Welcome to Computer Vision

Computer Vision

Dr. Muhammad Tahir

DEPARTMENT OF COMPUTER SCIENCE,

FAST-NUCES, Peshawar

Course Details

LECTURES: Monday & Wednesday

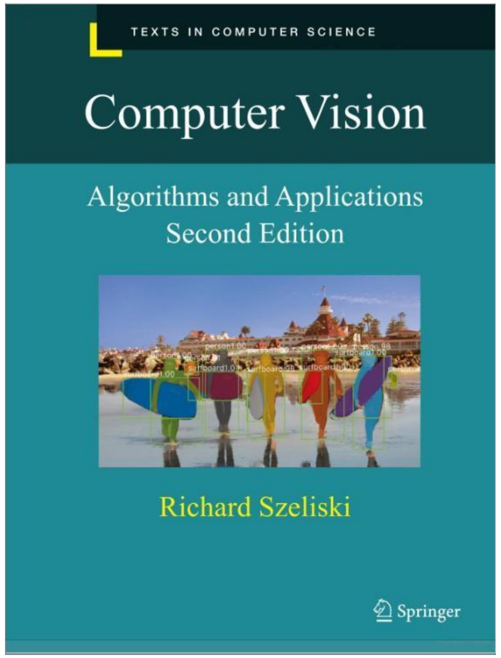
TIMINGS:

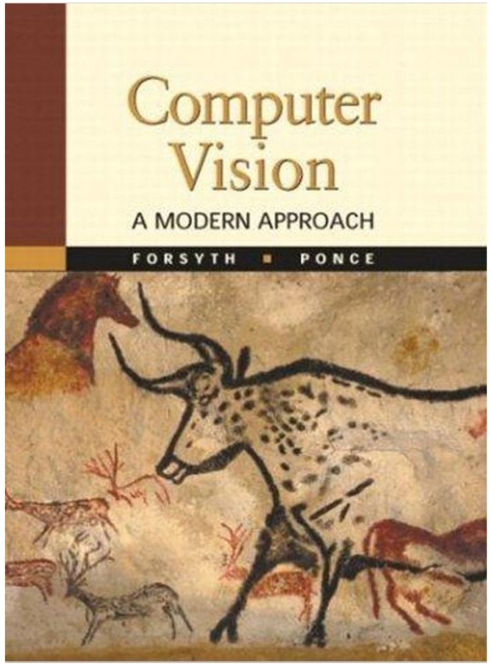
9:30 am – 11:00 am

MY OFFICE:

OFFICE HOURS:

EMAIL: m.tahir@nu.edu.pk





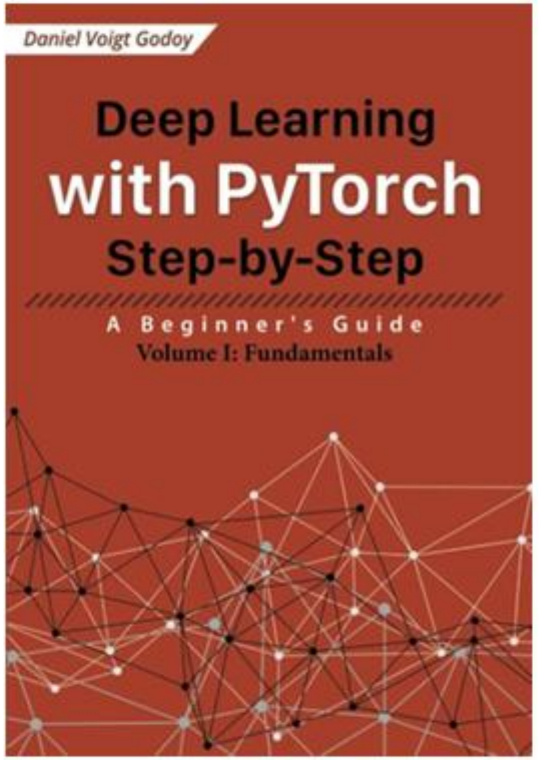
References

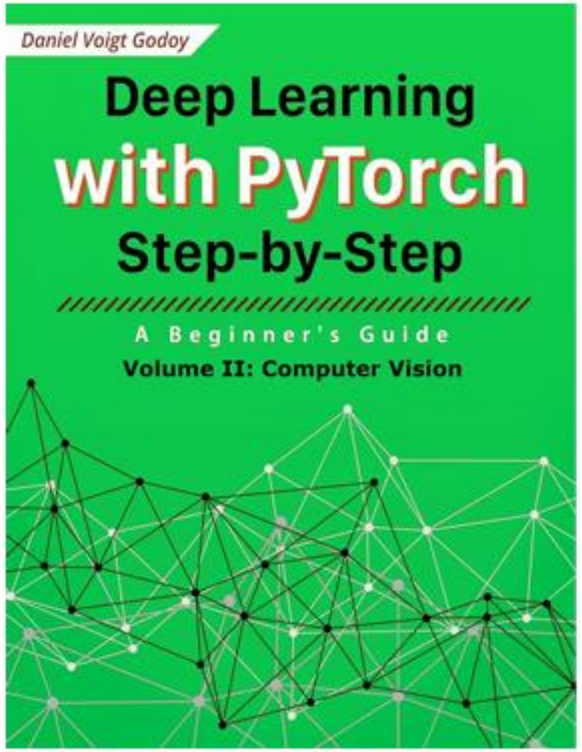
The material in these slides are based on:

1. Rick Szeliski’s book: [Computer Vision: Algorithms and Applications]
2. Forsythe and Ponce: [Computer Vision: A Modern Approach]

Recommended Books

Deep Learning with PyTorch Step-by-Step by Daniel Voigt Godoy





Course Learning Outcomes

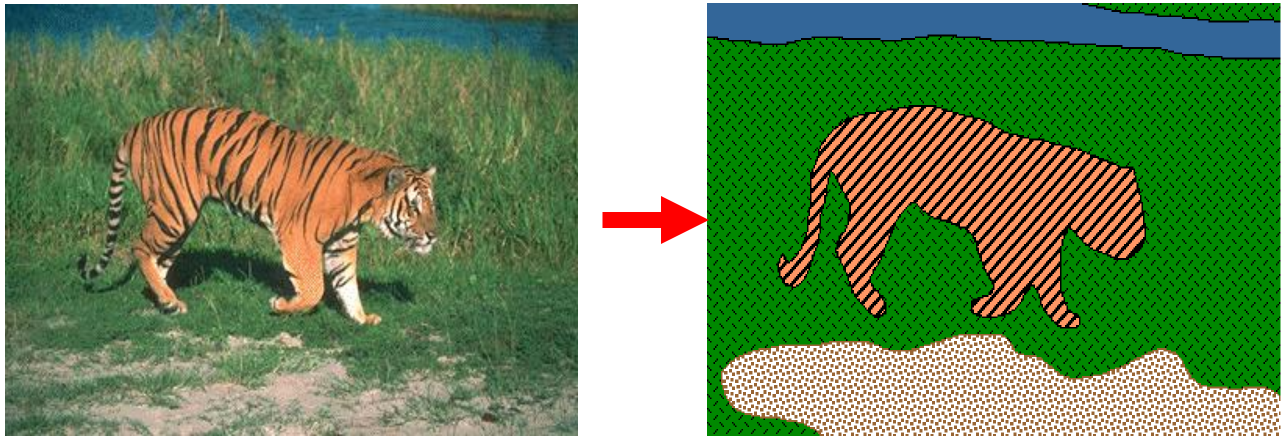
No	CLO	Domain	Taxonomy Level	PLO
1	Understand the view geometry concepts, multi-scale representation, edge detection and detection of other primitives, stereo, motion and object recognition.	Cognitive		
2	Assess which methods to use for solving a given problem, and analyse the accuracy of the methods Skills	Cognitive		
3	Apply appropriate image processing methods for image filtering, image restoration, image reconstruction, segmentation, classification and representation	Cognitive		

Outline

Image Segmentation - I

Image Segmentation

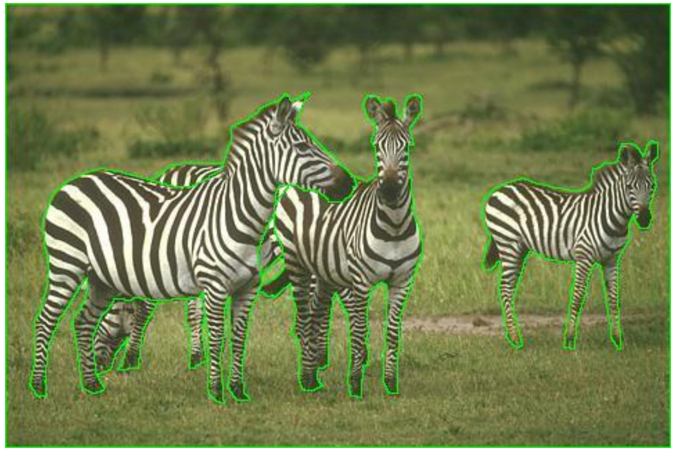
Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels

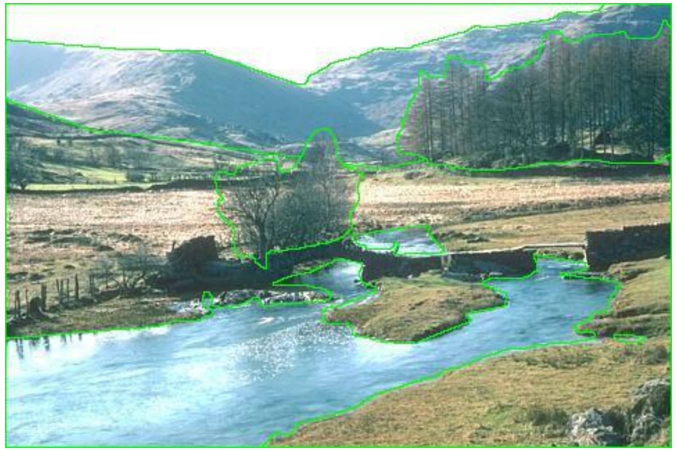


Segment image into:

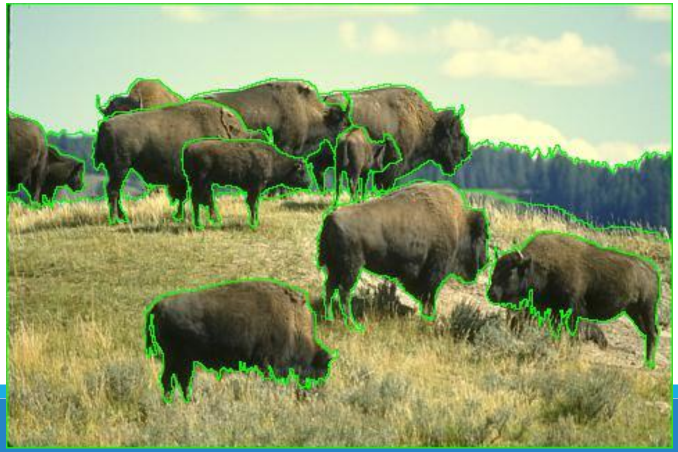
1. Meaningful regions (coherent objects)
2. Linear structures (line, curve,…)
3. Shapes (circles, ellipses, …)

