Computer Vision
and Geometry Lab

Computer Vision

Exercise Session 10 – Image Categorization

Object Categorization

- Task Description
  - “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”
- How to recognize ANY car





Object Categorization

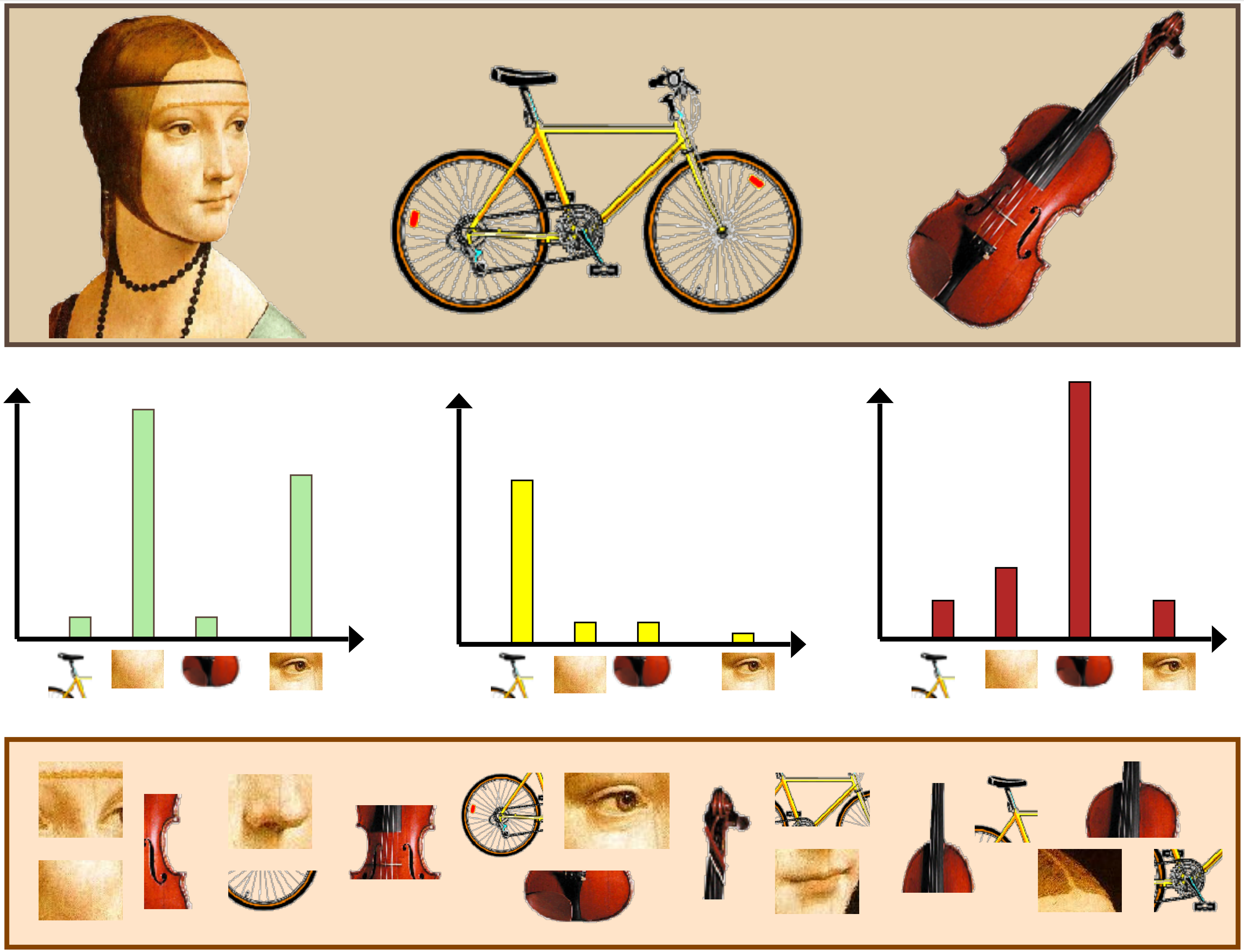
- Two main tasks:
  - Classification
  - Detection

