

# CDS6334 Visual Information Processing

## Lecture 8 Region Segmentation 2610

Faculty of Computing and Informatics



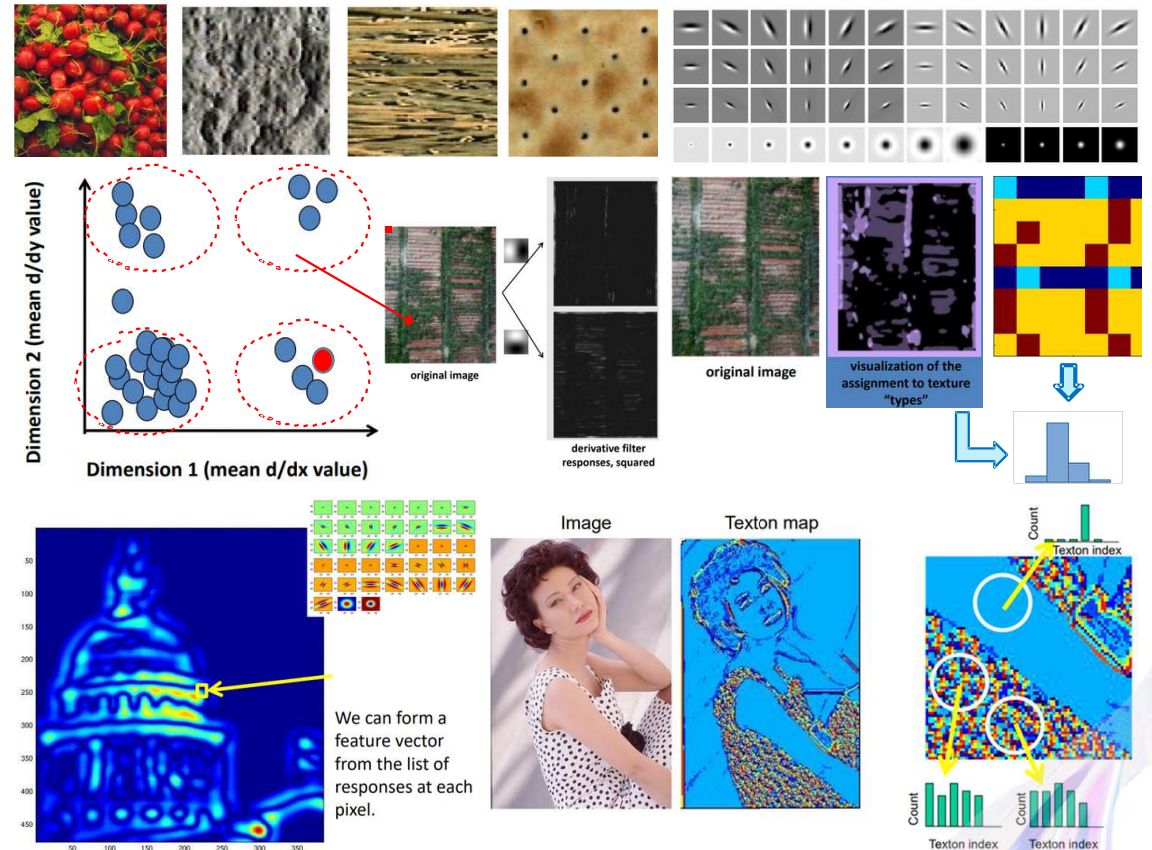
Some graphics and materials are taken from UTexas at Austin CS376 & CS381V notes, Stanford Uni's CS131 notes and Gonzalez & Wood's DIP textbook





## Previously

- Textures
  - Defining textures and representing them
- Texture Representation
  - Simple features (mean, std. dev.) from filters
  - Filter bank – a series of filters
  - Histogram of texture feature occurrences
- Applications
  - Scene classification
  - Texture Matching
  - Image Retrieval
- Recommended Reading
  - [Forsyth & Ponce] Chapter 10 (10.1 in particular)

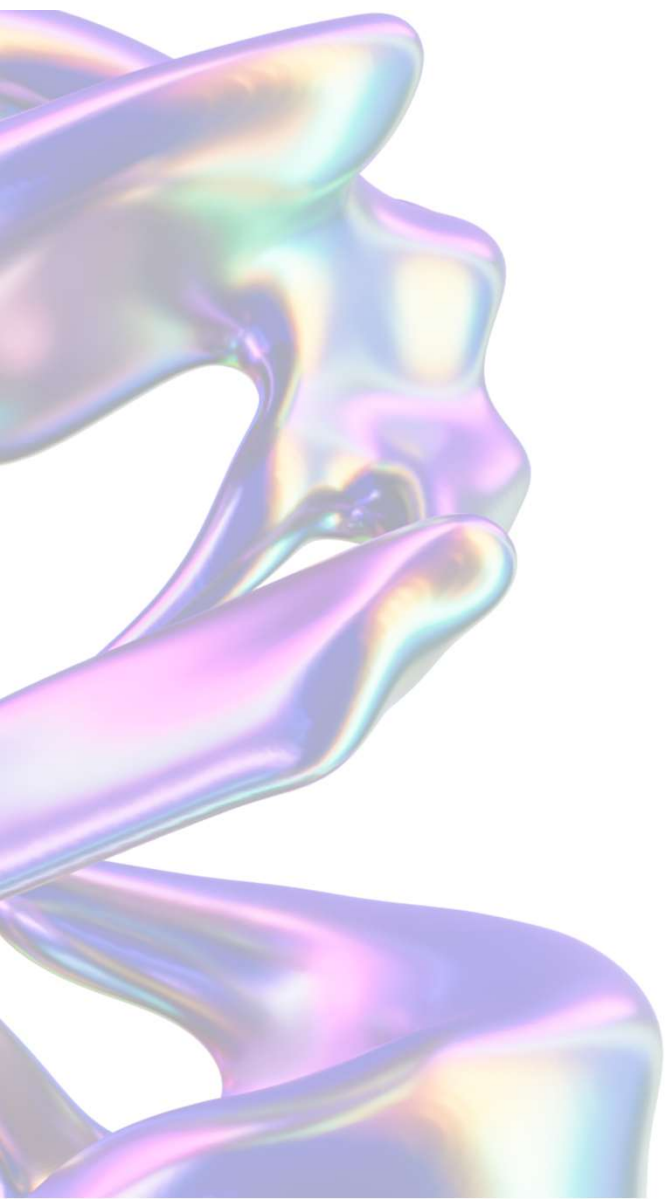




## Lecture Outline

- Why do segmentation?
  - Inspiration from human perception – Gestalt theory
- Segmentation as clustering
  - K-means
- Superpixels – Graph-based algorithms
  - Graph cuts / normalized cuts
  - Felzenswalb's method
- Deep Learning methods
  - FCN Semantic Segmentation
  - Mask R-CNN Instance Segmentation
  - ViT Semantic Segmentation





# Why do segmentation?

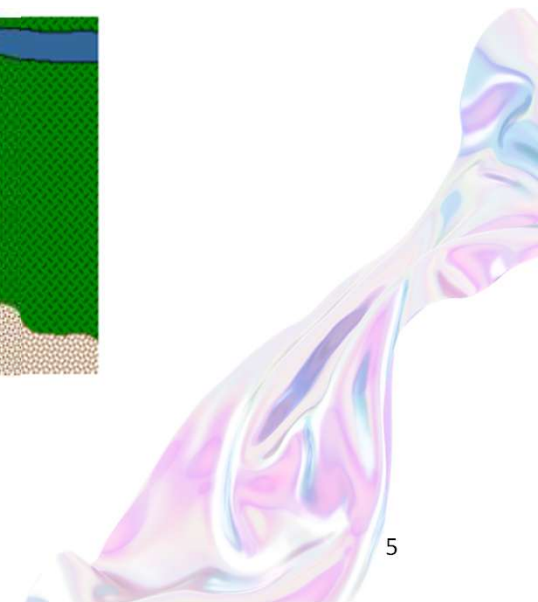
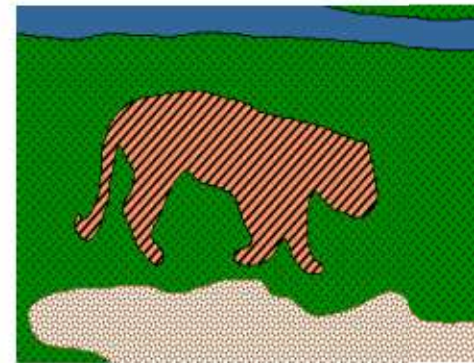


Image credit:  
<https://www.superannotate.com/blog/image-segmentation-for-machine-learning>



## Why do segmentation?

- Aims:
  - Gather or group features that belong to each other
  - Obtain an intermediate representation that compactly describes key image/video parts





# Examples of segmentation tasks



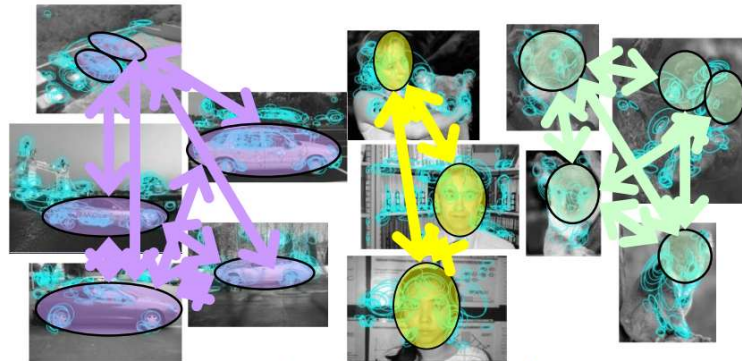
[Figure by J. Shi]

Determine image regions



[[http://poseidon.csd.auth.gr/LAB\\_RESEARCH/Latest/imgs/S\\_peakDepVidIndex\\_img2.jpg](http://poseidon.csd.auth.gr/LAB_RESEARCH/Latest/imgs/S_peakDepVidIndex_img2.jpg)]

Group video frames into shots



[Figure by Grauman & Darrell]

Object level grouping



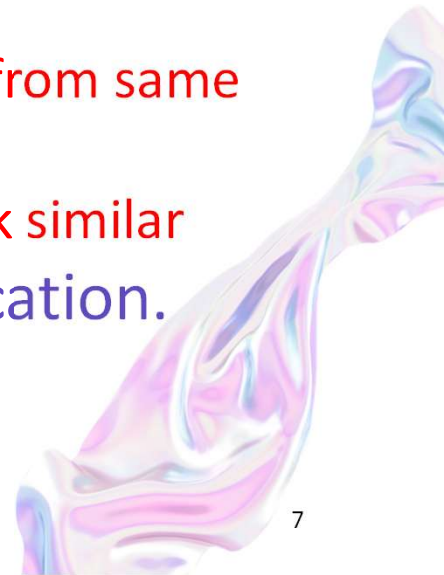
[Figure by Wang & Suter]

Figure-ground



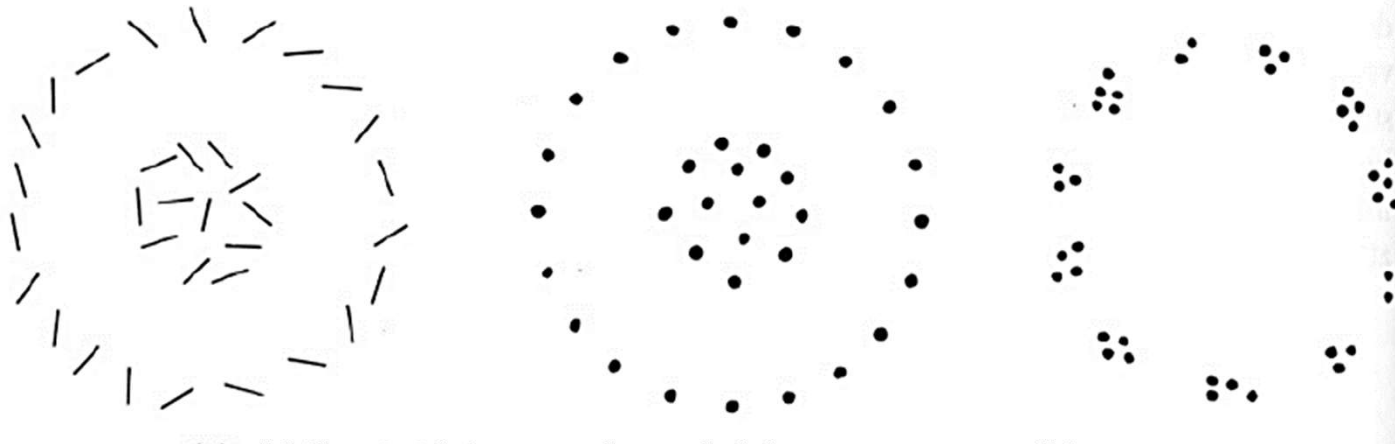
## Segmentation

- Aims:
  - Gather or group features that belong to each other
  - Obtain an intermediate representation that compactly describes key image/video parts
- Top-down vs. bottom-up segmentation
  - Top-down: Pixels belong together because **they are from same object**
  - Bottom up: Pixels belong together because **they look similar**
- **Hard to measure success – depends on application.**

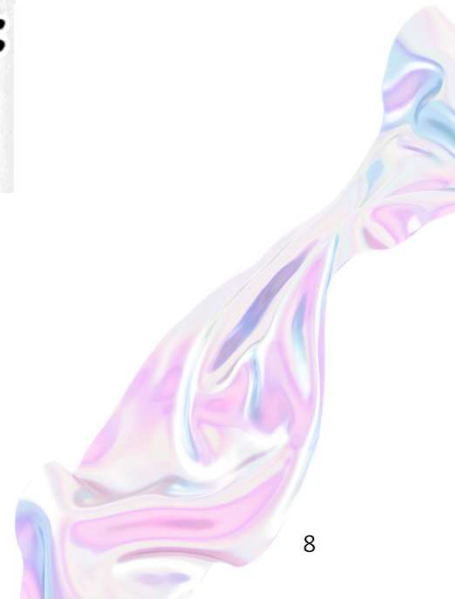




## Grouping what?



What things should be grouped?  
What cues indicate groups?





