Artificial Neural Networks and Deep Learning

- Introduction to Machine Learning and Deep Learning -

Giacomo Boracchi, PhD

https://boracchi.faculty.polimi.it/

Politecnico di Milano

Standard Programming

```python
sum = 0
a = int(input("Insert a: "))

while a > 0:
  sum += a
  a = int(input("Insert a: "))

print(f"Sum = {sum}")
```

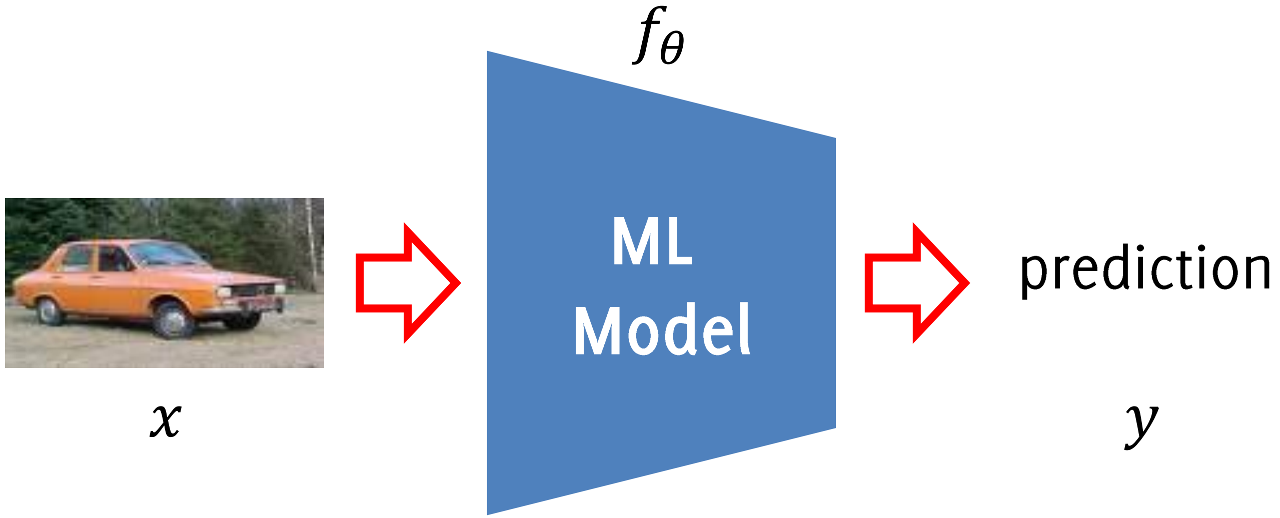
Standard Programming

```python
sum = 0
a = int(input("Insert a: "))

while a > 0:
  sum += a
  a = int(input("Insert a: "))

print(f"Sum = {sum}")
```

Can you write a program that takes as input an image and tells whether it contains a car or a motorbike?

Machine Learning Paradigms

ML is the solution! Here the C program **is replaced by a very big parameteric function** f_θ, whose paramters θ are learned from data!



$x \Rightarrow f_\theta(x) = y$

Machine Learning Paradigms

ML is the solution! Here the C program **is replaced by a very big parameteric function** f_θ, whose paramters θ are learned from data!

Learning consists is (automatically) defining the parameters θ of the **model** f.

Parameters θ are *learned* from data, following consolidated pipelines

Different settings applies, which give rise to the supervised or unsupervised settings

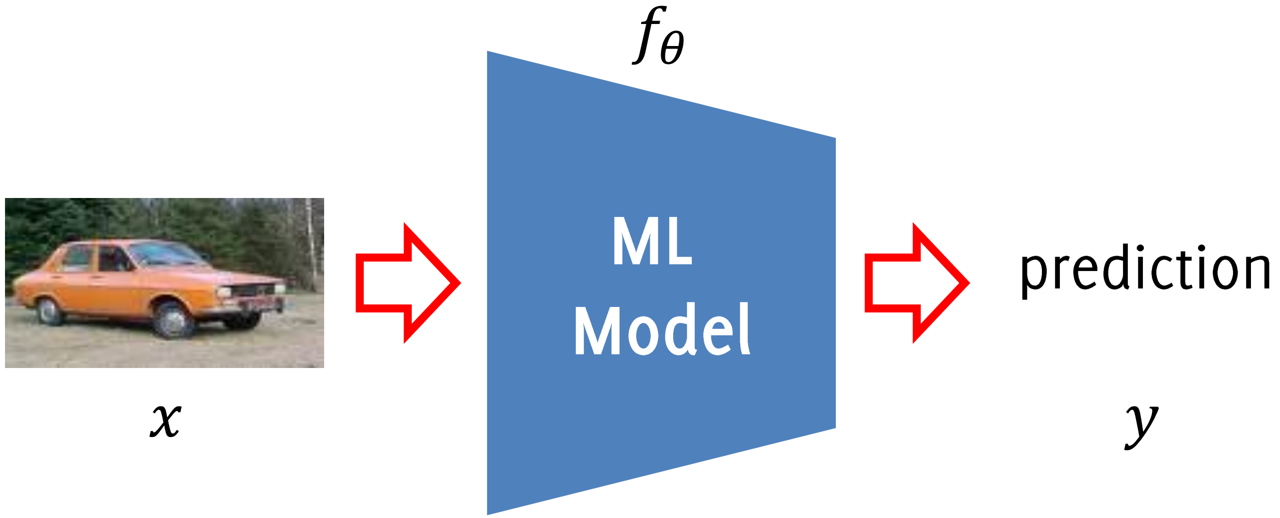
This course deals with a particular type of models: Neural Networks, which are very powerful in handling data like signals, images, videos, text…

Machine Learning Paradigms

ML is the solution! Here the C program **is replaced by a very big parameteric function** f_θ, whose paramters θ are learned from data!

Supervised Learning

- Classification
- Regression



$$x \Rightarrow f_\theta(x) = y$$

Supervised Learning

In **Supervised Learning** we are given a training in the form:

$$TR = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

where

- $x_i \in \mathbb{R}^d$ is the input
- $y_i \in \Lambda$ is the target, the expected output of the model to x_i

The set Λ can be

- A discrete set, as in classification $\Lambda = \{\text{"brown"}, \text{"green"}, \text{"blue"}\}$ (e.g., possible eye colors)
- An ordinal set (often continuous set, $\mathbb{R}$) in case of regression.

Λ can also be multivariate (e.g., regressing weight and height of an individual or estimating their eye colors and hair color)

Training Set for (binary) Image Classification

Cars

Motorcycles

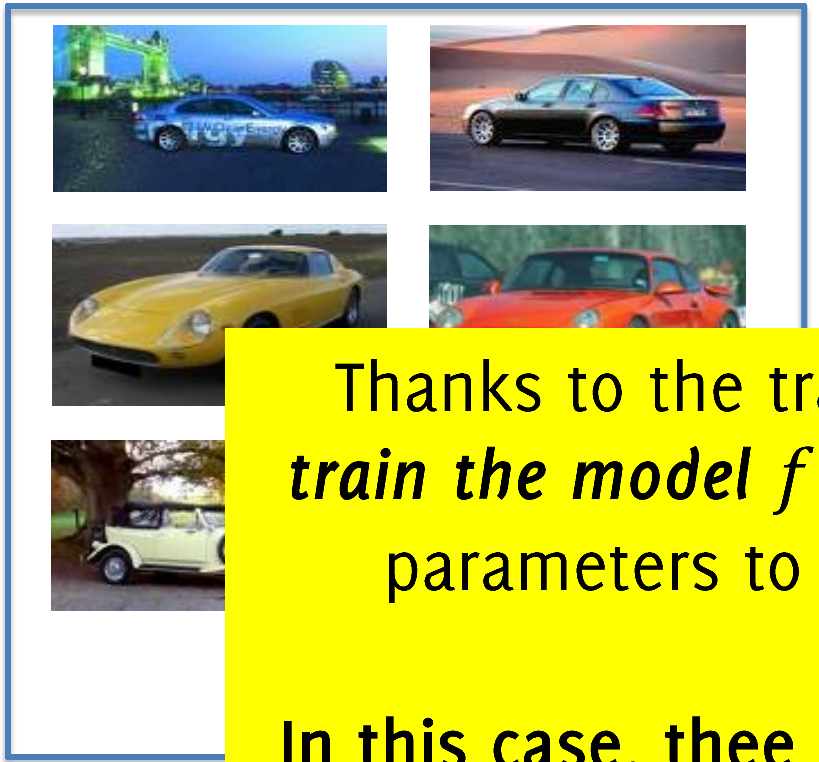
$$TR = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

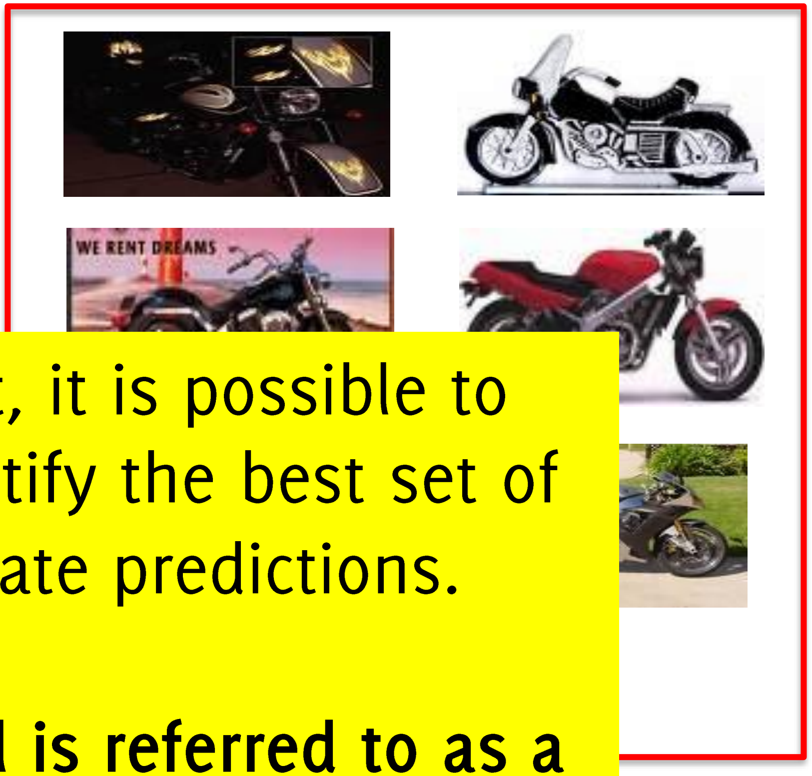
- $x_i \in \mathbb{R}^{R \times C \times 3}$ is the input image
- $y_i \in \{\text{"car"}, \text{"motorcycle"}\}$

An element in TR

, "car"

Training Set for (binary) Image Classification





Thanks to the training set, it is possible to ***train the model*** f and identify the best set of parameters to get accurate predictions.

In this case, thee ML model is referred to as a classifier!

- $x_i \in \mathbb{R}^{R\times C\times 3}$ is the input image
- $y_i \in \{"car", "motorcycle"\}$

ent in TR

(, "car")

Inference Using the Trained Classifier

Cars

Motorcycles

Classifier

Motorcycle

Training Set for Regression

12000 $

15000 $

6000 $

2000 $

8000 $

22000 $

4000 $

28000 $

6000 $

35000 $

$$TR = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

- $x_i \in \mathbb{R}^{R \times C \times 3}$ is the input image
- $y_i \in \mathbb{R}$

An element in TR

( ,"28000$")

Training Set for Regression

12000 $

15000 $

6000 $

2000 $

8000 $

22000

00 $

Thanks to the training set, it is possible to ***train the model*** f and identify the best set of parameters to get accurate predictions.

In this case, thee ML model is referred to as a regressor!

ement in TR

- $x_i \in \mathbb{R}^{R \times C \times 3}$ is the input image
- $y_i \in \mathbb{R}$

( ,"28000$")

Supervised learning: Regression

12000 $

15000 $

6000 $

2000 $

8000 $

22000 $

4000 $

28000 $

6000 $

35000 $

→ Regressor → 3800 $

Remarks on both Classification and Regression

- Number of classes can be larger than two: **multiclass classification**, (e.g., {"car", "motorcycle","truck"}).
- The input size needs to be fixed (in deep learning exception applies).
- Regression models can have more than 2 outputs (multivariate regression, e.g., estimating cost and weight of the vehicle).
- Training a Classifier or a Regressor requires different loss functions.
- Difference between classification or regression is not only on the fact that Λ discrete, but whether it is ordinal and on how we assess errors
  - Λ categorical (no ordinal) -> classification
  - Λ ordinal (either discrete or continuous) -> regression

Give a few examples of

Regression problems on images

•
•
•
•
•

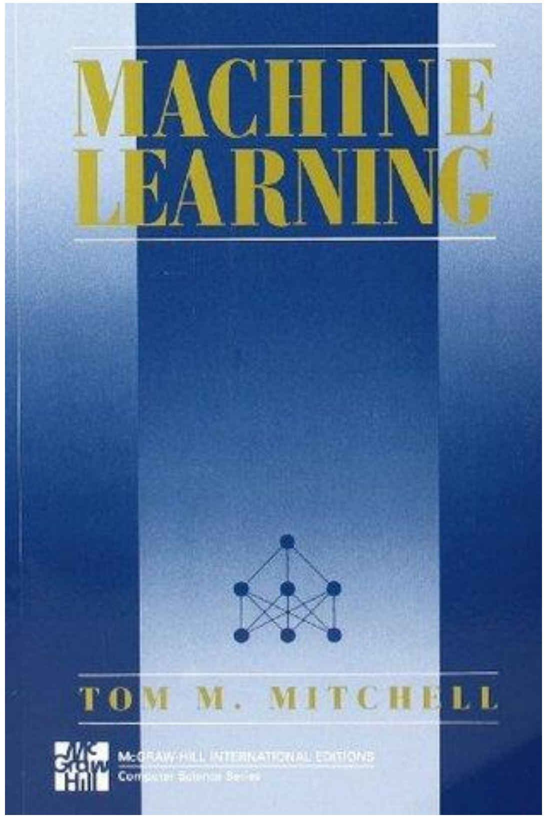
Classification problems on images

•
•
•
•
•

Machine Learning (Tom Mitchell – 1997)

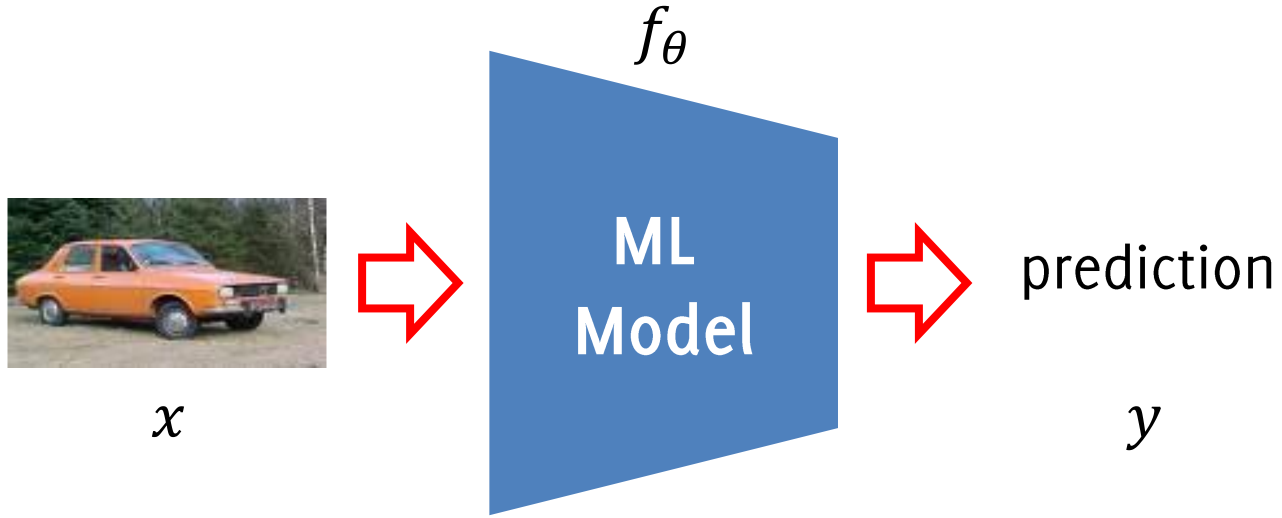
T = Regression/Classification/…
E = Training Data
P = Errors/Loss

*"A **computer program** is said to **learn from experience E** with respect to some class of **task T** and a **performance measure P**, if its performance at tasks in T, as measured by P, improves because of experience E."*



Machine Learning Paradigms

ML is the solution! Here the C program **is replaced by a very big parameteric function** f_θ, whose paramters θ are learned from data!



Supervised Learning

- Classification
- Regression

Unsupervised Learning

- Clustering
- Anomaly Detection
- …

$x \Rightarrow f_\theta(x) = y$

Unupervised Learning

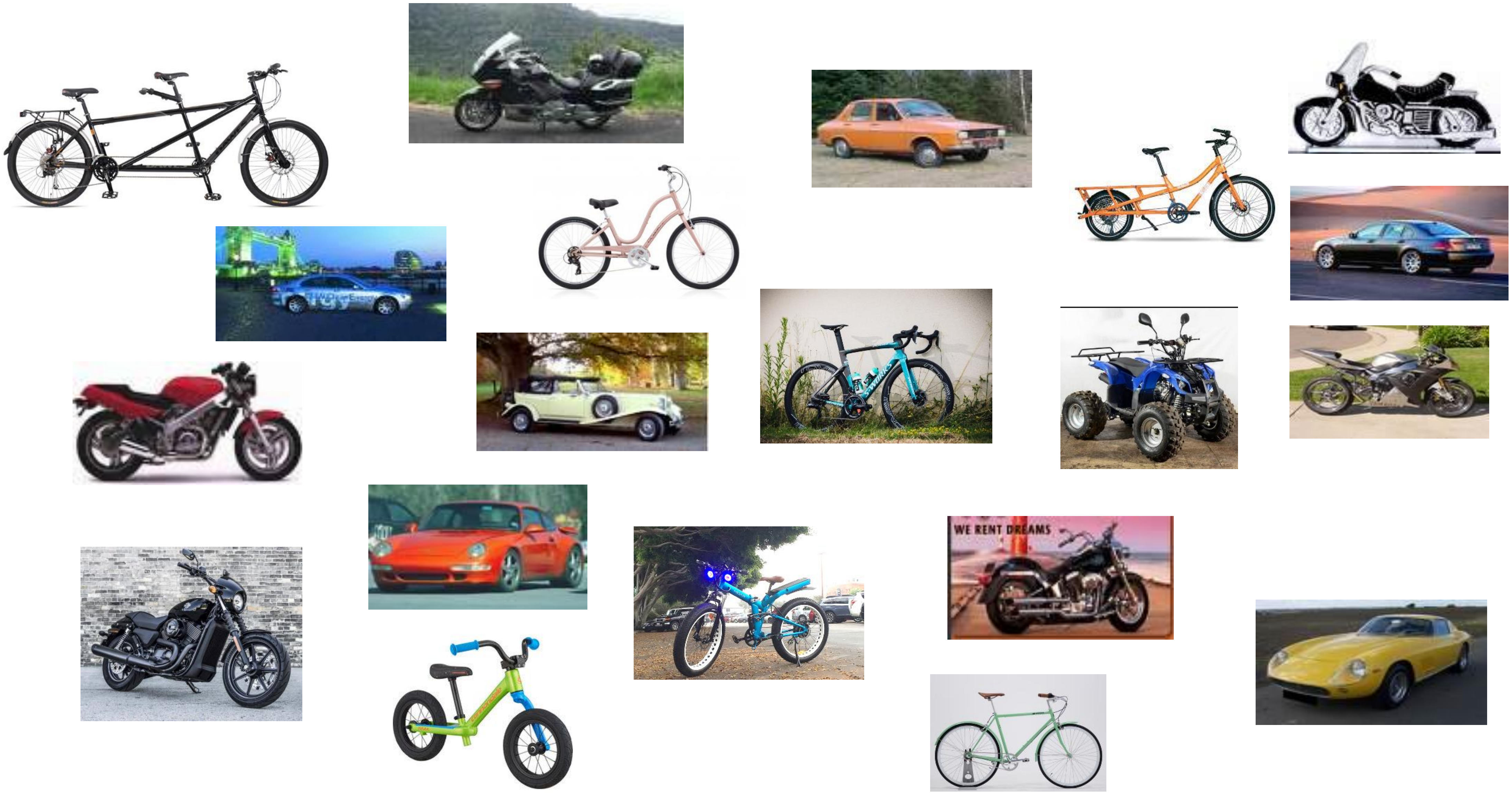
In **Unsupervised Learning**, the training set contains only inputs,

$$TR = \{x_1, \dots, x_n\}$$

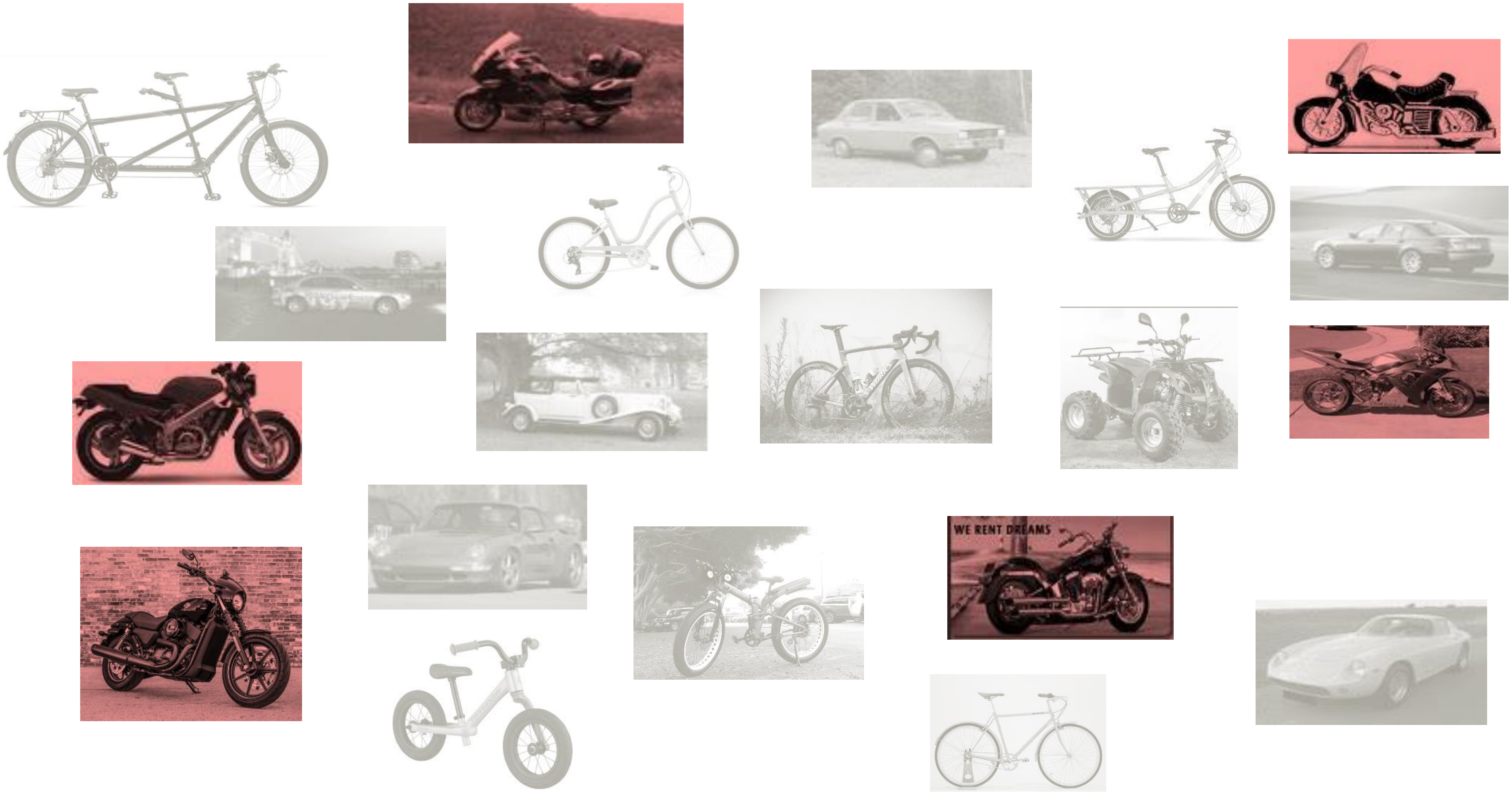
and the goal is to find structure in the data, like

