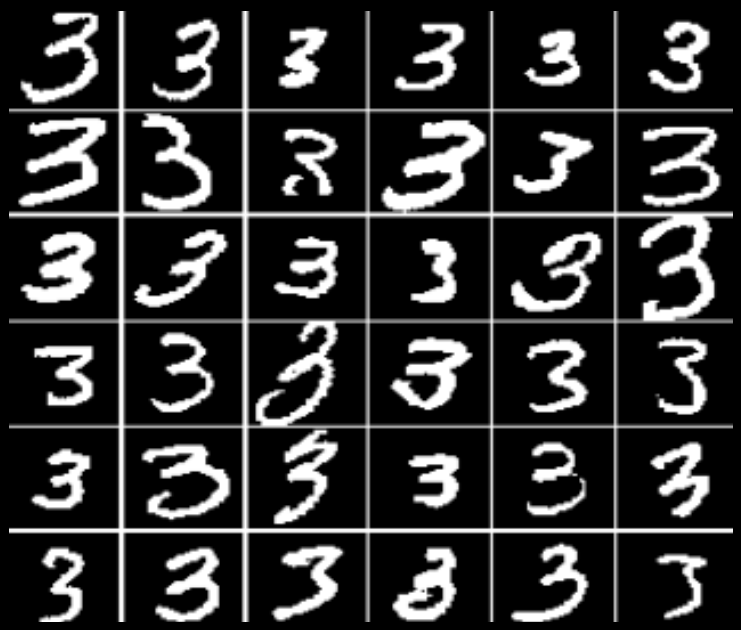
Today

- Supervised learning
- Nearest neighbor methods
  - What are they?
    - Distance functions.
  - When do they work and when do they not work?
    - Test setting is similar to training setting
    - Images are not overly variable.
  - Theoretical results
- Alternatives?

Supervised Learning

- Supervised learning:
  - Formalization of the idea of learning from examples.
- 2 elements:
  - Training data
  - Test data
- Training data:
  - Data in which the *class* has been identified.
    - Example: This is a “three”. 
- Test data:
  - Data which the algorithm is supposed to identify.
  - What is this? 

Recognizing Handwritten Digits



Supervised learning

- Formally:
  - n training data pairs:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$$

