UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

# Feature Extraction, Selection and Fusion; Image Classification

Dr. Eng. Laksmita Rahadianti, Muhammad Febrian Rachmadi, Ph.D., Dr. Dina Chahyati, Prof. Dr. Aniati M. Arymurthy

CSCE604133 Computer Vision
Fakultas Ilmu Komputer Universitas Indonesia

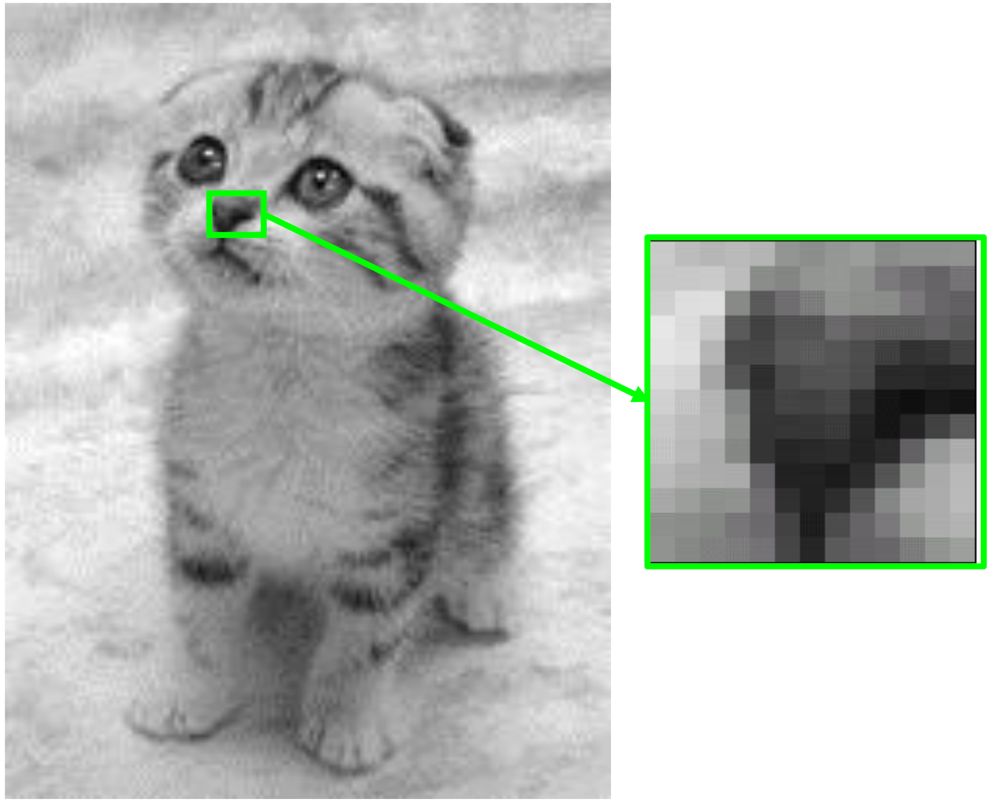
# Acknowledgements


- These slides are created with reference to:
  - Computer Vision: Algorithms and Applications, 2nd ed., Richard Szeliski https://szeliski.org/Book/
  - Digital Image Processing, Gonzales and Woods, 3rd ed, 2008.
  - Course slides for CSCE604133 Image Processing – Faculty of Computer Science, Universitas Indonesia
  - Introduction to Computer Vision, Cornell Tech https://www.cs.cornell.edu/courses/cs5670/2024sp/lectures/lectures.html
  - Computer Vision, University of Washington https://courses.cs.washington.edu/courses/cse576/08sp/

# What do you see in this image?

- How did you know?
- What did you look at?
- What **parts** of the image did you identify?

# What is a Digital Image?



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 201 | 188 | 181 | 185 | 180 | 147 | 140 | 149 | 155 | 138 | 144 | 144 | 145 |
| 199 | 200 | 201 | 188 | 139 | 132 | 147 | 150 | 143 | 123 | 112 | 102 | 117 |
| 207 | 221 | 222 | 136 | 90 | 111 | 125 | 145 | 140 | 138 | 122 | 104 | 97 |
| 231 | 219 | 200 | 90 | 65 | 84 | 84 | 107 | 95 | 92 | 92 | 99 | 89 |
| 227 | 223 | 181 | 74 | 72 | 89 | 92 | 86 | 77 | 63 | 50 | 55 | 65 |
| 217 | 211 | 166 | 85 | 47 | 75 | 82 | 83 | 75 | 42 | 42 | 39 | 40 |
| 208 | 195 | 179 | 131 | 54 | 68 | 66 | 72 | 46 | 21 | 15 | 24 | 19 |
| 198 | 187 | 181 | 141 | 53 | 54 | 55 | 59 | 37 | 21 | 37 | 66 | 90 |
| 195 | 184 | 170 | 134 | 52 | 38 | 42 | 45 | 35 | 43 | 98 | 152 | 172 |
| 186 | 175 | 171 | 169 | 100 | 34 | 34 | 27 | 44 | 85 | 139 | 170 | 184 |
| 167 | 156 | 142 | 144 | 112 | 48 | 32 | 46 | 84 | 133 | 166 | 172 | 186 |
| 142 | 139 | 131 | 120 | 108 | 67 | 30 | 76 | 102 | 123 | 153 | 171 | 178 |
| 145 | 134 | 128 | 125 | 117 | 70 | 38 | 91 | 101 | 105 | 125 | 146 | 157 |