# Gestalt Theory

- How to group pixels (the smallest element in images) based on these “Laws of **Gestalt Theory**”?



Law of Similarity



Law of Symmetry

**Gestalt:** Theory of the mind and how we acquire and maintain meaningful perceptions



Law of Symmetry



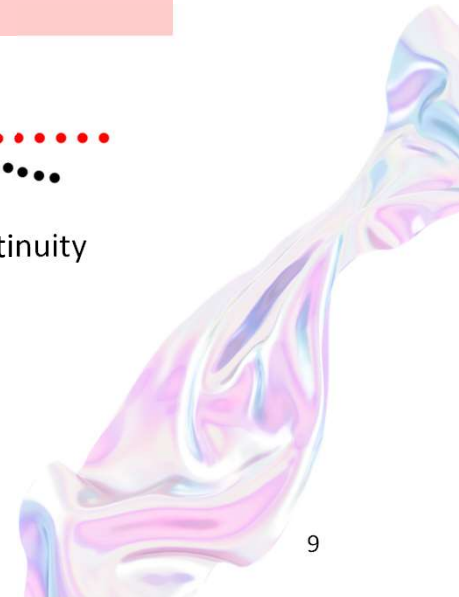
Law of Continuity



Law of Closure



Law of Common Fate





## Gestalt Theory

- Challenge
  - How to best map some of these ideas into algorithms?
    - At pixel level, we need to use some form of similarity between pixels as way to measure





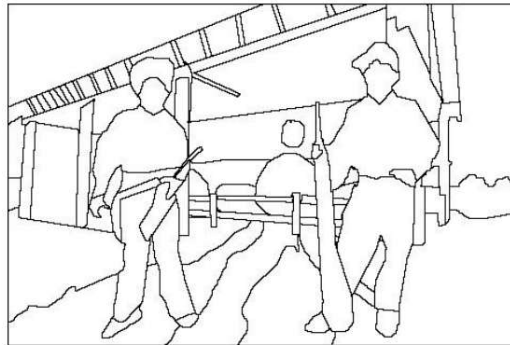
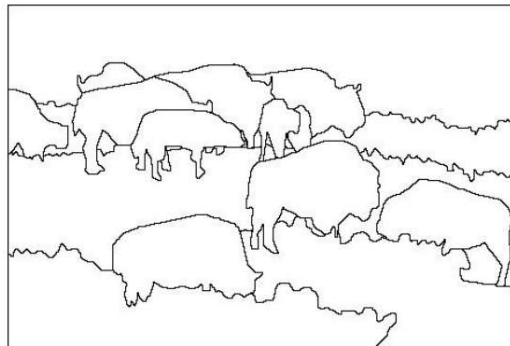
# The goal of segmentation

## 1. Separate image into coherent “objects”

image



human segmentation

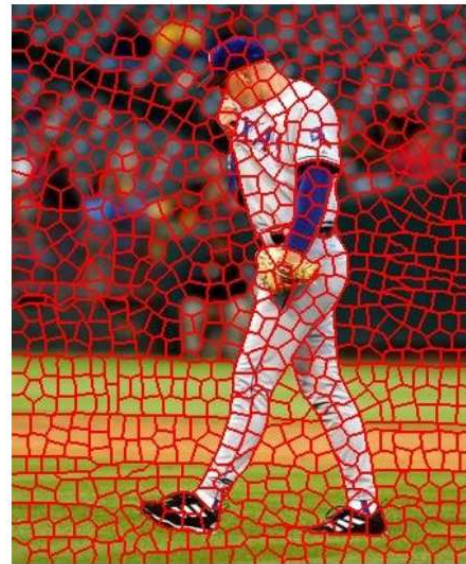
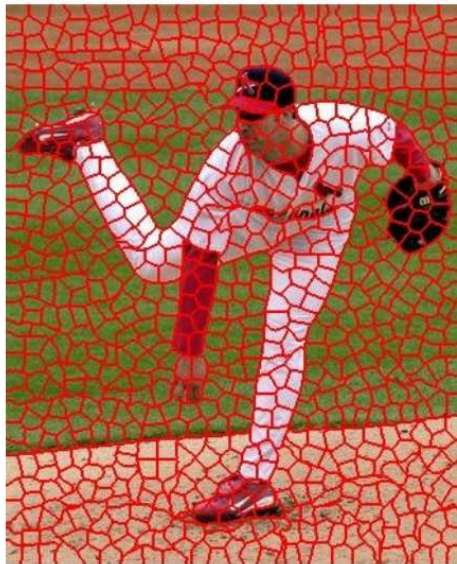




## The goal of segmentation

1. Separate image into coherent “objects”
2. Group together similar-looking pixels for efficiency of further processing

“superpixels”



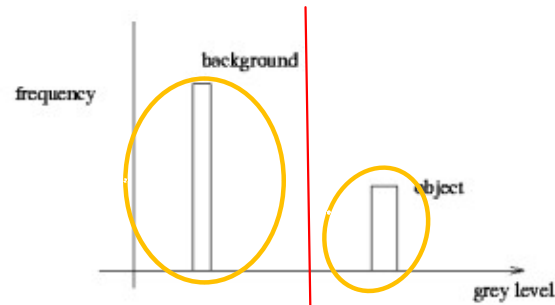
X. Ren and J. Malik. Learning a classification model for segmentation. ICCV 2003.



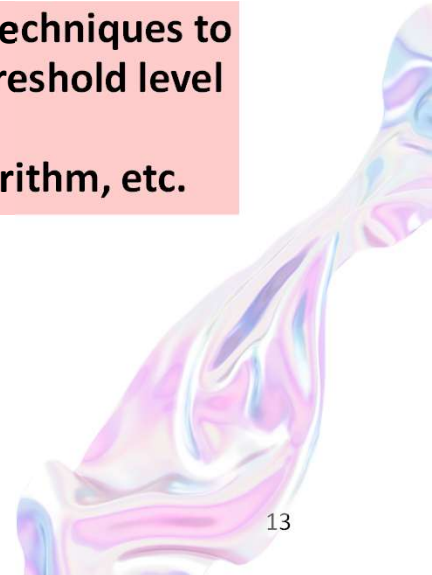
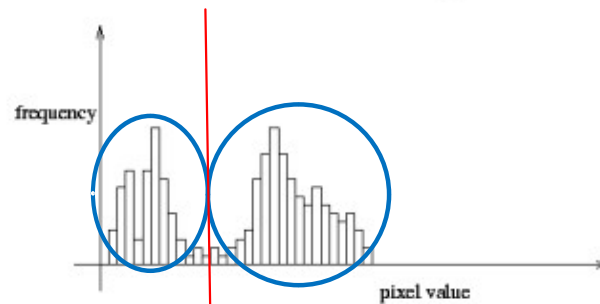
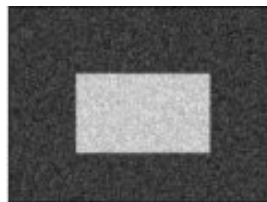


# Thresholding Revisited

- What thresholding can do:
  - Divide these intensities into two significant groups based on where the “mass” of pixels are located

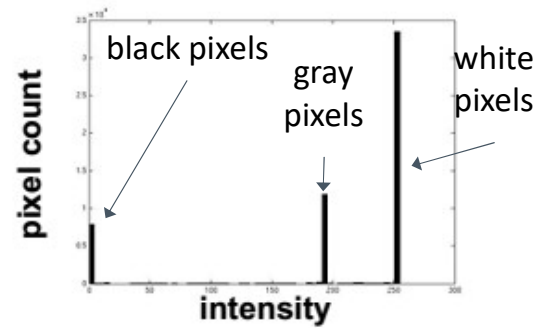
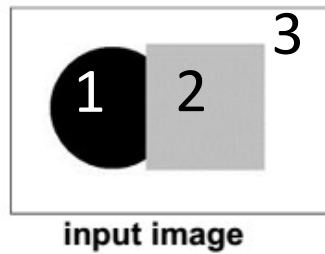


There are techniques to find the threshold level easily:  
Otsu's algorithm, etc.

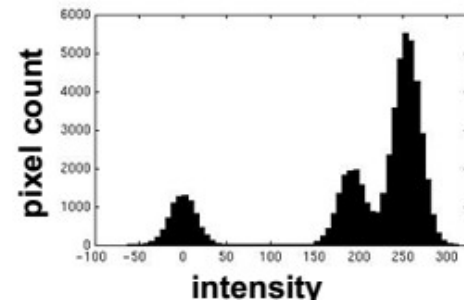
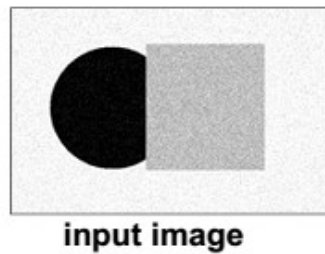




# Motivation of Clustering



**Threshold to 3 regions. Easy.**

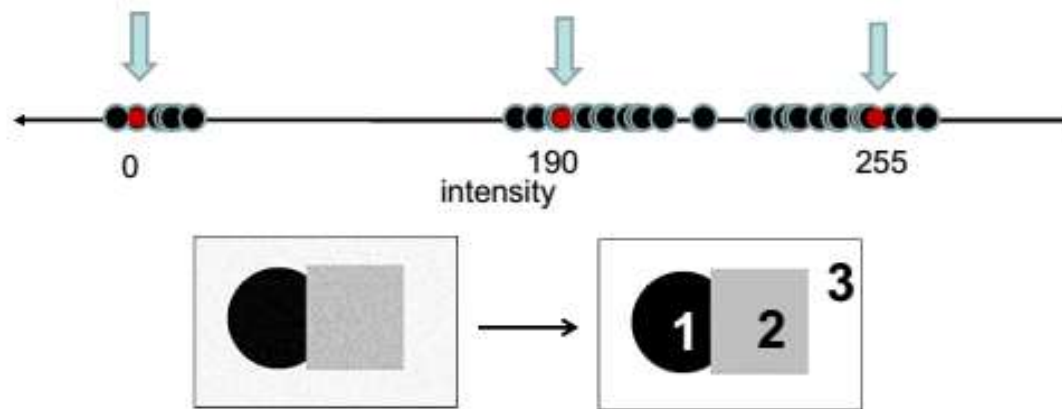


**Clustering can determine the "three" main intensities that define the groups.**





# Clustering Revisited



1. Choose the best cluster centres (as representatives) that minimizes SSD between all points and their nearest centre
1. Label every pixel according to which of these centres it is nearest to (sometimes known as “vector quantization”)

