Digital Image Processing

Lecture 6
Object Recognition 1

Rahul JAIN

Spring 2022

Outline

- Introduction to pattern (object) recognition
- Pattern recognition pipeline
  - How to recognition unknown patterns
- Two essential components of the pattern recognition
  - Feature extraction
  - Classification
- Simple recognition methods
  - *k*-nearest neighbor algorithm
  - Template matching

Pattern recognition

- Automatic (machine) recognition, description of patterns are useful in a wide range of field such as medical imaging, computer vision, AI and remote sensing.
- Pattern recognition: "The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories." [1]
- Example: The colors on the clothes, speech pattern, human face, handwritten word, fingerprint image, etc.
- The objective of pattern recognition is to automatically assign patterns to their respective classes.

[1] Bishop et al., "Pattern Recognition and Machine Learning (Information Science and Statistics).

Some pattern recognition applications

Problem Domain	Application	Input Pattern	Pattern Class
Bioinformatics	Sequence analysis	DNA / Protein sequence	Know types of genes / patterns
Data mining	Searching for meaningful patterns	Points in multi-dimensional space	Compact and well separated clusters
Document classification	Internet search	Text document	Semantic categories (e.g., business, sports, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g., action, dialogue, etc.)
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition	Telephone directory enquiry without operator assistant	Speech waveform	Spoken words

Pattern recognition

- Pattern Recognition is critical in most human decision-making tasks.
- A primary goal of pattern recognition is to be able to classify data into a set of related elements

Cognition vs. Recognition

- Humans can easily recognize faces, understand spoken words, and read handwritten text, but machines require to be trained using certain rules.
- An example of cognition was learning your friend's face in the first place.
- An example of recognition is seeing your friend's face in a sea of faces. Even though you have not seen this person in ten years, and he has changed a lot, you can you pick him out in a few seconds.

Pattern recognition

- Pattern recognition is the learning procedure for observing (sensing) the environment, learning to distinguish patterns of interest.
- Pattern recognition involves learning: "Learning is the process of estimating an unknown input-output dependency or structure of a system using a limited number of observations." [1]
- Some examples of learning problems:
  - Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack - the prediction is to be based on demographic, diet and clinical measurements for that patient.
  - Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data.

[1] Data Mining: Concepts, Models, Methods, and Algorithms

Pattern recognition

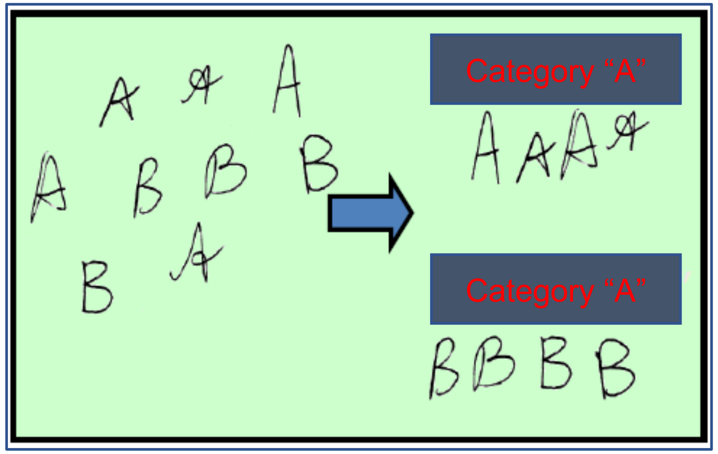
- Supervised learning – Classifying objects with (known) different labels.
  - We have training set, features (prediction or input), and outcome (response or output).
  - We can build a model to predict future data.
- Unsupervised learning – Classes or subclasses have to be derived from the data.
  - We observe only the features and have no outcome.
  - We need to cluster data or organize it.

Pattern recognition

Classification (Recognition)

(Supervised Classification)

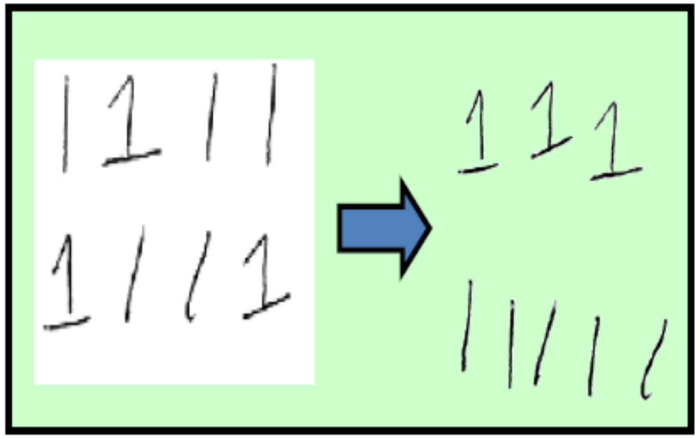
Known categories



Clustering

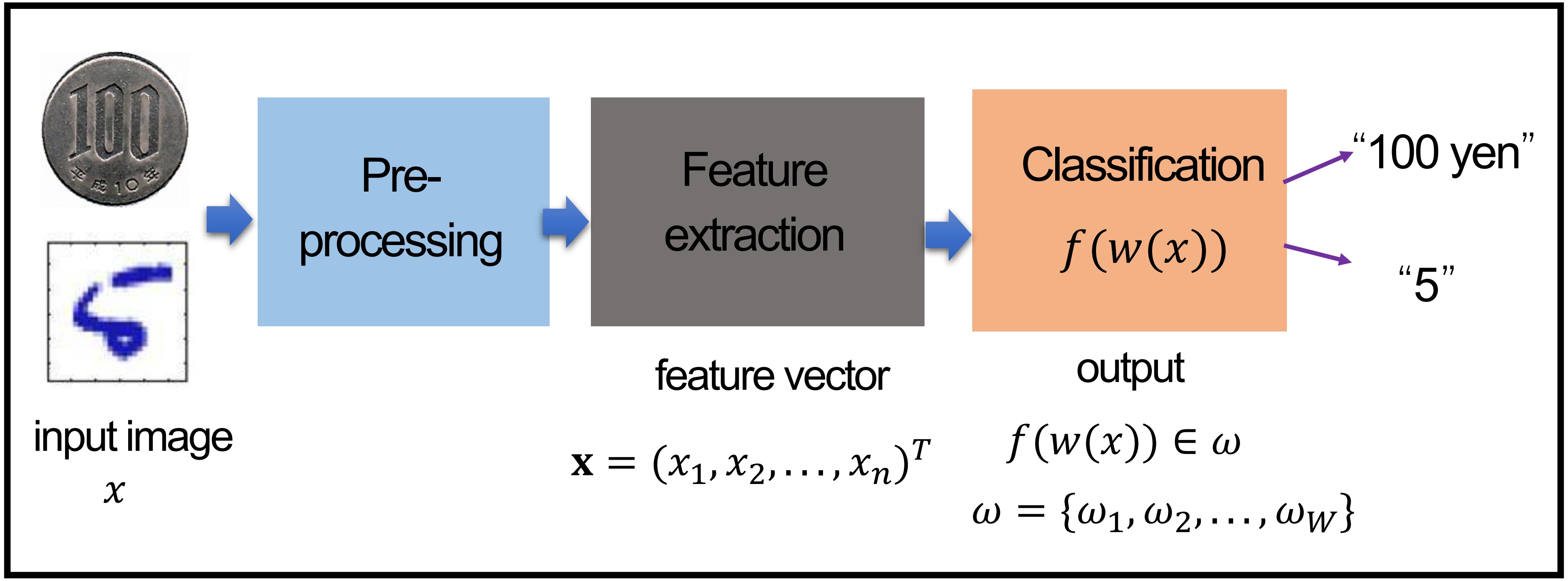
(Unsupervised Classification)

Creation of new categories



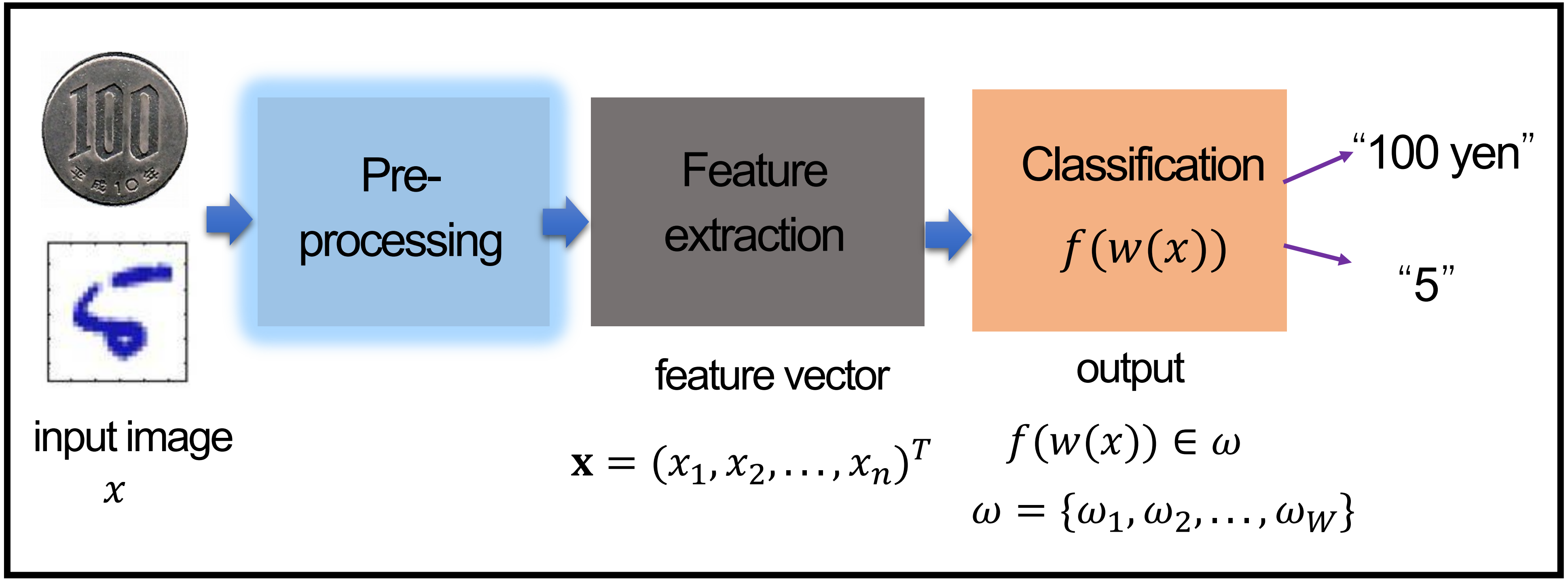
Pattern recognition

- Pattern recognition pipeline



Pattern recognition

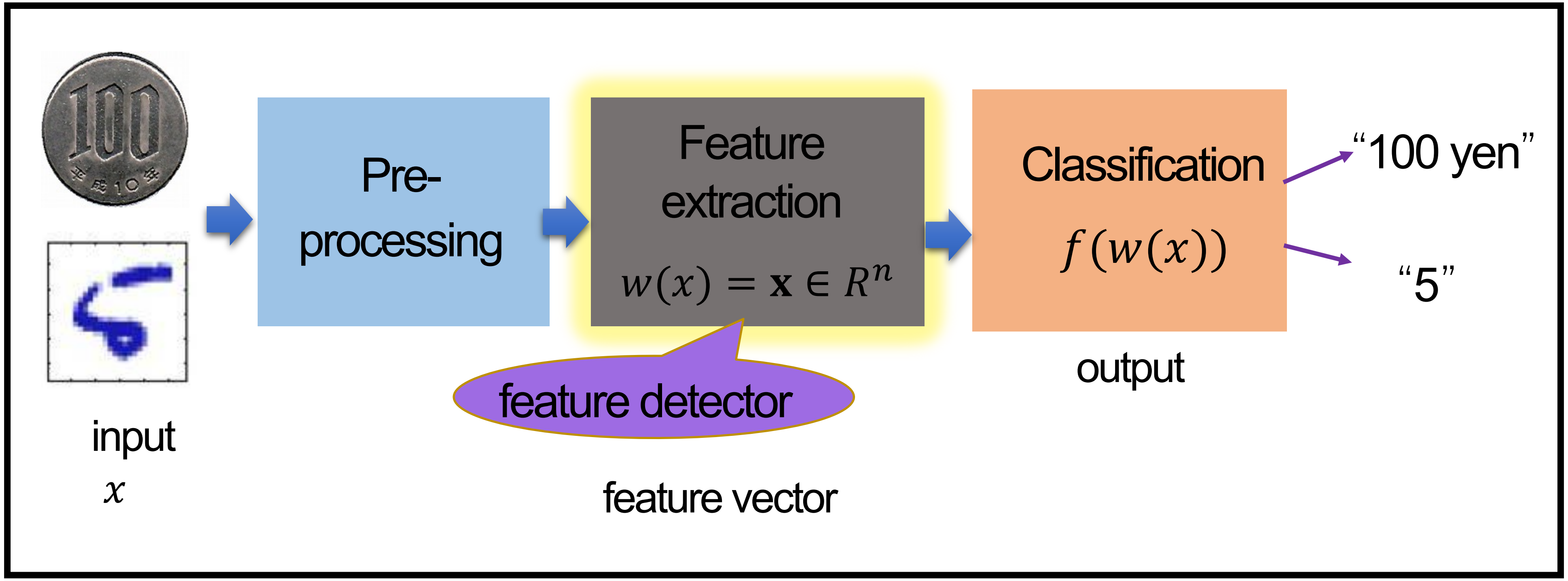
- Pattern recognition pipeline



Preprocessing: Image processing includes some data augmentation and image refinement methods such as smoothing or sharpening to effectively extract feature information and simplify subsequent processing