# Human vs Computer

- What you see

- What the computer sees

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 201 | 188 | 181 | 185 | 180 | 147 | 140 | 149 | 155 | 138 | 144 | 144 | 145 |
| 199 | 200 | 201 | 188 | 139 | 132 | 147 | 150 | 143 | 123 | 112 | 102 | 117 |
| 207 | 221 | 222 | 136 | 90 | 111 | 125 | 145 | 140 | 138 | 122 | 104 | 97 |
| 231 | 219 | 200 | 90 | 65 | 84 | 84 | 107 | 95 | 92 | 92 | 99 | 89 |
| 227 | 223 | 181 | 74 | 72 | 89 | 92 | 86 | 77 | 63 | 50 | 55 | 65 |
| 217 | 211 | 166 | 85 | 47 | 75 | 82 | 83 | 75 | 42 | 42 | 39 | 40 |
| 208 | 195 | 179 | 131 | 54 | 68 | 66 | 72 | 46 | 21 | 15 | 24 | 19 |
| 198 | 187 | 181 | 141 | 53 | 54 | 55 | 59 | 37 | 21 | 37 | 66 | 90 |
| 195 | 184 | 170 | 134 | 52 | 38 | 42 | 45 | 35 | 43 | 98 | 152 | 172 |
| 186 | 175 | 171 | 169 | 100 | 34 | 34 | 27 | 44 | 85 | 139 | 170 | 184 |
| 167 | 156 | 142 | 144 | 112 | 48 | 32 | 46 | 84 | 133 | 166 | 172 | 186 |
| 142 | 139 | 131 | 120 | 108 | 67 | 30 | 76 | 102 | 123 | 153 | 171 | 178 |
| 145 | 134 | 128 | 125 | 117 | 70 | 38 | 91 | 101 | 105 | 125 | 146 | 157 |

We need to be able to represent the image in a certain way so that the computer knows what it's looking at

# Representation and Description

- **Features** are used to **represent** the image and **describe** it.
- The motivation is that we need to **differentiate** images.
- We use attributes that:
  - are going to help us assign unique labels to objects in an image or
  - are going to be of value in differentiating images or families of images.

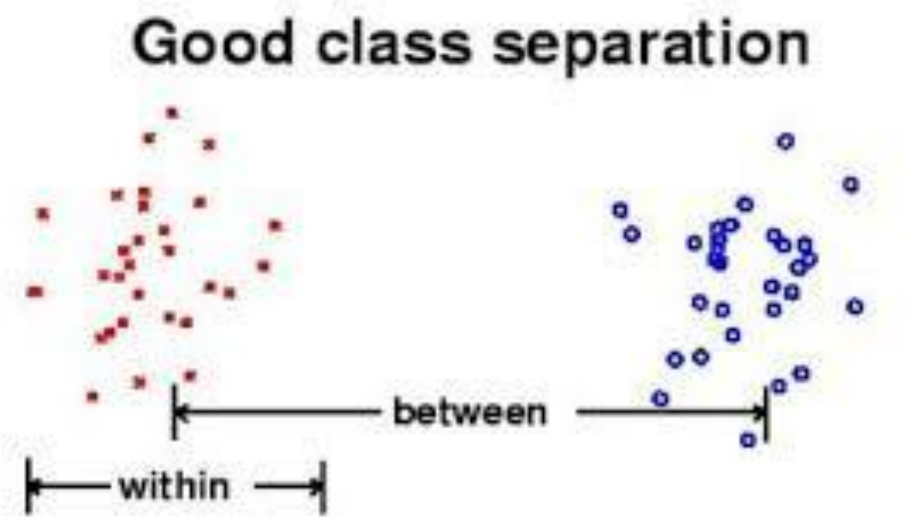
$F = 12$ (circumference)

- Representation : A region can be represented by its boundary
- Description: The boundary can be described by its length, number of concavities, etc.

# Feature Extraction

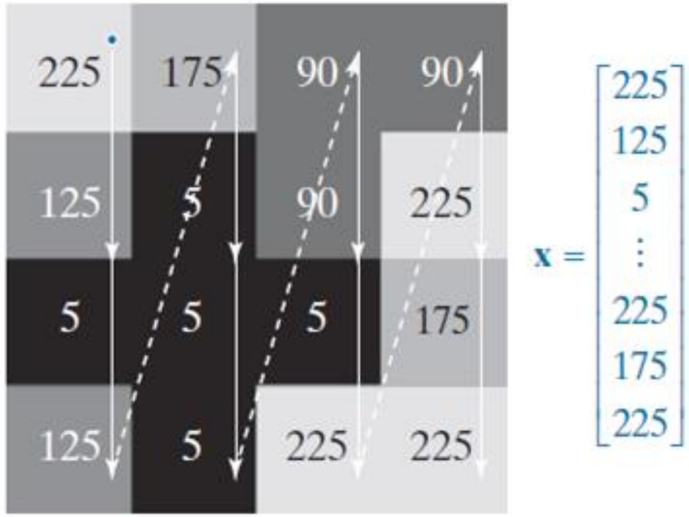
# Feature Extraction

- Feature Extraction: extracting attributes from the raw data (image)
  - minimizing the within class pattern variability
  - maximizing the between class pattern variability
- Steps:
  - Feature Detection: Finding the distinctive attributes
  - Feature Description: Assigning (usually) quantitative values to the attributes
- Whenever possible, preprocessing should be used to normalize input images before feature extraction.



# Descriptors

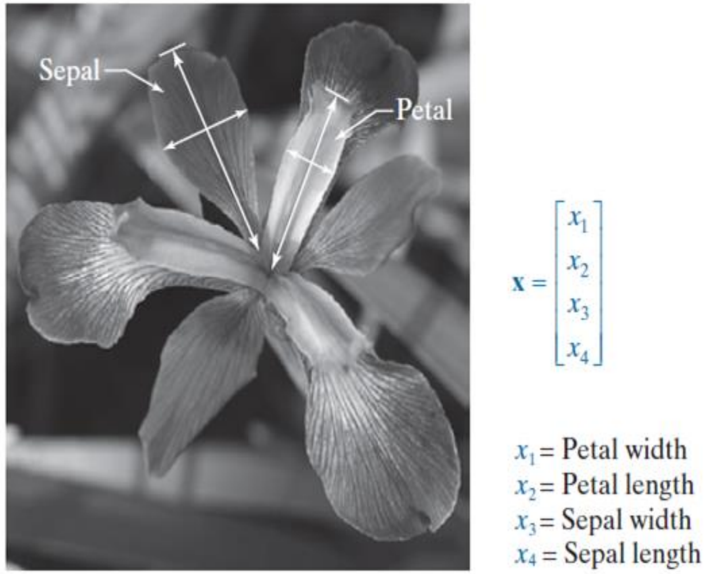
- Linear Indexing




a b
**FIGURE 12.1** Using linear indexing to vectorize a grayscale image.

