1

Introduction to Pattern Recognition by Examples

2

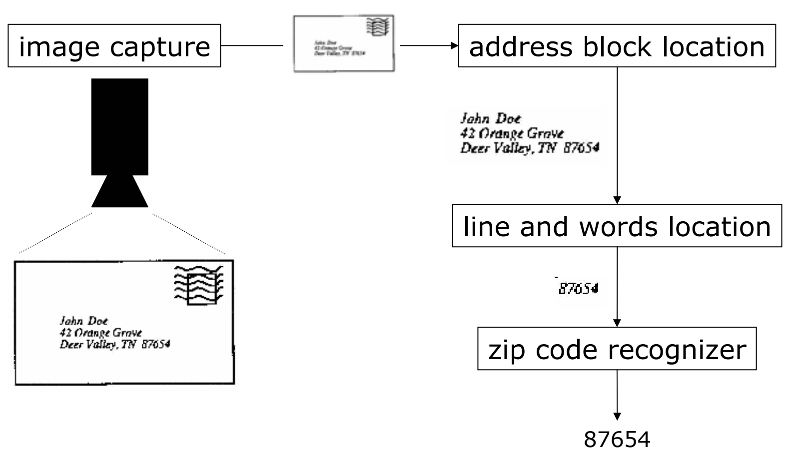
Introduction by Examples

- Optical Character Recognition (OCR),
- Skin Detection based on pixel color,
- Texture classification,
- Speech recognition,
- Email Spam Detection.

Example 1: OCR

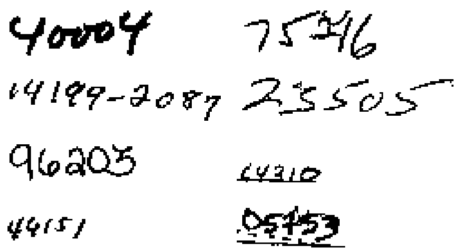
- Example of a problem: recognize *automatically* the *digits* of the zip code on the address *handwritten* on an envelope.
- Application: automatic sorting procedure for envelopes.

Zip Code Recognition

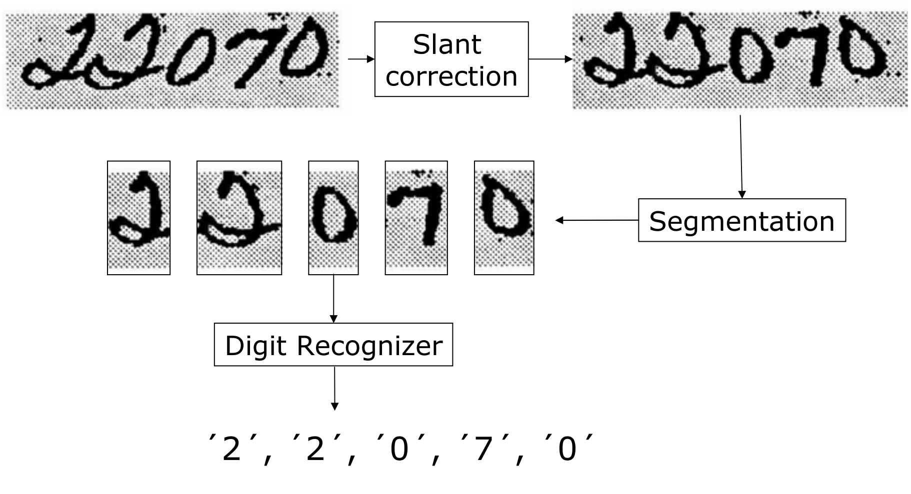


Zip Code Recognition

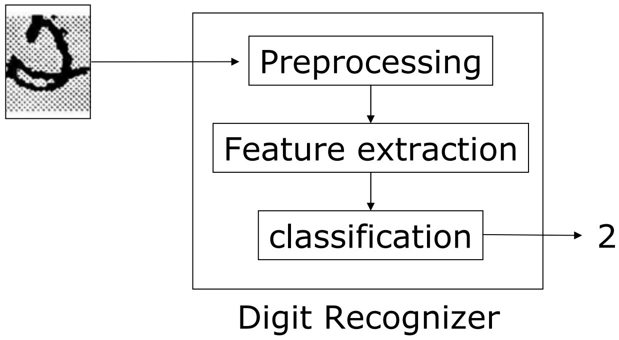
Examples:



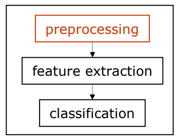
Zip Code Recognizer



Zip Code Recognizer



Zip Code Preprocessing



The preprocessing includes:

1. Thresholding, grey-level normalization, binarizing.
2. Noise correction due to bad image capture: speck removal.
3. Size normalization to a standard size.