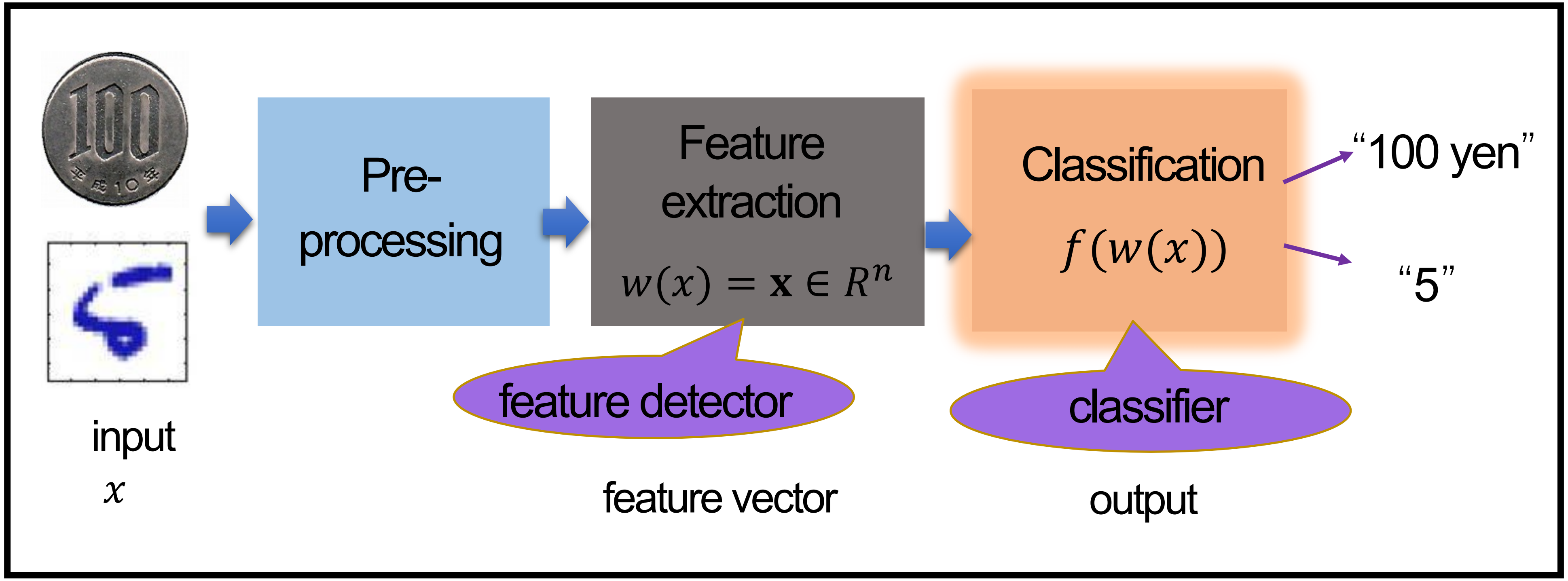
Pattern recognition

- Pattern recognition pipeline



Pattern recognition

- Pattern recognition pipeline



Pattern recognition

- In computer science, a pattern is represented using vector - feature values.
- Feature Extraction: Basically, at this stage you get all the features you want from your input data to give to the classifier (features are things like shape, weight, length, etc).
- Features may be represented as continuous, discrete, or discrete binary variables.
- A feature is a function of one or more measurements, computed so that it quantifies some significant characteristics of the object.

Pattern recognition

- A pattern (or an object) can be represented by a combination of attributes/descriptors.

$$\{x_1, x_2, \ldots, x_n\}$$

- A set of descriptors is referred to as "features" and a feature is defined as a vector (feature vector).

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$$

- A pattern class (category) ω_i is a group of patterns that some shared common characteristics:

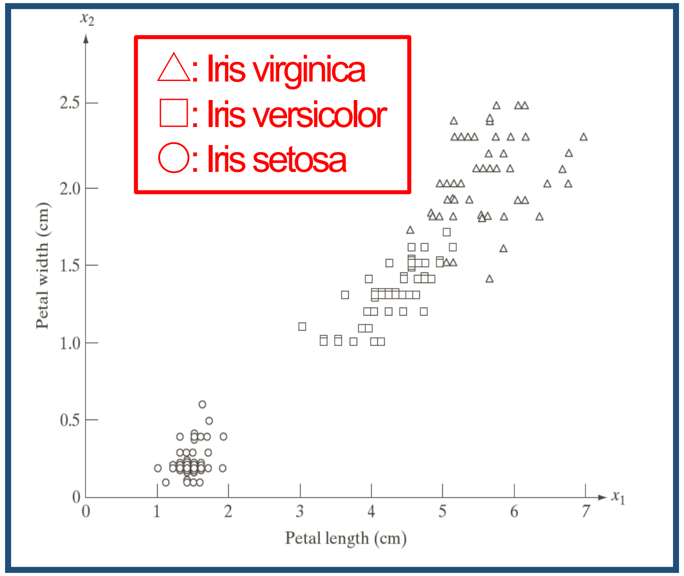
$$\omega = \{\omega_1, \omega_2, \ldots, \omega_W\}$$

where W is the number of classes

Feature vectors

- Feature Vectors can be represented as: $x = (x_1, x_2, \ldots, x_n)^T$.
  - where each component x_i represents the i^{th} descriptor and n is the number of descriptors associated with the pattern.
  - The nature of x_i is determined by the approach employed to describe the physical pattern.
- Example: *Iris* flower data set or Fisher's *Iris* data set [1]
  - The widths and lengths of the petals of three different species of iris flowers are identified.
  - 2-D feature vector can be defined as: $\mathbf{x} = (x_1, x_2)^T$

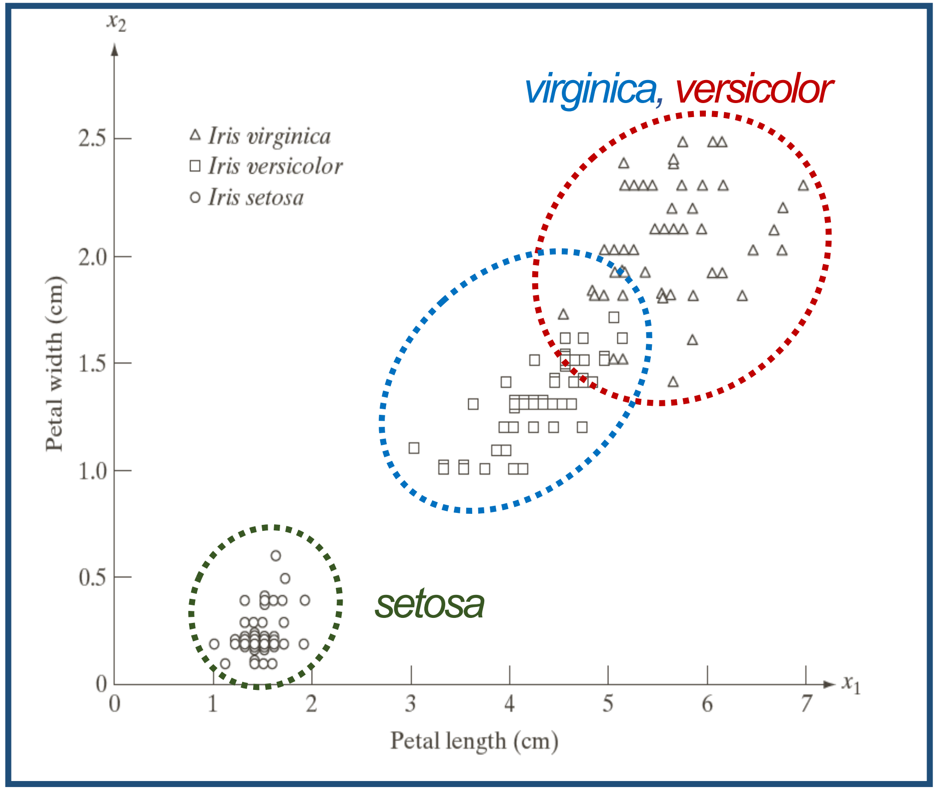
Petal length width



[1]. https://archive.ics.uci.edu/ml/datasets/iris

Feature vectors

- Each flower can be viewed as a point in two-dimensional Euclidean space (feature space).
  - The width and length of flower petals vary; therefore, pattern vectors also vary.
  - The width and length of these petals vary not just between classes, but also within a class.
- The 2-D features of the class 'setosa' are well separated from the other two classes, as shown in this figure.
  - The features of the other two classes are not fully separated from each other.
- The degree of class separability depends on the choice of features.
- This also affects recognition accuracy.

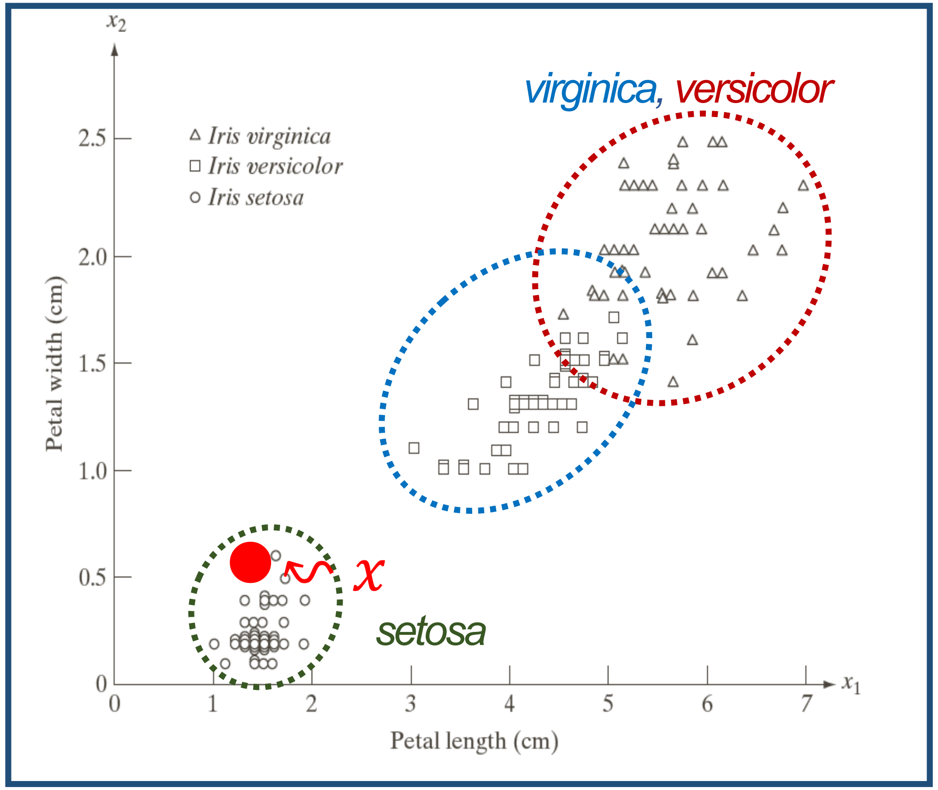


2-D Feature Space

A simple recognition approach

- For any unknown pattern x, our objective is to identify the class that x belongs to.
- We can do this by plotting x on the feature space.
- We believe that x belongs to the 'setosa' class since it is quite close to the patterns in this class.
- This is a simple and effective recognition approach.

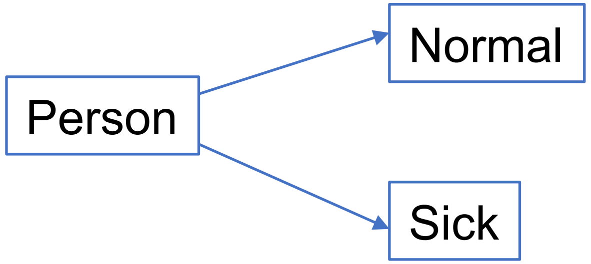
Based on this concept, the *k*-Nearest Neighbors Algorithm (*k*-NN algorithm) was developed.