- Features of flowers

**FIGURE 12.2** Petal and sepal width and length measurements (see arrows) performed on iris flowers for the purpose of data classification. The image shown is of the *Iris virginica* gender. (Image courtesy of USDA.)

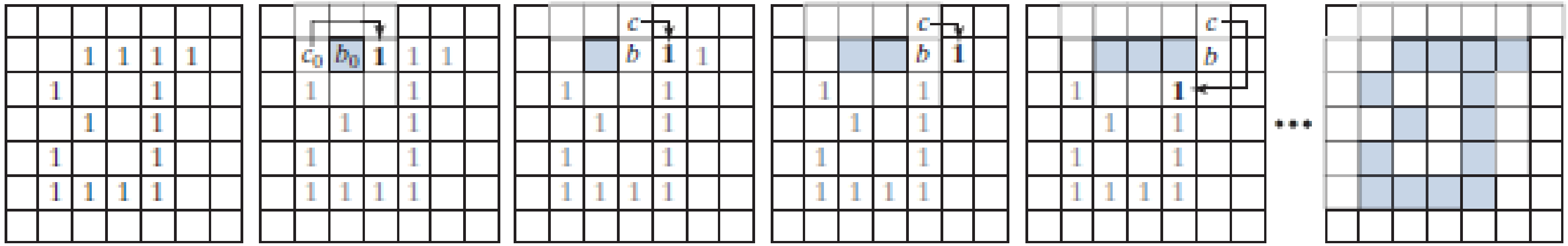


# Descriptors: Boundary Preprocessing

- Segmentation techniques give us raw data in the form of pixels along a boundary or pixels contained in a region.
- Then, the segmented images can be transformed into representations that facilitate the computation of descriptors, through boundary preprocessing
- Some methods:
  - Boundary Following (Tracing)
  - Chain Codes
  - Minimum-Perimeter Polygons
  - Signatures
  - Medial Axis Transform (MAT)

# Descriptors: Boundary Following (Tracing)

- A boundary-following algorithm’s output is an ordered sequence of points.
- Assumptions:
  - binary images in which object and background points 1 and 0,
  - images are padded with a border of 0’s to avoid object merging with the image border.



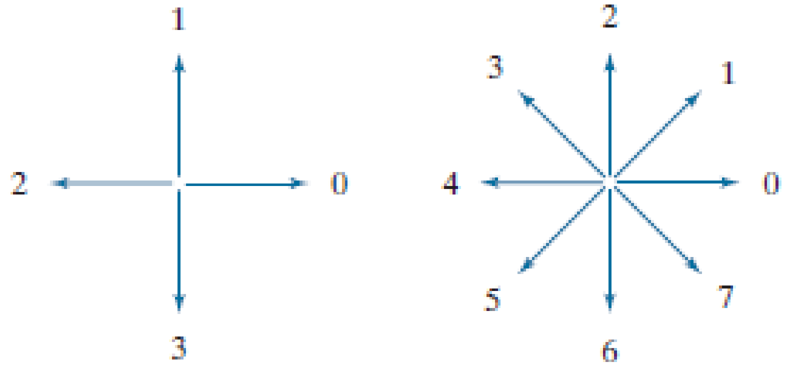
a b c d e f

**FIGURE 11.1** Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in bold, black; the points yet to be processed are gray; and the points found by the algorithm are shaded. Squares without labels are considered background (0) values.

# Descriptors: Chain Codes

- Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction.
- Result:
  - (a) Digital boundary with resampling grid superimposed.
  - (b) Result of resampling.
  - (c) 8-directional chain-coded boundary.

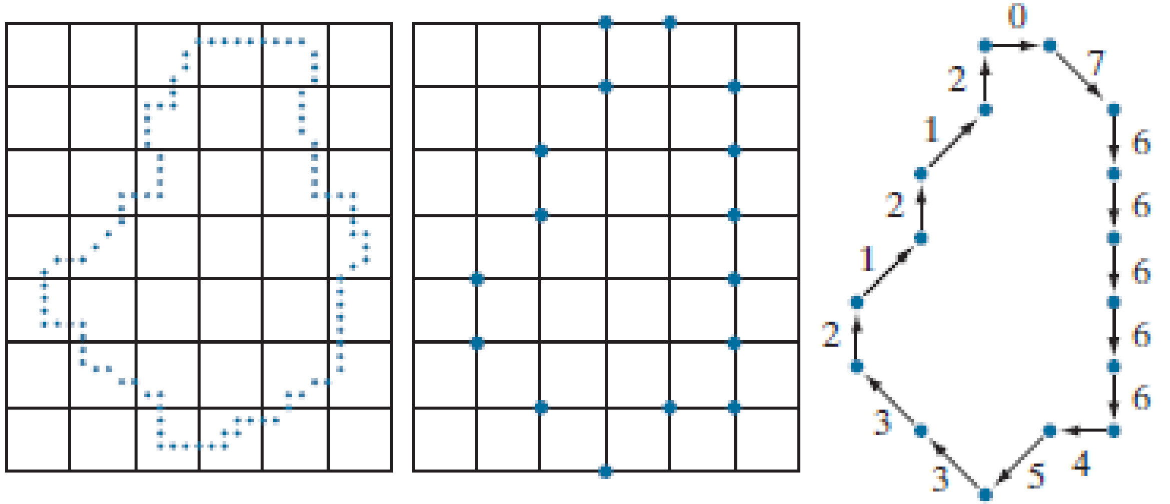
pilih 1 titik untuk 1 poin.



a b

**FIGURE 11.3** Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

ini artinya kearah 0



diagonal

jalo 4- (itu gak bisa detect diagonal)

ini 8-

# Descriptors: Minimum Perimeter Polygons

- A digital boundary can be approximated with arbitrary accuracy by a polygon
- Generally, it is not trivial and can be a time-consuming iterative search.
- However, approximation techniques of modest complexity are well suited for image-processing tasks, e.g., minimum-perimeter polygon (MPP).

