Feature engineering

- **Duration:** 4 hrs
- **Outline:**

1. Introduction
2. Feature engineering
3. Features in visual pattern recognition
4. Shape-based feature descriptors

Feature engineering

- **Duration:** 4 hrs
- **Outline:**

1. Introduction

2. Feature engineering

3. Features in visual pattern recognition

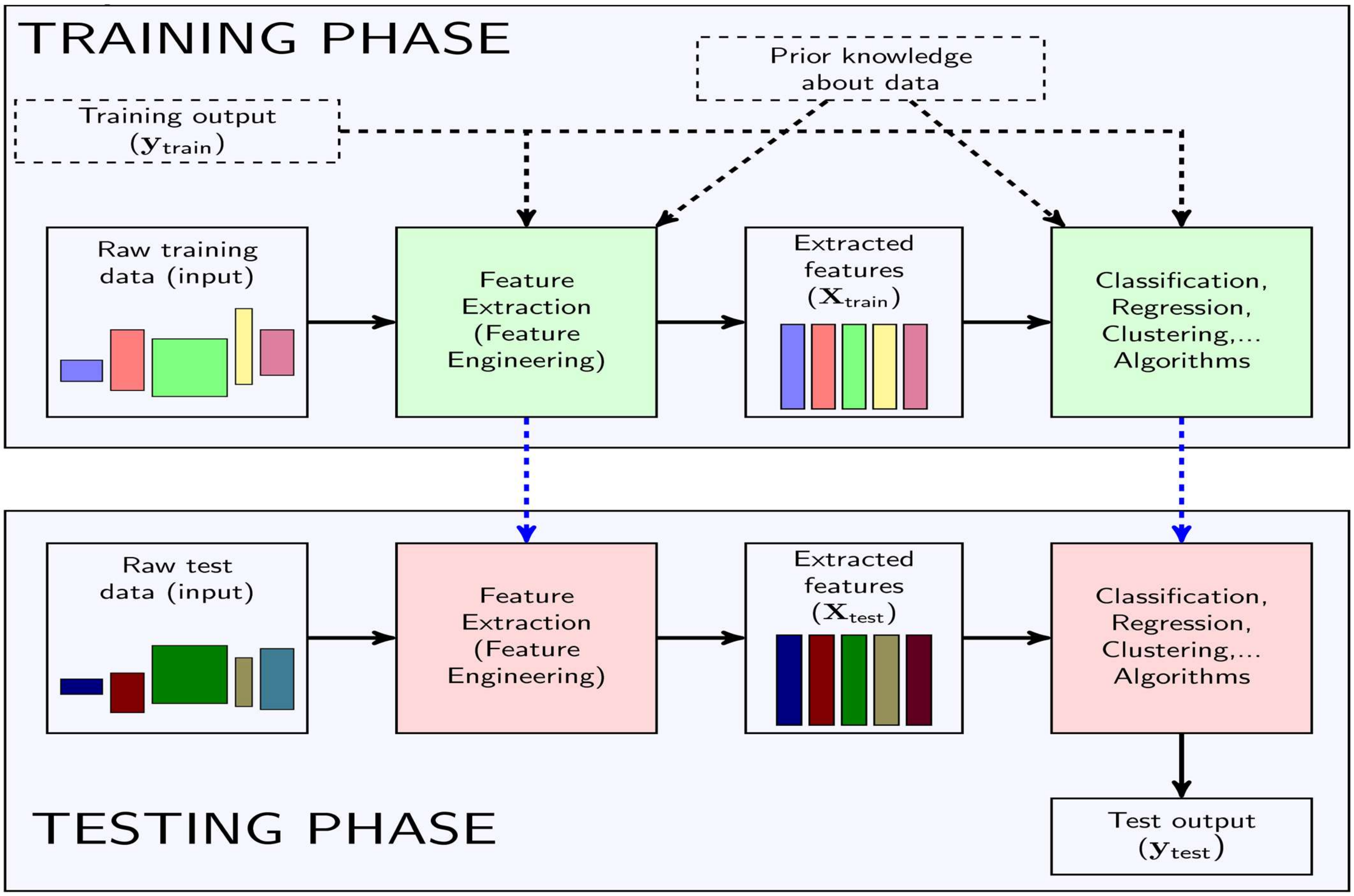
4. Shape-based feature descriptors

Introduction to feature & feature engineering

- Feature:
  - an individual measurable property or characteristic of a data example
  - describes the example
- Features are usually numeric.
- Feature engineering: transfer raw data into feature vector

Data → feature vector → ML model

The general framework for Machine Learning


TRAINING PHASE
Prior knowledge about data
Training output ($\mathbf{y}_{\text{train}}$)
Raw training data (input)
Feature Extraction (Feature Engineering)
Extracted features ($\mathbf{X}_{\text{train}}$)
Classification, Regression, Clustering,... Algorithms
Raw test data (input)
Feature Extraction (Feature Engineering)
Extracted features ($\mathbf{X}_{\text{test}}$)
Classification, Regression, Clustering,... Algorithms
Test output ($\mathbf{y}_{\text{test}}$)
TESTING PHASE

Curse of dimensionality

- Dimensionality: the number of features in feature vector.
- Curse of dimensionality:
  - The number of features is very large relative to the number of observations (examples) in dataset
  - Hard to train effective model

