CSSE463: Image Recognition Day 3

- Announcements/reminders:
  - Lab 1 should have been turned in last night.
  - Tomorrow: Lab 2: on color images. **Bring laptop**.

- Last class?
- Today:
  - Introduce **Fruit Finder**, due **next Friday**.
  - Lots of Helpful hints in Matlab.
  - Connected components and morphology

- Next week: Edge features

- Questions?

Project 1: Counting Fruit

- How many apples? Bananas? Oranges?

Goals

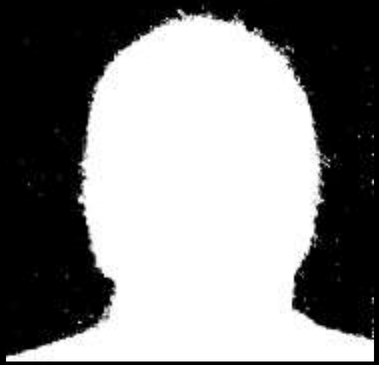
- Crash-course in using and applying Matlab
  - For this reason, I will direct you to some useful functions, but will not give details of all of them
- Practice feature extraction
- Practice writing a conference-paper style report
  - Formal and professional!
  - Could use style similar to ICME sunset paper

Fruit-finding technique

- Observe
  - What *is* a banana's "yellow" (numerically)?
- Model
  - Can you differentiate between yellow and orange? Orange and red? (Decisions)
  - Note: this isn't using a classifier yet; just our best guess at hand-tuned *boundaries*
- Classify pixels using your model
- "Clean up" the results
  - Discuss today
- Write up your results in a professional report (as you go)

Region processing

- Binary image analysis
  - Today, we'll only consider binary images composed of *foreground* and *background* regions
    - Example: apple and non-apple
    - Use **`find`** to create a mask of which pixels belong to each



Q1

Matlab How-to

- Lots of “Random” tidbits that I used in my solution:
  - zeros
  - size
  - find

Functions in Matlab

Contents of foo.m:

```
function retVal = dumbSum(x,y)
   …
   retVal = x+y;
```

Note that you don’t use ***return*** here.

Can return multiple values of any type:

```
[mask, count, threshold] = foo(img)
```

Neighborhoods

- Do we consider diagonals or not?

- 4-neighborhood of pixel p:
  - Consists of pixels in the 4 primary compass directions from p.
- 8-neighborhood of pixel p:
  - Adds 4 pixels in the 4 secondary compass directions from p.

Connected Components

- Goal: to label groups of connected pixels.
  - Assign each block of foreground pixels a unique integer
  - 4-connectivity vs. 8-connectivity matters
- Matlab help: search for *connected components*, and use *bwlabel* function
- Demo

- You'll likely devise an algorithm to do this as part of week 3 homework.

Q6

Morphological operations (Sonka, ch 13)