x's are "observations"
y's are the class labels

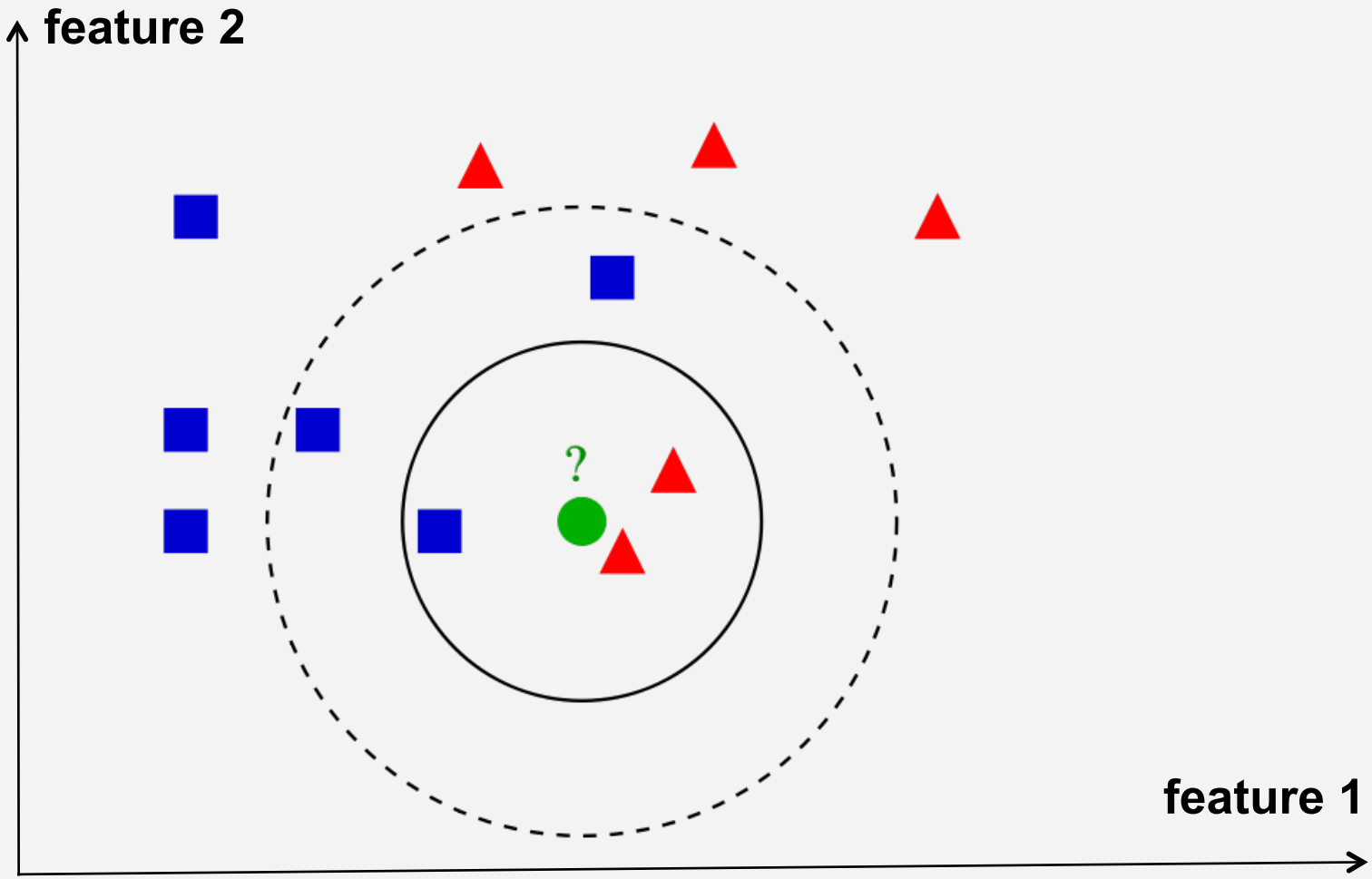
  - m test data samples:

$$(\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \ldots, \mathbf{x}_{n+m})$$

Nearest Neighbor Rule

- Choose label of training example closest to the test example.
- K-nearest neighbor rule (K-NN)
  - Choose some value for K, often dependent on the amount of data N.
    - K=sqrt(N) is a common choice.
    - For a two-class problem, K is usually odd. (Why?)
  - Among K nearest neighbors, have a vote for the label.
  - Break ties with a random choice.

Nearest Neighbor and K-NN



http://en.wikipedia.org/wiki/File:KnnClassification.svg

Euclidean distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

In two *dimensions*

or

Two *measurements* per point

Euclidean distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

In n *dimensions*

or

n *measurements* per point

Euclidean distance for images

```
>> I=rand(5,5)

I =

    0.0003    0.5830    0.4411    0.3030    0.5376
    0.0201    0.5671    0.1130    0.8868    0.6623
    0.9907    0.4703    0.1832    0.0563    0.4345
    0.6449    0.9332    0.3108    0.4623    0.4908
    0.9828    0.9349    0.0914    0.6733    0.8732

>> J=rand(5,5)

J =

    0.7968    0.2026    0.7233    0.1956    0.1599
    0.9779    0.3315    0.8022    0.5367    0.5875
    0.0319    0.2626    0.1441    0.5384    0.0207
    0.5556    0.1637    0.0796    0.4609    0.8187
    0.1396    0.0544    0.7685    0.1129    0.3389

>> diff=I(:)-J(:);
>> diffSquare=diff.^2;
>> EuclidDist=sqrt(sum(diffSquare));
>>
```