2-D Feature Space

Pattern recognition

- Example:



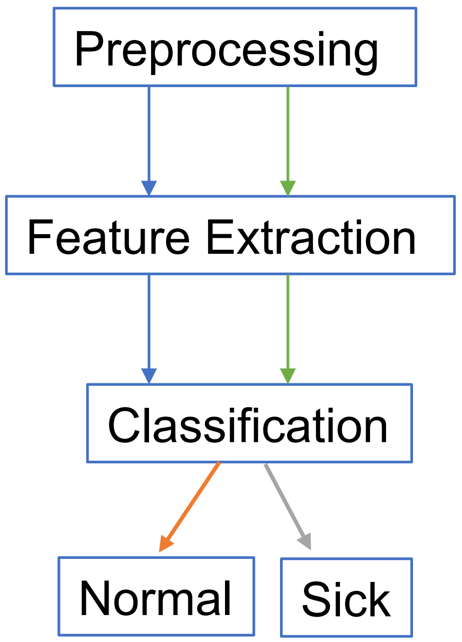
- You can set some measures to extract features:
  - Temperature
  - Blood Pressure
  - Age
  - Weight, etc
- Information from a single person is sent to a feature extractor whose purpose is to reduce the data by measuring certain features.
- Then, the features (attributes) are passed to a classifier.

Pattern recognition

- If a person comes to the hospital.
- The system takes the person's measurements (e.g., Pressure, Temperature, Weight, Age, etc.).
- The pre-processing in this case may be medical steps needed to measure the pressure, the temperature, the blood sugar, etc.
- The features are combined in a one vector representing the person (Feature Vector).

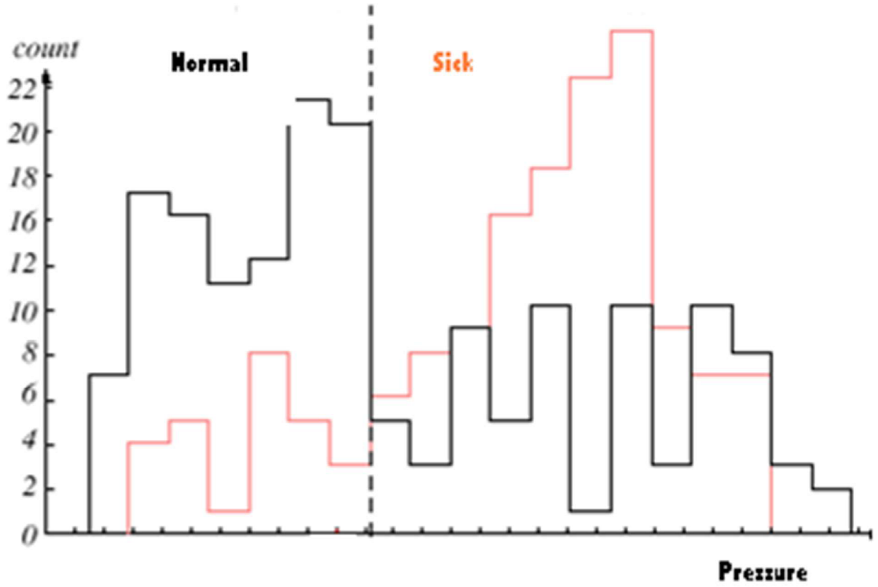
$$X = [130\ 85\ 37.5\ 77\ 45]$$

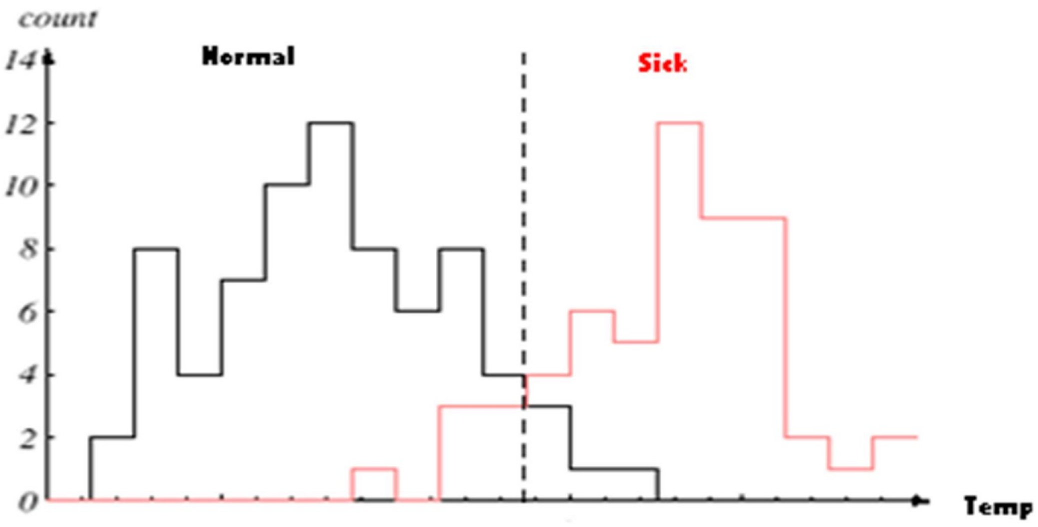
- This feature vector is the input to the classifier.



Pattern recognition

- In most real application, we need to obtain few features (attributes) so that each pattern (person) is represented by a vector of features.
- Using the training data, sketch the histogram of each class.
- Choose a threshold value above which the person is considered sick and vice versa.
- We can select some features values such as pressure and temperature as possible features for discrimination.



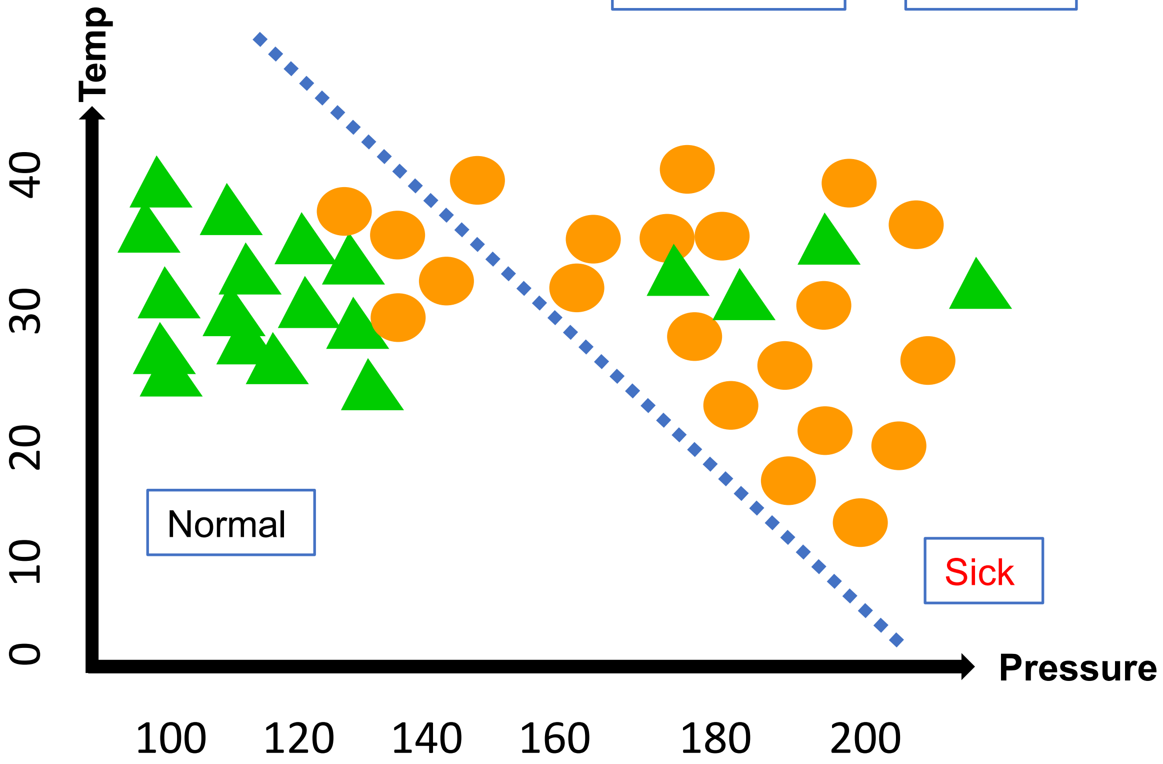


Pattern recognition

- Adopt the Pressure and add the Temperature to form a feature vector:

Person ⇨ $X^T = [X_1, X_2]$

X_1 → Pressure, X_2 → Temp.

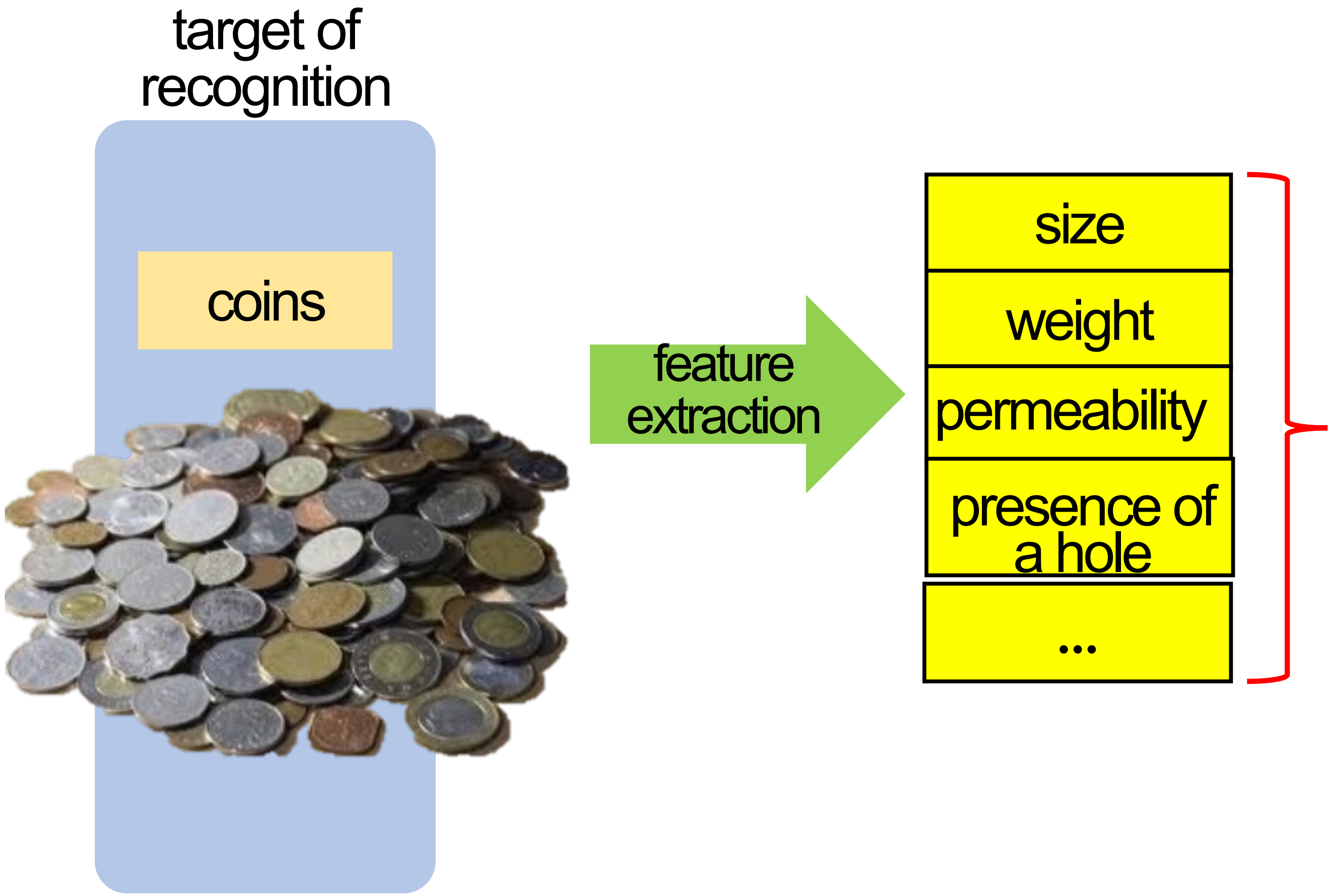


Pattern recognition

- Example: consider how a vending machine recognizes a coin.