- Classification
  - *Is there* a car in the image?
  - Binary answer is enough
- Detection
  - *Where* is the car?
  - Need localization e.g. a bounding box

Bag of Visual Words

Object → Bag of ‘words’

Bag of Visual Words



BoW for Image Classification

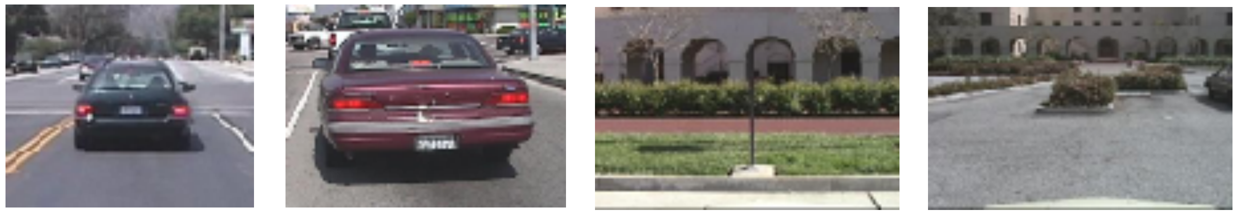
{face, flowers, building}

- Works pretty well for whole-image classification

BoW for Image Classification

1. Codebook construction
2. Training
3. Testing

positive
negative

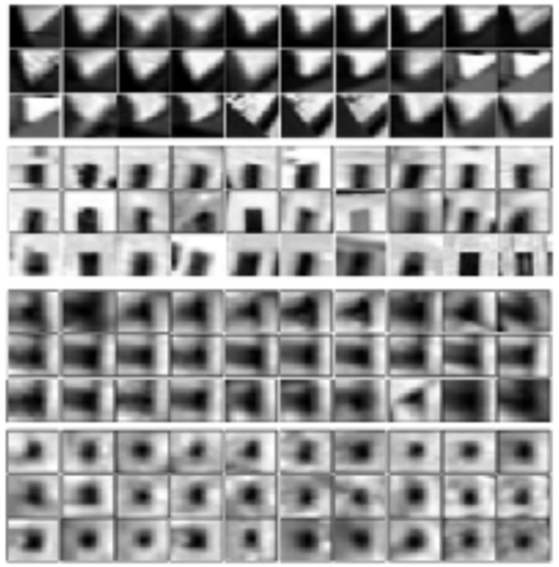


Images

Feature detection and description

Codebook construction

Codebook (visual words)