Image Segmentation





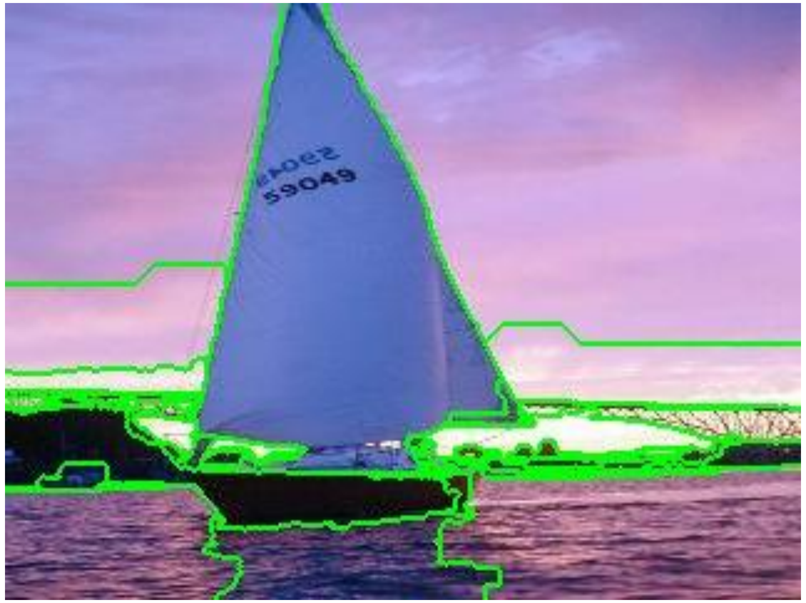




The goals of segmentation

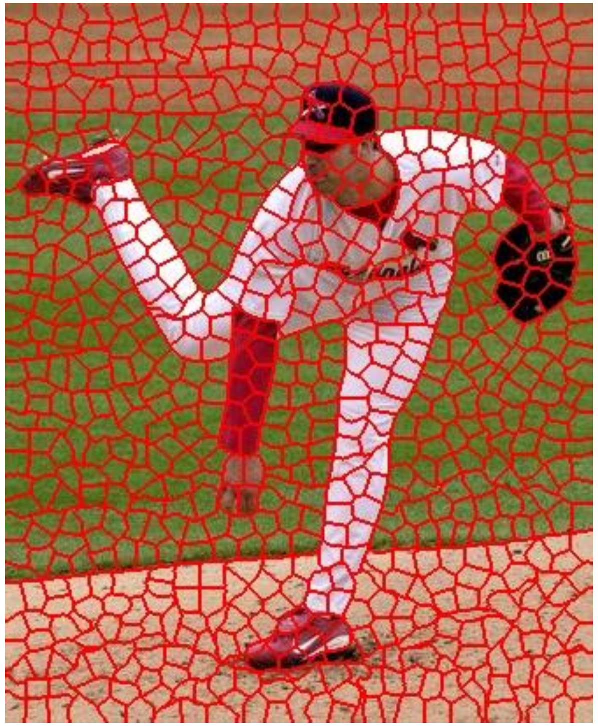
- Separate image into **coherent** "objects"

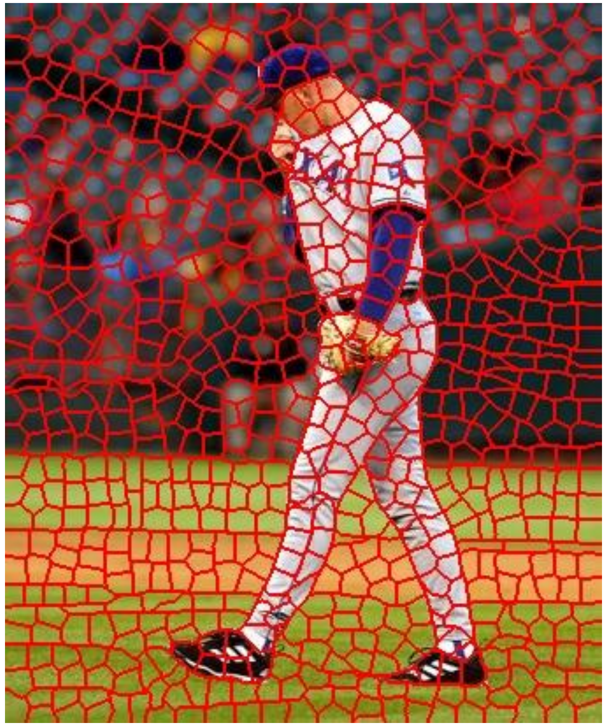
Image | Human segmentation



The goals of segmentation

- Separate image into **coherent** "objects"
- Group together **similar-looking** pixels for efficiency of further processing

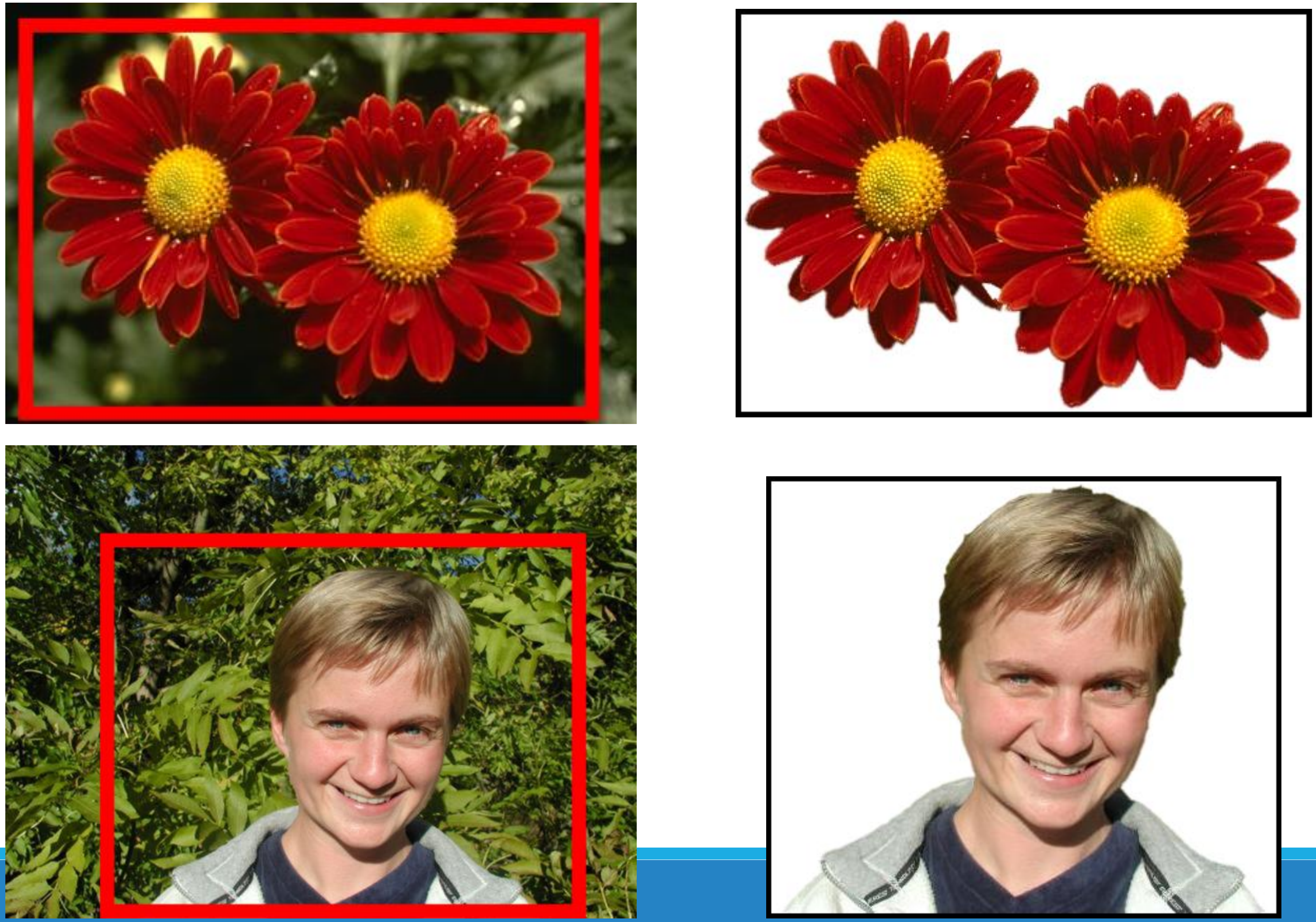




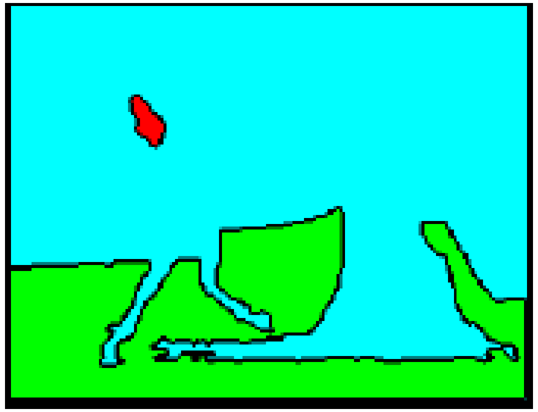
X. Ren and J. Malik. **Learning a classification model for segmentation.** ICCV 2003.

Segmentation for feature support

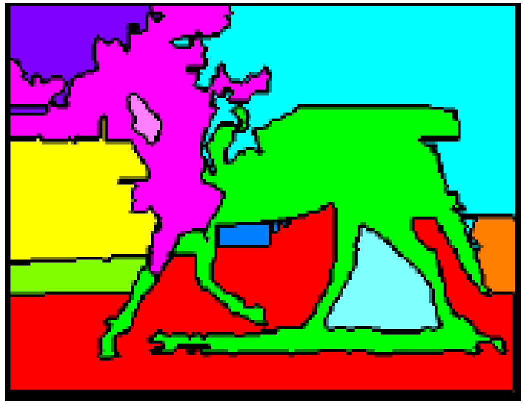
Segmentation as a result



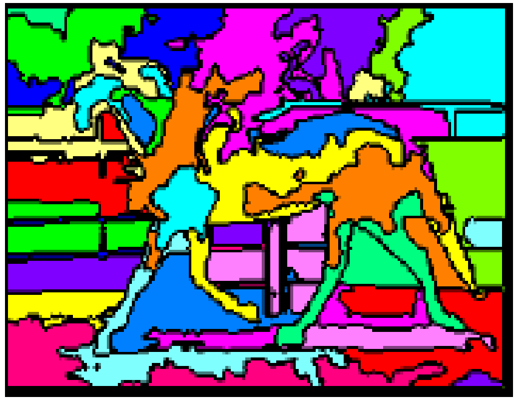
Segmentation



Under segmentation



Multiple Segmentations



Over segmentation

Clustering

- One way to think about "segmentation" is **Clustering**
- **Clustering:** group together similar data points and represent them with a single token
  - **Tokens:** whatever we need to group (pixels, points, surface elements, etc., etc.)
- **Key Challenges:**
  - What makes two points/images/patches similar?
  - How do we compute an overall grouping from pairwise similarities?

