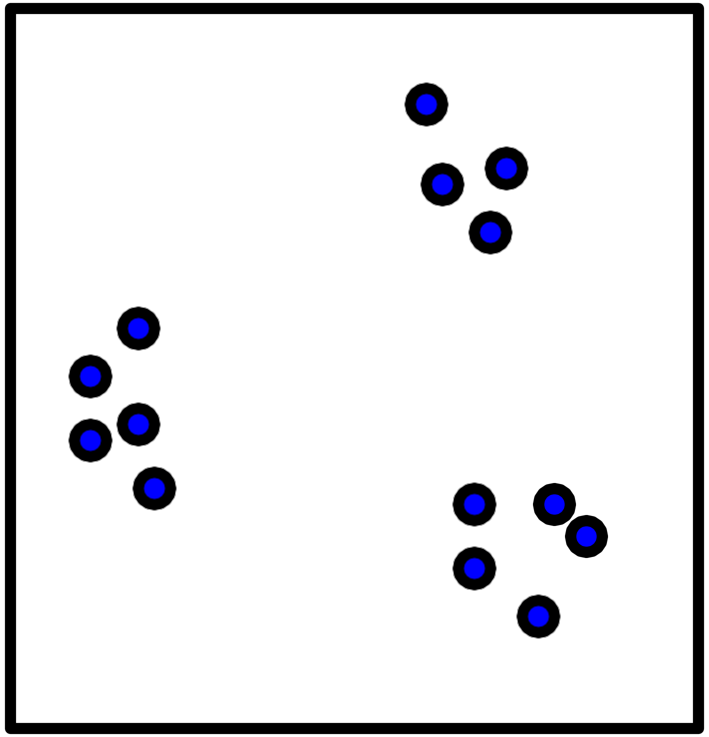
Note: RGB is generally on a different scale to x,y




Here we use a 5-dim space for clustering

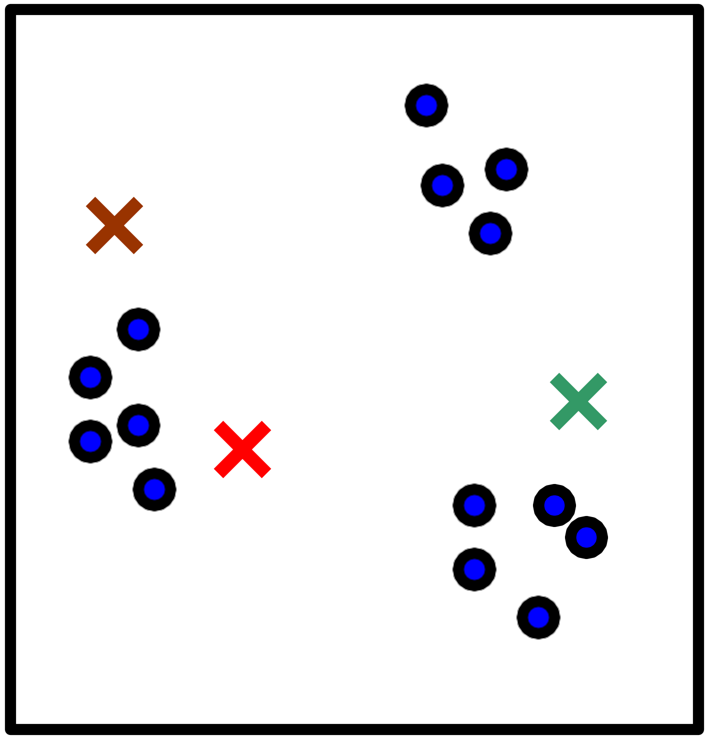
# Approaches: K-means Clustering

- “Guess” the number of clusters K before start



# Approaches: K-means Clustering

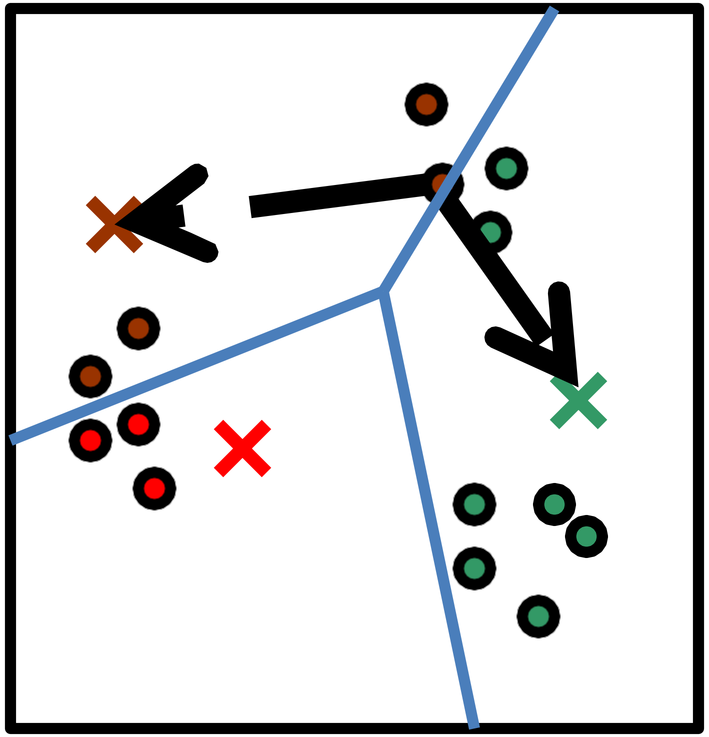
- Start with 3 *random* positions of the cluster *centres*



Iteration = 0

# Approaches: K-means Clustering

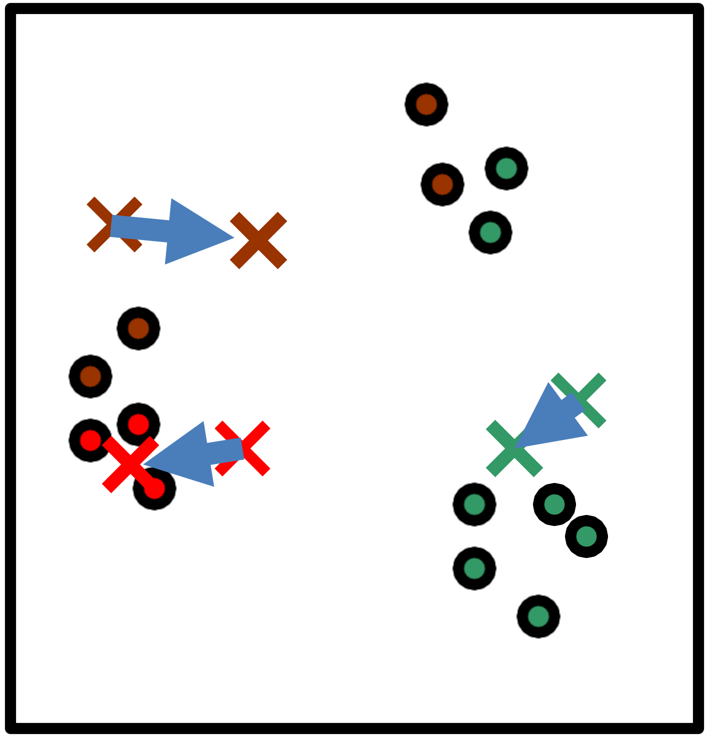
- Start with 3 *random* positions of the cluster *centres*
- By computing the distance, *assign* each data point to the closest centre



Iteration = 1

# Approaches: K-means Clustering

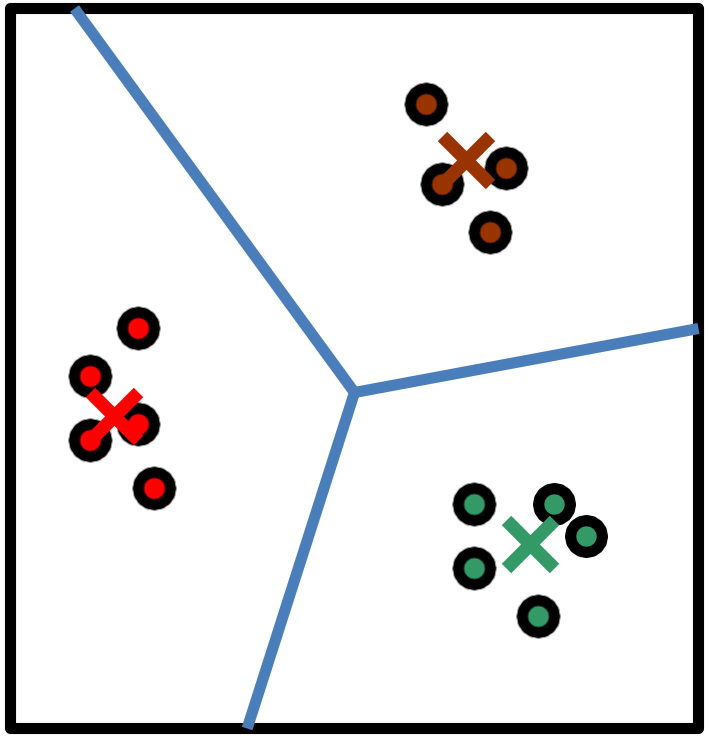
- Start with 3 *random* positions of the cluster *centres*
- By computing the distance, *assign* each data point to the closest centre
- *Re-compute* the centres after the assignments