$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$





## Clustering Methods

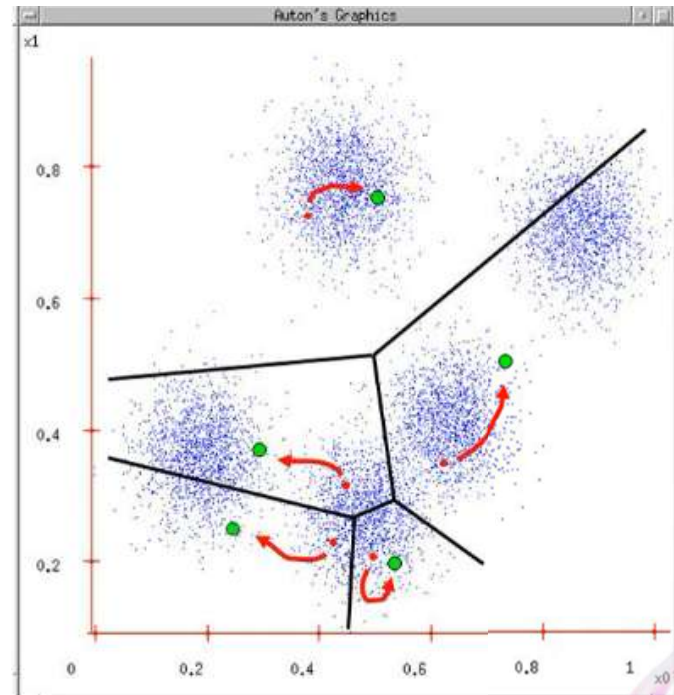
- › K-means clustering (top-down)
- › Mean shift clustering (top-down)
- › Hierarchical agglomerative clustering (bottom-up)





# K-means clustering

1. Determine beforehand how many clusters or value of  $k$
2. Randomly guess cluster centre locations
3. Each data point finds out which centre it is closest to (each centre “owns” a set of points)
4. Each centre finds the centroid of its own group
5. With the new centroid, repeat again the process from (3) until algorithm terminates





# Segmentation as Clustering



## Segmentation as clustering

- Depending on what we choose as the *feature space*, we can group pixels in different ways
- Grouping pixels based on **intensity** similarity

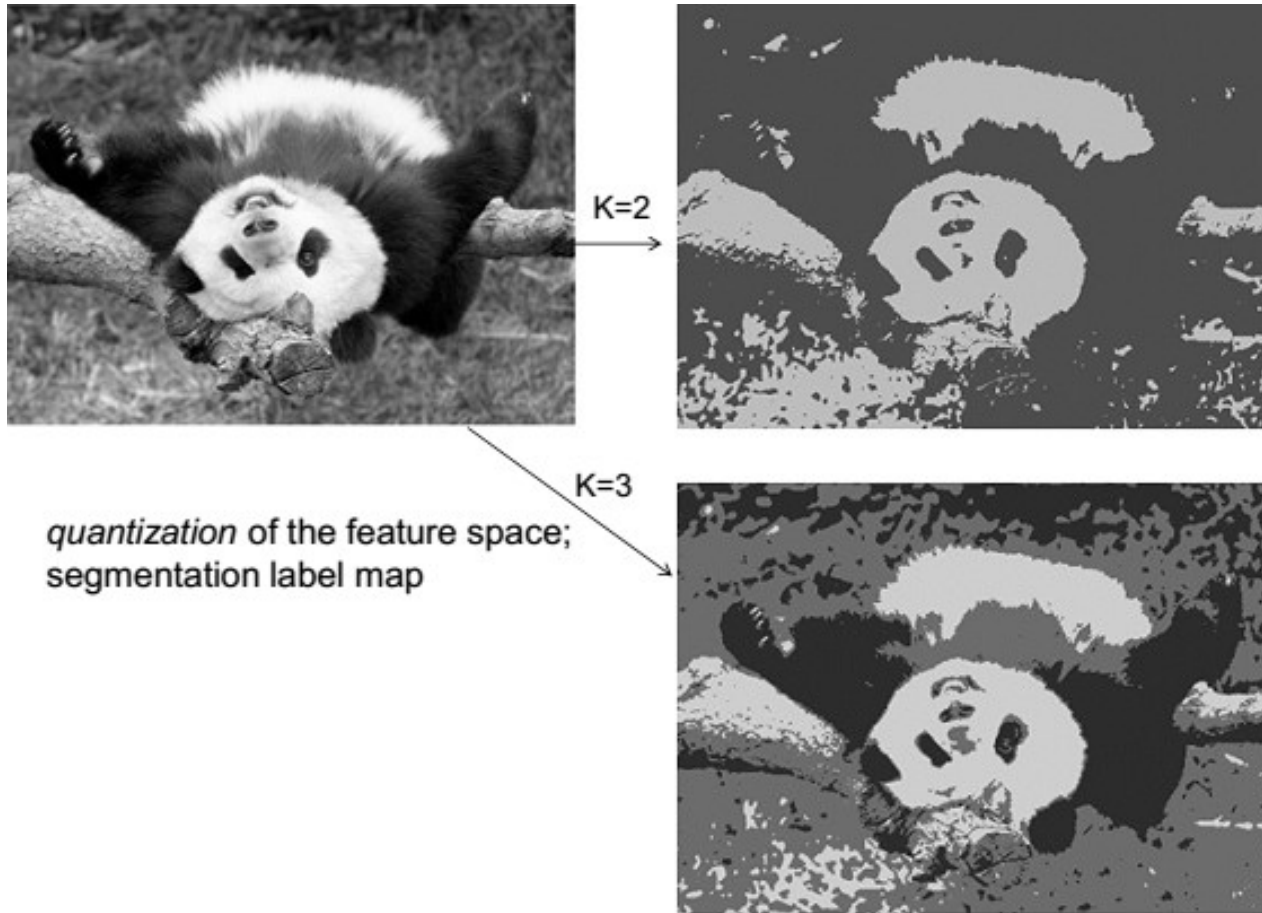


- Feature space: Intensity value (1-D)





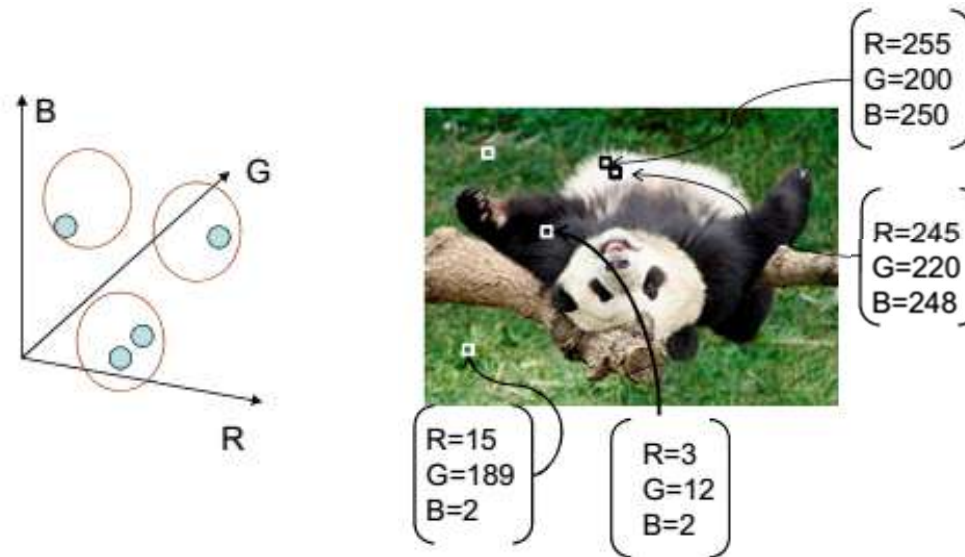
# Segmentation = quantization



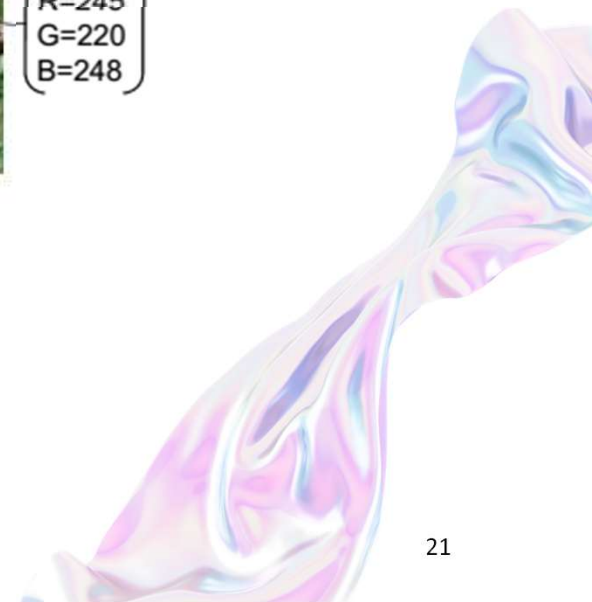


# Segmentation as clustering

- Grouping pixels based on color similarity



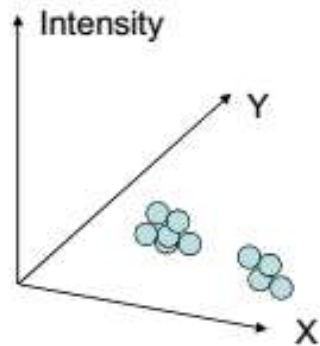
- Feature space: Color value (3-D)





## Segmentation as clustering

- Clusters based on intensity/color similarity are **not spatially coherent**  $\Rightarrow$  No idea where those pixels are...
- Grouping pixels based on **intensity+position** similarity



Both regions are black, but if we also include position  $(x,y)$ , then we could group the two into different segments

Encoding **similarity** & **proximity** (recall Gestalt theory)

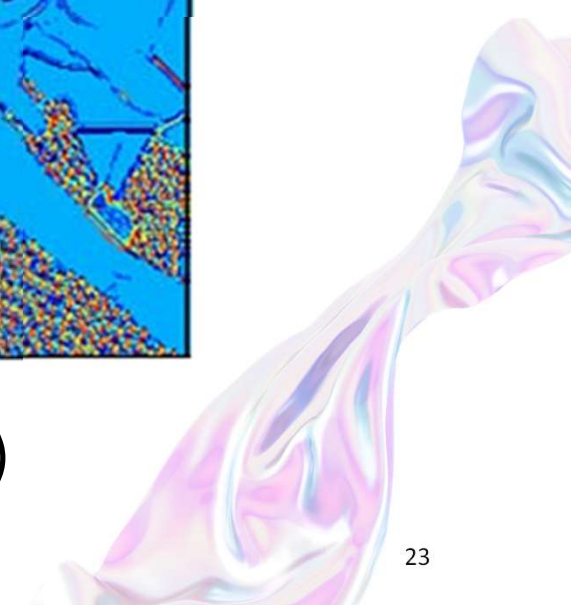


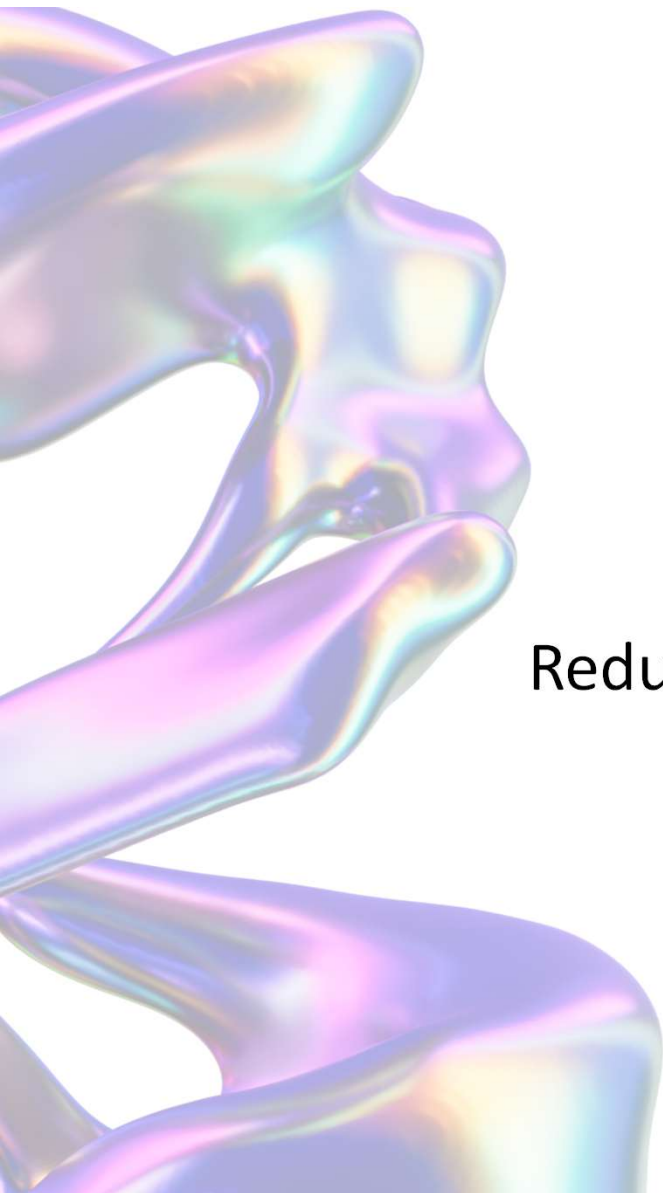
# Segmentation as clustering

- Grouping pixels based on texture similarity



- Feature space: Filter bank responses (e.g. 24-D)