- grouping or clustering of data according to *their similarity*
- estimating probability density distribution
- detecting outliers
- …

Unsupervised learning: Clustering

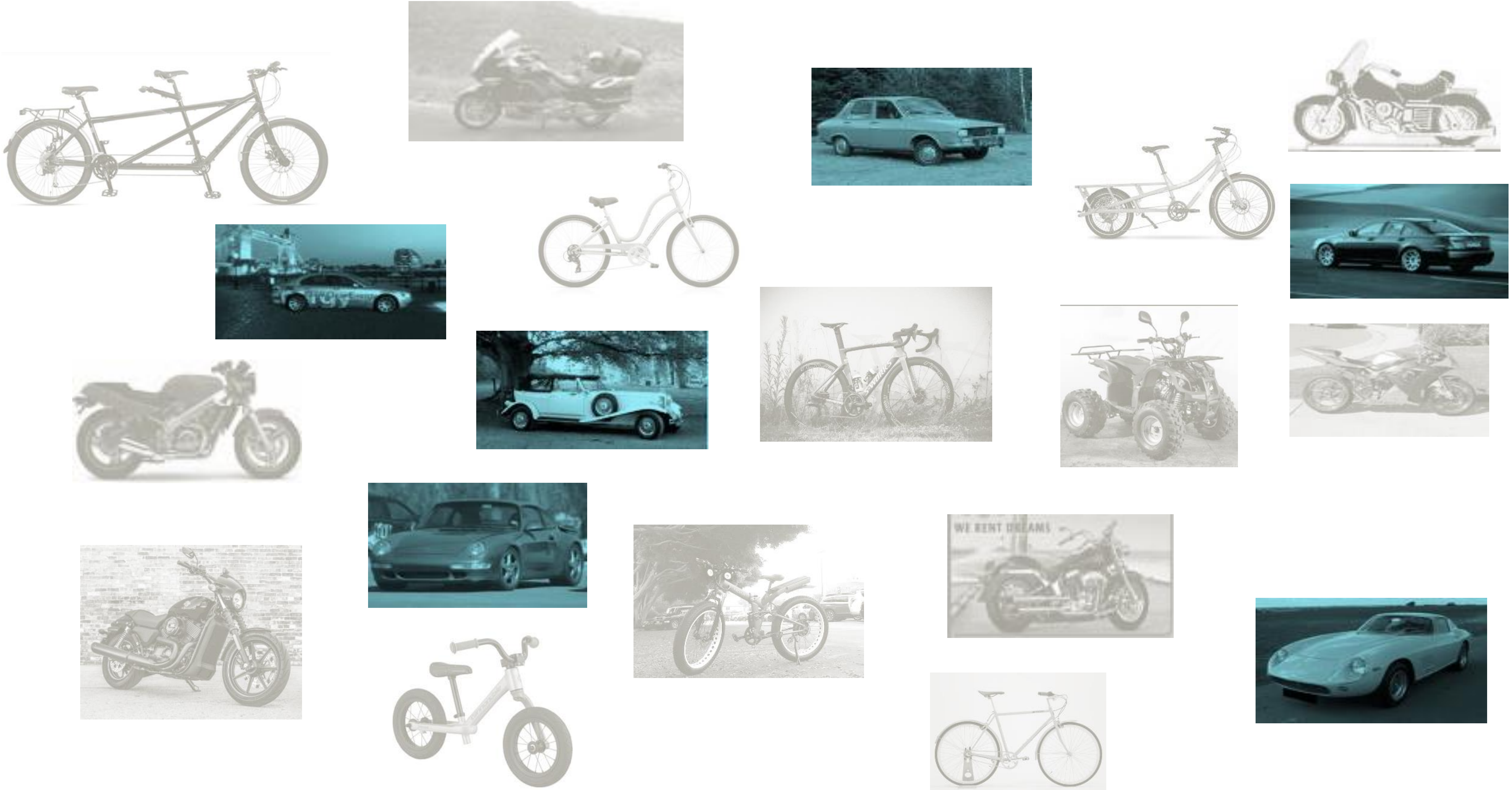


Unsupervised learning: Clustering

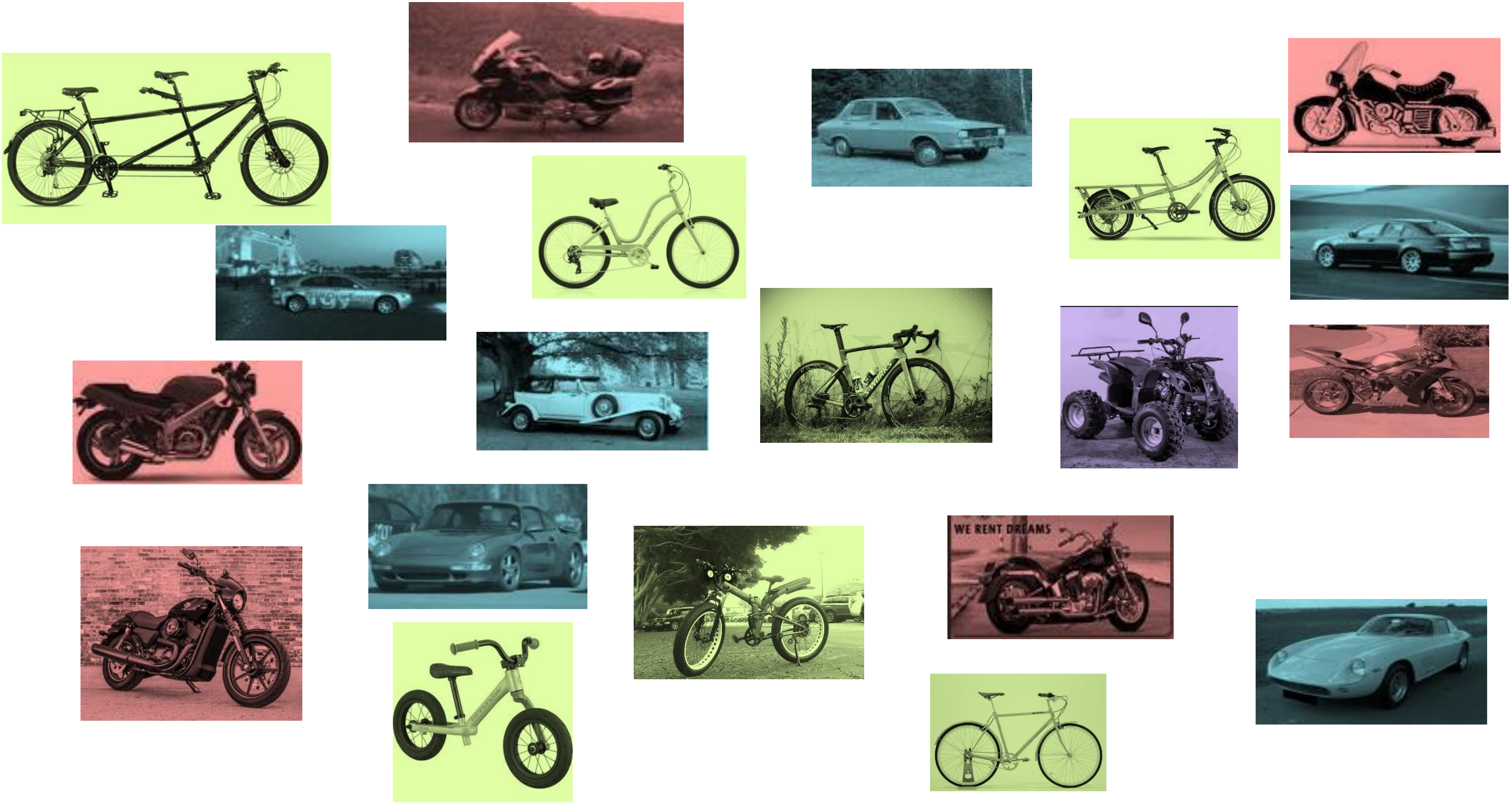


Unsupervised learning: Clustering

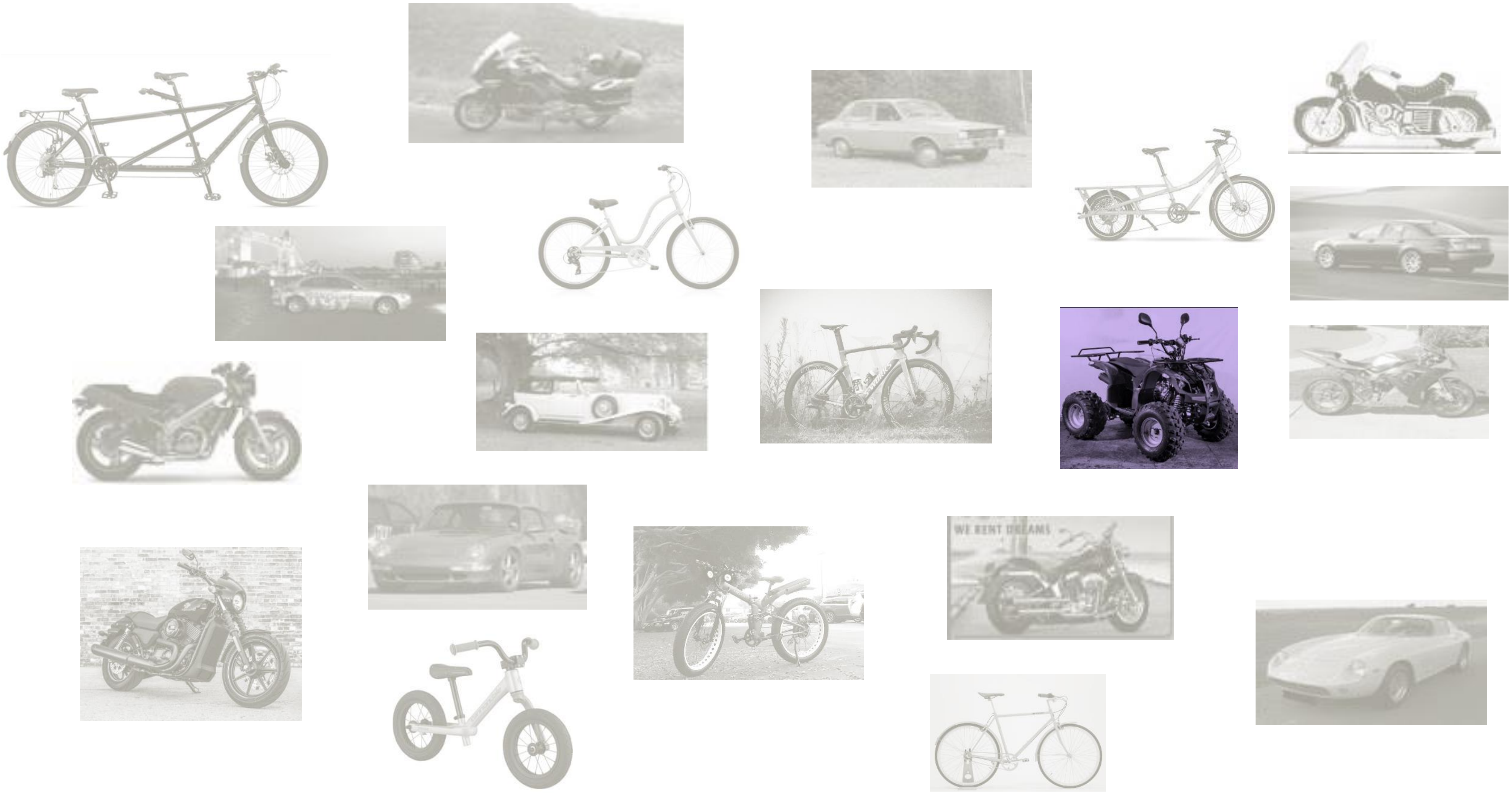
Unsupervised learning: Clustering



Unsupervised learning: Clustering



Unsupervised learning: Anomaly Detection



To Summarize: Machine Learning Paradigms

Immagine you have a certain experience E, i.e., data, and let's name it

$$D = x_1, x_2, x_3, \dots, x_N$$

- ***Supervised learning***: given a training set of pairs (input, desired output) $\{(x_1, y_1), \dots, (x_N, y_N)\}$, learn to produce the correct output for new inputs
- ***Unsupervised learning***: exploit regularities in D to build a meaningful/compact representation, to group, estimate densities, detect outliers…
- ***Reinforcement learning***: a different context where an agent is producing actions $a_1, a_2, a_3, \dots, a_N$, which affect the environment, and receiving rewards $r_1, r_2, r_3, \dots, r_N$. Learning how the agent should act in order to maximize rewards in the long term.

To Summarize: Machine Learning Paradigms

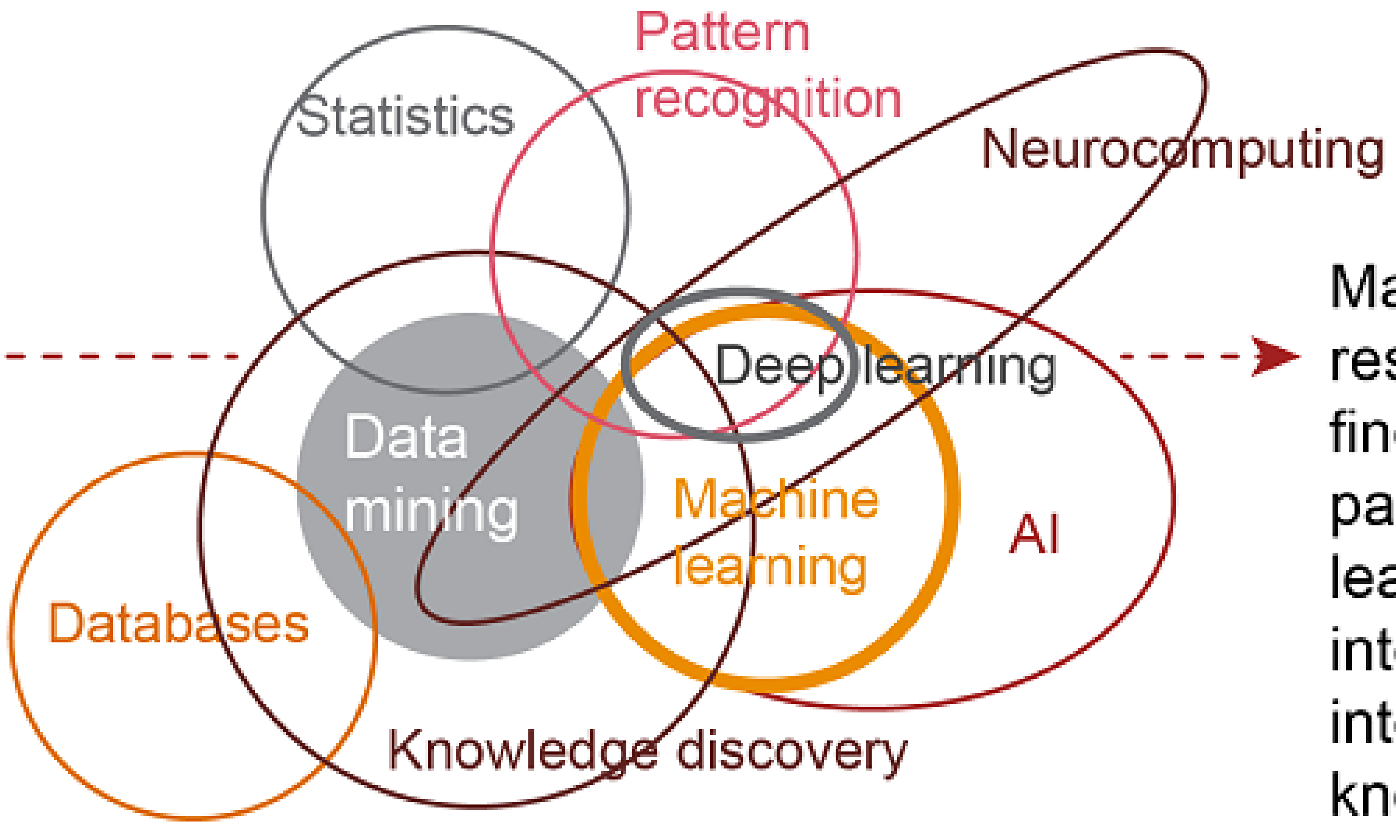
Immagine you have a certain experience E, i.e., data, and let's name it

$$D = x_1, x_2, x_3, \dots, x_N$$

- ***Supervised learning***: given a training set of pairs (input, desired output) $\{(x_1, y_1), \dots, (x_N, y_N)\}$, learn to produce the correct output for new inputs
- ***Unsupervised learning***: exploit regularities in D to build a meaningful/compact representation, to group, estimate densities, detect outliers…
- ***Reinforcement learning***: a different context where … $a_1, a_2, a_3, \dots, a_N$, which affect the environme… $r_1, r_2, r_3, \dots, r_N$. Learning how the agent shou… in the long term.

This course focuses most on Supervised Learning (with some unsupervised spots)

Machine Learning



Machine learning is a category of research and algorithms focused on finding patterns in data and using those patterns to make predictions. Machine learning falls within the artificial intelligence (AI) umbrella, which in turn intersects with the broader field of knowledge discovery and data mining.