Iteration = 1

# Approaches: K-means Clustering

- Start with 3 *random* positions of the cluster *centres*
- By computing the distance, *assign* each data point to the closest centre
- *Re-compute* the centres after the assignments
- Iterate until no points are reassigned



Iteration = 3

# Approaches: K-means Clustering

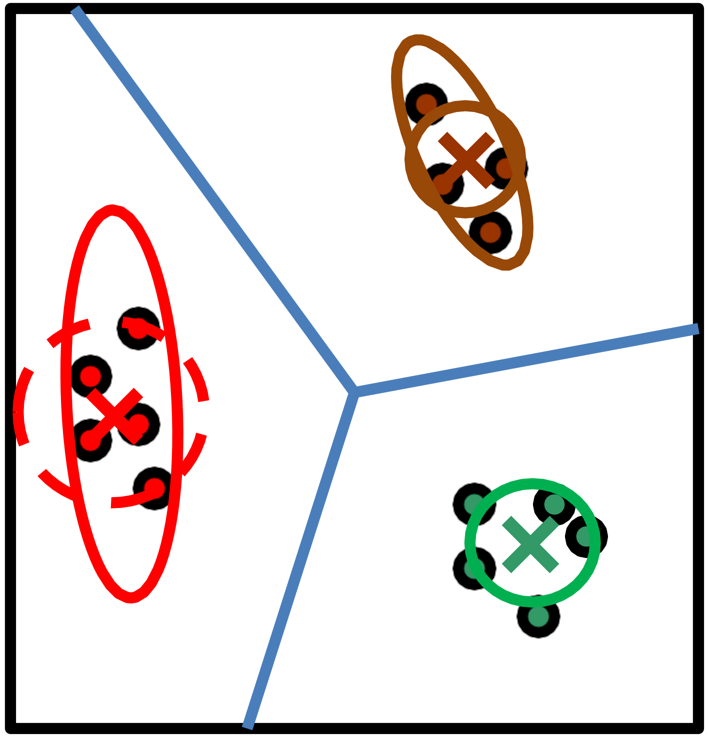
- What have we optimised?

- We have selected the means and membership which minimise the sum of squared distances to the centre means (centroids)
- The sum of squared distances to the means is the **variance** of the cluster

# Approaches: K-means Clustering

- Spherical variance is used

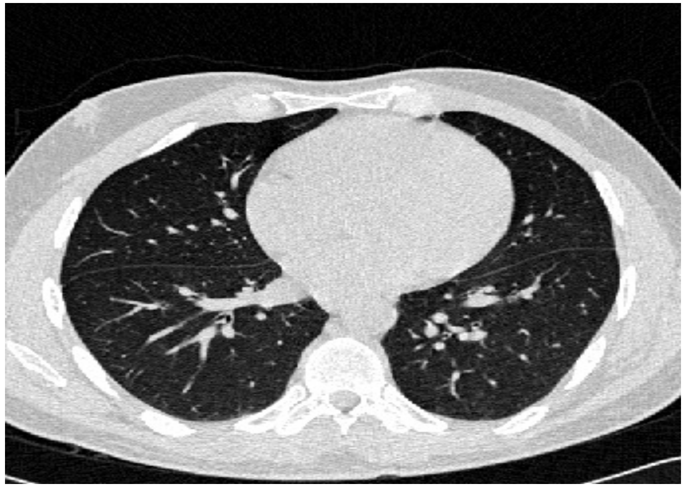
- Q: What about the elongated variances?
  - We can have multivariate Gaussians, or Gaussian mixture clustering
  - e.g., colour vs pixel distance



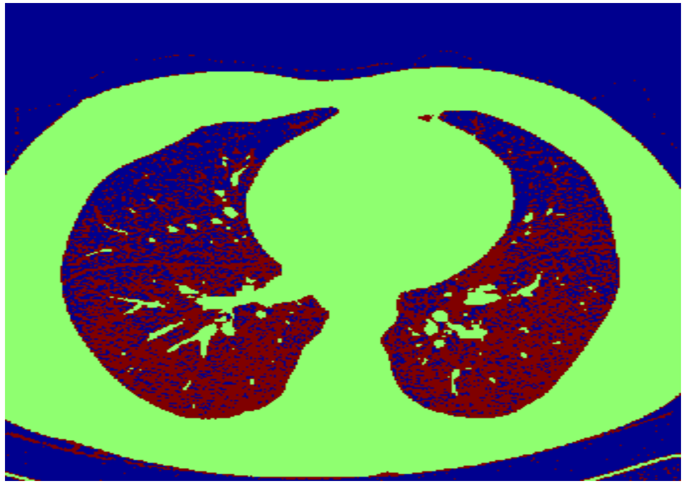
# Approaches: K-means Clustering

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

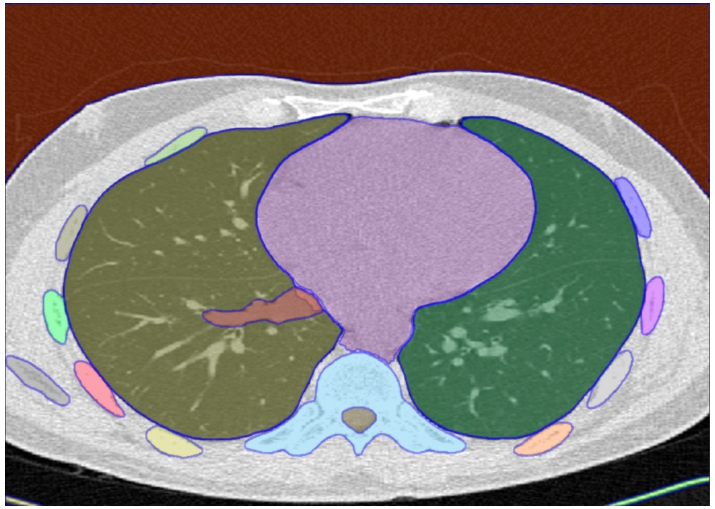
# Approaches: K-means Clustering – Colour



Input image (*I*)



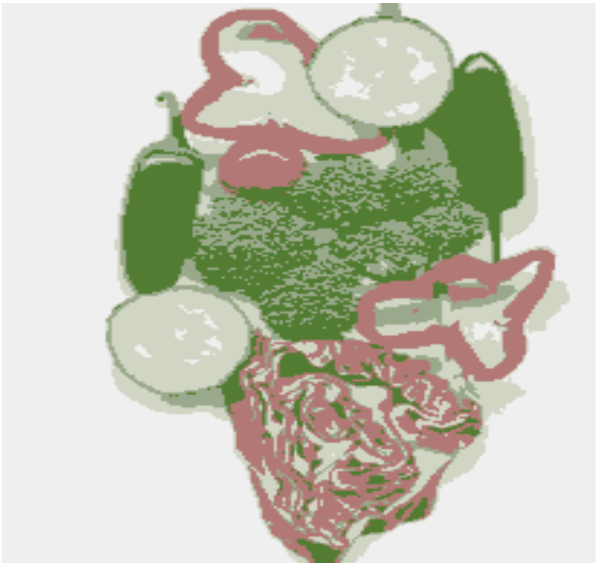
Three-cluster image using the gray levels of *input image*