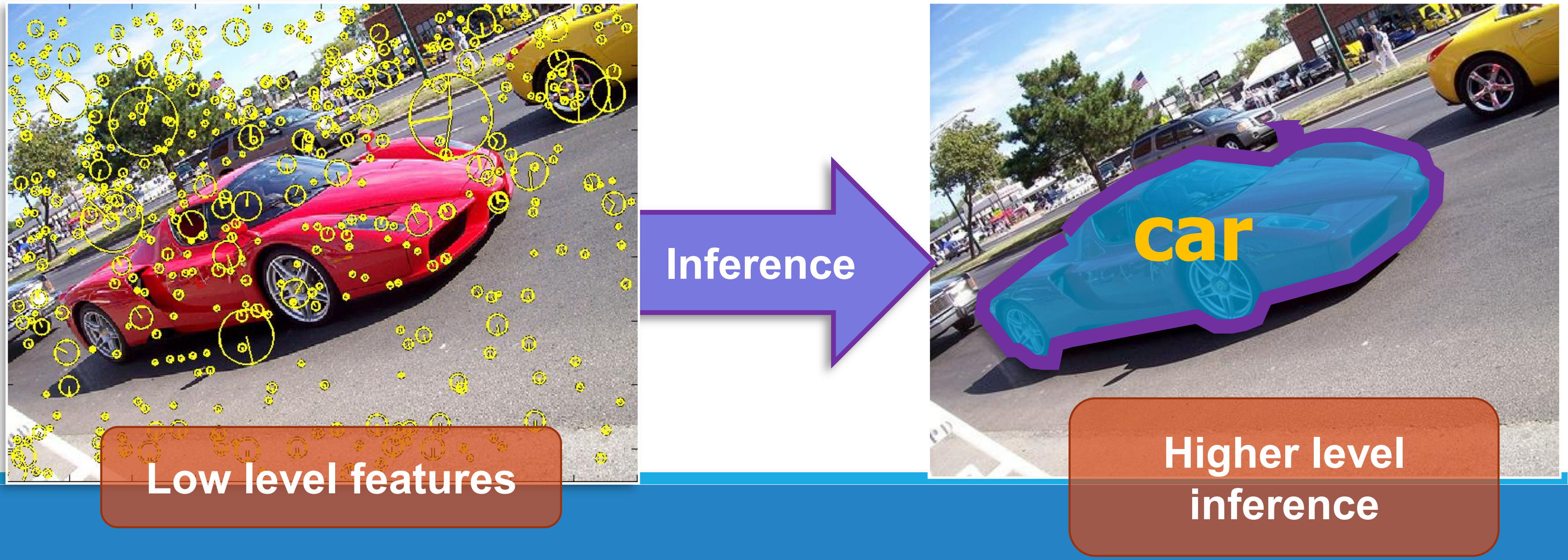
Clustering (grouping)

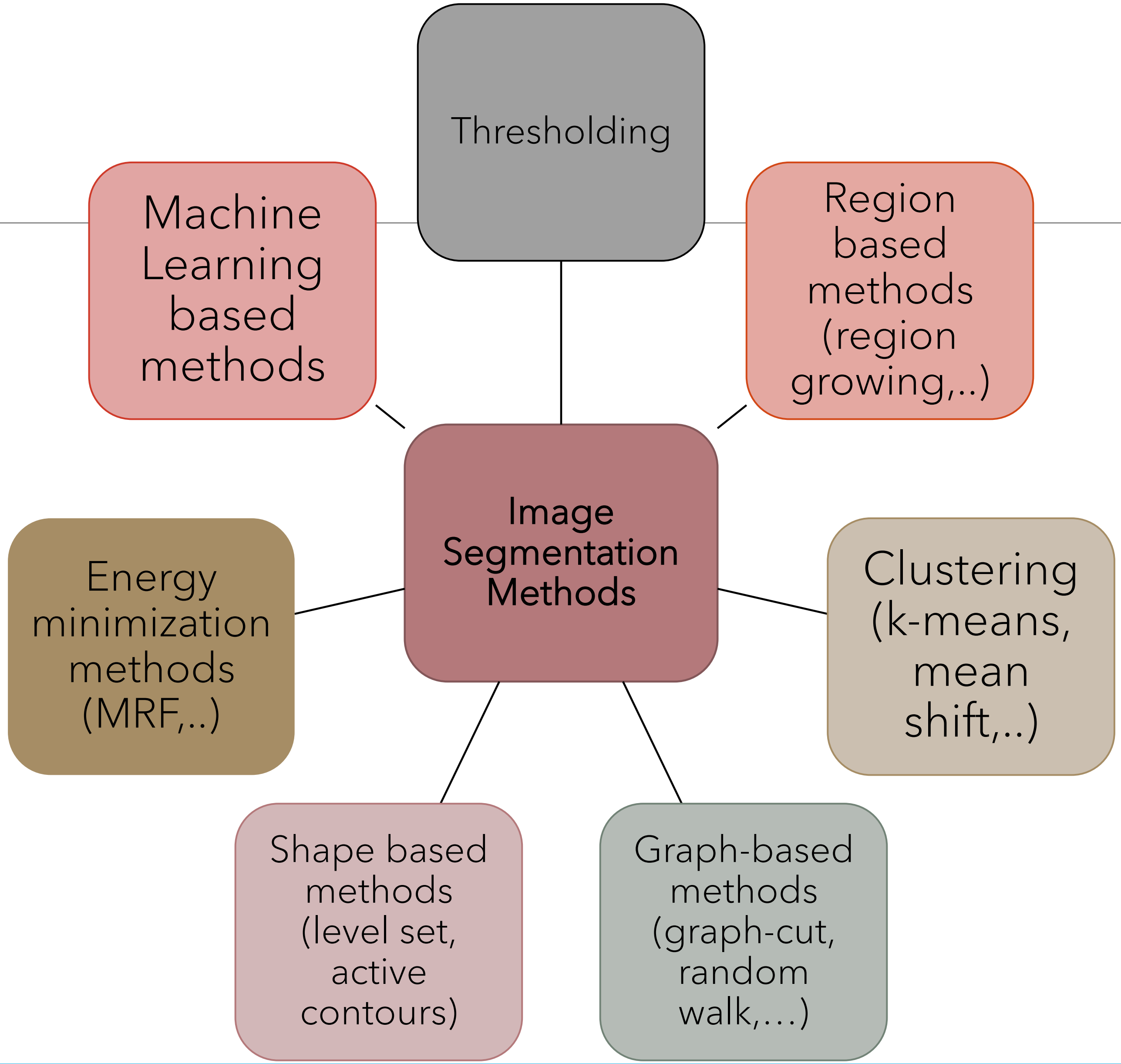
- Top-down vs. bottom-up segmentation
  - **Top down:** pixels belong together because they are from the same object
  - **Bottom up:** pixels belong together because they look similar

- Hard to **measure** success
  - What is interesting depends on the **application**.

Image Segmentation

- In computer vision, image segmentation is one of the oldest and most widely studied problems
  - Early techniques -> region splitting or merging
  - More recent techniques -> Energy minimization, hybrid methods, and deep learning




Thresholding
Machine Learning based methods
Region based methods (region growing,..)
Image Segmentation Methods
Energy minimization methods (MRF,..)
Clustering (k-means, mean shift,..)
Shape based methods (level set, active contours)
Graph-based methods (graph-cut, random walk,…)

K-Means Clustering Results

- **K-means** clustering based on intensity or color is essentially **vector quantization** of the image attributes
  - Clusters don't have to be spatially coherent

Image

Intensity-based clusters

Color-based clusters

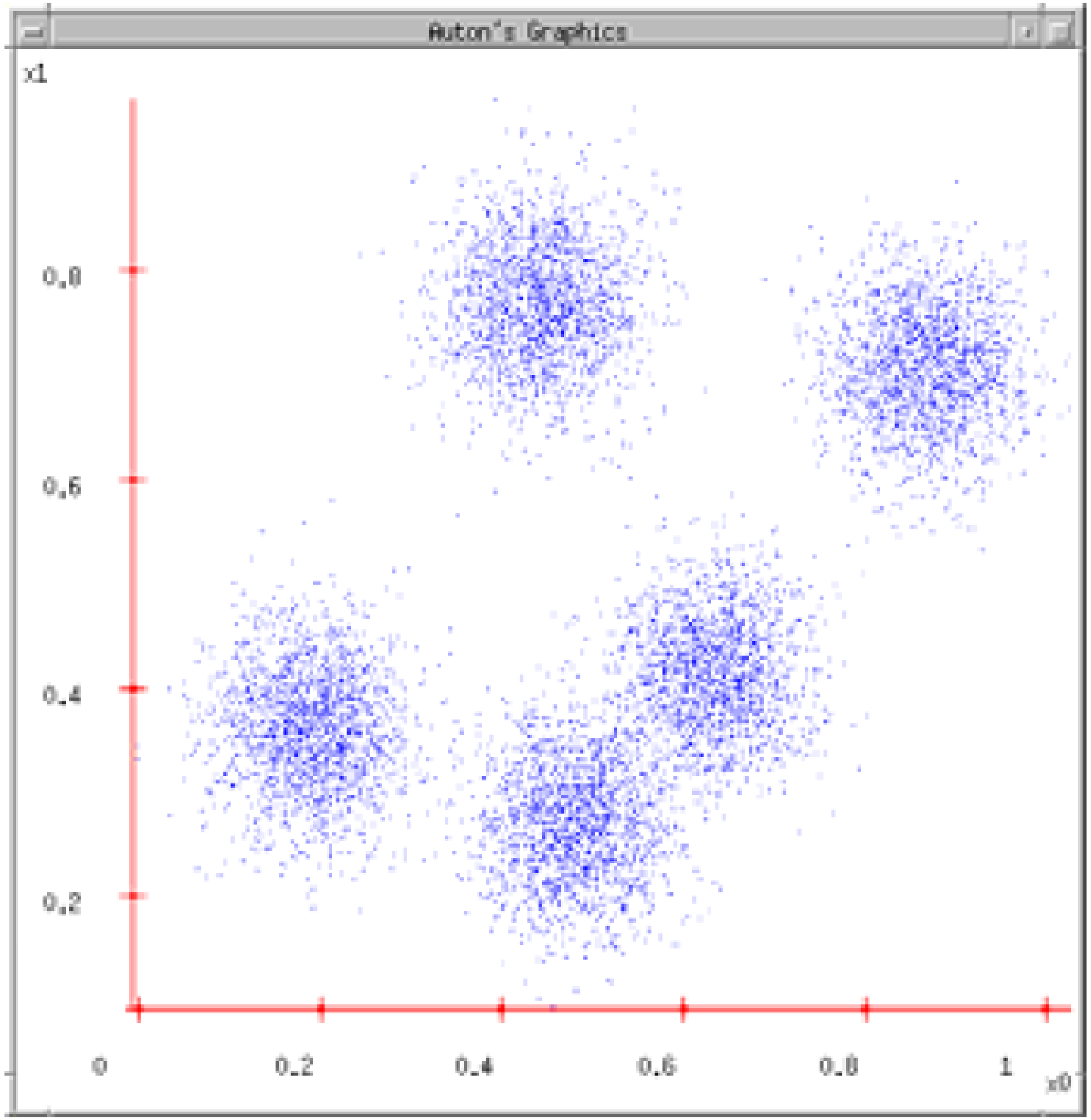
K-means clustering: Algorithm

1. Randomly initialize the **cluster centers**, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
   - For each point p, **find the closest c_i**. Put p into cluster i
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

- **Properties**
  - Will always converge to ***some*** solution
  - Can be a "local minimum"
    - does not always find the global minimum of objective function