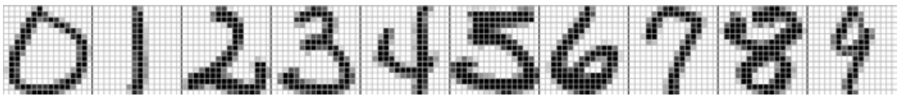
9

Zip Code Data

A standardized set of normalized digit data is available at:
http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/

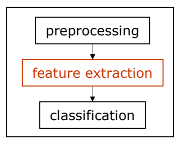
- 7291 digits used for training
- 2007 digits used for testing
- 1 digit = 16x16 grey level value

Example:

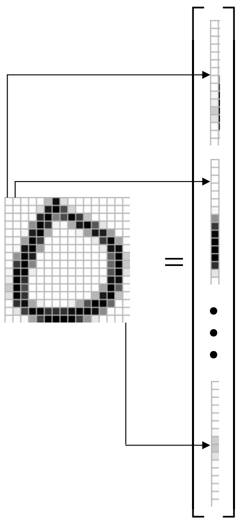


10

Zip Code Feature Extractions



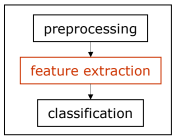
Feature 1: Pixel Intensity



$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 15 \\ 24 \\ 43 \\ 0 \\ \vdots \\ 12 \\ 0 \end{bmatrix}$$

column vector with 256 elements in the range [0, 255]

Zip Code Feature Extractions



Feature 1: Pixel Intensity

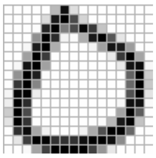
Advantage:

- Easy, raw data after segmentation and size normalization are used.

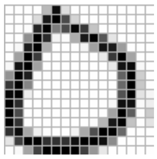
Disadvantage:

- NOT invariant to geometric transformations such as translation, rotation and scaling change.

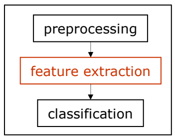
Example:



≠



Zip Code Feature Extractions

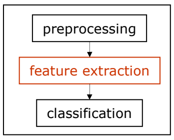


Feature 2: Fourier transformation of the Contour

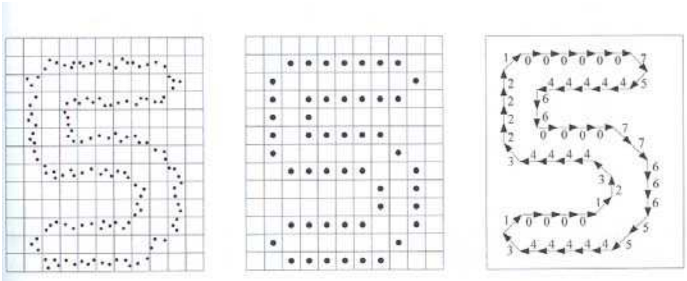
Advantage:

By a clever normalization of the Fourier coefficients, they can be made invariant to translation, rotation and scaling.

Zip Code Feature Extractions



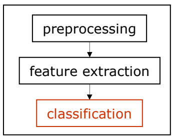
Feature 2: Chain Codes



To make the feature vector invariant to the contour length, a *relative histogram* of the direction is computed and serves as feature vector:

$$f(1) = \frac{\#[1]}{\#[\text{total}]} = \frac{3}{50} = 0.06$$

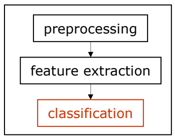
Zip Code Classifier



As the feature vectors are general (no time dependence, many examples per digit), any general classifier can be used:

- Bayes classifier,
- Nonparametric classifier,
- Linear or Nonlinear classifier,
- Support Vector Machine,
- Neural Network, or
- Decision Trees.

Zip Code Classifier



The choice of the classifier will depend on the (assumed) adequacy between the distribution of the training feature vectors in the feature space and the assumption made by the classifier.