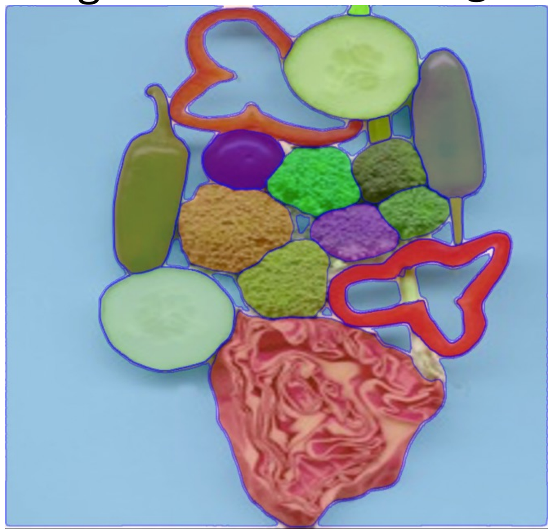
“Segment Anything”

# Approaches: K-means Clustering – Colour

Colour Image



Segmented on color



"Segment Anything"

# Approaches: K-means Clustering – Pros/Cons

**Advantages**

- Finds cluster centres that minimise variance (good representation of data)
- Simple to implement
- Widespread applications

**Disadvantages**

- All clusters have a spherical distribution (isotropic)
- Hard membership/assignment (i.e., 1 or 0 membership)
- Prone to local minima
- Need to choose K
- Can be very slow: each iteration is O(KN) for N-dimensional points

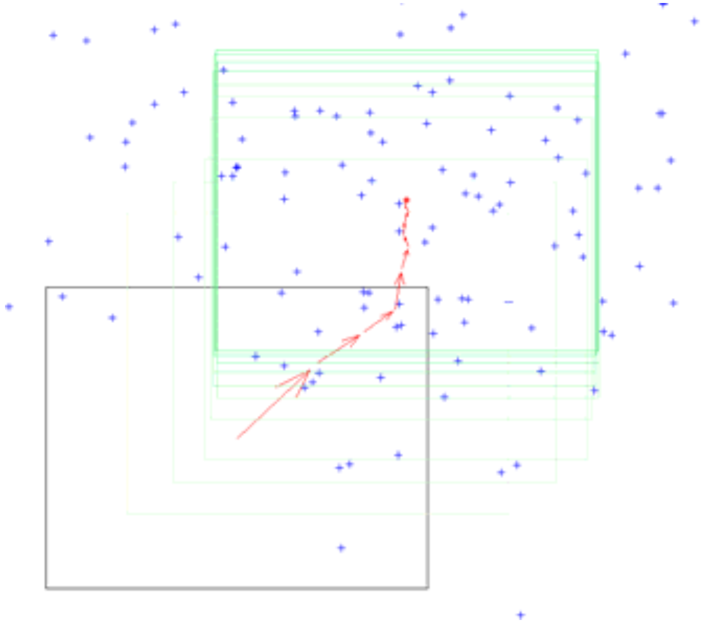
# Approaches: Mean-Shift Clustering

- An advanced and versatile technique for clustering-based segmentation
- The mean shift algorithm seeks a *mode* or local maximum of density of a given distribution
- Perform by computing the colour histogram, looking for modes

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002

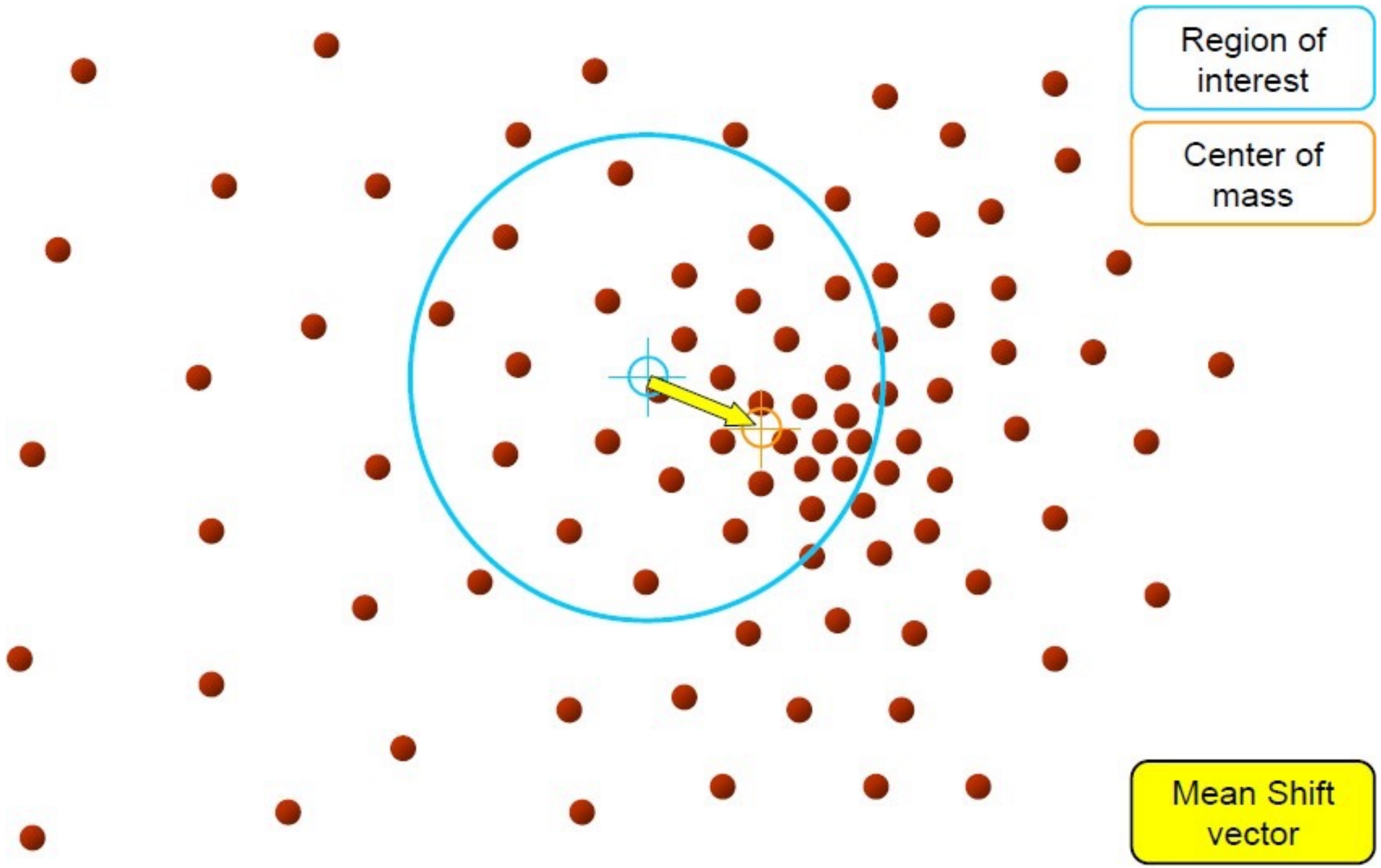
# Approaches: Mean-Shift Clustering

1. Choose a search window (width and location)
2. Compute the mean of the data in the search window
3. Centre the search window at the new mean  location
4. Repeat until convergence