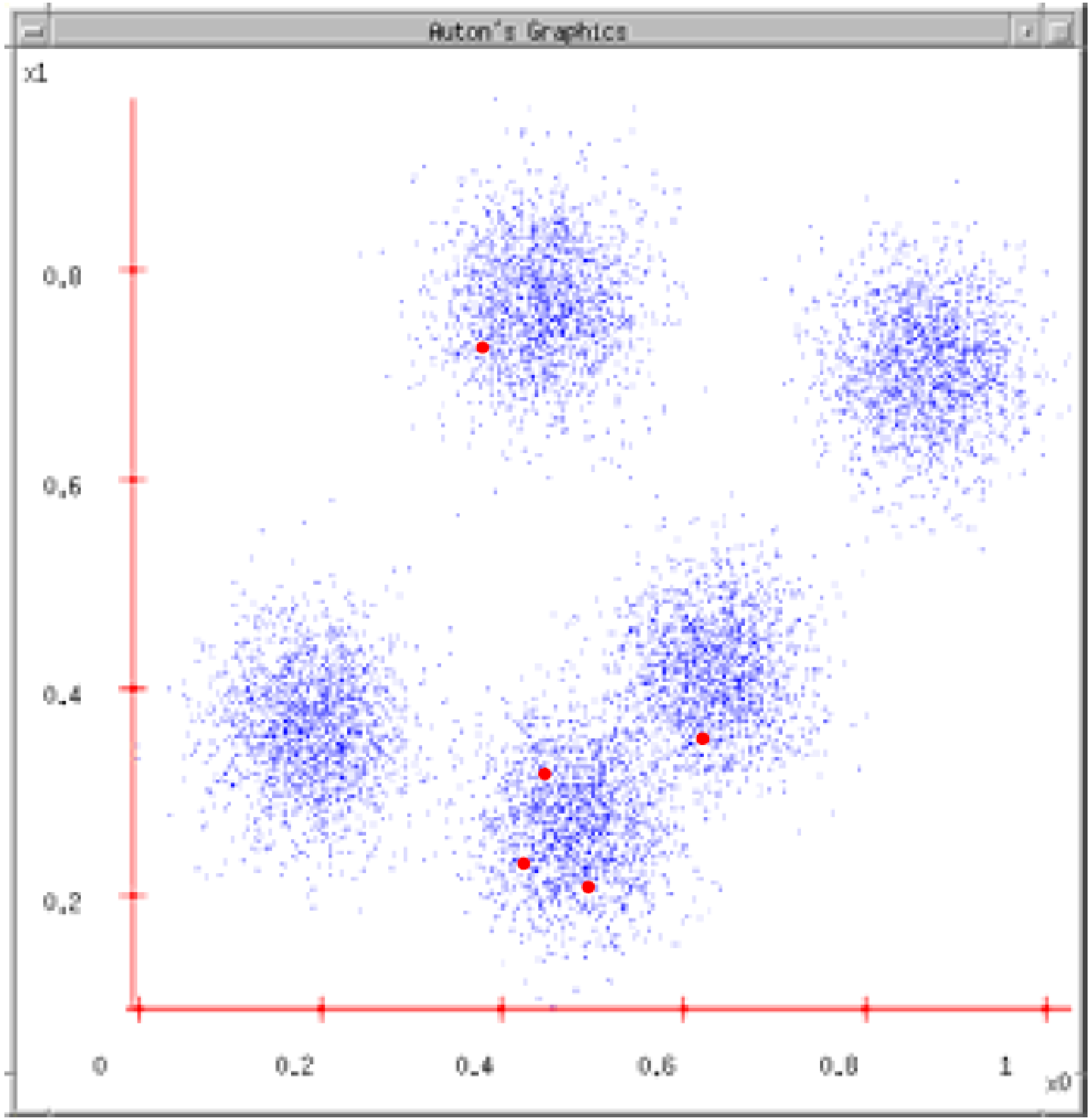
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*



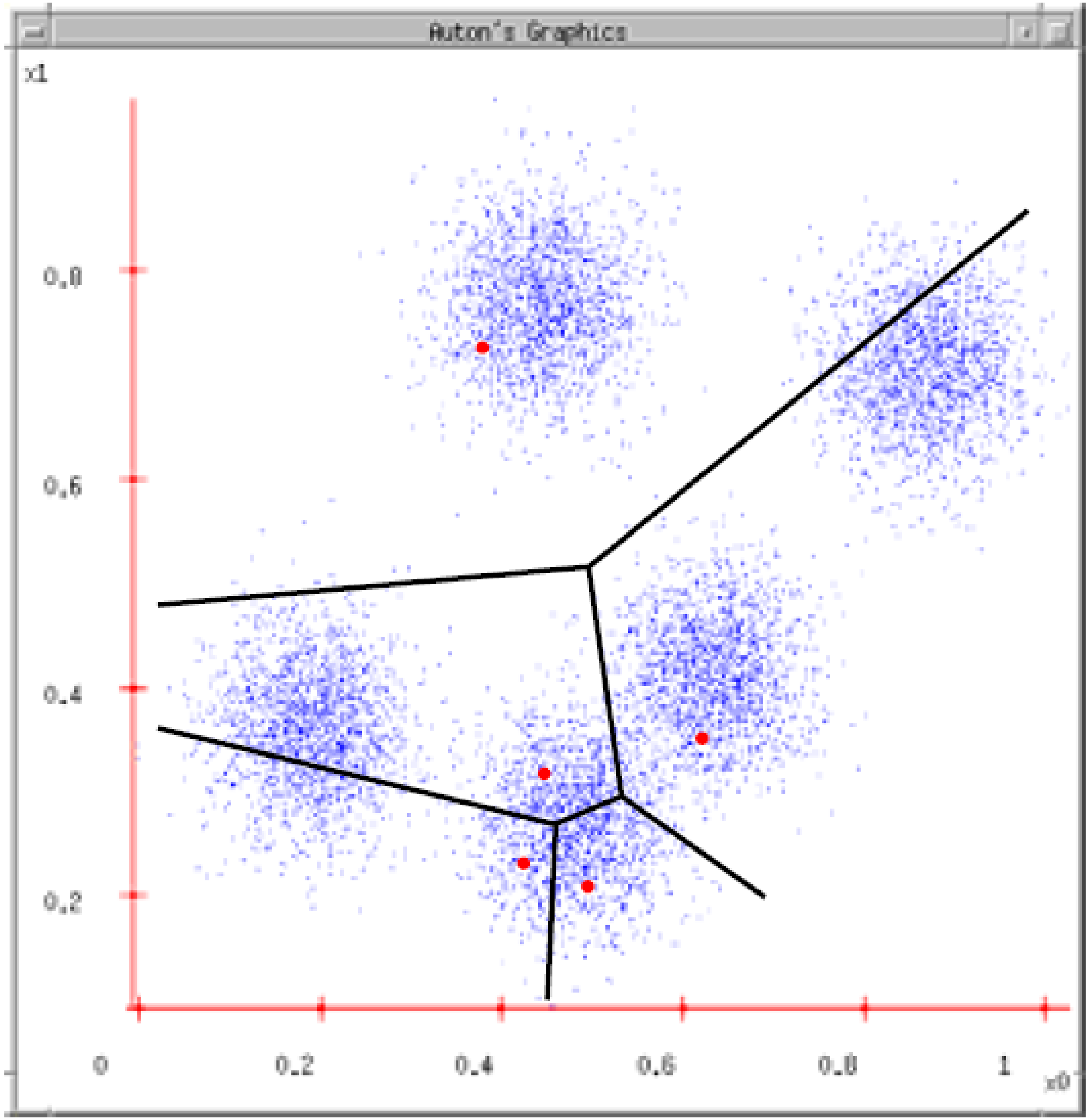
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



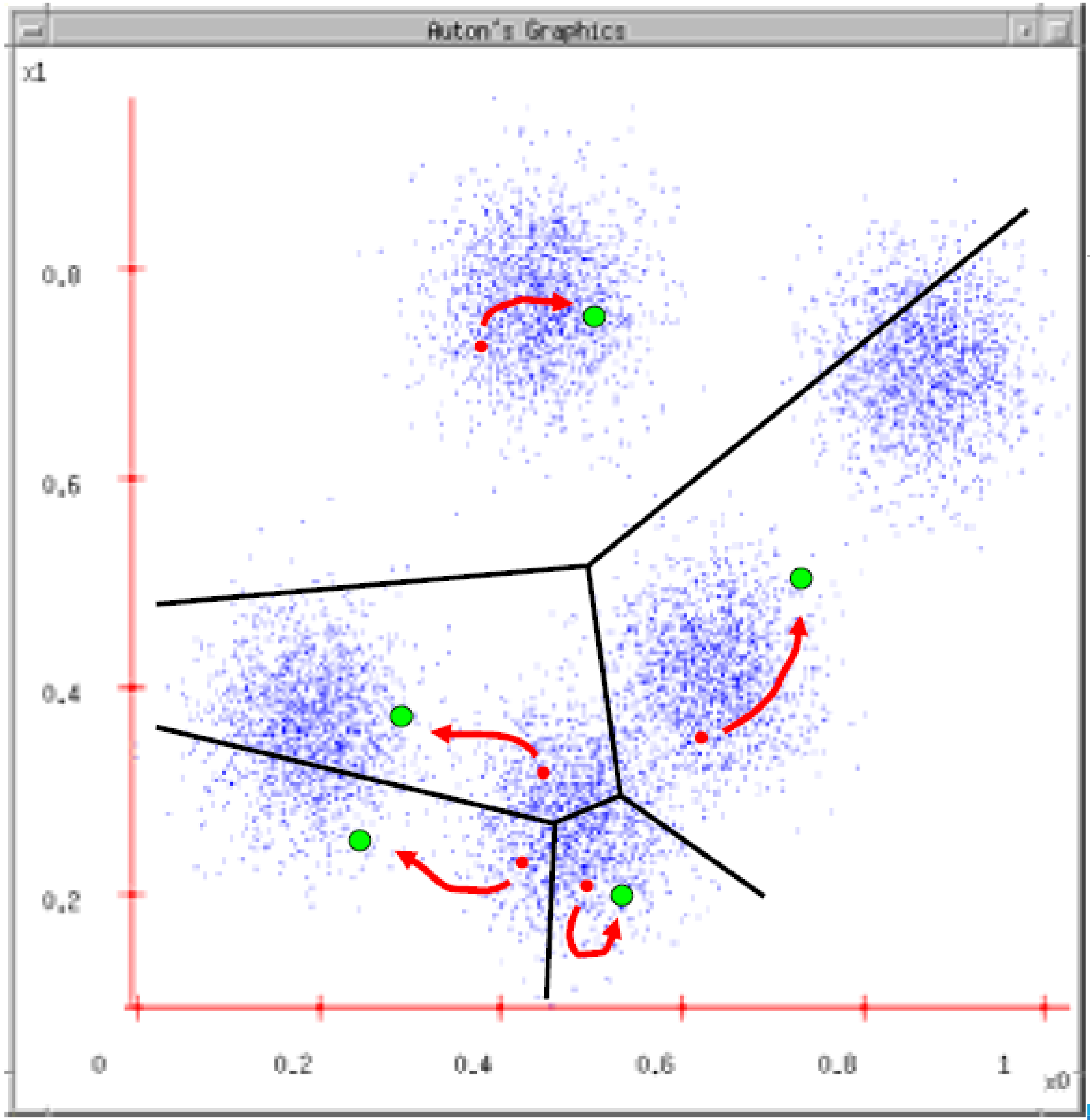
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