jadi ambil diagonal gara2 cost nya paling kecil

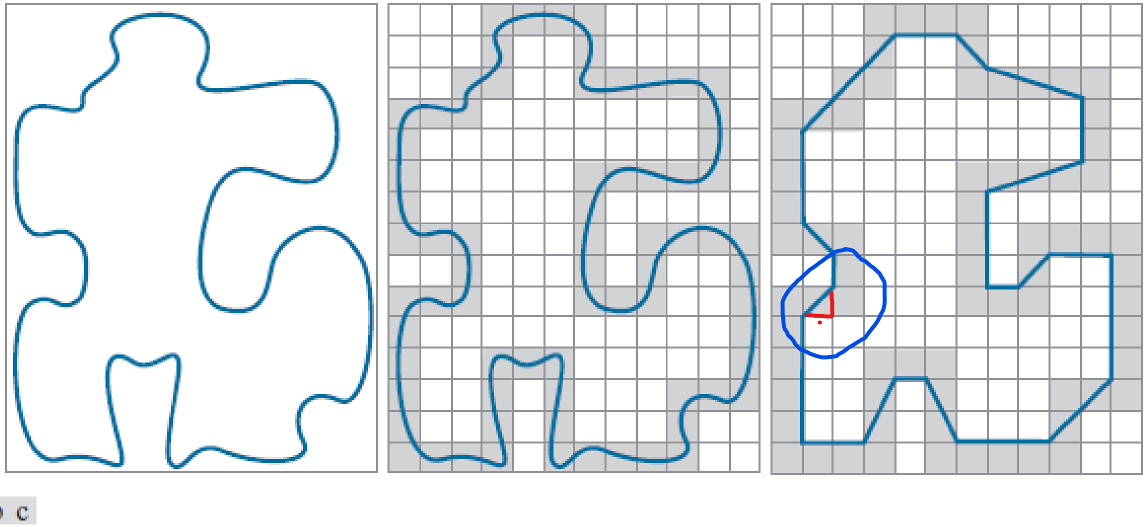
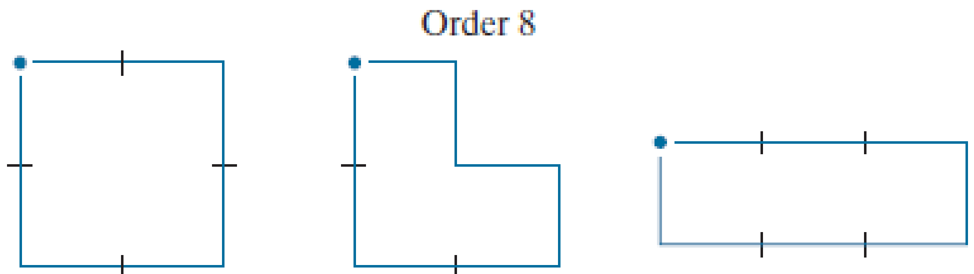



**FIGURE 11.7** (a) An object boundary. (b) Boundary enclosed by cells (shaded). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region.

# Descriptors: Boundary Features

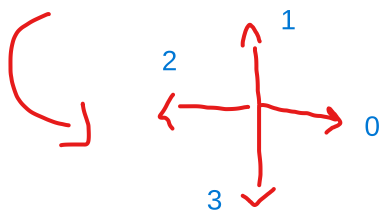
- Statistics
  - Diameter, Length, Angle
  - Statistical moments
- Shape numbers
  - of a Freeman chain-coded boundary, based on the 4/8-directional code




chain code dikasih, nah disini iitik awal itu 1 (artinya awal-awal, dari 1 ke 0 butuh berapa rotasi ada 3 rotasi (nah ini sebagai difference)

| | | | |
|---|---|---|---|
| Chain code: | 0 0 3 3 2 2 1 1 | 0 3 0 3 2 2 1 1 | 0 0 0 3 2 2 2 1 |
| Difference: | 3 0 3 0 3 0 3 0 | 3 3 1 3 3 0 3 0 | 3 0 0 3 3 0 0 3 |
| Shape no.: | 0 3 0 3 0 3 0 3 | 0 3 0 3 3 1 3 3 | 0 0 3 3 0 0 3 3 |

kalau kita pake 4 mata arah




chain code pakai 0 1 2 3 4 (arah mata angin)

karena counter clockwise, kalau 0->3, cost nya itu 3

Kalau 1 ke 3, costnya itu 2, 2 ke 3 itu cost nya 1.

# Descriptors: Region Features

- Compactness, circularity, and eccentricity

$$Compactness = \frac{p^2}{A}$$

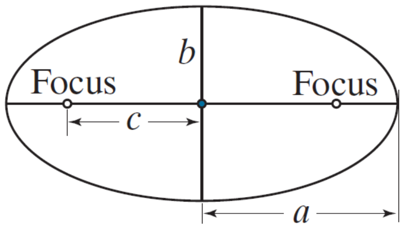
$$Circularity = \frac{4\pi A}{p^2}$$

$$Eccentricity = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a}$$

| **Descriptor** |  |  |  |  |
|---|---|---|---|---|
| *Compactness* | 10.1701 | 42.2442 | 15.9836 | 13.2308 |
| *Circularity* | 1.2356 | 0.2975 | 0.7862 | 0.9478 |
| *Eccentricity* | 0.0411 | 0.0636 | 0 | 0.8117 |

a b c d
**FIGURE 11.22** Compactness, circularity, and eccentricity of some simple binary regions.

