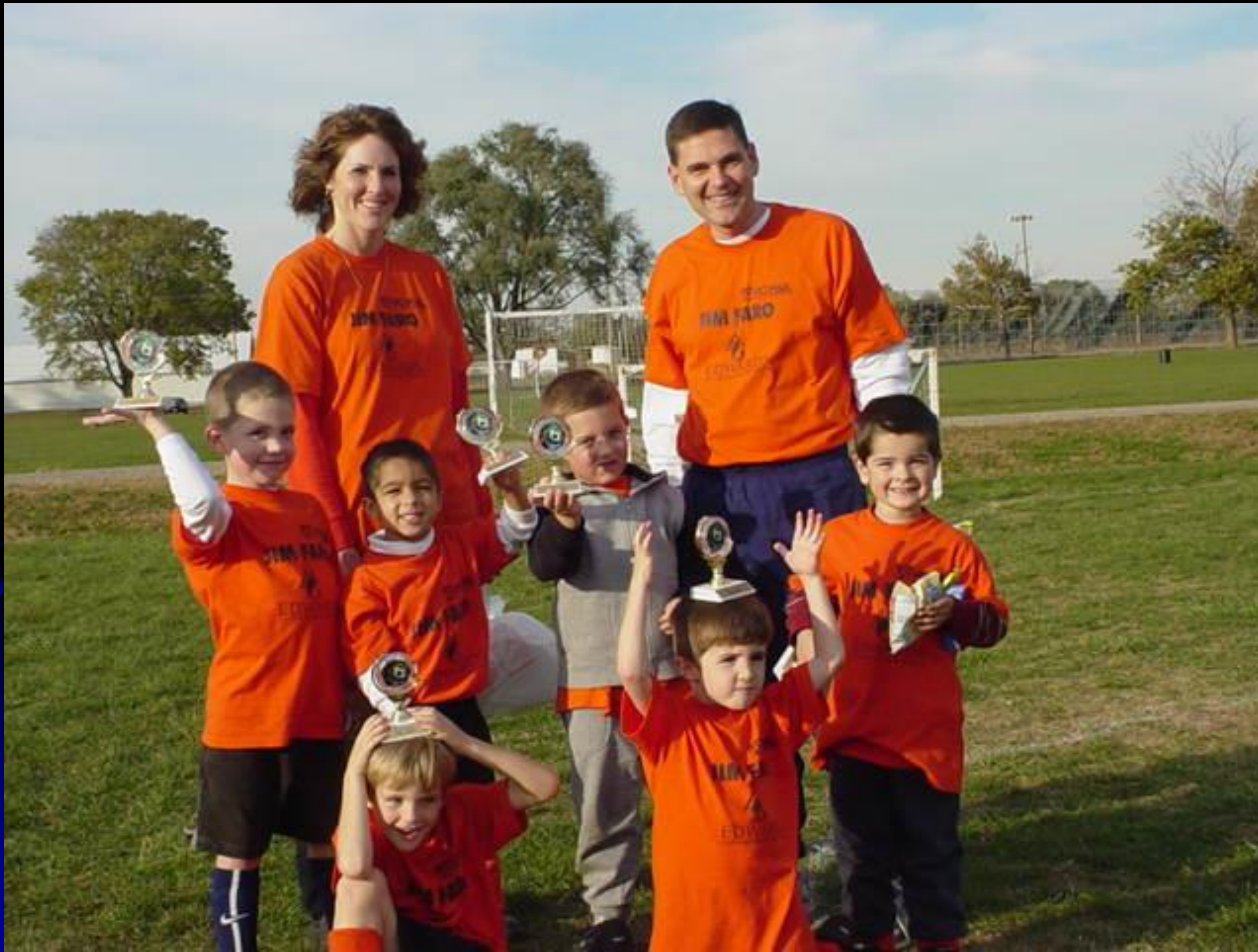
# CSSE463: Image Recognition Day 23

- This week:
  - K-means: a method of image segmentation
  - Lab 6 on k-means tomorrow
  - Sunday night: literature review due
- Project Teams:

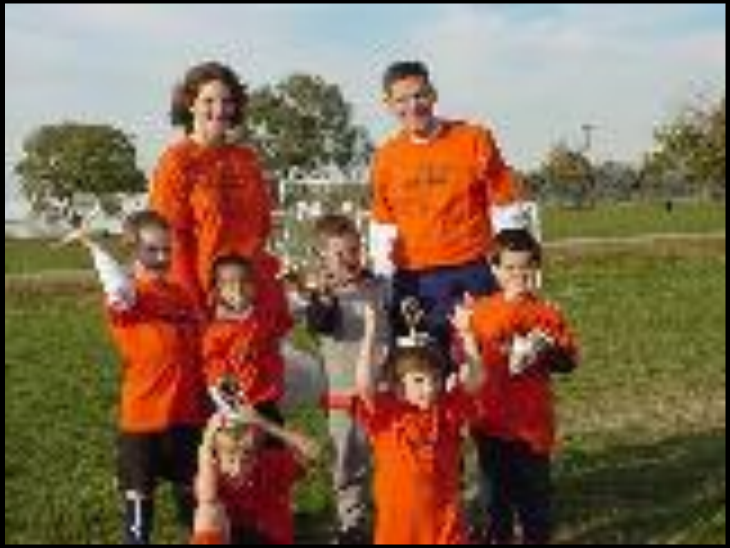
1. All about that Money!: Payden B, Graham F, Jacob O, Sydney S
2. Rhythm Game Detector: Tianyu L, Chris O, Caio, Luan
3. IGVC Obstacle Detection: Allison C, Joe S, Gustavo R
4. Drive smarter/safer or Sudoku solver: John S, Mohammed A, Orry J, Ben P
5. Sheet Music to MIDI: Austin Uphus, Christian Schultz, Man Chi Huen
6. Wordsearch : Garrett Barnes, Eric Yuhas, Zane Geiger

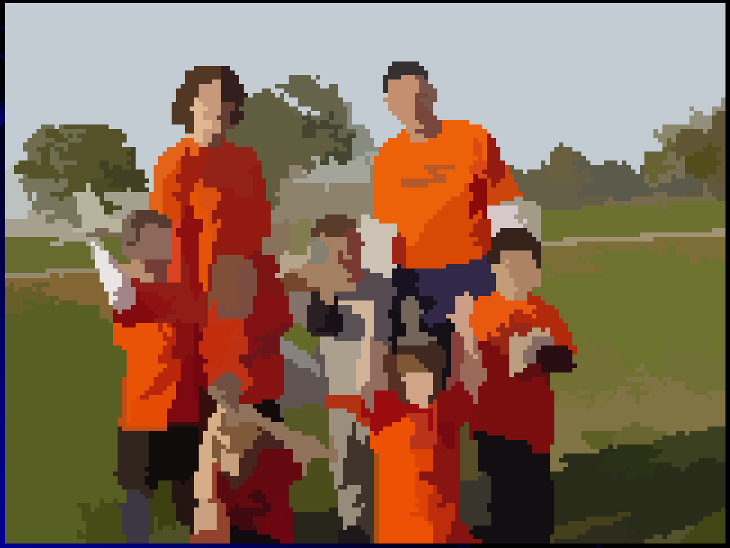
- Literature review: rubric and samples.
- Questions?

An image to segment…



# Segmentation





- The process of breaking an image into regions.
- Two types:
  - General-purpose
    - “One size fits all”
    - Very difficult…
  - Specialized
    - Intended for a specific domain (say fruit-, circle- or skin-finding)
    - Can be successful
- One to right is created using the mean-shift algorithm
  - D. Comaniciu, P. Meer: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell*, **24**, 603-619, 2002.
  - EDISON code downloadable at http://www.caip.rutgers.edu/riul/research/robust.html

# What properties can we use to segment?

- Regions homogeneous wrt. color, texture, etc.
- Adjacent regions different (else merge)
- Smooth boundaries