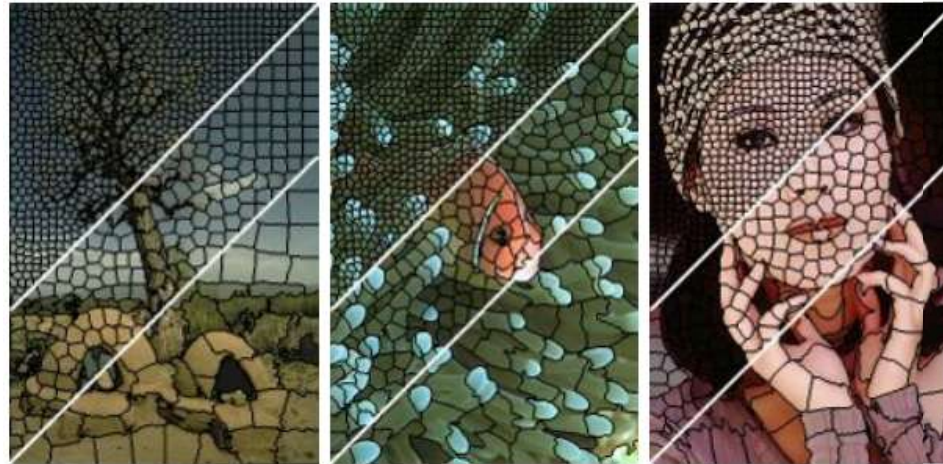
# Superpixels

Reducing spatial information for segmentation



## Superpixels

- **Issue:** We have way too many pixels in a reasonably sized image.
- **Idea:** Merge similar pixels into way less pixels  
⇒ “Superpixels”





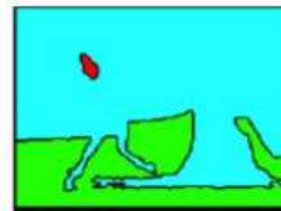
## Superpixels

- **Superpixels** is a result of “oversegmentation”
  - Intentionally segment an image into **many smaller parts**
- Justification
  - The standard “**pixel-grid**” is **not a natural** representation of visual scenes, but an “artifact” of a digital imaging process

Types of segmentations



Oversegmentation

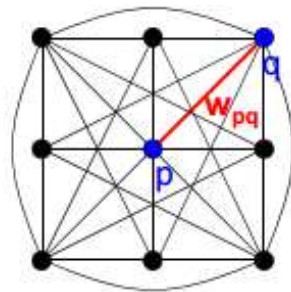


Undersegmentation





## Images as Graphs

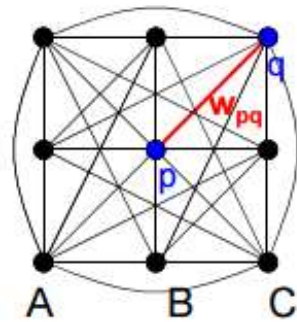


- Fully-connected Graph
  - Node (vertex) for every pixel
  - Link between every pair of pixels,  $p, q$
  - Affinity weight for each edge
    - ⇒ measures **similarity** (inverse to **distances**)

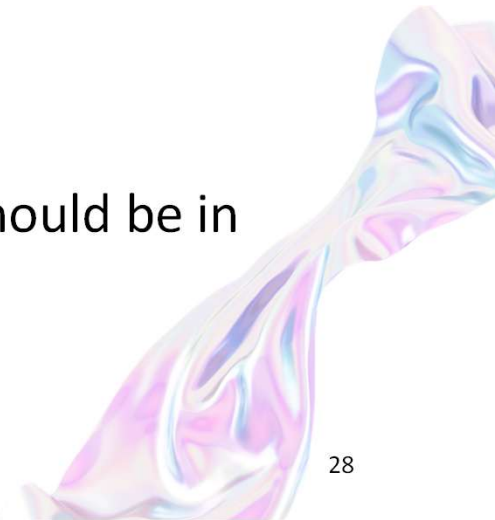




## Graph Cuts



- Break or “cut” graph into segments
  - Delete edges that span across segments
  - Good candidates are edges with low similarities (they should be in different segments)
  - Eventually, groups of edges left represent the segments



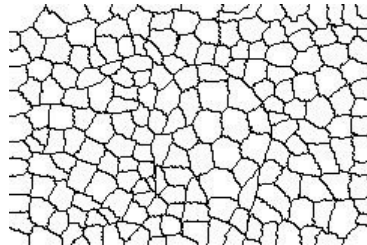


## Superpixels from Normalized Cuts

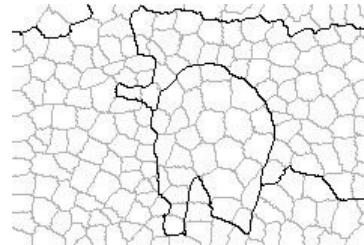
- **Normalized Cuts** – classical region segmentation method developed at UC Berkeley
  - Uses graph-based representation of pixels
  - Spectral clustering used to **exploit pairwise brightness, color and texture affinities** between pixels
  - **Graph is “cut” into segments** by normalized cuts



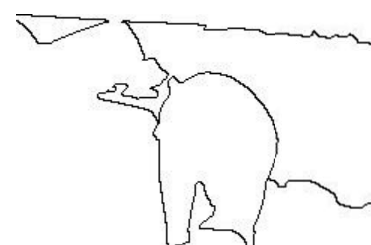
Original image



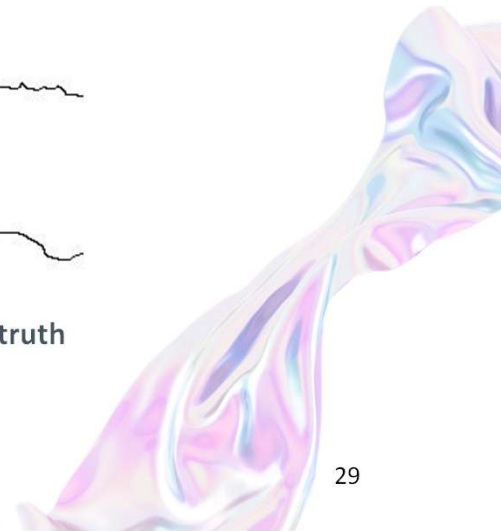
Superpixel segmentation



Region segmentation by  
normalized cut

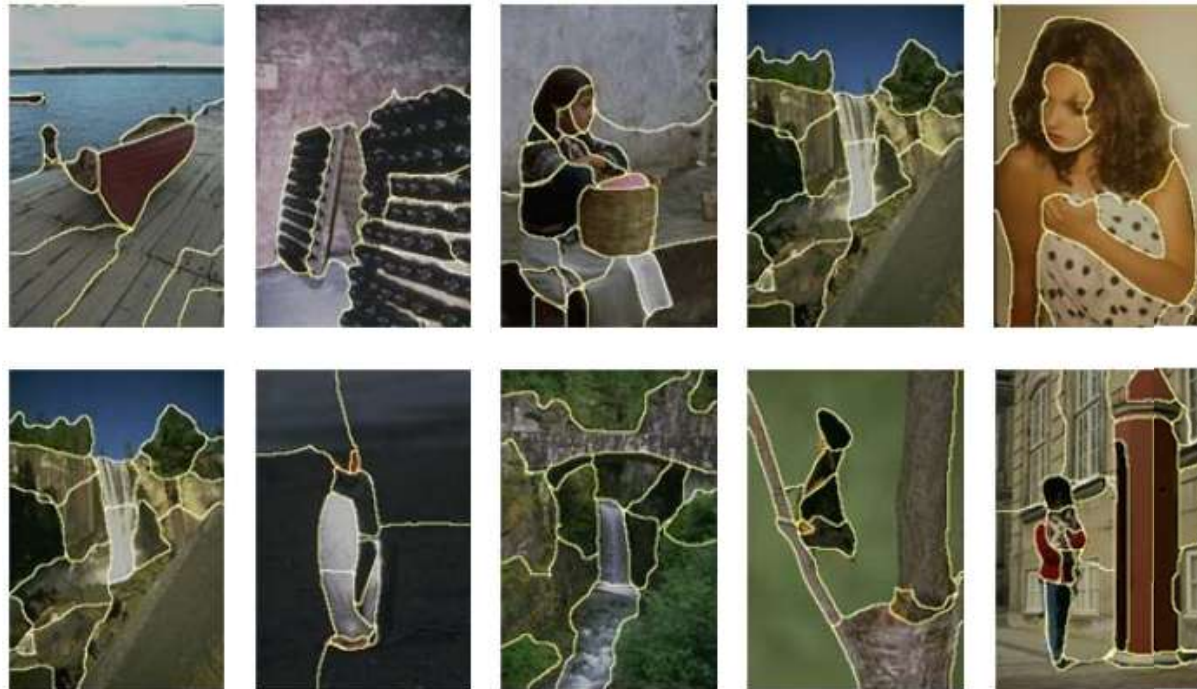


Human ground-truth





## Results from Berkeley Segmentation Engine

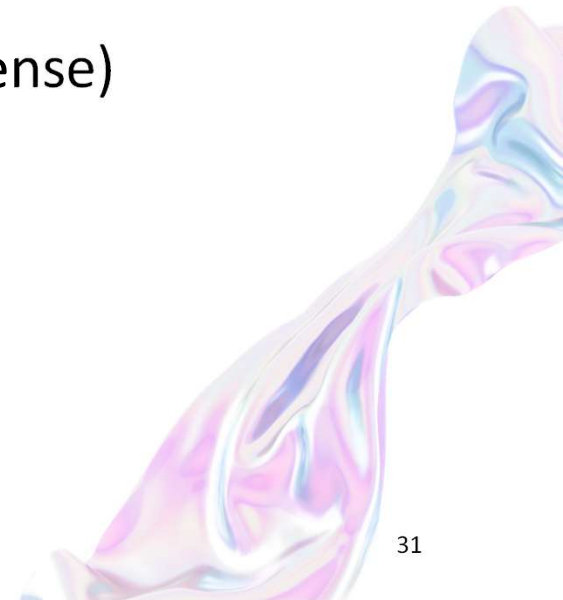


- <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>



## Graph Cuts

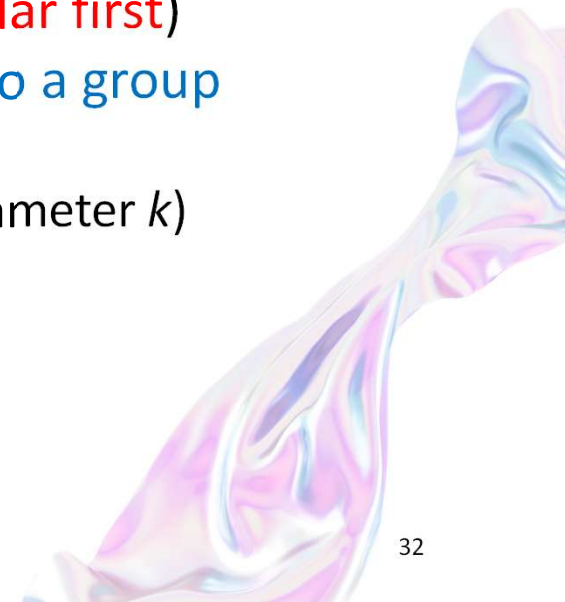
- **Pros**
  - Generic framework, **flexible** to choice of function that computes affinities
  - **Does not require model** of data distribution
- **Cons**
  - **Time complexity** can be high (if graphs are dense)
  - Some applications prefer balanced segments





