Digital Image Processing (CSE/ECE 478)

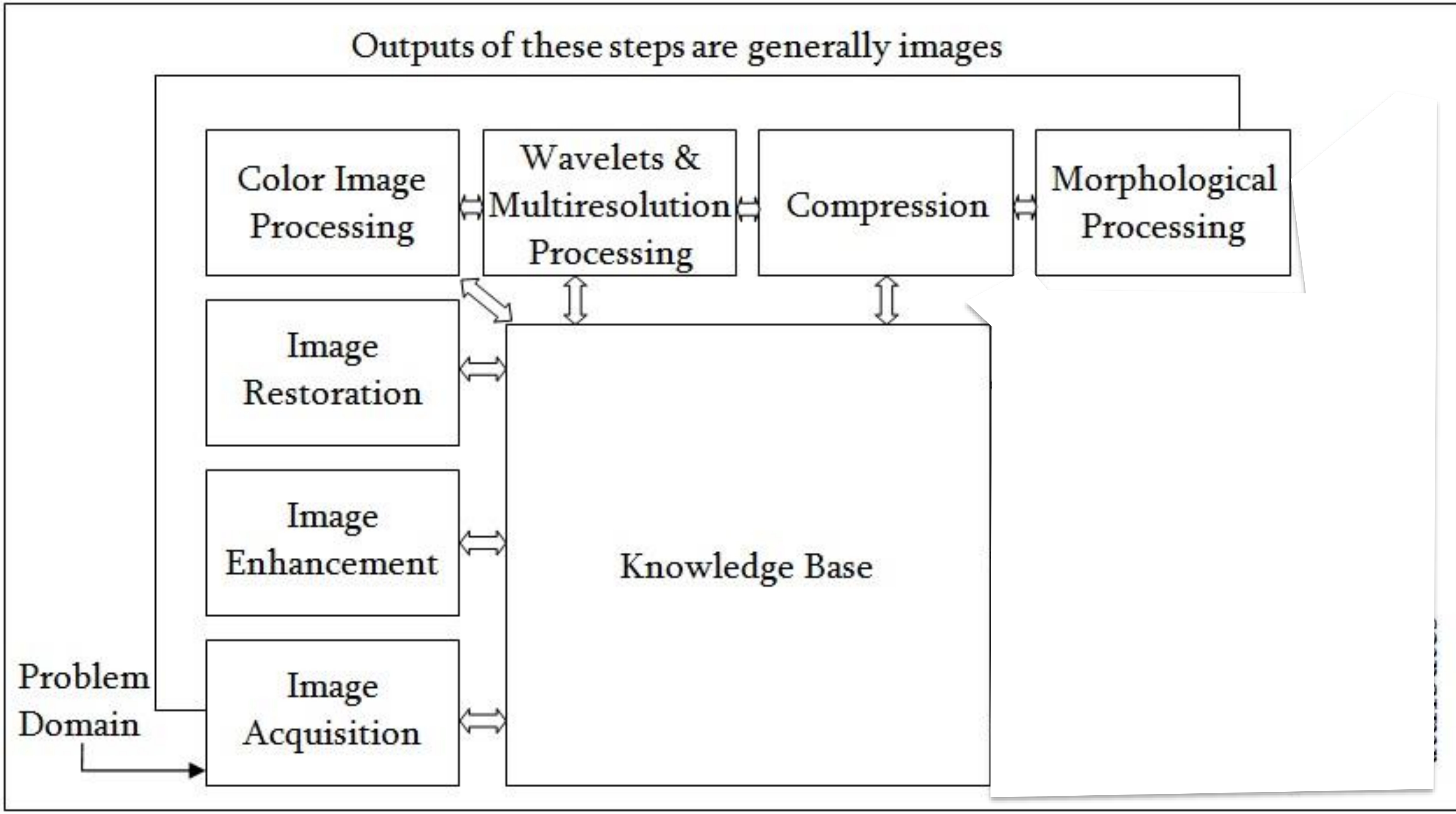
Lecture-21: Machine Learning / Pattern Recognition for Image Processing

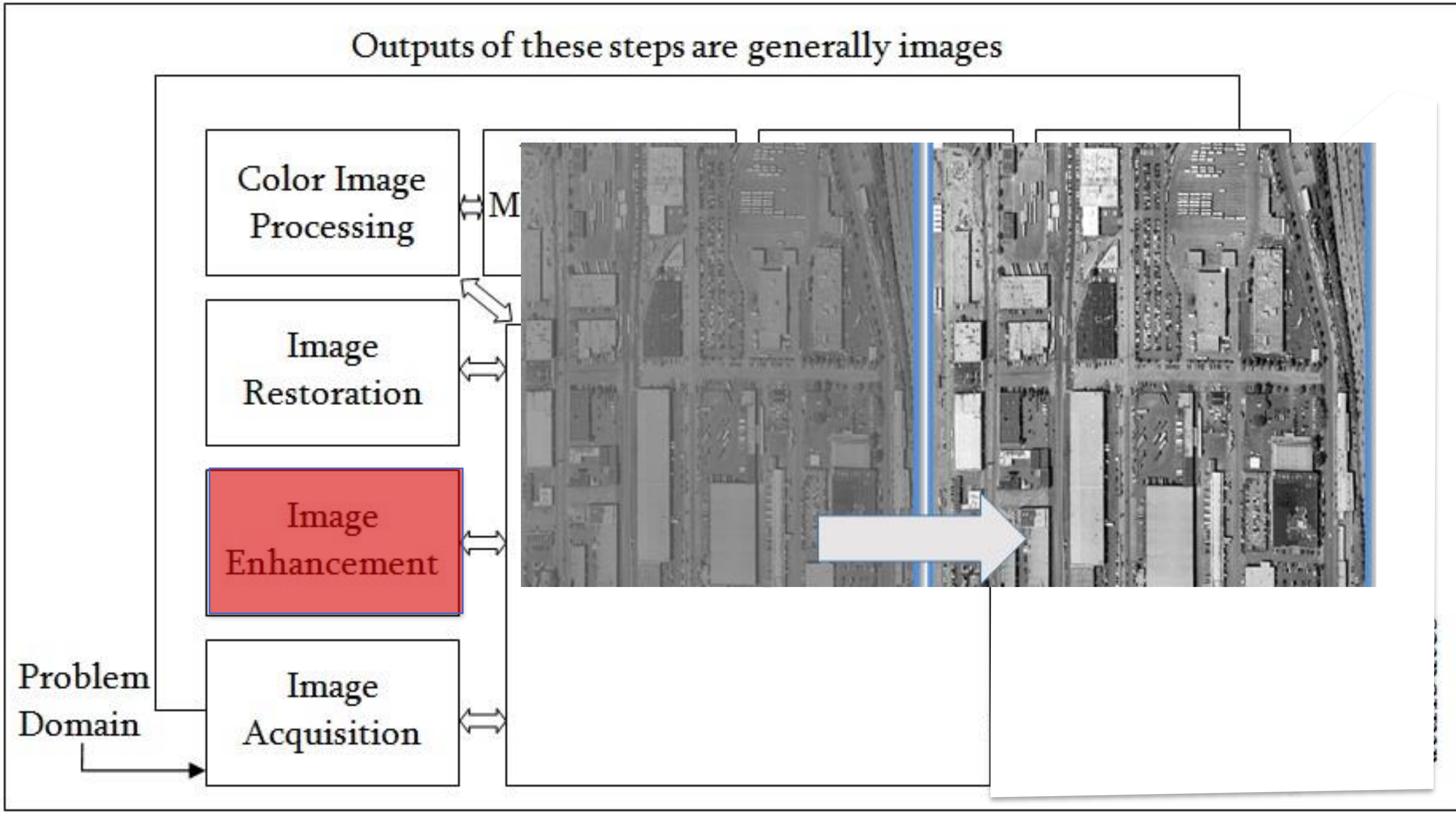
Ravi Kiran

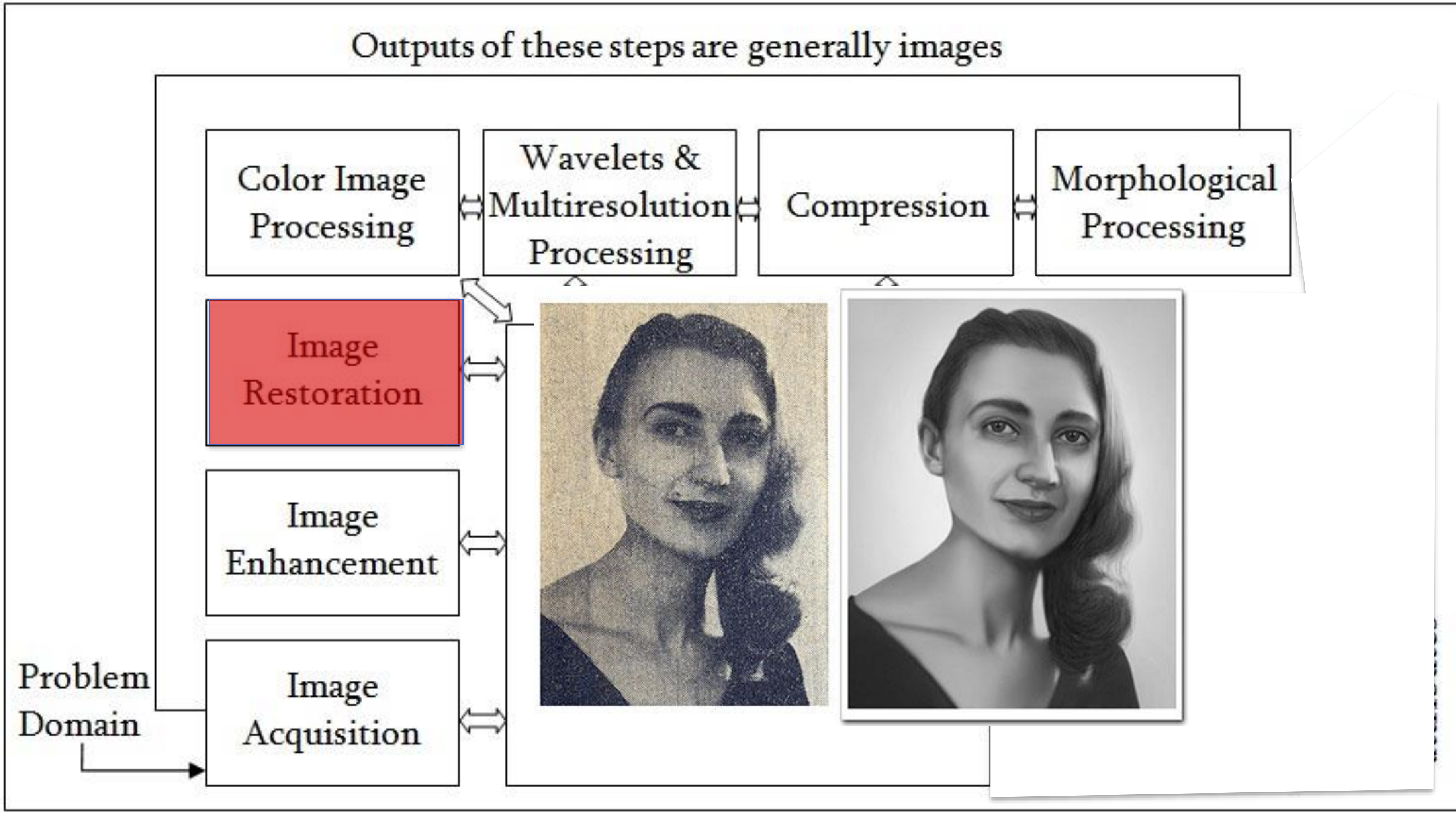
Rajvi Shah

Center for Visual Information Technology (CVIT), IIIT Hyderabad

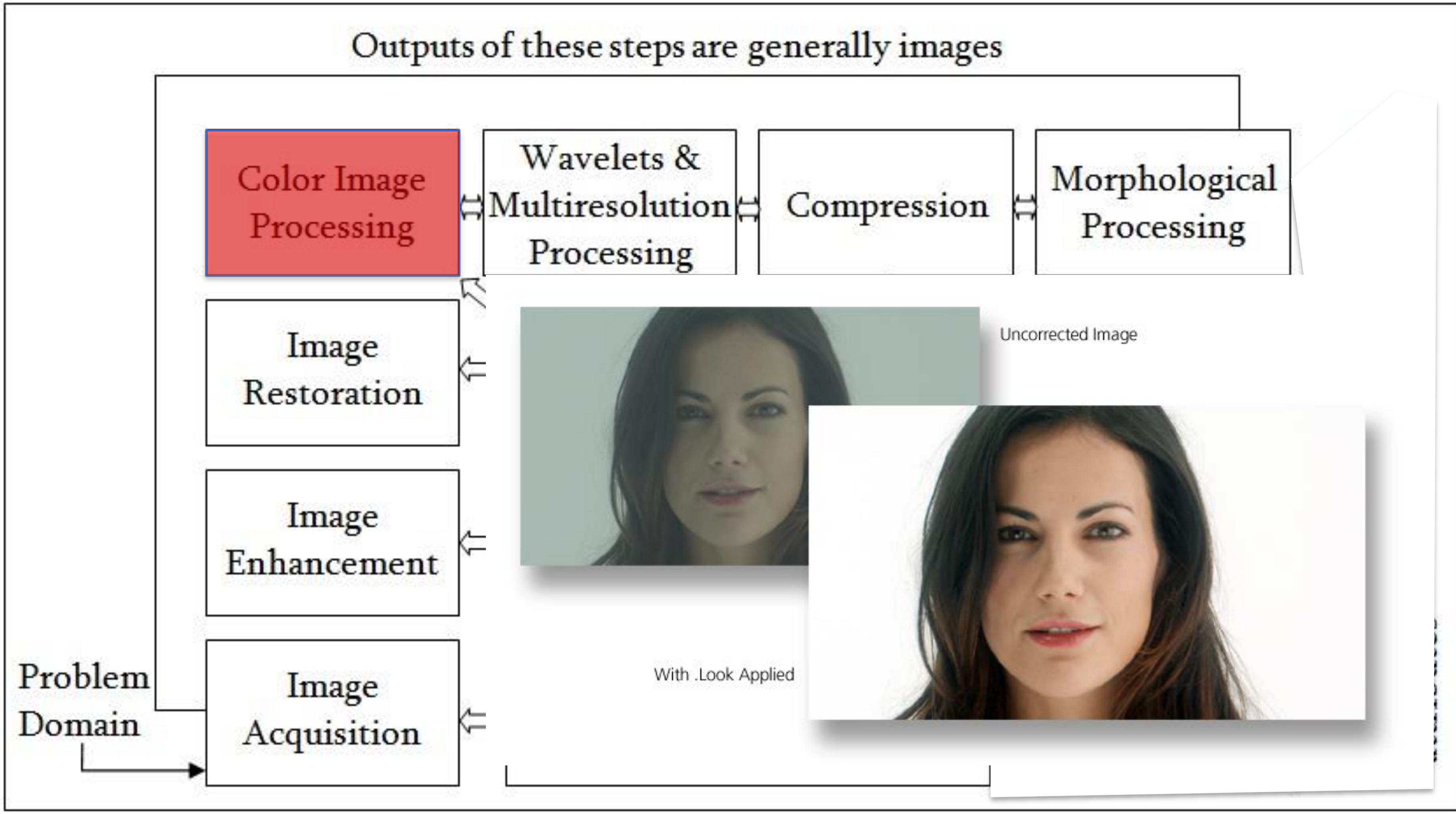
Some slides from: Alexandre Gramfort (Univ. Paris-Saclay), Castillo (NIH, US)

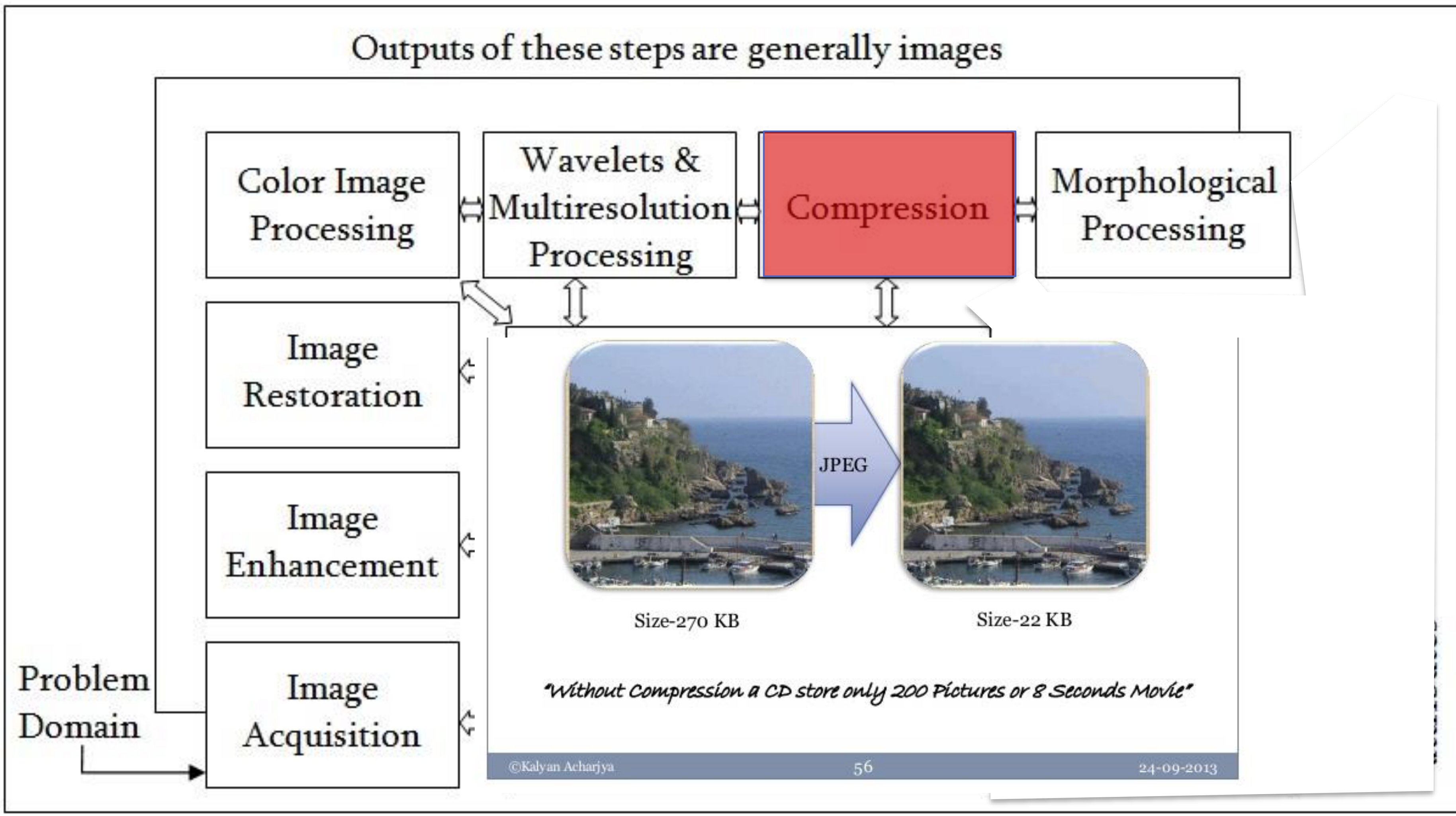

Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Image Enhancement
Image Acquisition
Knowledge Base
Problem Domain