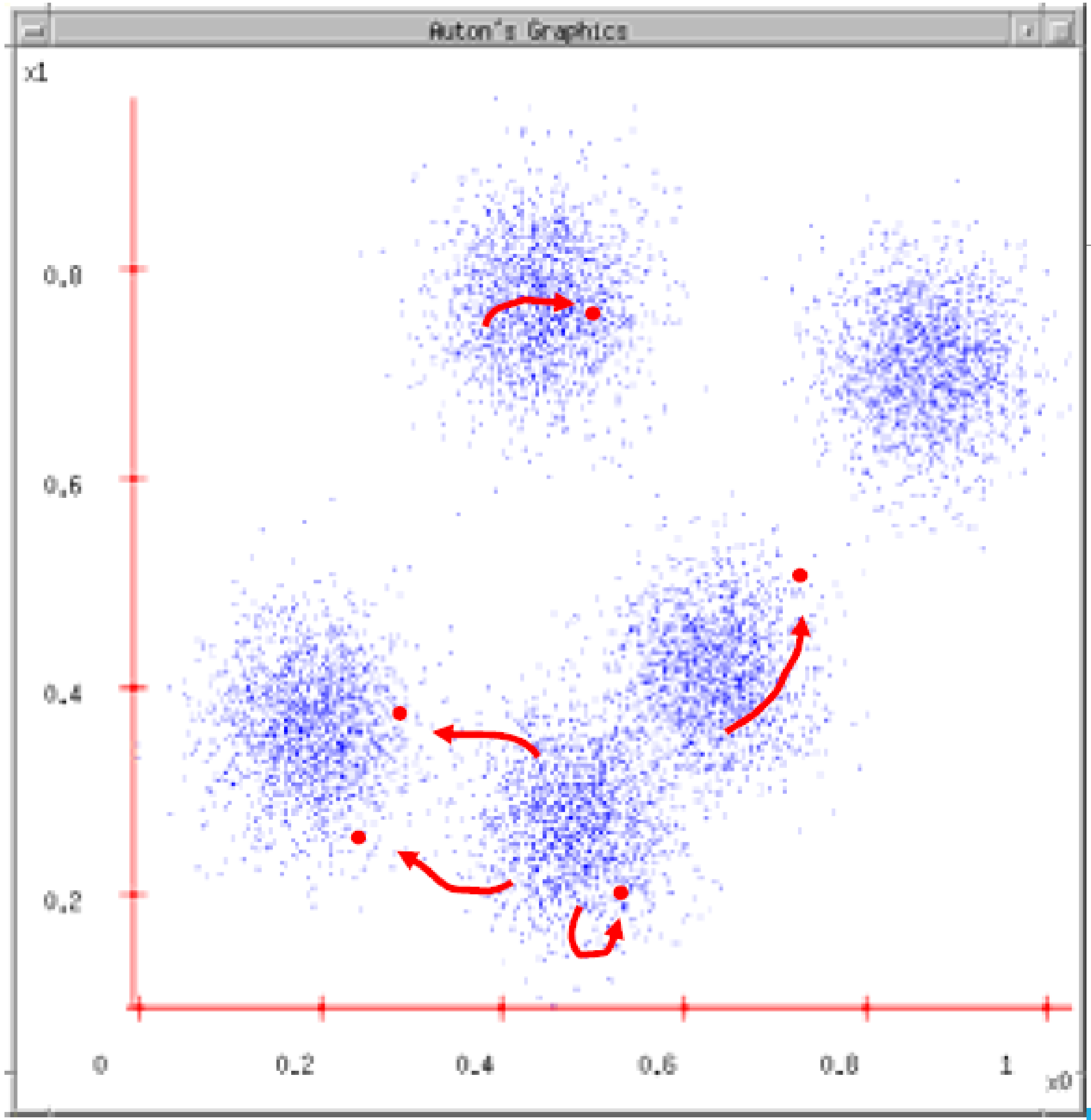
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

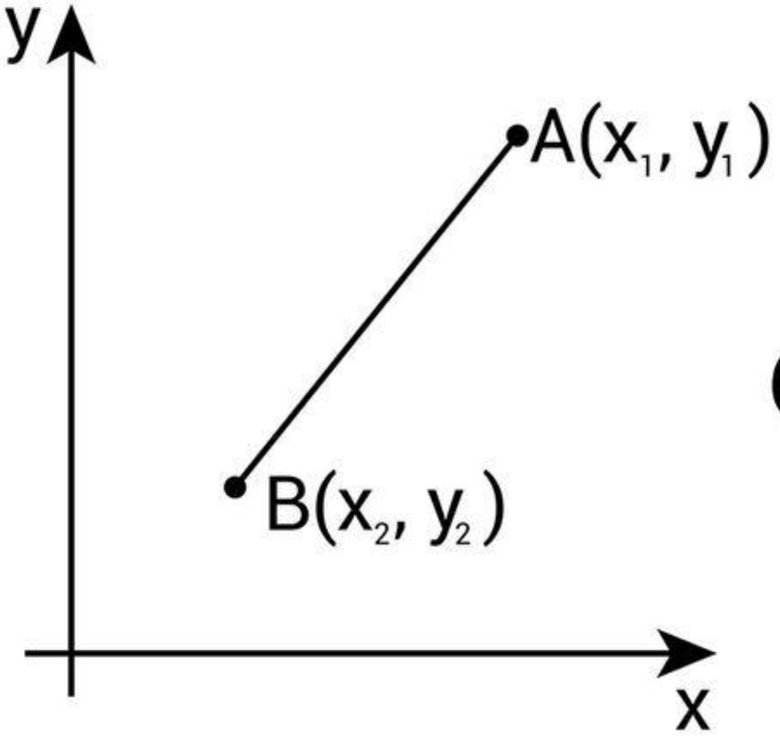


K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



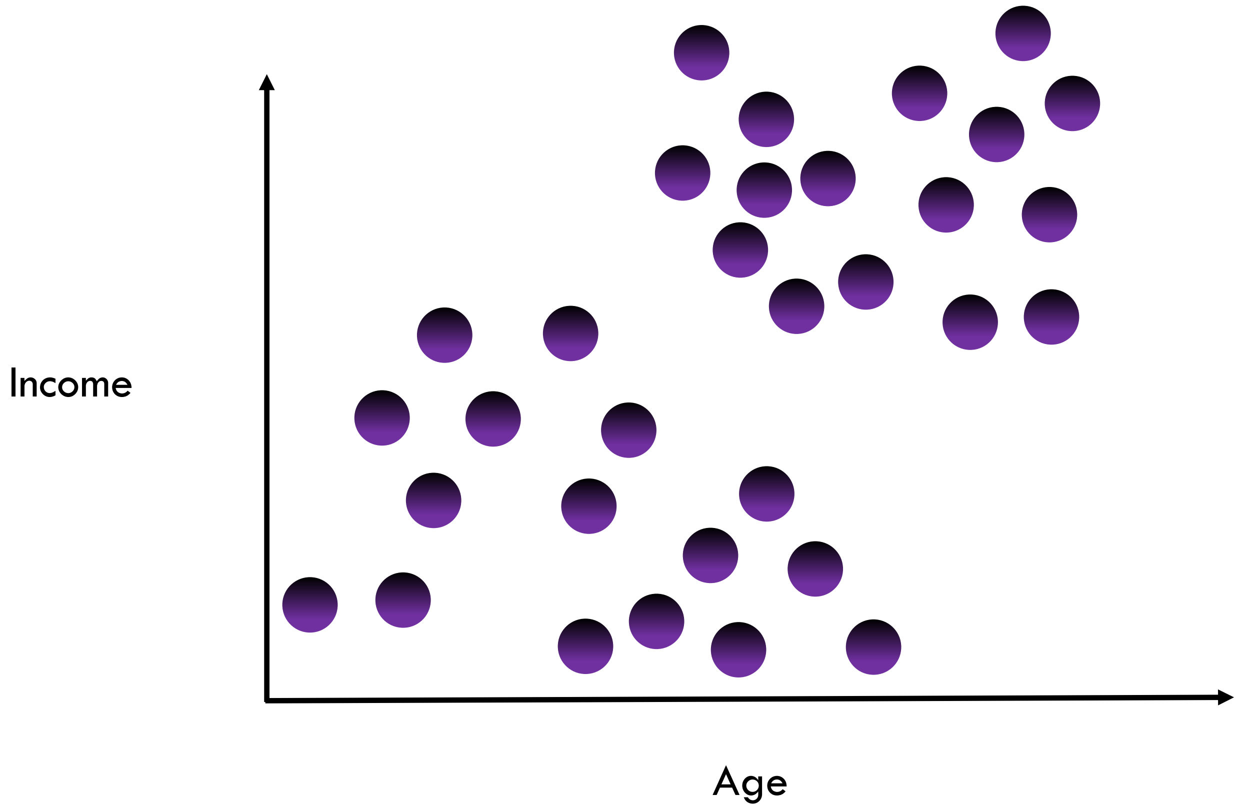
Distance Formula



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

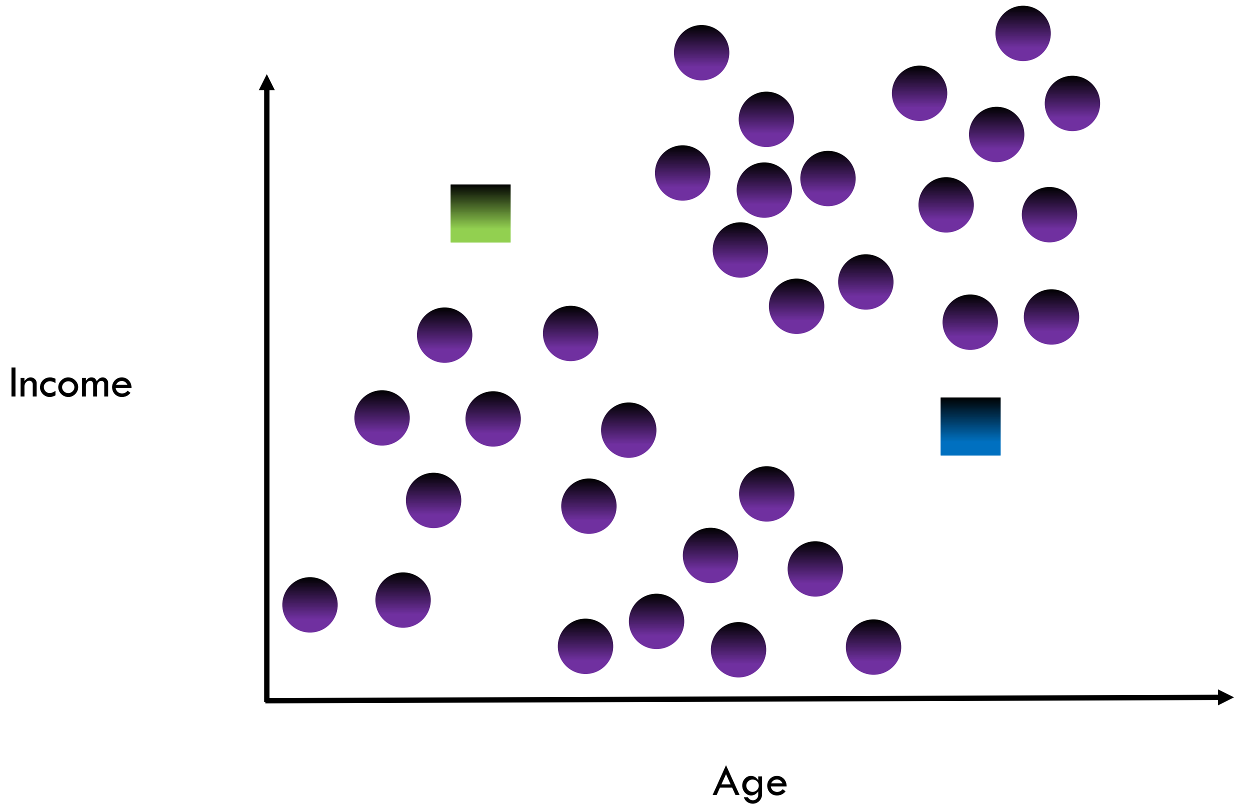
K-Means algorithm

- **Objective:** Find 2 clusters
- Similar example, this time we have 2 features. Age and Income
- Visually, how would you cluster this dataset into 2 clusters?
- The answer is obvious to us, but let's see if we can get there algorithmically.