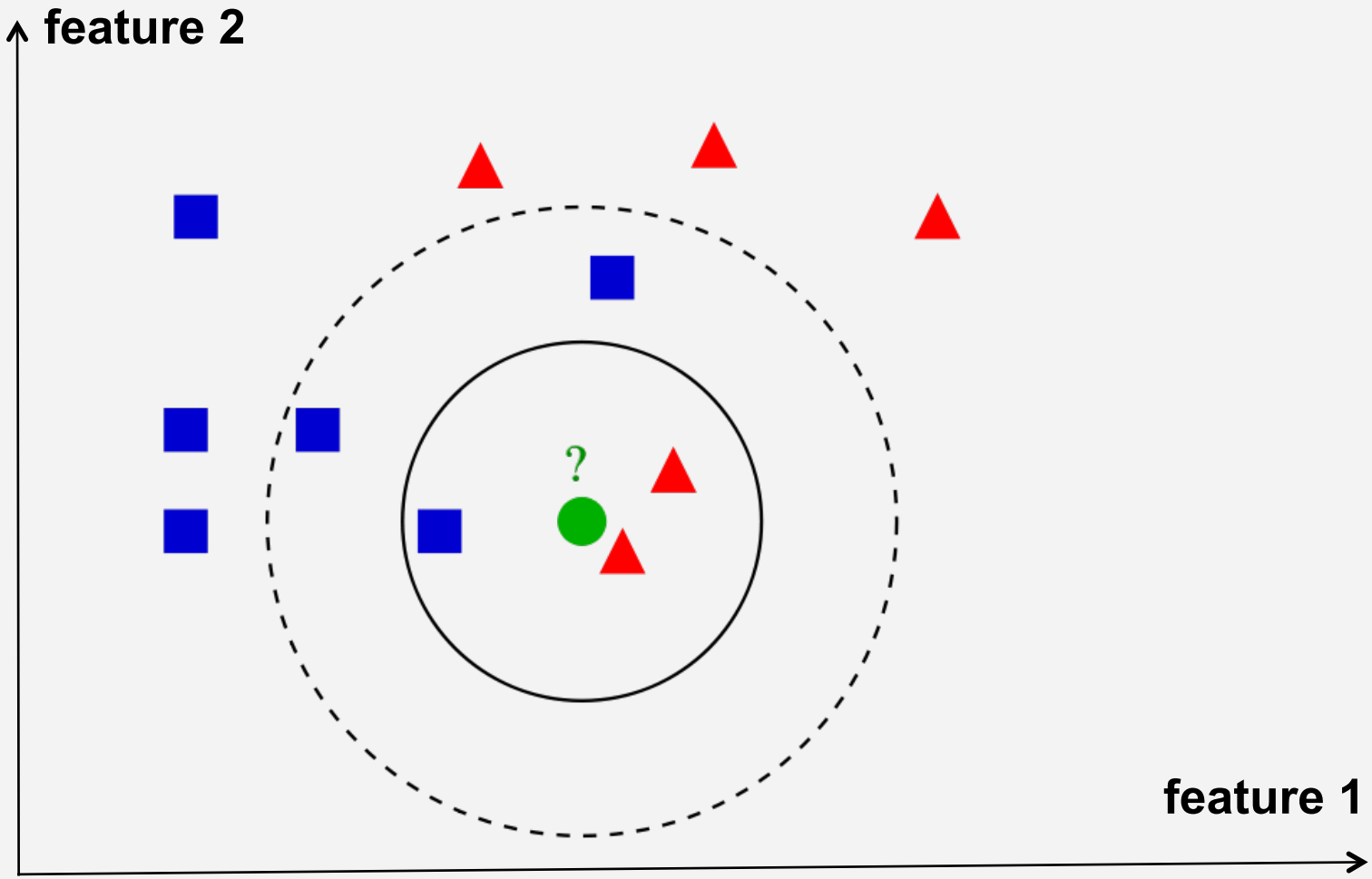
or... in 1 line

```
>> EuclidDist=sqrt(sum((I(:)-J(:)).^2))

EuclidDist =

    2.6947
```

Nearest Neighbor Rule

- Choose label of training example closest to the test example.