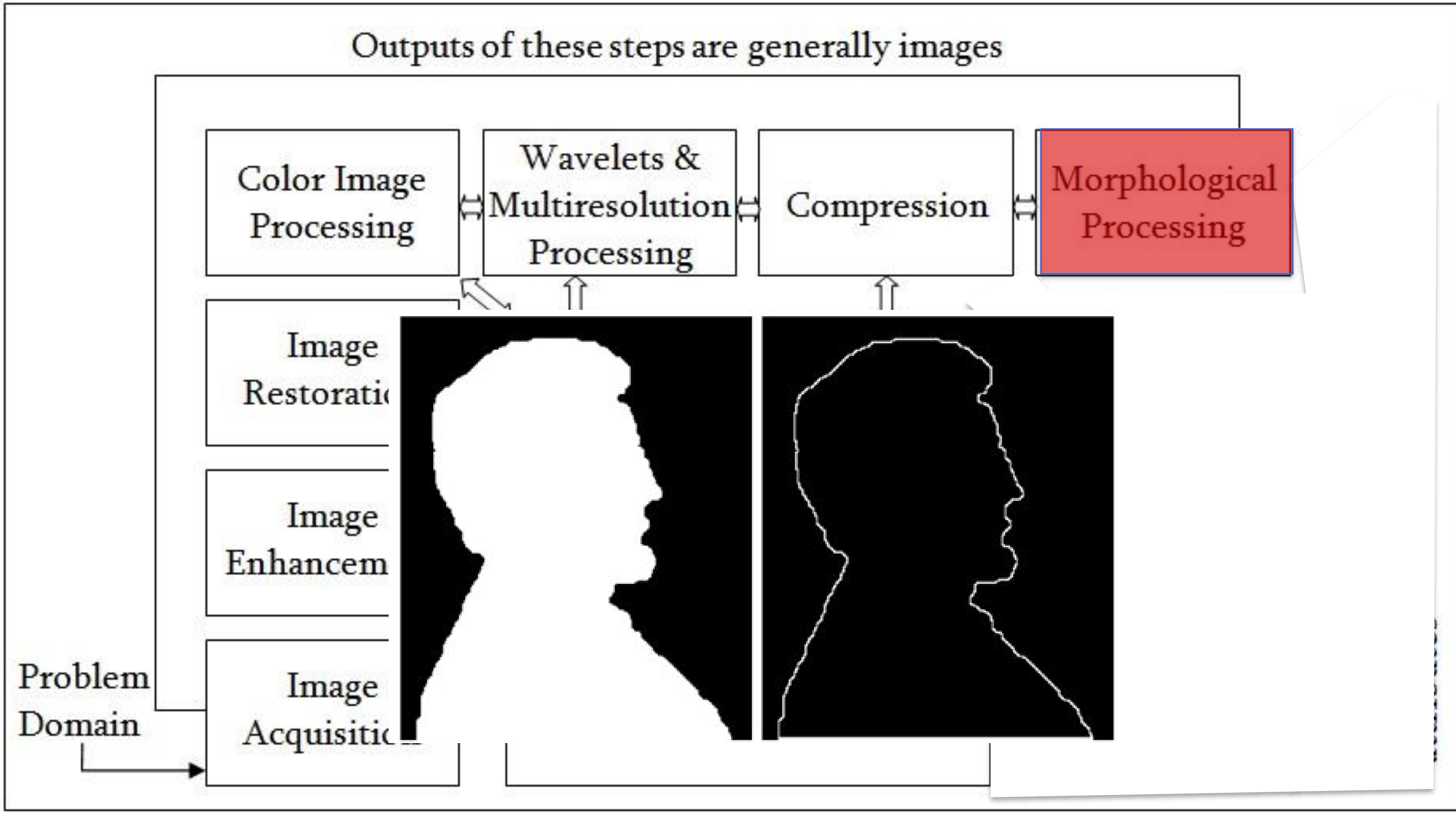

Outputs of these steps are generally images
Color Image Processing
M
Image Restoration
Image Enhancement
Image Acquisition
Problem Domain

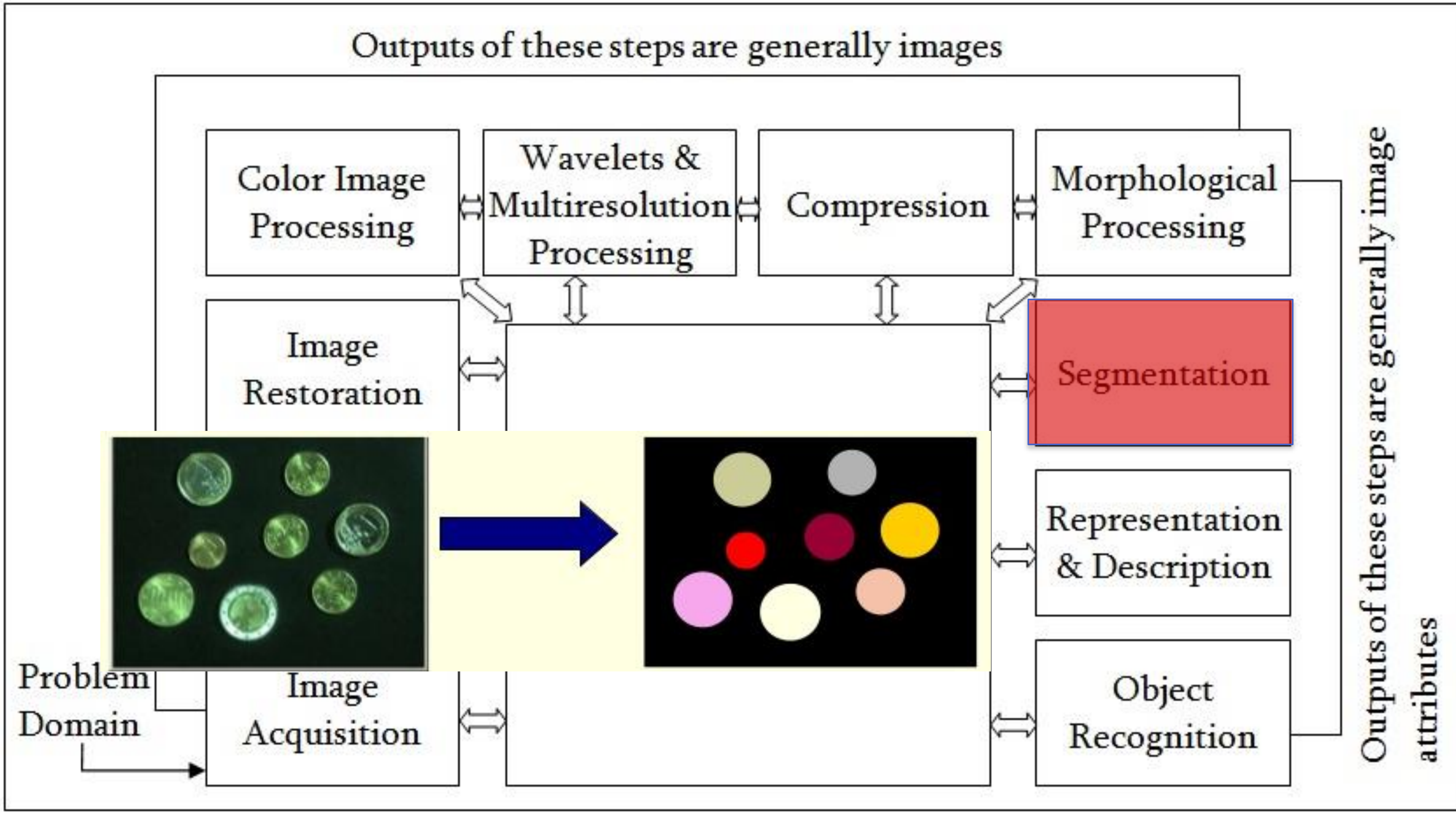


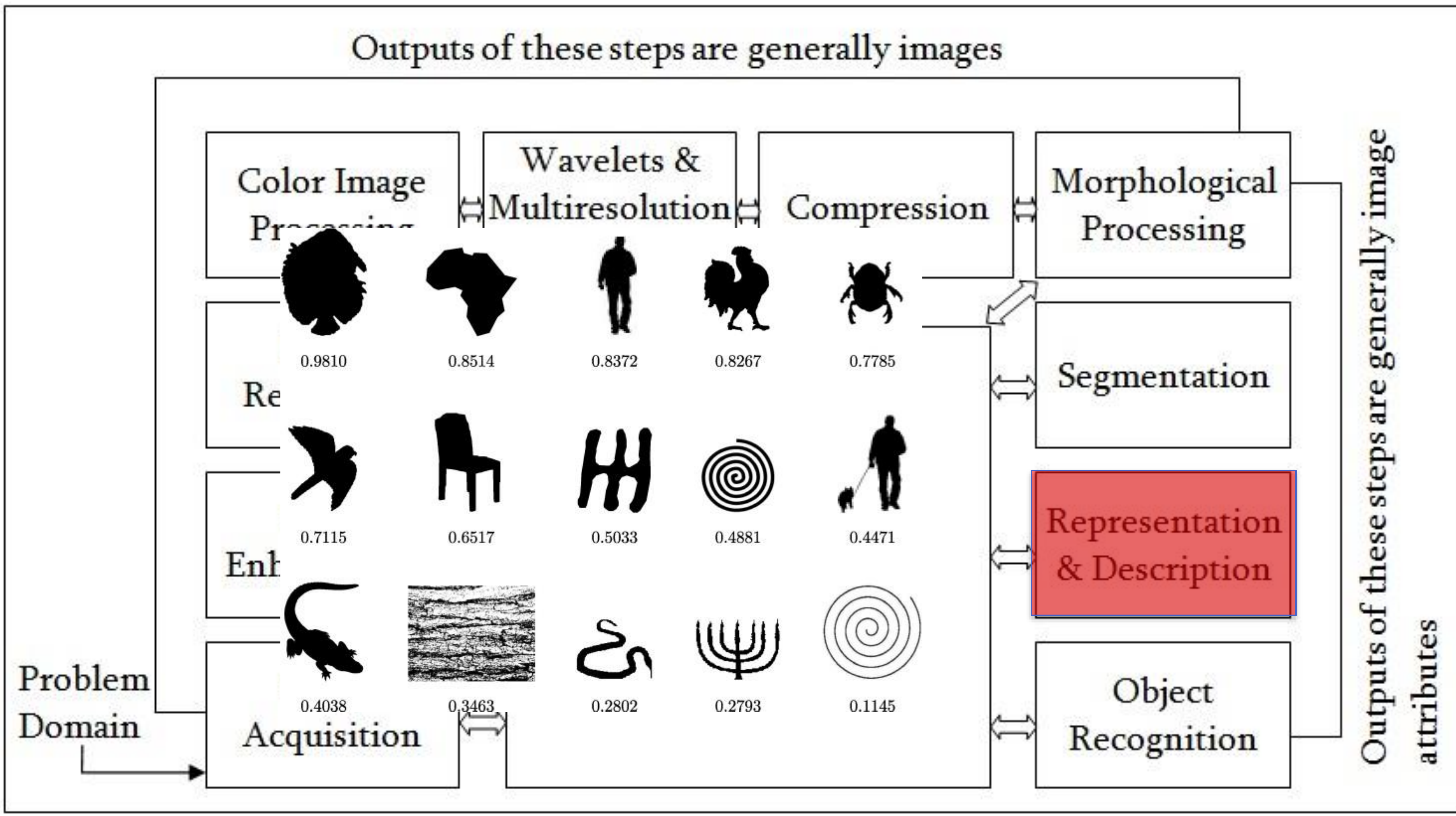
Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Image Enhancement
Image Acquisition
Problem Domain


Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Image Enhancement
Image Acquisition
Problem Domain
Uncorrected Image
With .Look Applied

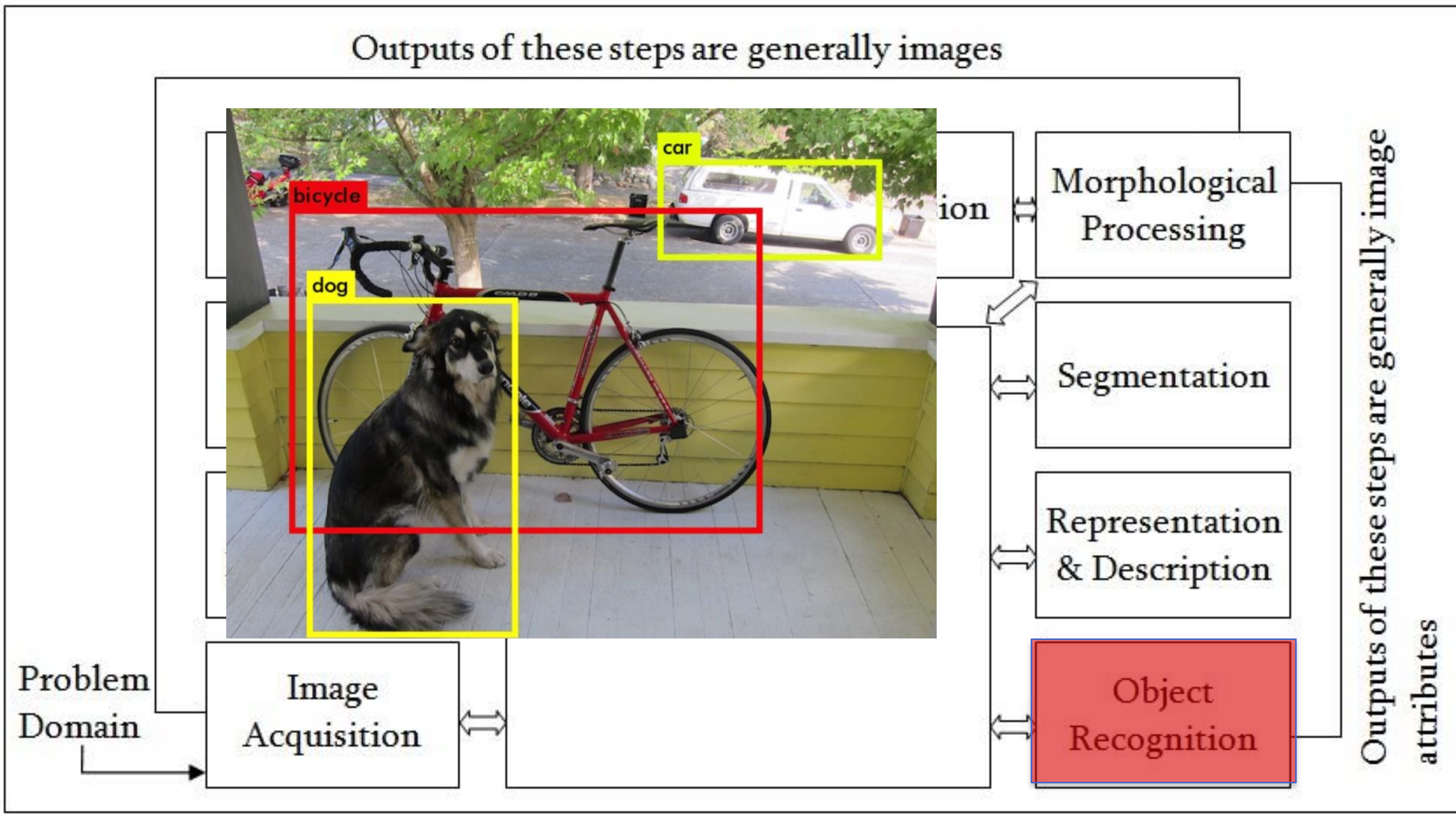

Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Image Enhancement
Image Acquisition
Problem Domain
JPEG
Size-270 KB
Size-22 KB
"Without Compression a CD store only 200 Pictures or 8 Seconds Movie"
©Kalyan Acharjya
56
24-09-2013


Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restorati
Image Enhancem
Image Acquisitic
Problem Domain


Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution Processing
Compression
Morphological Processing
Image Restoration
Segmentation
Representation & Description
Object Recognition
Image Acquisition
Problem Domain
Outputs of these steps are generally image attributes

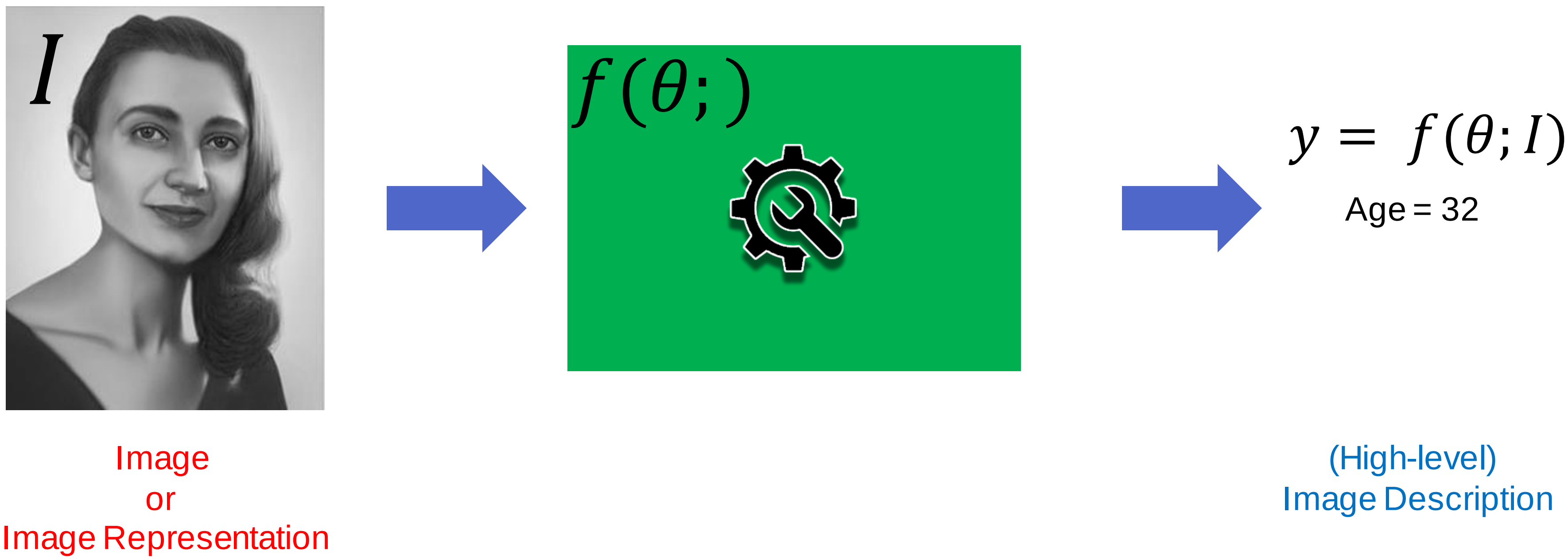


Outputs of these steps are generally images
Color Image Processing
Wavelets & Multiresolution
Compression
Morphological Processing
Segmentation
Representation & Description
Object Recognition
Problem Domain
Acquisition
Outputs of these steps are generally image attributes
0.9810
0.8514
0.8372
0.8267
0.7785
0.7115
0.6517
0.5033
0.4881
0.4471
0.4038
0.3463
0.2802
0.2793
0.1145



Outputs of these steps are generally images
car
bicycle
dog
ion
Morphological Processing
Segmentation
Representation & Description
Object Recognition
Problem Domain
Image Acquisition
Outputs of these steps are generally image attributes

Example of a system which outputs high-level image description



I

Image
or
Image Representation

- Spatial Dimension-preserving
- Partial dimension-preserving
- Non-dimension preserving