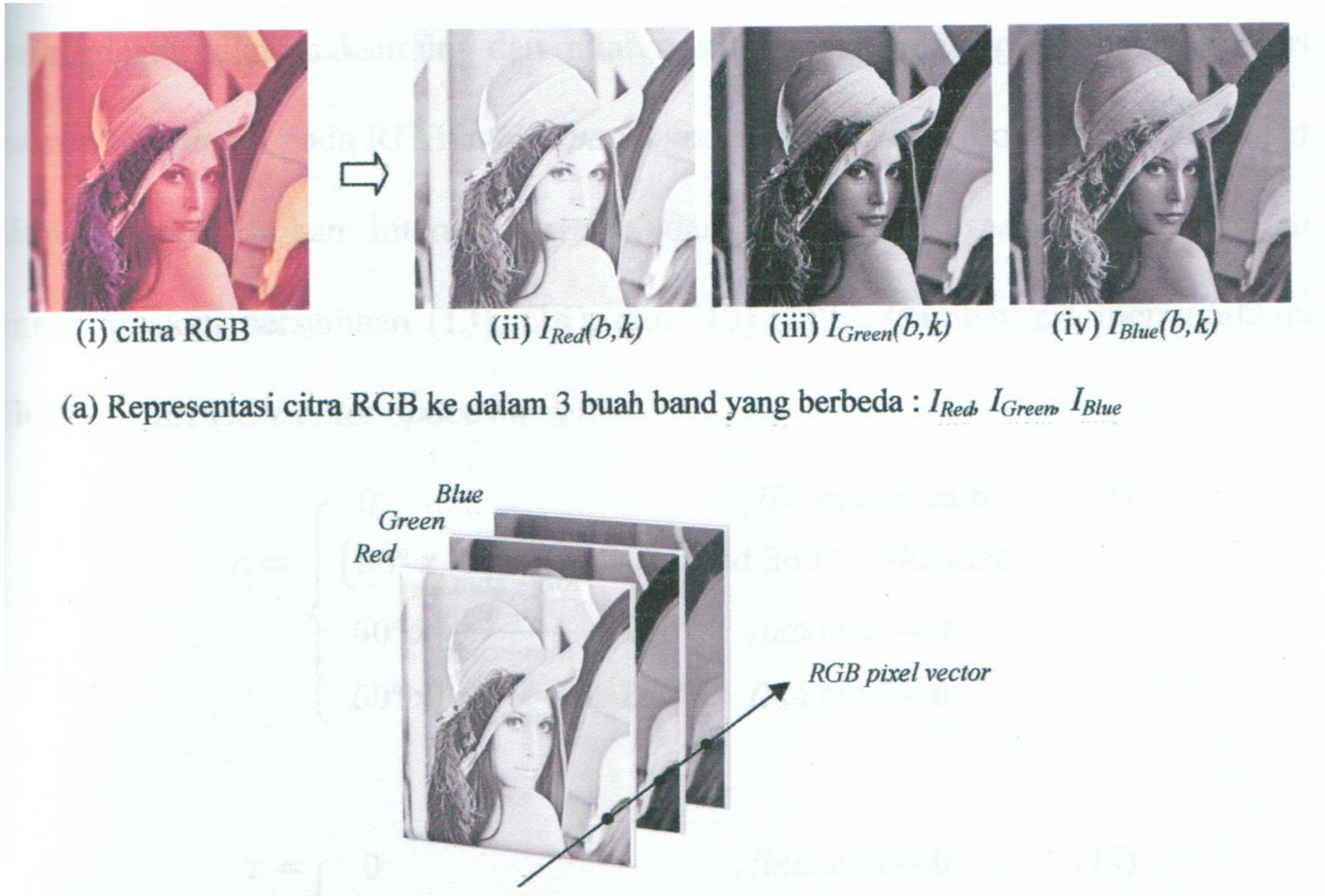
$$c^2 = a^2 - b^2$$




a, b, c of an ellipse
A= Area
P = perimeter

- Topological Descriptors
- Texture

# Color Descriptors

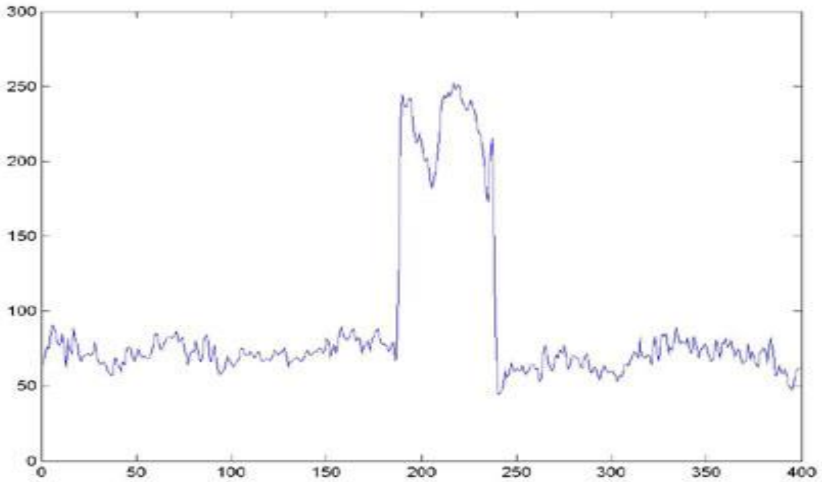


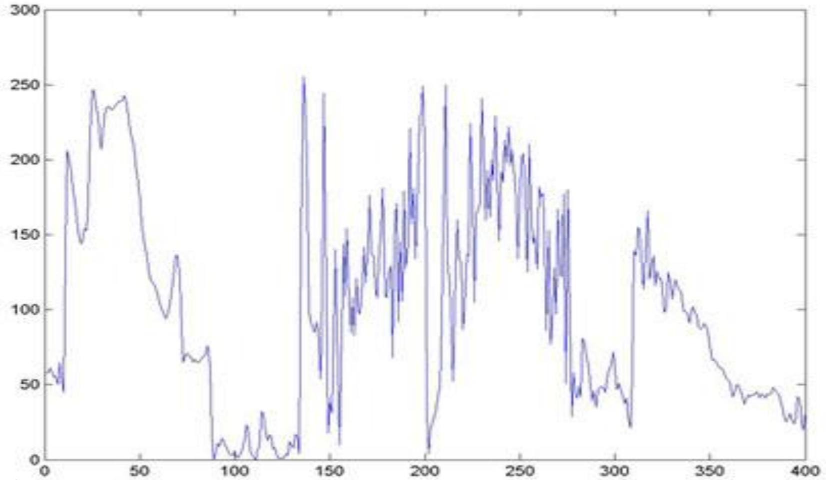

(a) Representasi citra RGB ke dalam 3 buah band yang berbeda : $I_{Red}$, $I_{Green}$, $I_{Blue}$