Pattern recognition

- When we put a coin into the machine, certain features are extracted from it.

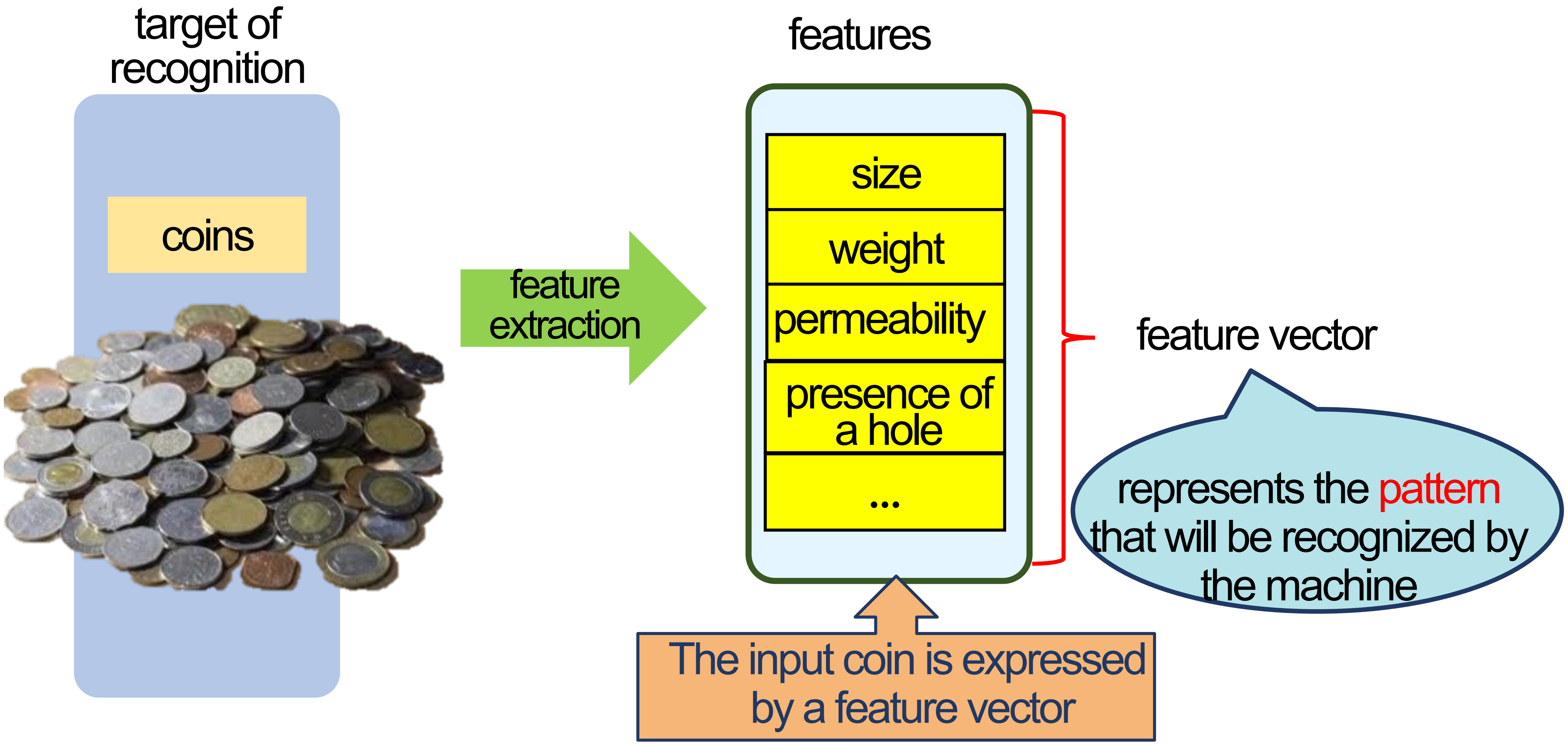


These descriptors (feature vector) are believed to be useful in identifying coins

Pattern recognition

- Steps for pattern recognition:

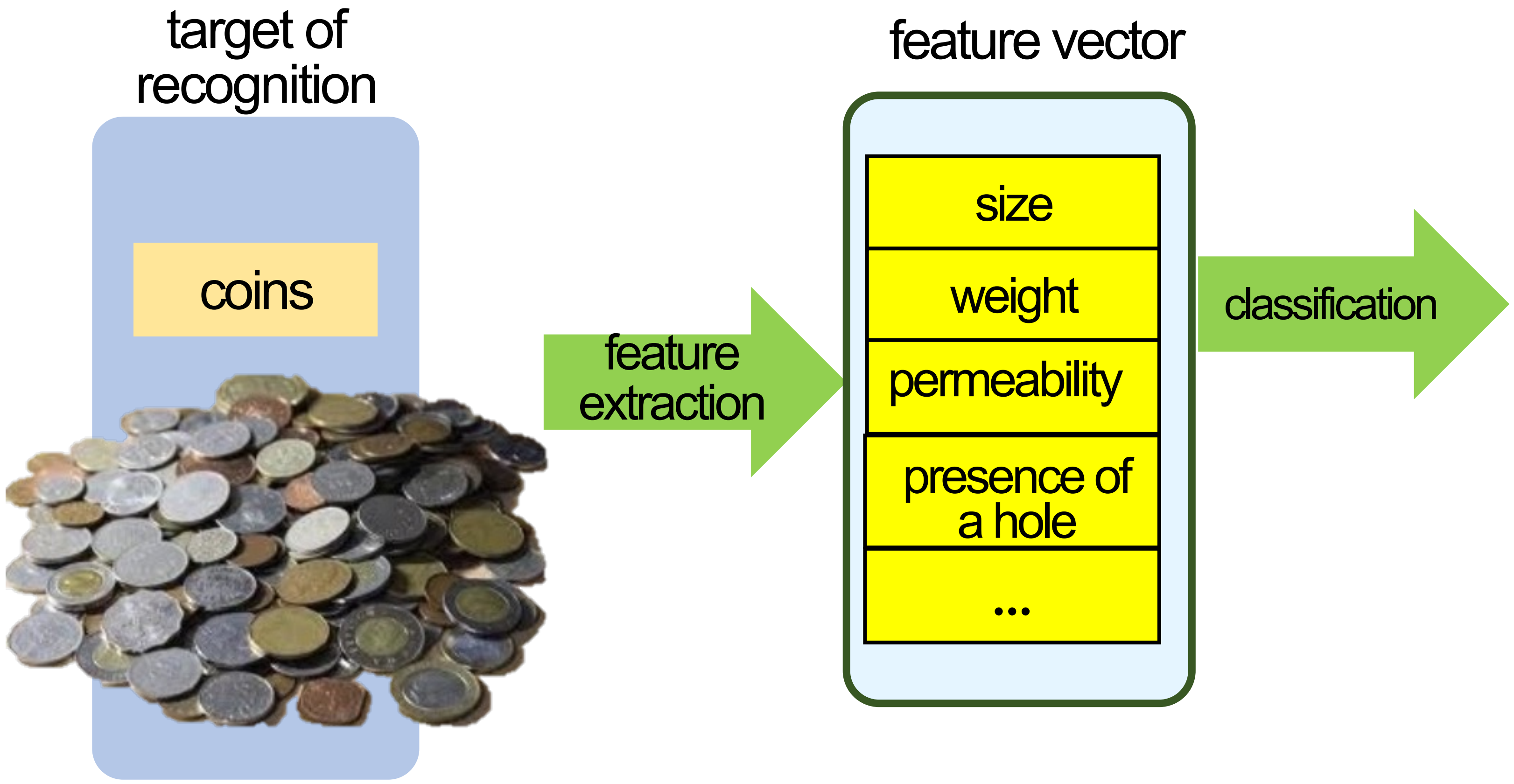
1. Feature extraction



Pattern recognition

- Steps for pattern recognition:

2. Classification or recognition - Input coins are assigned to a class based on some decision rules.



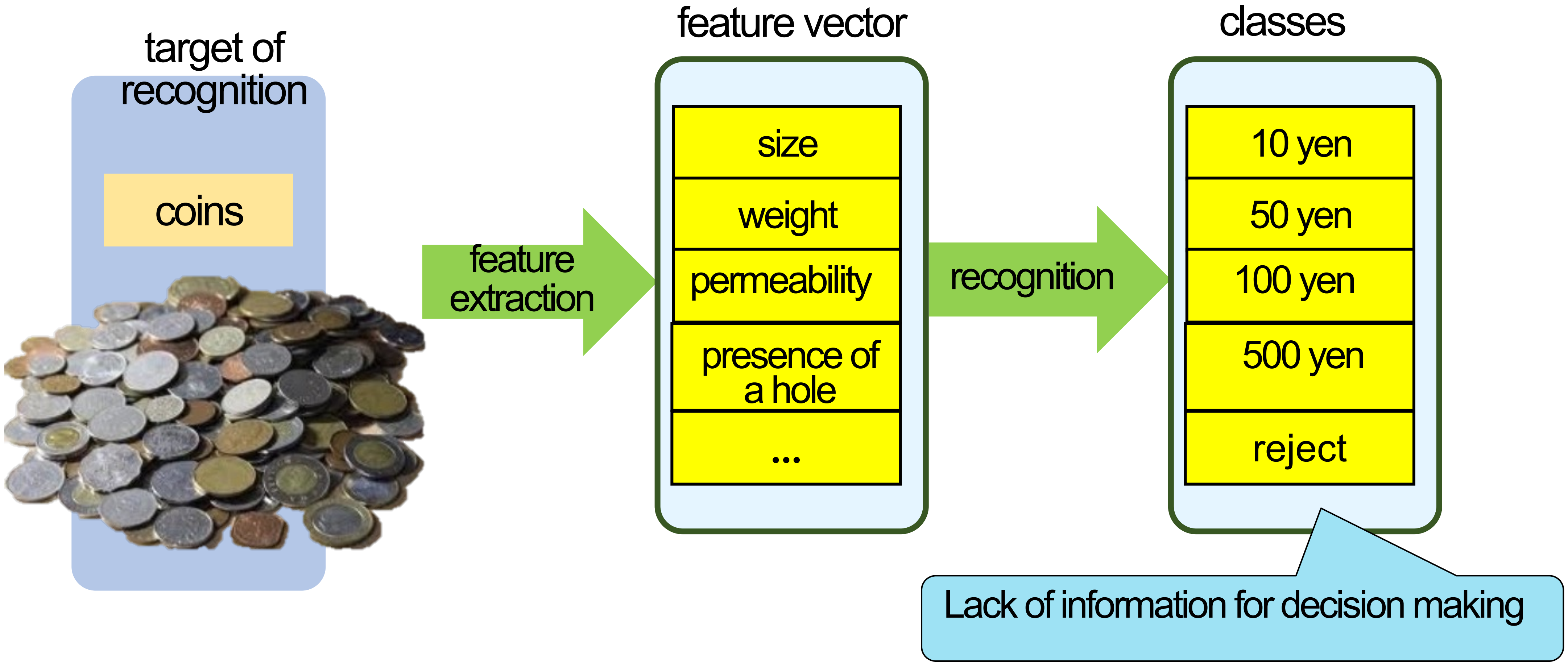
decision rules:

ϕ: [23.5,23.6](cm) &
w: [4.4, 4.6](g) &
μ: $[1.24\times10^{-6}, 1.26\times10^{-6}]$
h: no

10-yen

Pattern recognition

- In this case, the output would be one of five classes.
- The option "reject" indicates there is not enough information to make a decision.