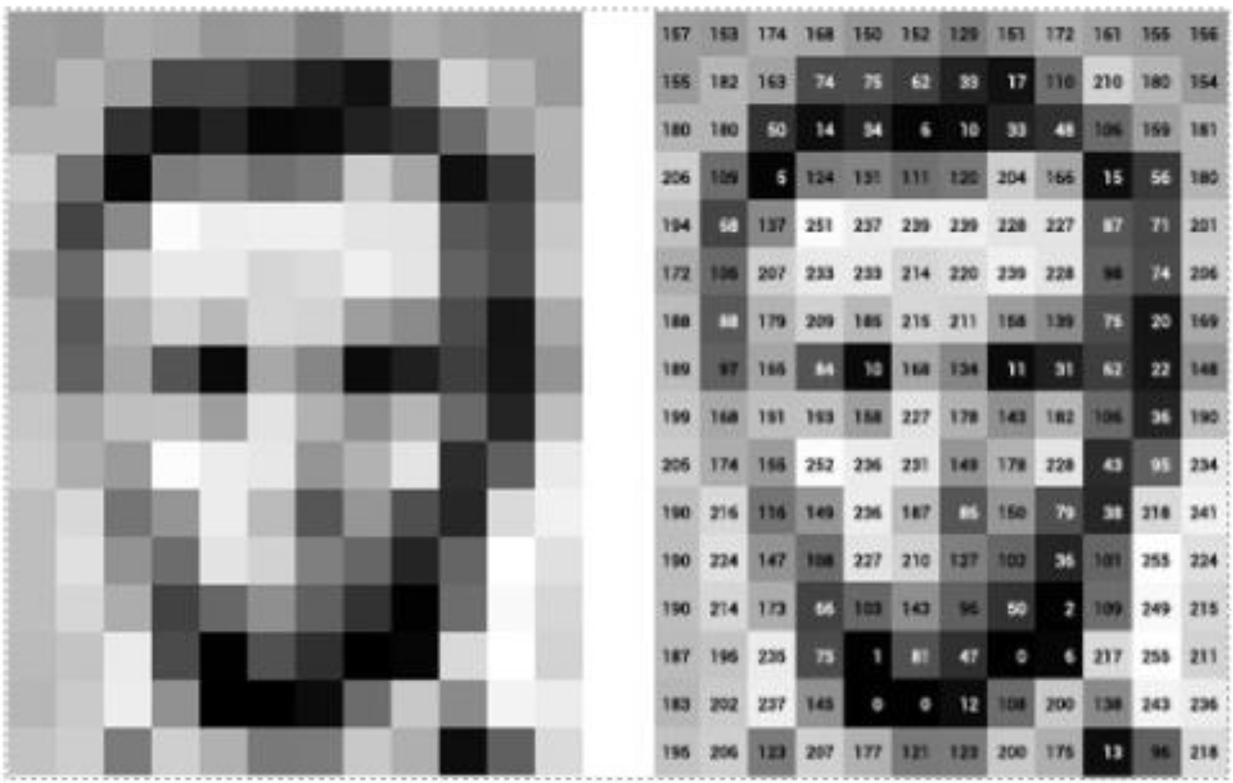
I

Image
or
Image Representation

$f(\theta;\,)$

$y = f(\theta; I)$

(High-level)
Image Description



Raw image

I

Image
or
Image Representation

$f(\theta;)$

$y = f(\theta; I)$

(High-level)
Image Description

Gradient magnitude

Orientation at each pixel

$$\theta = \tan^{-1}\frac{gy}{gx}$$

39

https://images.slideplayer.com/17/5266552/slides/slide_39.jpg

$$I$$

Image
or
Image Representation

$$f(\theta; \ )$$

$$y = f(\theta; I)$$

(High-level)
Image Description

Intensity Transformation

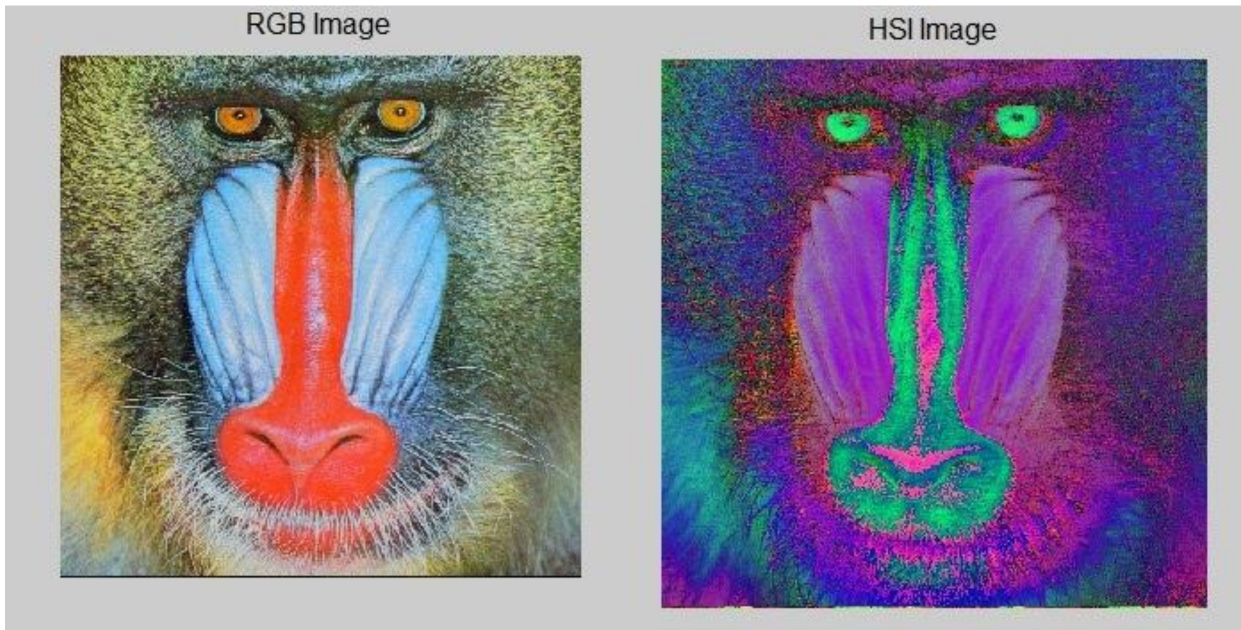
$$I$$

Image
or
Image Representation

$$f(\theta;\ )$$

$$y = f(\theta; I)$$

(High-level)
Image Description



Colorspace Transformation

$f(\theta;\,)$

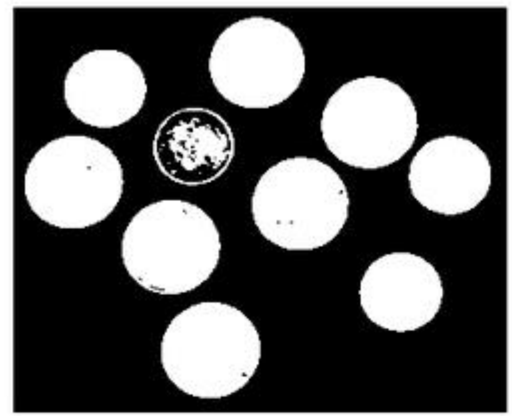
I

Image
or
Image Representation

$y = f(\theta; I)$

(High-level)
Image Description

Input Image



threshold output

Morphological operations

I

Image
or
Image Representation

- Spatial Dimension-preserving
- Partial dimension-preserving
- Non-dimension preserving

I

Image
or
Image Representation

$f(\theta; \ )$

$y = f(\theta; I)$

(High-level)
Image Description

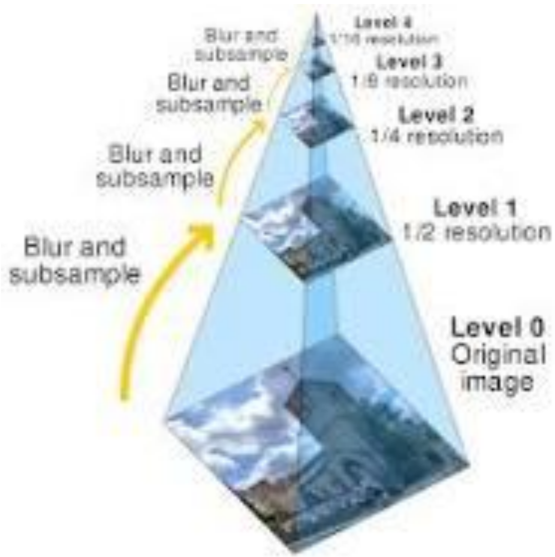


Image Pyramid

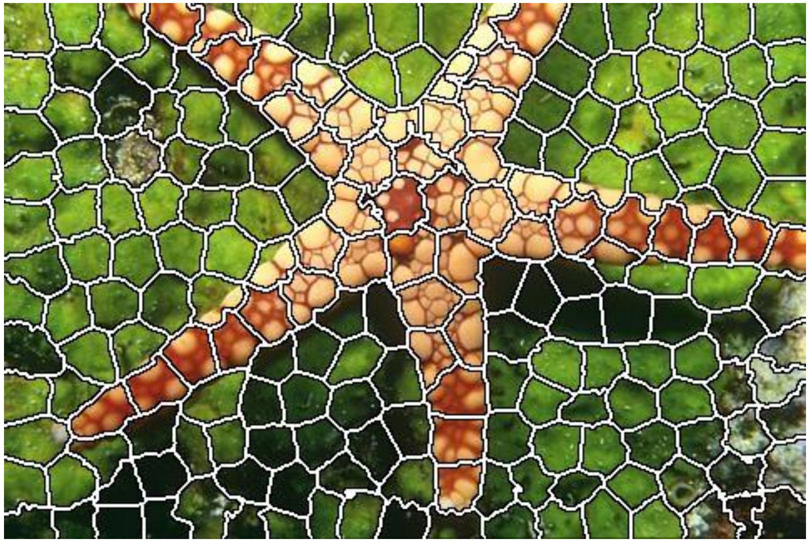
$$I$$

Image
or
Image Representation

$$f(\theta;)$$

$$y = f(\theta; I)$$

(High-level)
Image Description



Superpixels

I

Image
or
Image Representation

- Spatial Dimension-preserving
- Partial dimension-preserving
- Non-dimension preserving

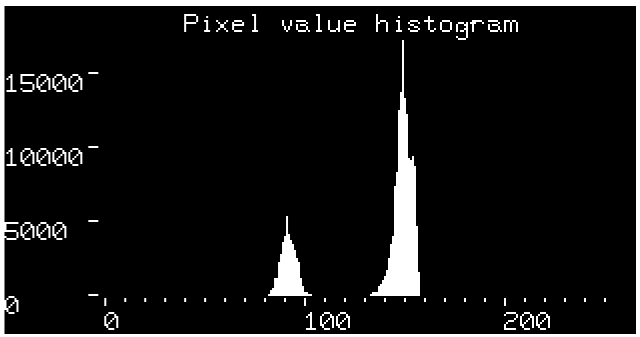
I

Image
or
Image Representation

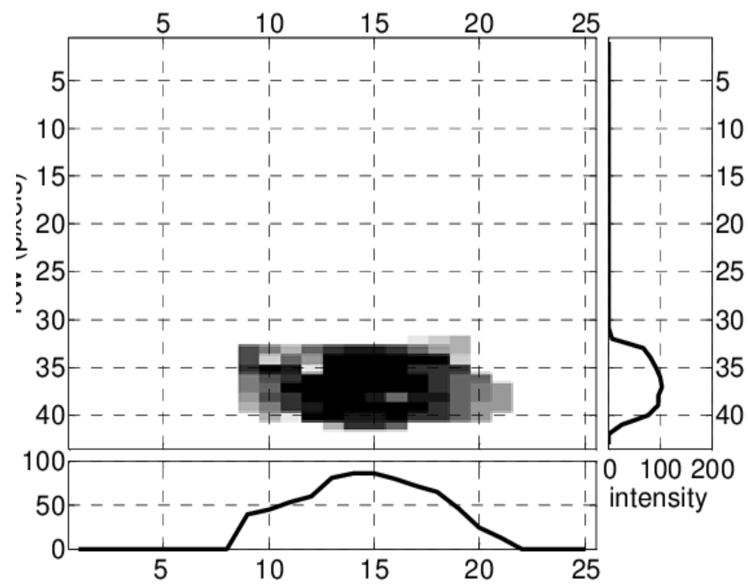
$f(\theta;\,)$

$y = f(\theta; I)$

(High-level)
Image Description



Histogram



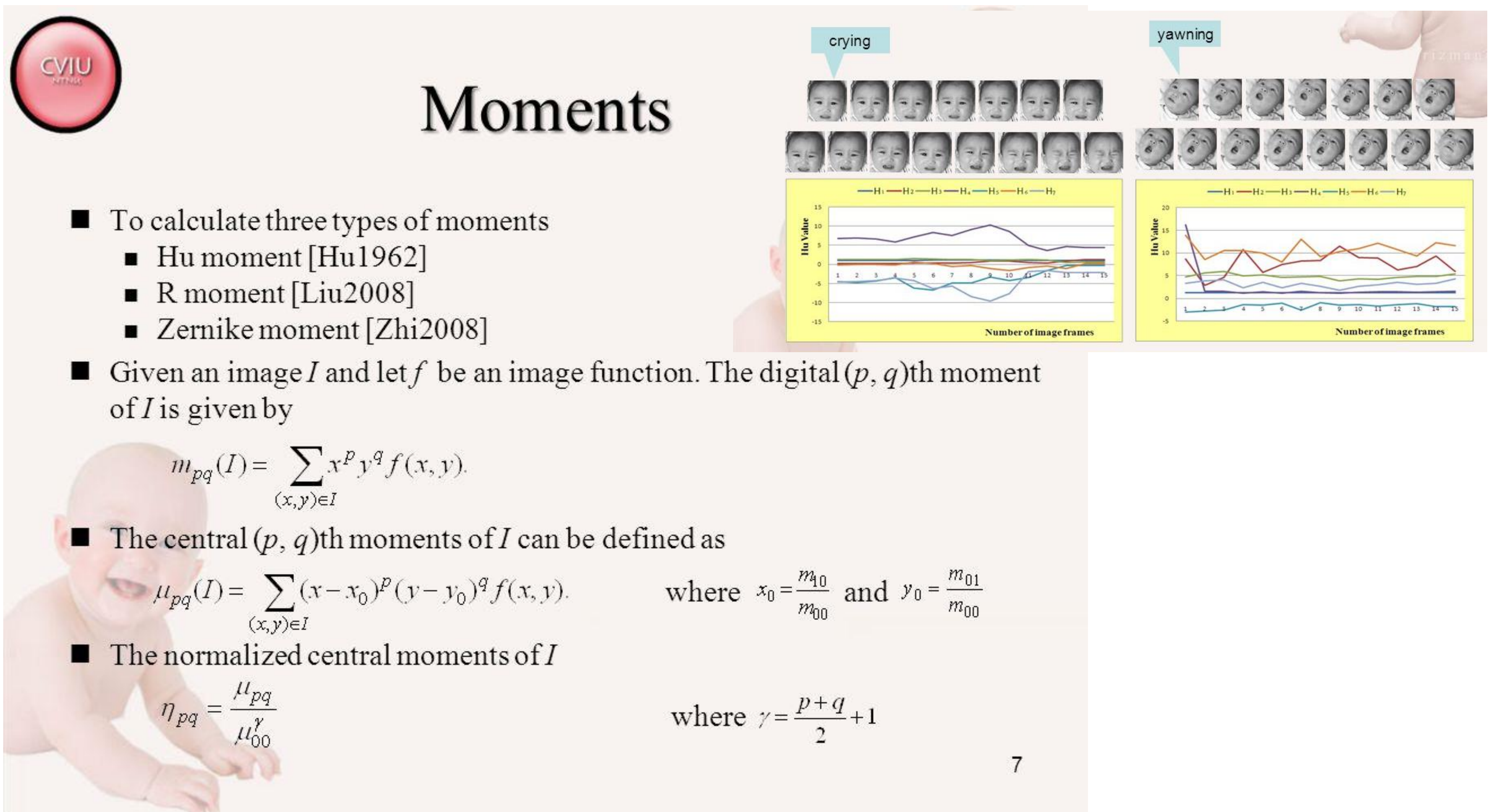
I

Image
or
Image Representation

$f(\theta;)$

$y = f(\theta; I)$

(High-level)
Image Description



Moment-based

https://images.slideplayer.com/14/4273740/slides/slide_12.jpg

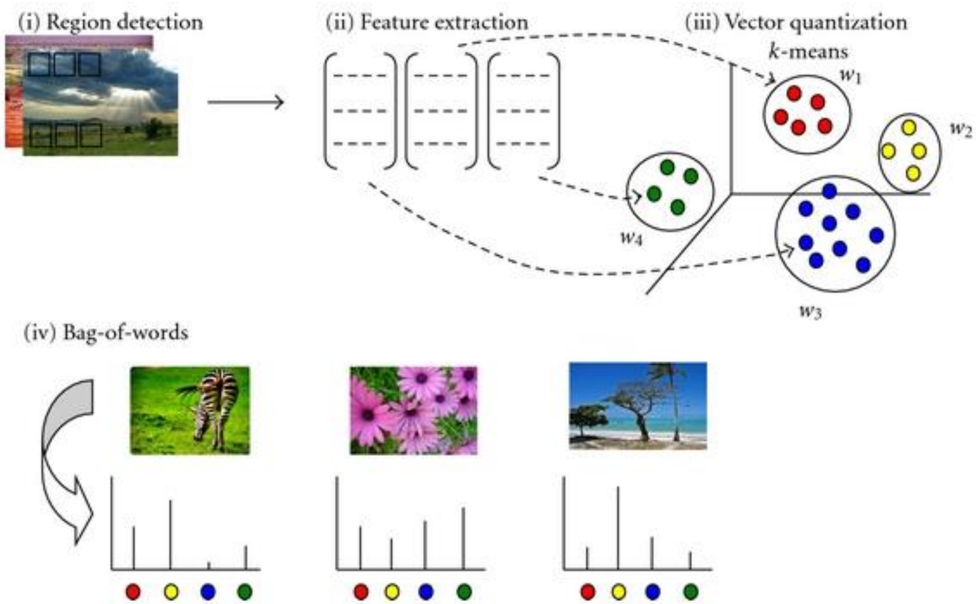
$f(\theta;\ )$

I

Image
or
Image Representation

$y = f(\theta; I)$

(High-level)
Image Description



https://www.hindawi.com/journals/isrn/2012/376804.fig.001.jpg

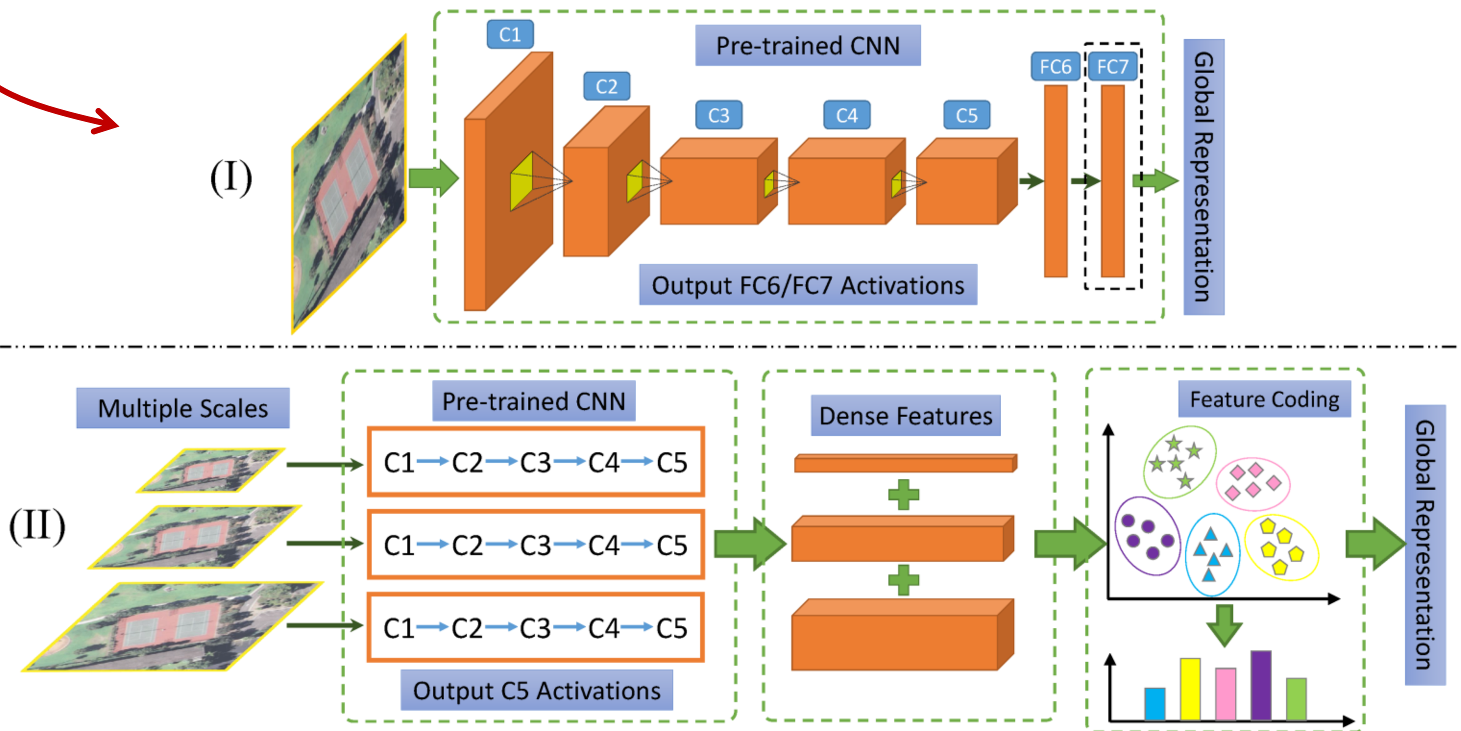
I

Image
or
Image Representation

$f(\theta;\ )$

$y = f(\theta; I)$

(High-level)
Image Description



Mathematical abstraction for input image representation

- Tensor (Spatial info preserving)
  - 2-D array / 3-D array / n-D array
- Vector (Distributed)
  - 1-D array
  - Simplest: Flattened tensor