(b) RGB *pixel vector* terdiri atas RGB *pixel value* pada koordinat baris ($b$) dan kolom ($k$) yang sama

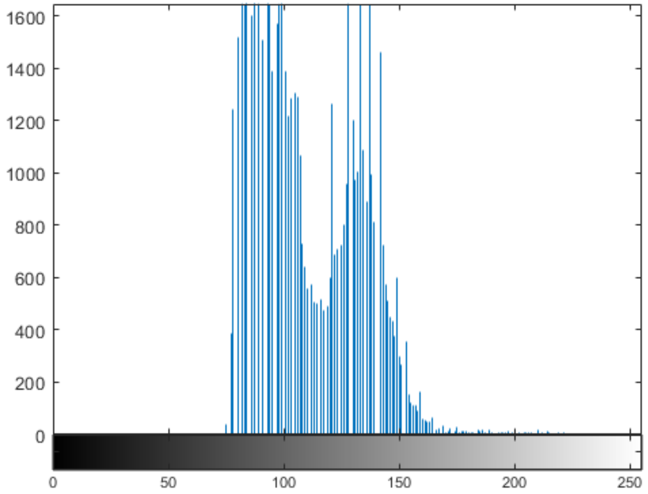
# Texture Descriptors

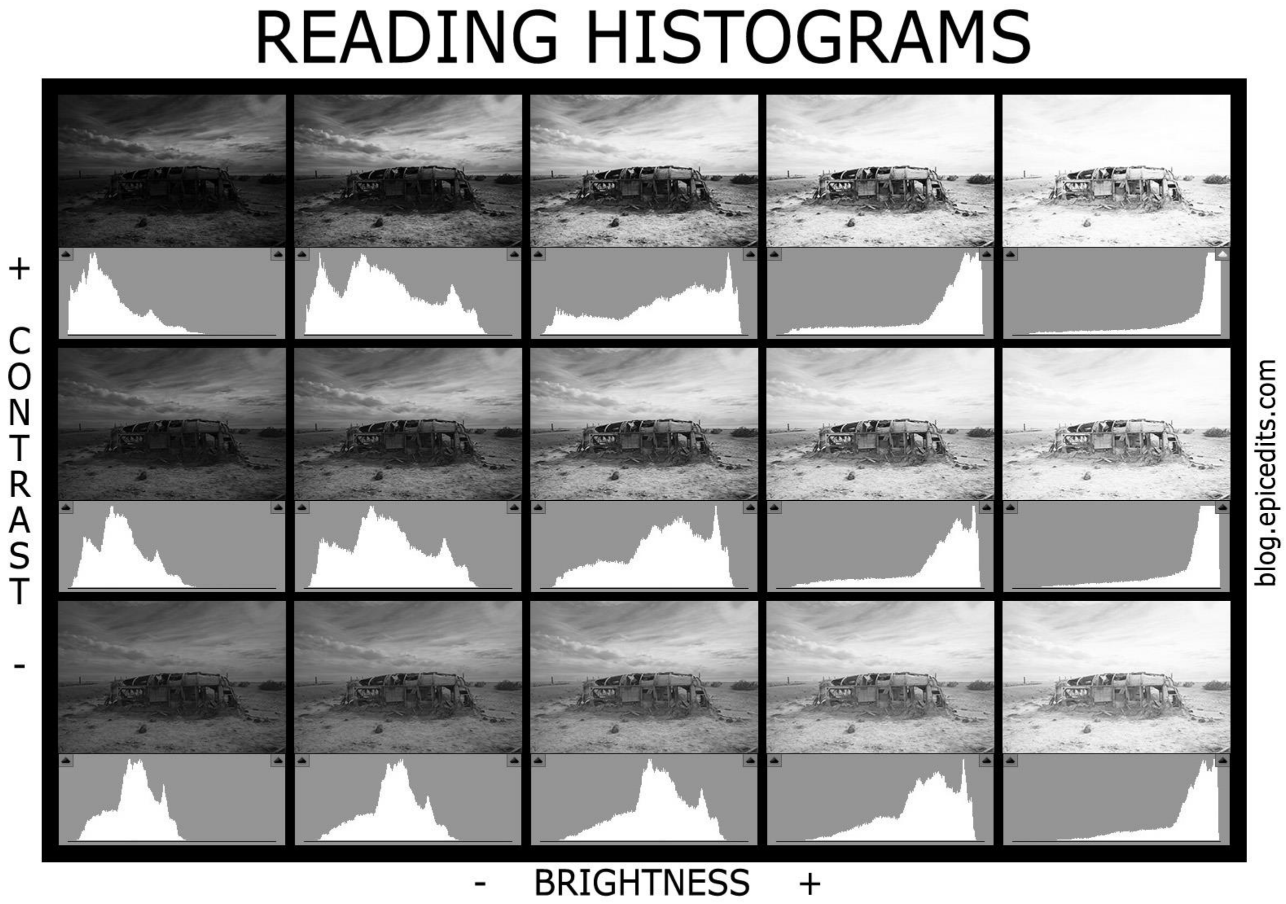
- Fourier Transform
- Hadamard Transform
- Wavelet Transform