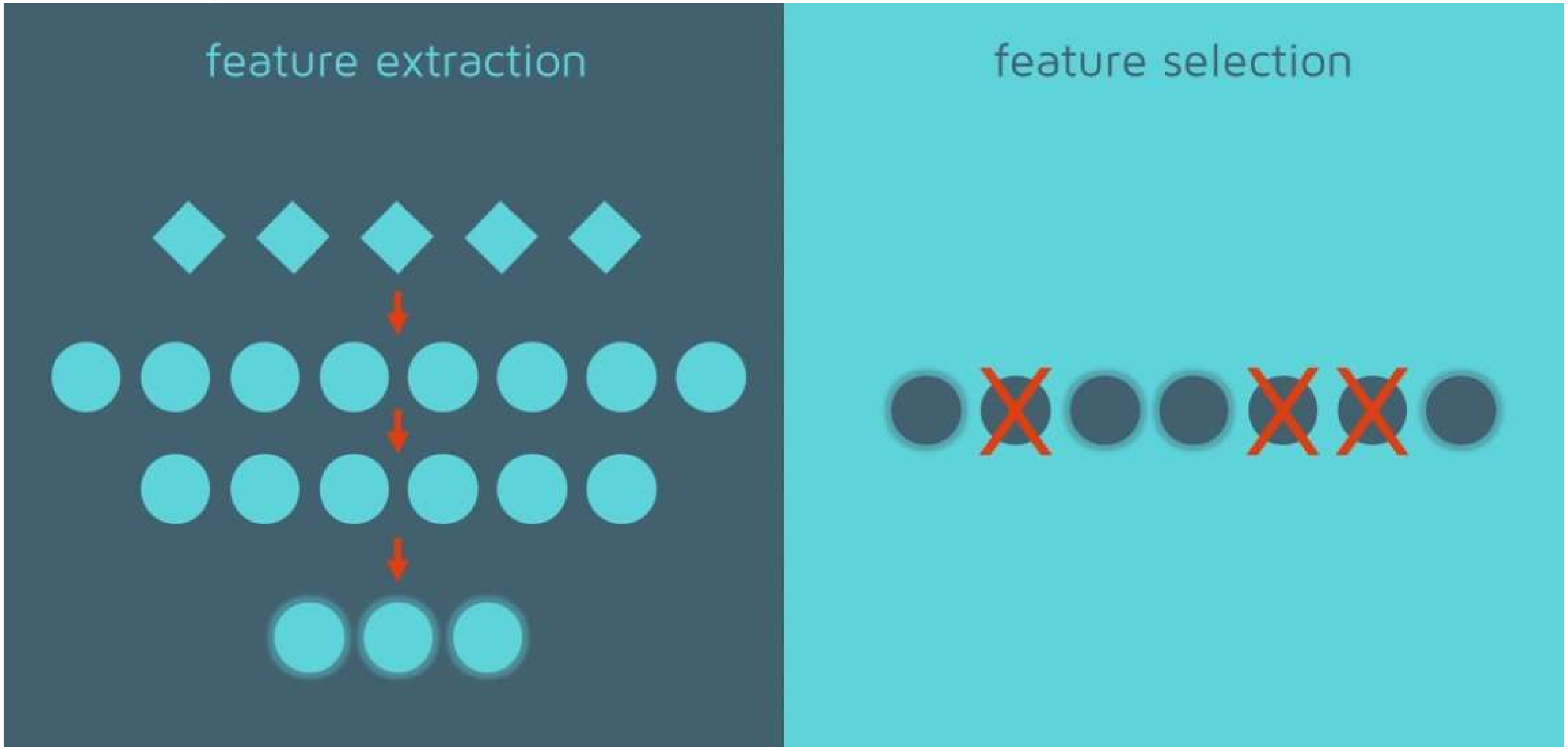
→ Dimensionality reduction

- Feature selection
- Feature extraction

Feature extraction vs. feature selection

- Feature selection:
  - Filtering irrelevant or redundant features from dataset
  - Choosing a subset of the original features
- Feature extraction:
  - Creating a new smaller set of features
  - Getting useful features from existing data
- Feature need to be informative, discriminating and independent

Feature extraction vs. feature selection



Feature engineering

- **Duration:** 4 hrs
- **Outline:**

1. Introduction
2. **Feature engineering**
3. Features in visual pattern recognition
4. Shape-based feature descriptors

Feature engineering

- One-hot encoding
- Binning
- Normalization
- Standardization
- Dealing with missing feature
- Data imputation techniques

One-hot encoding

- Transform a categorical feature into several binary features
- Example: feature “color” has 3 values “red”, “yellow”, green”

➢ “red” = 1, “yellow” = 2, “green” = 3

➢ ?

$$\begin{aligned} red &= [1,0,0] \\ yellow &= [0,1,0] \\ green &= [0,0,1] \end{aligned}$$

Binning (bucketing)

- Transform a numerical feature into categorical feature
- Example: feature “age”
  - Put all ages between 0 and 5 years-old into one bin
  - Put ages from 6 to 10 years-old in the second bin
  - Put ages from 11 to 15 years-old in the third bin, and so on.

Normalization

- Converting an actual range of values of a numerical feature into a standard range of values, typically in the interval [-1, 1] or [0, 1].
- Example: natural range = [350, 1450]
  - Subtracting 350 from every value of the feature
  - Dividing the result by 1100 → normalized range = [0, 1].

Normalization Formula

$$X_{normalized} = \frac{(X - X_{minimum})}{(X_{maximum} - X_{minimum})}$$

Standardization

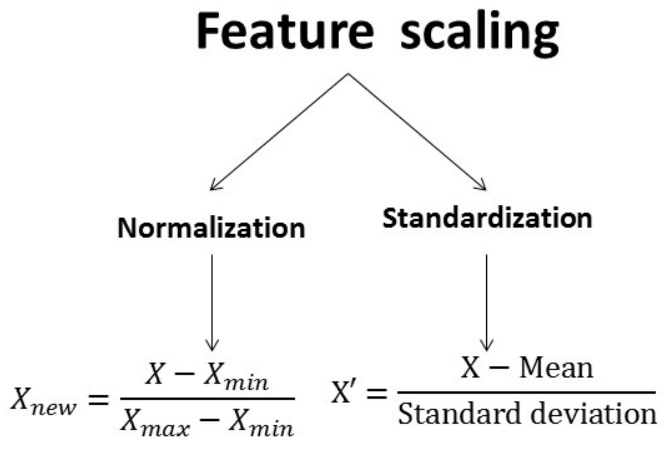
- Rescaling the feature values so that they have the properties of a standard normal distribution with $\mu = 0$ and $\sigma = 1$
- Formula:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}.$$

Standardization or normalization?