I

Image
or
Image Representation

$f(\theta;\ )$

$y = f(\theta; I)$

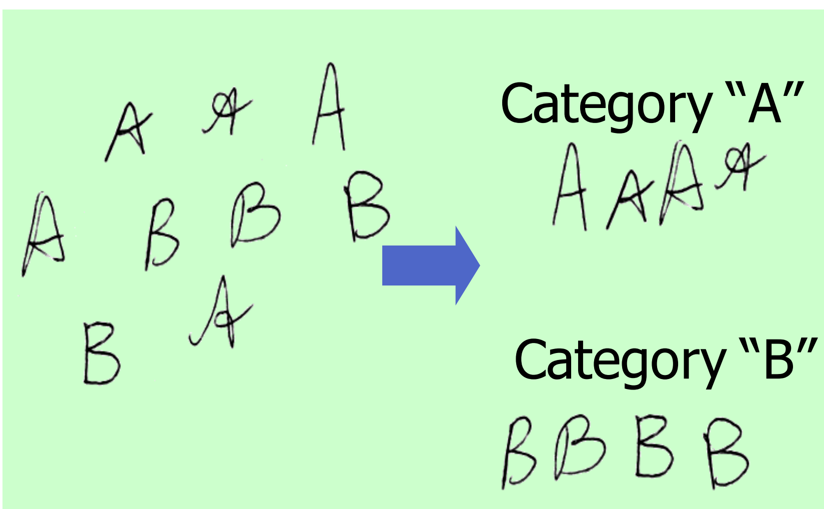
(High-level)
Image Description

Nature of mapping from I → y

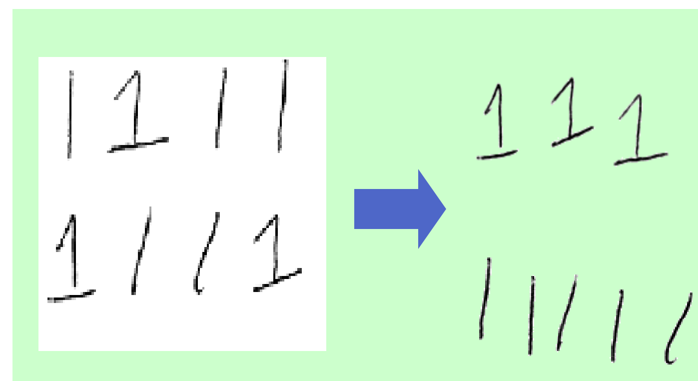
- Target-based
- Non-target based

Classification vs Clustering

- Classification (known categories)
- Clustering (unknown categories)



Classification (Recognition)
(Supervised Classification)



Clustering
(Unsupervised Classification)

I → $f(\theta; \ )$ → $y = f(\theta; I)$

Image
or
Image Representation

(High-level)
Image Description

- f
  - Simple ; deterministic
  - Complex ; non-linear/stochastic

Learning '*f*' from (I,y) sample pairs → Machine Learning

What is ML ?

- ML is
  - .. Programming computers
  - .. To optimize a performance criterion
  - .. Using example data and/or encoding of prior knowledge

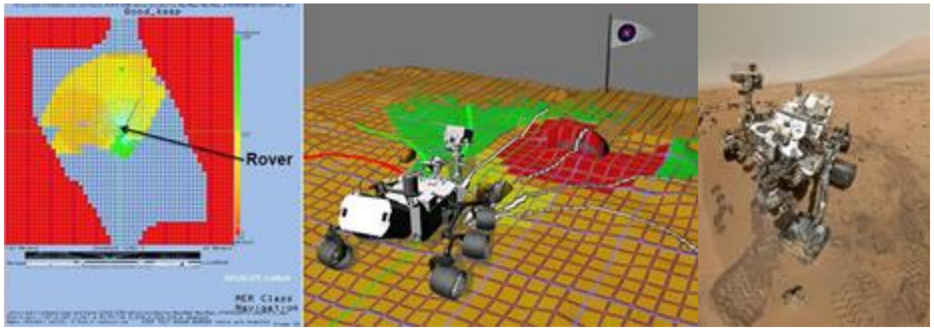
- Not required to calculate your GPA :)
  - Fixed, small set of rules

What is ML ?

- Computer program whose behavior evolve based on empirical data (Wikipedia)
- Computer program that learns from experience E in order to improve its performance P on a task T (Tom Mitchell)

experience E	:	images, text, sensor measurements, biological data
task T	:	estimating probabilities, predicting object label, dimensionality reduction, clustering
performance P	:	probability of success, money/time saved,

When to "Learn"

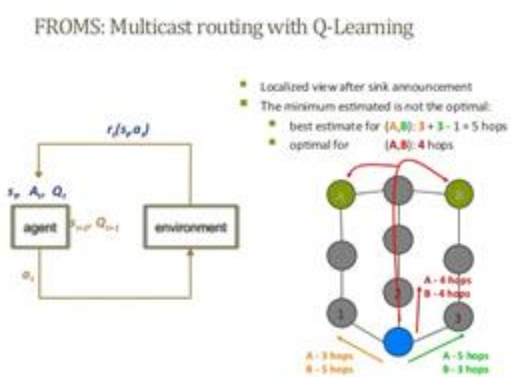


Human expertise does not exist

('learning' to navigate on Mars)

Humans unable to explain their expertise

('learning' to understand speech)



Solution changes over time

('learning' to route network packet traffic)

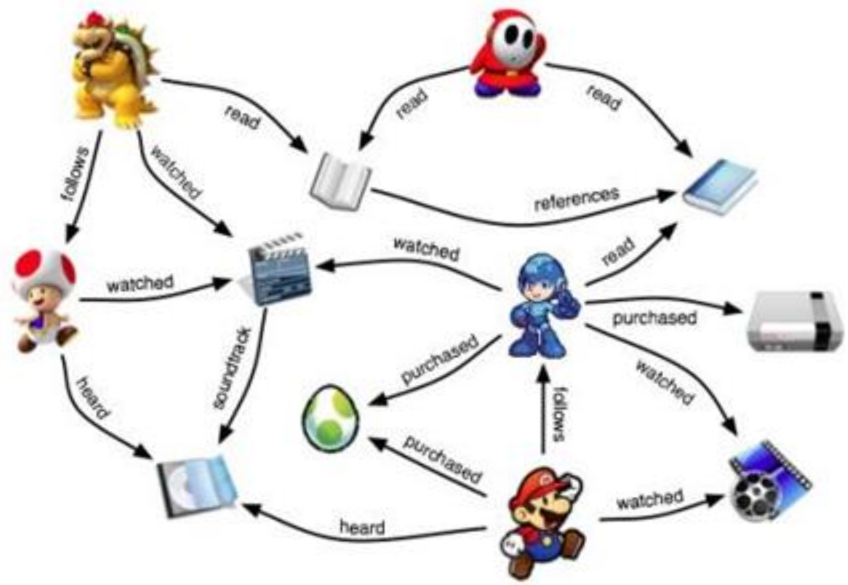
Solution needs to be adapted to particular cases

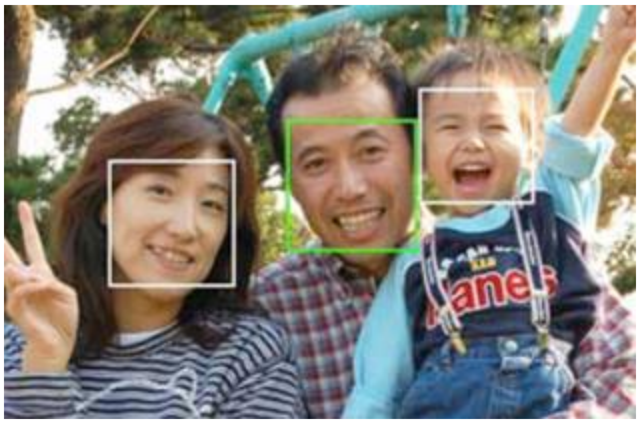
(user-specific 'learning')

If you have data, use Machine Learning!

Applications of ML

E-biz/social Media analytics



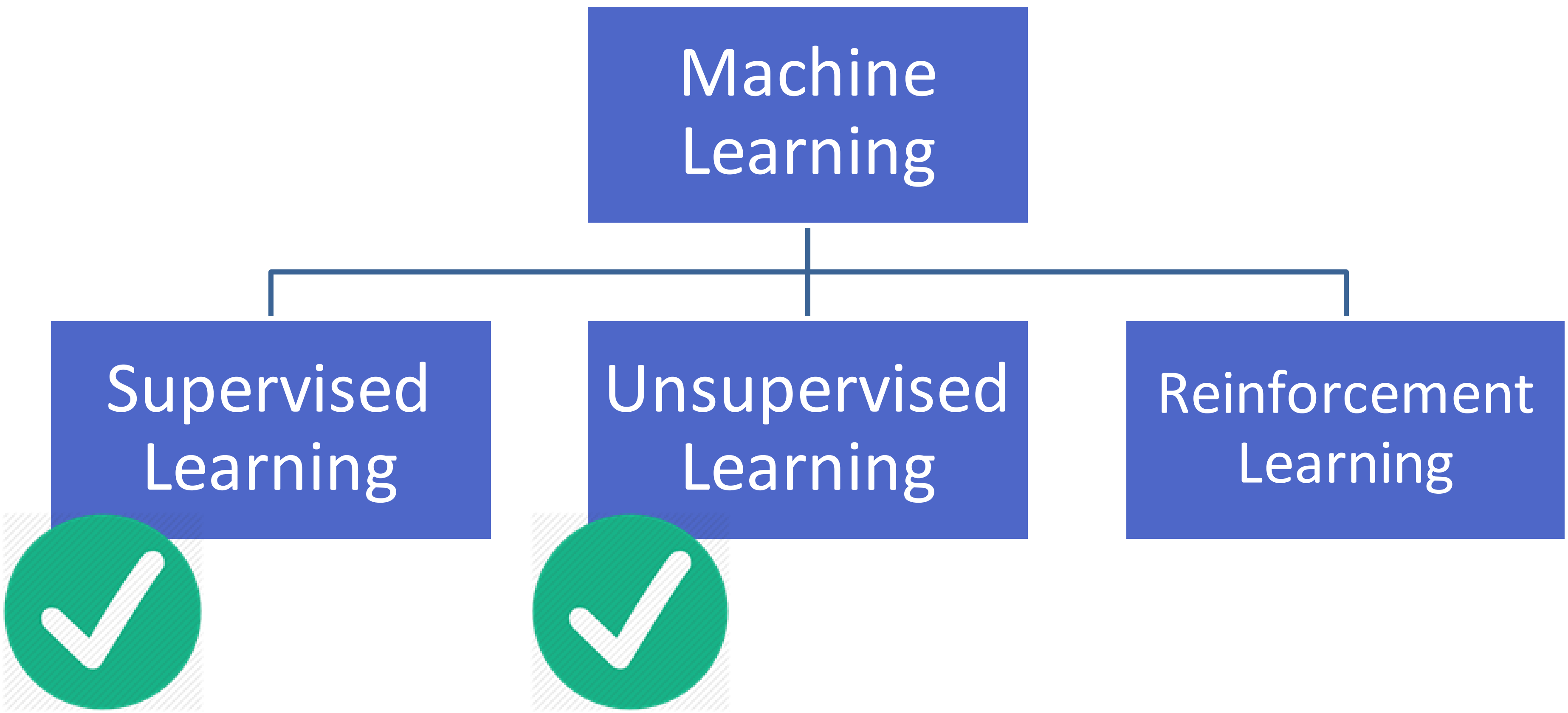


Face detection

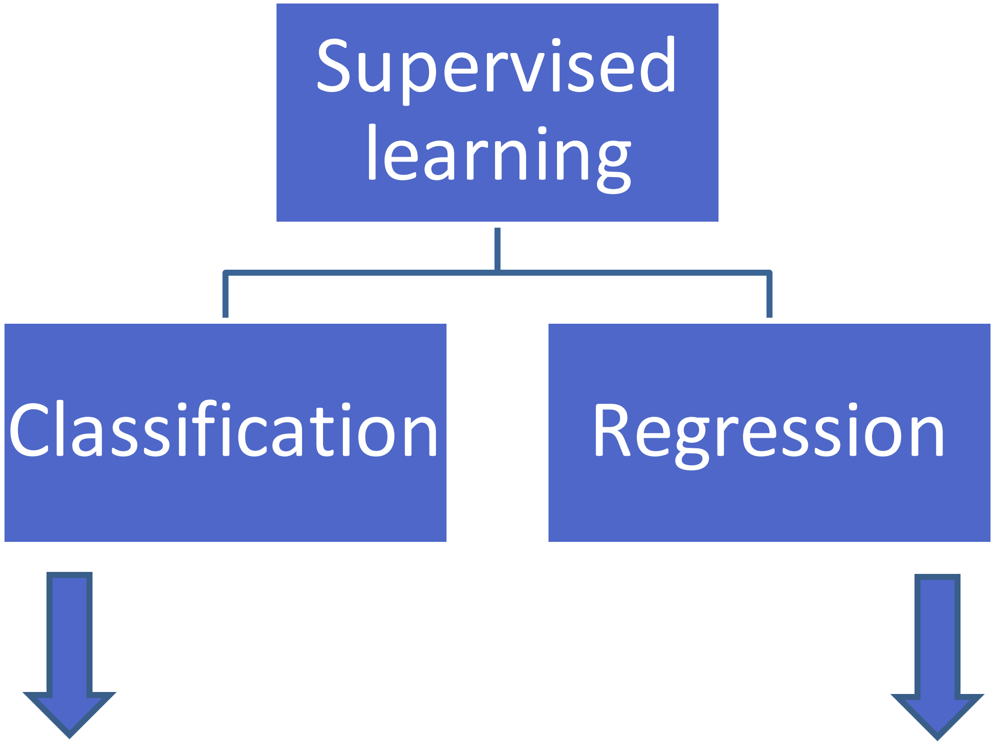
OK, Google !

Spam Filtering

Types of Learning



Types of Supervised learning



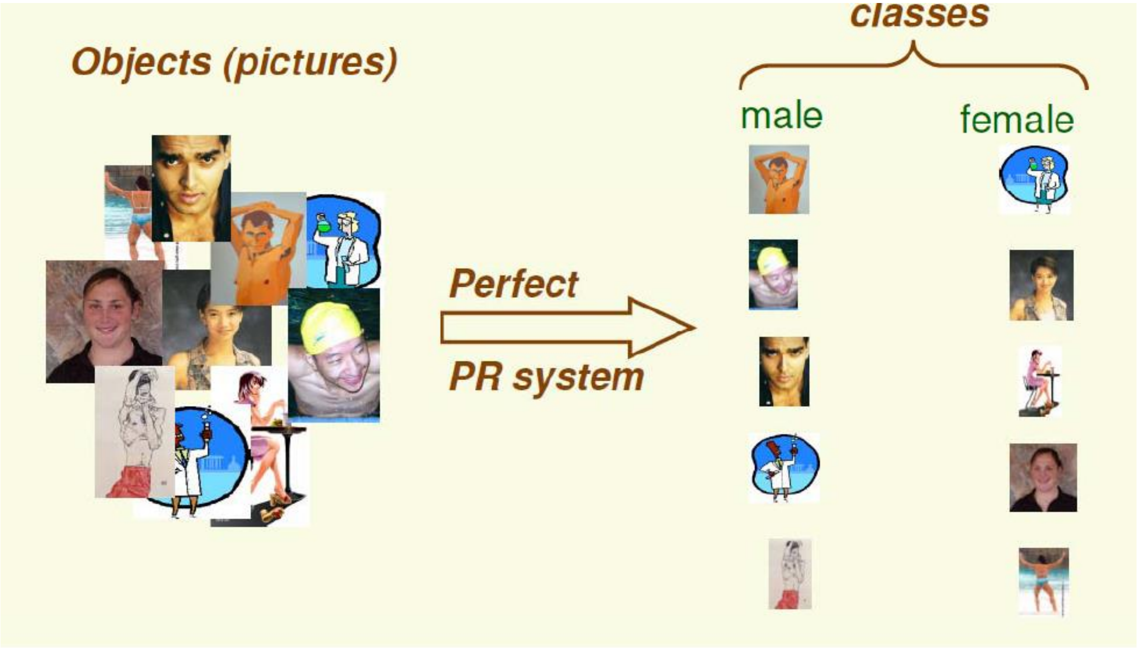
"Categorical labels"
Eg: 'cat', 'dog',..

"Continuous labels"
Eg: 0.5, -1.7, …

Mapping an input to a target !

What is Pattern Recognition?

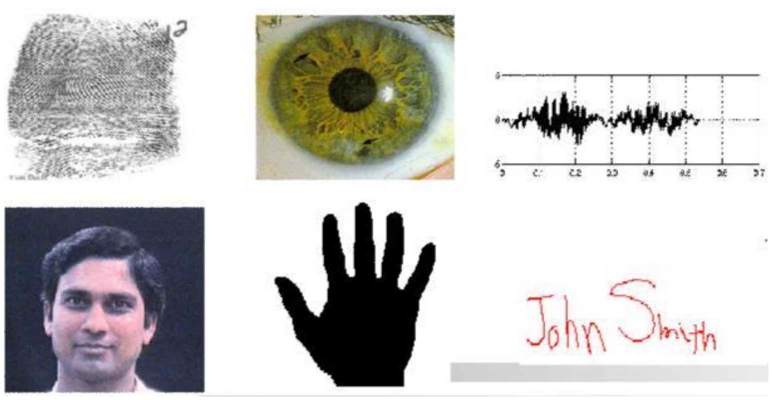
- Assign an unknown pattern to one of several known categories (or classes).



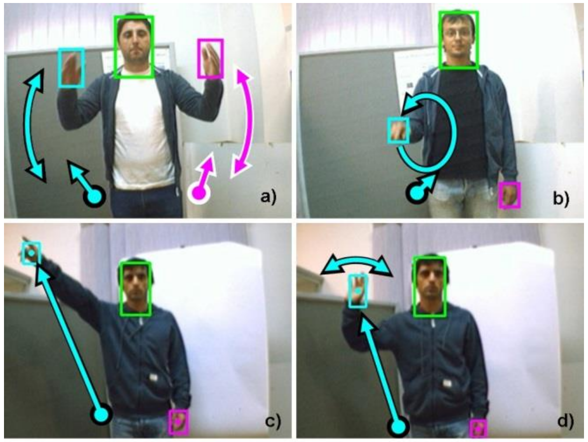
What is a Pattern?

- A pattern could be an object or event.

biometric patterns



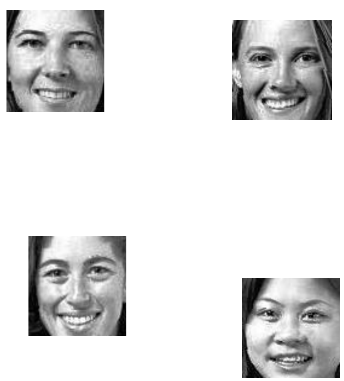
hand gesture patterns



Pattern Class

- A collection of "similar" objects.