Source: SAS, 2014 and PwC, 2016

Hand-Crafted Features

How images / signals were classified before deep learning

Assume you need to automatize this process

Assume you need to automatize this process

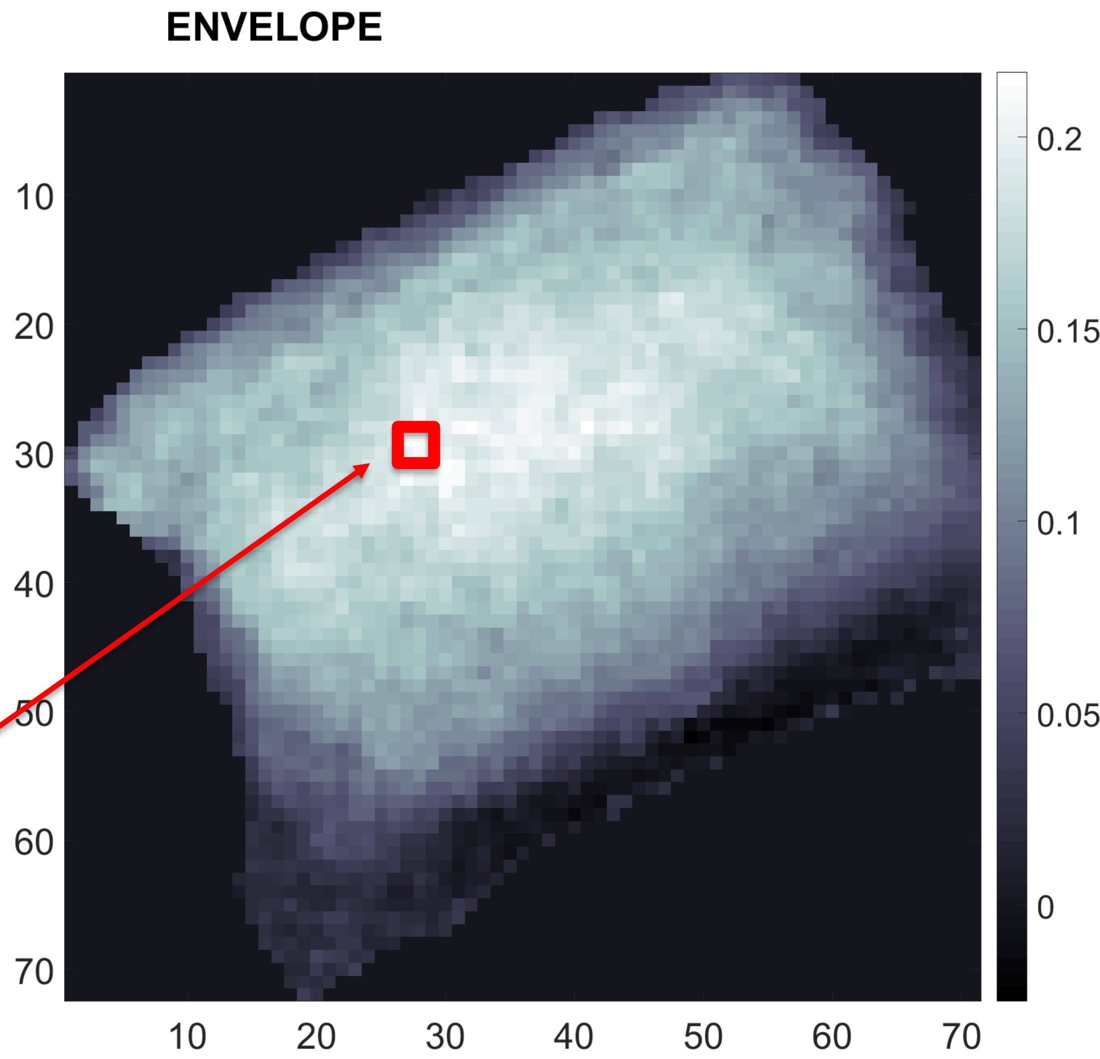
Assume you need to automatize this process

An Illustrative Example: Parcel Classification

Images acquired from a RGB-D sensor:

- No color information provided
- A few pixels report depth measurements
- Images of 3 classes
  - ENVELOPE
  - PARCEL
  - DOUBLE

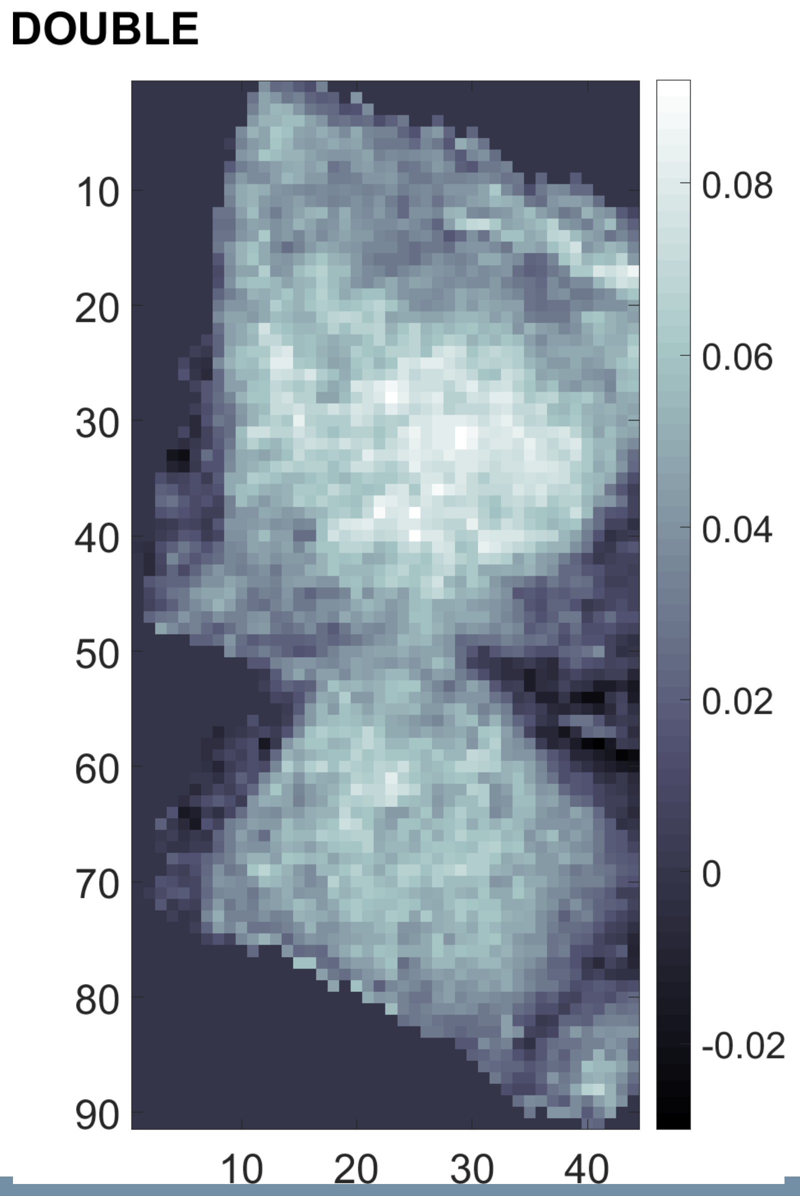
Envelop height at that pixel



An Illustrative Example: Parcel Classification

Images acquired from a RGB-D sensor:

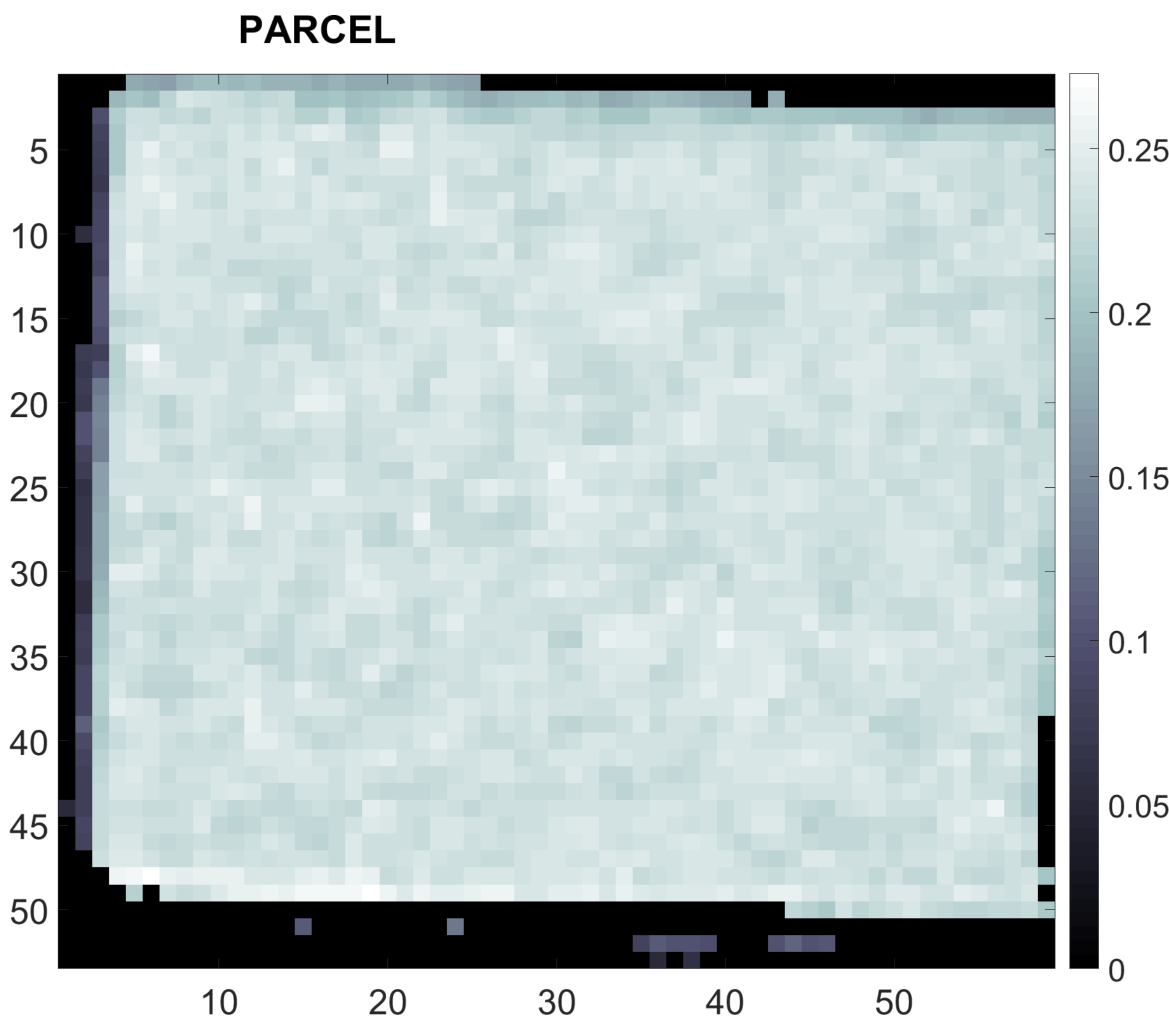
- No color information provided
- A few pixels report depth measurements
- Images of 3 classes
  - ENVELOPE
  - PARCEL
  - DOUBLE



An Illustrative Example: Parcel Classification

Images acquired from a RGB-D sensor:

- No color information provided
- A few pixels report depth measurements
- Images of 3 classes
    - ENVELOPE
    - PARCEL
    - DOUBLE



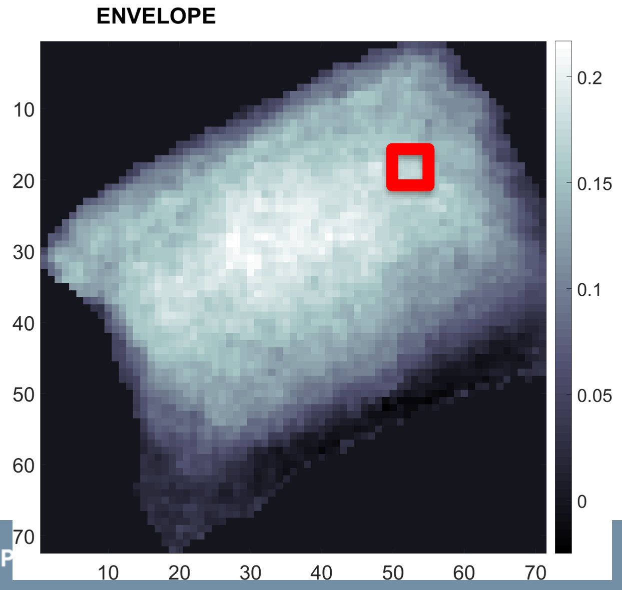
An Illustrative Example: Parcel Classification

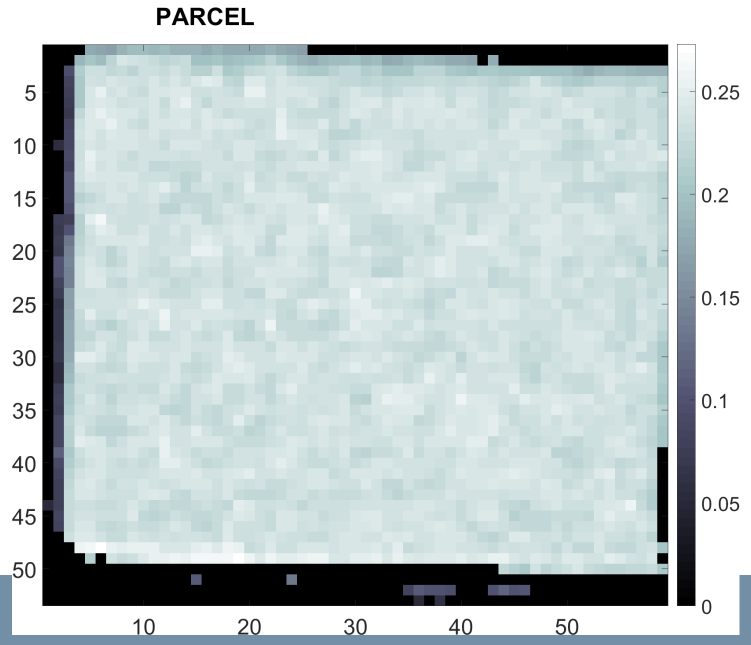
Images acquired from an RGB-D sensor:

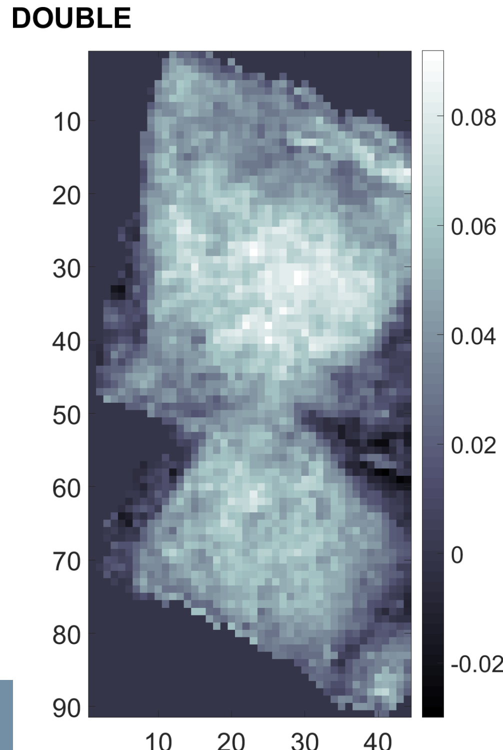
No color information provided

Images of 3 classes

- ENVELOPE
- PARCEL
- DOUBLE



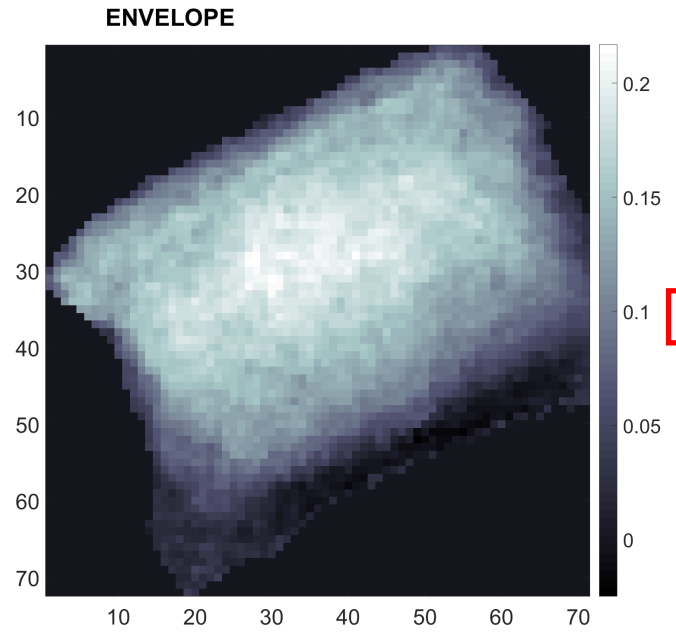


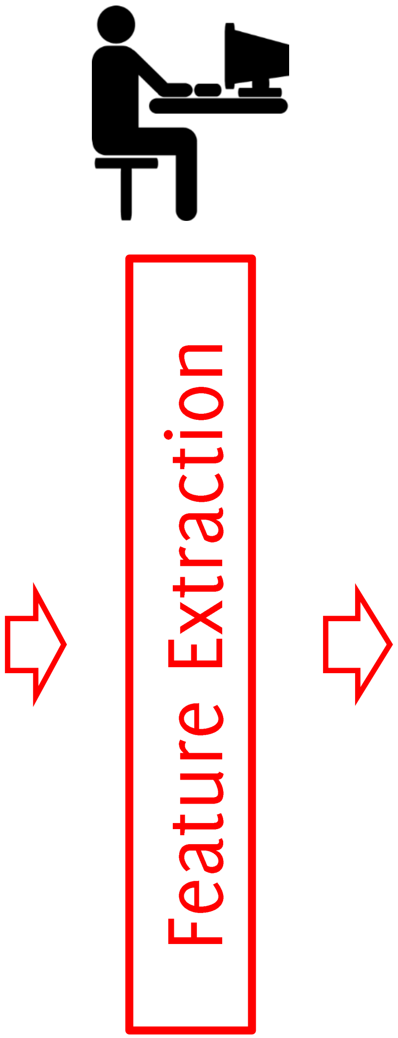


Hand Crafted Featues

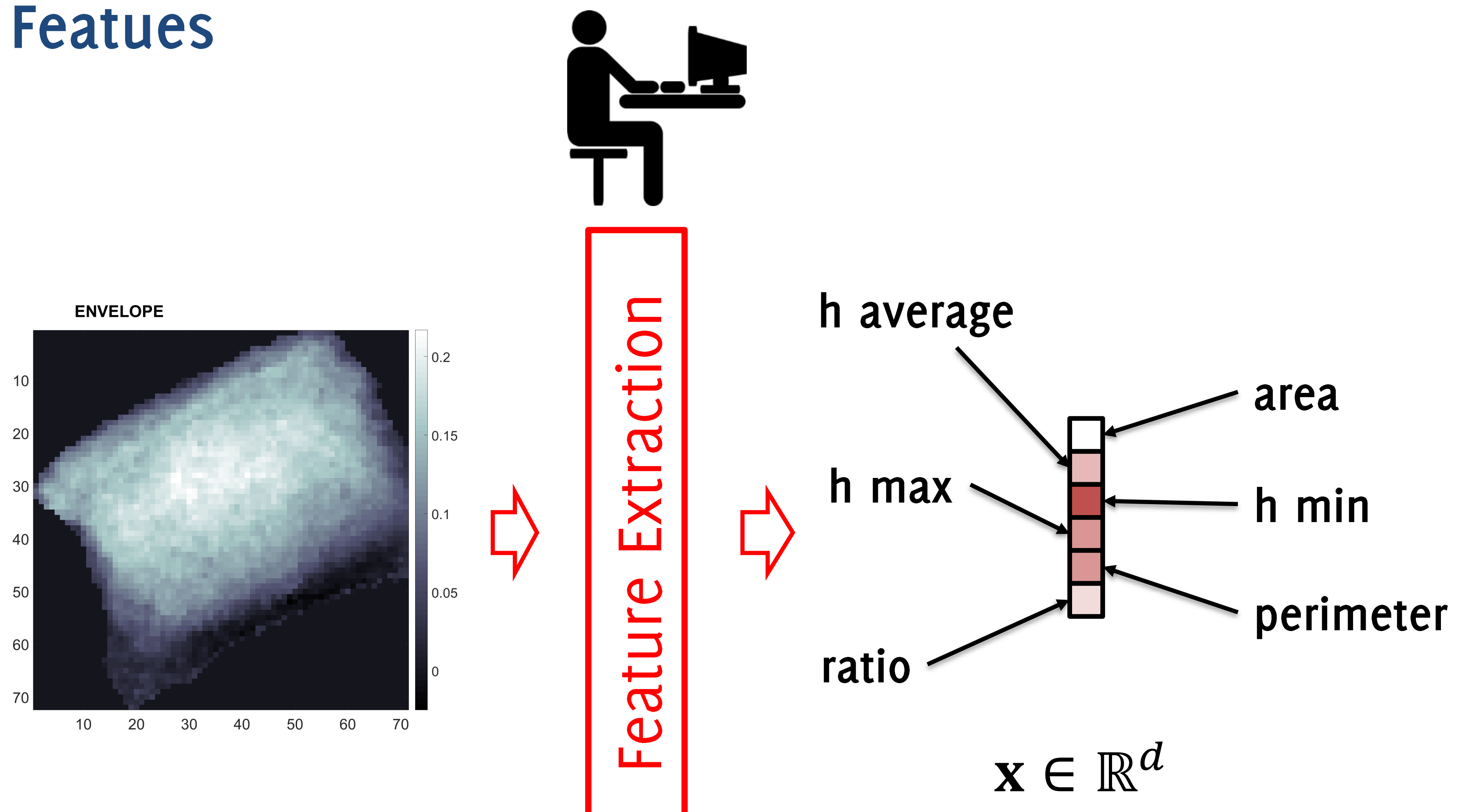
Engineers:

- know what's meaningful in an image (e.g. a specific color/shape, the area, the size)
- can implement algorithms to map this information in a set of measurements, a **feature vector**





Hand Crafted Featues



Here you get to «tabular data» which can be traditionally handled by Machine Learning models.