Bag of words image representation

Train image classifier

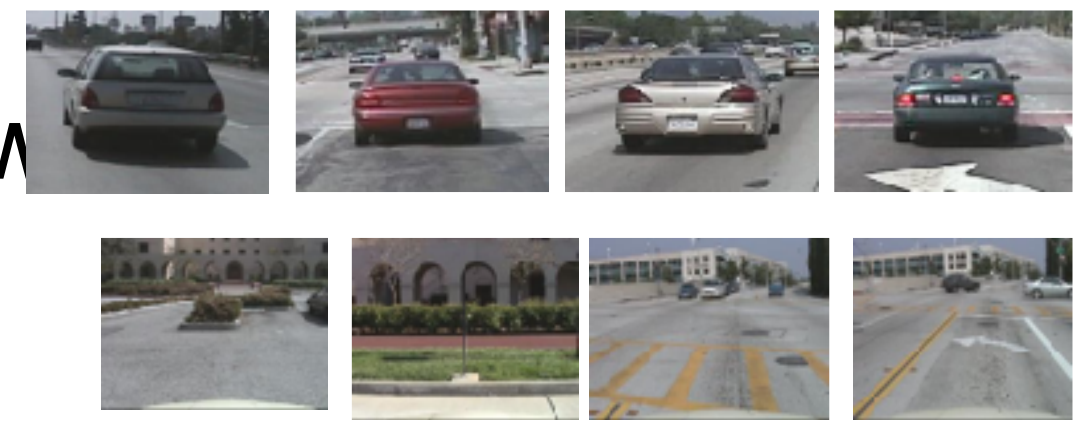
Classifier

Image classification

Binary classification

Dataset

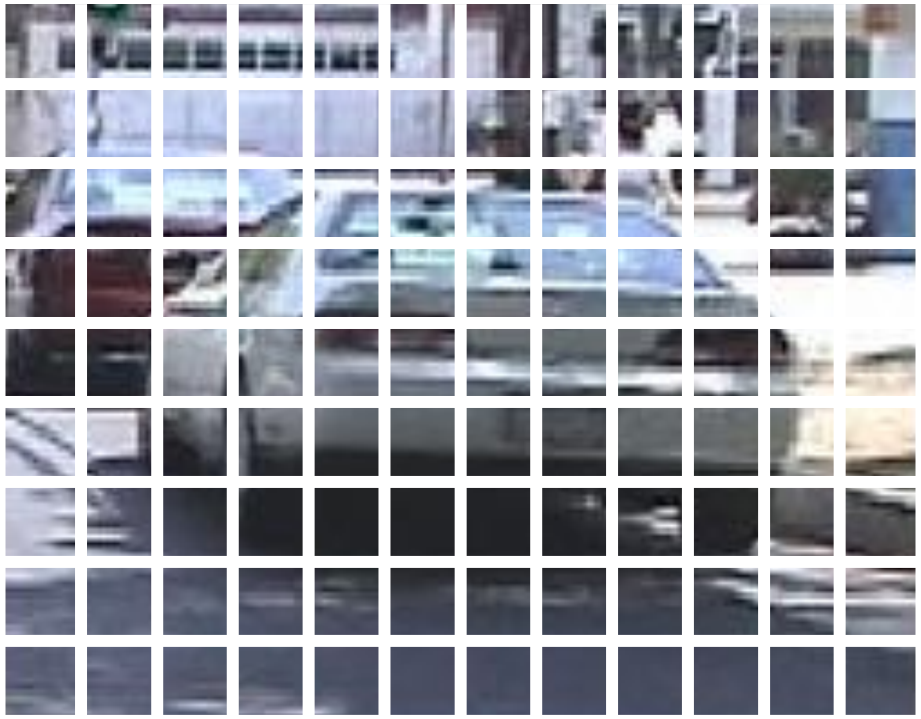
- Training set
  - 50 images CAR - back view
  - 50 images NO CAR



- Testing set
  - 49 images CAR - back view
  - 50 images NO CAR

Feature Extraction

- Feature detection
  - For object classification, dense sampling offers better coverage.
  - Extract interest points on a grid



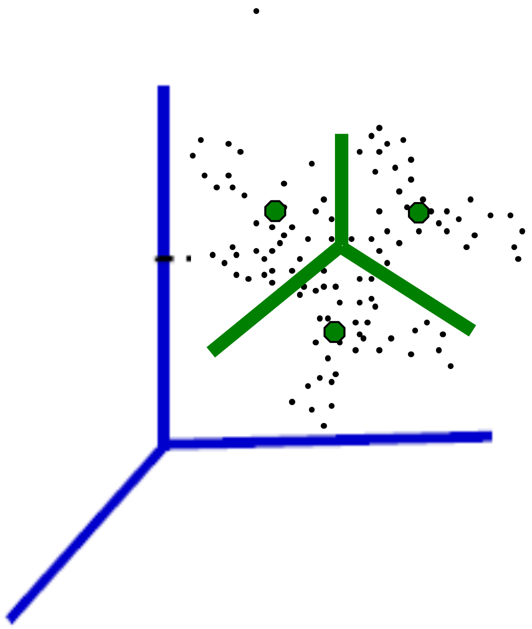
- Feature description
  - Histogram of oriented gradients (HOG) descriptor