Female

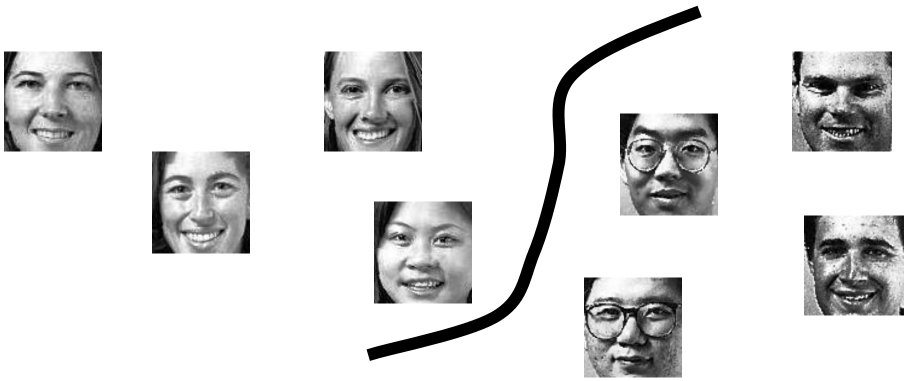


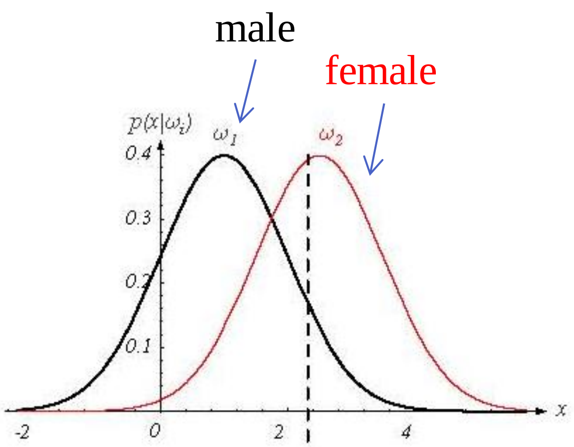
Male

How do we model a Pattern Class?

- Typically, using a statistical model.
  - probability density function (e.g., Gaussian)

Gender Classification



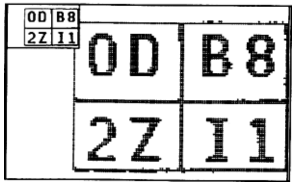


How do we model a Pattern Class? (cont'd)

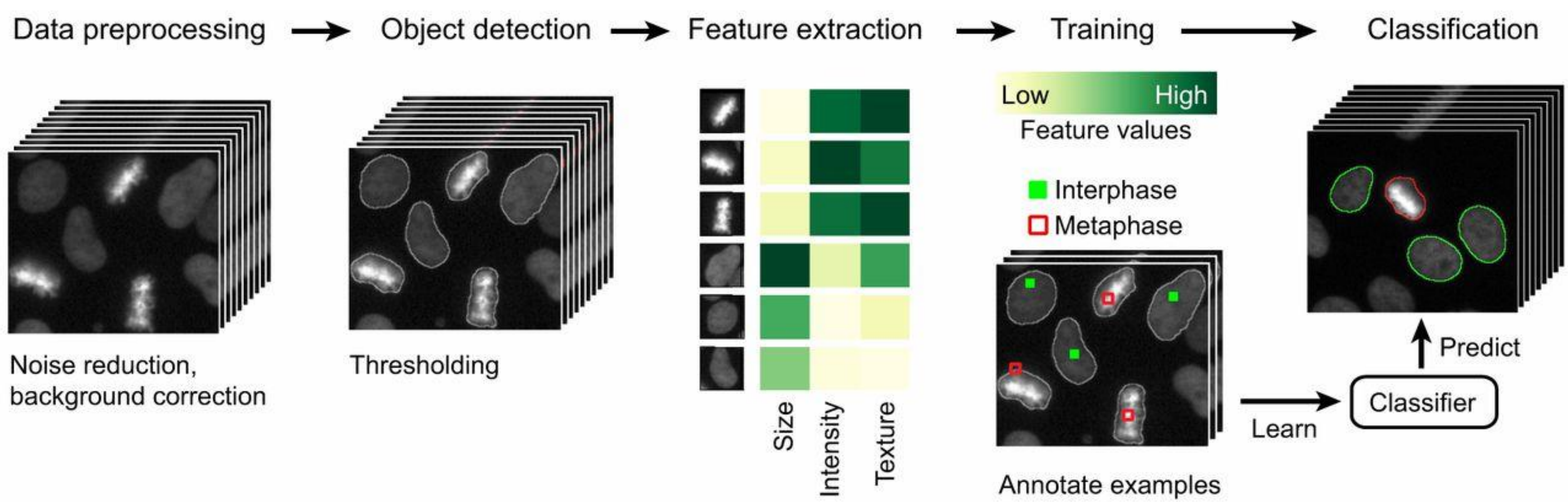
- Key challenges:
  - Intra-class variability

The letter "T" in different typefaces

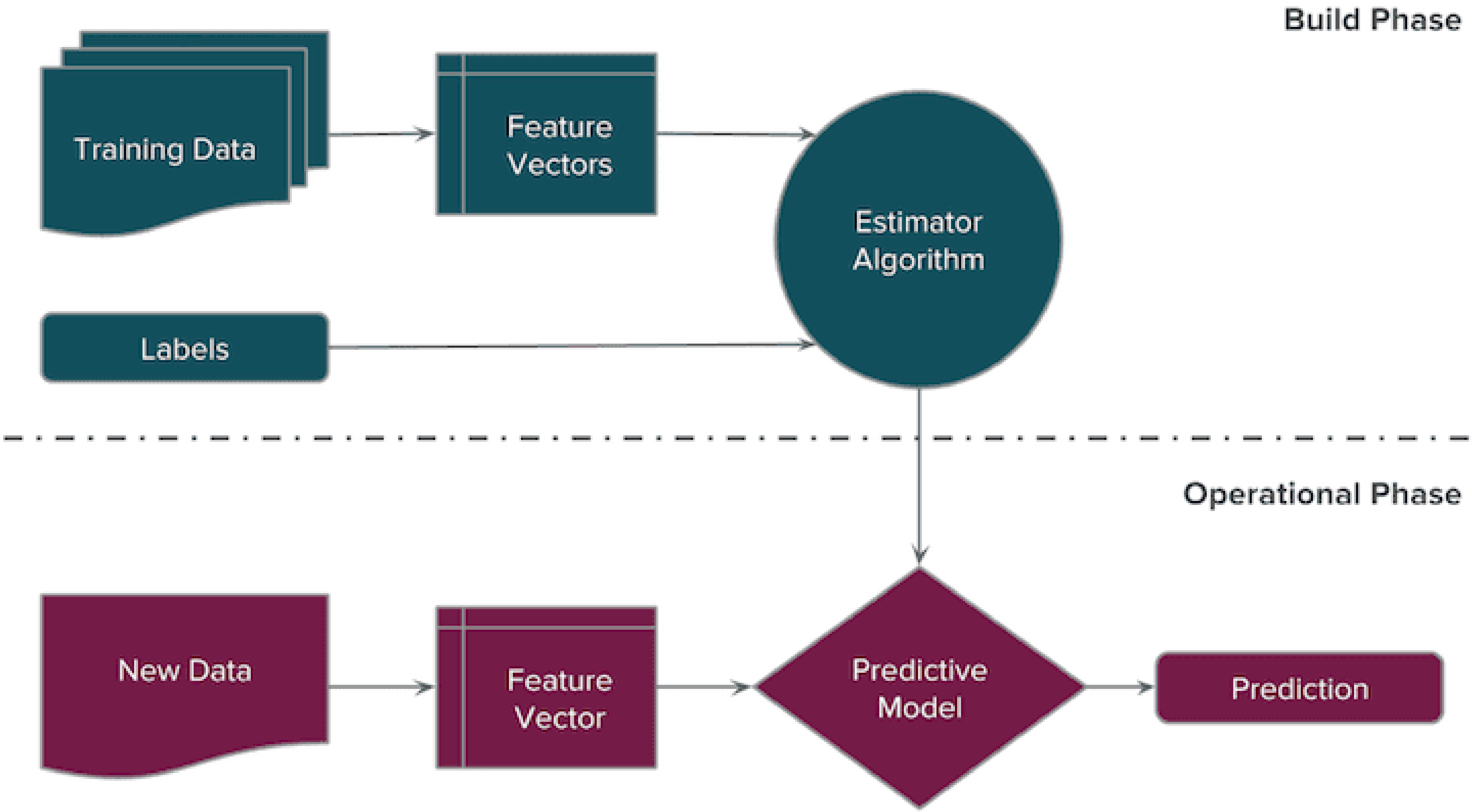
  - Inter-class variability



Letters/Numbers that look similar

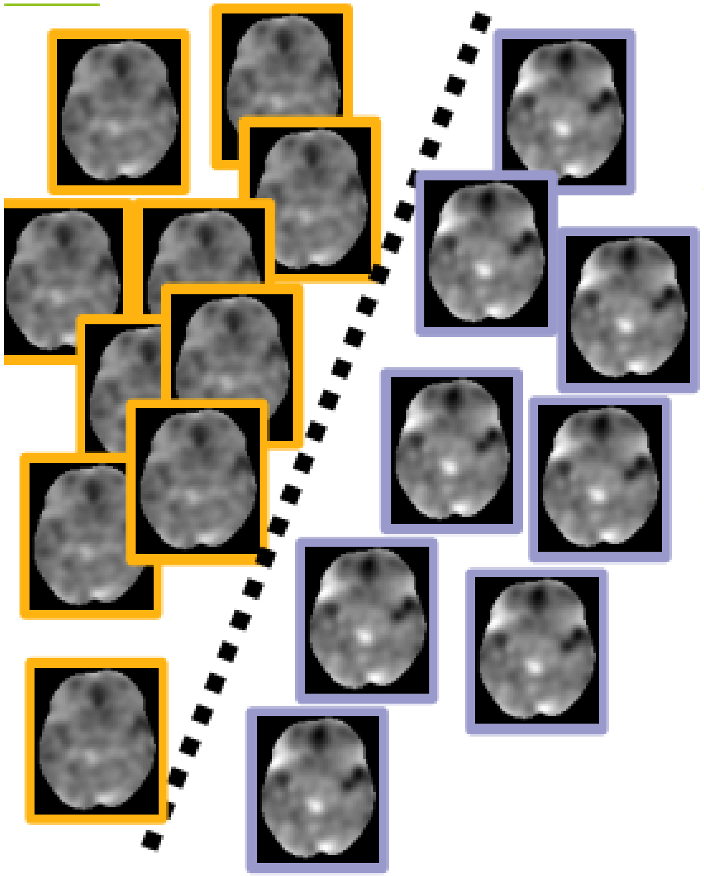


http://jcs.biologists.org/content/joces/126/24/5529/F1.large.jpg



http://dataaspirant.com/wp-content/uploads/2017/06/machine_learning_workflow.png

101 Classification example



The **objective** is to be able to **predict** ■ or ■ given an fMRI activation map

■		■
Patient	vs.	Controls
Faces	vs.	Houses
...	vs.	...
1	vs.	-1

ie. $y = \{-1, 1\}$

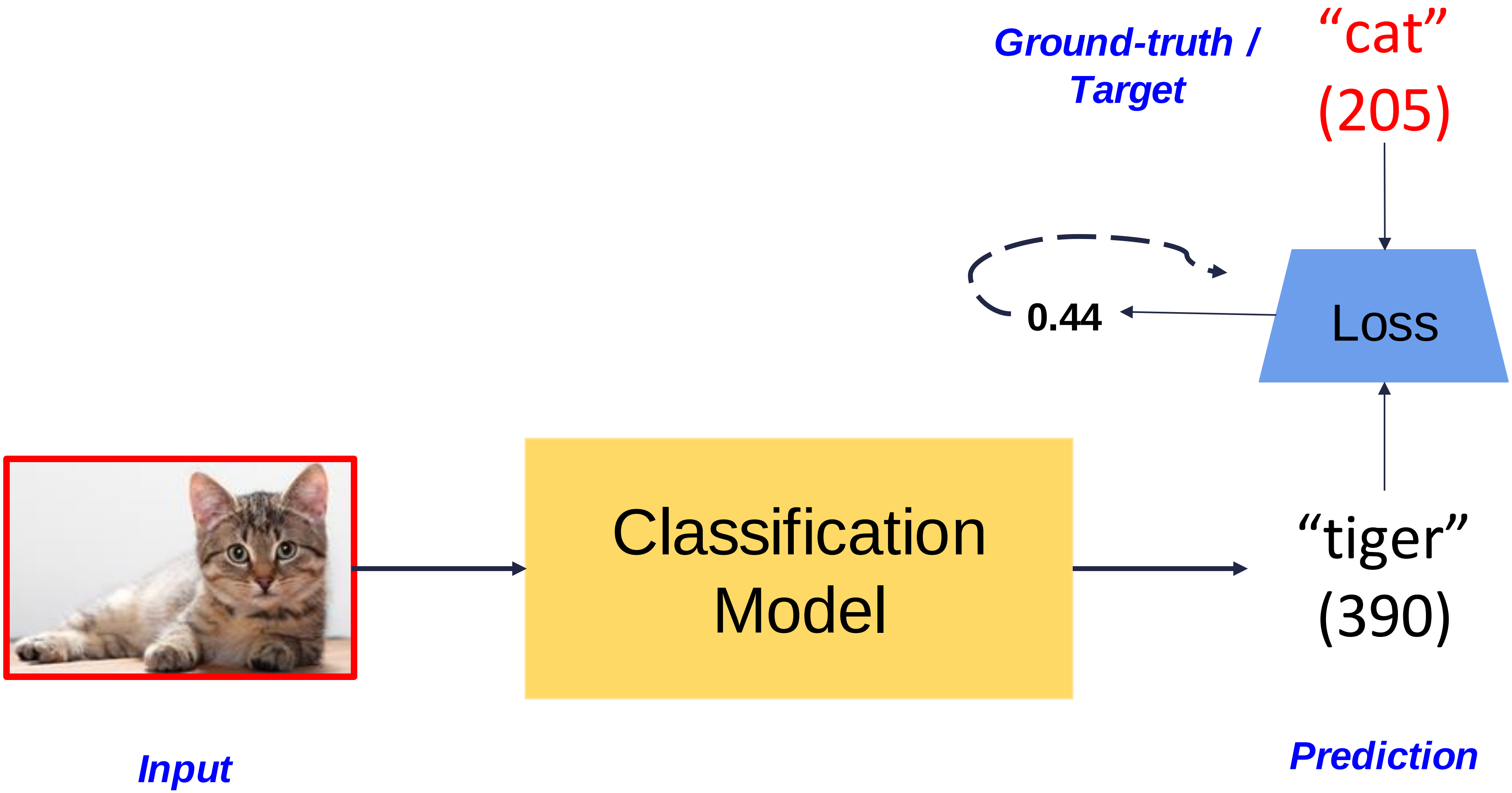
objective: Predict $y = \{-1, 1\}$ given $x \in \mathbb{R}^p$

Supervised Learning / Classification

= Cat (1)

= Dog (2)

Supervised Learning / Classification



Supervised Learning / Classification

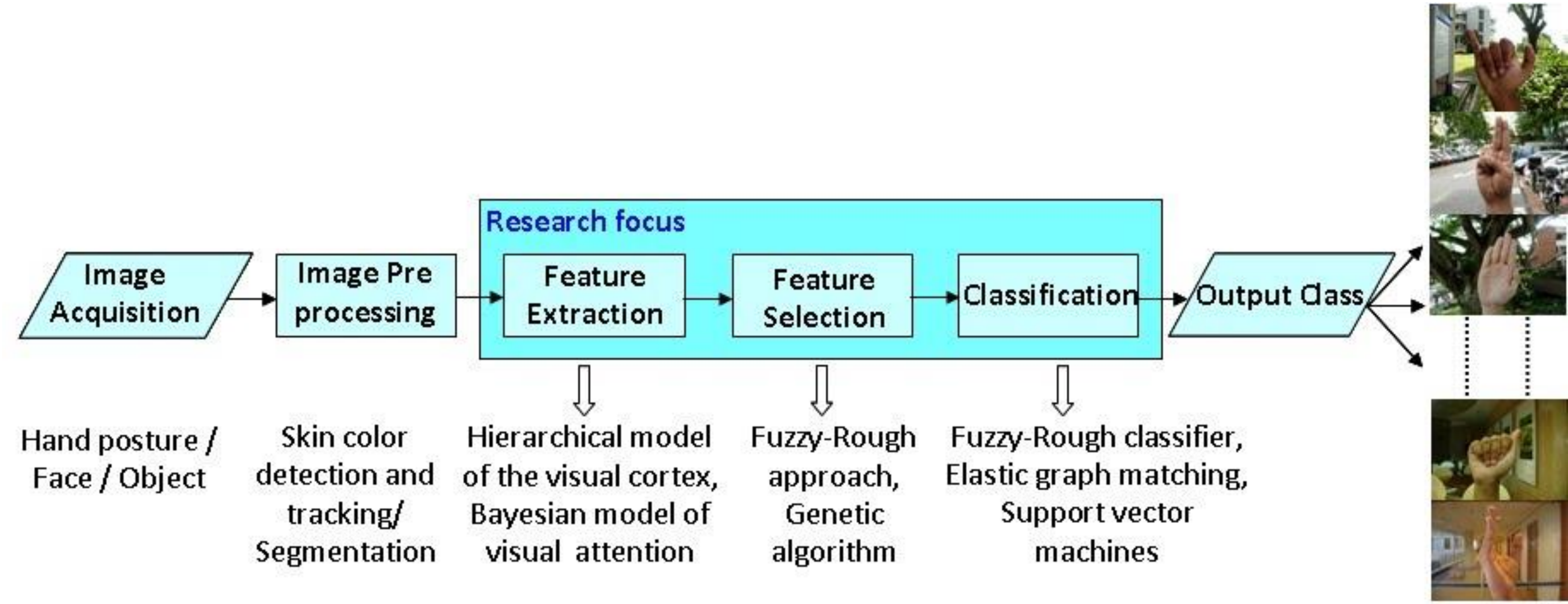
- Components
  - **Input Data**
    - “cat picture”
  - **Paired output Data**
    - 227 -- {1,2,...227,..1000}
  - **Model**
    - “Deep Neural Network”
  - **Loss**
    - How to penalize a mis-match ? (e.g. what is the cost of predicting label as “tiger”, cost for prediciting as “car”)

Multi-class supervised classification

*https://miro.medium.com/max/908/1*toRnJNryryhKETfBqOcvVg.png*

Supervised Learning / Classification

- Classification : Output is
  - … a discrete set {1,2, … ,K}
  - … with limited outcomes (1-of-K)
  - … variant: multi-label (m-of-K)



http://www.pramodkp.com/_/rsrc/1440453983012/research/research.jpg

Types of Supervised learning

Supervised learning

Classification

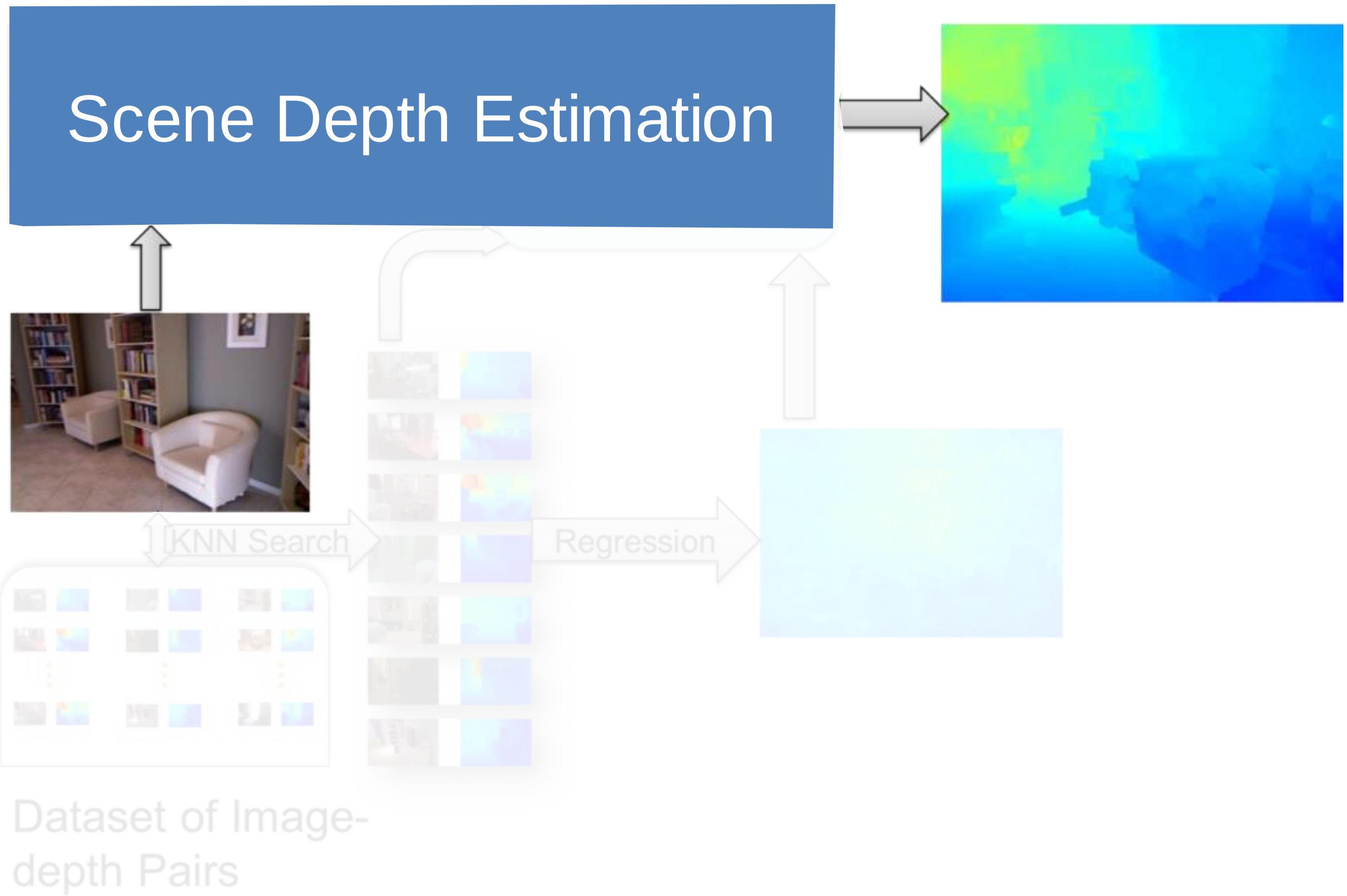
Regression

“Categorical labels”
Eg: ‘cat’, ‘dog’,..