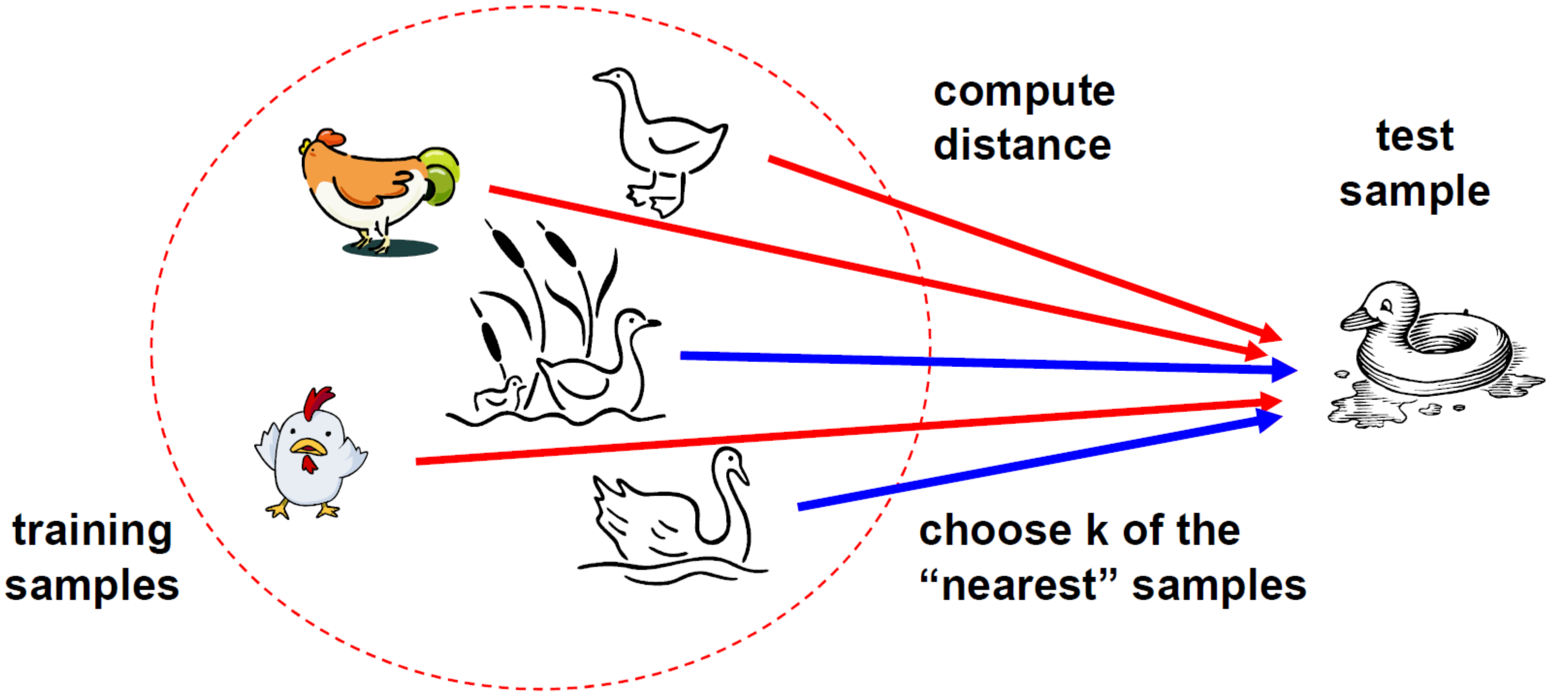


k-Nearest Neighbors Algorithm (k-NN)

- *k*-NN algorithm is the widely used basic pattern recognition algorithm.
- It is a non-parametric method used for classification and regression task (we will only focus on classification).
- Approach:
  - The input consists of test x (target for recognition) and some training samples (training data) in the feature space.
  - Each training sample has a class label (ground truth value).
  - The output is the class assigned to target x based on the estimated feature space.
- The *k*-NN technique is known as the nearest neighboring algorithm (NN algorithm) when *k* = 1.

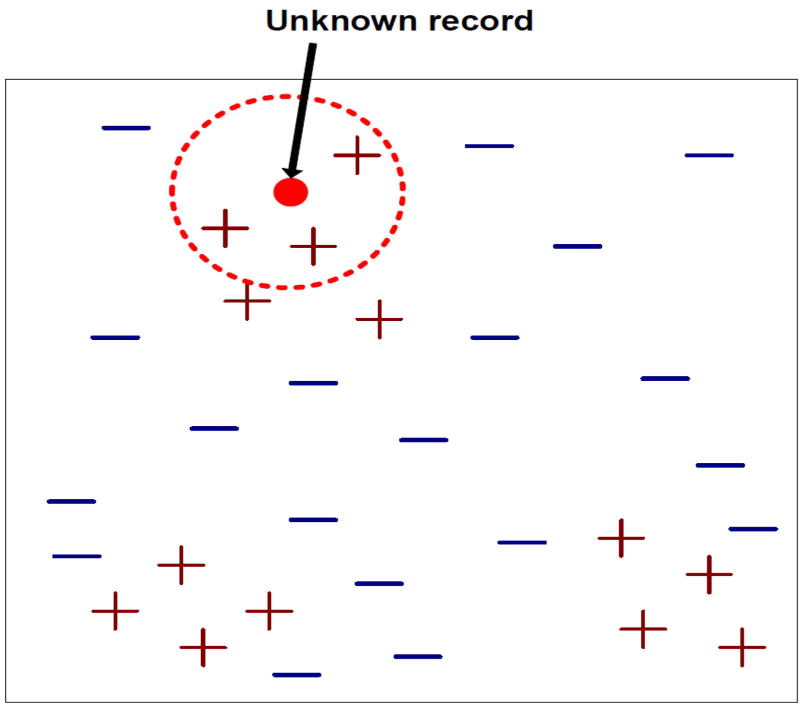
k-Nearest Neighbors Algorithm (k-NN)

- Basic idea:
- If it walks like a duck, quacks like a duck, then it's probably a duck.



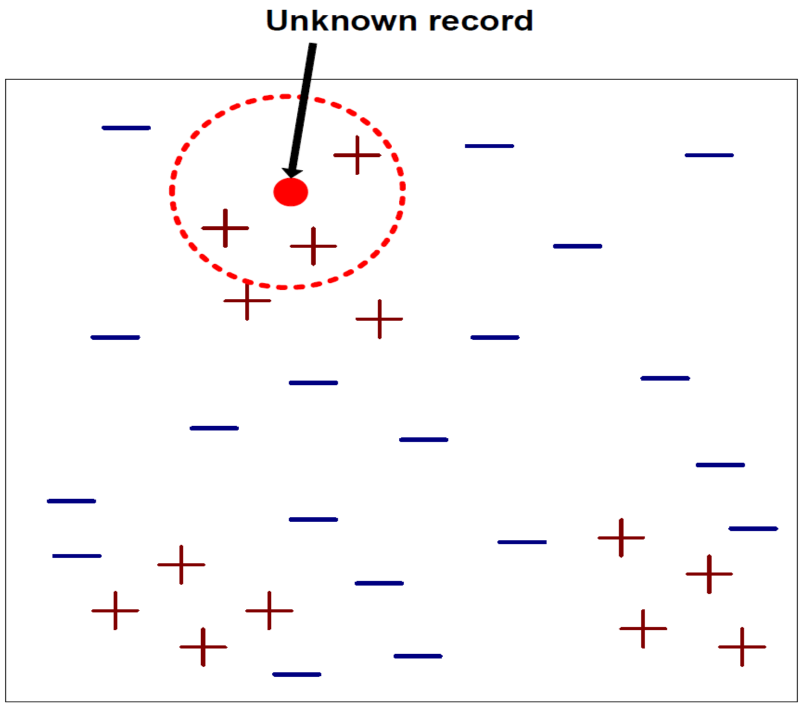
k-Nearest Neighbors Algorithm (k-NN)

- It requires three inputs:
  - The set of stored samples.
  - Distance metric to compute the distance between samples.
  - The value of *k*, the number of nearest neighbor to retrieve.



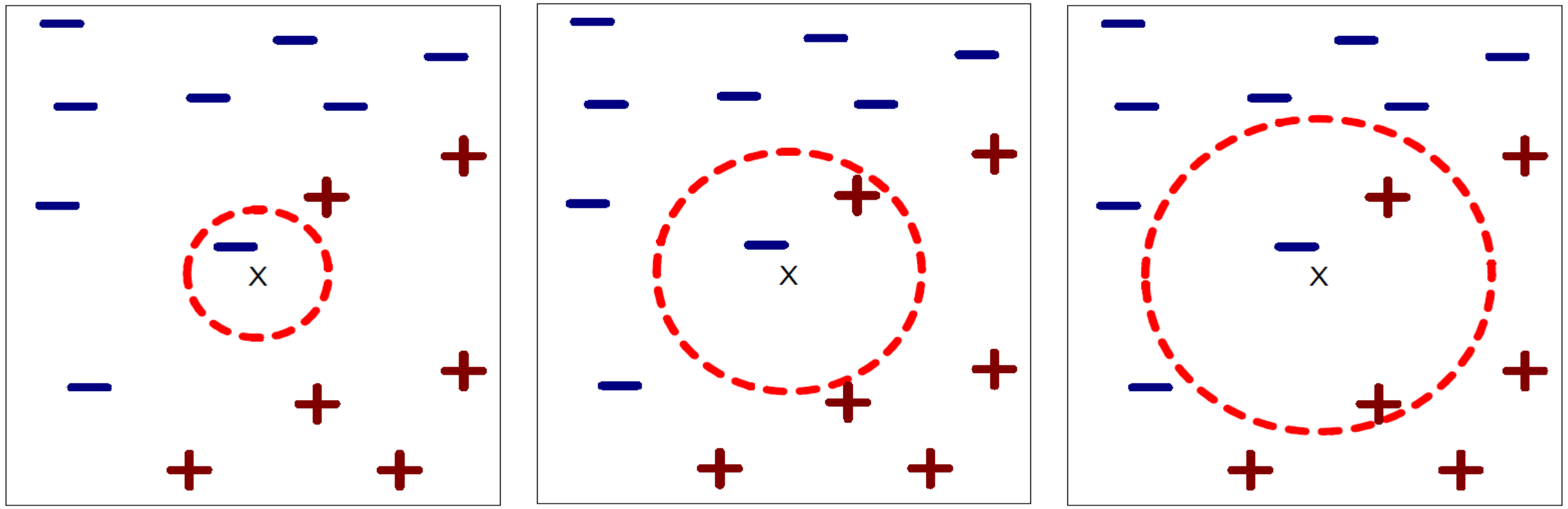
k-Nearest Neighbors Algorithm (k-NN)

- To classify unknown record:
  - Compute distance to other training records.
  - Identify *k* nearest neighbor.
  - Use class label of nearest neighbor to determine the label of unknown record (e.g., by taking majority vote).



k-Nearest Neighbors Algorithm (k-NN)

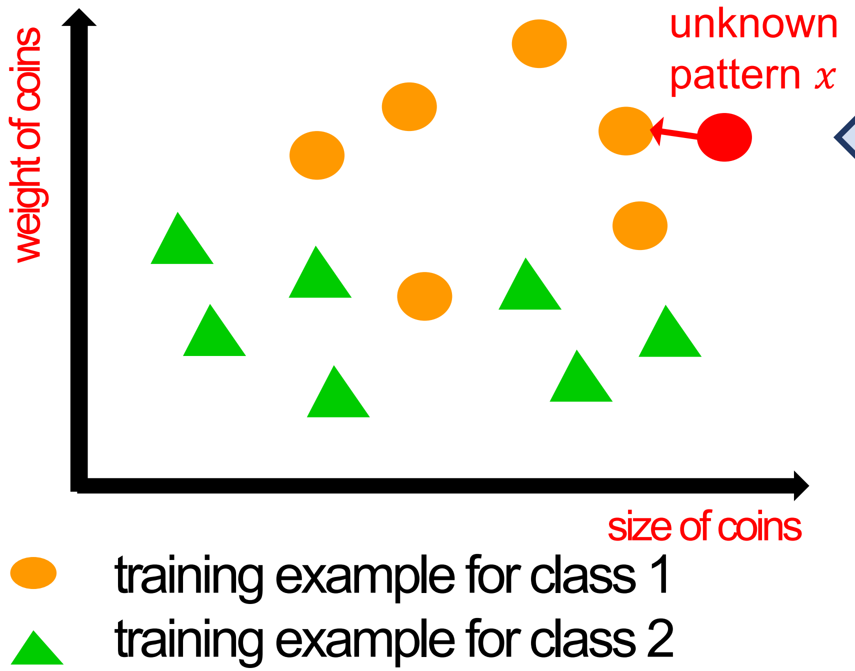
- Definition of nearest neighbor:
- *k*-nearest neighbors of a sample *x* are data points that have the *k* smallest distances to *x*.