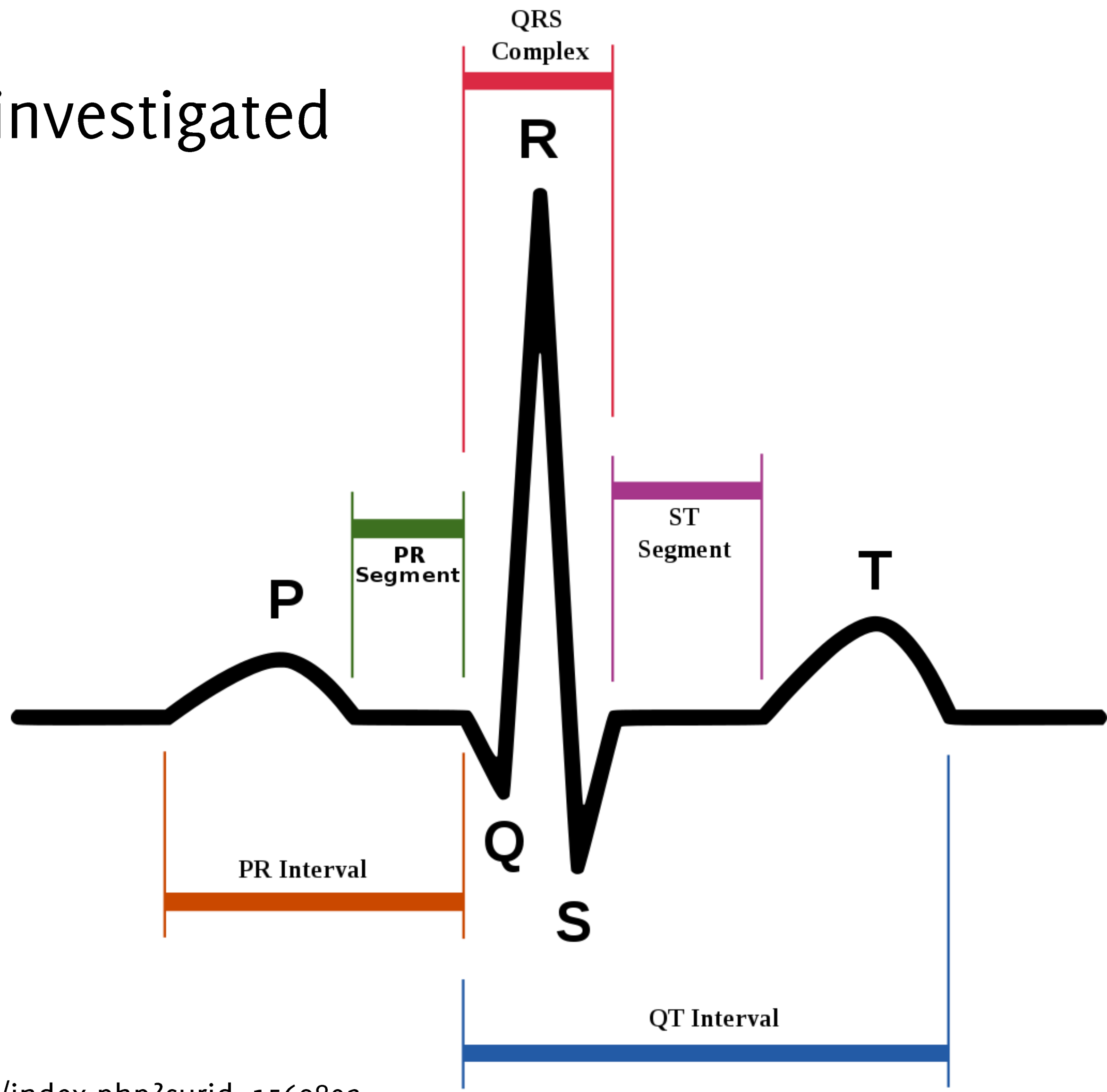
This is exactly what a doctor would to to classify ECG tracings

Heartbeats morphology has been widely investigated

Doctors know which patterns are meaningful for classifying each beat

Features are extracted from landmarks indicated by doctors:

e.g. QT distance, RR distance…



Created by Agateller (Anthony Atkielski), Public Domain, https://commons.wikimedia.org/w/index.php?curid=1560893

The Training Set

The training set is a set of annotated examples

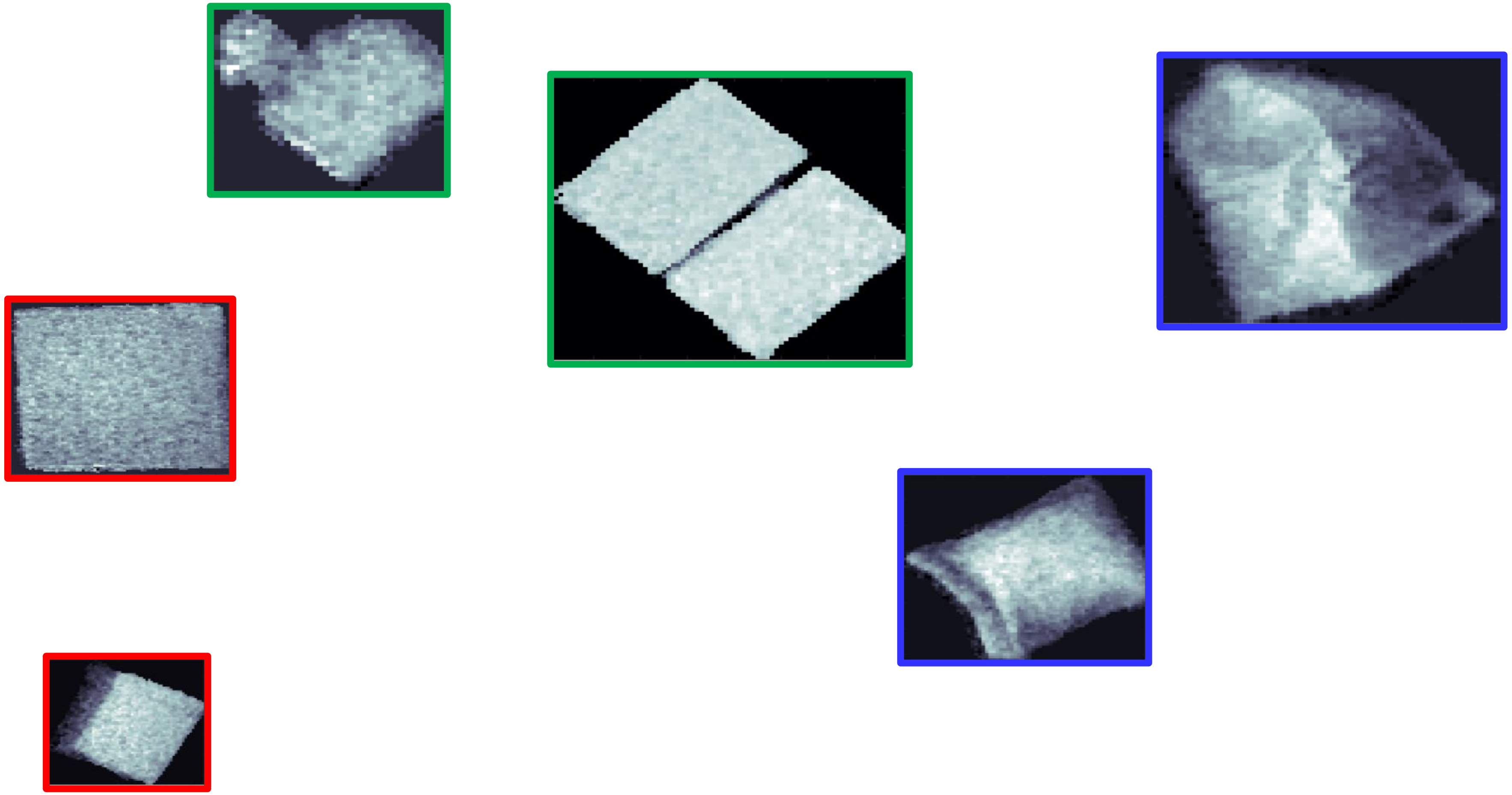
$$TR = \{(\boldsymbol{x}, \boldsymbol{y})_i, i = 1, \dots, N\}$$

Each couple $(\boldsymbol{x}, \boldsymbol{y})_i$ corresponds to:

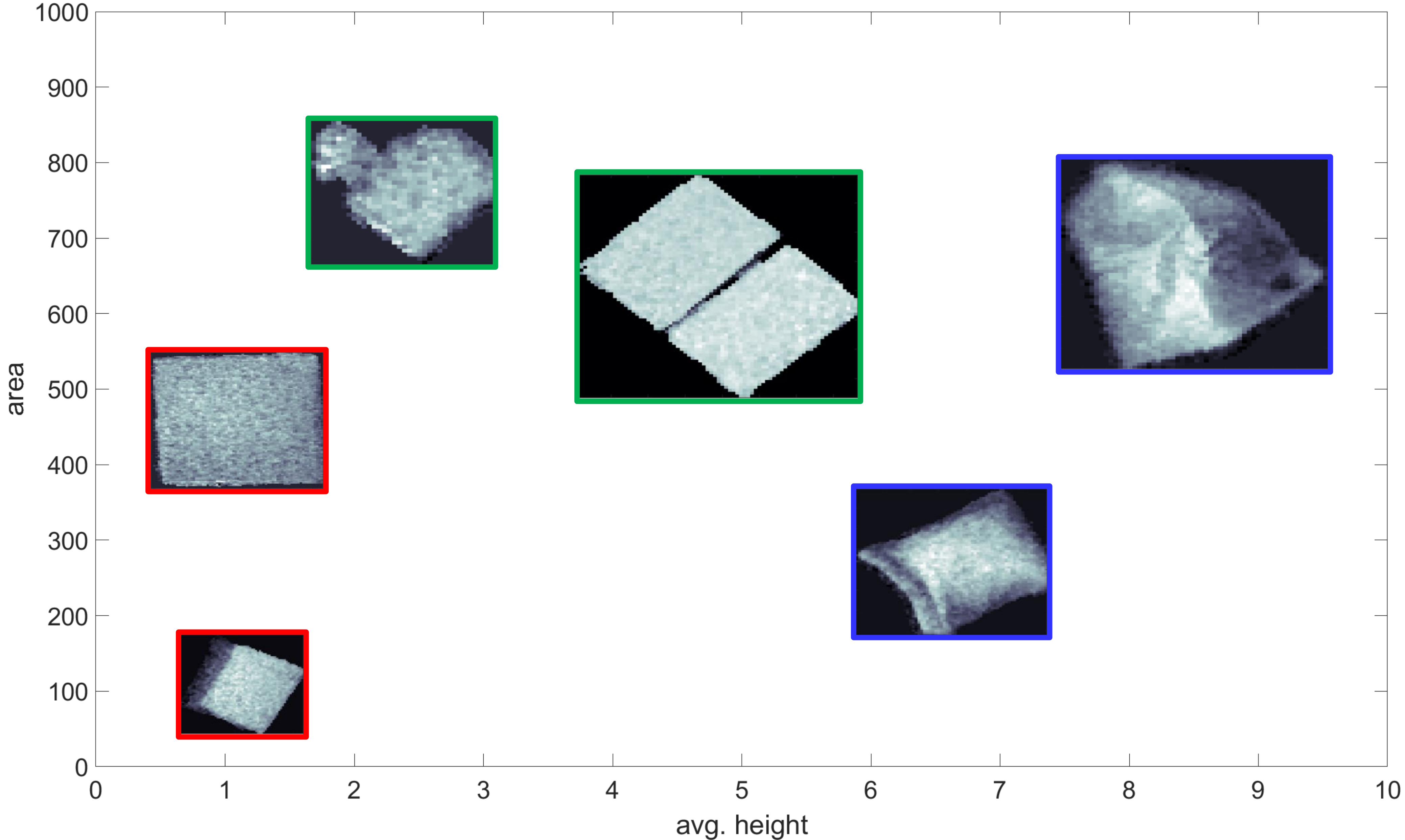
- an image $\boldsymbol{x}_i$
- the corresponding label $\boldsymbol{y}_i$

This is meant for a **Supervised** Learning Problem!