- Try two if have time ☺
- Rule of thumbs:

- unsupervised learning algorithms, in practice, more often benefit from standardization than from normalization;
- standardization is also preferred for a feature if the values this feature takes are distributed close to a normal distribution (so-called bell curve);
- again, standardization is preferred for a feature if it can sometimes have extremely high or low values (outliers); this is because normalization will "squeeze" the normal values into a very small range;
- in all other cases, normalization is preferable.



$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad X' = \frac{\text{X} - \text{Mean}}{\text{Standard deviation}}$$

Dealing with missing features

- Removing the examples with missing features.
- Use data imputation technique

Data imputation techniques

- Technique 1: Replacing the missing value of a feature by an average value of this feature in the dataset
- Technique 2: Replacing the missing value by the same value outside the normal range of values.
- Technique 3: Replacing the missing value by a value in the middle of the range.

…etc…

Feature engineering

- **Duration:** 4 hrs
- **Outline:**

1. Introduction
2. Feature engineering
3. **Features in visual pattern recognition**
4. Shape-based feature descriptors

Image feature extraction

- Purpose:
  - To reduce the dimensionality of input image
  - To transform each input image into a corresponding multi-dimension feature vector
  - To perform the predefined classification tasks with sufficient accuracy without using the entire input image
- Requirements:
  - Features should extract the most suitable characteristics from the input image

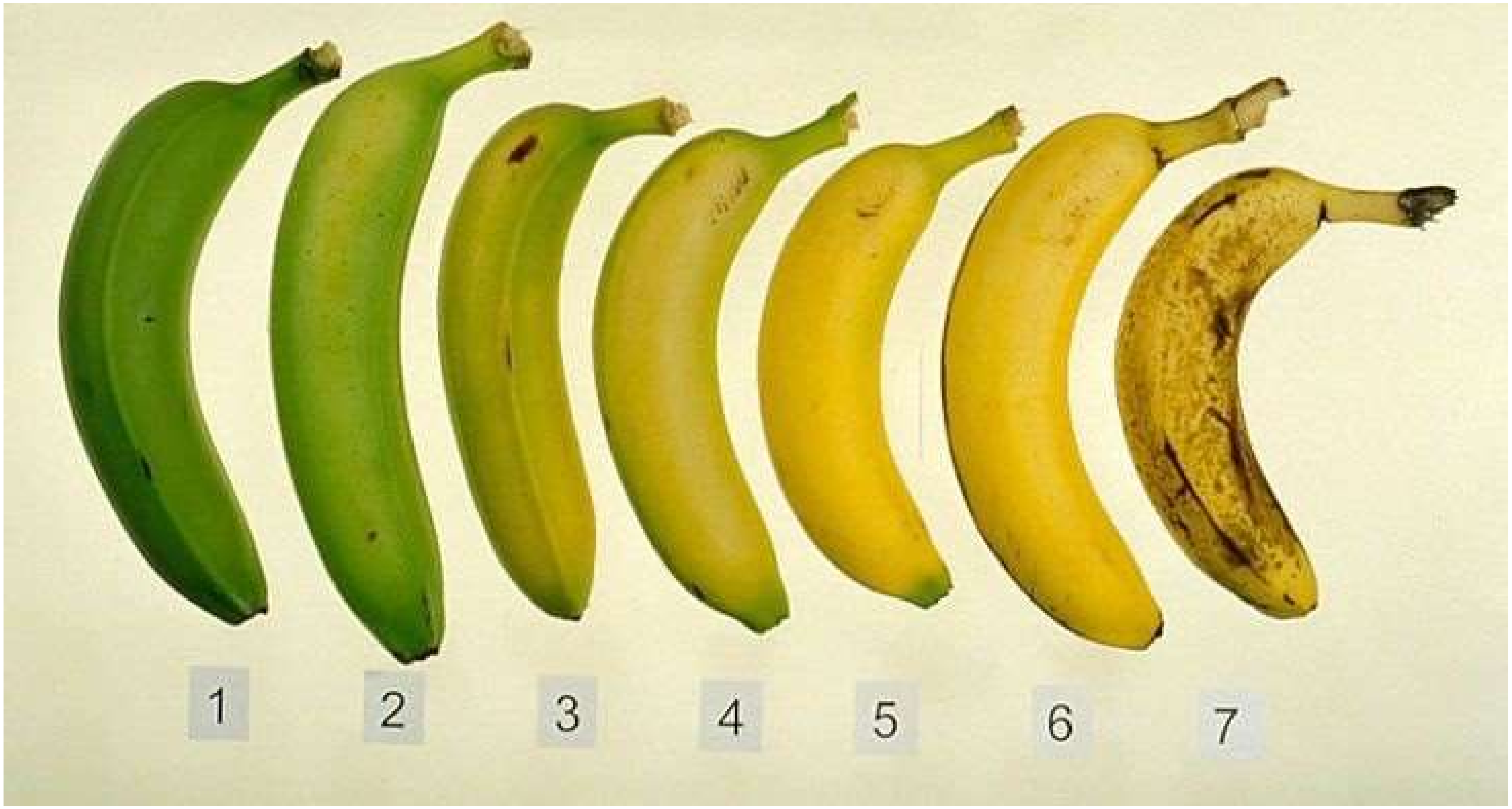
An example of feature extraction

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Moment	0.3461	0.0738	0.0093	0.0042	2.61E-05	0.0011	4.85E-06
logHu	-1.0610	-2.6069	-4.6771	-5.4646	-10.5523	-6.8463	-12.2370

.