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Nearest Neighbor Algorithm

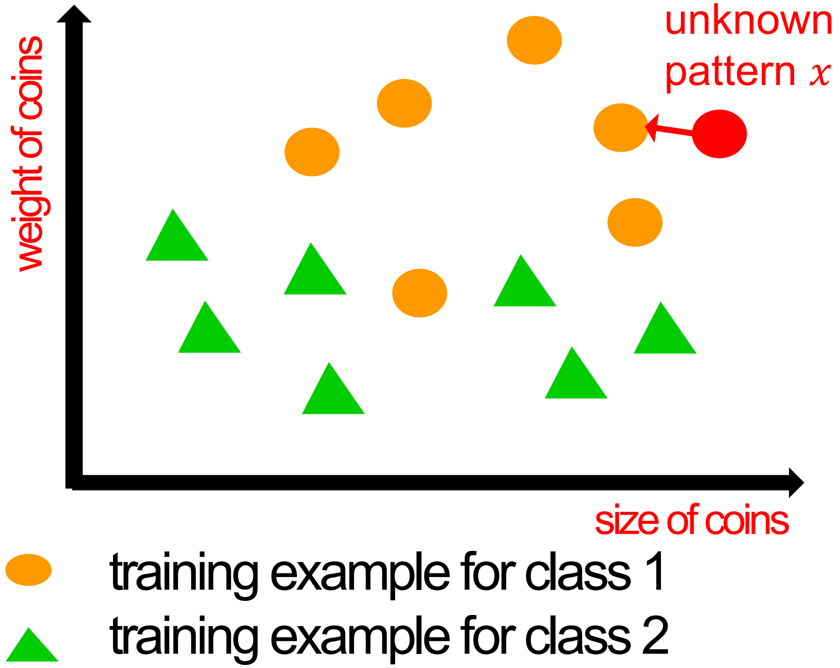
- The basic principle is that the unknown pattern is simply assigned to the same class as its closest training sample.
- Consider a case in which there are only two classes.



x is assigned to class 1, because its nearest neighbor (a training data) belongs to class 1

Nearest Neighbor Algorithm

- What is the best way to calculate the distance between two patterns?
- We can use Euclidean distance, but other types of distance are also possible.



Euclidean distance between unknown pattern x and the j-th example in class i can be calculated by the length of the vector:

$$d(x, x_j^{(i)}) = \left\| x - x_j^{(i)} \right\|$$

Nearest Neighbor Algorithm

- Compute distance between two points:

Euclidean distance - $\mathrm{d}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_i (x_i - y_i)^2}$

- Determining the class from nearest neighbor list
  - Take the majority vote of the class labels among the *k*-nearest neighbors.
  - Weight the votes according to distance.

    Weight factor $w = 1/d^2$

Decision Rule for NN Algorithm

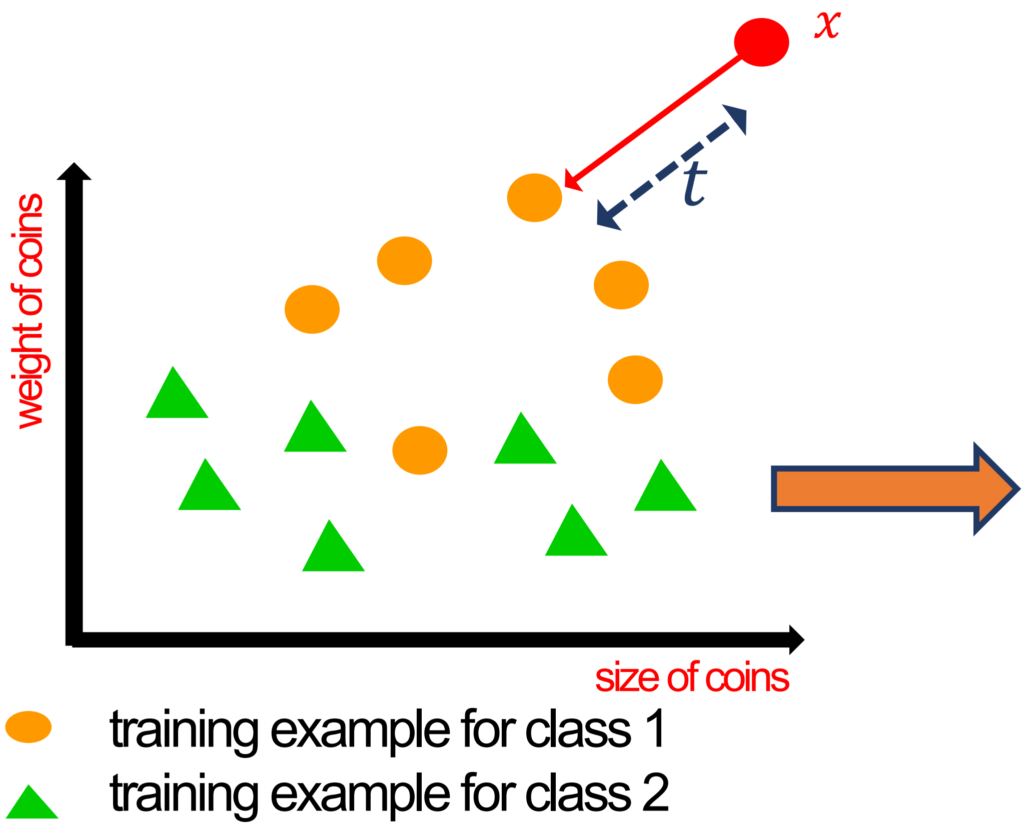
- The decision rule for NN algorithm:
  - When the distance between a pattern x and its nearest training example is less than a certain threshold t, the output class membership should be the same as the nearest training example's.
  - Otherwise, "reject" should be output (unrecognizable).
- This can be described in the form of a formula such as:

$$\text{Output} = \begin{cases} \boxed{\operatorname*{argmin}_i d\left(x, x_j^{(i)}\right)} & \min d\left(x, x_j^{(i)}\right) \le \mathrm{t} \\ \text{reject} & \text{otherwise} \end{cases}$$

* argmin: argument of the minimum distance

Decision Rule for NN Algorithm

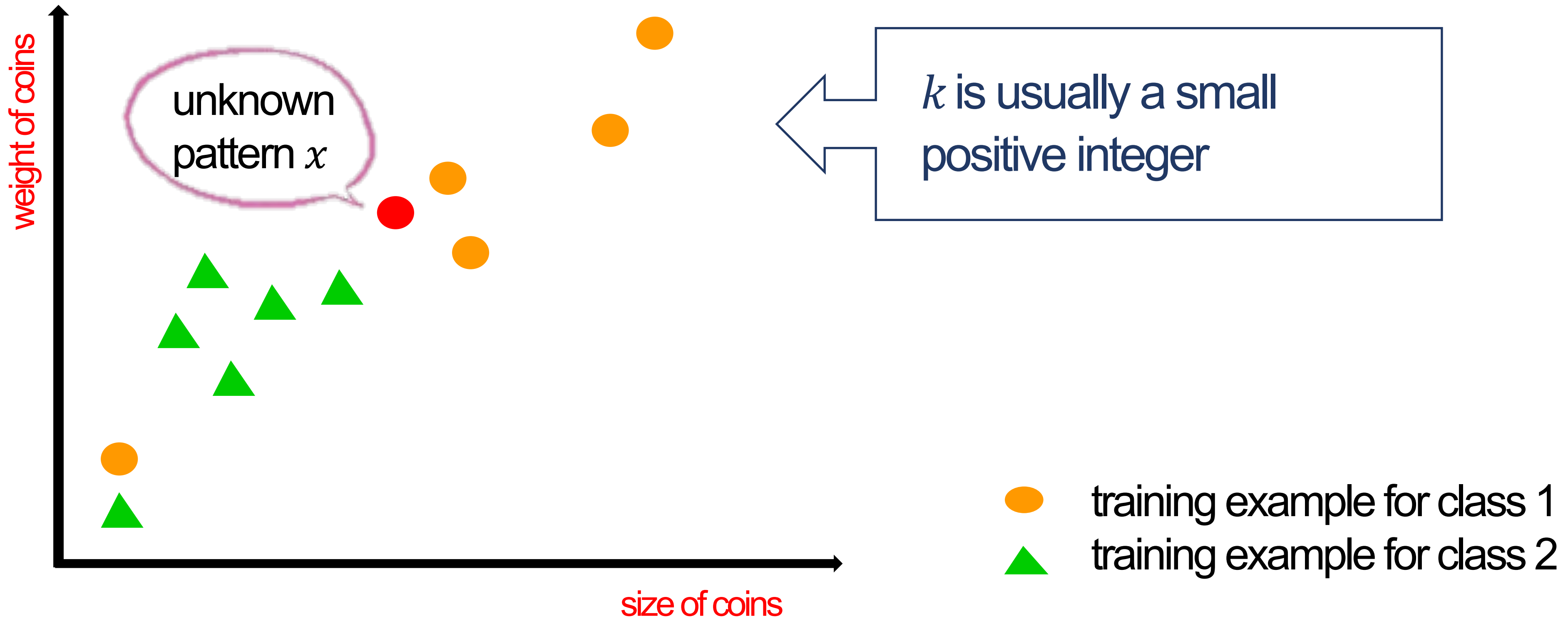
- Constrain/threshold



"reject" The distance from x to the nearest training example exceeds certain threshold t.

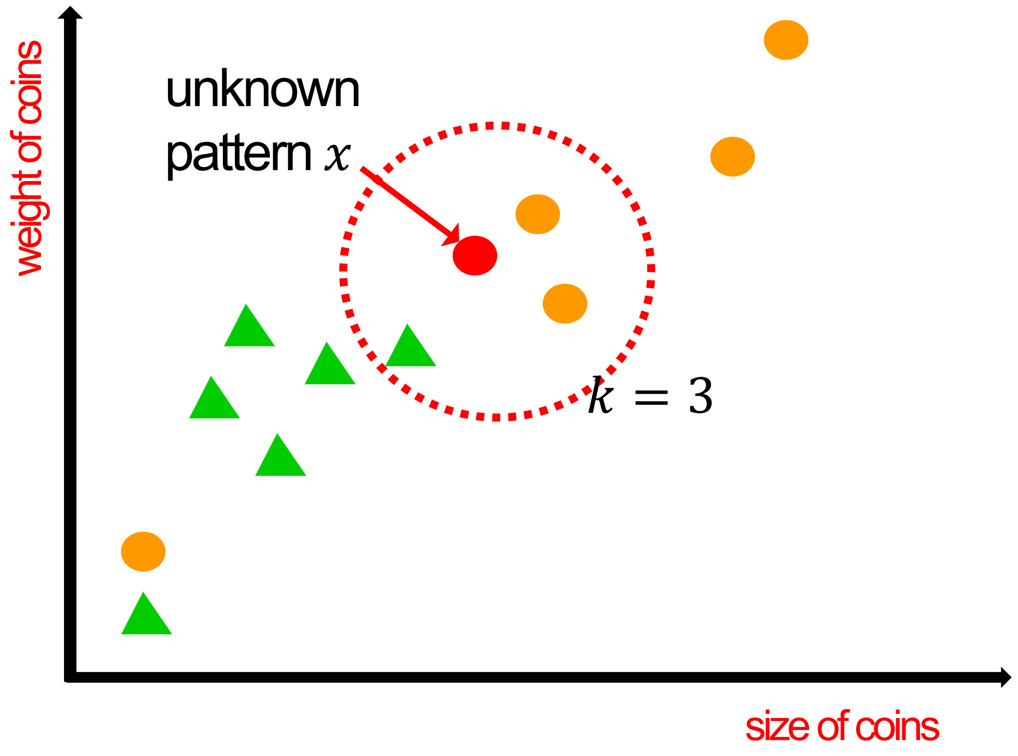
k-Nearest Neighbors Algorithm (k-NN)

- A pattern is classified by a majority vote of its neighbors.
  - An unknown pattern is assigned to the class most common among its k-nearest training examples.



k-Nearest Neighbors Algorithm (k-NN)

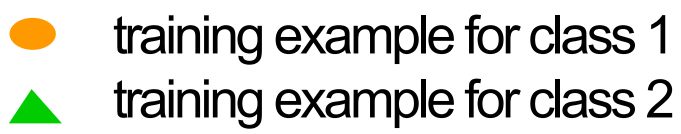
- The unknown pattern is assigned to the most common class among its k-nearest neighbors.