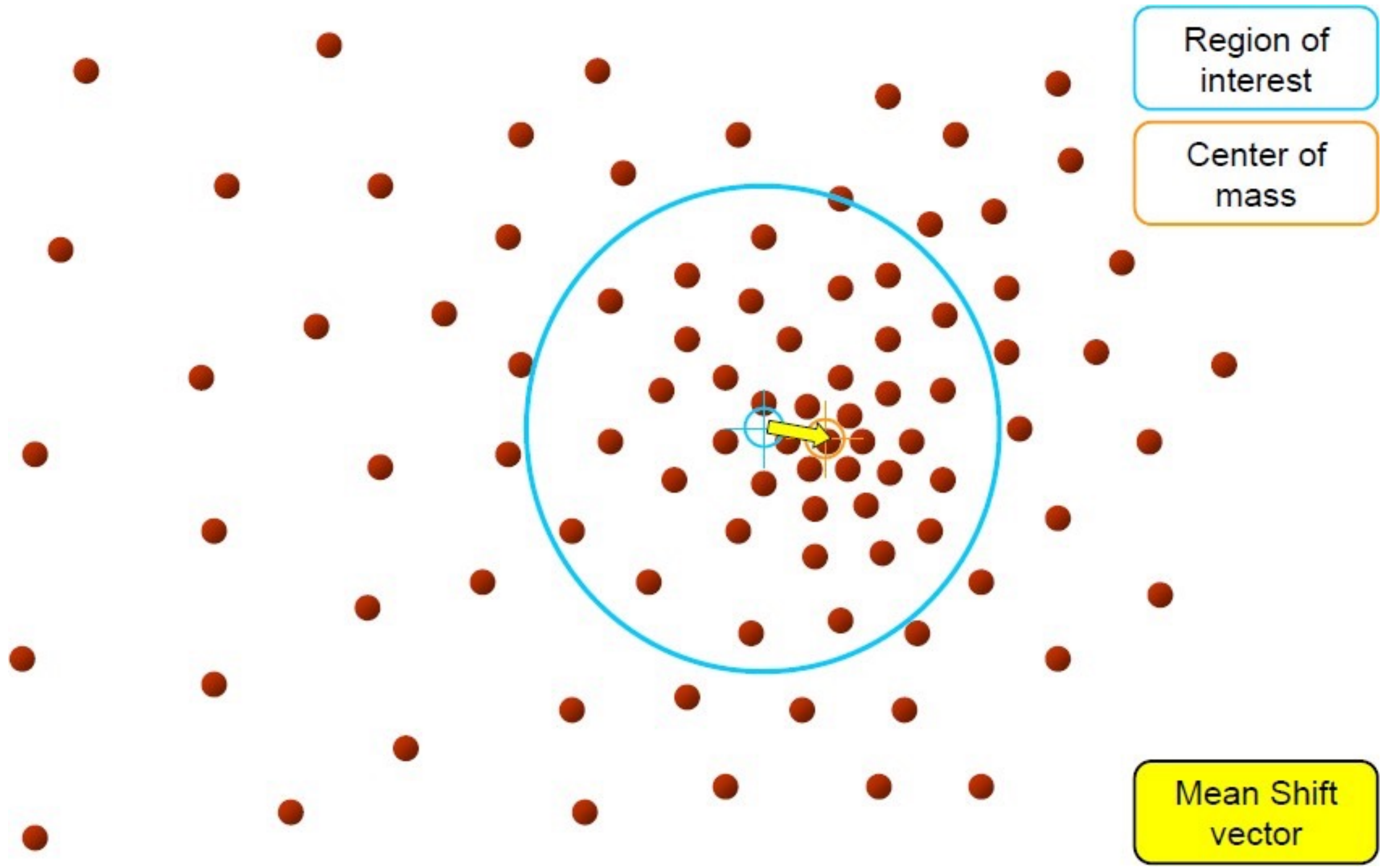
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature  Space Analysis, PAMI 2002