“Continuous labels”
Eg: 0.5, -1.7, …

Mapping an input to a target !

Regression



https://research.csiro.au/data61/wp-content/uploads/sites/85/2015/02/OverView-1024x681.png

Examples

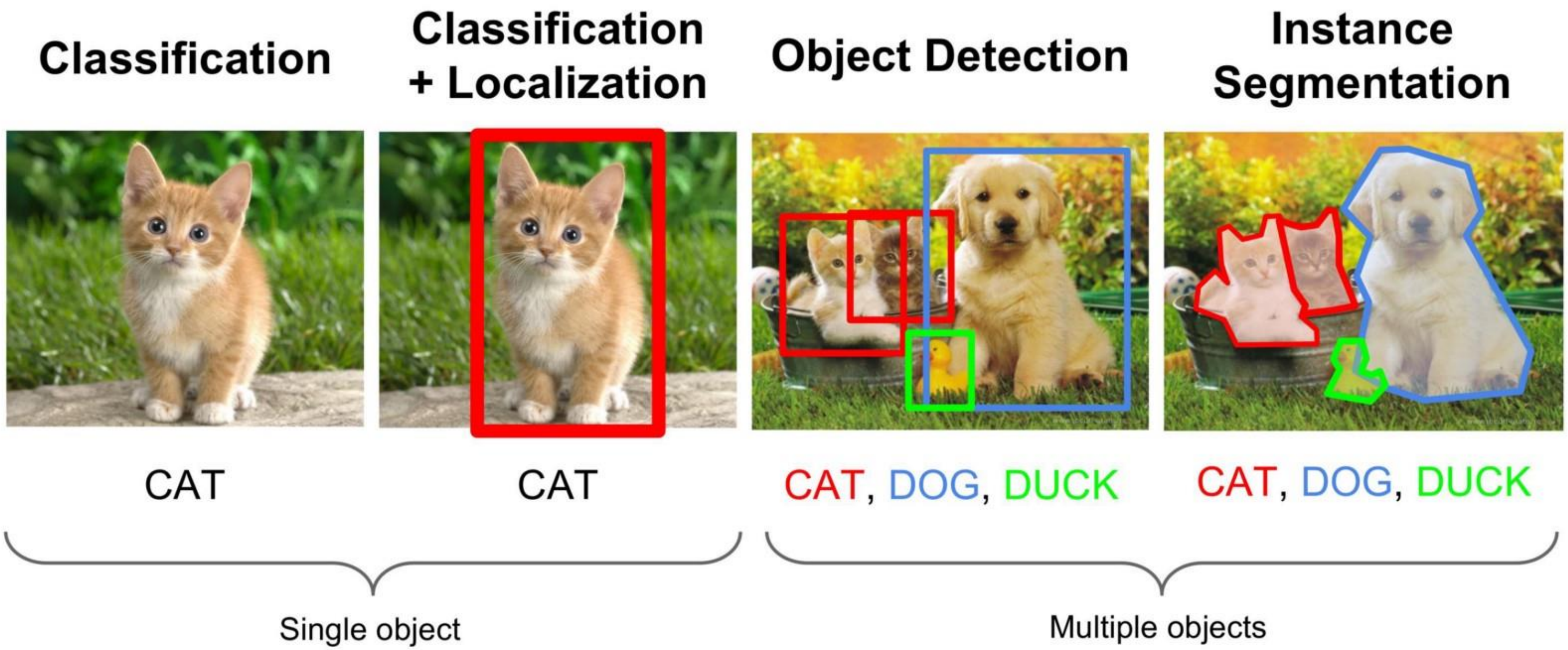
X	y
MRI volume (cortical thickness, brain/structures volumes, etc.)	Sick or Healthy? Clinical score (e.g. AD/MCI/HC)
EEG signals	move cursor left or right?
fMRI or M/EEG or ...	Was the stimulus A or B?

With $y \in \mathbb{R}$ it is a regression problem

With $y \in \{0, 1\}$ or $y \in \{-1, 1\}$: binary classification

With $y \in \{1, 2, \ldots K\}$: multi-class classification

With $y \in \{1, 2, \ldots K\}$ ordered: ranking / ordinal regression

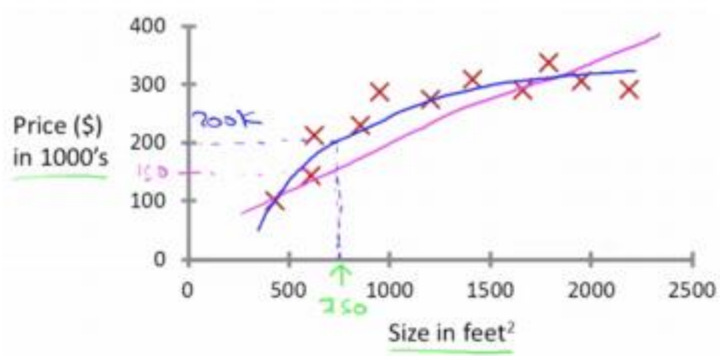


https://cdn-images-1.medium.com/max/1600/1*Hz6t-tokG1niaUfmcysusw.jpeg

REGRESSION	CLASSIFICATION

REGRESSION

Predict a Continuous Variable



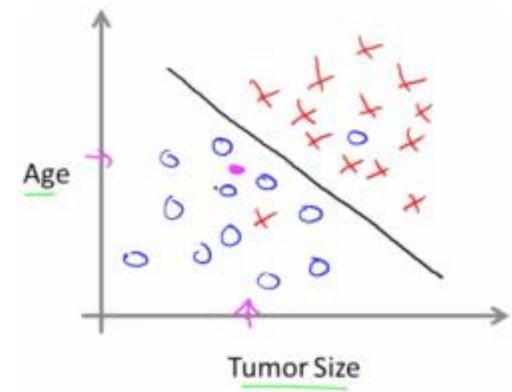
$f_w(x_1,..x_n)$ = Real Number ?

Independent Variables

Dependent Variable

CLASSIFICATION

Predict a Discrete Variable



$f_w(x_1,..x_n)$ = Class ID

Features

Label

Measuring the quality of a prediction

- We have to define a **loss** function : $\ell(y, \hat{y})$
- Measures how close y is from the predicted output $\hat{y}$
- The smaller the better
- Typical losses:
  - Classification: $\ell(y, \hat{y}) = \mathbf{1}_{y \neq \hat{y}}$ zero-one loss
  - Regression: $\ell(y, \hat{y}) = |y - \hat{y}|^2$ squared loss

Some more definitions

- A prediction function : mapping from the input to output space

$$f : X \rightarrow f(X)$$

- The quality of a prediction function is given by:

$$R(f) = \mathbb{E}_P[\ell(y, f(X))]$$

- $R(f)$: the risk of f (the <u>loss you get of average</u>)

Metrics

Classification metrics

See the Classification metrics section of the user guide for f

`metrics.accuracy_score` (y_true, y_pred[, ...])
`metrics.auc` (x, y[, reorder])
`metrics.average_precision_score` (y_true, y_score)
`metrics.brier_score_loss` (y_true, y_prob[, ...])
`metrics.classification_report` (y_true, y_pred)
`metrics.confusion_matrix` (y_true, y_pred[, ...])
`metrics.f1_score` (y_true, y_pred[, labels, ...])
`metrics.fbeta_score` (y_true, y_pred, beta[, ...])
`metrics.hamming_loss` (y_true, y_pred[, classes])
`metrics.hinge_loss` (y_true, pred_decision[, ...])
`metrics.jaccard_similarity_score` (y_true, y_pred)
`metrics.log_loss` (y_true, y_pred[, eps, ...])
`metrics.matthews_corrcoef` (y_true, y_pred)
`metrics.precision_recall_curve` (y_true, ...)
`metrics.precision_recall_fscore_support` (...)
`metrics.precision_score` (y_true, y_pred[, ...])
`metrics.recall_score` (y_true, y_pred[, ...])
`metrics.roc_auc_score` (y_true, y_score[, ...])
`metrics.roc_curve` (y_true, y_score[, ...])

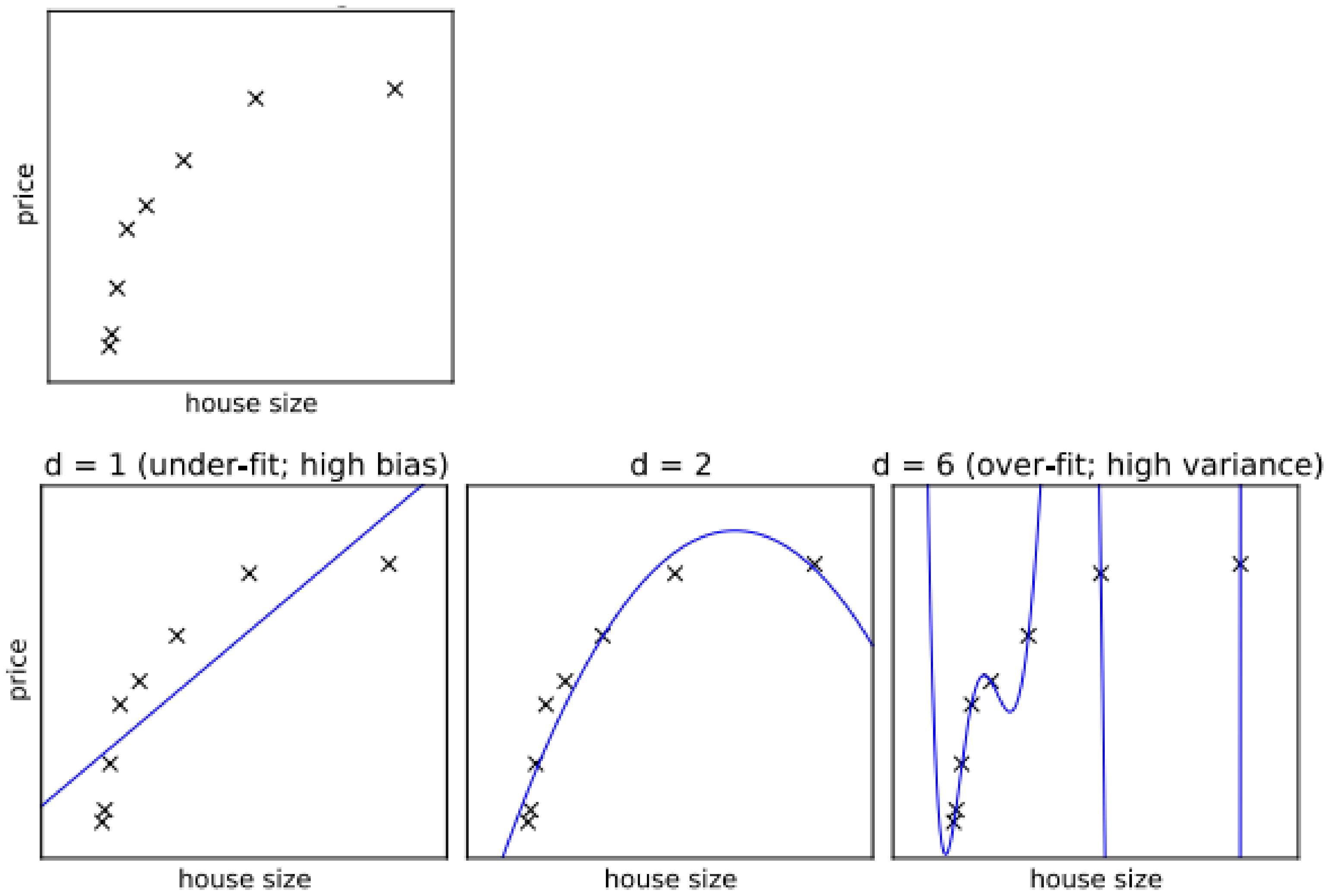
Regression metrics

See the Regression metrics section of the user guide for further deta

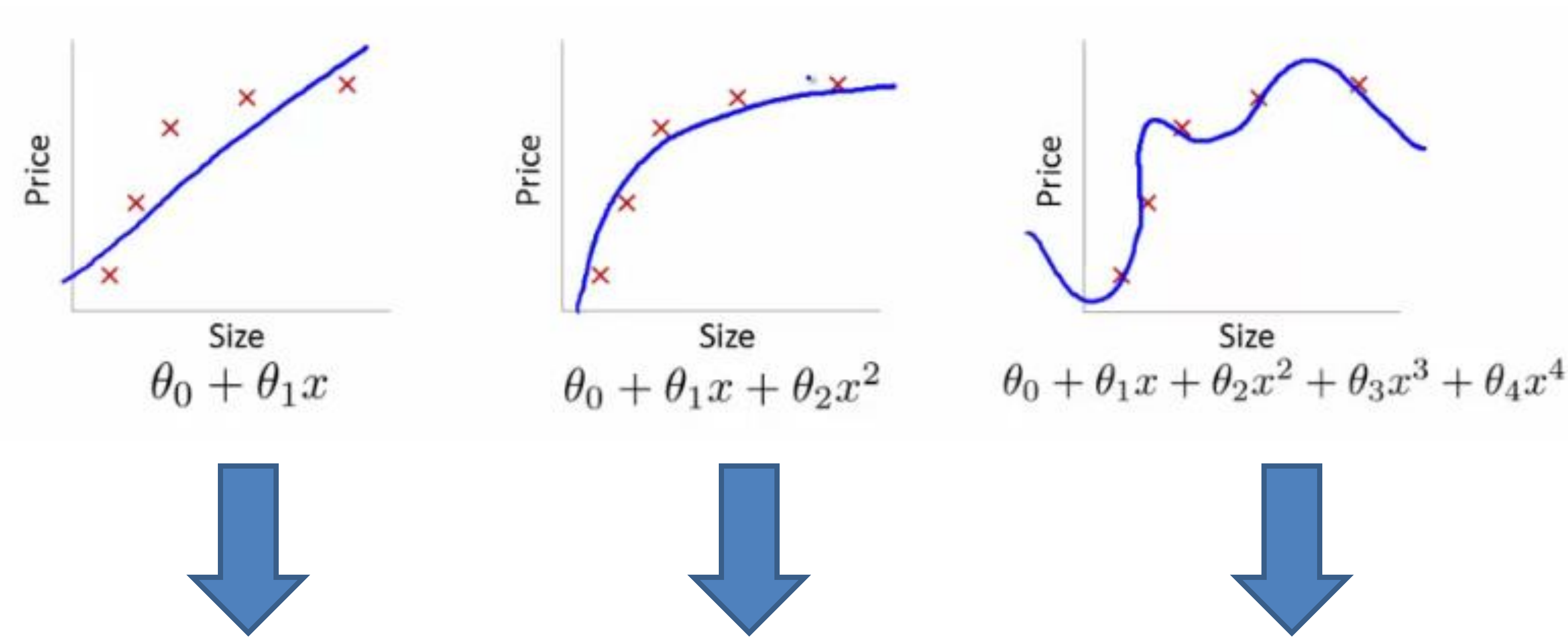
`metrics.explained_variance_score` (y_true, y_pred)	Explaine function
`metrics.mean_absolute_error` (y_true, y_pred)	Mean ab
`metrics.mean_squared_error` (y_true, y_pred[, ...])	Mean sq
`metrics.median_absolute_error` (y_true, y_pred)	Median a
`metrics.r2_score` (y_true, y_pred[, ...])	R^2 (coe regressic

(cropped) list of metrics from scikit-learn

Regression and overfitting



Overfitting and Underfitting



Underfitting

"Just right"

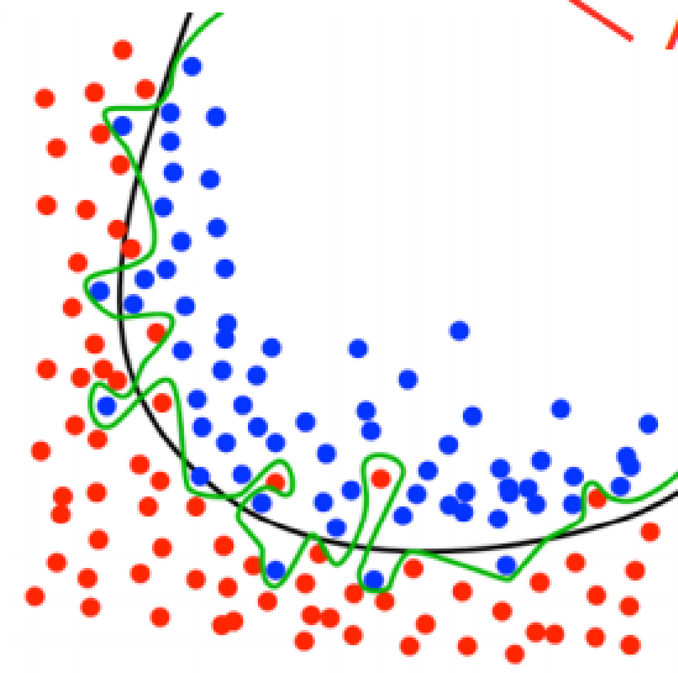
Overfitting

Classification and overfitting

- Empirical risk minimization:

$$\hat{f} \in \arg\min_{\underline{f\in\mathcal{F}}} \frac{1}{n}\sum_i \mathbf{1}_{y_i\neq f(X_i)}$$

Allowed models should be in a limited set



Good model in $\mathcal{F}$

Too complex model

Performance should be evaluated on left-out data otherwise learn by heart!

What is this F in practice?

