UMassAmherst

Features

What are features?

What are features?

- Given an image in some format
  - Gray scale (0-255)
  - Color: RGB (0-255)
  - Binary image (0,1)
- 3 definitions:
  - Any *function* of the image:
    What features will you be using in your classifier?
    - Also called a "feature function".
  - The **result** of any function of the image:
    How many components does a SIFT feature have?
    - Also called a feature value or feature descriptor.
  - A property of an image which satisfies some predicate function:
    Does the image have that feature?

Some categories of features (not mut. exclusive)

- Single pixel features
- Multipixel features
- Edge features
- Linear filters
- Histograms
  - Brightness values, colors, edges, etc.
- Array of histogram features
- Neural network and deep belief net features

Some categories of features continued...

- Principal components
- Moments (mean, variance, skew, kurtosis):
  - brightness, color, etc.
- Locally weighted histograms
- Position or location features

Position and location features

- Positions which satisfy some property
  - Position of an image patch that looks like an eye
  - Position of a patch that looks like the letter “B”.
  - Positions of “corners” in the image.
  - Positions of edges in the image.
  - “Keypoints” (as used for SIFT descriptors)

Features are the result of a piece of code

- If you can’t write a program to compute it, then you can’t use it as a feature:
  - An “eye” is not a feature we can use in computer vision.
  - The output of a classifier that tries to say whether a patch of an image is an eye or not is a feature we can use. However, it is often misleading (because it is wrong).

Single pixel features

Single pixel features

- The brightness of a single pixel
- The bin of a single pixel
- The percentile of a single pixel
- The z-score of a single pixel (standard deviations from the mean)
- The cluster id of a single pixel.
- The mean of the nearest cluster.

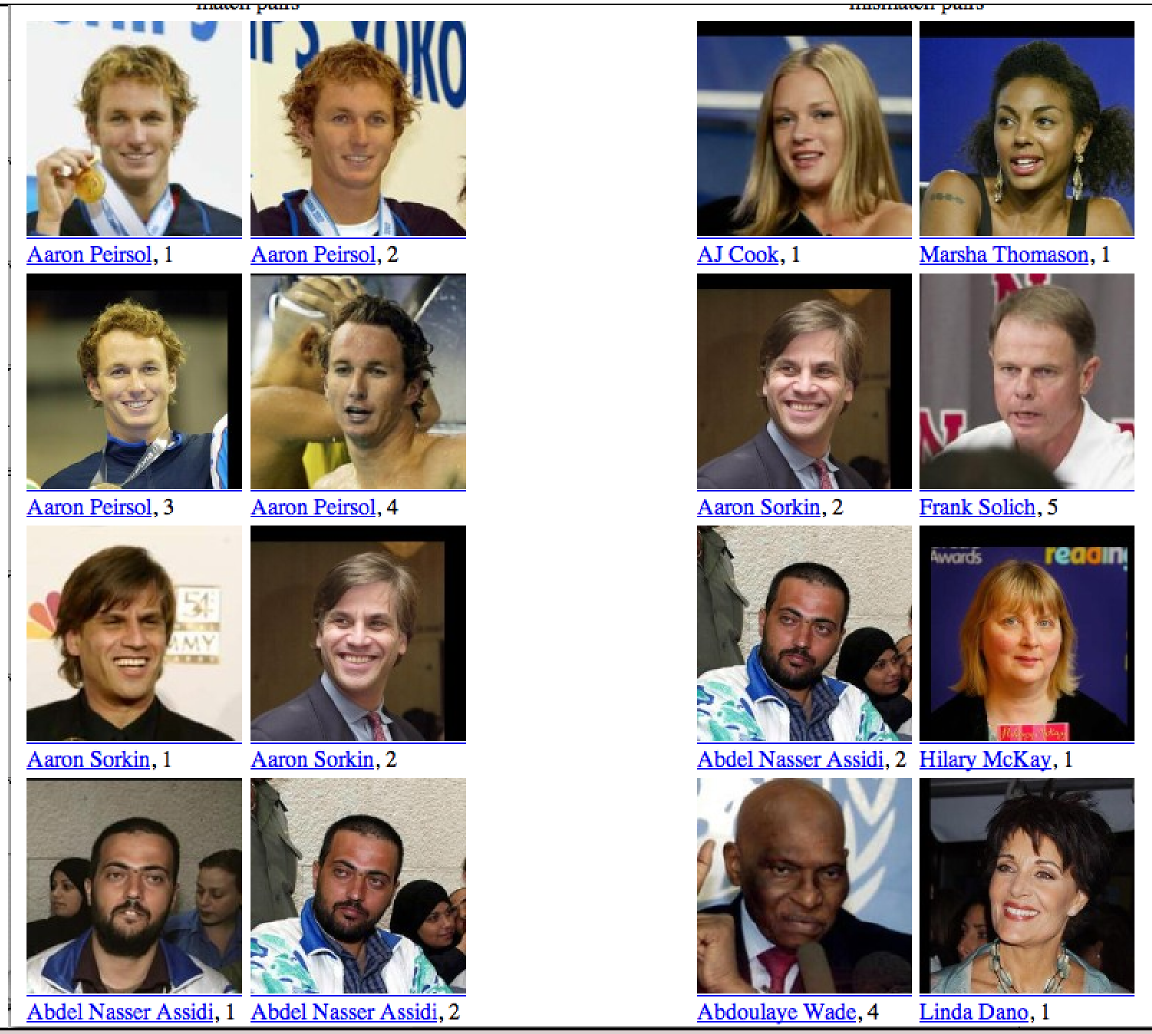
Wait a minute…Why are we using features?