Codebook Construction

- Map high-dimensional descriptors to words by quantizing the feature space

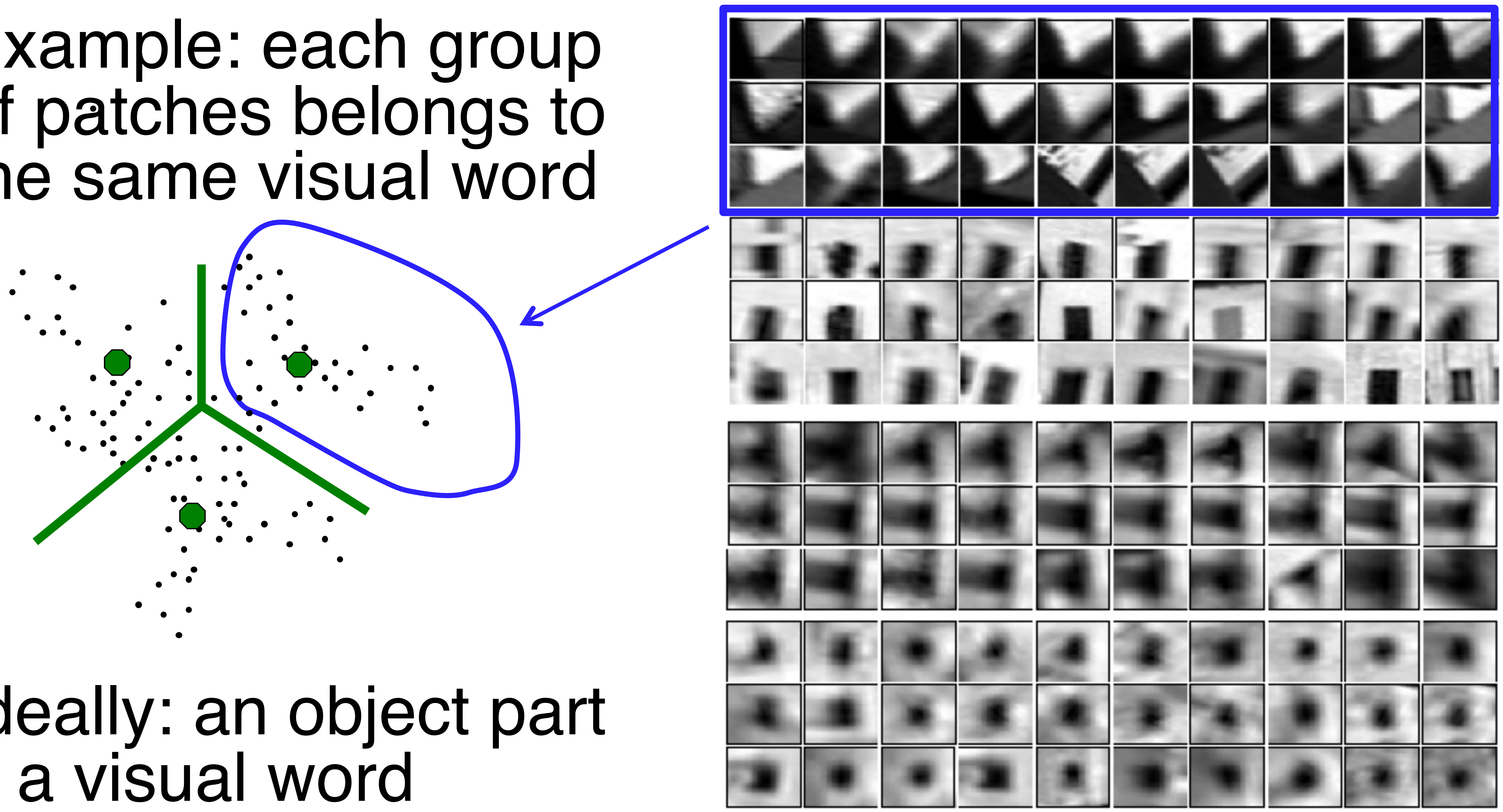
- Quantize via clustering   K-means

- Let cluster centers be the prototype “visual words”