# Approaches: Mean-Shift Clustering

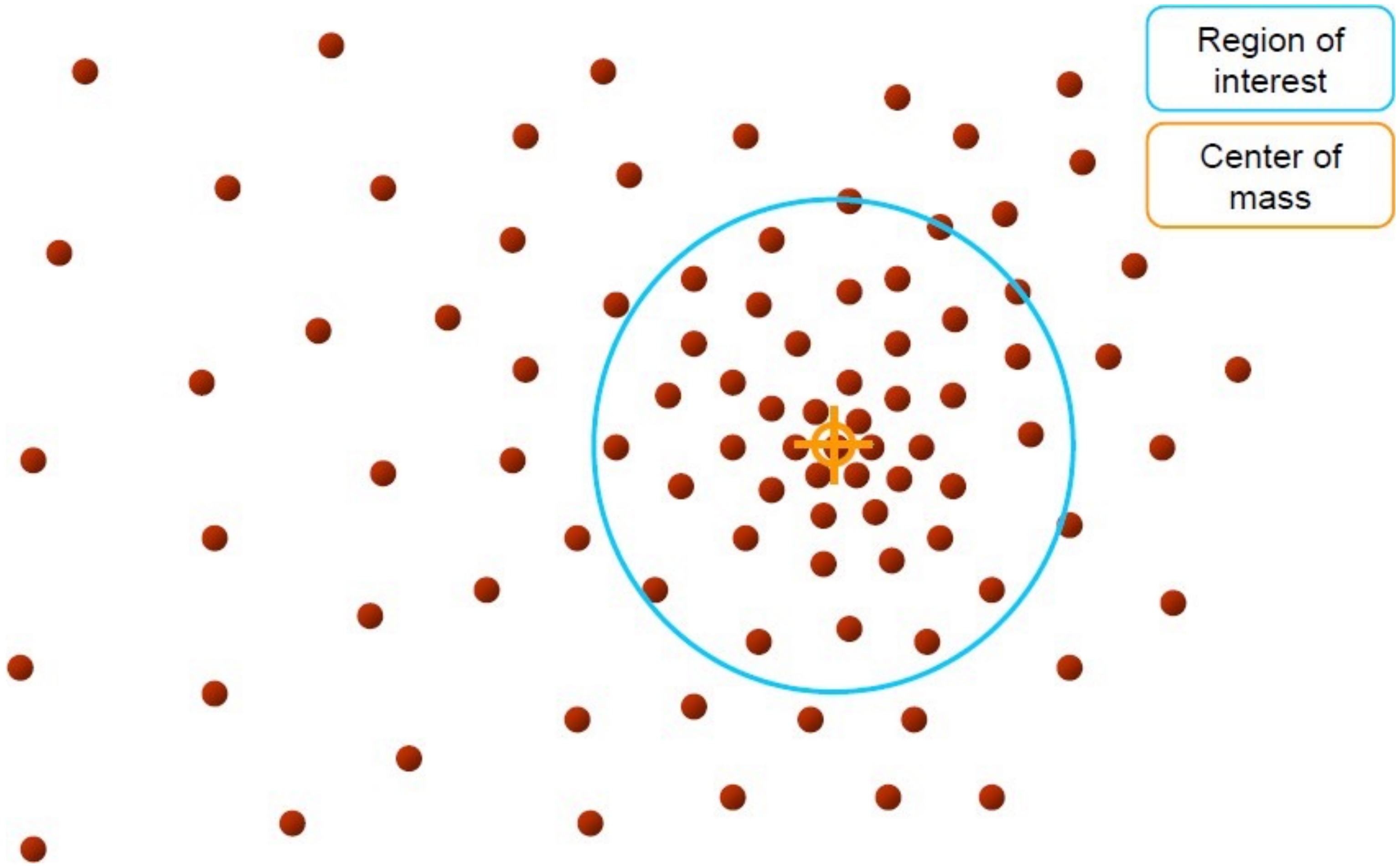



Slide Credit: Y. Ukrainitz & B. Sarel

# Approaches: Mean-Shift Clustering

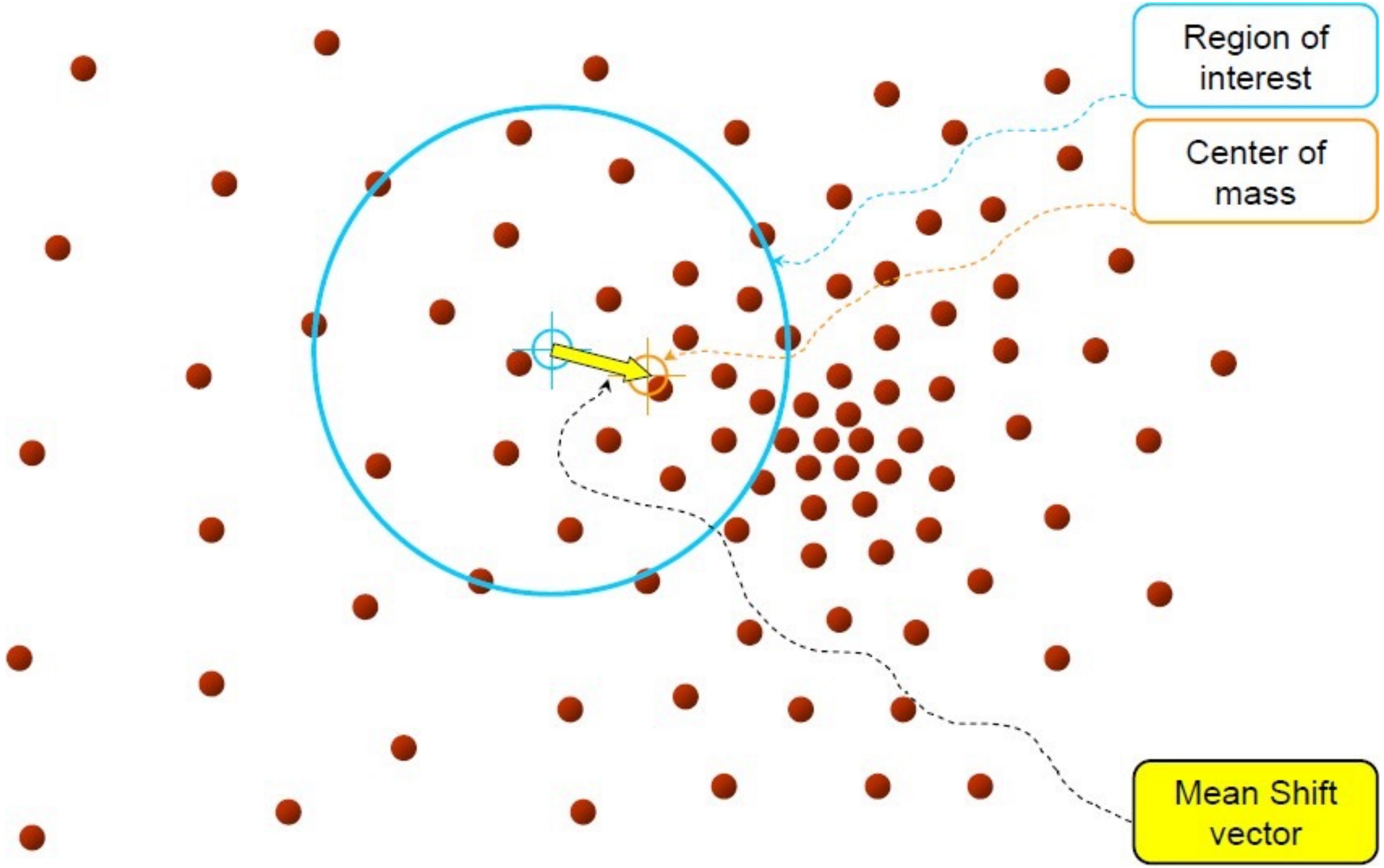


# Approaches: Mean-Shift Clustering




Slide Credit: Y. Ukrainitz & B. Sarel

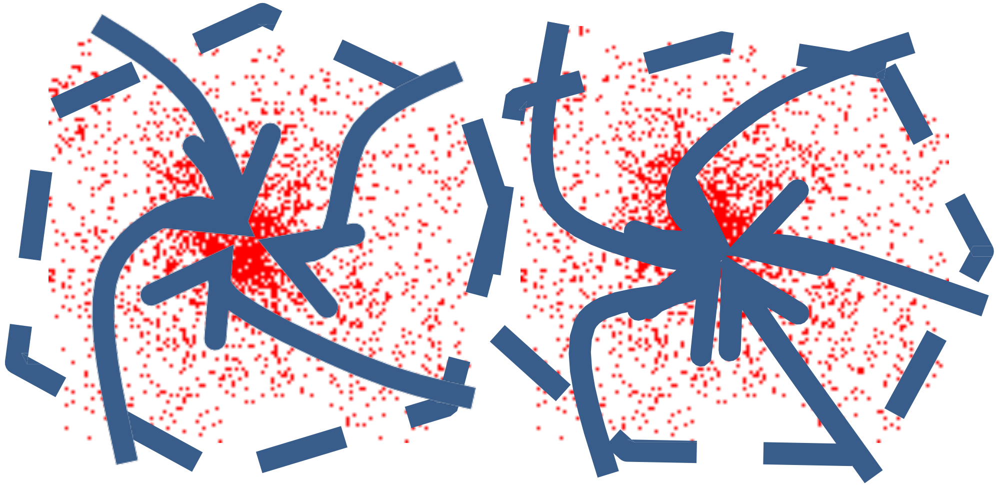
# Approaches: Mean-Shift Clustering



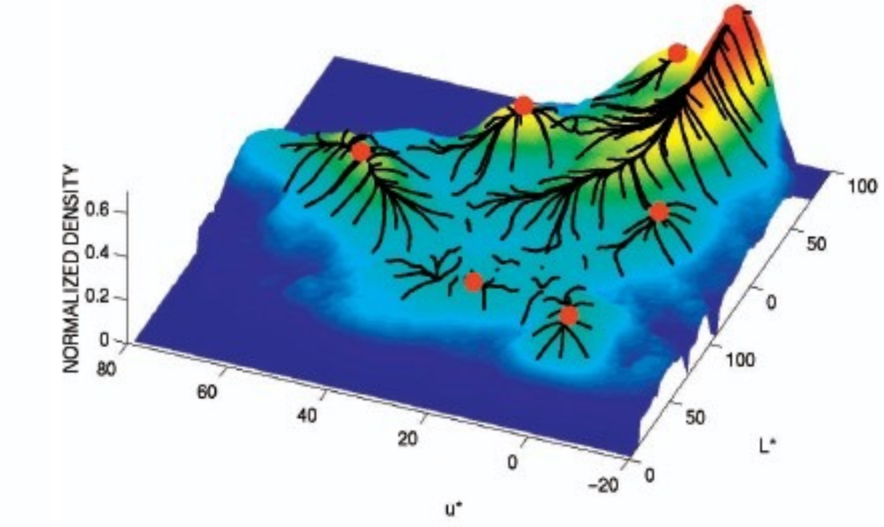

Slide Credit: Y. Ukrainitz & B. Sarel

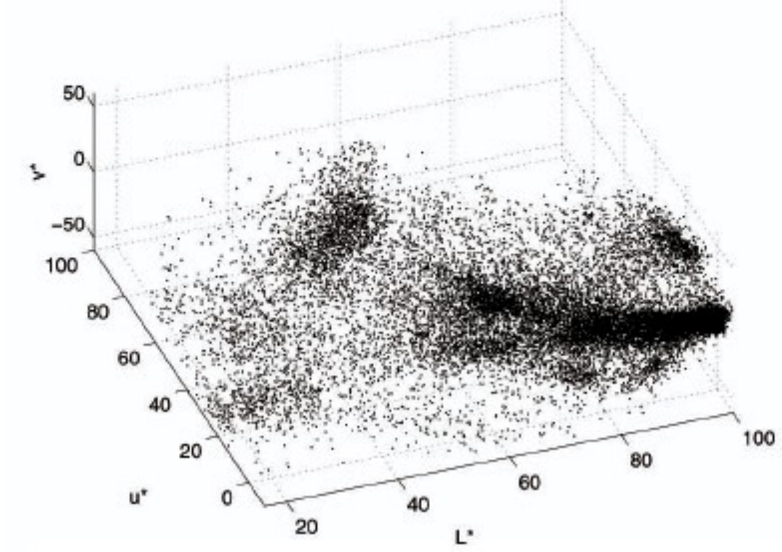
# Approaches: Mean-Shift Clustering

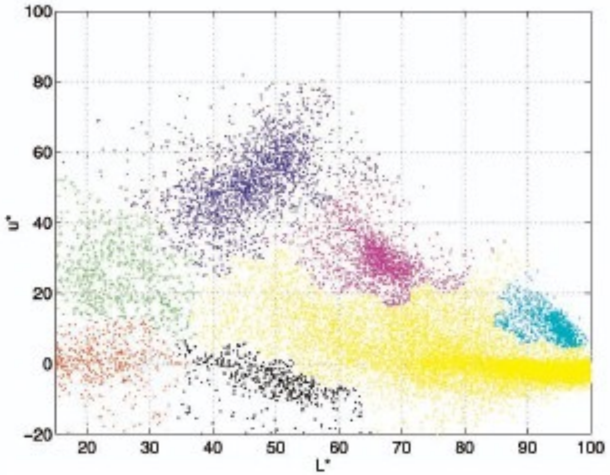
- **Attraction basin:** the region for which all trajectories lead to the same mode
- **Cluster:** all data points in the attraction basin of a mode