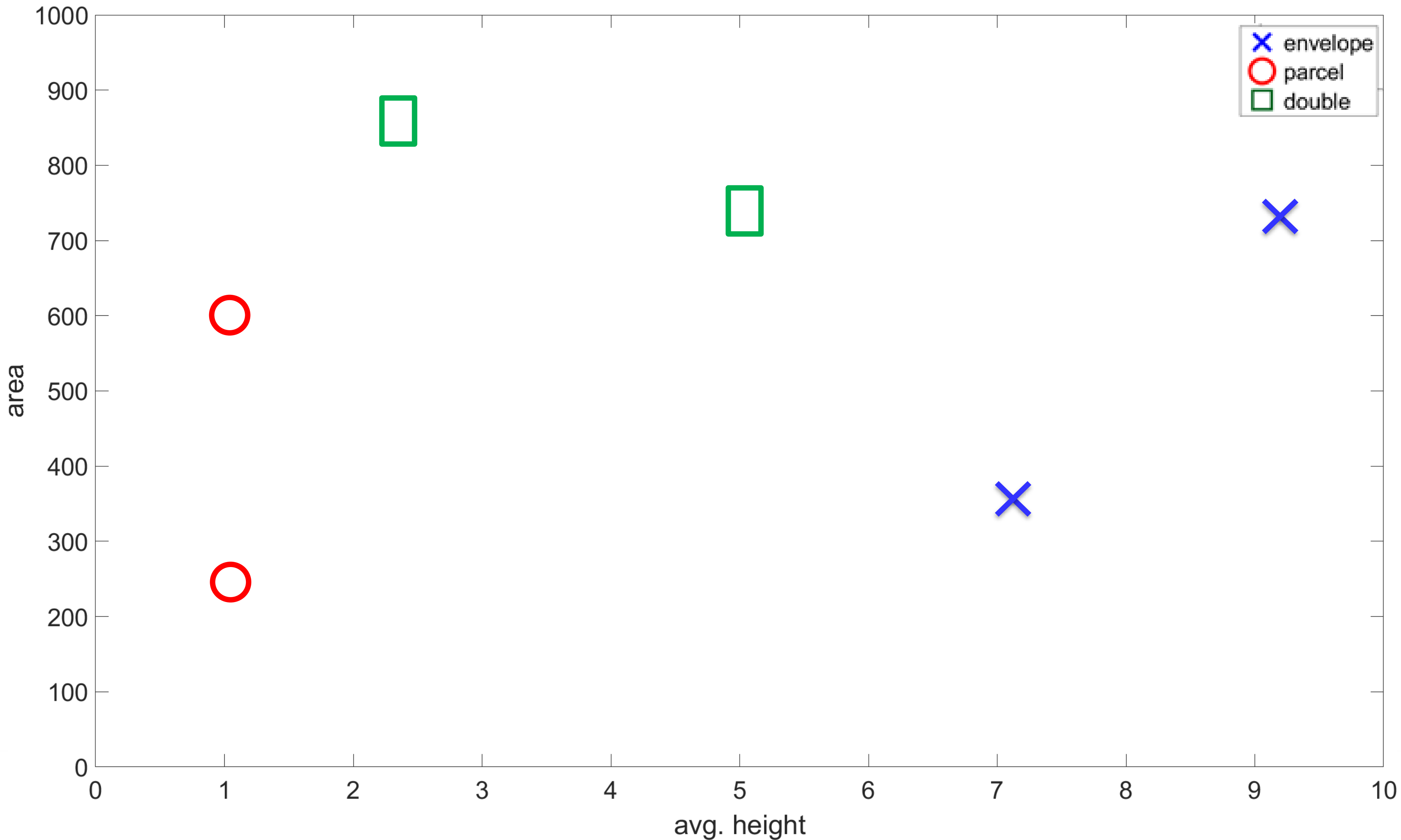
The Training Set: images + labels



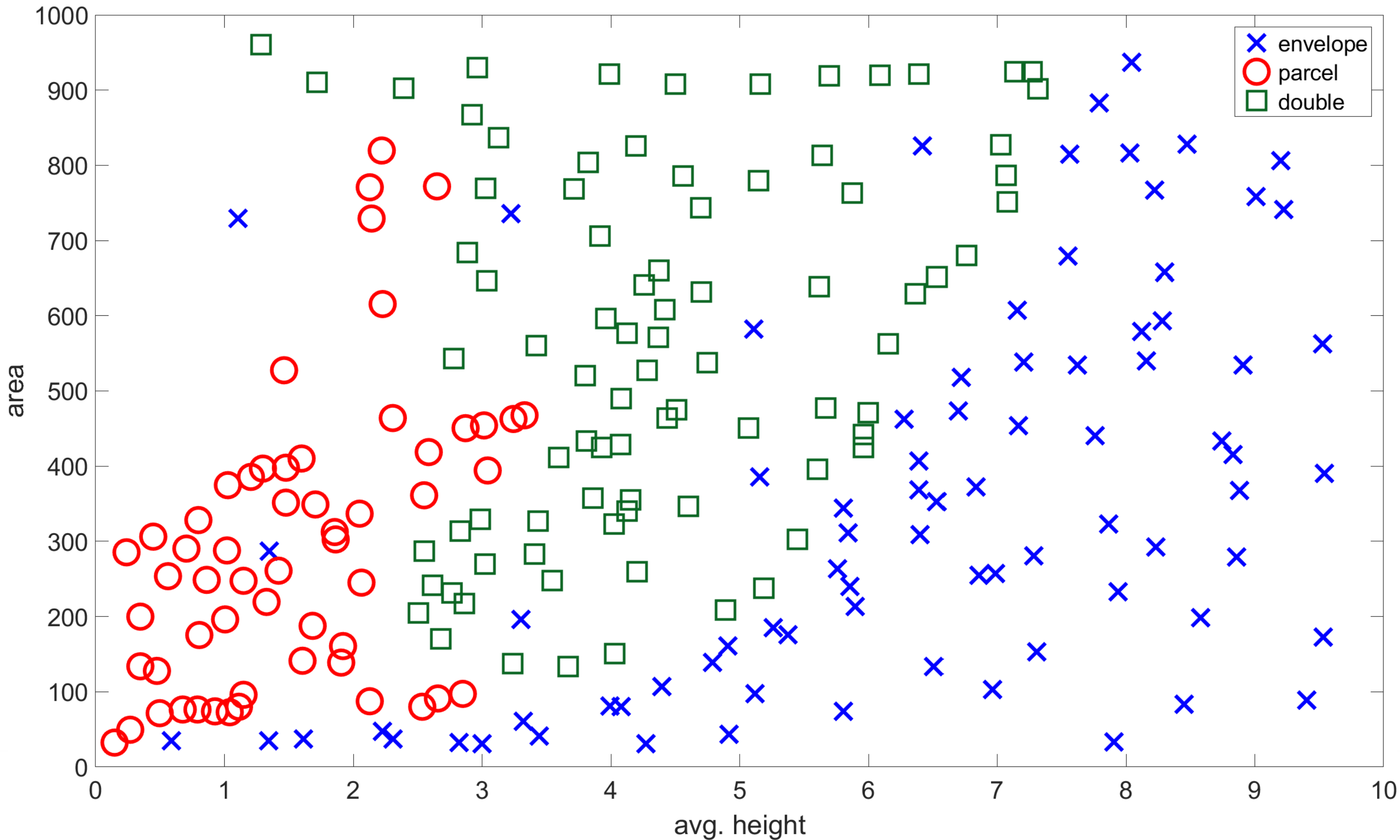
The Training Set: images + labels



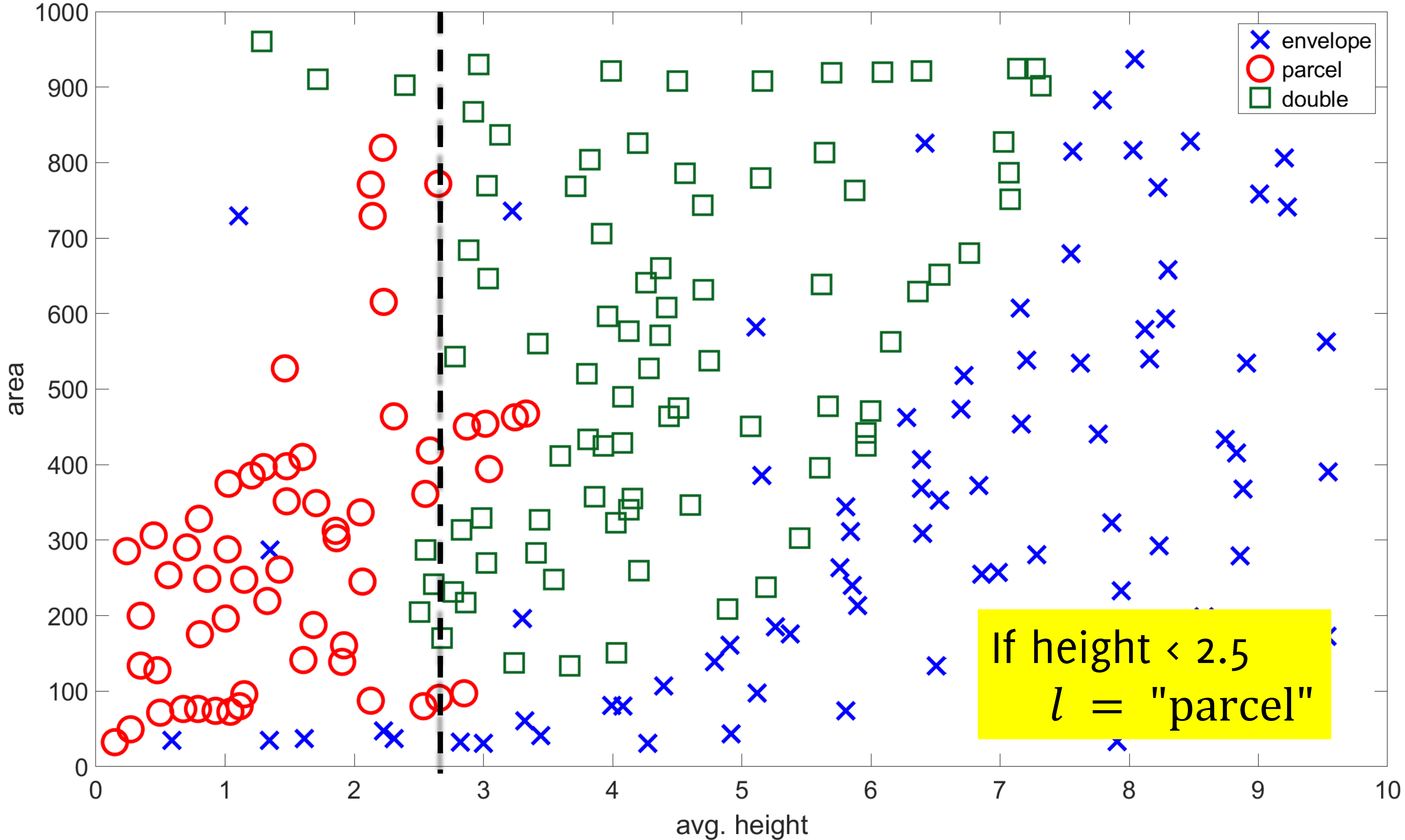
The Training Set: features + labels



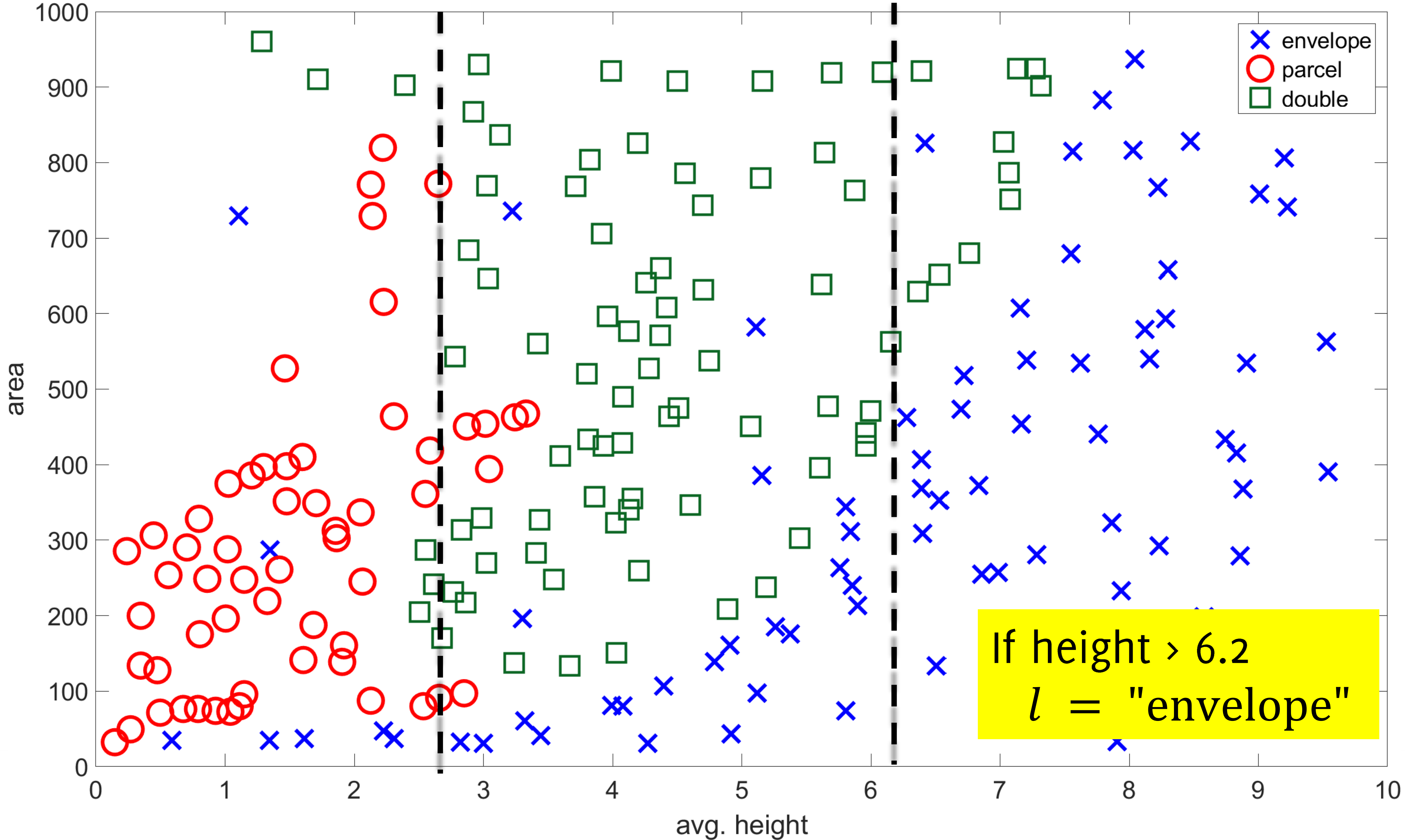
The Training Set



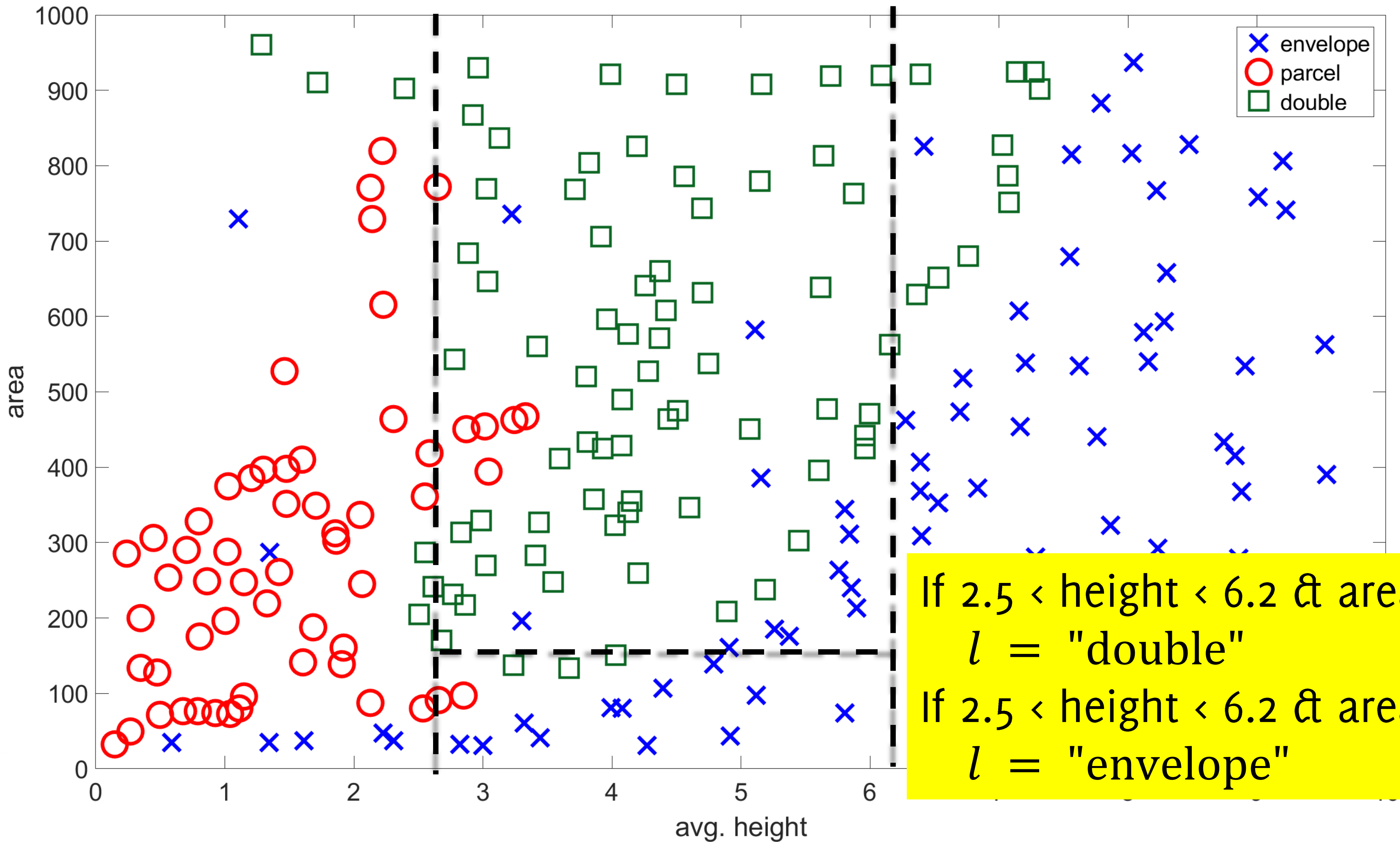
Training Set



Training Set



Training Set



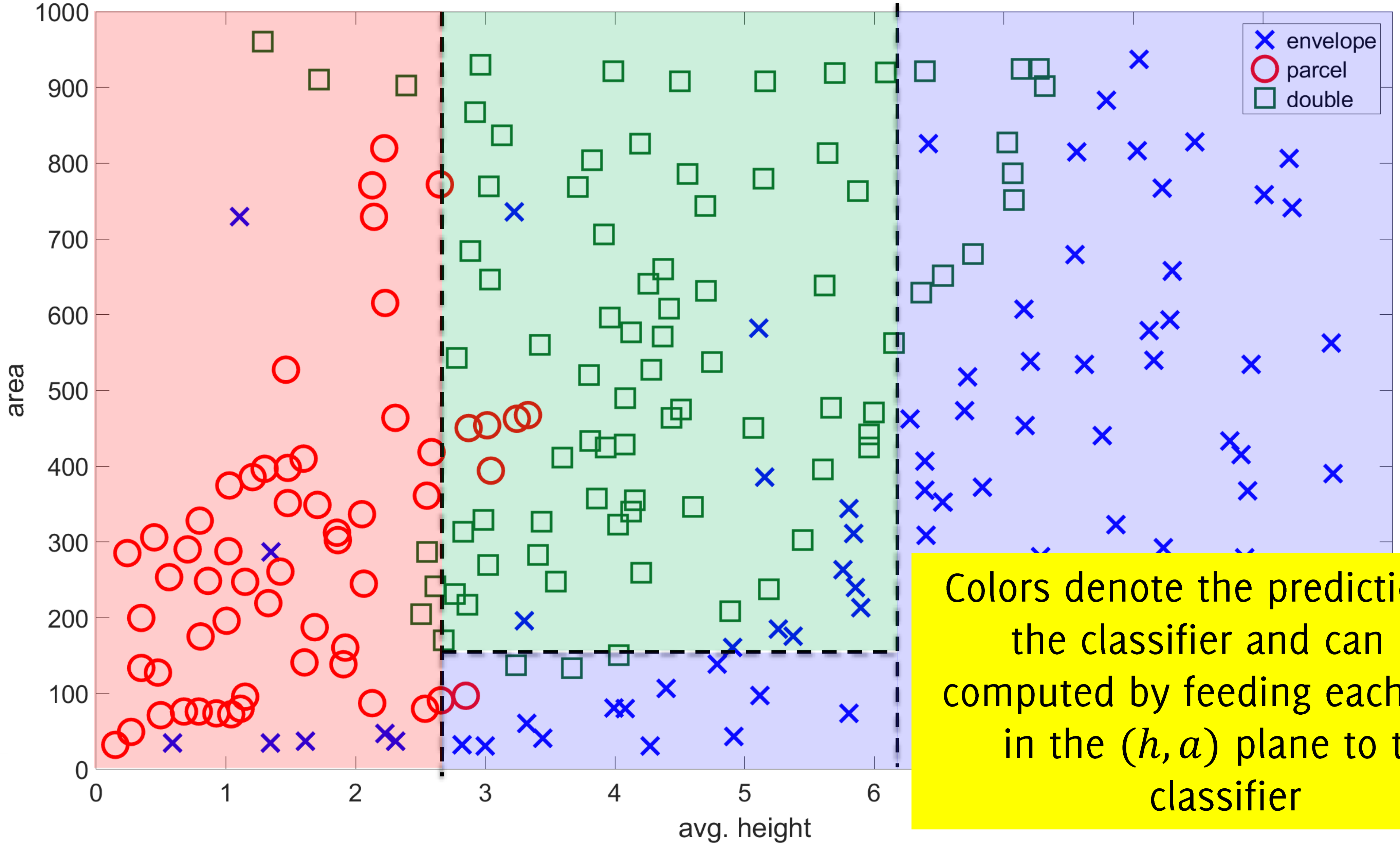
If $2.5 < \text{height} < 6.2$ & $\text{area} > 200$

l = "double"

If $2.5 < \text{height} < 6.2$ & $\text{area} < 200$

l = "envelope"

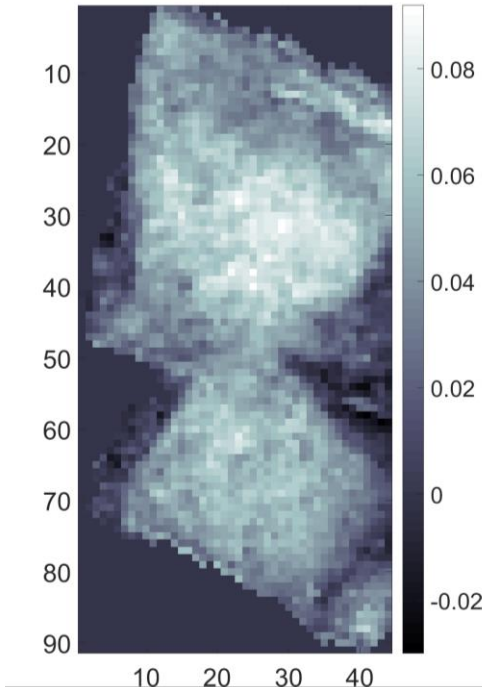
Classifier Output



Colors denote the predictions of the classifier and can be computed by feeding each point in the (h, a) plane to the classifier

A tree classifying image features

Input image



$I_1 \in \mathbb{R}^{r_1 \times c_1}$

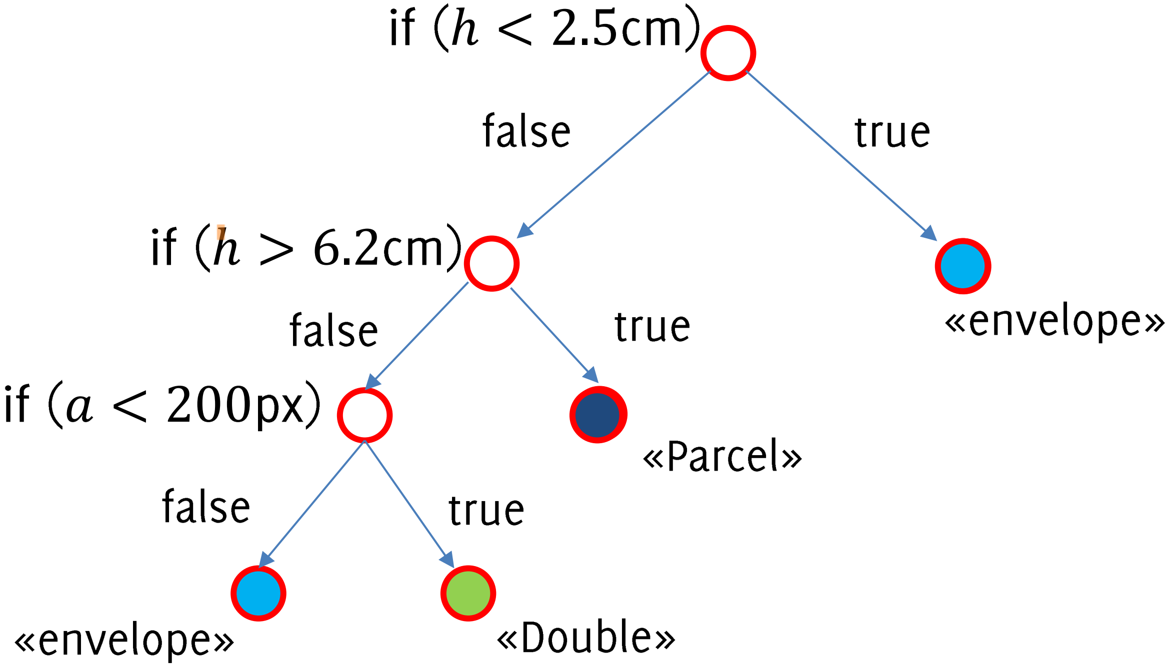
Feature Extraction Algorithm

$\mathbf{x} \in \mathbb{R}^2$

h
a

$\mathbf{x} \in \mathbb{R}^2$

h
a



"double" "envelope" "parcel"

Limitations of Rule Based Classifier

It is difficult to grasp what are meaningful dependencies over multiple variables (it is also impossible to visualize these)

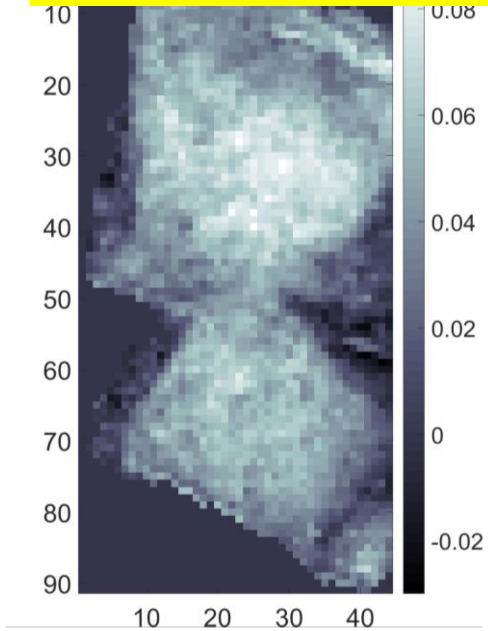
Let's resort to a **data-driven model** for the only task of separating feature vectors in different classes.

How can a classifier achieve better performance?

A tree classifying image features

The classifier has a few patameters θ:

- The **splitting criteria**
- The **splitting thresholds T_i**



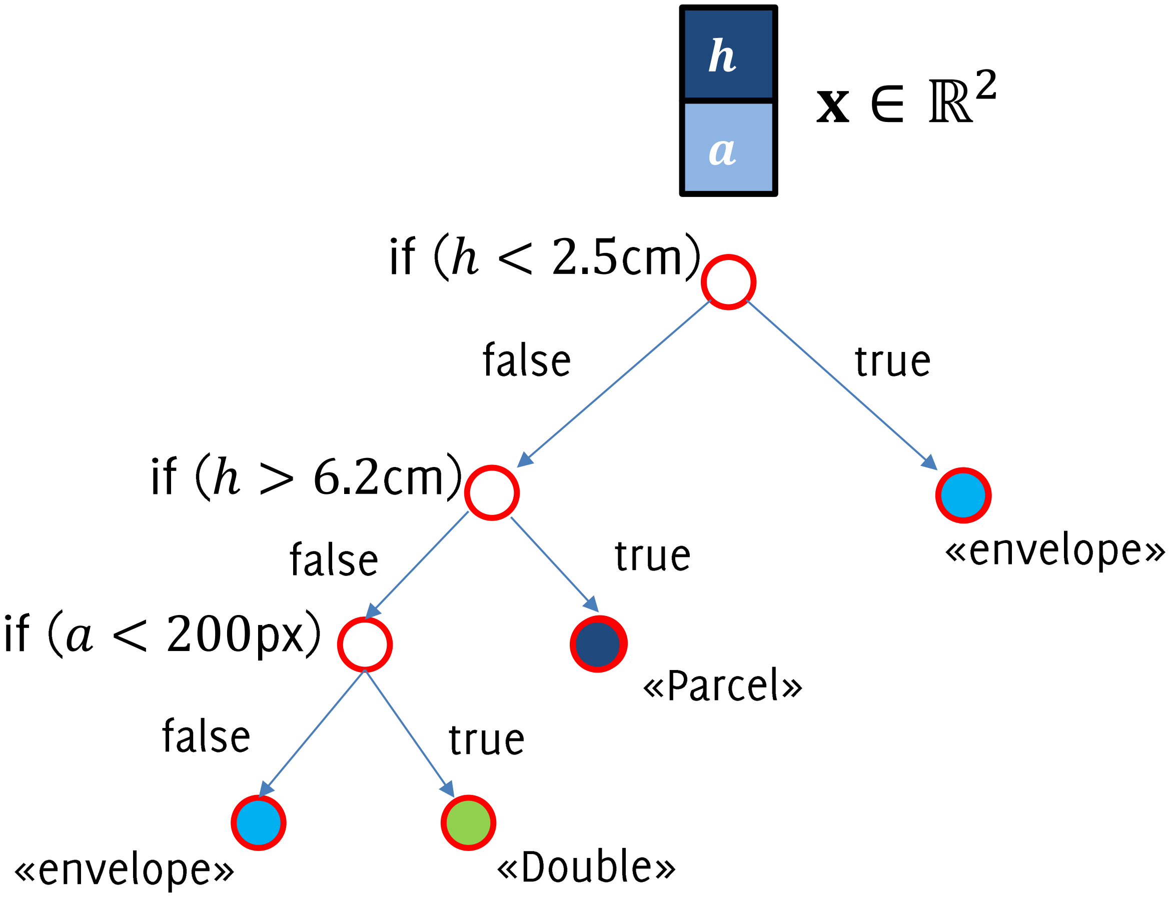
$I_1 \in \mathbb{R}^{r_1 \times c_1}$

Feature Extraction Algo

$\mathbf{x} \in \mathbb{R}^2$

h

a



"double" "envelope" "parcel"