Codebook Construction

- Example: each group of patches belongs to the same visual word



- Ideally: an object part = a visual word

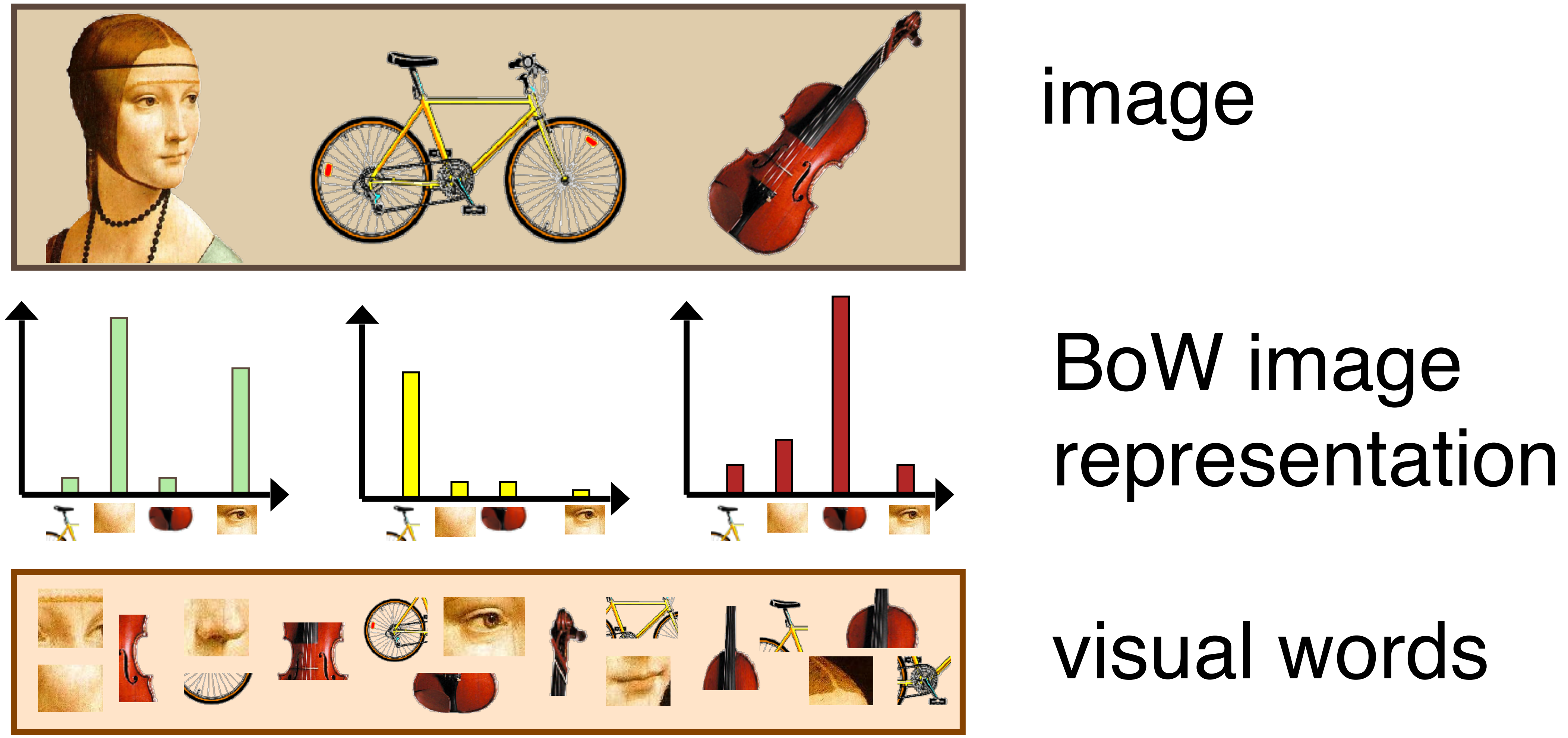
Codebook Construction

- K-means

1. Initialize K clusters centers randomly
2. Repeat for a number of iterations:
   a. Assign each point to the closest cluster center
   b. Update the position of each cluster center to the mean of its assigned points