K-Means algorithm

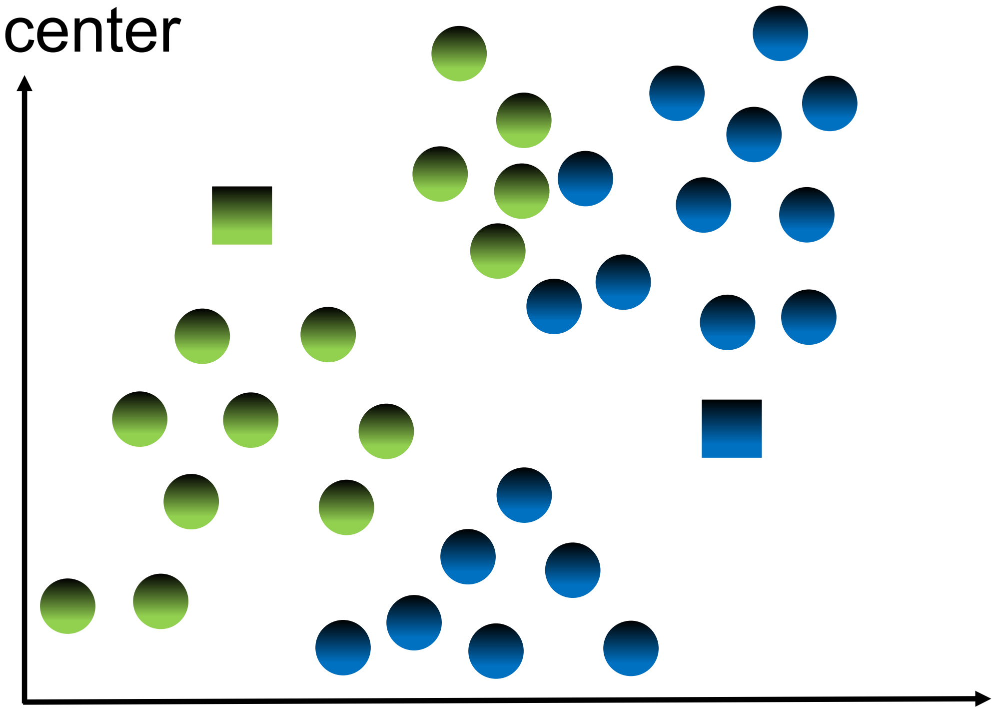
- **K = 2**, Randomly assign cluster centers
- Since we prescribed 2 clusters, initialize the algorithm by picking 2 random points.
- These are going to be the center of the clusters.
- Name the clusters. Here we're doing this by colorcoding.



K-Means algorithm

- For each example in our space, determine which cluster it belongs to by computing distance and getting the closest.
- Here, in the first iteration, the examples are color coded like this.
- So now, every point belongs to a cluster, but we are not done, as this assignment is somewhat arbitrary, and it has not converged yet.

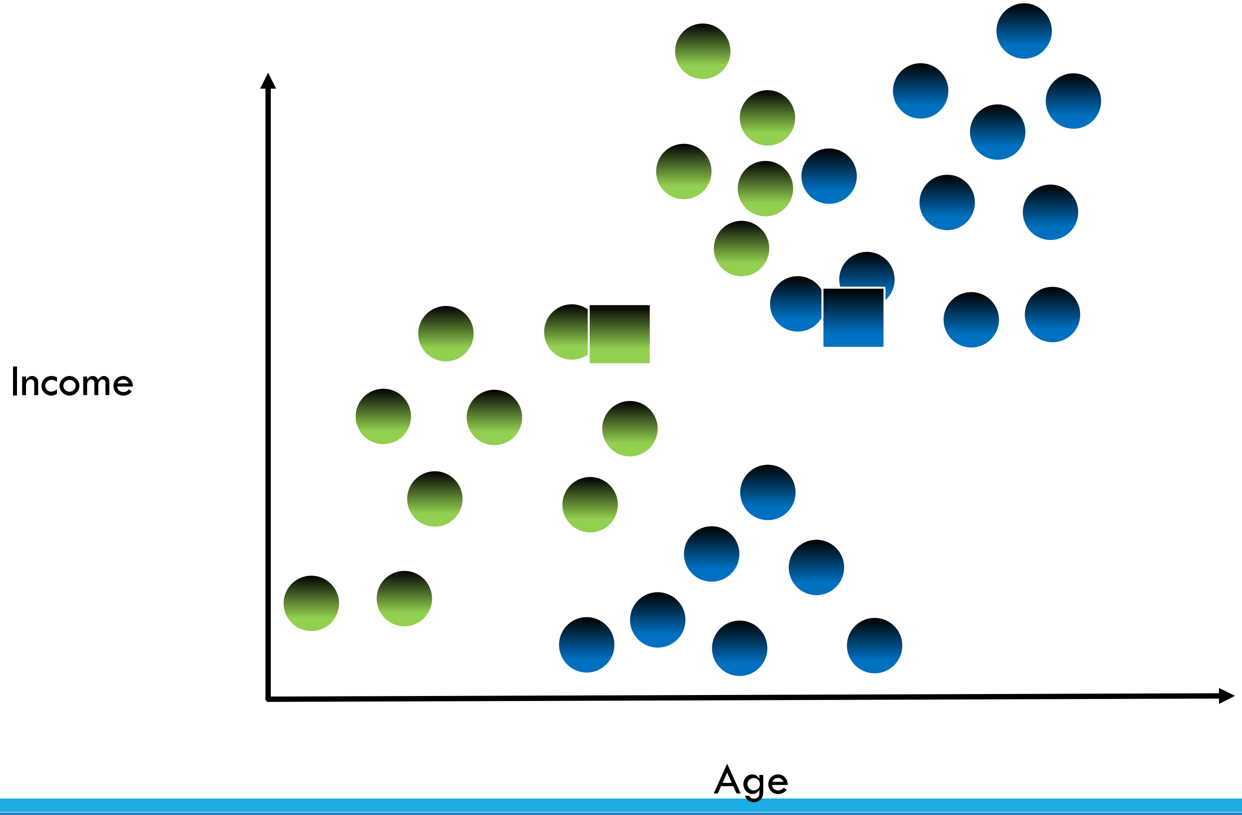
K-Means K = 2,
Each point belongs to closest center



K-Means algorithm

- Second step is to adjust the points (so called cluster centers) to the new center of the clusters.
- The new location of green square is right on the middle point of all the green circles. Same for blue.
- We’re thru the first iteration. We are going to keep repeating this process, until no example is assigned to a different cluster.

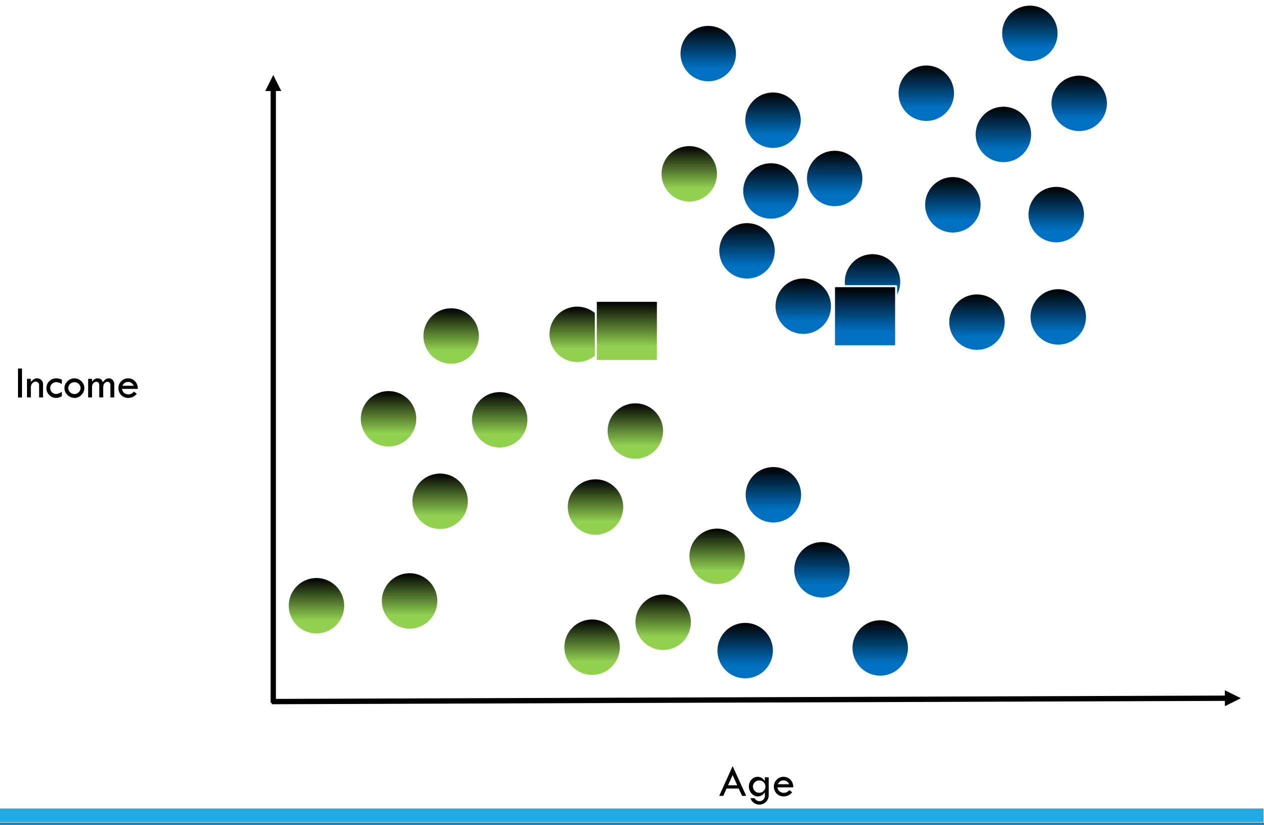
K = 2, Move each center to cluster's mean



K-Means algorithm

- So let us see the first step of the 2nd iteration.
- With our new cluster centers in place, identify which cluster each point belongs to again.

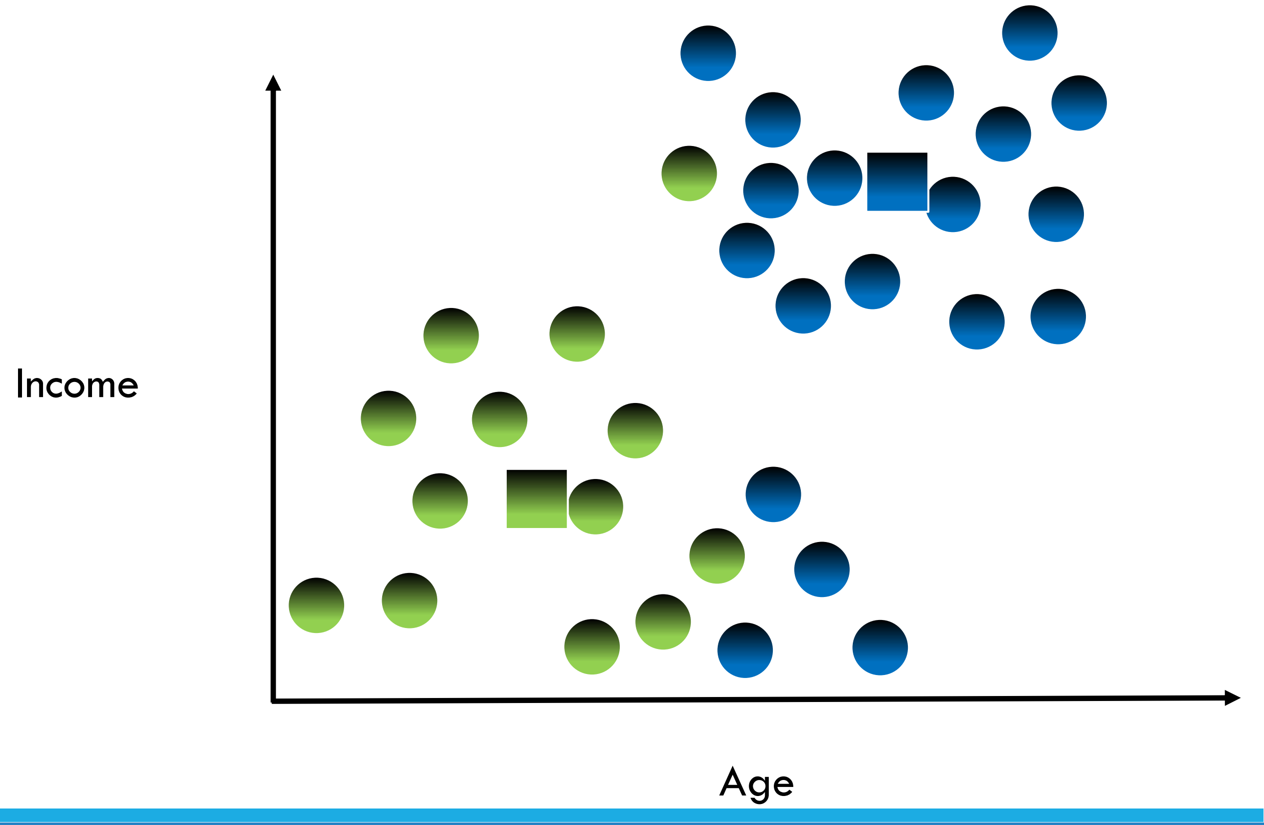
K = 2, Each point belongs to closest center



K-Means algorithm

And move the centers to the new center.