# Statistical Based Features: Image Histograms

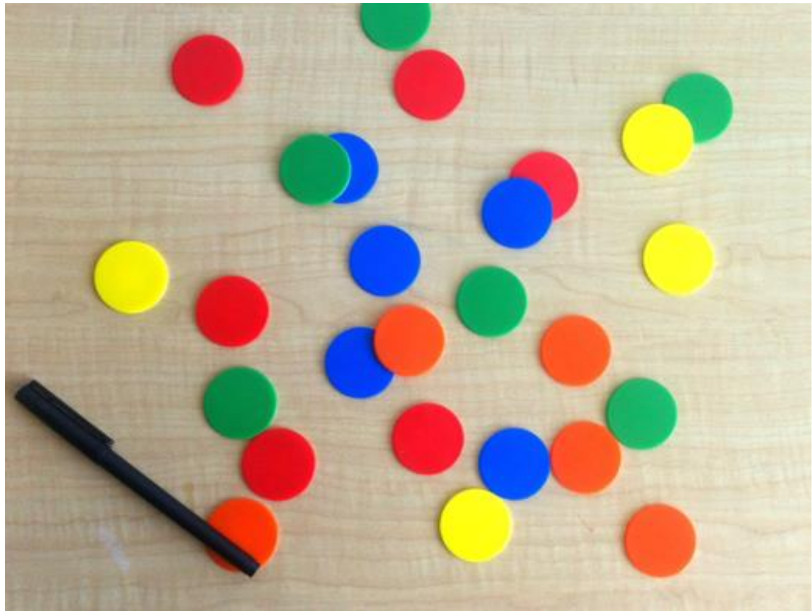


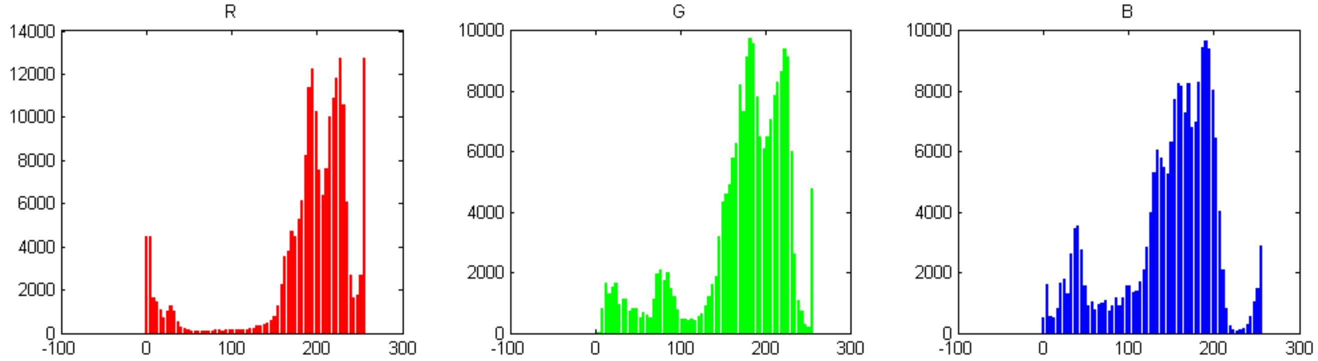




- Variations:
  - histogram of gray levels
  - histogram of gradients (HOG)
  - histogram of textons

# Statistical Based Features: Color Histograms

- Can be used as color features
- The challenge shall be in combining the color channel's histograms





# Statistical Based Features: Moments

moments = center of mass gambar kita

- Moments of the gray-level histogram z where discrete pixel intensity *(i = 1,2, … L)* and the $p(z_i)$ is the histogram, then $n^{th}$ moment of $z$ about mean is:

$$\mu_n(z) = \sum_{i=1}^{L} (z_i - m)^n p(z_i)$$

setiap pixel dikurangin dengan mean, terus dia ngitung berapa kali intensitas itu muncul di image.

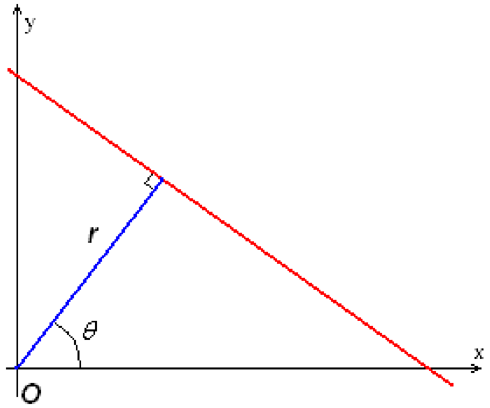
- where *m* is the mean of *z* (average intensity): $m = \sum_{i=1}^{L} z_i p(z_i)$

- The moments define the image texture based on the histogram:
  - 2nd moment gives the information about contrast (relative smoothness)
  - 3rd moment gives the information about skewness of the histogram
  - 4th moment gives the information about the flatness of the histogram

# Global Features: Hough Transform

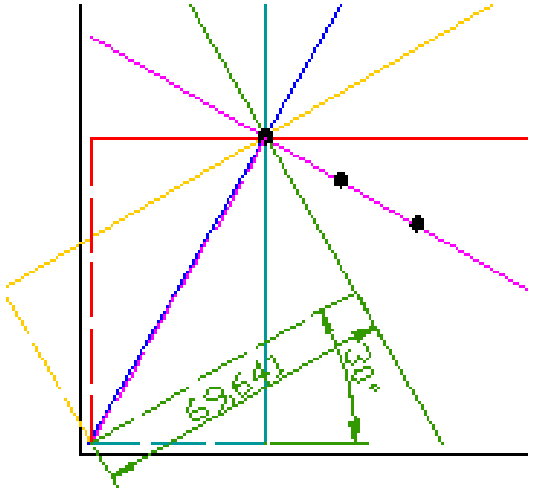
dapetin edge dari suatu image

- Implemented on edge detected images - Finds lines in an image, but also can be generalized to find circles or ellipses
- Line equation $r = x\cos\theta + y\sin\theta$




parameternya ini

- Every line is represented as $(\mathrm{r}, \theta)$
- The information is then stored in a table of features: the **accumulator array**
- For every **desired** angle:
  - Store the information of every line found in the image
  - Save line as $(\mathrm{r}, \theta)$