In case $k = 3$, we check the nearest three training data

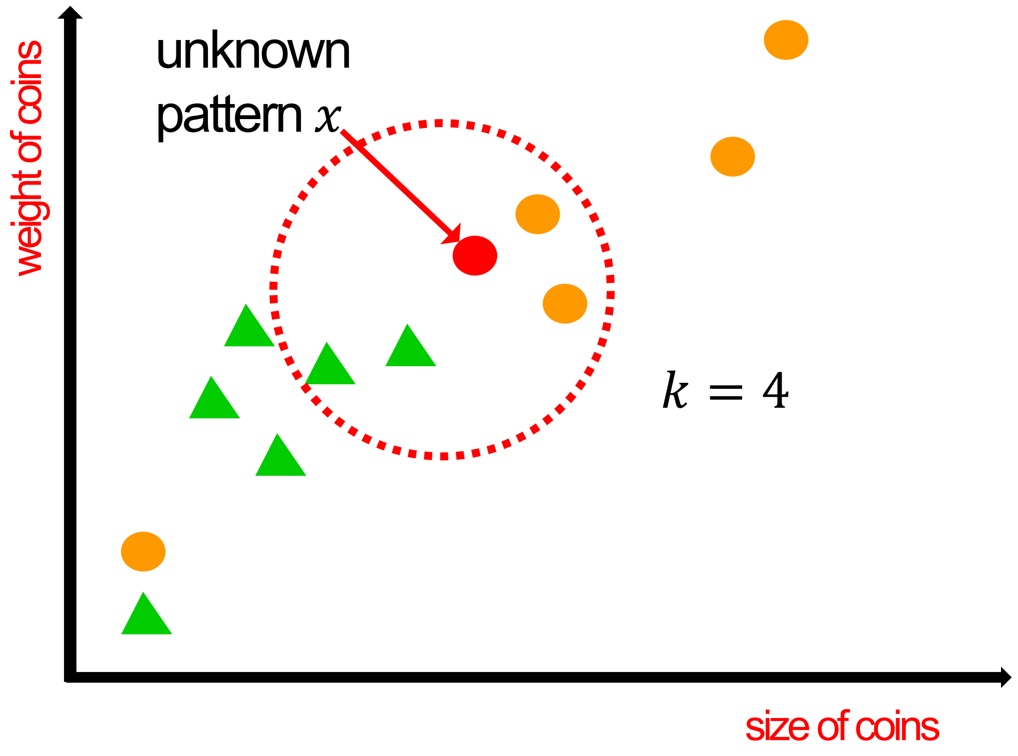
- Two data points in class 1
- One data point in class 2

The output "class 1" in this case



k-Nearest Neighbors Algorithm (k-NN)

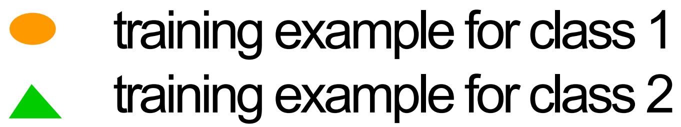
- The unknown pattern is assigned to the most common class among its k-nearest neighbors.



In case $k = 4$, we check the nearest four training data

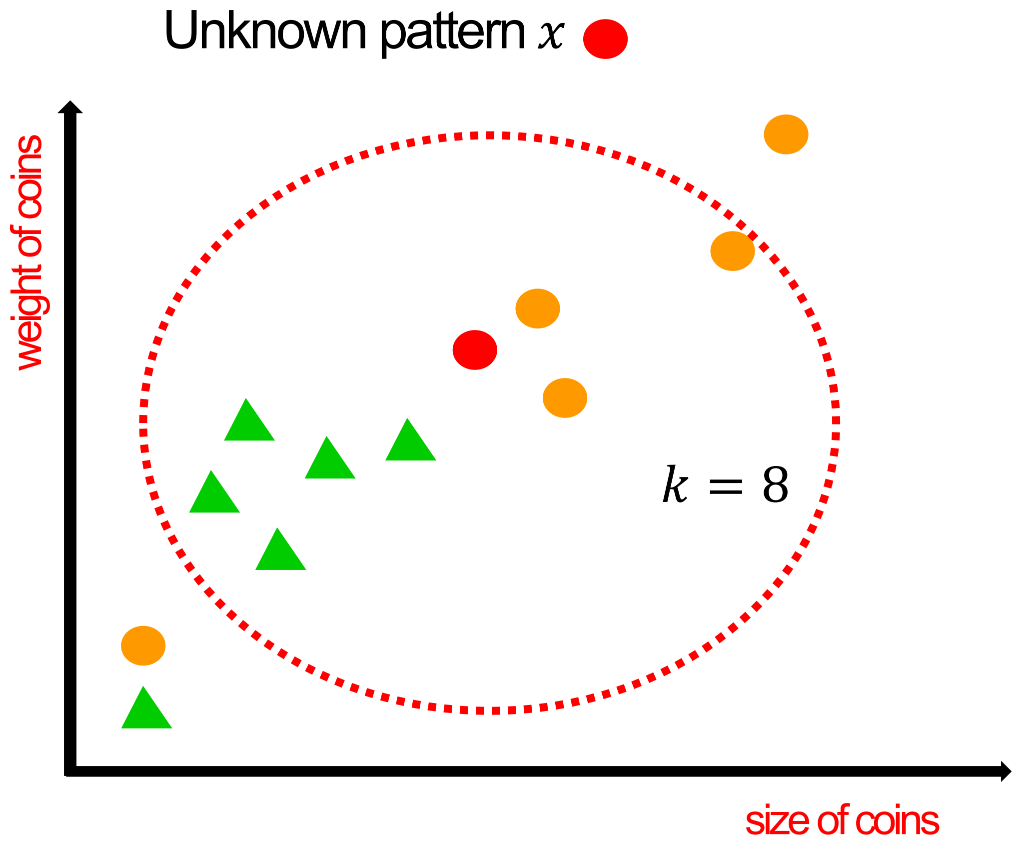
- Two data points in class 1
- Two data point in class 2

output “reject"



k-Nearest Neighbors Algorithm (k-NN)

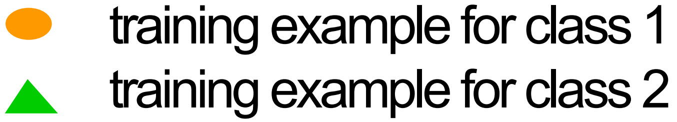
- The unknown pattern is assigned to the most common class among its k-nearest neighbors.



In case $k = 8$, we check the nearest eight training data

- Three data points in class 1
- Five data point in class 2

output “class 2"

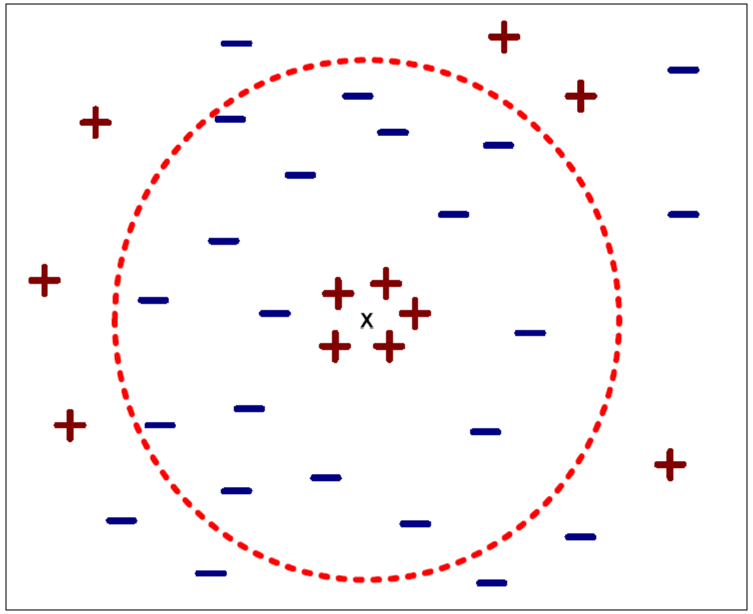


Decision Rule for k-NN Algorithm

- The decision rule for k-NN algorithm:
  - When the distances between a pattern x and its k-nearest training examples are less than a certain threshold t, the output class membership as same as the class most common among its k-nearest neighbors.
  - Otherwise, output “reject” (unrecognizable).

Nearest Neighbor Classification

- Choosing the value of *k*:
  - If k is too small, sensitive to noise points.
  - If k is too large, neighborhood may include points from other classes.



Nearest Neighbor Classification

- Features scaling: Features may have to be scaled to prevent distance measures from being dominated by one of the attributes.

For example: height of a person may vary from 1.5 m to 1.8 m
weight of a person may vary from 90 lb to 300 lb

- Generalization of feature values is required to make correct decisions.

Nearest Neighbor Algorithm

- For any given two images (test and training) and representing them as vectors l_1, l_2.
- Compare them using $L1$ distance to classify the image category.

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

Where the sum is taken over all pixels. Here is the procedure visualized:

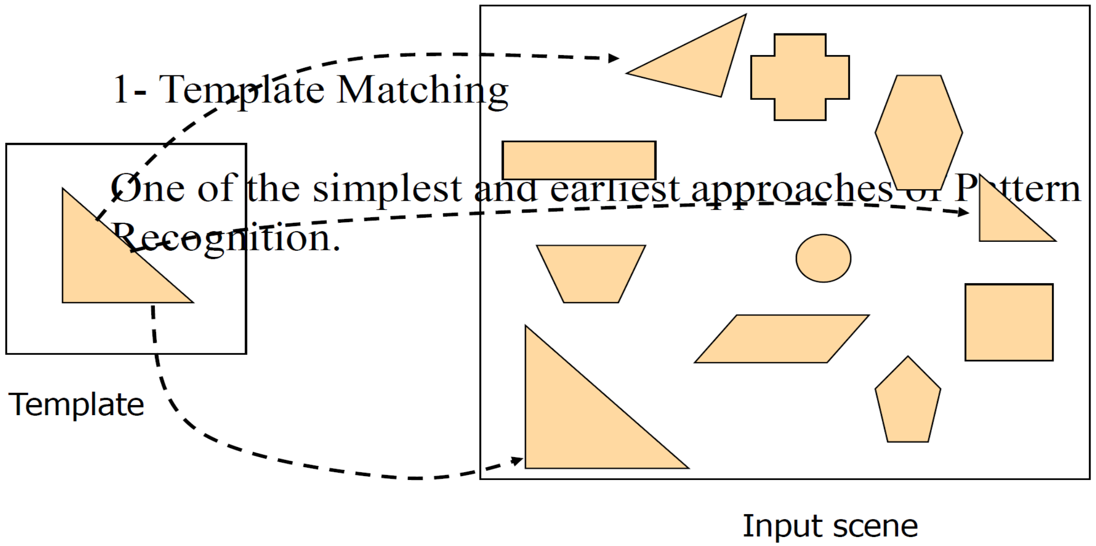
$$\left| \overset{\text{test image}}{\begin{bmatrix} 56 & 32 & 10 & 18 \\ 90 & 23 & 128 & 133 \\ 24 & 26 & 178 & 200 \\ 2 & 0 & 255 & 220 \end{bmatrix}} - \overset{\text{training image}}{\begin{bmatrix} 10 & 20 & 24 & 17 \\ 8 & 10 & 89 & 100 \\ 12 & 16 & 178 & 170 \\ 4 & 32 & 233 & 112 \end{bmatrix}} \right| = \overset{\text{pixel-wise absolute value differences}}{\begin{bmatrix} 46 & 12 & 14 & 1 \\ 82 & 13 & 39 & 33 \\ 12 & 10 & 0 & 30 \\ 2 & 32 & 22 & 108 \end{bmatrix}} \rightarrow 456$$

An example of using pixel-wise differences to compare two images with L1 distance (for one color channel in this example). Two images are subtracted elementwise and then all differences are added up to a single number. If two images are identical the result will be zero. But if the images are very different the result will be large.

Image Source: https://cs231n.github.io/classification/

Template matching

- Template matching is pattern recognition approach.
- Samples, Pixels or Curves are used as representations to determine the match area.
- Correction and Distance Measure are examples of recognition functions.
- Typical criterion like Classification error can be used.

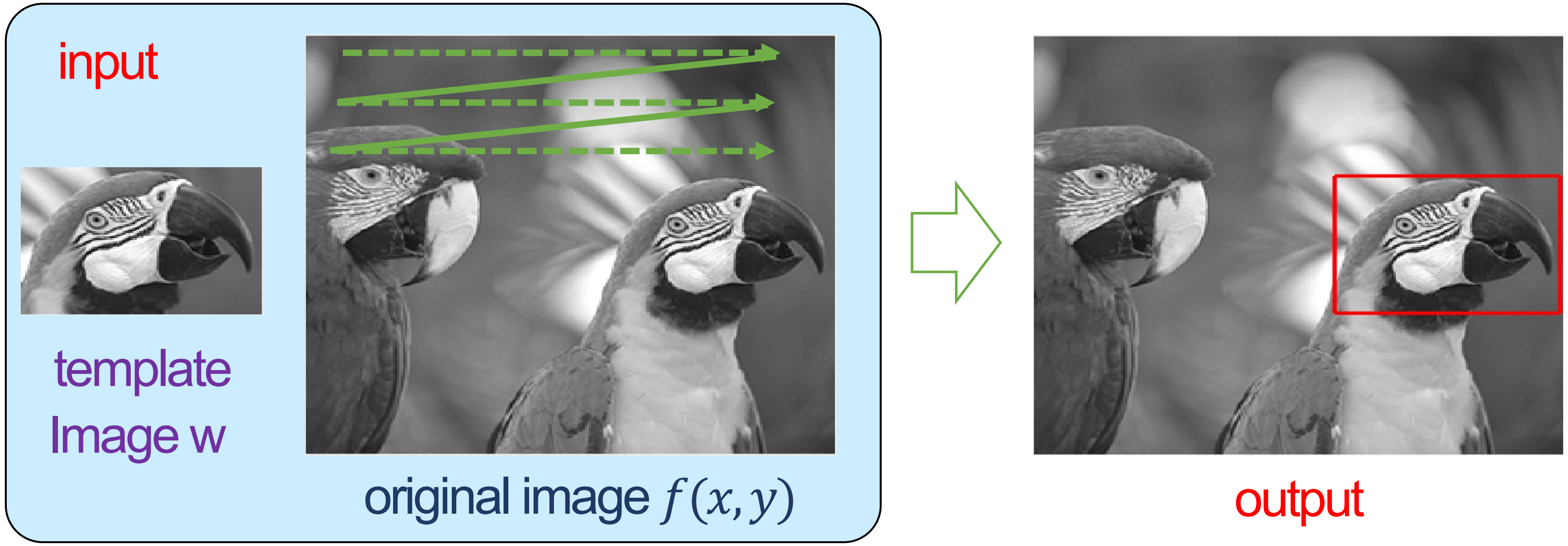


Template matching

- A template (typically, a 2D shape) or a prototype of the pattern to be recognized is available.
- The pattern to be recognized is matched against the stored template while taking into account all possible pose (translation and rotation) and scale changes.
- The similarity measure, often a correlation, may be optimized based on the available training set.