Nearest Neighbor and K-NN

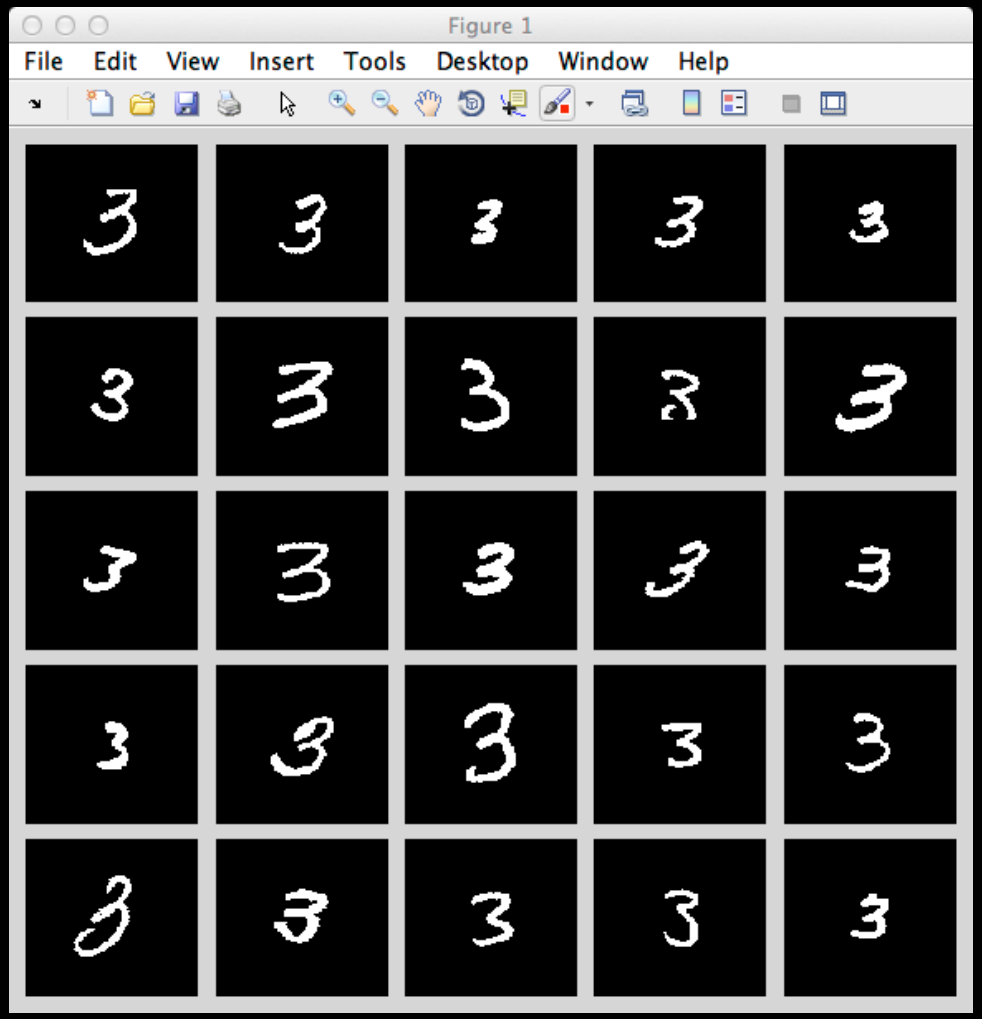


http://en.wikipedia.org/wiki/File:KnnClassification.svg

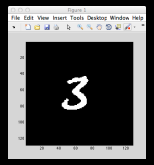
NN for image classification

- Binary images: 128x128.
  - How many measurements?
  - How many dimensions?
- 25 training images of 3's
- 25 training images of 5's
- One test example