Example 2: Skin Detection

A pre-processing step for many applications:

- Face detection & tracking

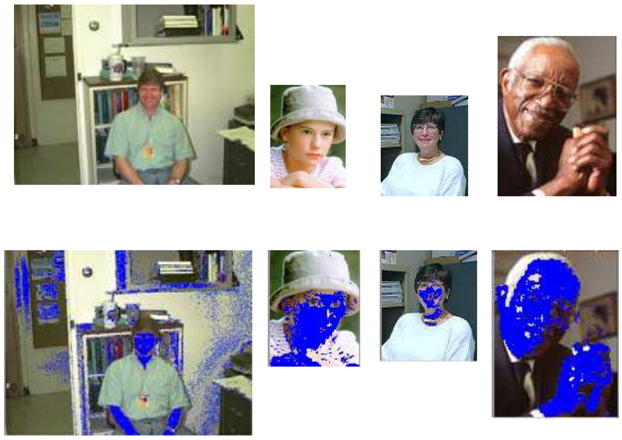
  Images containing people should contain skin → quickly finding skin patches can speed the search for faces, limbs, etc.

- Hand tracking, gesture analysis

- Image content filtering

Skin Detection

17

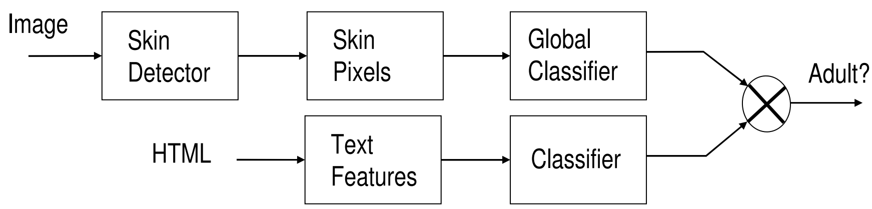


Skin Detection

18

Example: adult image detection:

- Take into account skin area (relative amount of skin pixels) and spatial distribution (connectivity).
- Explore joint image/text analysis.

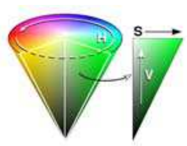


Skin Detection

- Skin color arises due to melanin and hemoglobin; saturation varies with concentration of these components (e.g. lips).
- Hence, it might be possible to represent skin in a color space invariant across humans.
- Problem:
  This representation may not be invariant to illumination.

Skin Detection Features

- Usual RGB-representation is not optimal:
  - High correlation between channels, mixing of chrominance & luminance, etc.
- Normalized RGB: $r=\frac{R}{R+G+B},\ g=\frac{G}{R+G+B},\ \left[b=\frac{B}{R+G+B}\right]$
- Separation of lum./chrom.:
  HSV, YCrCb, YUV



- Perceptually uniform color systems:
  CIELAB, CIELUV

Skin Detection Classifiers

- Classifiers are trained using a set of example pixels with labels (skin or non-skin).

- Bayes Classifiers with:
  - Parametric: single and mixed Gaussians.
  - Non-parametric methods (histograms or parzen-windows) make no assumptions about the shape of the distribution.

Example 3: Texture Classification

Applications:

- Content-based access to image database:

  Which image resembles to: 

  in :  , ..., 

23

Texture Classification

Other application:

- Industrial surface inspection for defects,
- biomedical surface inspection for disease,
- ground classification of satellite or aerial imagery,…..

24

Texture Classification

A texture is an image region with a characteristic distribution of the pixels.

Examples:

As always in a pattern recognition application, the first task is to extract features from the input. Here, the question is:

How to describe these textures mathematically?

Texture Features

To describe a texture, we can use as feature a mathematical measure of:

- fine or coarse ?
- smooth or rough ?
- homogeneous or inhomogeneous ?
- spatial structure, orientation,
- contrast, etc.

Texture Features

A major problem is that textures in the real world are often not uniform, due to changes in orientation, scale or other visual appearance. Hence the features must be invariant to those variations.

Texture feature 1: First Order Statistics

The most well known first order statistics is the *histogram*:

$$P(I) = \frac{\text{number of pixel with color I}}{\text{total number of pixel}}$$

Advantage: simple and easy.

Disadvantage: only pixel intensities are represented, no spatial interaction:

histogram() = histogram()

Texture feature 2: Second Order Stat.

Second order statistics consider pixels in pairs.

Hence, it represents bilateral spatial interactions.

It depends on 2 more parameters:

- d: relative distance between pixels.
- Φ: relative orientation between pixels.