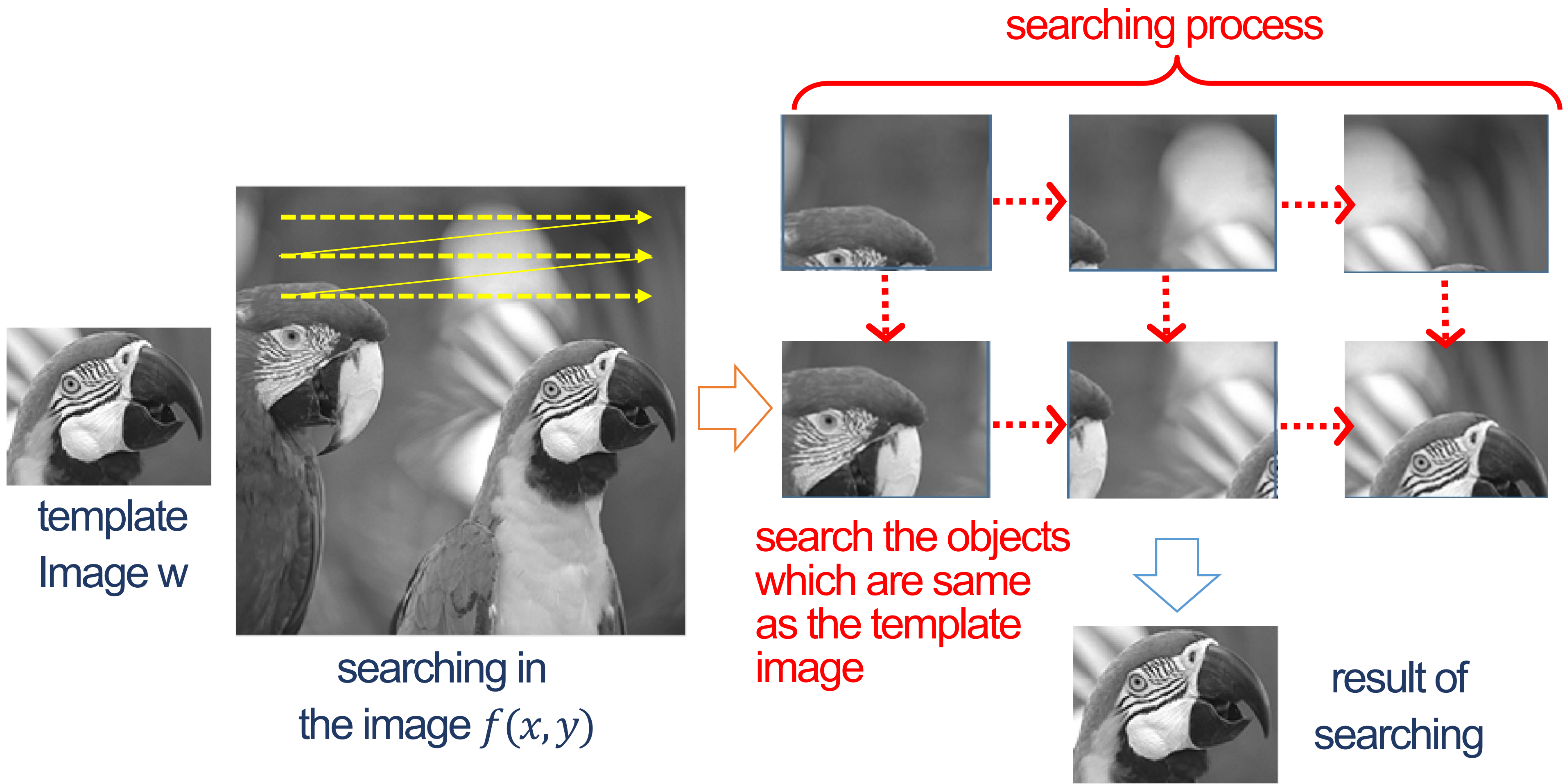
Template matching

- In image processing for finding small sections of an image that match a template image.
  - This is considered as a simple recognition problem in which no features are extracted (use the image itself as features).
  - Input: an original image $f(x, y)$ and a template image w.
  - Output: image with the location of the template.



Template matching

- If the image is significantly larger than the template image/
- Template w is moved for all values of the displacement variables x and y of $f(x, y)$.



Template matching

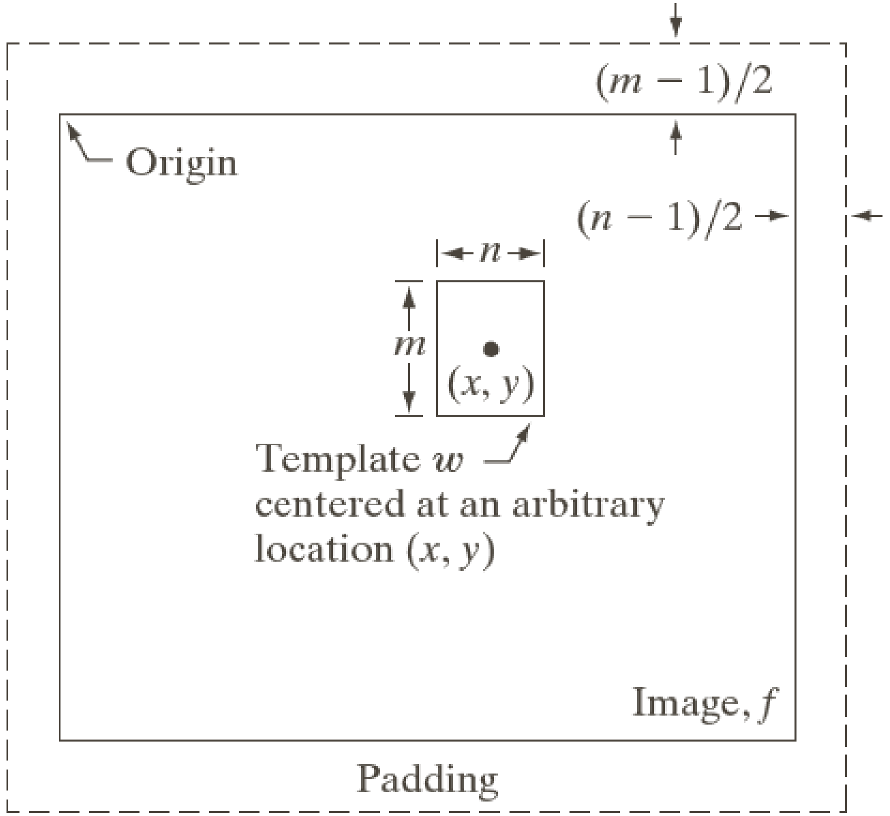
- At each position, the similarity is evaluated to find where the best match occurs.
- Often, Normalized Correlation Coefficient is used

$$\gamma(x,y) = \frac{\sum_s \sum_t [w(s,t)-\bar{w}] \sum_s \sum_t [f(x+s,y+t)-\bar{f}(x+s,y+t)]}{\left\{\sum_s \sum_t [w(s,t)-\bar{w}]^2 \sum_s \sum_t [f(x+s,y+t)-\bar{f}(x+s,y+t)]^2\right\}^{\frac{1}{2}}} \qquad (1)$$

  - $\bar{w}$: the average value of the template (computed only once)
  - $\bar{f}(x+s,y+t)$: the average value of f in the region coincident with w
  - The limits of summation are taken over the region shared by w and f
  - $\gamma(x,y)$ has the values in the range [-1,1]
  - If w is a part of the image $f(x,y)$, its maximum value (1) occurs

Procedure of template matching

- Padding the border around f. Padding is necessary when the center of w is on the boundary of f.
- The correlation at an arbitrary point is obtained by applying Eq.1 (slide 28).
- The center of w is moved to an adjacent location and the procedure is repeated.
- The complete $\gamma(x, y)$ is obtained by moving the center of w so that the center of w visits every pixel in f.
- Find for the maximum value in $\gamma(x, y)$ to find where the best match occurs.



A template of size $m \times n$ whose center is at an arbitrary location (x, y)

Example of template matching

(A) Satellite image (913×913) of Hurricane Andrew*

- where the eye of the storm is clearly visible

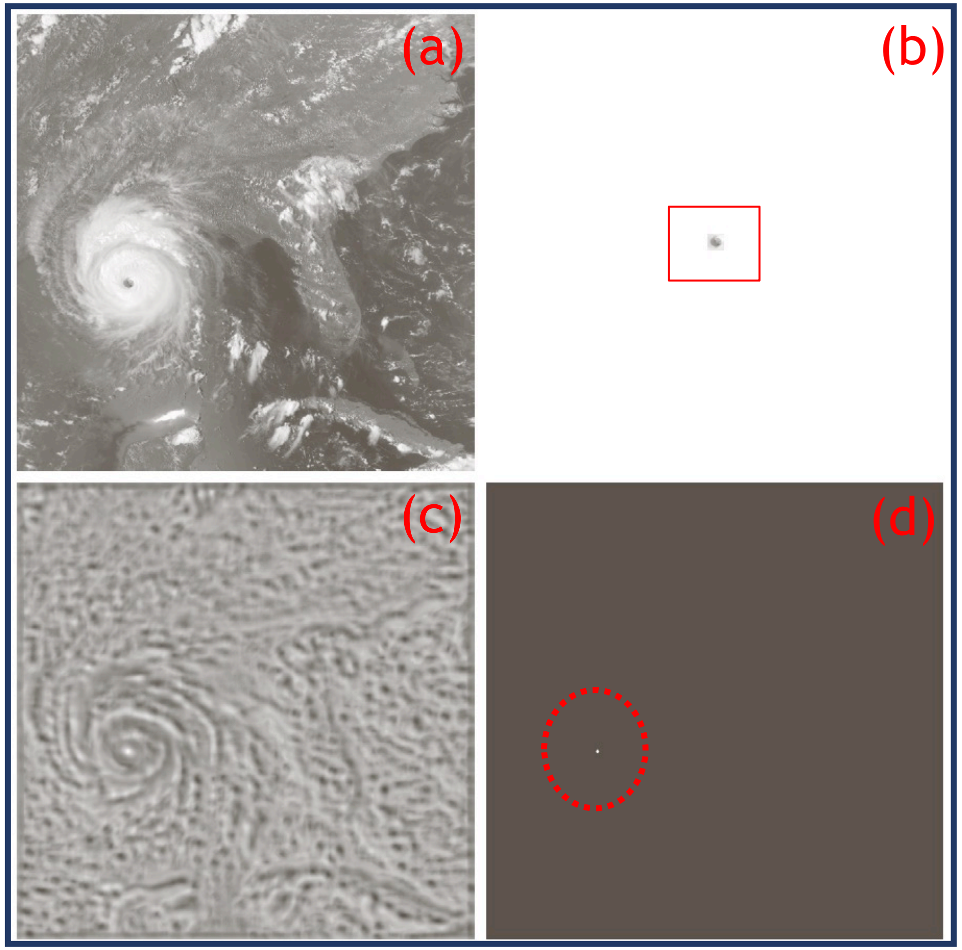
(B) Template w, a small sub-image (31×31) of the eye of the storm

- We want to find the location of the best match in f of w by template matching

(C) The result of computing the correlation coefficients

(D) The location of the maximum correlations (a white dot)

- Coincident with the eye of the storm



A big hurricane that struck some states of U.S.A. in 1992.

Thank you for your attention