K = 2, Move each center to cluster's mean

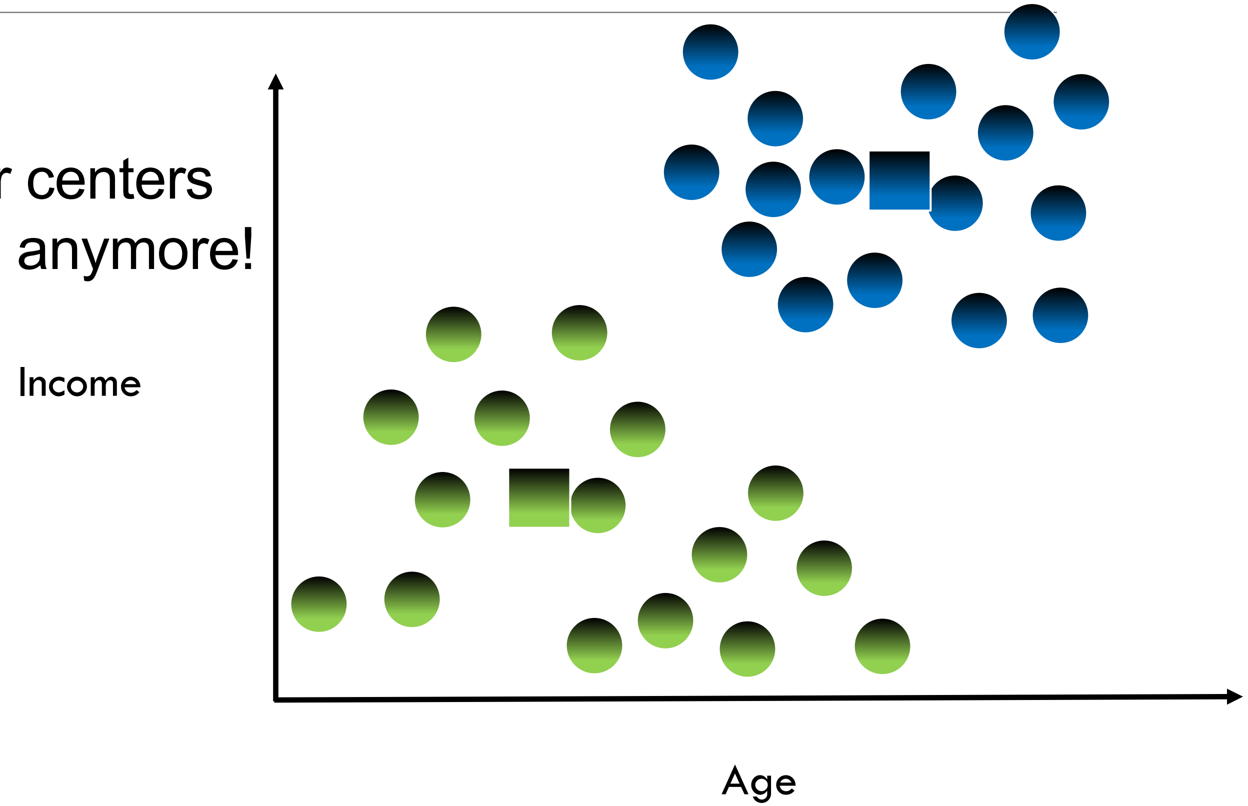


K-Means algorithm

K = 2, Points do not change
→ Converged!

And the 3rd iteration

The cluster centers
do not move anymore!

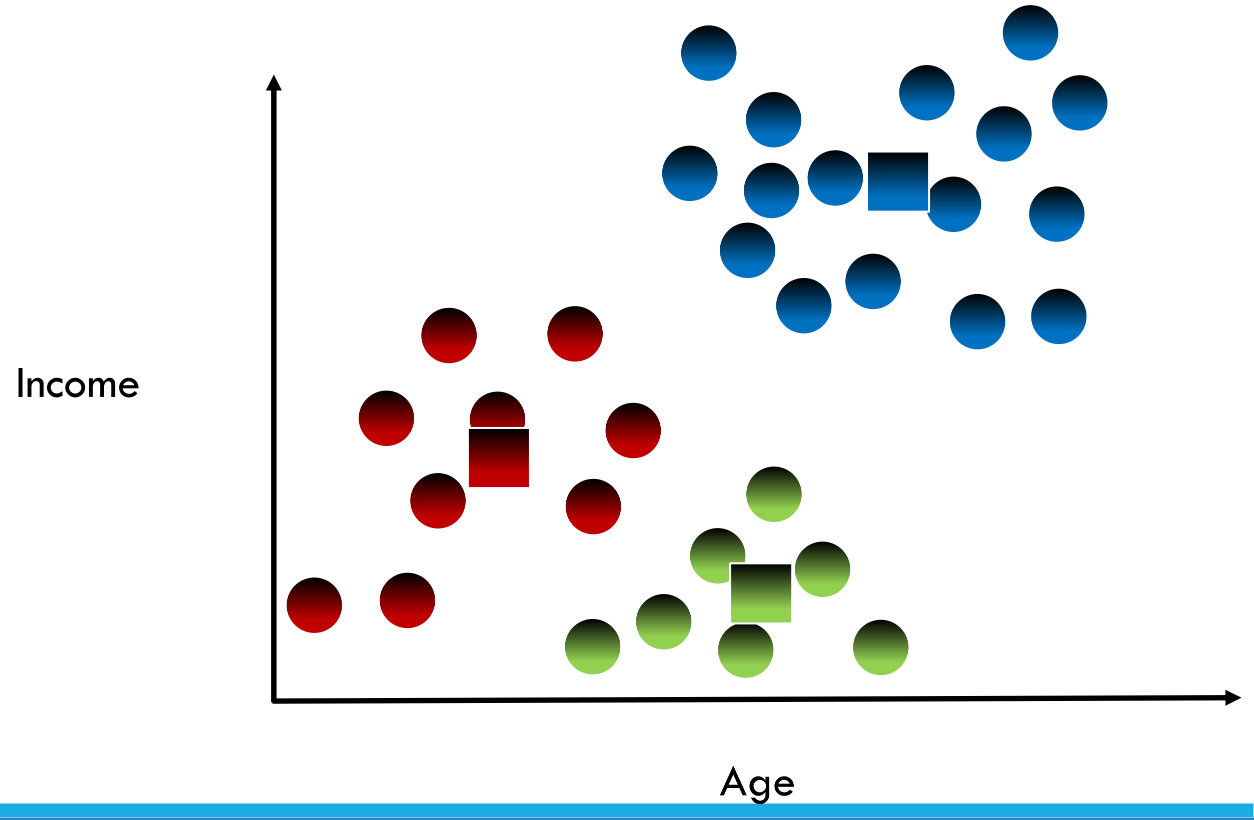


Each point belongs to
closest center

K-Means algorithm

- For 3 clusters, clusters can look like this.
- However, there can be other solutions.

K-Means Algorithm: K = 3

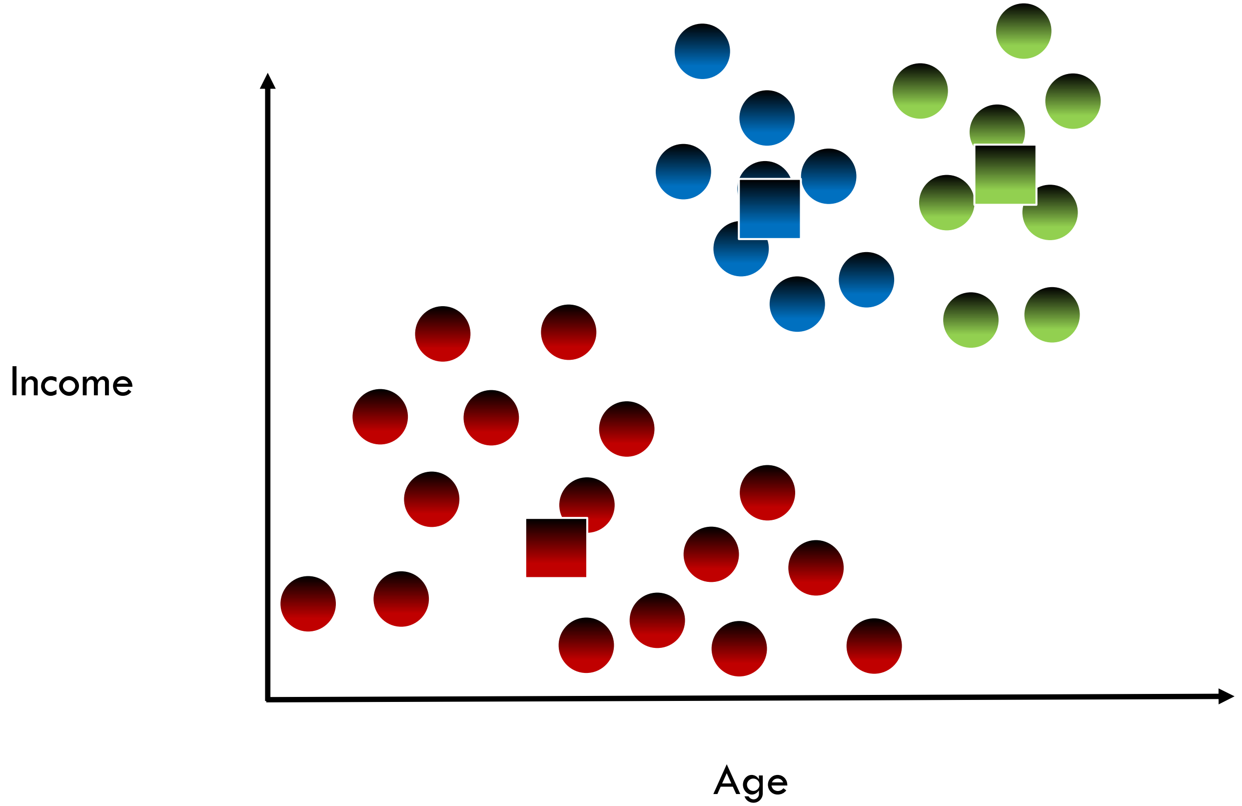


K-Means algorithm

K-Means Algorithm: K = 3

Such as this.