Wait a minute…Why are we using features?

- In supervised learning, 2 basic choices:
  - Work directly on full images
    - Can’t estimate distributions of full images.
  - Work on features of the image
    - We *can* estimate distribution of features, but they may not be as informative.

What features to use?

“same”



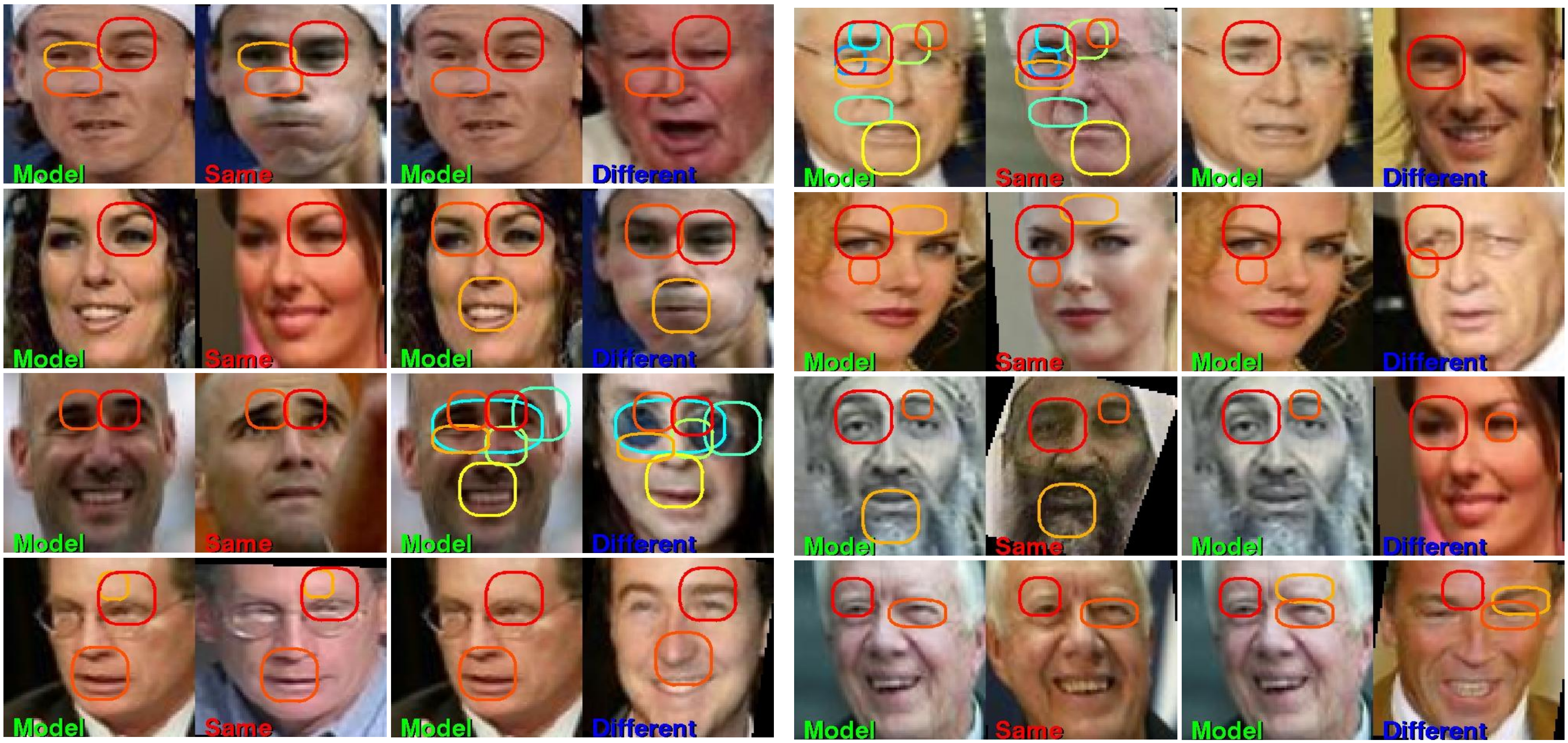
“different”