Constrained formulation (to limit the space of f):

$$\underset{\underline{f \in \mathcal{F}}}{\arg\min} \frac{1}{n} \sum_i \ell(y_i, f(X_i))$$

where $\underline{\mathcal{F}} = \{f / \|f\| \leq \tau\}$

Variational formulation:

$$\arg\min_f \frac{1}{n} \sum_i \ell(y_i, f(X_i)) + \lambda \|f\|$$

Error on data Regularization

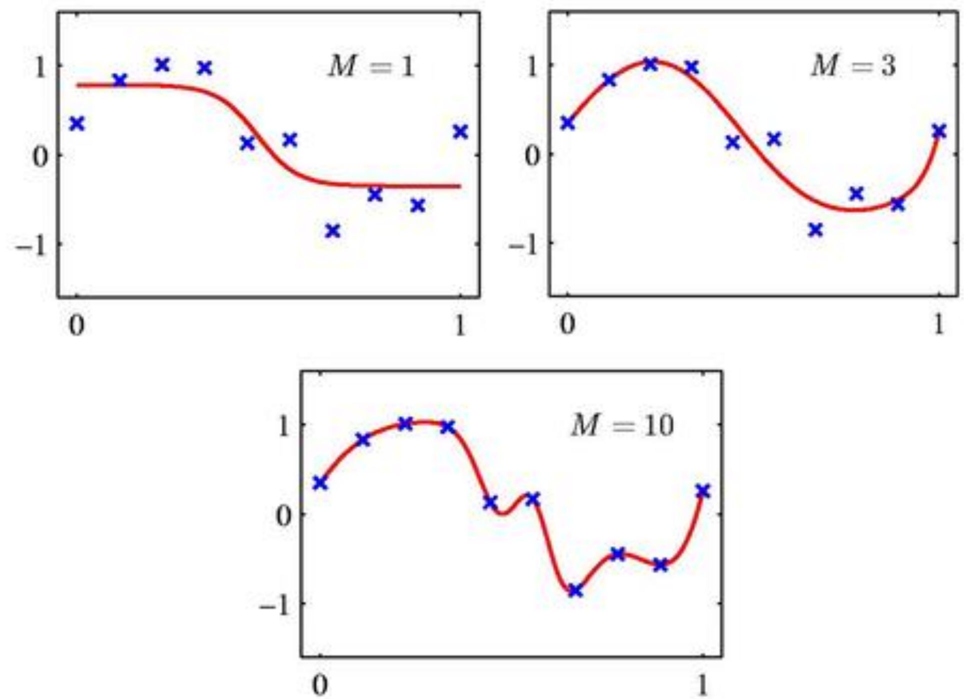
The large λ leads to simpler models

Remark: λ is what we call an hyperparameter

Regularization

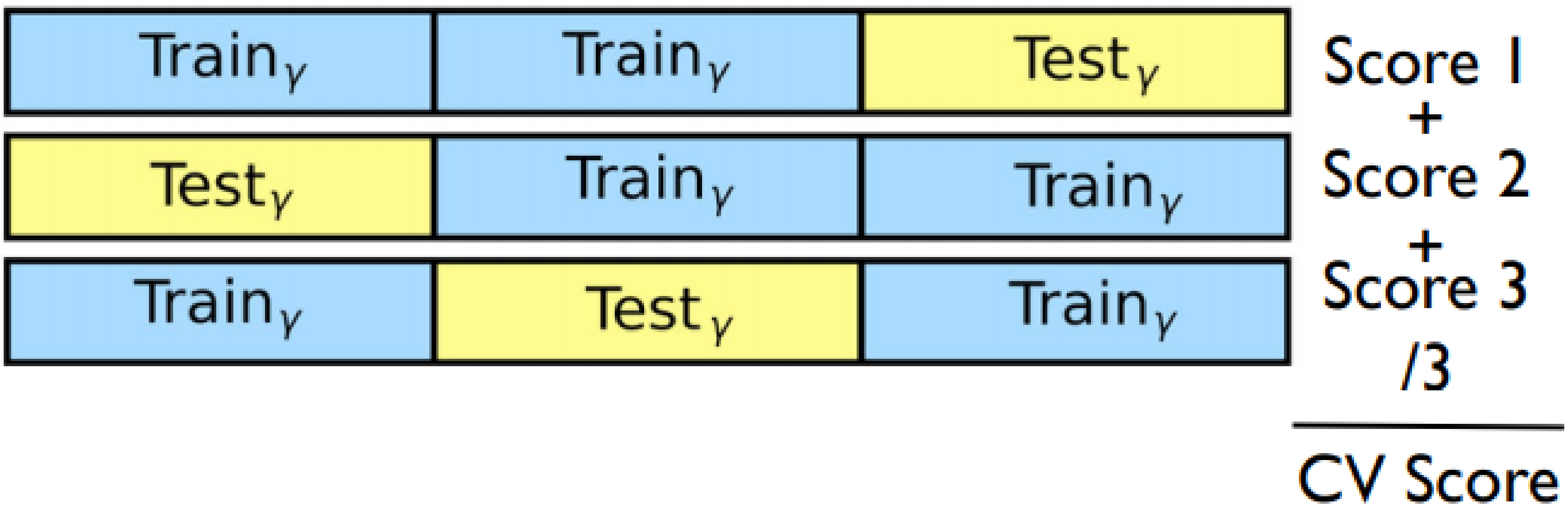
$$cost(a, b) = \sum_i (y_i - h(x, \theta))^2 + \frac{1}{M}\|\theta\|^2$$

Crucial aspect of many modern ML systems!



Cross-validation (CV)

Example with 3-Folds CV



Remark: You should report the average score and the standard deviation

Training vs. Test vs. Validation error

Full dataset

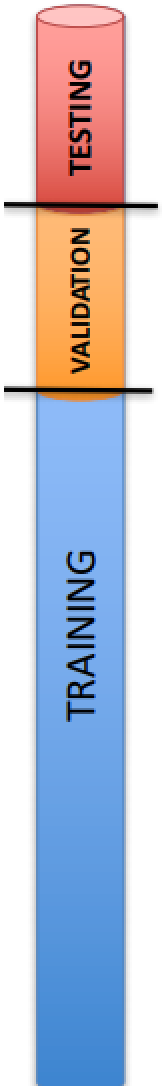
~~Learning hyperparameters with test set~~

~~Training | Test~~

Learning hyperparameters with validation set and use test for test !

Training | Validation | Test

To reliably assess the performance of your model don't optimize hyperparameters on test set



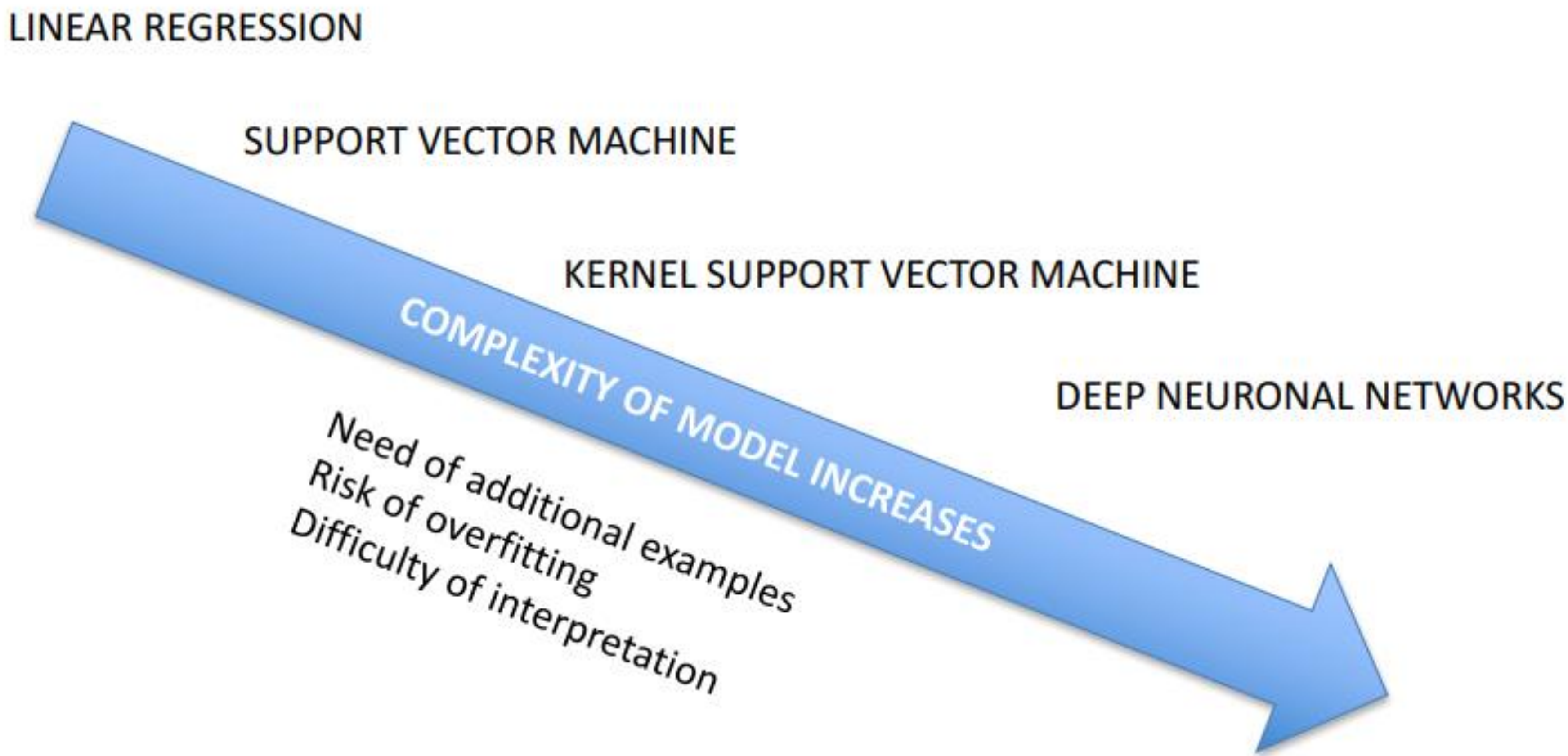
OPTION 3 (MODEL SELECTION PROBLEM):

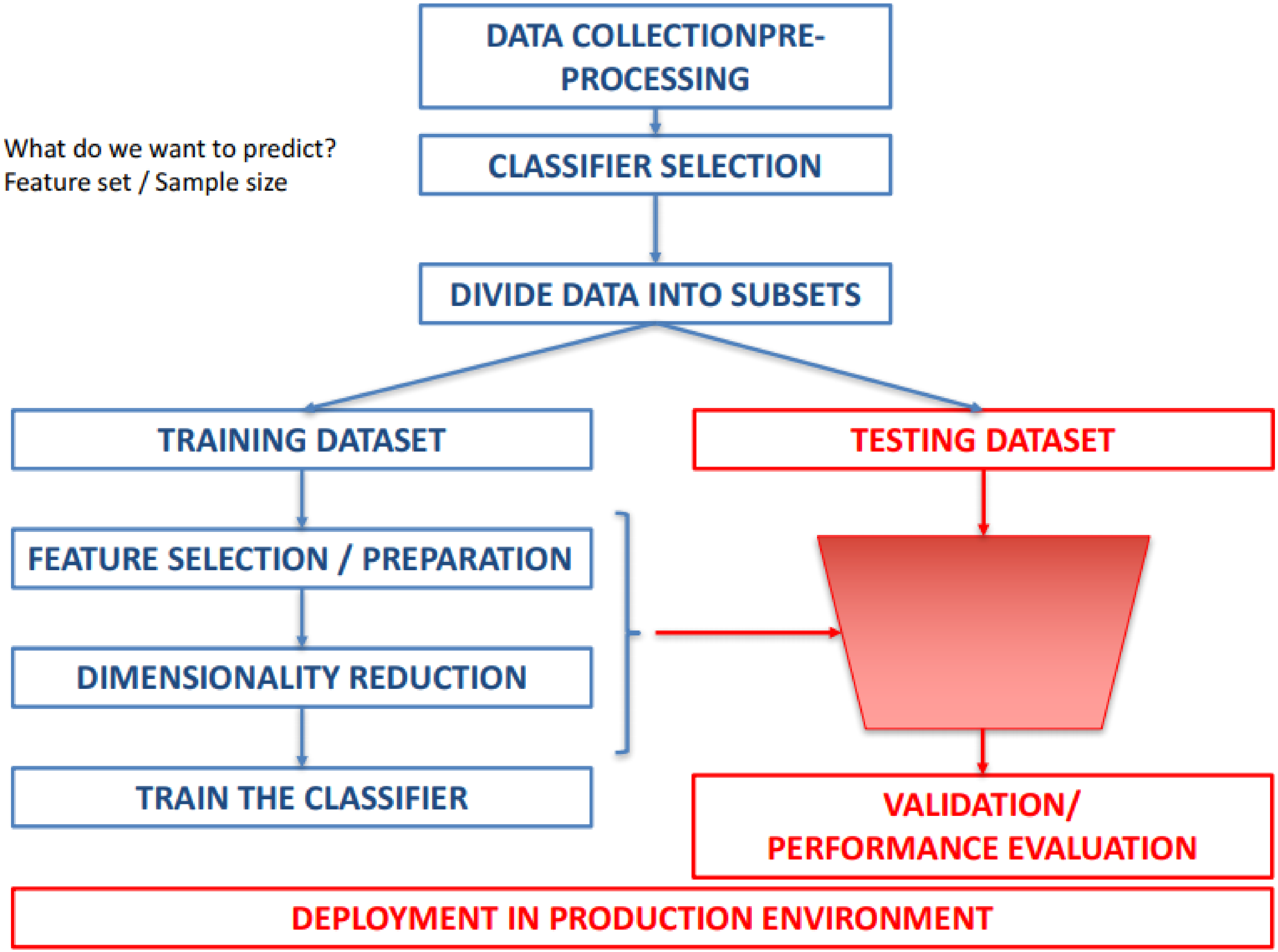
- We want to train, but also do some sort of model selection.
- Example: Not sure which one of three linear models to use

$$\textit{Degree } d=1 \rightarrow f_w(x) = w_o + w_1x_1$$
$$\textit{Degree } d=2 \rightarrow f_w(x) = w_o + w_1x_1 + w_2x_1^2$$
$$\textit{Degree } d=3 \rightarrow f_w(x) = w_o + w_1x_1 + w_2x_1^2 + w_3x_1^3$$

In addition to training each model, we want to automatically pick d

- We need to subdivide our dataset in three subsets:
  - **TRAINING**: We use this to train all models (estimate w^T for all models)
  - **VALIDATION**: We use this to select the best model
  - **TESTING**: We use this to estimate final performance (generality)


LINEAR REGRESSION
SUPPORT VECTOR MACHINE
KERNEL SUPPORT VECTOR MACHINE
DEEP NEURONAL NETWORKS
COMPLEXITY OF MODEL INCREASES
Need of additional examples
Risk of overfitting
Difficulty of interpretation


DATA COLLECTIONPRE-PROCESSING
What do we want to predict?
Feature set / Sample size
CLASSIFIER SELECTION
DIVIDE DATA INTO SUBSETS
TRAINING DATASET
TESTING DATASET
FEATURE SELECTION / PREPARATION
DIMENSIONALITY REDUCTION
TRAIN THE CLASSIFIER
VALIDATION/
PERFORMANCE EVALUATION
DEPLOYMENT IN PRODUCTION ENVIRONMENT

Steps to solving a supervised learning problem

1. Decide what the input-output pairs are.
2. Decide how to encode inputs and outputs.

   This defines the input space $\mathcal{X}$, and the output space $\mathcal{Y}$.
3. Choose a class of hypotheses/representations $\mathcal{H}$.
4. Choose an error function (cost function) to define the best hypothesis
5. Choose an algorithm for searching efficiently through the space of hypotheses.

Types of Learning

Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

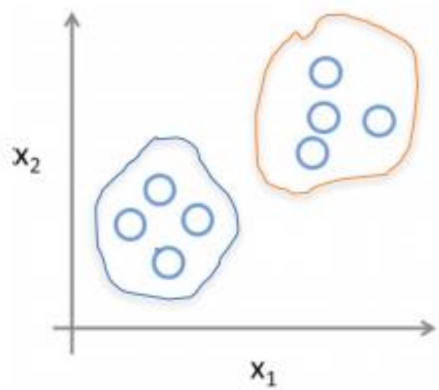
Unsupervised Learning

- No “target value”

Neural Networks practitioner

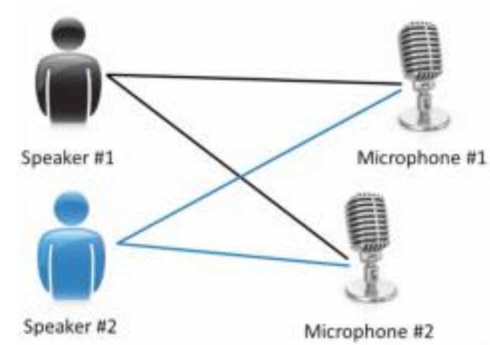
Algorithms used to draw inferences from unlabeled datasets

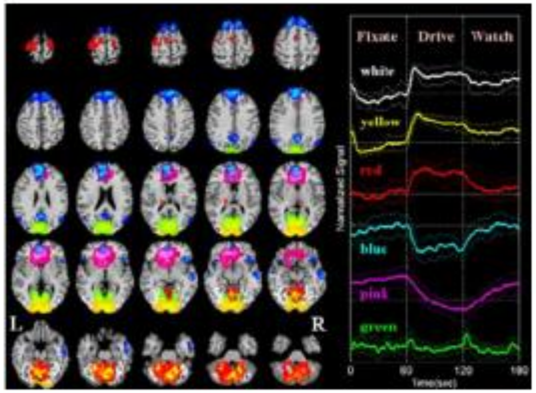
CLUSTERING ALGORITHMS



K-Means, Fuzzy K-means,
Hierarchical Clustering, DBSCAN, …

SOURCE SEPARATION





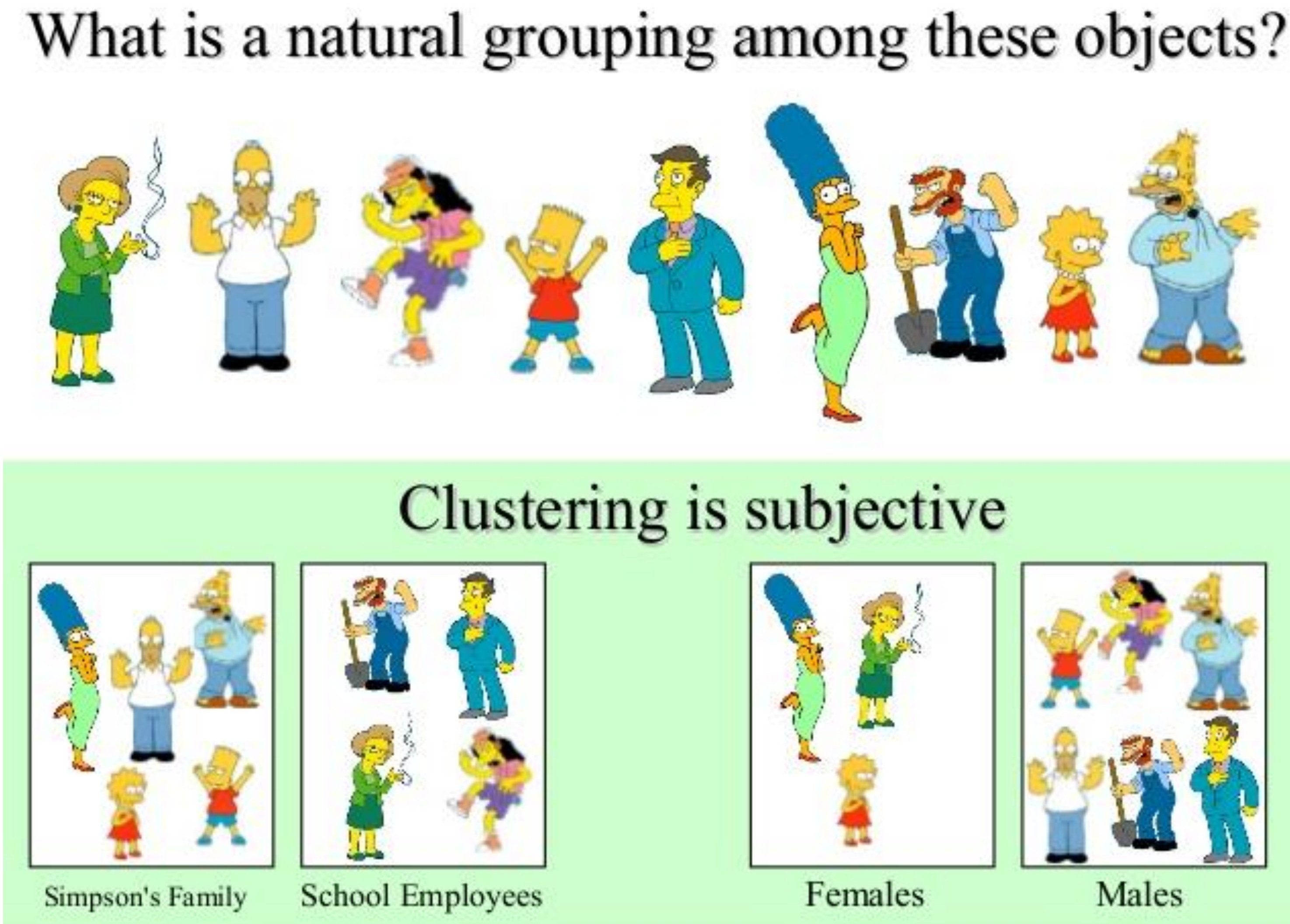
PCA, ICA, SVD, …

Unsupervised Learning

- Learning what “normally happens”
- No target value
- Example applications
  - Customer segmentation (‘Moms’, ‘Teenagers’, ‘Men’)
  - Organize photo collections ({‘Person A’, ‘Person B’}, {‘Birthday’,’Job Treat’,’Road Trip’})