## Felzenswalb's Segmentation Method

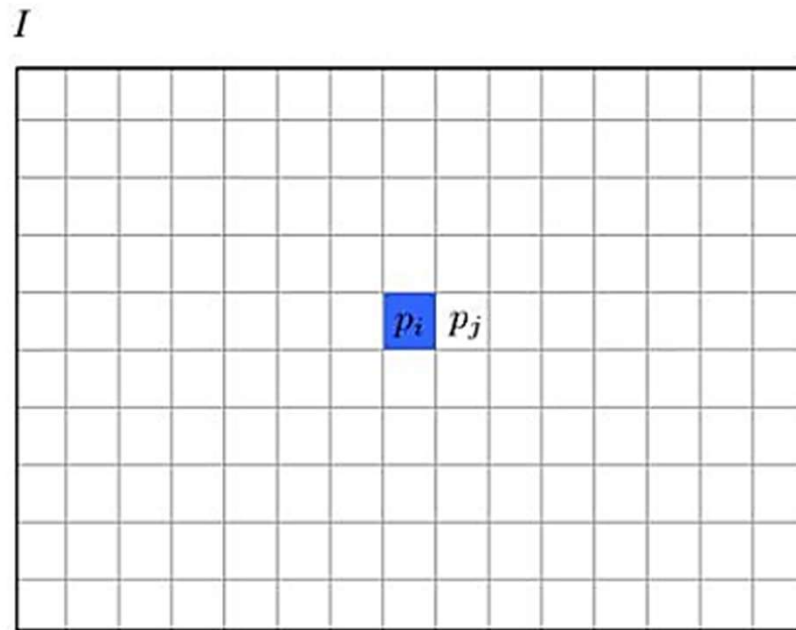
- Also called Felzenswalb-Hutterlocher Method
- Efficient method to **capture perceptually important groupings**
- Steps:
  - Calculate “weights” of all pixels with their respective neighbors
  - Sort the weights by lowest till highest (**most similar first**)
  - From lowest weight, **merge pixels / segments into a group**
    - **if weight < threshold**  
(determined by minimum internal difference and parameter  $k$ )
  - **Repeat** until end of list of weights





# Felzenswalb's Segmentation Method

- Take an image



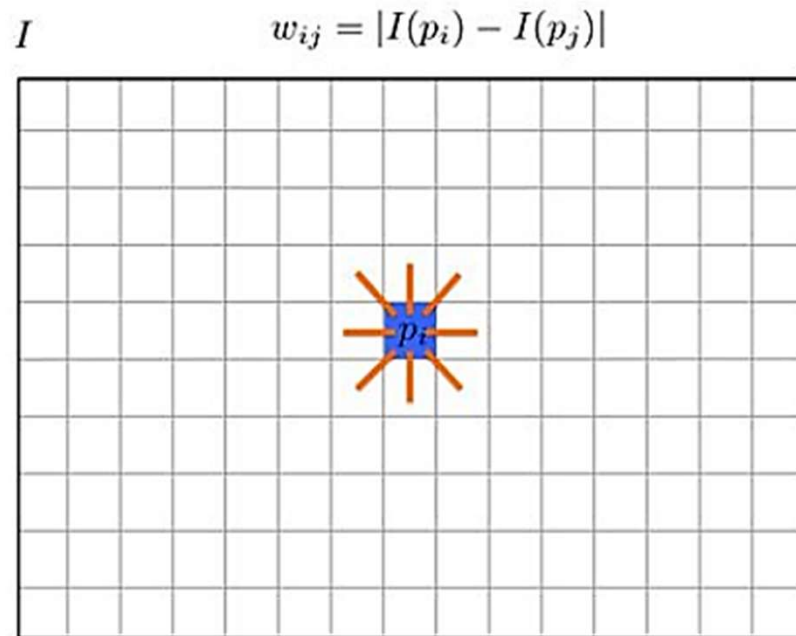
Credits to Sanja Fidler





# Felzenswalb's Segmentation Method

- Compute weights between a pixel and all 8 of its neighbours



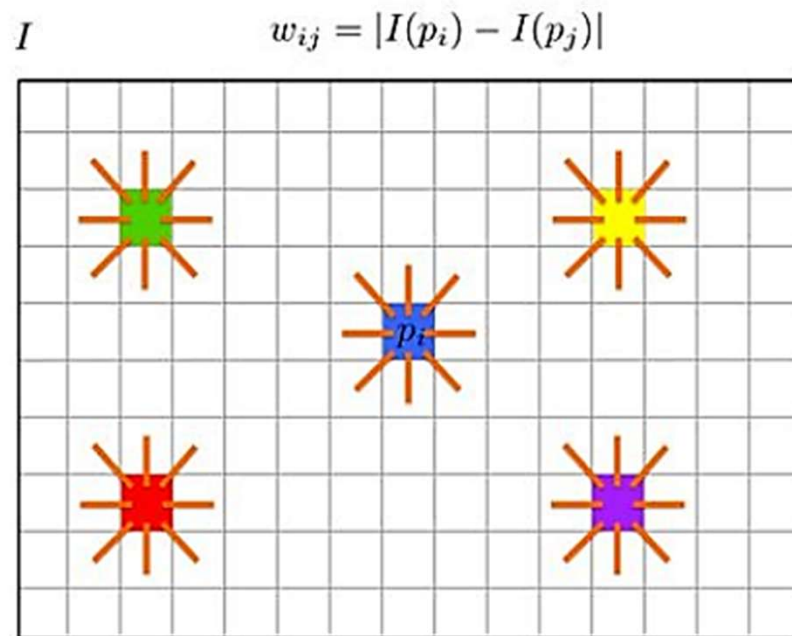
Credits to Sanja Fidler





# Felzenswalb's Segmentation Method

- And do that for **all pixels** in the image



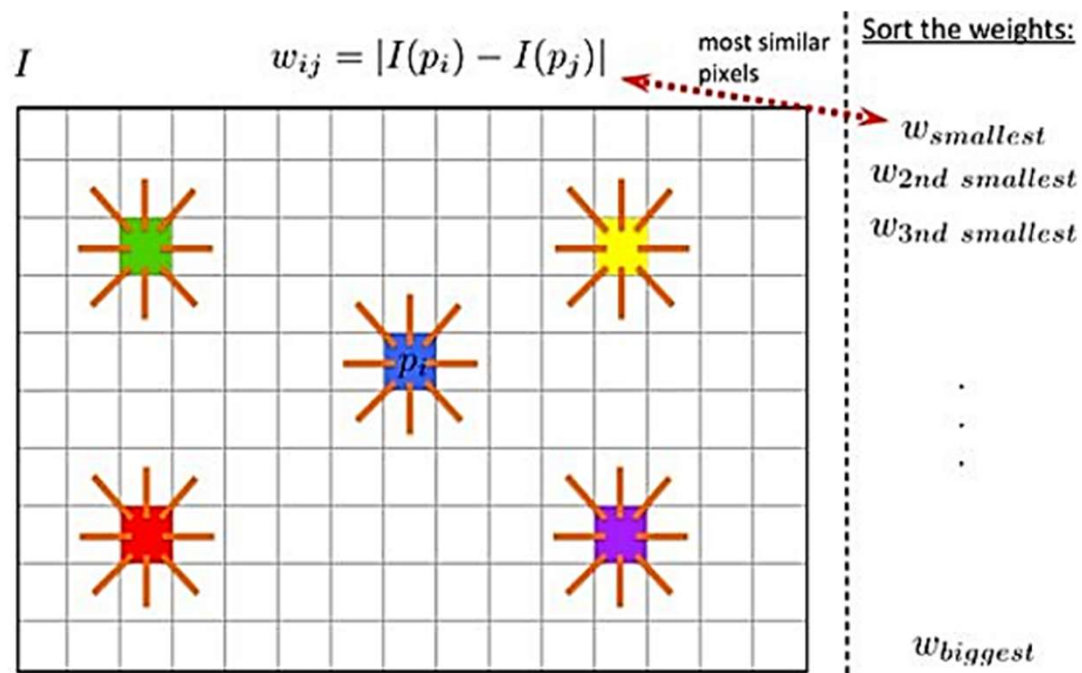
Credits to Sanja Fidler





# Felzenswalb's Segmentation Method

- Now sort all weights in the image by non-decreasing values

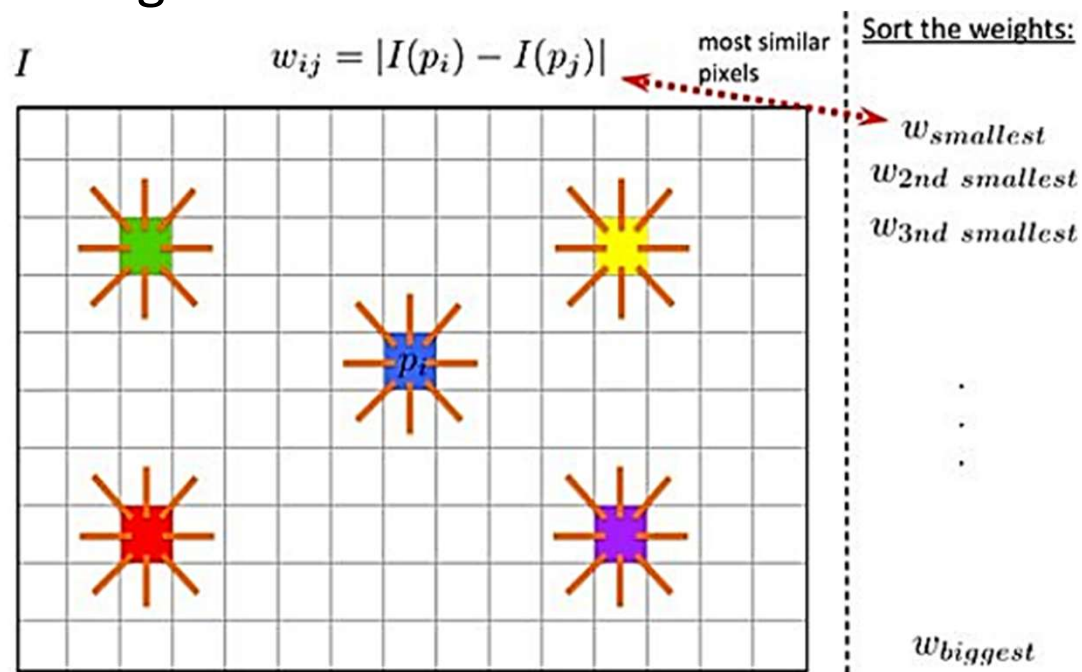


Credits to Sanja Fidler

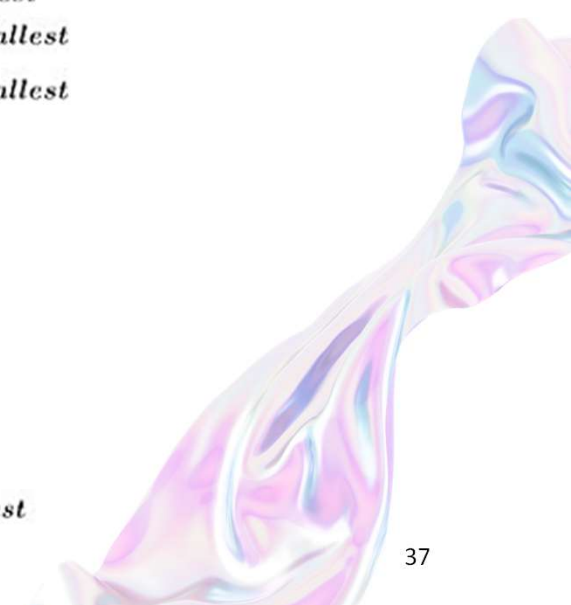


# Felzenswalb's Segmentation Method

- Pick the top-most weight. If **weight < threshold** for both segments, **merge** the pixels . Then, **update threshold** for new segment based on size of segment