Properties of good features

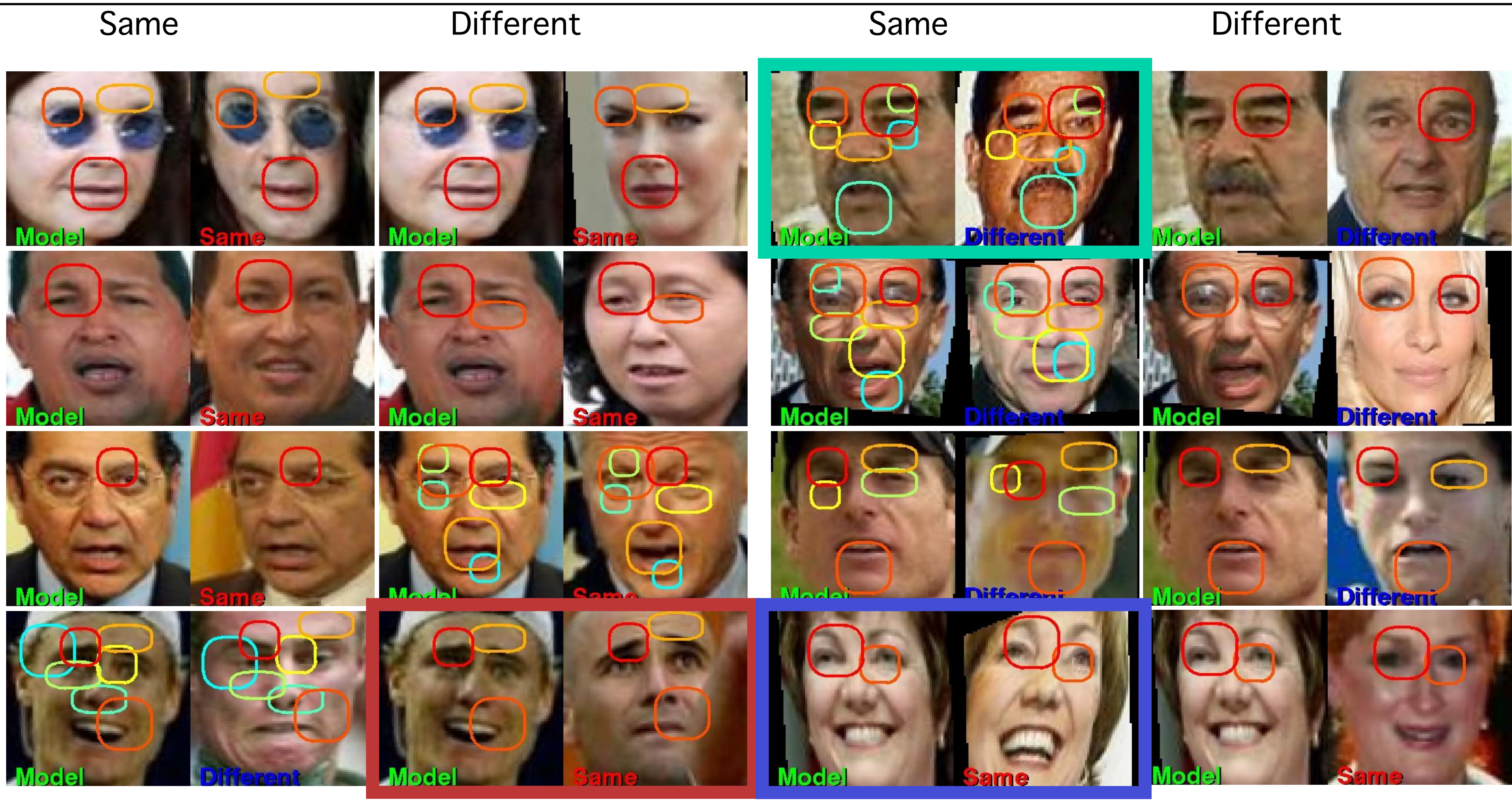
- Depends upon the problem we're trying to solve
- What makes a good feature (in the sense of a property that we are trying to detect)
  - Repeatable—occurs in both examples of an object.
  - Discriminative—occurs in one class but not the other class.