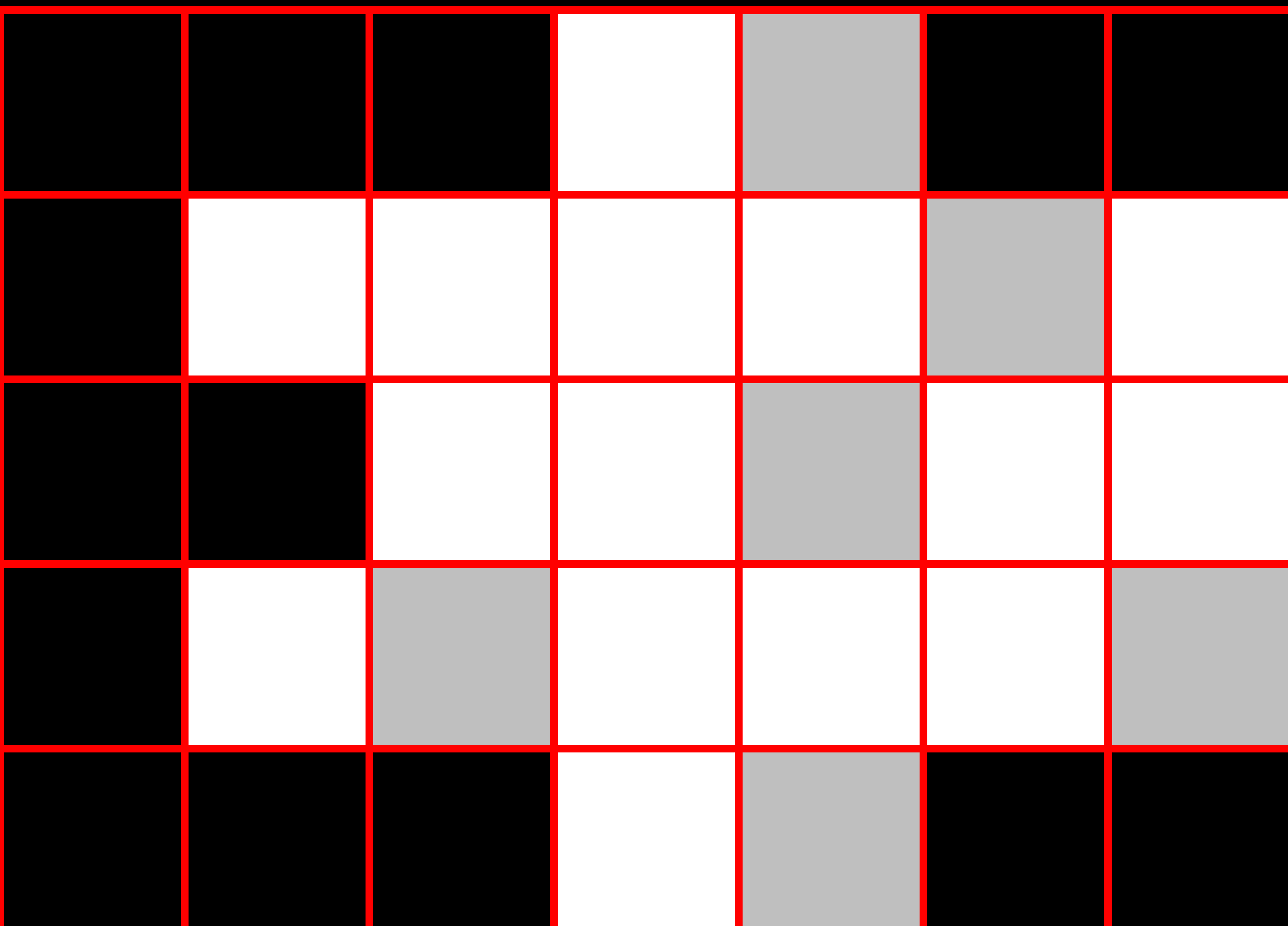
- Morphology = form and structure (shape)
- For binary images
  - Done via a structuring element (usually a rectangle or circle)
- Basic operations:
  - Dilation, erosion, closing, opening

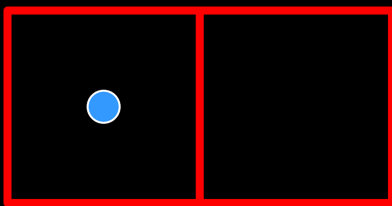
Dilation

- Given a structuring element, adds points in the union of the structuring element and the mask
- Intuition: Adds background pixels adjacent to the boundary of the foreground region to the foreground.
- Def, for image X and structuring element B:

$$X \oplus B = \ p \in \varepsilon^2 : p = x + b, x \in X \text{ and } b \in B$$

Dilation in action





Strel = 2x1,
centered on dot

Dilation

- Matlab: **`imdilate(bw, structureElt)`**
  - Typically want symmetric structuring elements
  - structureElt (for 8 neighborhood) found by:
    - **`structureElt = strel('square', 3);`** % for erosion using 3x3 neighborhood
  - structureElt (for 4 neighborhood) found by:
    - **`structureElt = strel([0 1 0; 1 1 1; 0 1 0]);`**
  - `help strel` lists 11 others
- Demo for intuition: Enlarges a region
- Def:

$$X \oplus B = \ p \in \varepsilon^2 : p = x + b, x \in X \text{ and } b \in B$$

Q7a

Erosion

- Removes all pixels on the boundary
- Matlab: imerode(bw, structureElt)

$$X \Theta B = \ p \in \varepsilon^2 : p = x + b \in X \forall b \in B$$

Q7b

Closing and Opening

- Closing (imclose)
  - Dilate, then erode
  - Fills internal holes in a region, while maintaining approximatel pixel count
  - Eliminates inlets on the boundary
- Opening (imopen)
  - erode, then dilate
  - Removes small regions
  - Eliminates peninsulas on the boundary

- To make dilation more aggressive,
  - Dilate n times, then erode n times.
  - Or, use a larger structuring element
  - Example: compare *dilating twice using a 3x3 square* with dilating *once using a 5x5 square*.