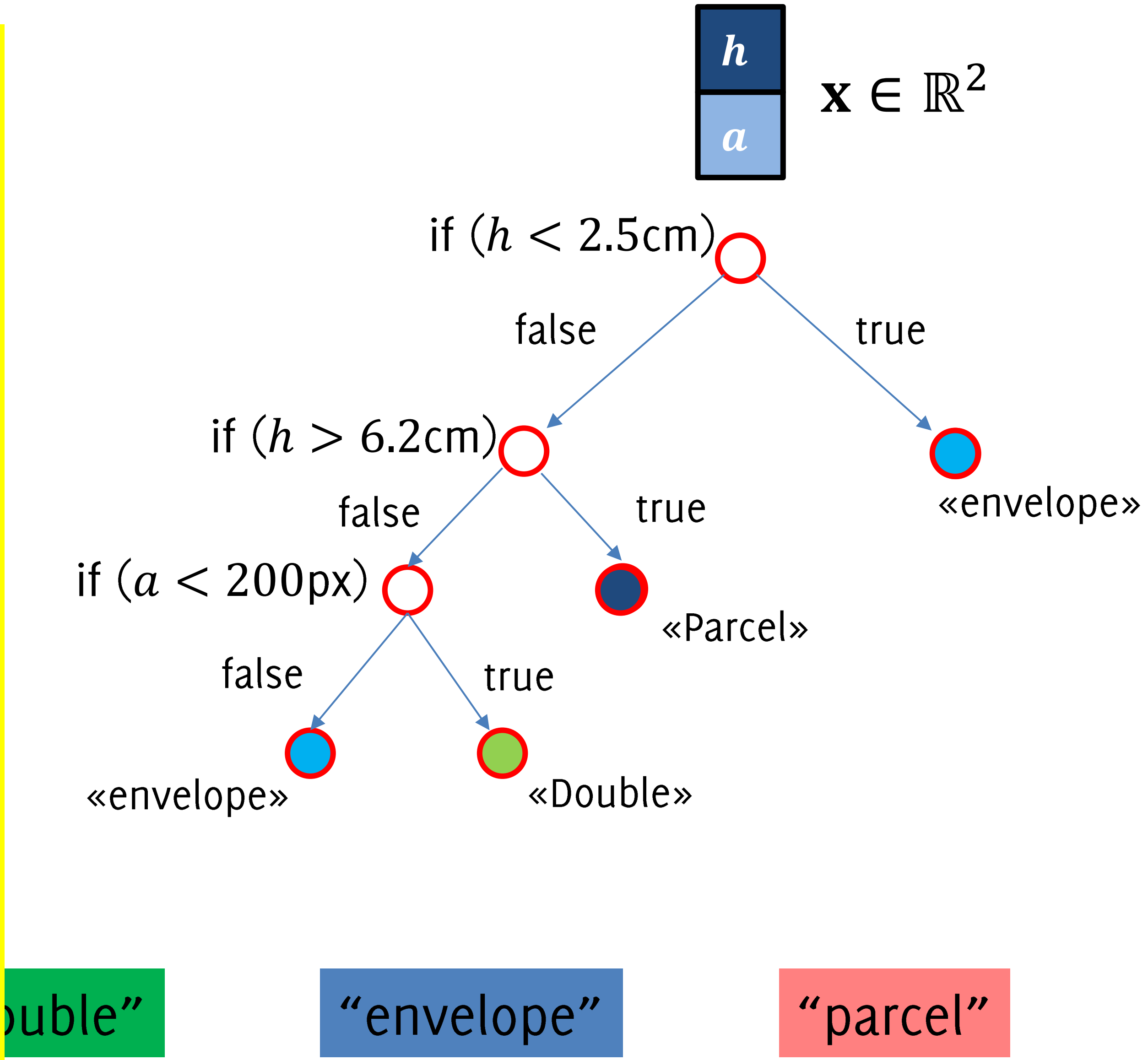
A tree classifying image features

The classifier has a few patameters θ:

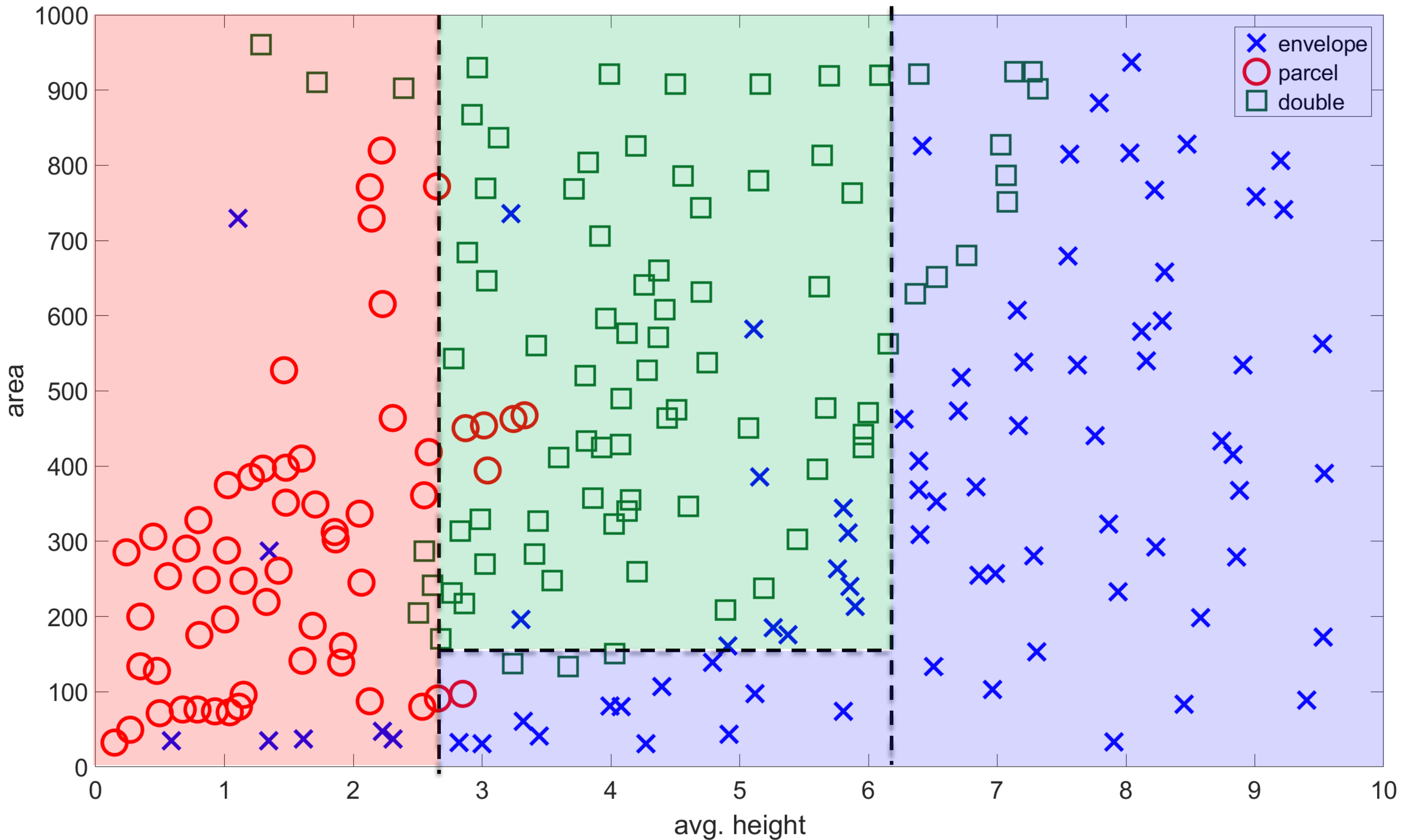
- The **splitting criteria**
- The **splitting thresholds** $\boldsymbol{T_i}$

Summarizing:

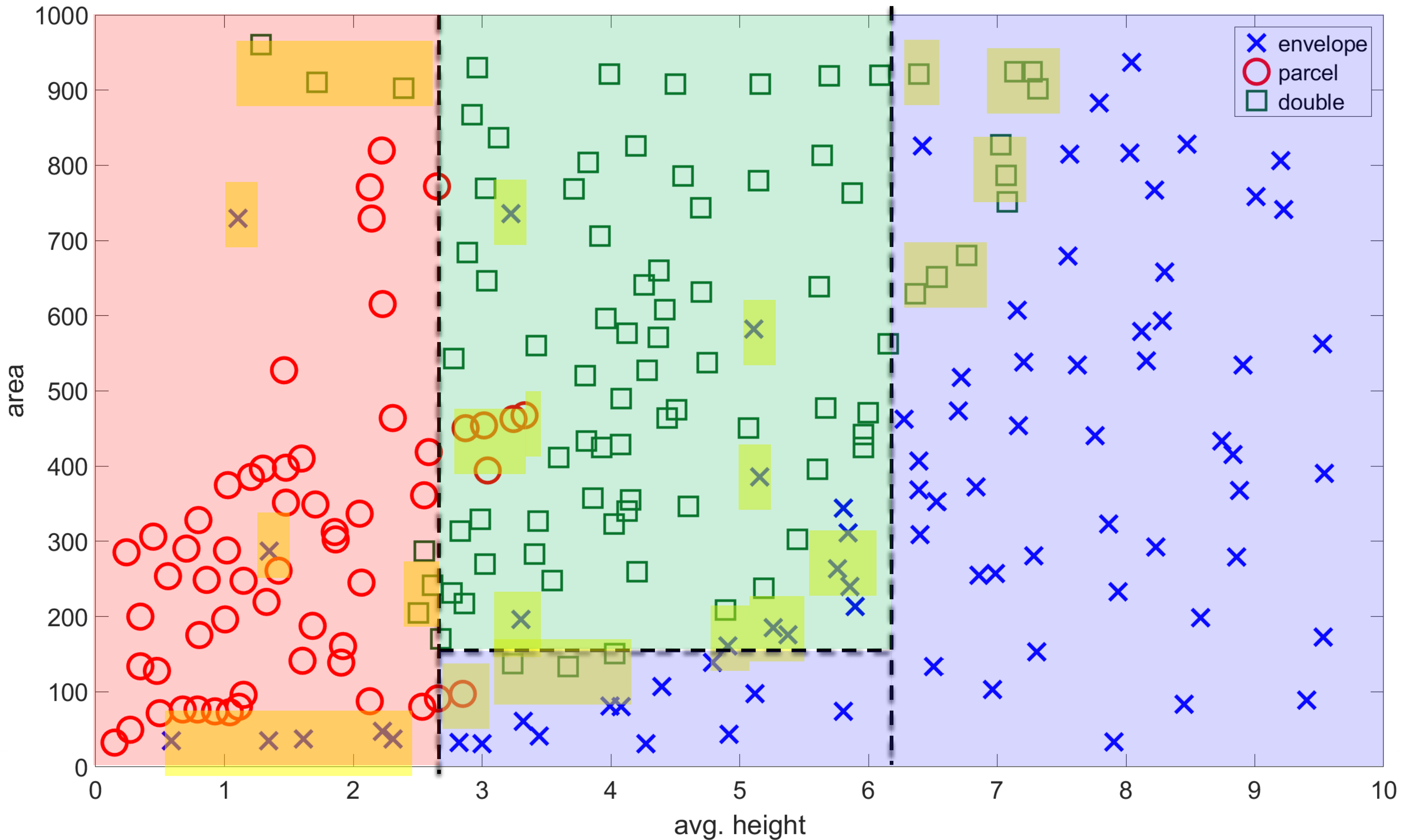
- **The model:** the (decision) tree with its own parameters θ
- **The task:** multi-class classification
- **The experience:** the training set
- **The performance:** the classification accuracy



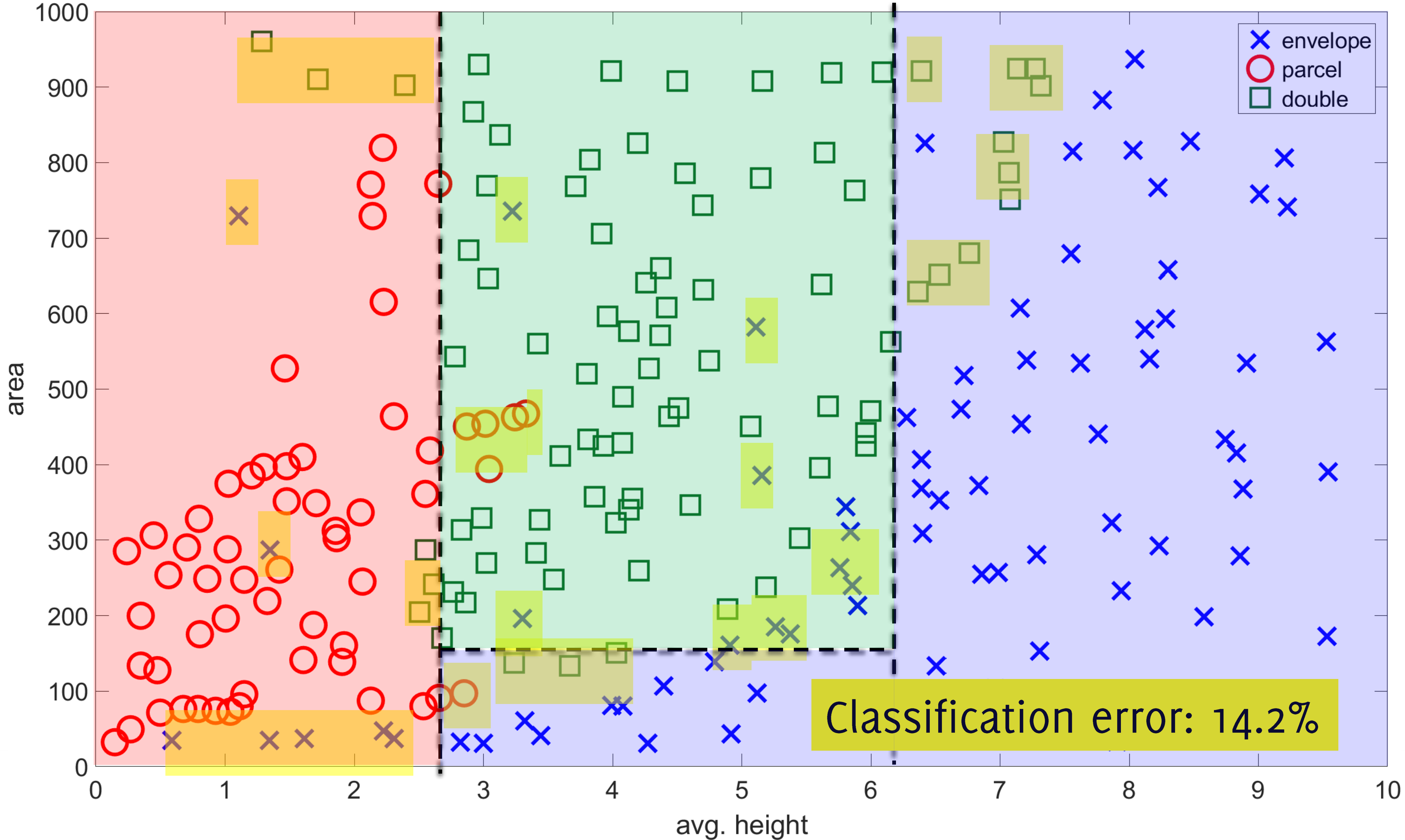
This is our first solution



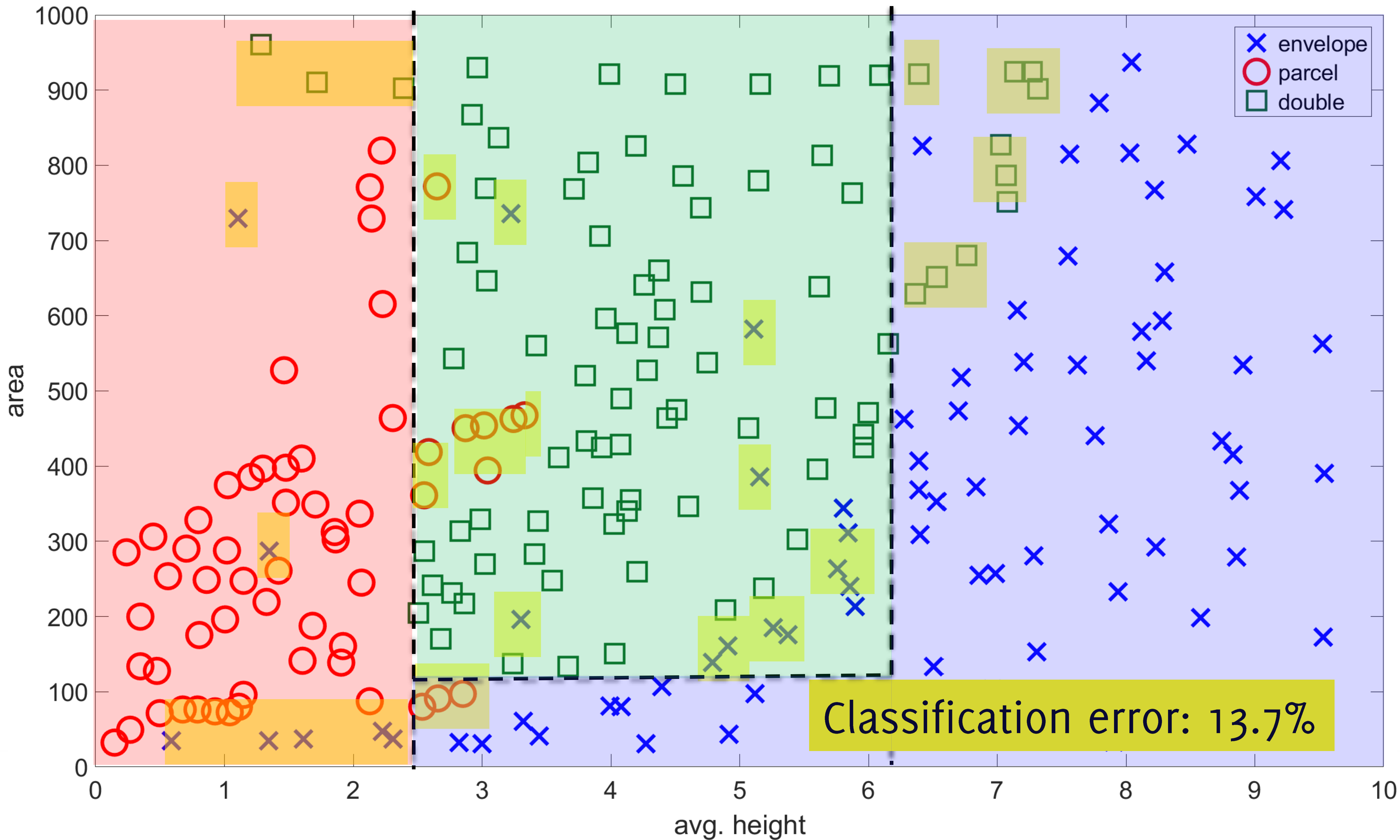
There are a few errors



Can I do better?



Let's try different parameters



Data Driven Models

Data Driven Models are defined from a training set of (supervised) pairs

$$TR = \{(\boldsymbol{x}, \boldsymbol{y})_i, i = 1, \dots, N\}$$

The model parameters θ (e.g. Neural Network weights) are set to minimize a **loss function** (e.g., the classification error in case of discrete output or the reconstruction error in case of continuous output)

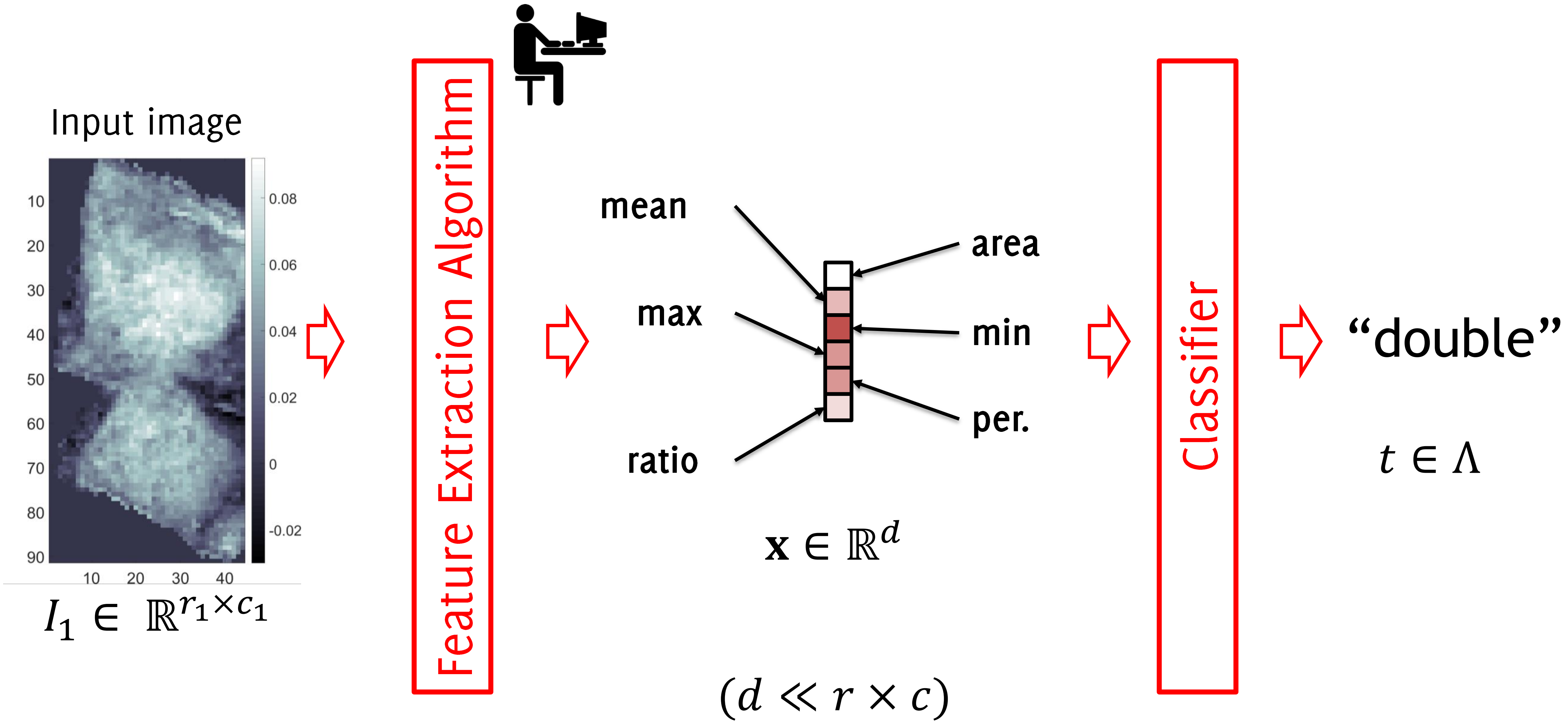
$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta, TR)$$

Network training is an optimization process to find params minimizing the loss function.