# Approaches

1. Models
   - Uses an expected shape, color, etc. (fruit- and circle-finders)
   - Can use probabilities
2. Clustering
   - An *unsupervised* machine learning technique
     - No class labels used in learning!
   - Groups pixels “close” to each other by some metric.
     - Color distance, texture, intensity, spatial location, etc.
   - Regions are then found using connected components

# K-means clustering

$$\min_{C} D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - m_k \right\|^2$$

- D= total distance
- K = # of clusters
- x are pixels
- $C_k$ is the set of pixels in cluster *k*
- $m_k$ is the center of cluster *k*
- ||.|| is a distance

- **Goal:** given K clusters, assign each pixel to one of the clusters such that the *total* distance from each pixel to the center of its cluster is minimized.

- We control C, the assignment of pixels to each cluster. (We will actually do this by specifying the location of their means)

Q2

# K-means clustering

$$\min_{C} D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - m_k \right\|^2$$

- K = # of clusters
- x are pixels
- $C_k$ is the set of pixels in cluster *k*
- $m_k$ is the center of cluster *k*
- ||.|| is a distance

**Problems:**

- What's K?
- How do we know which pixel belongs to each cluster?

- K-means is an answer to the second question.

# K-means clustering

- Iterative process to group into *k* clusters.
- Algorithm (Sonka, p 403; Forsyth&Ponce, p. 315; Shapiro, p. 282)
- Initialize K cluster means
- Repeat until convergence:
  - For each pixel, find the closest mean and assign it to that cluster
  - Re-compute the mean of all pixels assigned to the cluster
- Label each pixel with its current cluster
- Example on board using 2D spatial distance

# K-means clustering

- We are trying to find out where the clusters are and which points are assigned to each cluster. We iteratively solve half the problem. Notice the overall structure:

- Repeat until convergence:
  - **Assume you know where the cluster centers are.** For each pixel, find the closest mean and assign it to that cluster
  - **Assume you know which points belong to each cluster.** Recompute the mean of all pixels assigned to the cluster
- Label each pixel with its current cluster

# K-means clustering

- Pros:
  - Easy to implement
  - Finds local optimum (best we can hope for)
- Cons:
  - The number of clusters, *K*, must be known in advance
  - Some clusters might have 0 points
  - Local optimum is not guaranteed to be global optimum
- Ideas:
  - Can re-run with several initializations
  - Can choose *K* based on observation or statistical means
  - *Adaptive* k-means:
    - split a cluster if the total distance to that cluster is too large. Do if you lose a mean along the way
    - Can try to merge adjacent clusters

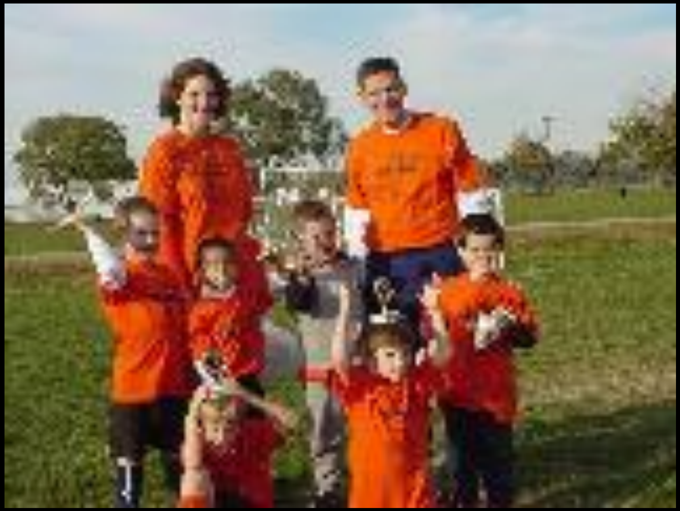
# K-means clustering

$$\min_{C} D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - m_k \right\|^2$$

- K = # of clusters
- x are pixels
- $C_k$ is the set of pixels in cluster *k*
- $m_k$ is the center of cluster *k*
- ||.|| is a distance: could be 2D distance in image or 3D Euclidean distance between colors (or combination of both)

(On Lab: will produce disconnected regions)

# K-means results



Original (120x160)

K=3

K=5

K=7