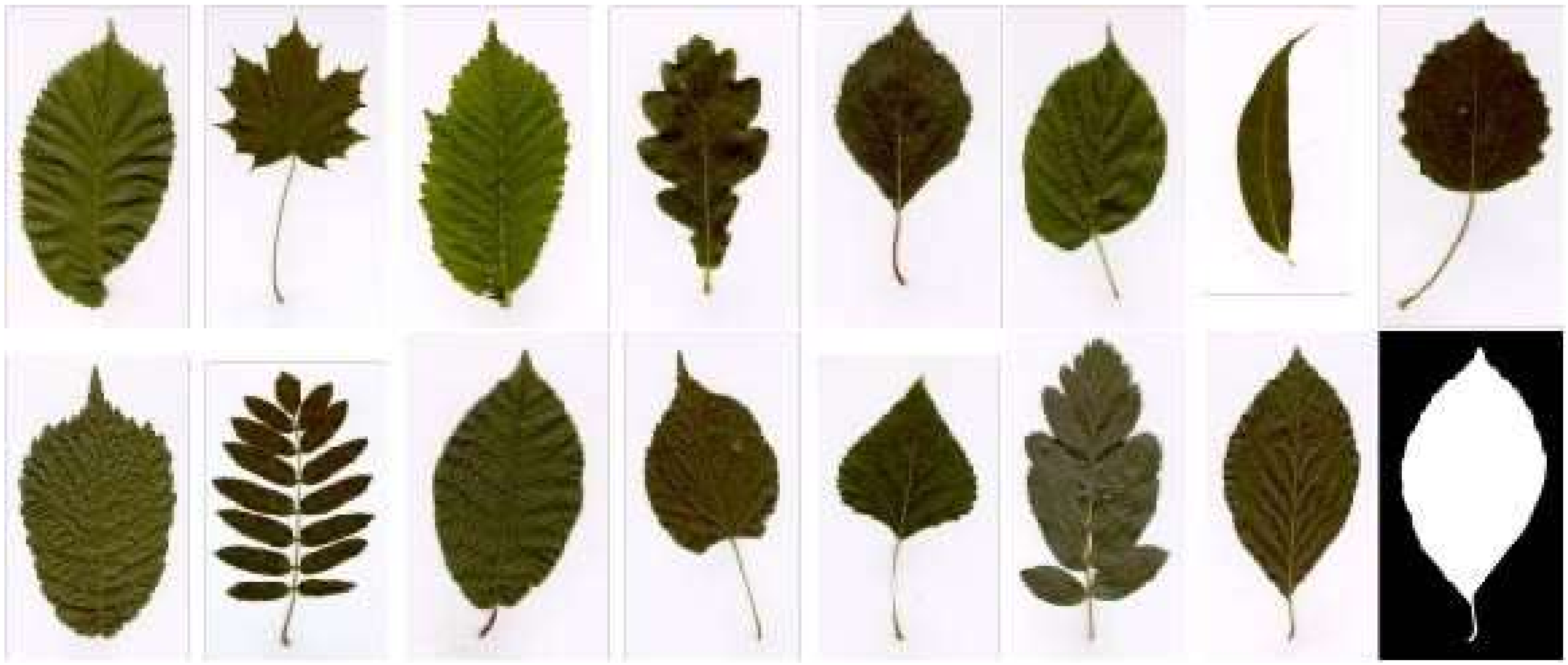
Visual features

- Color-based features



Visual features

- Shape-based features



Visual features

- Texture-based features

Which feature is the best?

- **Example:** plant recognition
- Plant features: leaf, fruit, flower, root, branch,…
- Leaf features: shape, vein, margin, texture
- No single best feature for a given leaf identity → combination of different features
- No single best presentation for a given feature → multiple descriptors to characterize the feature from different perspectives

→
- Challenging
- Innovative

→ Deep learning

Feature engineering

- **Duration:** 4 hrs
- **Outline:**

1. Introduction
2. Feature engineering
3. Features in visual pattern recognition
4. **Shape-based feature descriptors**

Shape-based feature descriptor

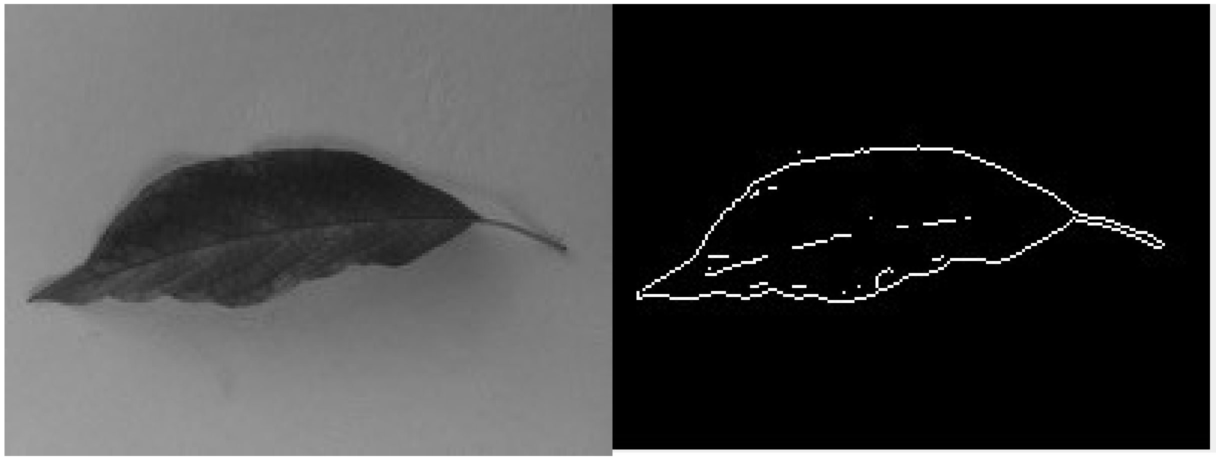
- Shape: important
- Good shape descriptor: invariant to geometrical transformations (rotation, reflection, scaling, translation)
- Types of shape descriptors: simple and morphological shape descriptor (SMSD), contour-based, region-based