- The problem with the K-means algorithm is, that it is sensitive to the choice of initial points.
- Different initial configurations may yield different results (it may converge to local optimas)

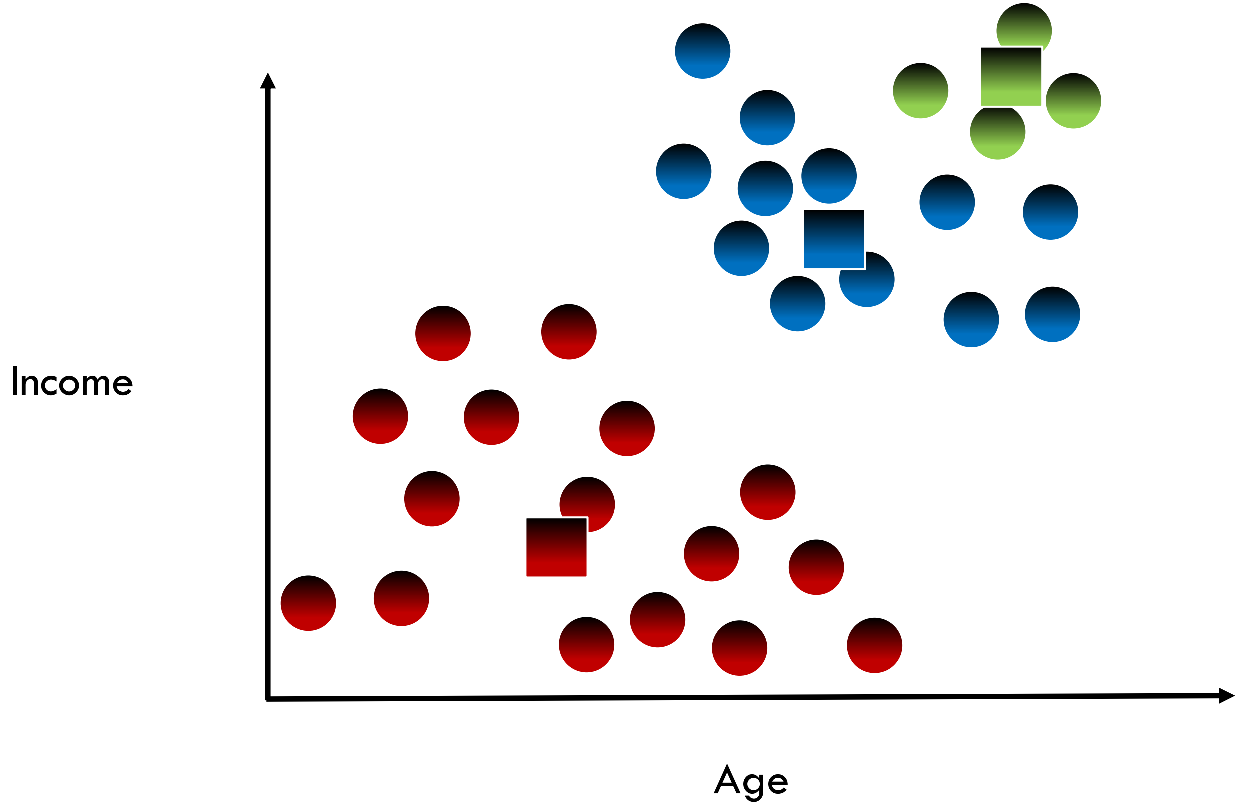


Results depend on initial cluster assignment!

K-Means algorithm

- Here is another local optimum. Which clustering makes more sense?
- We need a way of judging the converged results, and rank them according to "goodness"

K-Means Algorithm: K = 3



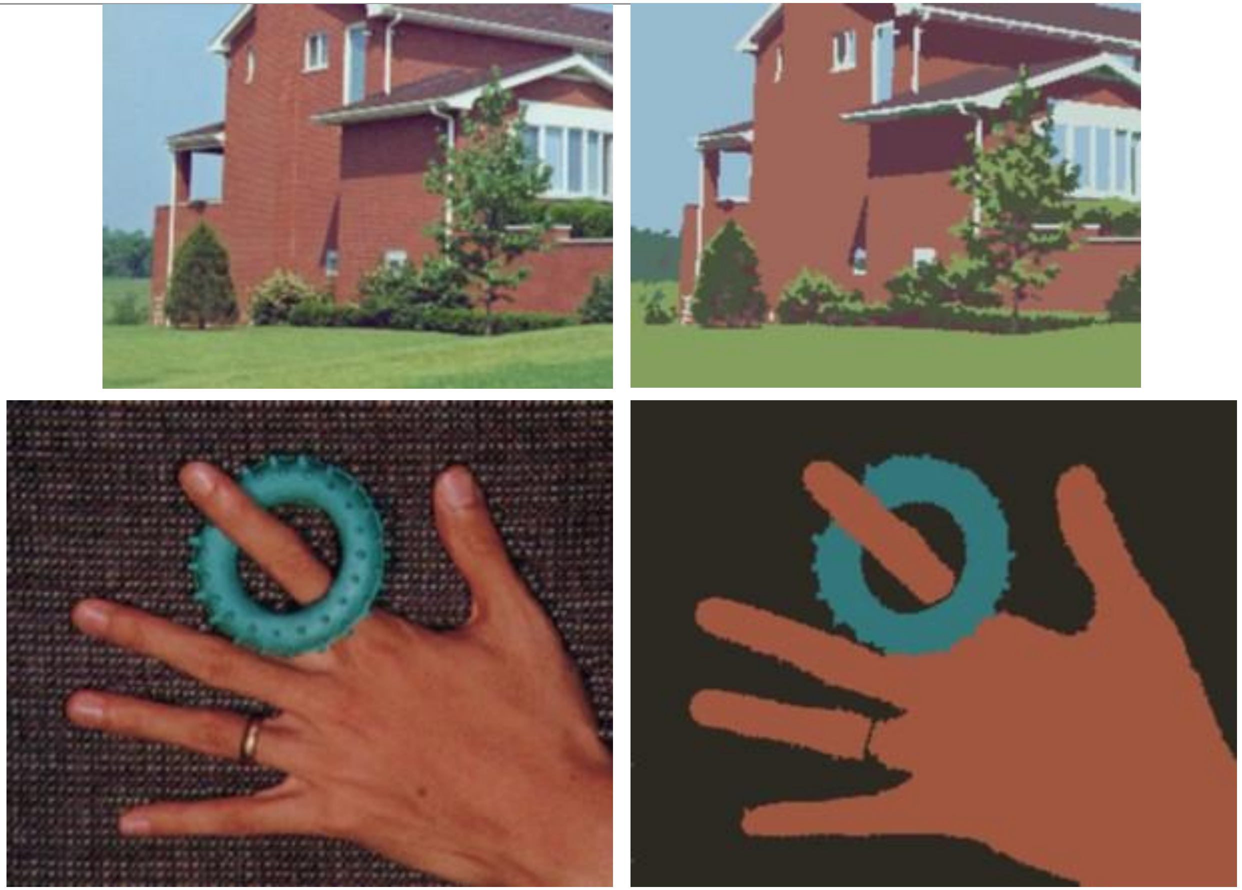
Which Model is the Right One?

K-Means Algorithm: Student Activity

K-Means Algorithm: K =2

Person	Age
1	23
2	32
3	44
4	15
5	25
6	18
7	28

Example



D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.

Example

Image

Clusters on intensity

Clusters on color

K-means clustering using intensity alone and color alone

Example

Original

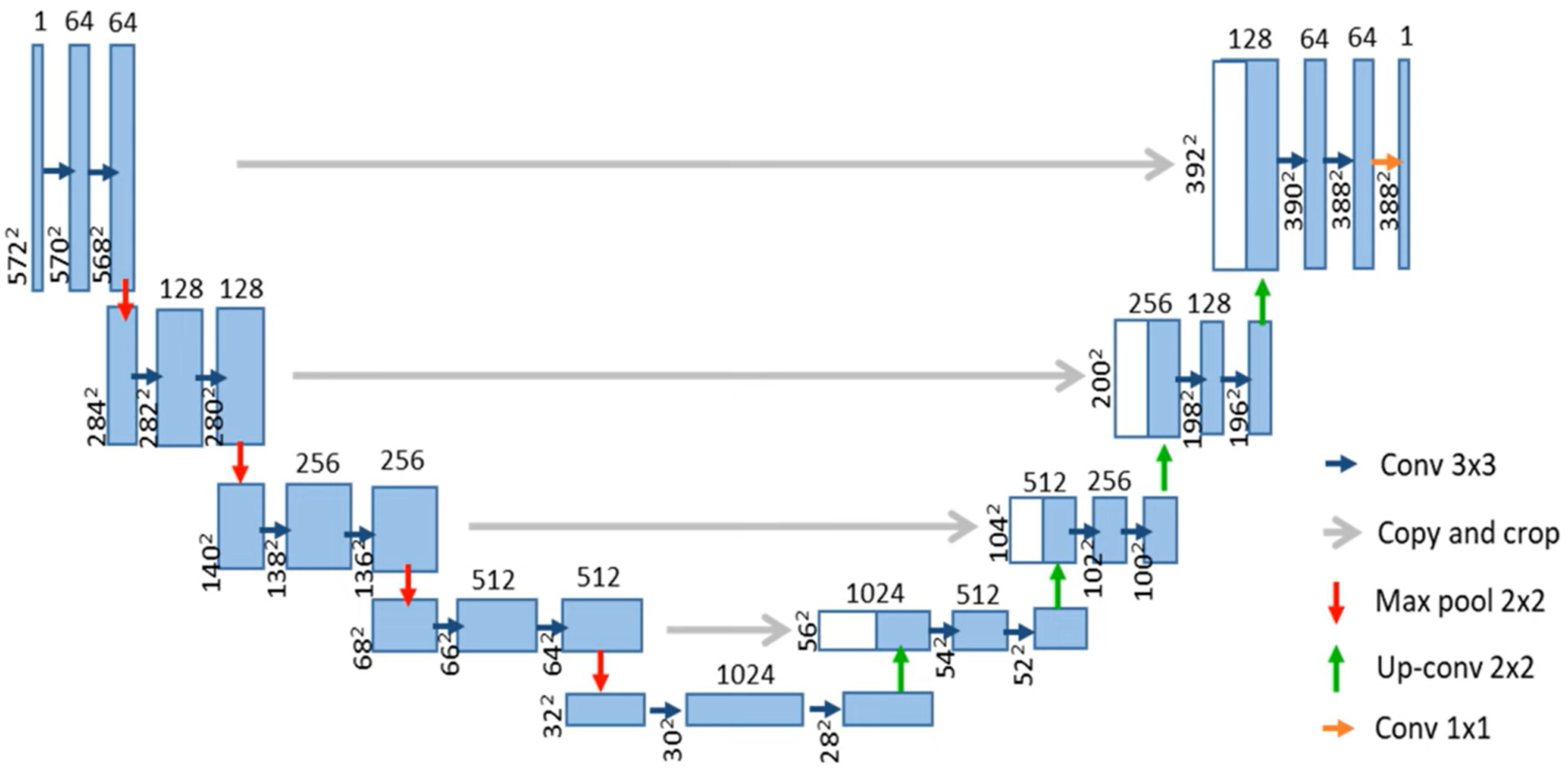
K=5

K=11

How to find a good “K”?

- Use prior knowledge about image.
- Apply the algorithm for different values of K and test for goodness of clusters.
- Analyze Image Histograms.

UNET : Automatic Image semantic



Thank you