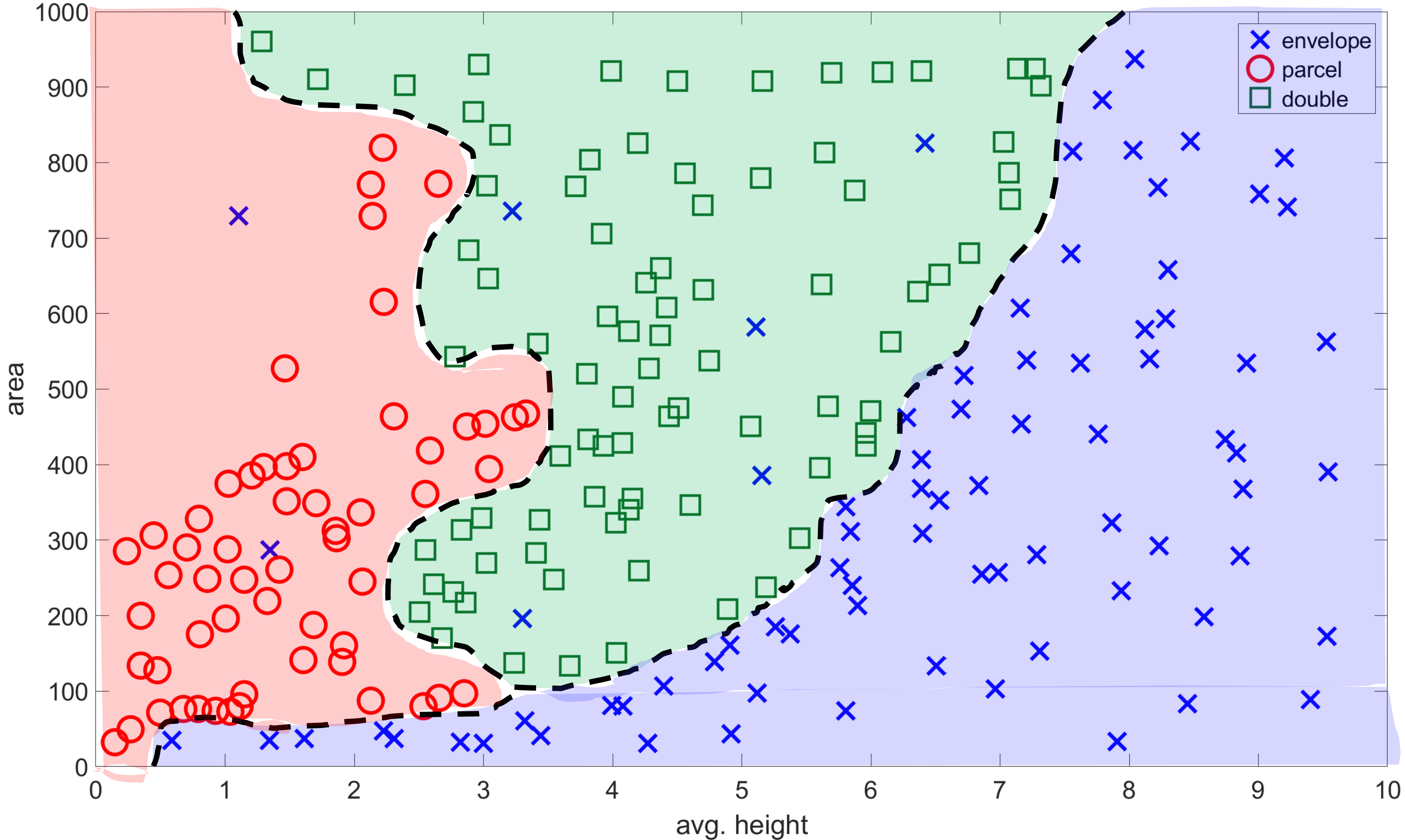
Can definitvely boost the image classification performance

- Annotated training set is always needed
- Classification performance also depends on the training set
- Generalization is not guaranteed

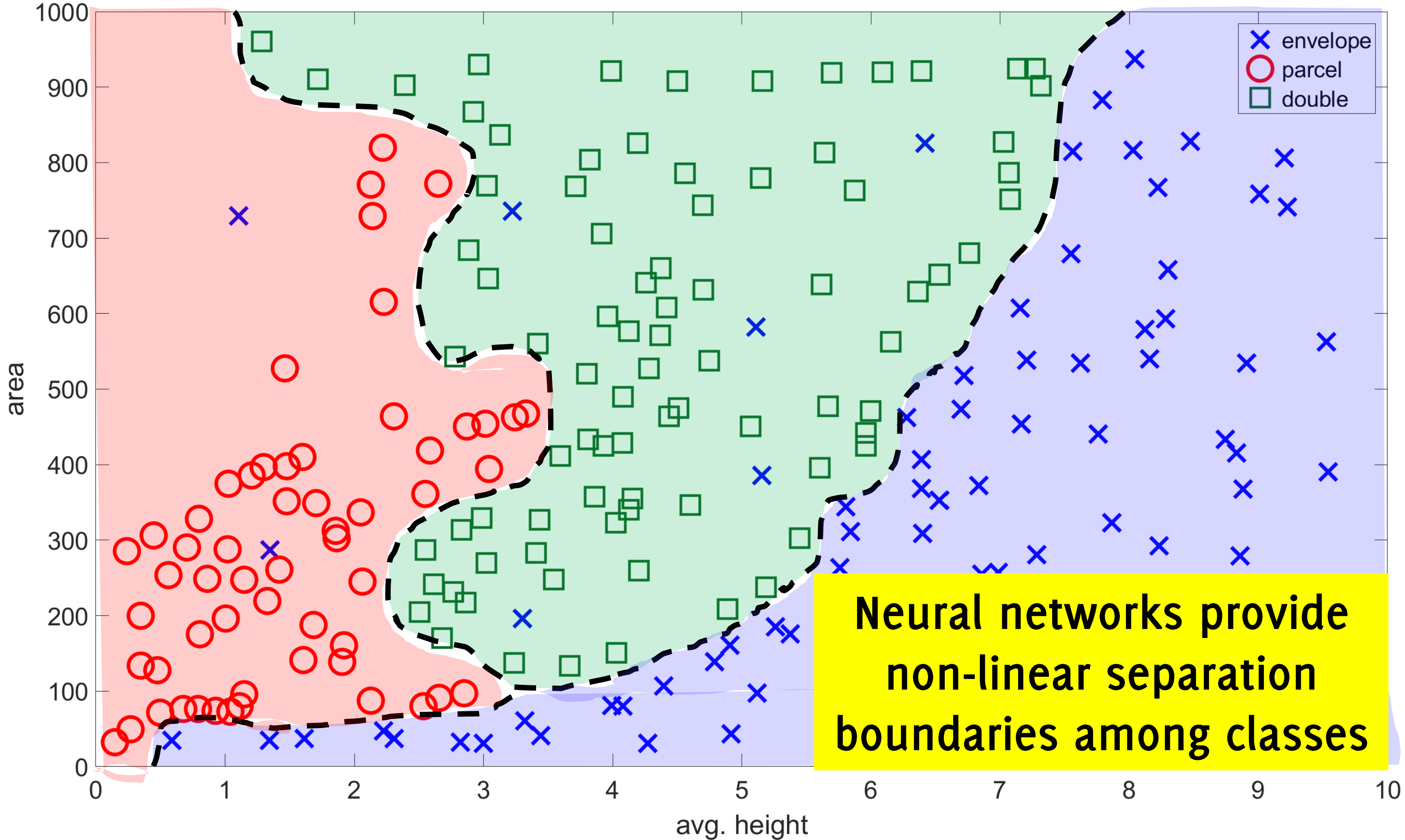
Hand Crafted Feature Extraction, data-driven Classification



Are there better classifiers?

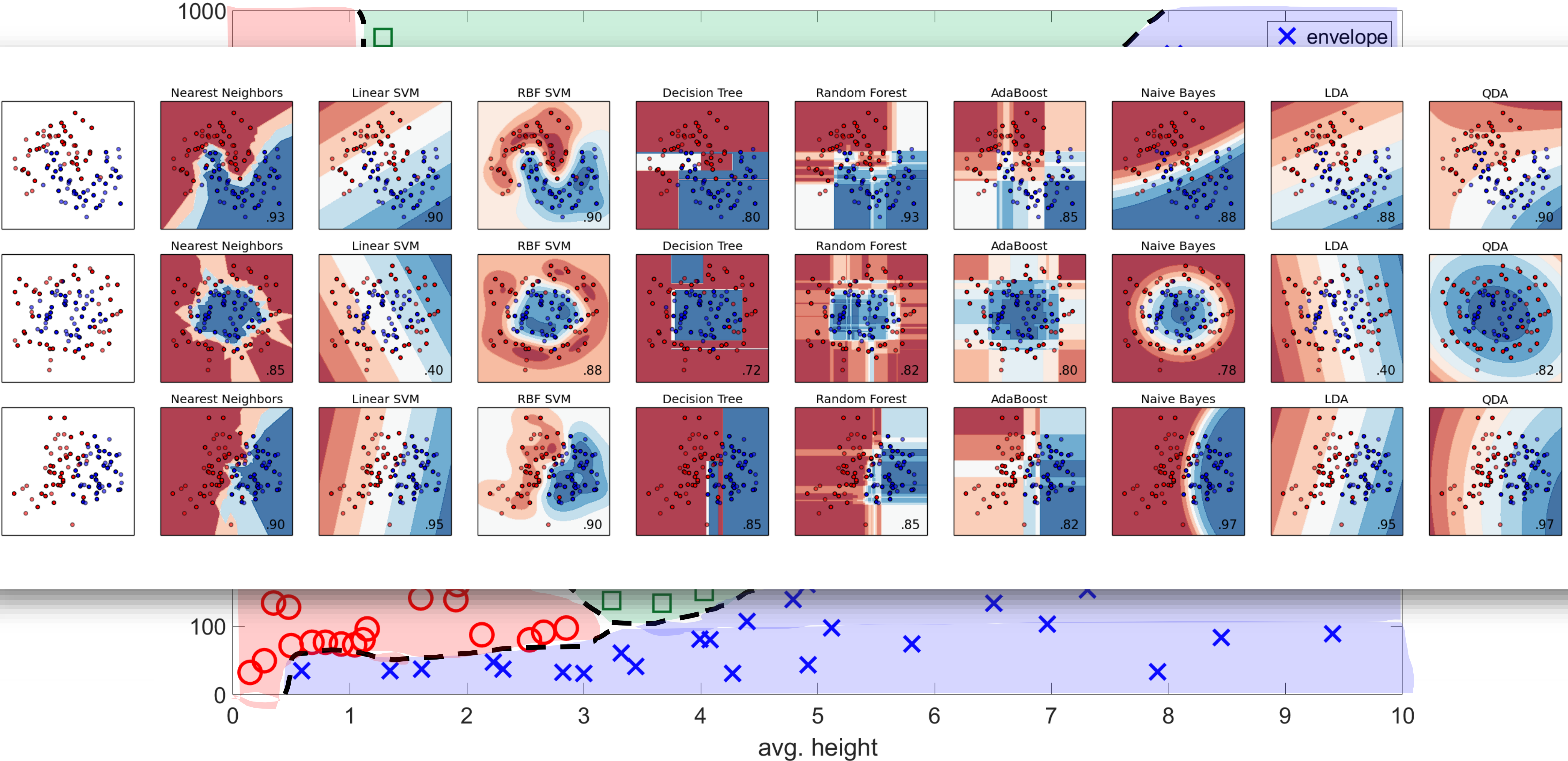


Are there better classifiers?



Neural networks provide non-linear separation boundaries among classes

And Neural Networks are not the only..



Neural Networks

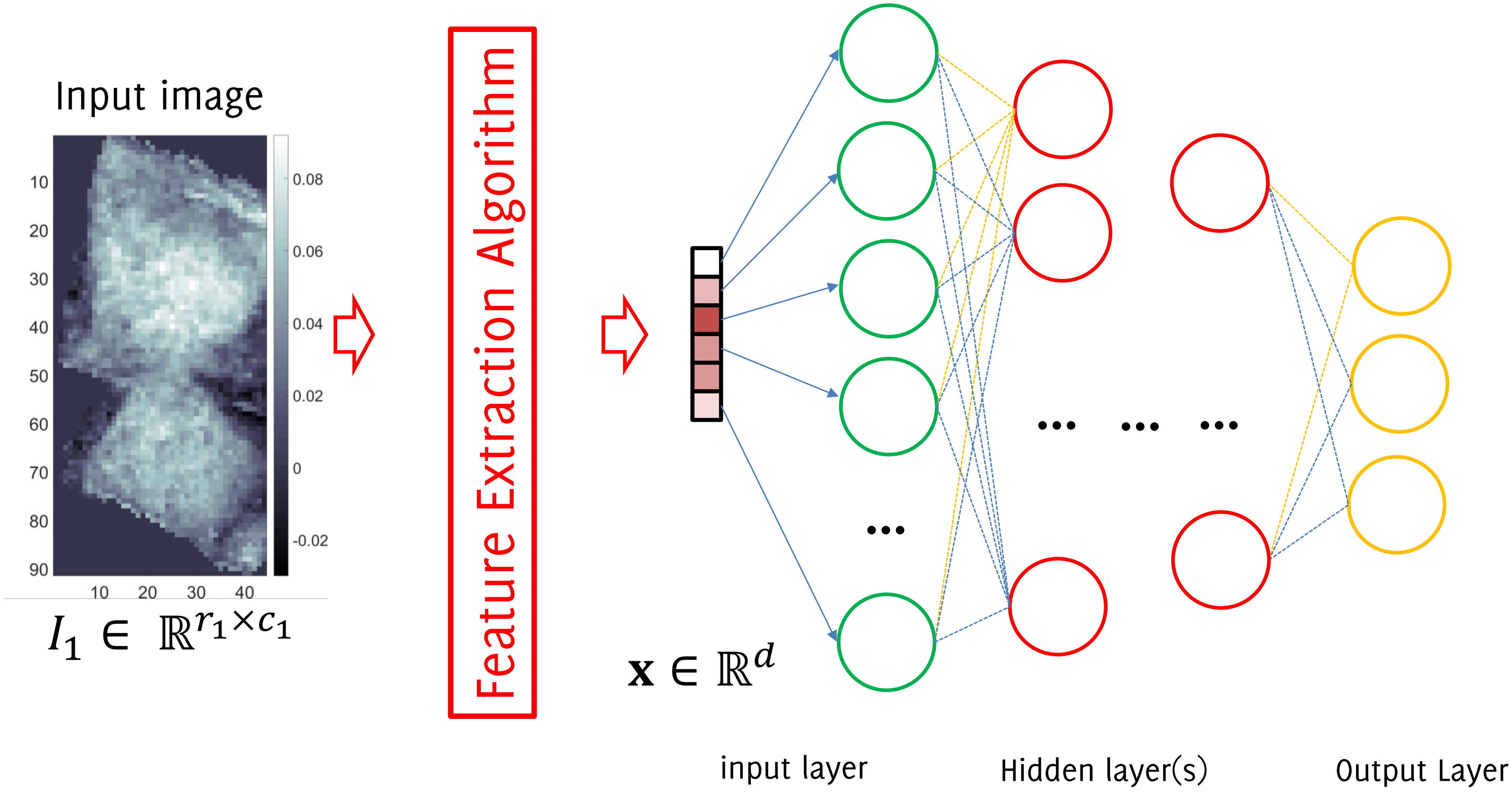
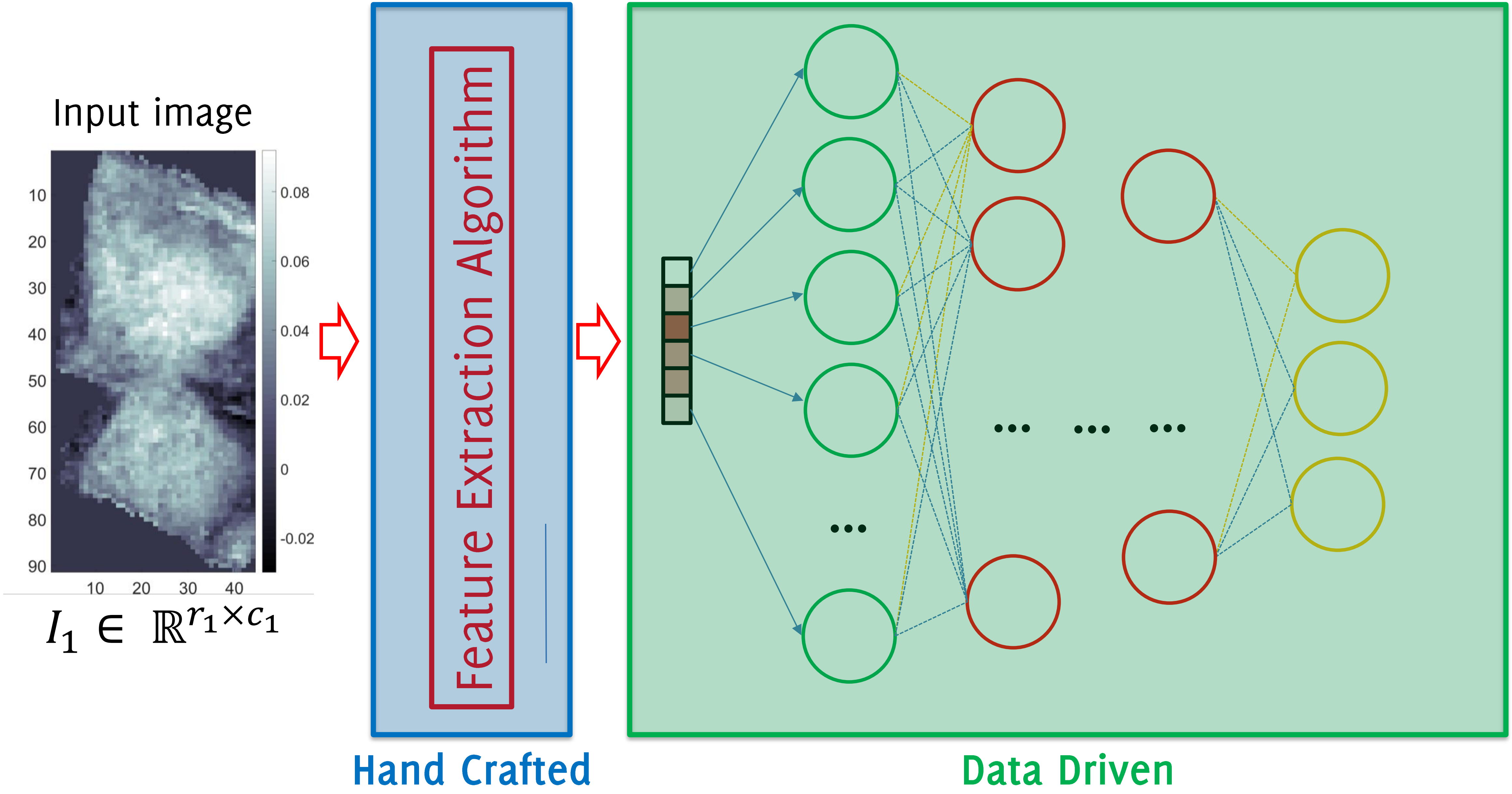


Image Classification by Hand Crafted Features



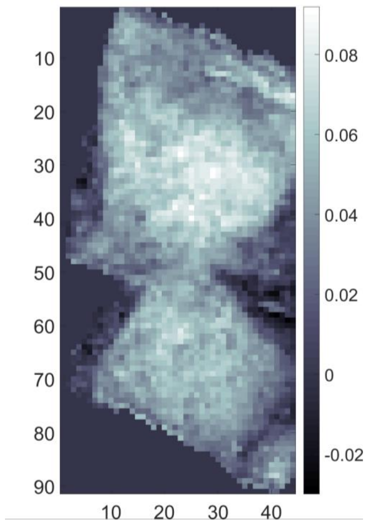
Hand Crafted Featues, pros:

- **Exploit a priori** / expert **information**
- Features are **interpretable** (you might understand why they are not working)
- You can **adjust features** to improve your performance
- **Limited amount of training data** needed
- You can give more relevance to some features

Hand Crafted Featues, cons:

- Requires a lot of **design/programming efforts**
- **Not viable** in many **visual recognition** tasks that are easily performed by humans (e.g. when dealing with natural images)
- **Risk of overfitting** the training set used in the feature design
- **Not very general and "portable"**

What is Deep Learning after all?

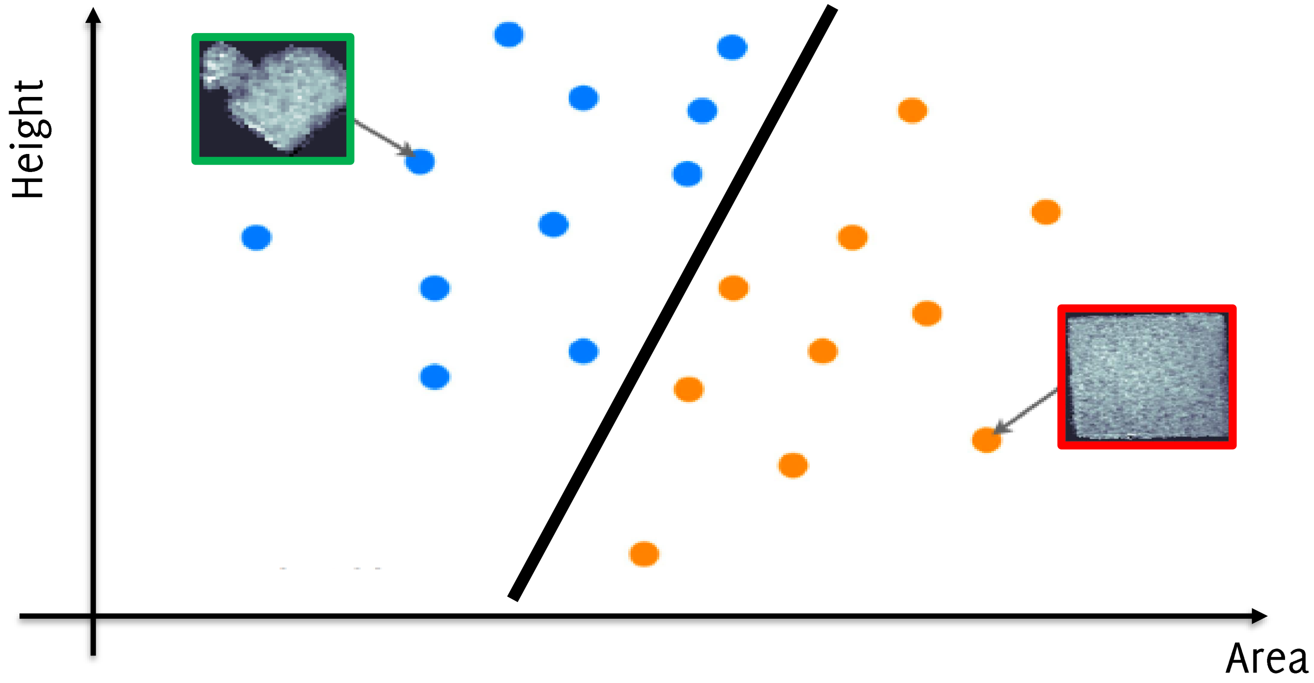


Machine learns how to take samples from different classes apart

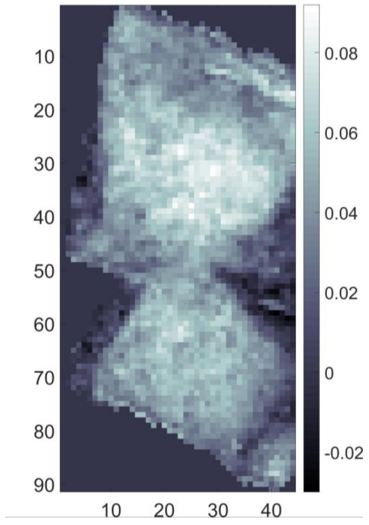
Hand-crafted Features

Learned Classifier

double



What is Deep Learning after all?