Simple and morphological shape descriptor

- Refer to basic geometric properties of the shape
- Basic descriptor: diameter, major axis length, minor axis length, area, perimeter, centroid,…
- Morphological descriptor: aspect ratio, perimeter to area ratio, rectangularity measures, circularity measures,…

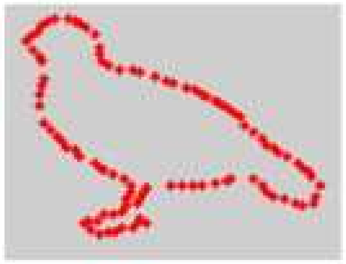
Contour-based feature descriptor

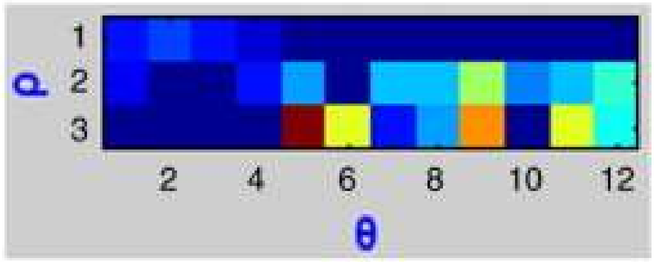
- Consider the boundary of a shape and neglect the information contained in the shape interior
- Ex: CCD (centroid contour distance), Fourier descriptor computed on CCD.



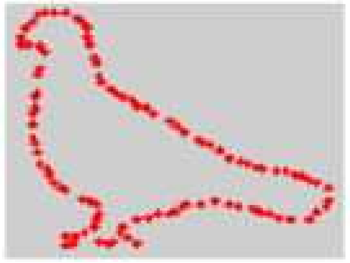
Contour-based feature descriptor

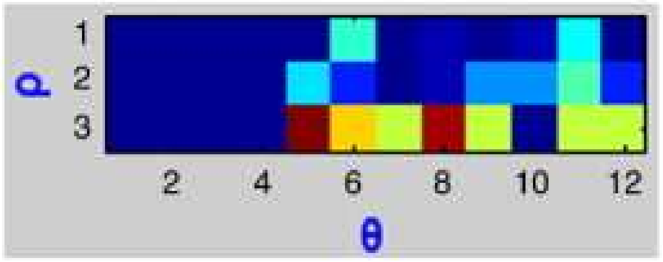
(a) Original image


(b) Contour image


(c) Histogram of the contour points distribution

(d) Original image


(e) Contour image


(f) Histogram of the contour points distribution

Region-based feature descriptor

- Take all the pixels within a shape region into account to obtain the shape representation
- Image moments: statistical descriptor of a shape. Ex: Hu moments
- Local features: select key points in image. Ex: HOG (histogram of oriented gradients), SIFT (scale-invariant feature transform)

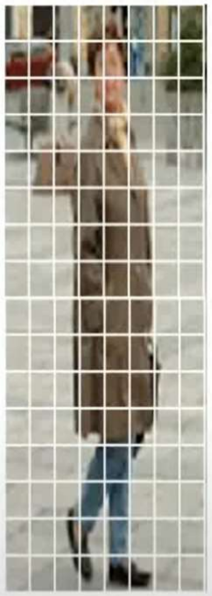
Histogram of Oriented Gradients (HOG) ALGORITHM

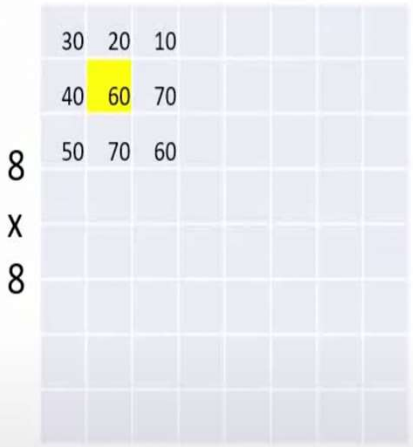
- HOG stands for histogram of oriented gradients.
- The hog descriptor focuses on structure or shape of the object.
- It uses magnitude as well as direction of the gradient to compute the features.
- It generates histogram by using magnitude and direction of the gradient.

HOG ALGORITHM

150x300

64x128

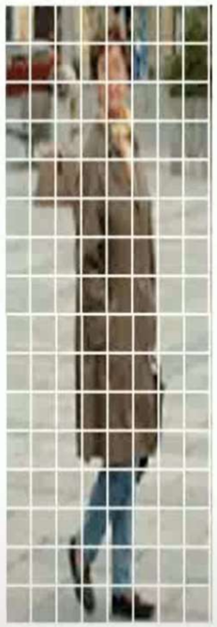

8x16



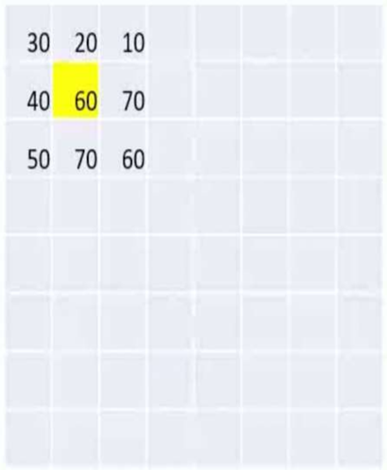
HOG ALGORITHM

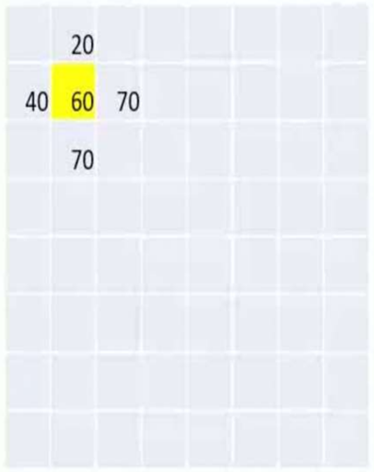
	20	
40		70
	70	

- **Here we calculating gradient magnitude and direction, to calculate pixels intensity we need**
- **X direction=|40-70|=30**
- **Y direction=|20-70|=50**
- **By these values we are calculating magnitude and direction of the gradient**
- **By using magnitude and direction we calculate feature vectors**