BoW Image Representation

- Histogram of visual words



BoW Image Classification

- Nearest Neighbor Classification
- Bayesian Classification

Nearest Neighbor Classifier

Training:

- Training images i -> BoW image representation $\mathbf{y}_i$ with binary label c_i

Testing:

- Test image -> BoW image representation $\mathbf{x}$
- Find training image j with $\mathbf{y}_j$ closest to $\mathbf{x}$
- Classifier test image with binary label c_j

Bayesian Classifier

- Probabilistic classification scheme based on Bayes' theorem
- Classify a test image based on the posterior probabilities

Bayesian Classifier

- Test image -> BoW image representation
- Compute the posterior probabilities

$$P(Car|hist) = \frac{P(hist|Car) \cdot P(Car)}{P(hist)}$$

$$P(!Car|hist) = \frac{P(hist|!Car) \cdot P(!Car)}{P(hist)}$$

- Classification rule

$$P(Car|hist) > P(!Car|hist) => Car$$
$$P(Car|hist) <= P(!Car|hist) => !Car$$

Bayesian Classifier

- In this assignment consider equal priors

$$P(Car) = P(!Car) = 0.5$$

- Notice that the posterior probabilities have the same denominator – normalization factor $P(hist)$

- Classification rule

$$P(hist|Car) > P(hist|!Car) => Car$$

$$P(hist|Car) <= P(hist|!Car) => !Car$$

Bayesian Classifier

- **How to compute the likelihoods?**

$$P(hist|Car), P(hist|!Car)$$

- Each BoW image representation is a K-dimensional vector

hist = [2 3 0 0 0 . . . 1 0]

$\uparrow$ Number of counts for the 2nd visual word in the codebook

$\uparrow$ Number of counts for the K-th visual word in the codebook

Bayesian Classifier

- Consider the number of counts for each visual word a random variable with normal distribution

$$counts(i) \rightsquigarrow \mathcal{N}(\mu(i), \sigma(i))$$

Warning: this is a very non-principled approximation as *counts(i)* is discrete and non-negative!

  - For positive training images estimate:

$$\mathcal{N}(\mu_p(i), \sigma_p(i))$$

  - For negative training images estimate:

$$\mathcal{N}(\mu_n(i), \sigma_n(i))$$

Bayesian Classifier

- BoW test image representation= [U_1 U_2 … U_K]
- Probability of observing U_i counts for the ith visual word
  - in a car image $P(U_i|\mathcal{N}(\mu_p(i), \sigma_p(i)))$
  - In a !car image $P(U_i|\mathcal{N}(\mu_n(i), \sigma_n(i)))$

Bayesian Classifier

- Using independence assumption:

$$P(hist|Car) = \prod_{i=1}^{K} P(U_i|\mathcal{N}(\mu_p(i), \sigma_p(i)))$$

$$P(hist|!Car) = \prod_{i=1}^{K} P(U_i|\mathcal{N}(\mu_n(i), \sigma_n(i)))$$

- Numerical stability – use logarithm
- $$log(\prod_{i=1}^{K} p_i) = \sum_{i=1}^{K} log(p_i)$$
- **Now we have the likelihoods**

Hand-in

- Report should include:
  - Your classification performance
    - Nearest neighbor classifier
    - Bayesian classifier
  - Variation of classification performance with K
  - Your description of the method and discussion of your results
- Source code
- Try on your own dataset (for bonus marks!)