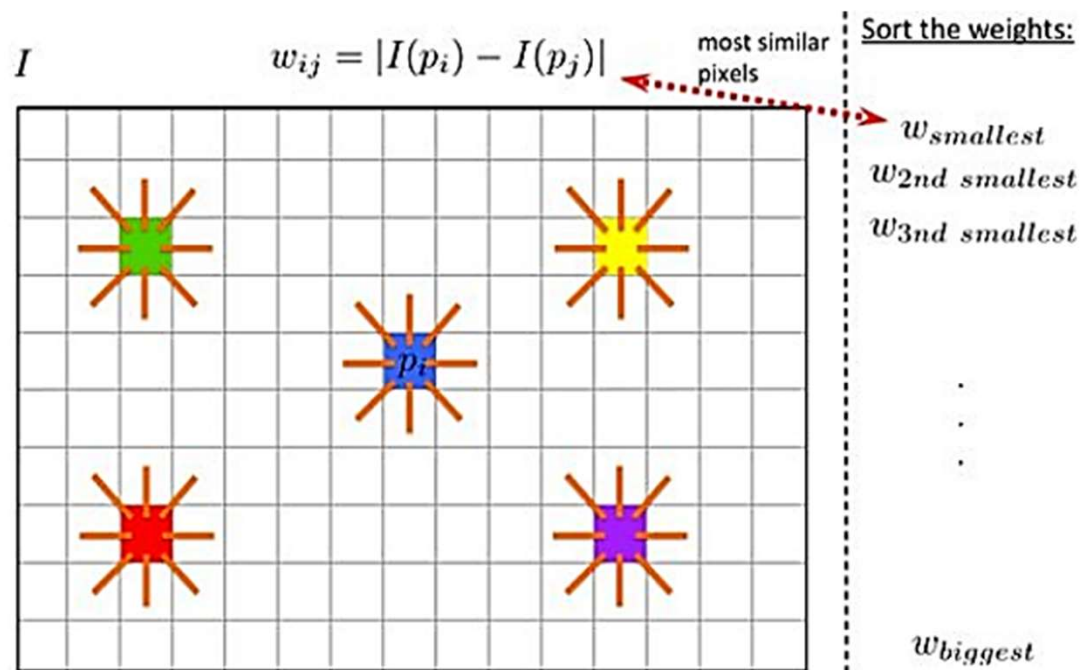
Credits to Sanja Fidler





# Felzenswalb's Segmentation Method

- Repeat until the end of the list

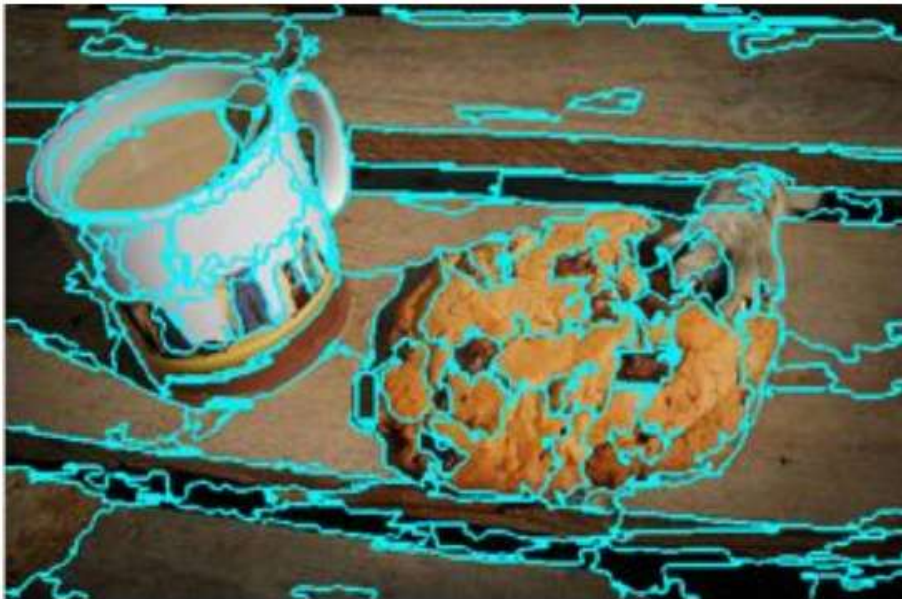


Credits to Sanja Fidler



## Felzenswalb's Segmentation Method

- **Result:** Algorithm runs **real-time**.



Credits to Sanja Fidler





## SLIC Superpixels

- Simple Linear Iterative Clustering Algorithm for Superpixel generation
- Another algorithm that can generate compact, **nearly uniform superpixels**
- Low computational cost
- Only a **single parameter** required: Number of superpixels.





## Why are superpixels useful?

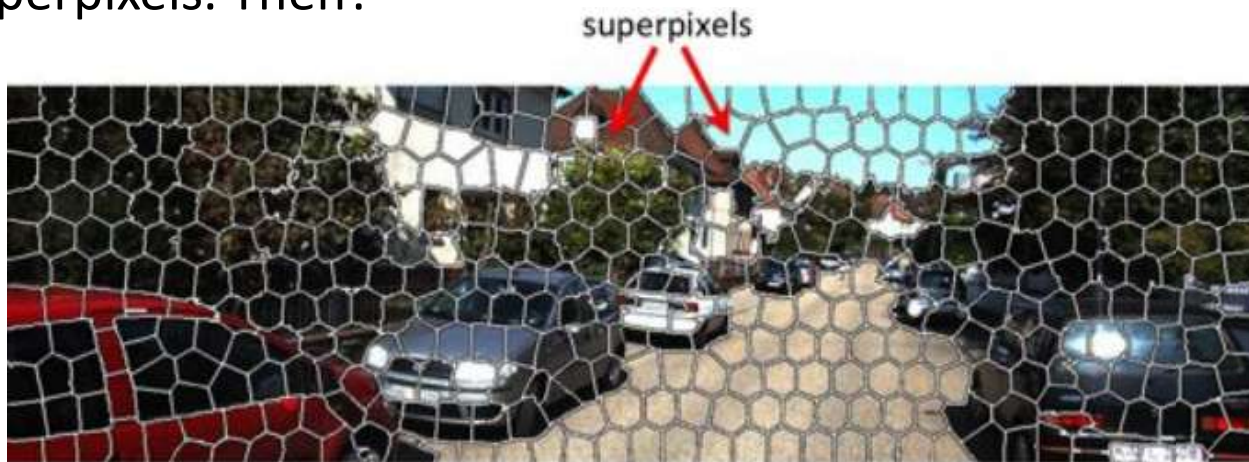
- **Application Example:** How can we find all road pixels in this image?





## Why are superpixels useful?

- **Compute superpixels:** A 4-million pixel image converted to only 500 superpixels. Then?

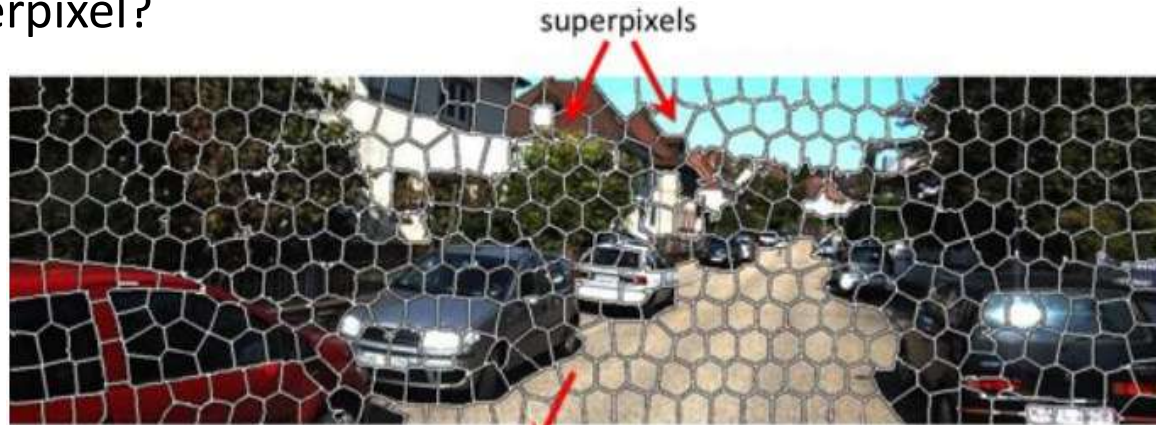


- **Possible idea:** Compute features on each superpixel and train a classifier for road/non-road



## Why are superpixels useful?

- If we use a set of features, different superpixels have different number of pixels. How to get a feature vector that has same dimension for each superpixel?



$$\mathbf{f} = [\text{color}, \text{gradients}, x, y, \dots]^T$$

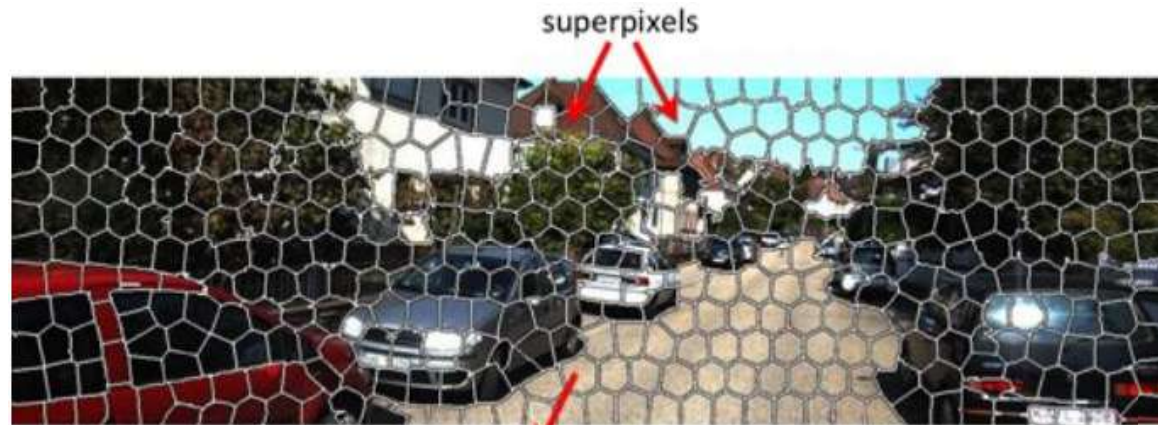
Can be: 1) average across pixels in superpixel, 2) a histogram, 3) a histogram of visual words





## Why are superpixels useful?

- Typically, the more dimensions we use, the better the ability to distinguish between road and non-road.



$$\mathbf{f} = [\text{color, gradients, } x, y, \dots]^T$$

Can be: 1) average across pixels in superpixel, 2) a histogram, 3) a histogram of visual words

- Next, we need to classify them (to be covered later)

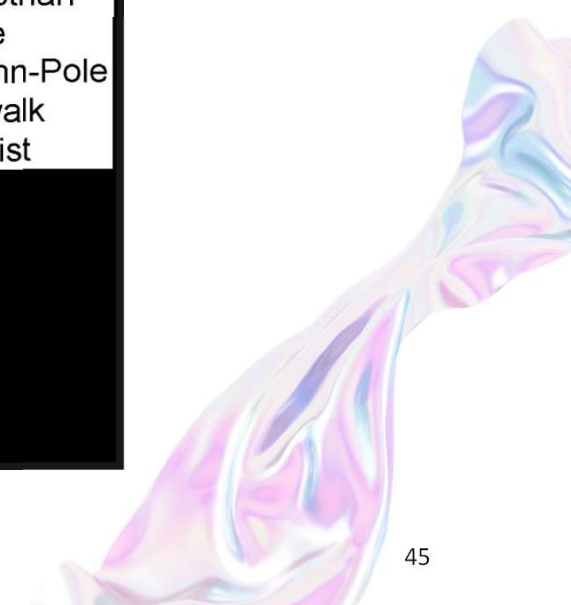


# Scene Segmentation from Dashcams



**SuperParsing: Scalable Nonparametric Image Parsing with Superpixels**

[https://www.youtube.com/watch?v=UZ\\_NTYCFIHk](https://www.youtube.com/watch?v=UZ_NTYCFIHk)