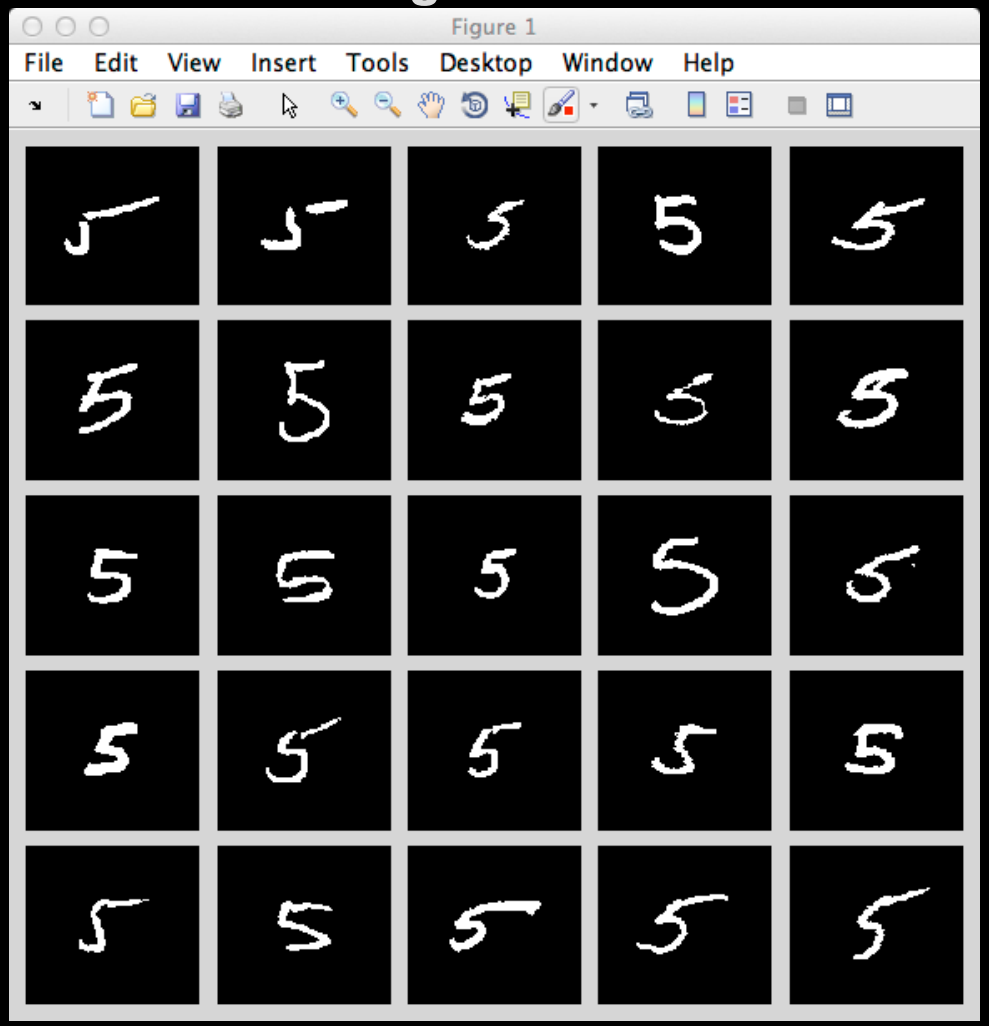
Training Data for 3's



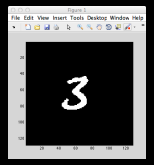
Test Image



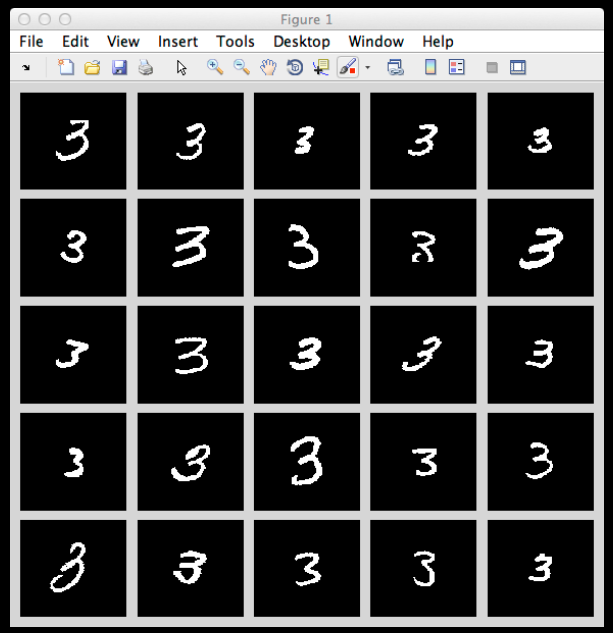
Training Data for 5's