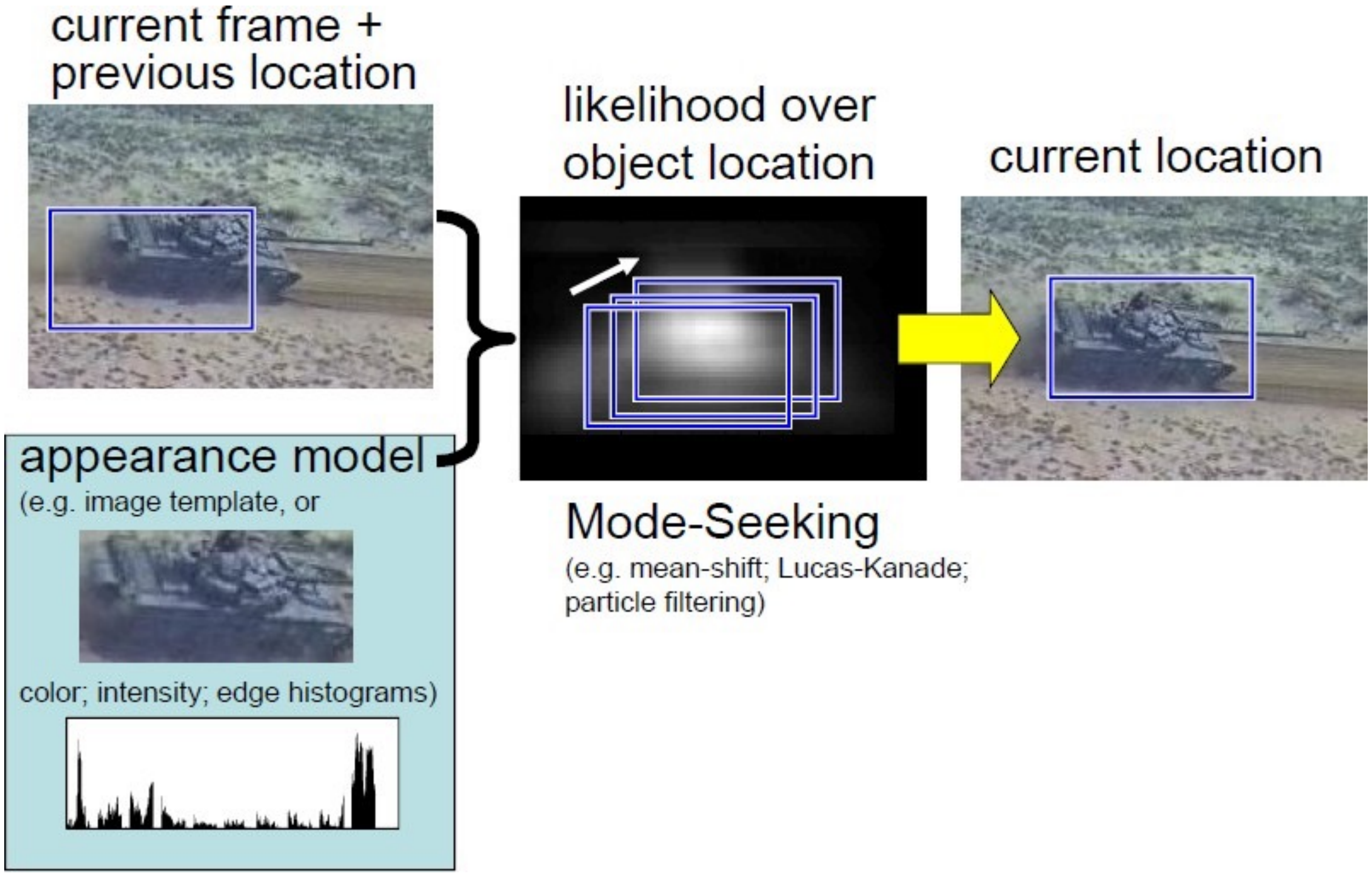
# Approaches: Mean-Shift Clustering



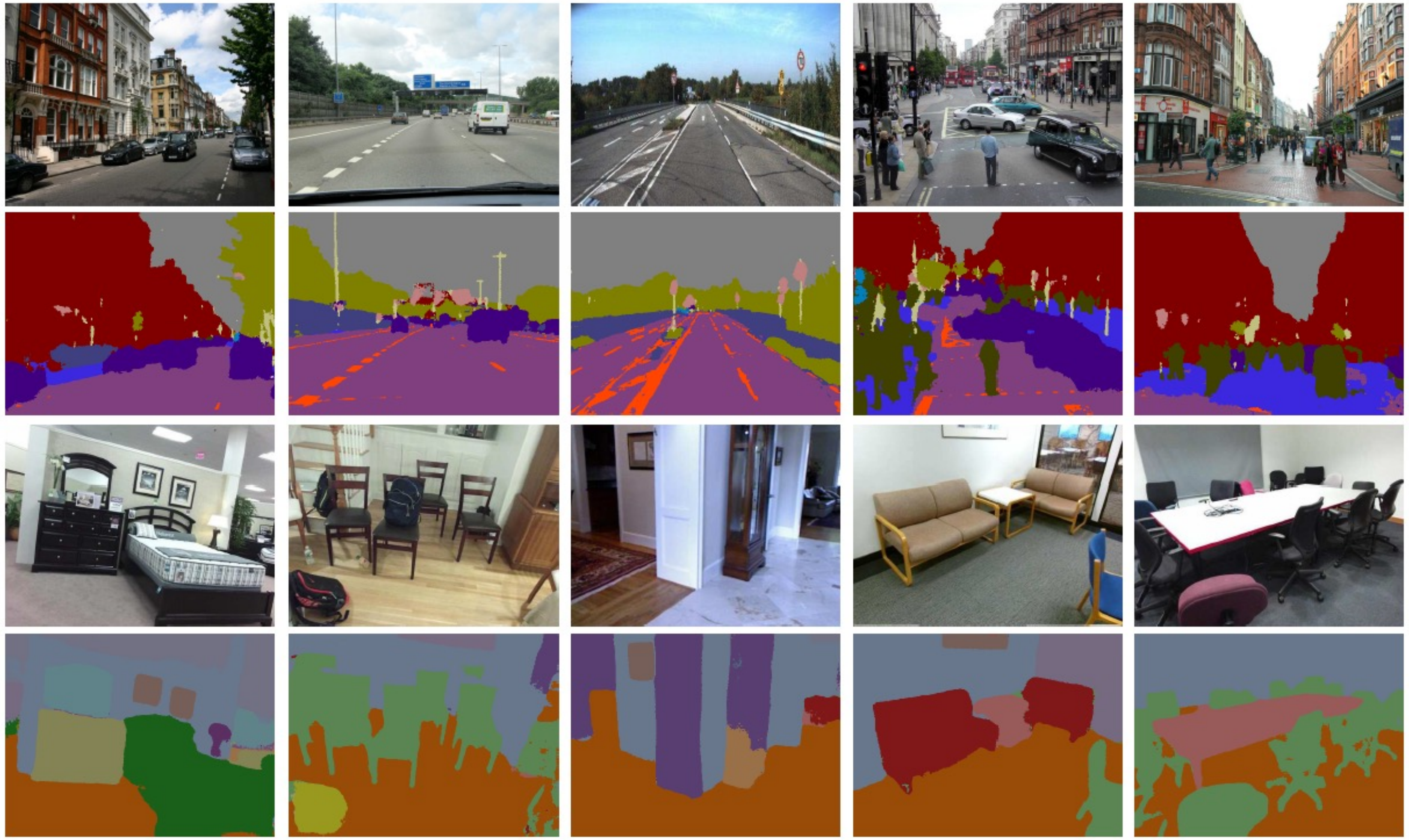



# Approaches: Mean-Shift Tracking



# Approaches: CNNs

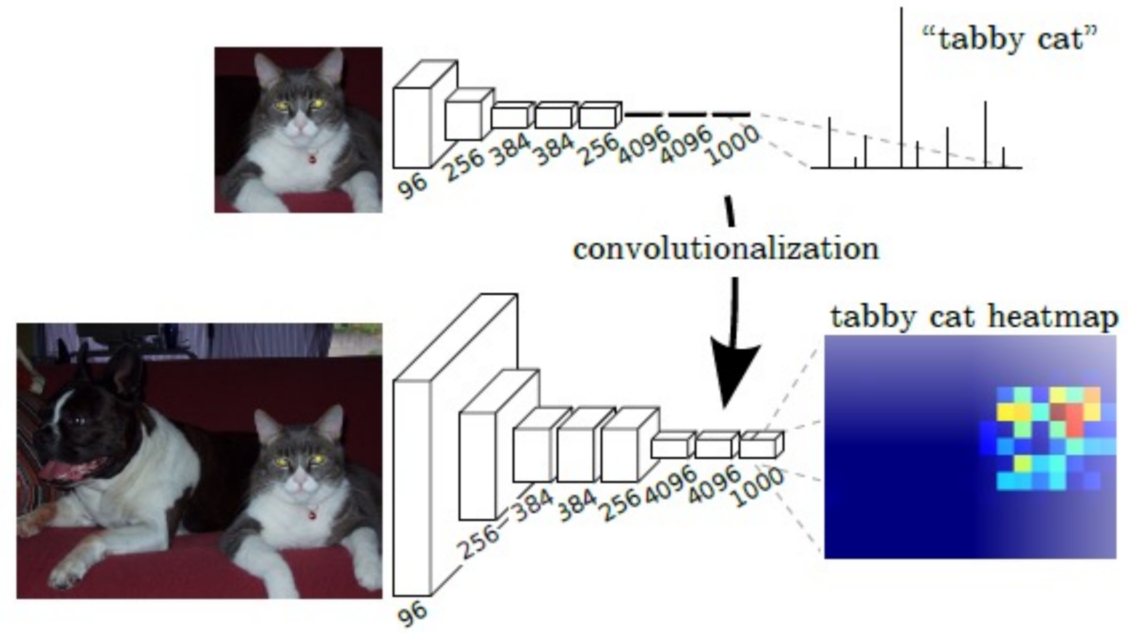
- Label each pixel with a category label
- No difference for instances