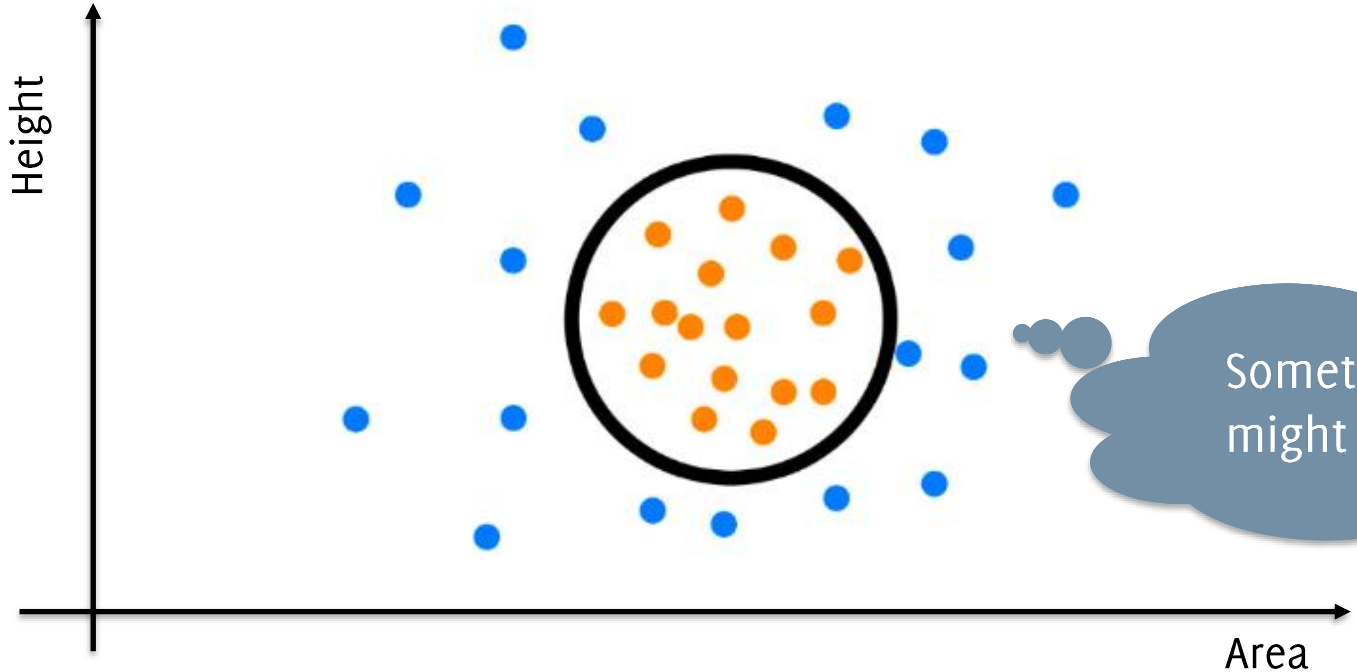


Hand-crafted Features

Learned Classifier

double

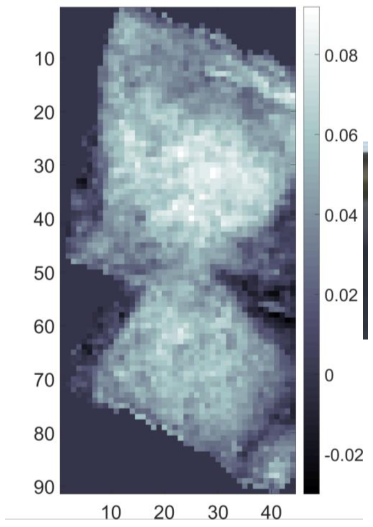
Machine learns how to take samples from different classes apart

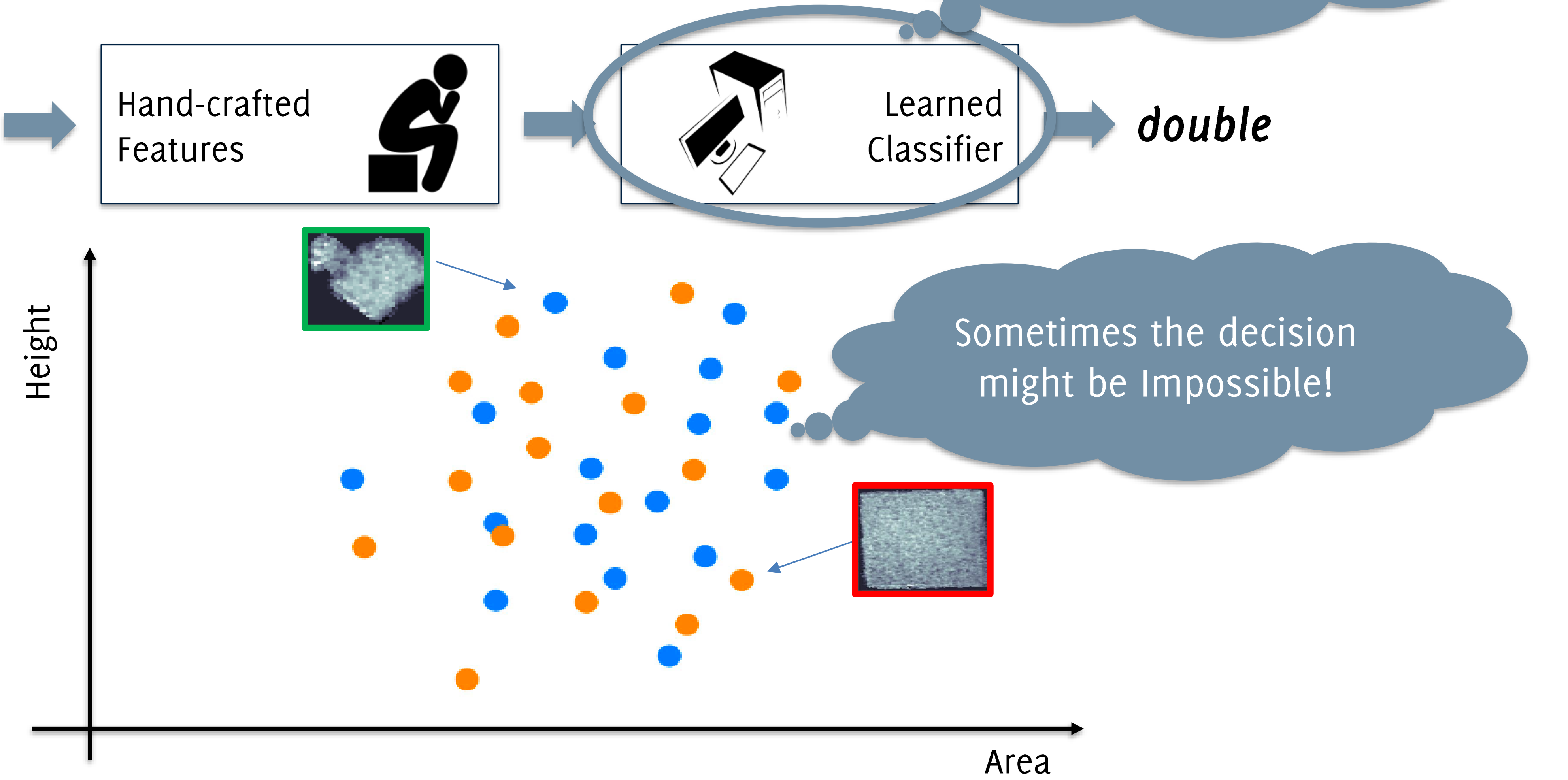


Sometimes the decision might be more complex

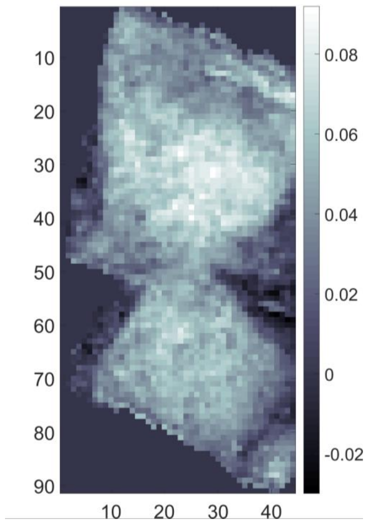
What is Deep Learning after all?

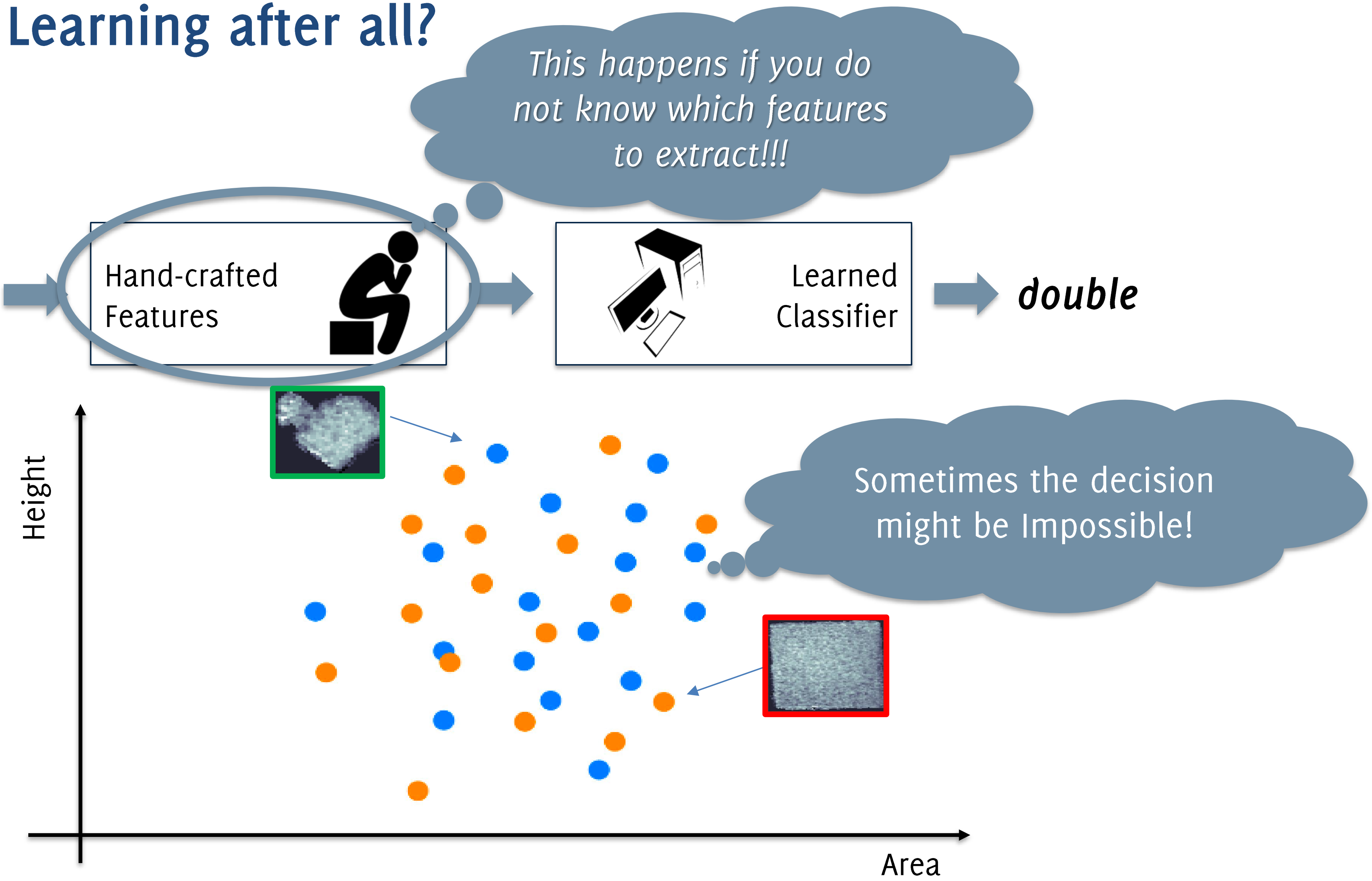
Machine learns how to take samples from different classes apart





What is Deep Learning after all?

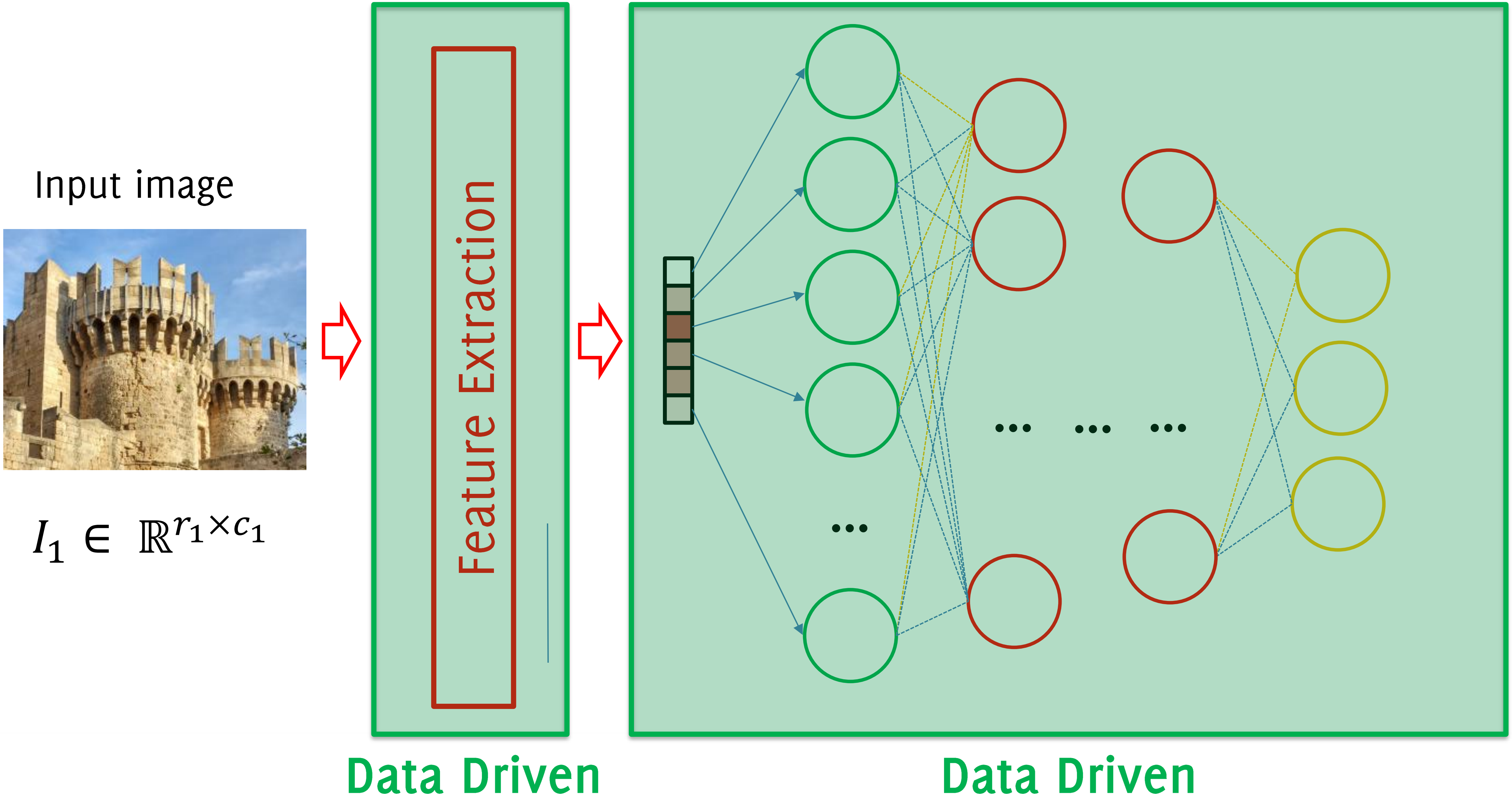




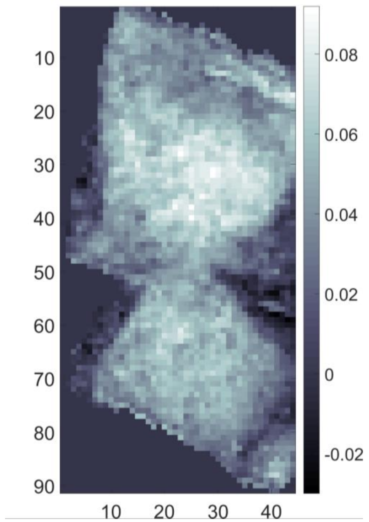
Data Driven Features

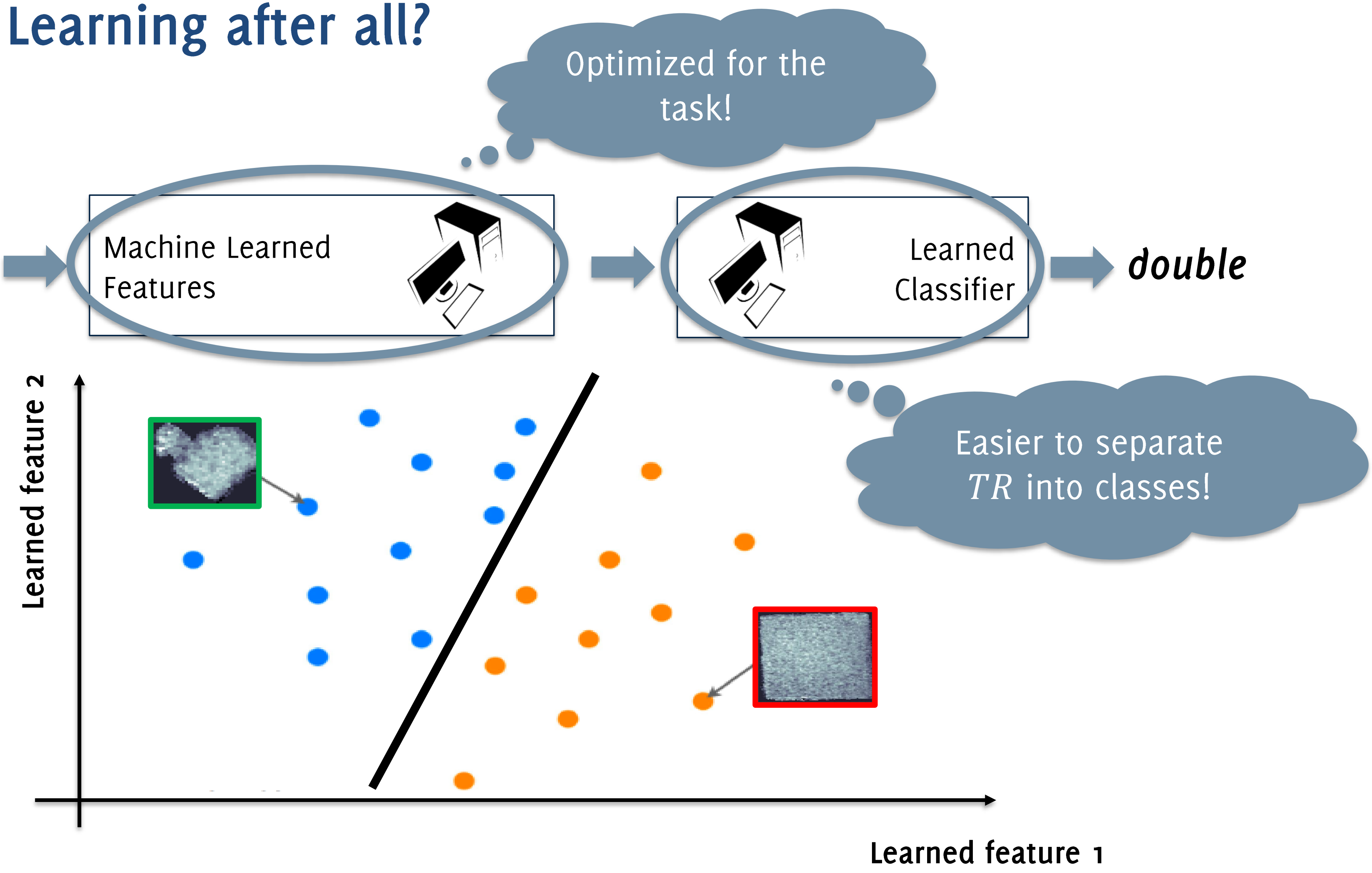
… the advent of Deep Learning

Data-Driven Features



What is Deep Learning after all?





What is Deep Learning after all?