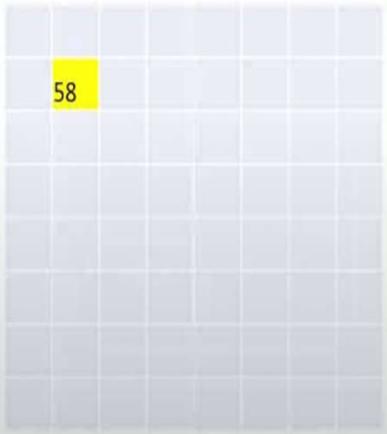
8 x 8



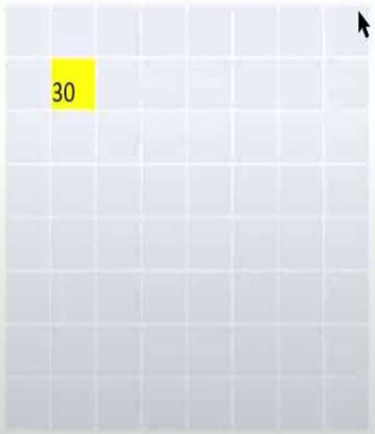


X direction = |40 – 70| = 30

Y direction = |20 – 70| = 50



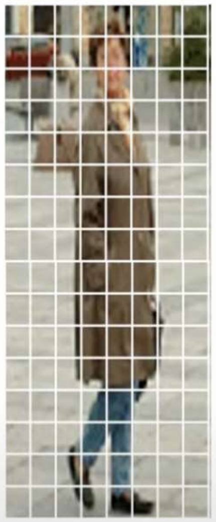
Grad Magnitude

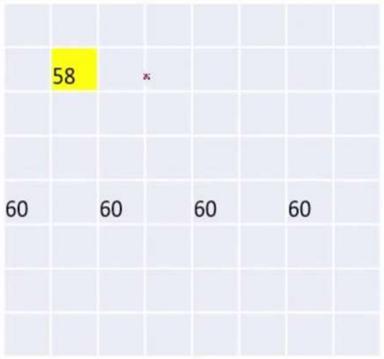


Grad Direction

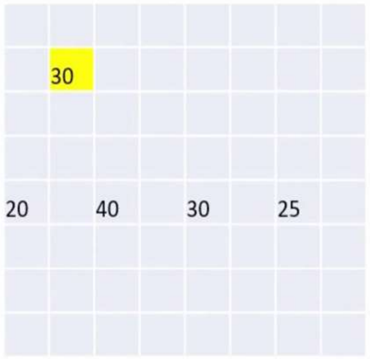
Grad Mag = sqrt(30^2 + 50^2) = ~58

Grad Direction = $\tan^{-1}(30/50)$ = ~30^0

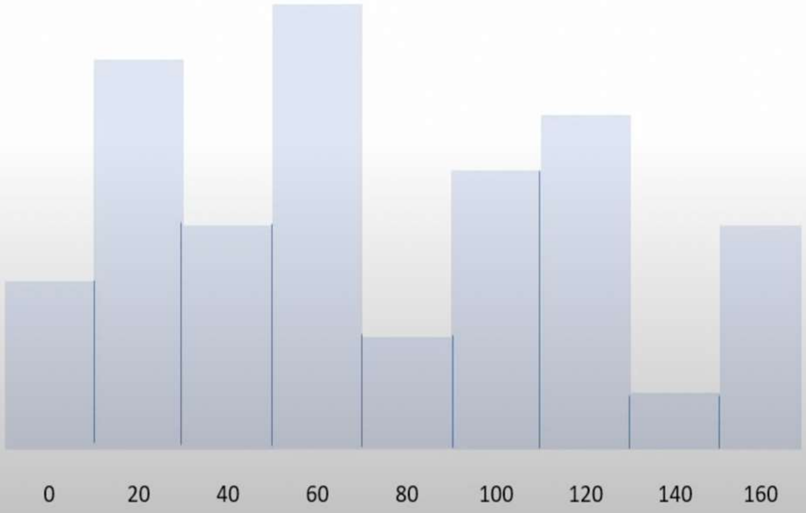


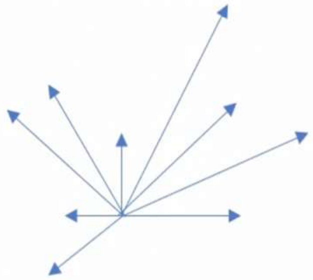


Grad Magnitude



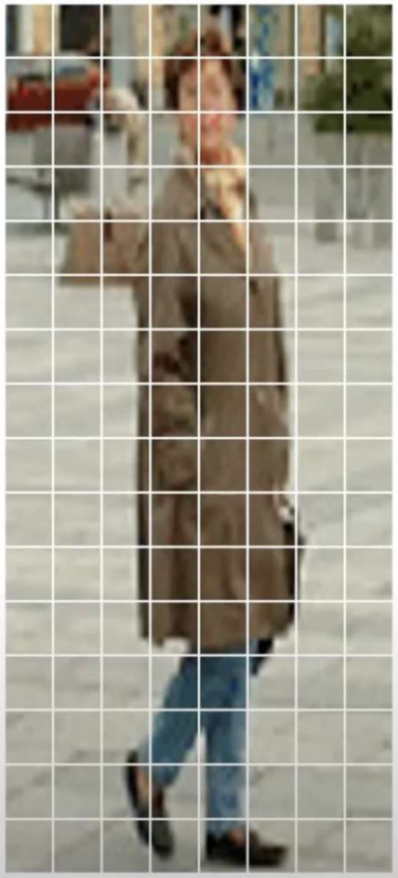
Grad Direction





Feature Vector of size 9

HOG ALGORITHM



8x16 7*15*36 = 3780

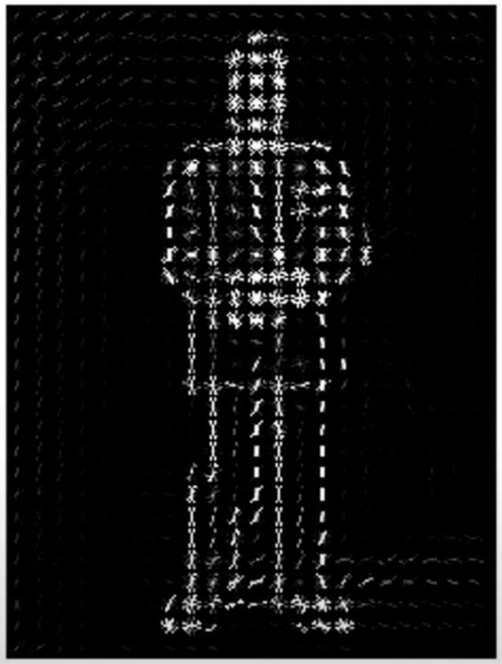
HOG ALGORITHM

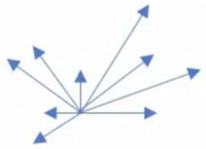
- Before getting the hog feature and after concatenating feature vectors we are supposed to do normalize.
- Suppose we have taken 150*300 pixels and multiply with 2 to increase the brightness and divided by 2 to decrease the brightness, then you cant compare two images without normalization bec’z the pixels intensity will be changed.