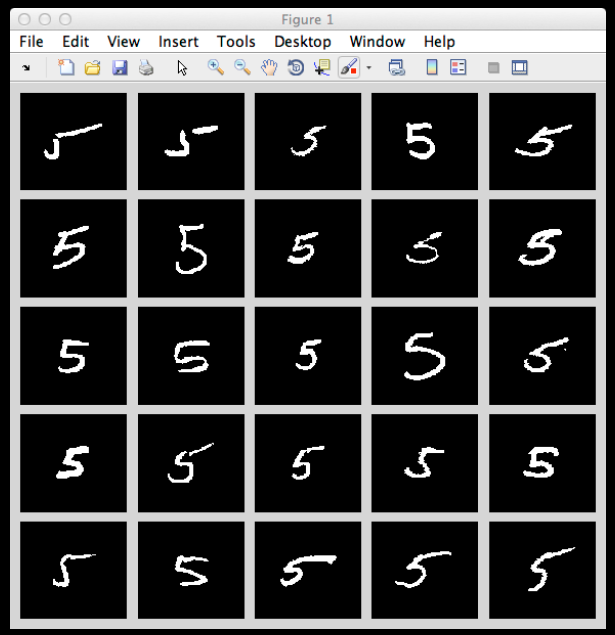


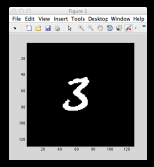
Test Image



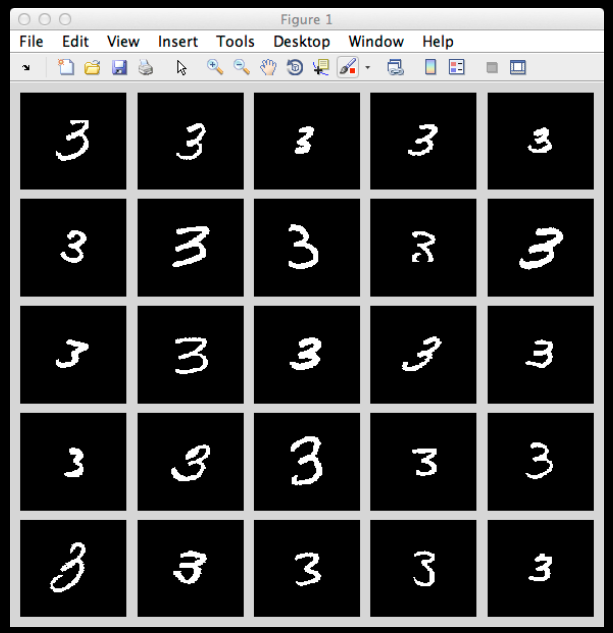
What's the Nearest Neighbor?