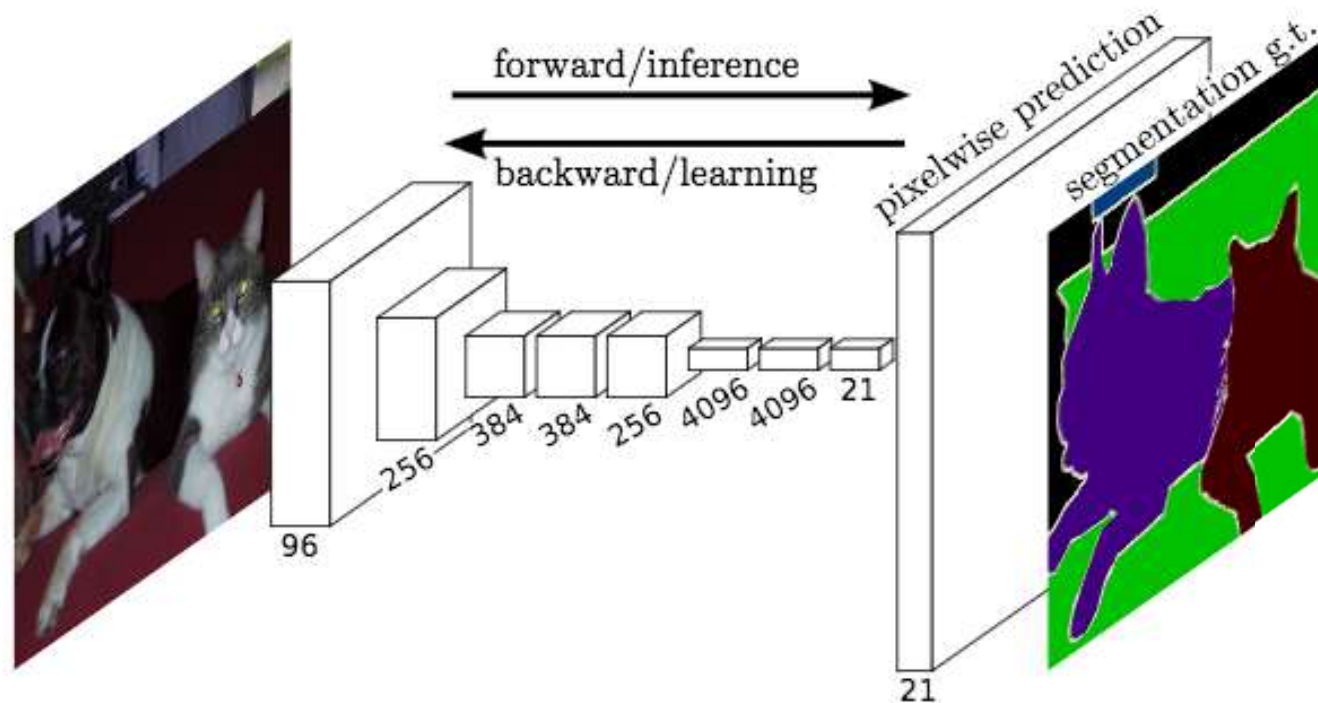


# Deep Learning Methods

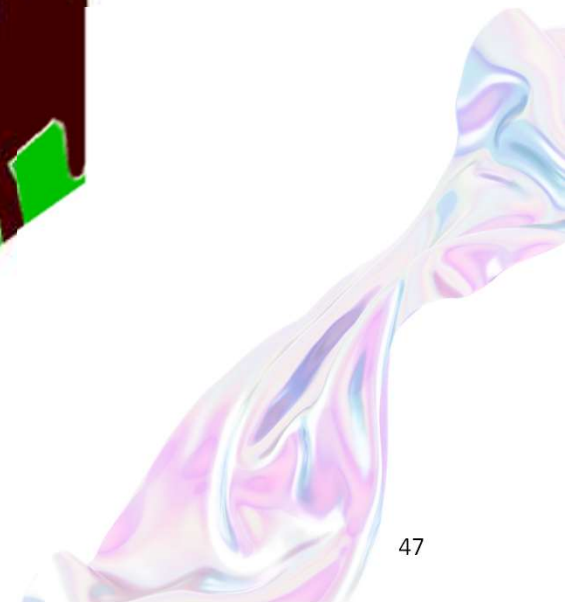
End-to-end models for semantic segmentation



# Fully Convolutional Network (FCN)

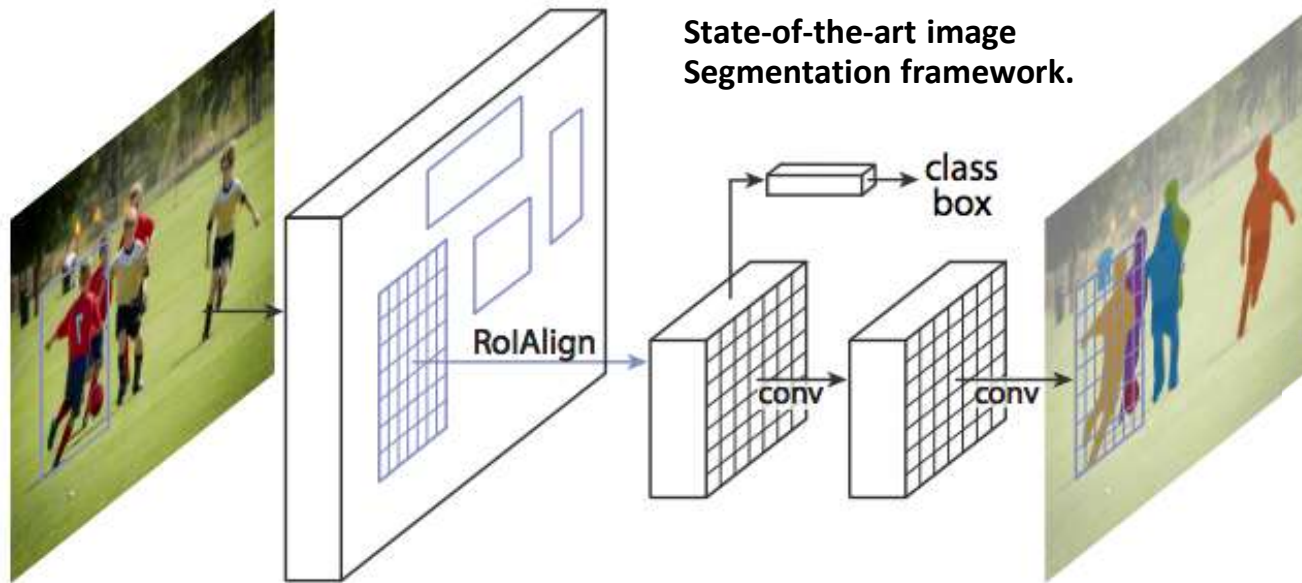


Long, Jonathan, Evan Shelhamer, and Trevor Darrell.  
"Fully convolutional networks for semantic segmentation." *CVPR 2015*.





## Mask Region-based CNN (R-CNN)

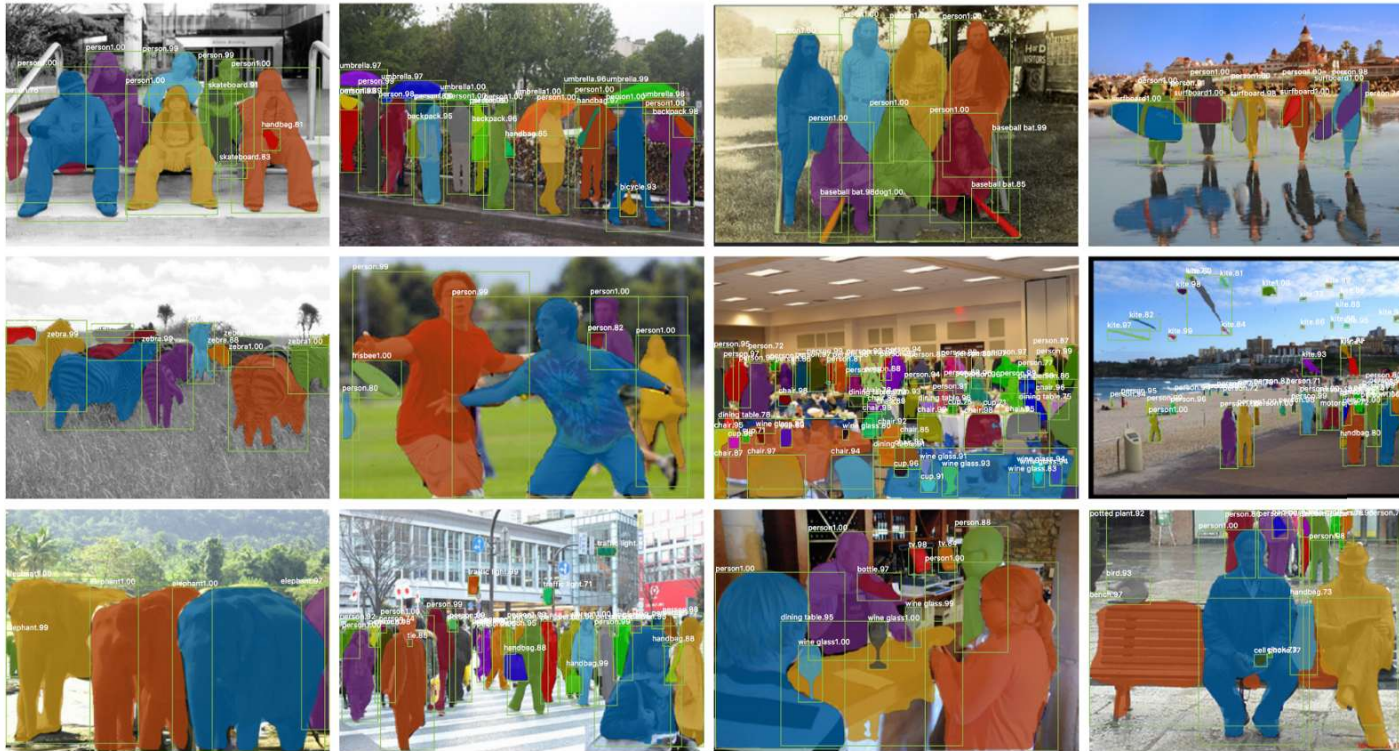


**Mask R-CNN architecture.** The first layer is a RPN extracting the RoI. The second layer processes the RoI to generate feature maps. They are directly used to compute the bounding box coordinates and the predicted class.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell.  
"Fully convolutional networks for semantic segmentation." *In ICCV 2017.*



# Mask Region-based CNN (R-CNN)

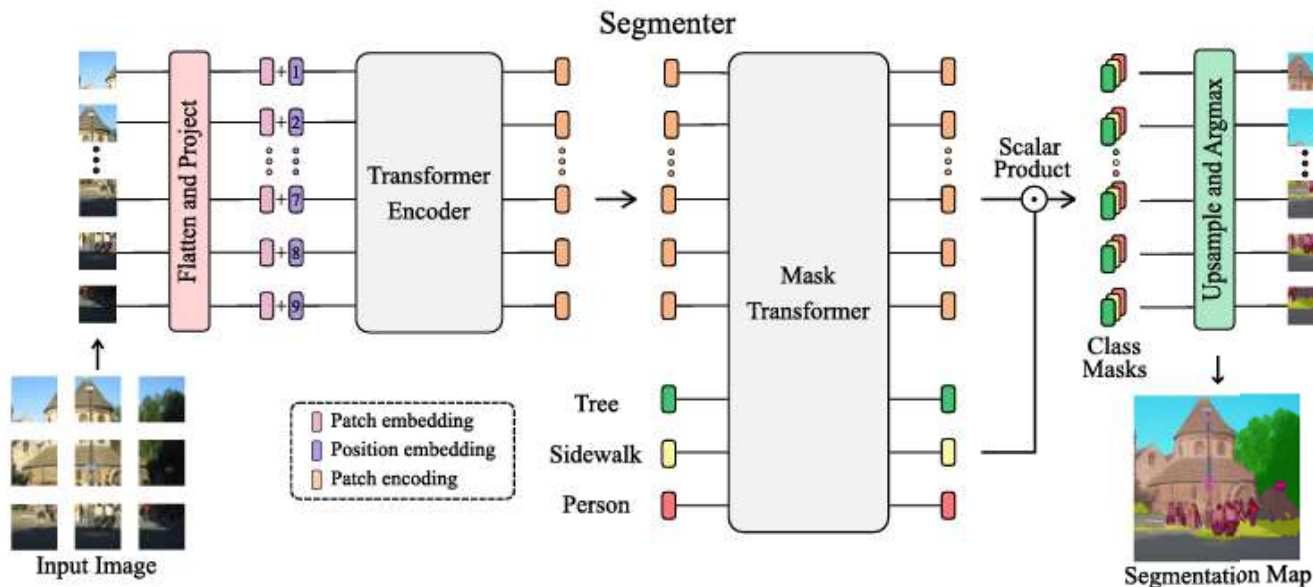


Long, Jonathan, Evan Shelhamer, and Trevor Darrell.  
"Fully convolutional networks for semantic segmentation." *In ICCV 2017.*