- But if you normalize the feature vectors it is easy to compare

HOG ALGORITHM

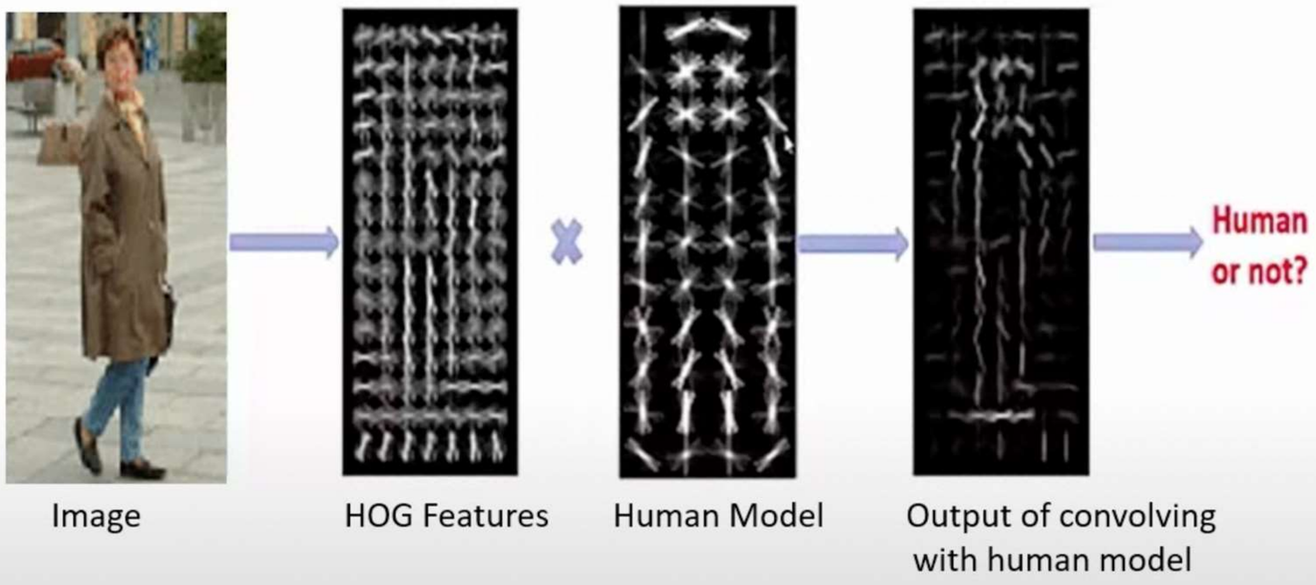
Input image

Histogram of Oriented Gradients





HOG ALGORITHM



- For hog features giving human template and giving output for convolving with human model
- Then it will predict whether it is human or not.

Image moments

- Weighted average of image pixel intensities.
- The pixel intensity at location (x,y) is given by s(x,y)
- Binary image: s(x,y) = 0/1
- Simplest moment: $M = \sum_x \sum_y s(x,y)$

➢ The number of white pixels

➢ The area of white region (object) in the image

Hu moments feature descriptor

Central moments:

$$m_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q s(x,y) \quad p,q = 0,1,2,3$$

Central normalized moments:

$$M_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}}$$

Centroid of the image:

$$\bar{x} = \frac{\sum_x \sum_y x.s(x,y)}{\sum_x \sum_y s(x,y)}, \bar{y} = \frac{\sum_x \sum_y y.s(x,y)}{\sum_x \sum_y s(x,y)}$$