Badrinarayanan et al, SegNet: A Deep Convolutional Encoder-Decoder Architecture for image segmentation, 2015

# Approaches: Fully Convolutional Networks

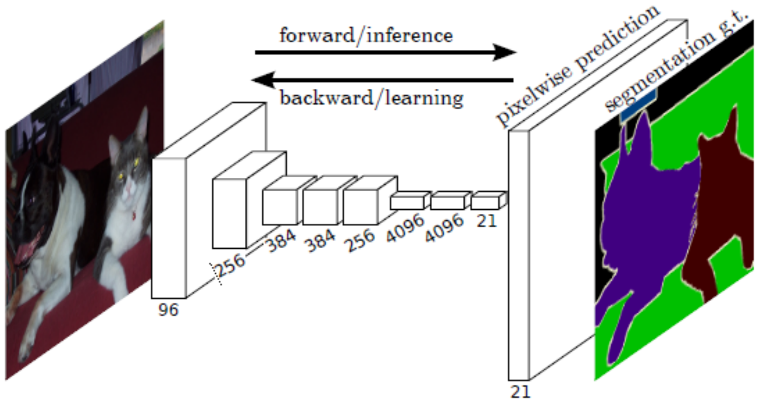
- Rather than convolution layers yielding an image classification
- Convert to the spatial representation of a heat map giving classification probabilities, and upsample
- Maps are highly overlapping: cost amortised




Long et al., Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

# Approaches: Fully Convolutional Networks

- Take classification maps and output a heatmap
- Reconstruct a segmentation map from this downsampled map
- Upsampling back up to image size




- A general approach for dense prediction tasks
- Encoder–decoder architecture

Long et al., Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

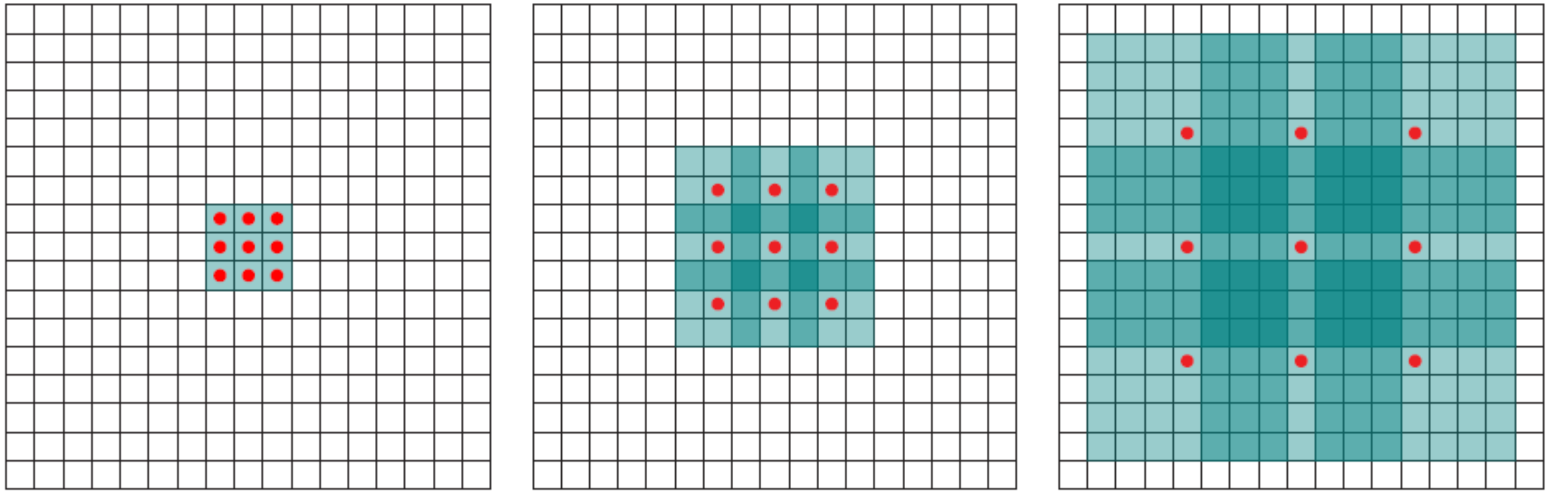
# Dilated Convolution

- We want large spatial fields for classification, detection and segmentation in order to include the spatial context of an object
- For classification, done by layers of convolution and pooling
  - Do not require fine spatial information; a global prediction
- This has been directly applied to segmentation
  - E.g., FCN, DeConvNet, U-Net use this, then up-sample
- Instead, replace pooling by **Dilated Convolution**:
  - No pooling or subsampling
  - Specifically for pixel-based, dense prediction tasks

Yu & Koltun, ICLR 2016

# Dilated Convolution

- Rather than downsample explicitly, space the kernel elements
  - A sample only
- Same number of parameters, larger field of view or spatial context
- For segmentation: dilated convolution rather than convolution and max pooling, then fractionally-strided convolution
  - Same effect, less parameters and memory