# Vision Transformer (ViT) - Segmenter

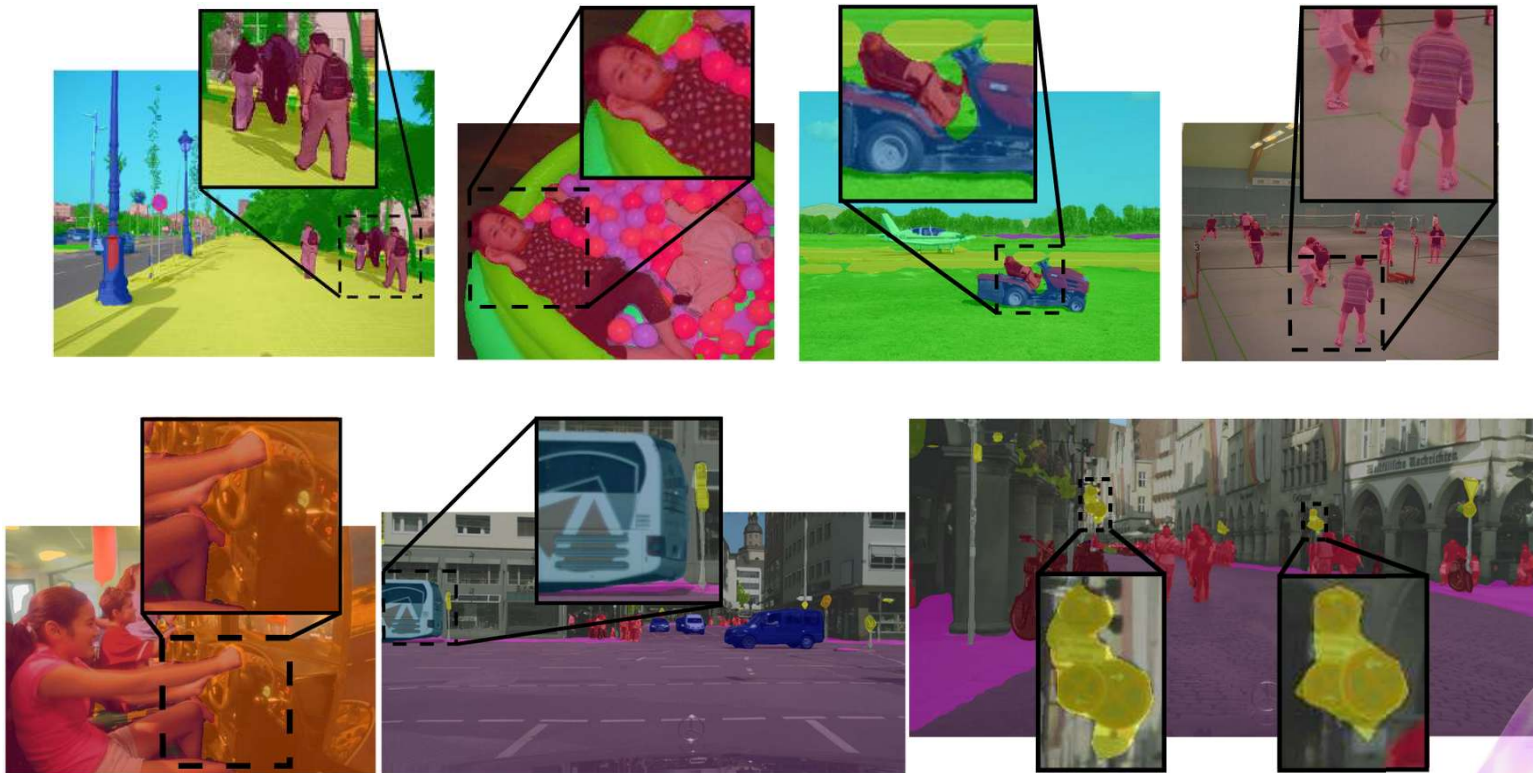


**Segmenter architecture.** The image is first separated into patches and projected to a sequence of embeddings and then encoded with a transformer. A mask transformer then takes as input the output of the encoder and class embeddings to predict segmentation masks.

Strudel, Robin, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid.  
"Segementer: Transformer for semantic segmentation." *In ICCV 2021.*



# Vision Transformer (ViT) - Segmenter



Strudel, Robin, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid.  
"Segmenter: Transformer for semantic segmentation." *In ICCV 2021.*





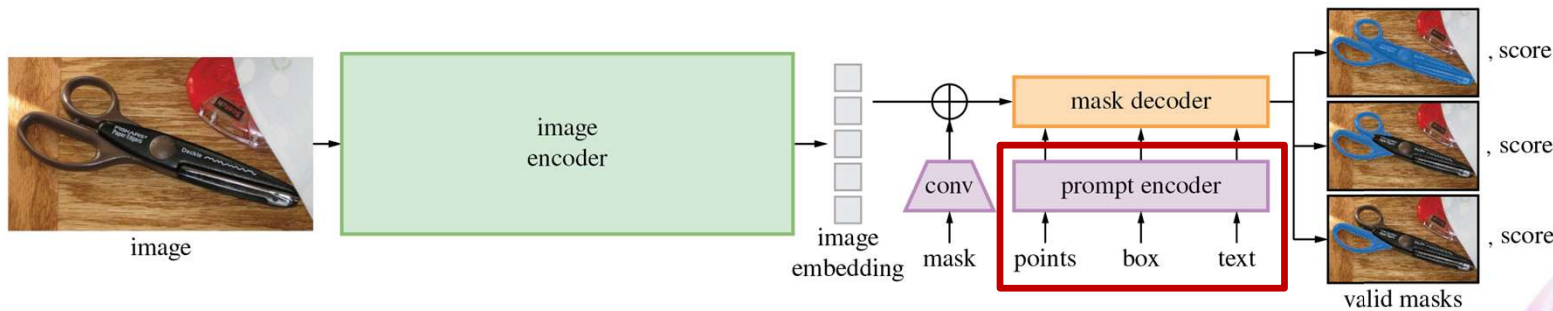
# Segment Anything (SAM)

The **Segment Anything Model (SAM)** introduced by Meta produces **high quality object masks from input prompts**, and it can be used to generate masks for all objects in an image.

## SAM 1: The Foundation (Promptable Segmentation)

**Core Idea:** Introduced promptable segmentation, focusing on interactive input (clicks, boxes, masks) to define an object. It has been trained on a **dataset of 11 million images and 1.1 billion masks**

**Key Feature:** Zero-shot transfer to new tasks and domains, making segmentation highly adaptable.



Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R.  
"Segment anything." *In ICCV 2023.*



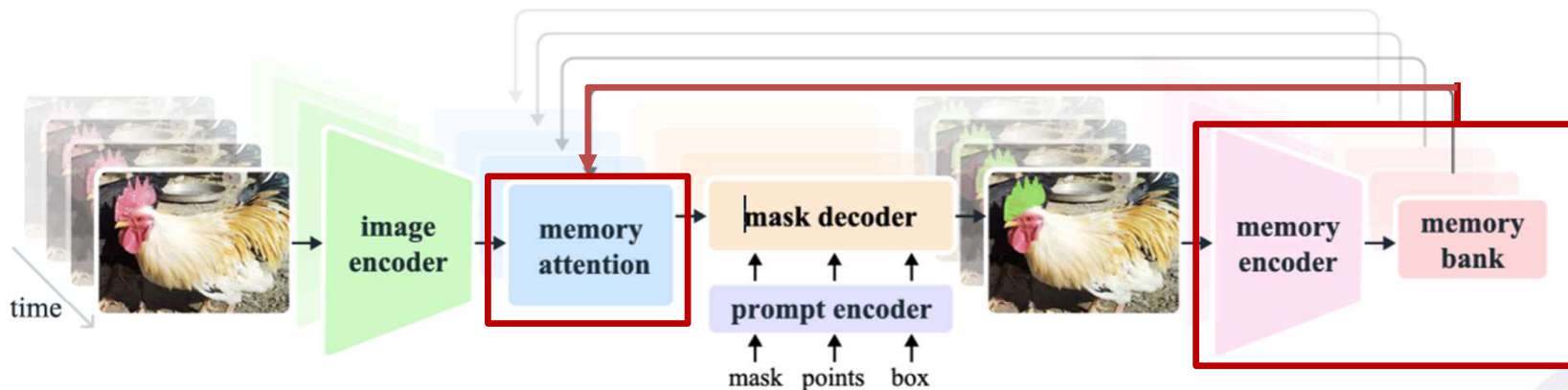
# Segment Anything (SAM)

## SAM 2: Video & Tracking Enhancement

**Core Idea:** Extended SAM's capabilities to **real-time video segmentation**.

**Key Feature:** Incorporated a temporal memory mechanism to track objects across frames, maintaining segmentation consistency.

**Limitation:** Still relied heavily on initial visual prompts for guidance.



<https://ai.meta.com/sam2/>

Carion, N., Gustafson, L., Hu, Y. T., Debnath, S., Hu, R., Suris, D., ... & Feichtenhofer, C.  
"Sam 3: Segment anything with concepts." In *arXiv 2025*.

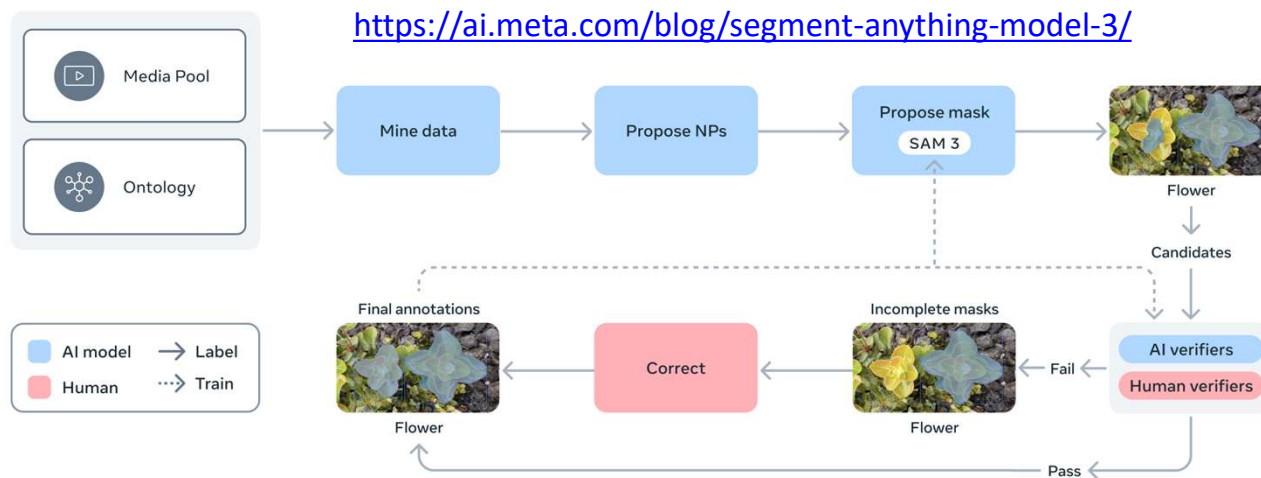


# Segment Anything (SAM)

## SAM 3: Promptable Concept Segmentation (The Big Leap)

**Core Idea:** Introduce the promptable concept segmentation capability: finding and segmenting all instances of a concept defined by a text or exemplar prompt.

**Key Feature: Unified Vision-Language Intelligence** — accepts **text prompts** — open-vocabulary short noun phrases — **and image exemplar prompts**, eliminating the constraints of fixed label sets.



Ravi, N., Gabeur, V., Hu, Y. T., Hu, R., Ryali, C., Ma, T., ... & Feichtenhofer, C.  
“Sam 2: Segment anything in images and videos”. In *ICLR 2025*.



# Summary

- Why do segmentation?
  - Inspiration from human perception – Gestalt theory
- Segmentation as clustering
  - K-means
- Superpixels – Graph-based algorithms
  - Graph cuts / normalized cuts
  - Felzenswalb’s method
- Deep Learning methods
  - FCN Semantic Segmentation
  - Mask R-CNN Instance Segmentation
  - ViT Semantic Segmentation
- Next
  - Local Invariant Features
- Recommended Reading
  - [Forsyth & Ponce] Chapter 15