Hu moments feature (cont)

$$S_1 = M_{20} + M_{02}$$

$$S_2 = (M_{20} - M_{02})(M_{20} + M_{02}) + 4M_{11}M_{11}$$

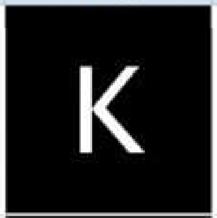
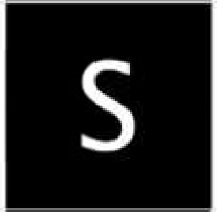
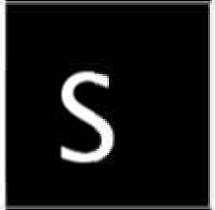
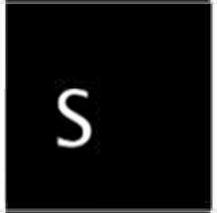
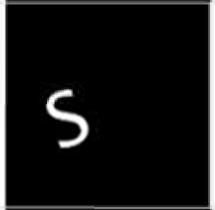
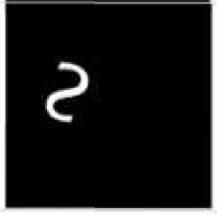
$$S_3 = (M_{30} - 3M_{12})^2 + (M_{30} - 3M_{21})^2$$

$$S_4 = (M_{30} + M_{12})^2 + (M_{03} + M_{21})^2$$

$$S_5 = (M_{30} - 3M_{12})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{03} + M_{21})^2] + (3M_{21} - M_{03})(M_{03} + M_{21})[3(M_{30} + M_{12})^2 - (M_{03} + M_{21})^2]$$

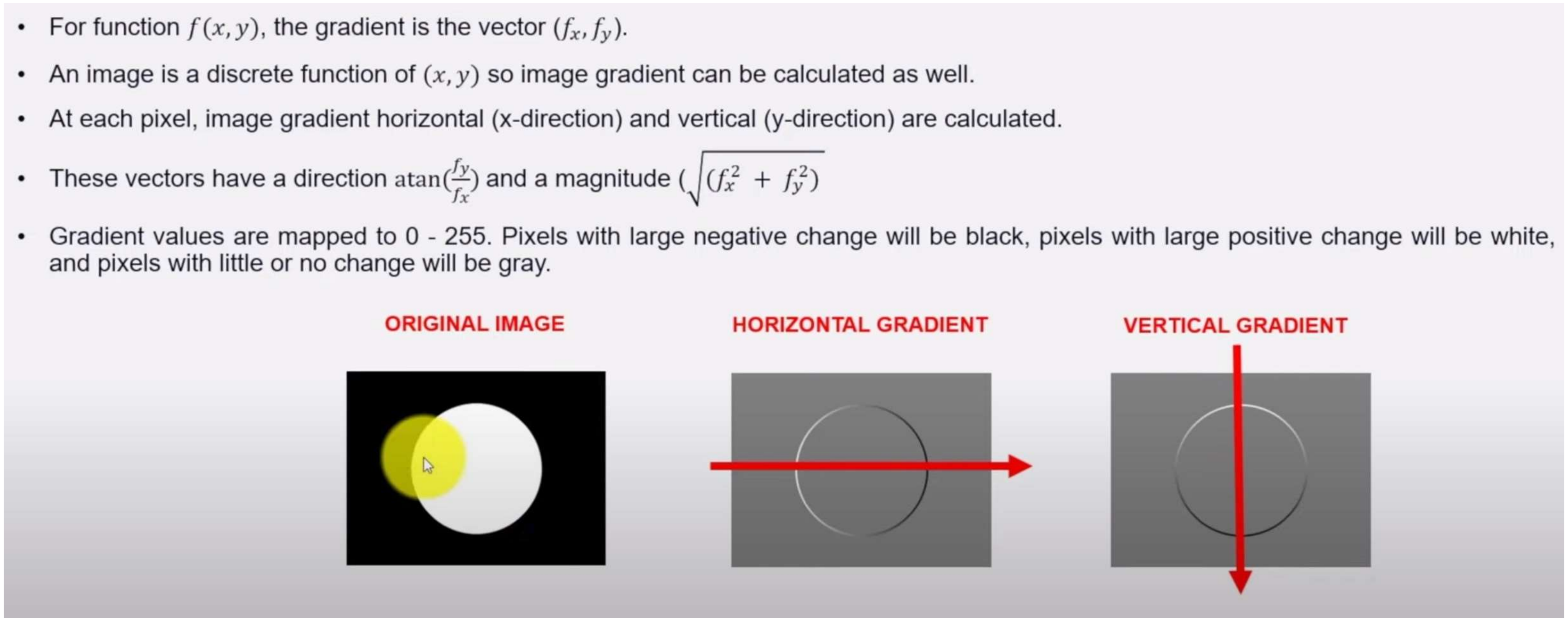
$$S_6 = (M_{20} - M_{02})[(M_{30} + M_{12})^2 - (M_{03} + M_{21})^2] + 4M_{11}(M_{30} + M_{12})(M_{03} + M_{21})$$

$$S_7 = (3M_{21} - M_{03})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{03} + M_{21})^2] + (M_{30} - 3M_{12})(M_{21} + M_{02})[3(M_{30} + M_{12})^2 - (M_{03} + M_{21})^2]$$

Id	image	S1	S2	S3	S4	S5	S6	S7
1		2.78871	6.50638	9.44249	9.84018	-19.593	-13.1205	19.6797
2		2.67431	5.77446	9.90311	11.0016	-21.4722	-14.1102	22.0012
3		2.67431	5.77446	9.90311	11.0016	-21.4722	-14.1102	22.0012
4		2.65884	5.7358	9.66822	10.7427	-20.9914	-13.8694	21.3202
5		2.66083	5.745	9.80616	10.8859	-21.2468	-13.9653	21.8214
6		2.66083	5.745	9.80616	10.8859	-21.2468	-13.9653	-21.8214

HOG

https://www.youtube.com/watch?v=XmO0CSsKg88&t=41s



- For function $f(x, y)$, the gradient is the vector (f_x, f_y).
- An image is a discrete function of (x, y) so image gradient can be calculated as well.
- At each pixel, image gradient horizontal (x-direction) and vertical (y-direction) are calculated.
- These vectors have a direction $\mathrm{atan}(\frac{f_y}{f_x})$ and a magnitude $(\sqrt{(f_x^2 + f_y^2)}$
- Gradient values are mapped to 0 - 255. Pixels with large negative change will be black, pixels with large positive change will be white, and pixels with little or no change will be gray.