Unsupervised Learning

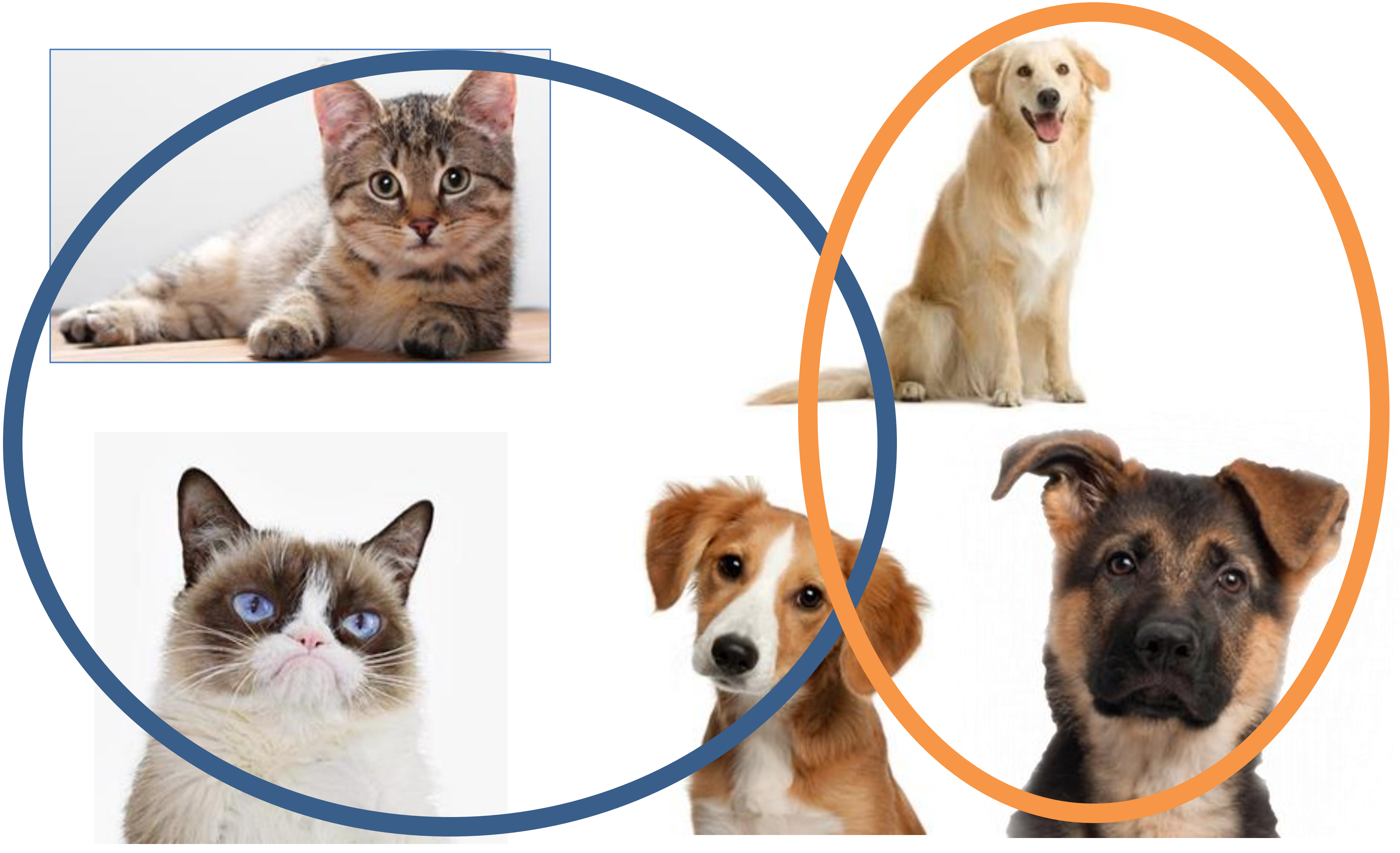
Unsupervised Learning

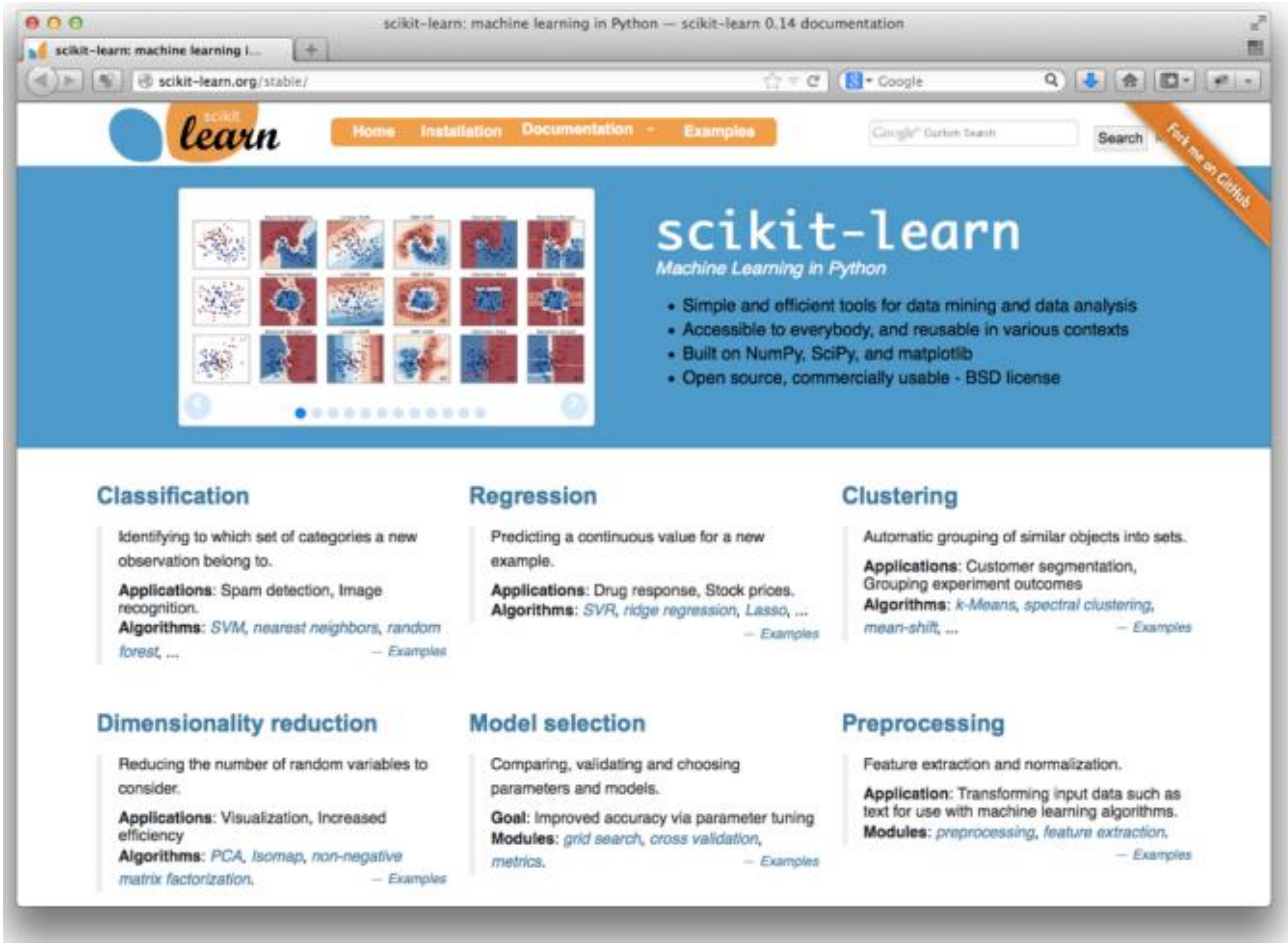


Unsupervised Learning

Unsupervised Learning

Unsupervised Learning




scikit-learn: machine learning in Python — scikit-learn 0.14 documentation
scikit-learn: machine learning i...
scikit-learn.org/stable/
Google
scikit
learn
Home
Installation
Documentation
Examples
Search
Fork me on GitHub
scikit-learn
Machine Learning in Python
• Simple and efficient tools for data mining and data analysis
• Accessible to everybody, and reusable in various contexts
• Built on NumPy, SciPy, and matplotlib
• Open source, commercially usable - BSD license
Classification
Identifying to which set of categories a new observation belong to.
Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...
— Examples
Regression
Predicting a continuous value for a new example.
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...
— Examples
Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...
— Examples
Dimensionality reduction
Reducing the number of random variables to consider.
Applications: Visualization, Increased efficiency
Algorithms: PCA, Isomap, non-negative matrix factorization.
— Examples
Model selection
Comparing, validating and choosing parameters and models.
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.
— Examples
Preprocessing
Feature extraction and normalization.
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.
— Examples

ML model ~ Learning ‘implicit’ rules

- Prediction of future cases: Use the rule to predict output for future inputs
- Knowledge extraction: Rule is easy to understand (e.g. ‘customer’ profiles)
- Compression: Rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule (e.g. fraud detection)

Classical stats vs ML

- Classical statistics (t-tests, GLM):
  - Statistical significance assessed using full datasets
  - <u>Question:</u> Is the mean of the distribution different between my 2 conditions?
  - <u>Outcome:</u> Effect size & P-values
- Statistical machine learning:
  - Performance evaluated on left-out data
  - <u>Outcome:</u> Metric above chance when evaluated on left-out and independent samples & P-values

Components of an Image Processing System:

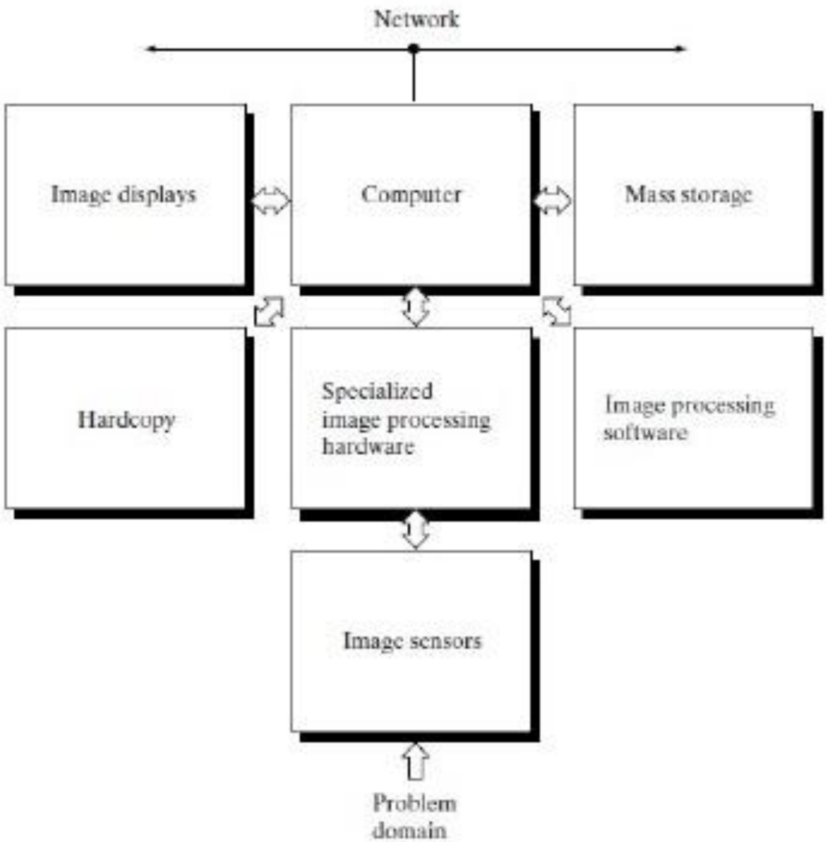


FIGURE 1.24 Components of a general-purpose image processing system.

Ref: Digital image processing ,Gonzalez & Woods

Object-recognition: conventional approach

Image → Hand designed feature extraction → Classifier → Object Class

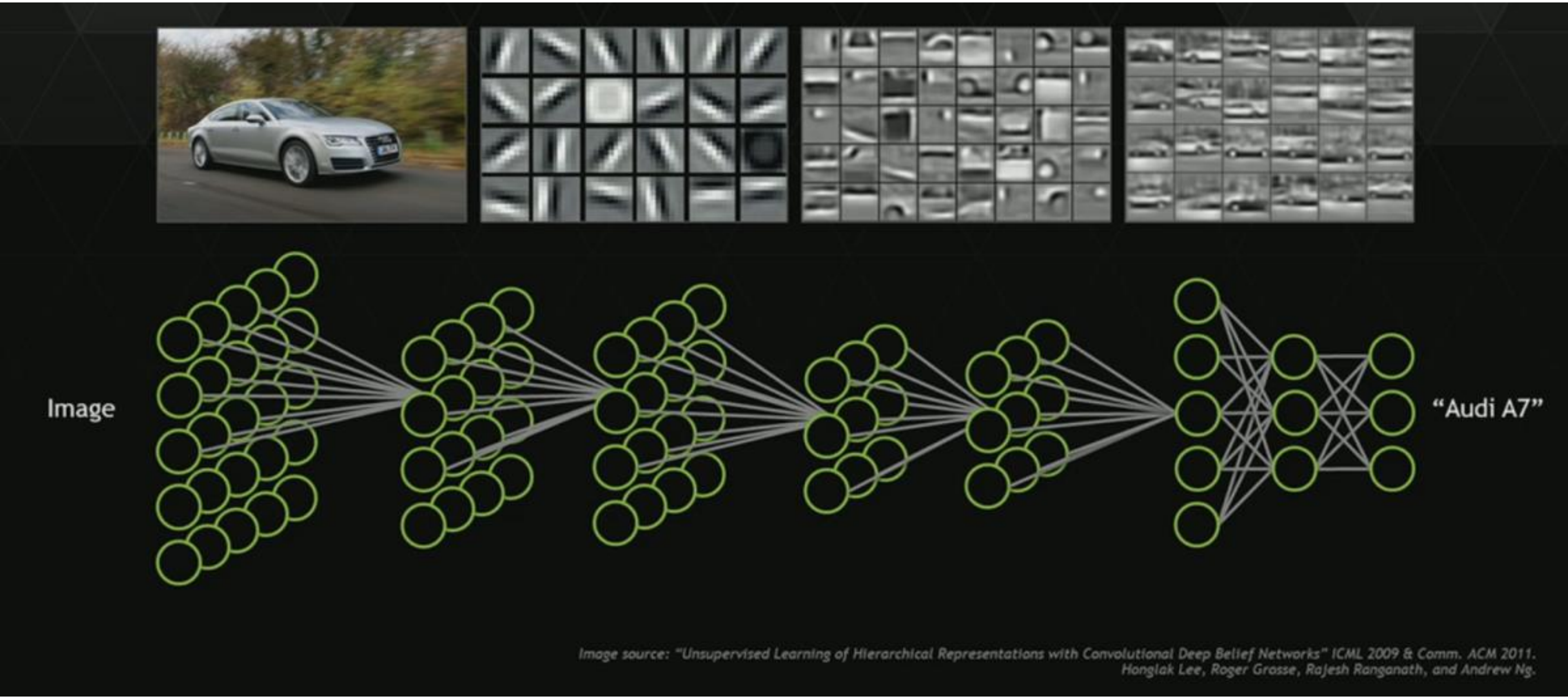
"dog"

Object-recognition: conventional approach

Image → Hand designed feature extraction → Classifier → Object Class

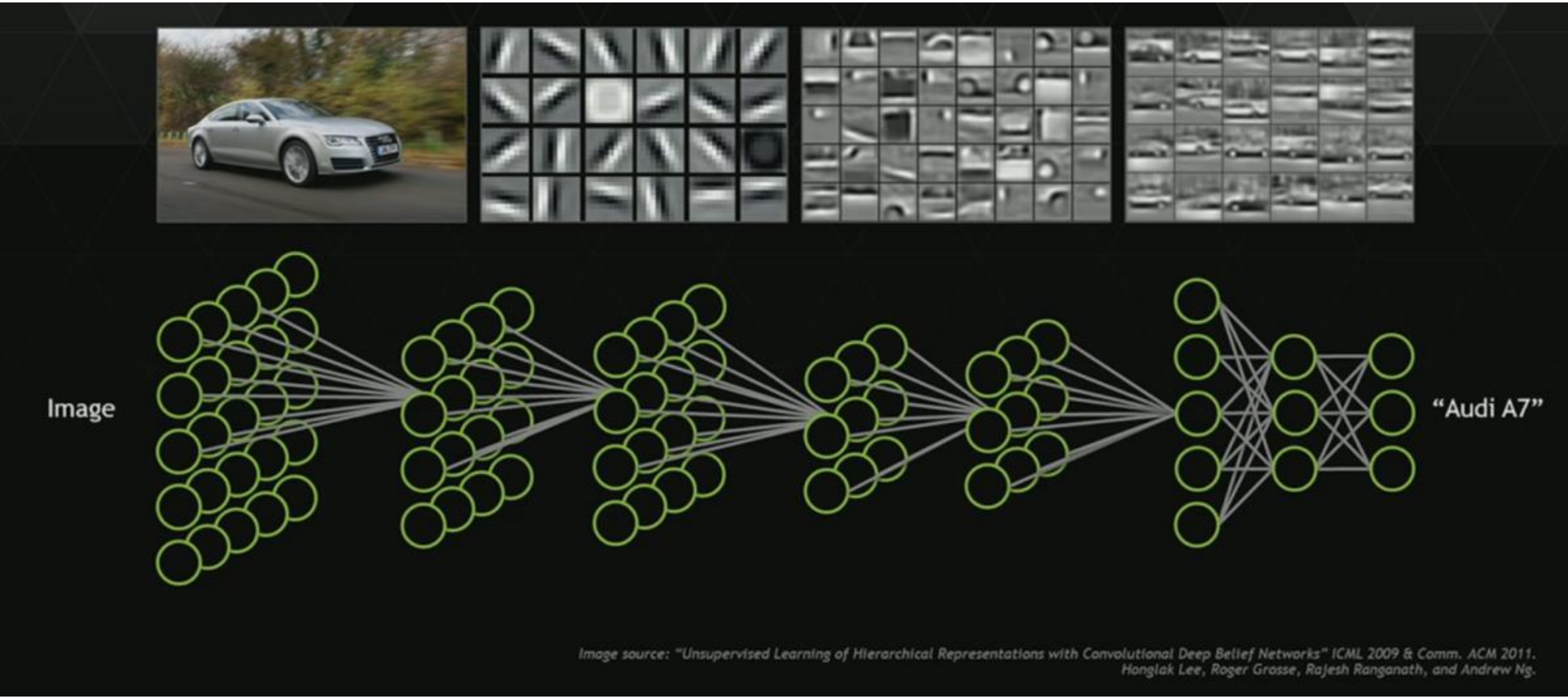
“dog”

Object Recognition: Deep Neural Networks



Data-driven, End-to-End learning

Object Recognition: Deep Neural Networks



Data-driven, End-to-End learning, Task-specific feature hierarchy

Why deep learning