For computational reasons, Φ is discretized, for example in 4 regions arond (0°, 45°, 90°, 135°).

Texture feature 2: Second Order Stat.

Example: second order histogram

$$hh(I_1, I_2, d, \Phi) = \begin{cases} P(I(m,n) = I_1, I(m \pm d, n) = I_2) & \text{for } \Phi = 0° \\ P(I(m,n) = I_1, I(m \pm d, n \mp d) = I_2) & \text{for } \Phi = 45° \\ P(I(m,n) = I_1, I(m, n \mp d) = I_2) & \text{for } \Phi = 90° \\ P(I(m,n) = I_1, I(m \pm d, n \pm d) = I_2) & \text{for } \Phi = 135° \end{cases}$$

Second order statistics may be used to represent smoothness, contrast, coarseness, regularity and other measurement that do not have a physical interpretation but a good discrimination power.

Texture Classification

As the feature vectors are general (no time dependence, many examples per texture), many different classifiers are currently in use:

- Bayes classifier,
- Nonparametric classifier,
- Linear or Nonlinear classifier,
- Support Vector Machine,
- Neural Network and
- Decision Trees.

Example 4: Speech Recognition

31

Applications:

- Search recorded audio and video.

- Machine interaction for people whose hands are busy, for example in hospitals.

- Machine accessible for people who do not know how to type or who can not type: disabled people, telephony applications.

- If combined with Natural Language Understanding, would make computer accessible to any people.

Speech Recognition

32

Why is it difficult ?

- 2 persons uttering the same word will say it differently (inter-speaker variation).

- The same person will say the same word differently on different occasions (intra-speaker variation).

- The waveform of a word depends on recording conditions (noise, reverberation).

Speech Recognition

Speech Recognition is *different* from the previous examples:

- Speech unfolds in *time*.
- Word = succession in time of phonemes.
- Problem: Phoneme at time `t` is influenced by phoneme at time `t-1` and at time `t+1`.
  - Example: `skloo` and `sklee`

    The shape of the mouth when `skl` is pronounced depends on what follows.

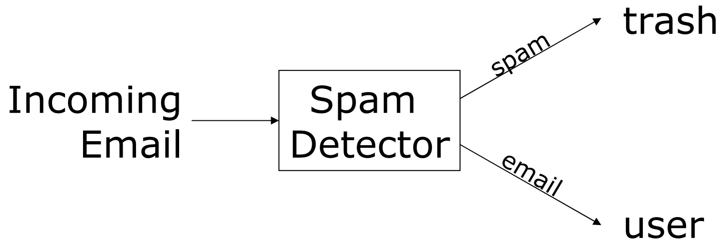
Speech Recognition

Hence, we will have to use a classifier that model this time (neighborhood) dependence.

The most widely used classifier is the Hidden Markov Model (HMM).

Example 5: Email Spam Detection

35



In this application, not all errors are equal.
The cost of misclassification is assymetric:

email misclassified as spam
should happen *less often* than
spam misclassified as email.

Features for Email Spam Detection

36

Features are similar to those used for the texture classification but using words instead of pixels:

First order: word frequencies

Second order: word co-occurences

Classifier:

- Decision tree,
- Bayes classifier,
- etc.

Essence of Pattern Recognition

In Pattern Recognition, there are *many* classifiers.

So far there is no automated optimal classifier selection!

It depends on the *distribution* of the feature vectors in the feature space.

As this distribution is in general *inaccessible*, the practitioner must *assume* one.

The *accuracy* and the *efficiency* of the system will depend on this choice.

The other choice of the practitioner, which is not independent of the first one, is the one of the *features*.

Essence of Pattern Recognition

Good features:

- have a simple distribution, that can be discriminated by simple and efficient classifiers.
- are invariant to transformations of the data present in the training set which are unrelated to the classification problem.

Conclusions

39

We have seen several examples of pattern recognition problems.

The two major questions of the design of a pattern recognition application are:

1. Which *feature* to use ?

2. Which *classifier* to use ?

The *feature* to be used depends heavily on the problem: feature for texture classification are not the same as those for OCR.

Conclusions

40

However, the *classifier* used may be the same: Usually, a feature is a vector, any classifier taking vectorized input may be applied on OCR, texture classification, etc.

The choice of the classifier depends on the distribution of the training feature vectors in the feature space.