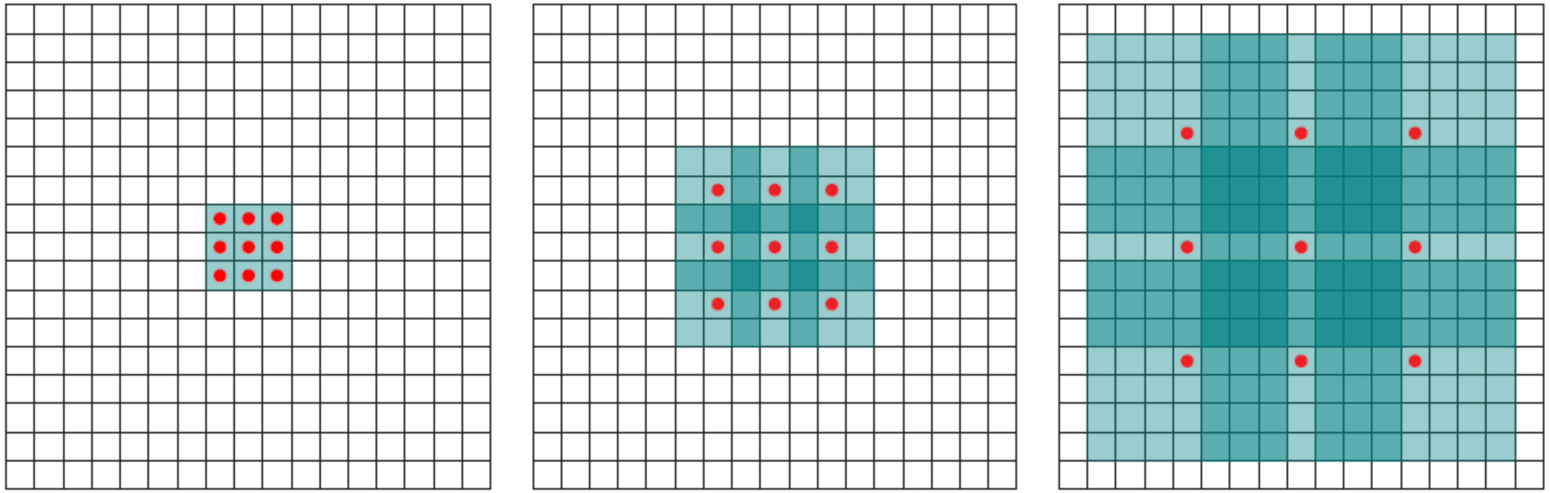


Yu & Koltun, ICLR 2016

# Dilated Convolution

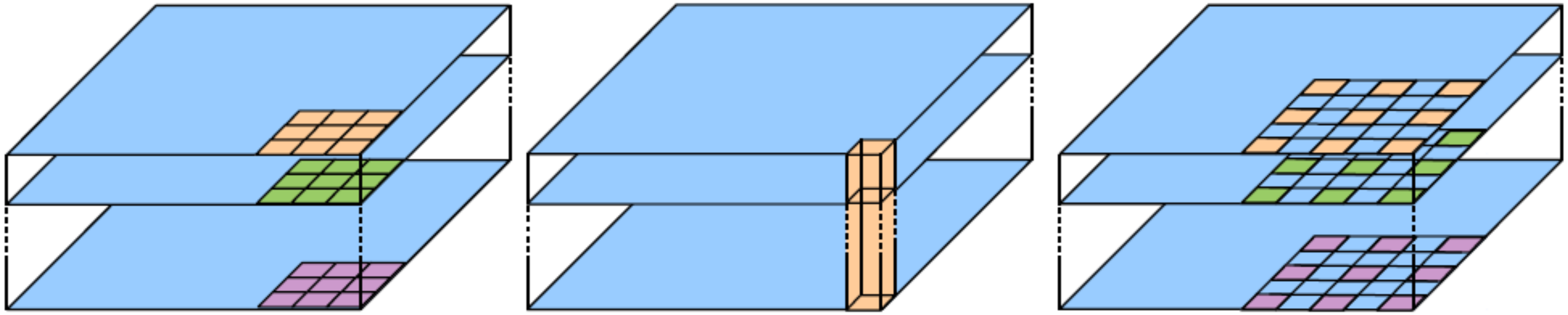
- For location *i* in the output map feature map, *y*
- Convolution filter *w*
- Input feature map *x*
- *r* is the dilation rate
- Standard filter has a dilation rate of 1

$$y[i] = \sum_{k} x[i + r \cdot k] w[k]$$



Yu & Koltun, ICLR 2016

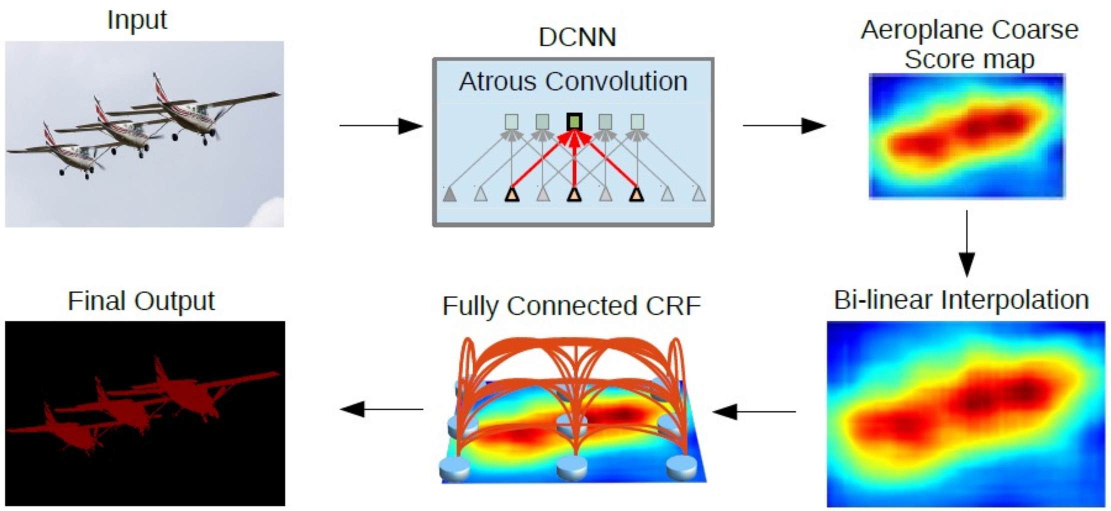
# Dilated Convolution



(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

# Approaches: DeepLab

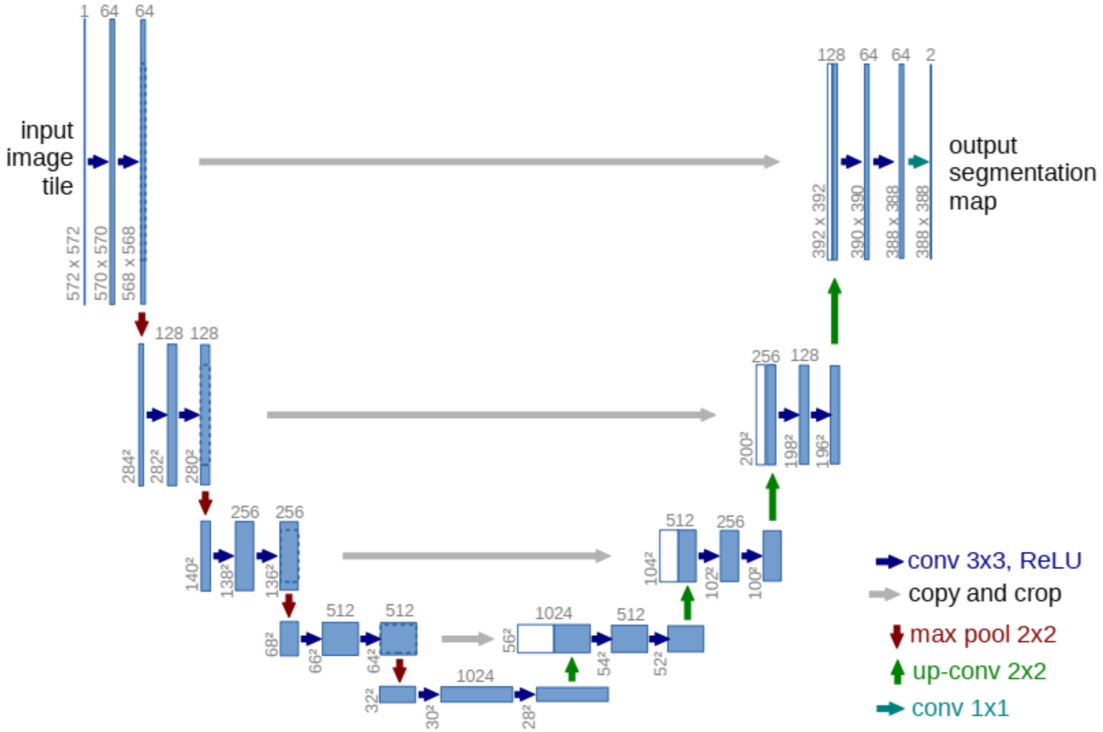
- Dilated convolution = *atrous* convolution
- Atrous spatial pyramid pooling
  - Similar to multi-scale dilated convolution
- Improve localisation of object boundaries
  - Performs smoothing with a Conditional Random Field

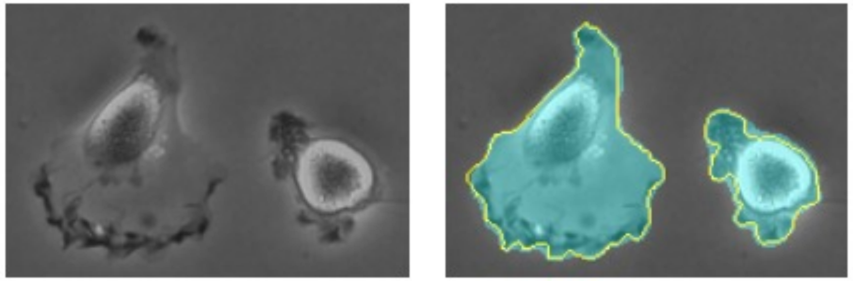



Chen et al., IEEE TPAMI, 2018

# Approaches: U-Net

- U-Net: segmentation model
  - Encoder–decoder
  - Multi-level U-shaped
  - Bottleneck
  - Up-convolutions
  - *Skip connections*



# Image Segmentation: Summary

- Goal: break up the image into semantically-meaningful or perceptually-similar regions

- Classic: histograms, K-means, mean-shift

→

- Deep learning: CNNs, encoder–decoder, atrous convolutions
  - Transformers (e.g., Mask2former, Segment Anything)

# Next Week

- Course review
- Practice exam questions
- Q&A