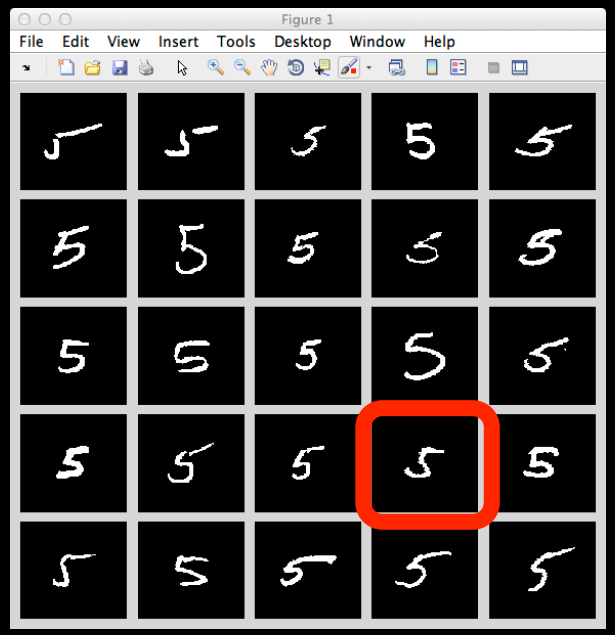


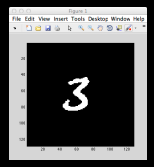




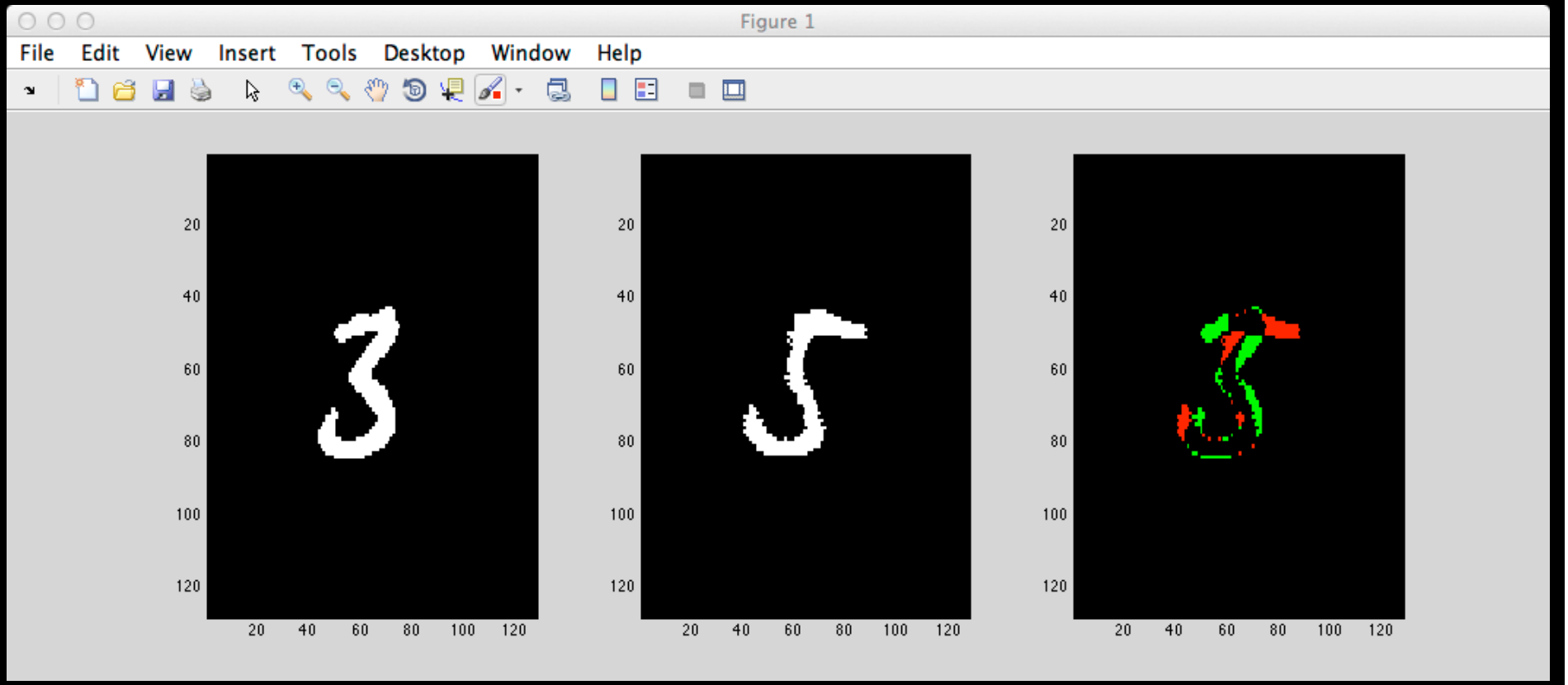
What's the Nearest Neighbor?







Visualizing the image difference



Problems with Nearest Neighbor for Images

- Treats every pixel the same
- Is thrown off by changes in position, scale, or orientation of image
- Can’t handle changes in color