Classification Results (Correct)

Same Different Same Different



Classification Results (Errors)

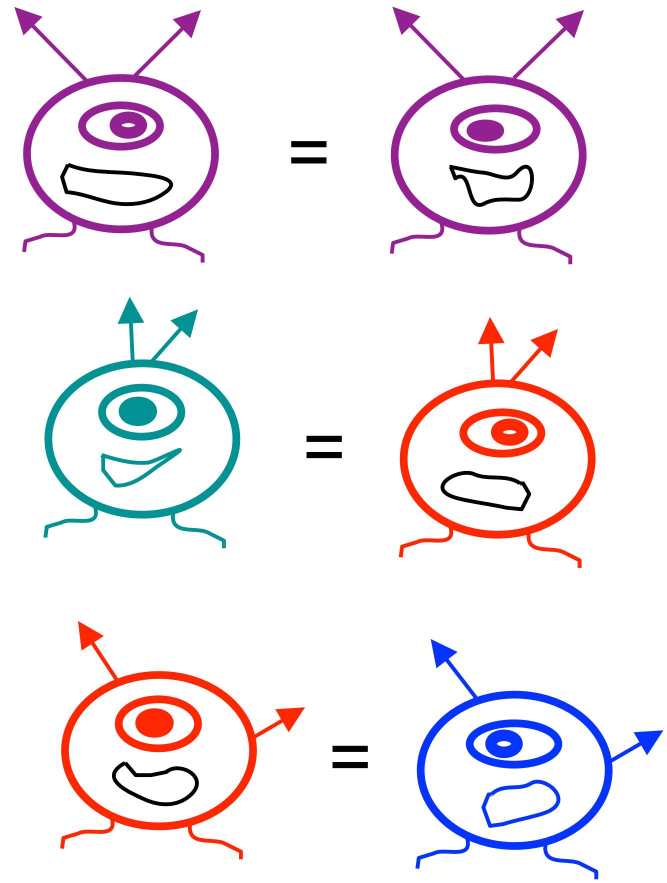


Challenges of Identification (one "model" image)

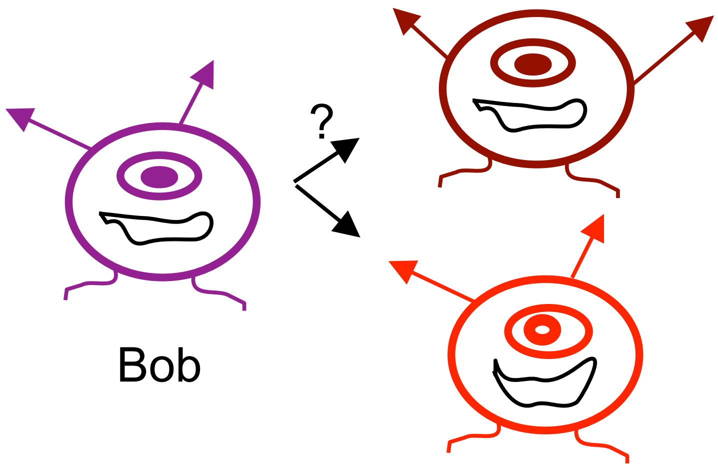



Crash Course on Martian Identification

Martian training set



Test: Find Bob after one meeting



Bob

How are we doing (on faces)?

How are we doing? Continued.

What do we do once we have decided on features?

- Generative models and Bayes rule
- Discriminative methods

Some standard approaches: generative models

- For training set:
  - Convert each training image to a “feature vector”
  - Calculate likelihoods of features for each class
- For test set:
  - Convert test image to feature vector.
  - Use Bayes rule to calculate posterior of class given feature vector.

Some standard approaches: nearest neighbor

- For training set:
  - Convert each training image to a “feature vector”
- For test set:
  - Convert test image to feature vector.
  - Use nearest neighbor, or k-nearest neighbors to pick a class.

Examples

- Handwritten digit recognition from a single pixel: Bayesian approach.
- Scene recognition using color histograms

Why use features?

- Probabilistic approaches:
  - Difficult to estimate distributions based on large numbers of measurements.
    - We don’t have enough data.
    - Will overfit to training data.
    - Won’t generalize to new examples.
- Discriminative approaches
  - Difficult to estimate boundaries in high dimensional spaces
  - Again, overfitting is a problem.

Feature principle #1

- *If we can "throw away" the right parts of the image, we can estimate distribution of the good parts better, and improve generalization performance.*

Another reason for features

- Making the boundary easier to find.
  - Adding squared features.