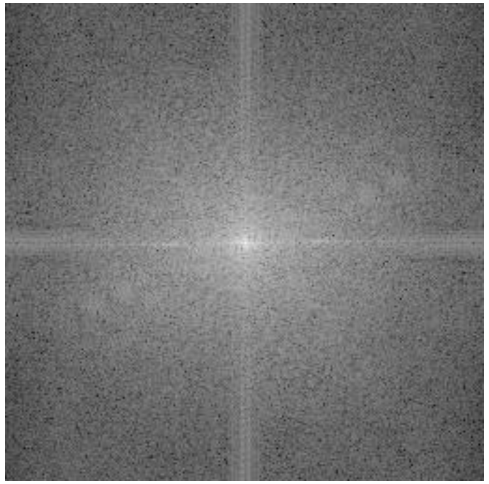



| Angle | Dist. |
|---|---|
| 0 | 40 |
| 30 | 69.6 |
| 60 | 81.2 |
| 90 | 70 |
| 120 | 40.6 |
| 150 | 0.4 |

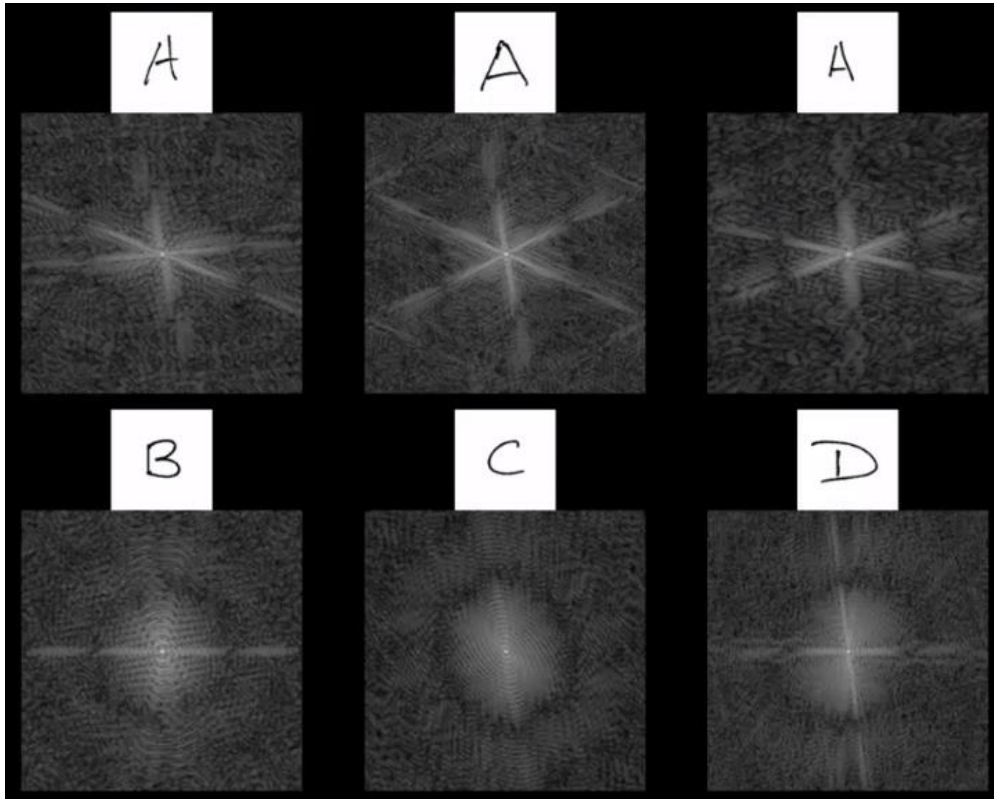
# Global Features: Fourier Transform

- The *Fourier* transform describes the image as a combination of various frequencies or textures or patterns

$f\ (x, y)$



$|F\ (u, v)|$




The Fourier transform for each letter is distinct

# Types of Features

## Raw features

- are obtained directly from the data
- no extra data manipulation or engineering

## Derived features

- are obtained from feature engineering
- created features from existing data attributes.
- manipulated the existing data/ features into a better representation

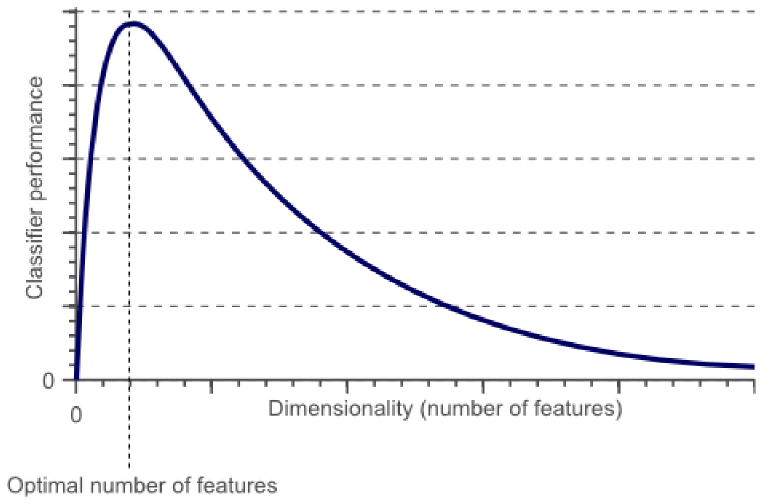
# Feature Selection and Dimensionality Reduction

# Discussion

- What attributes would you use to describe “Orange”??
- Do you need all those attributes to distinguish “Orange” from “Apple”?

# Hughes Phenomenon (Landgrebe *et al.*, 1993)

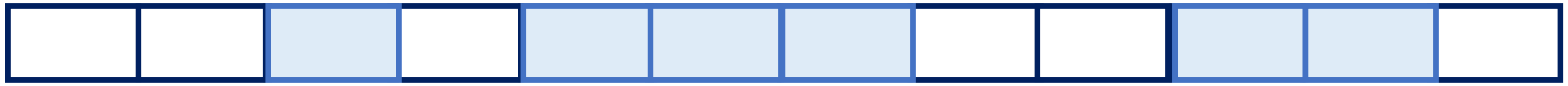
- Curse of dimensionality deal with this with PCA maybe
  - A higher dimension of feature does not guarantee a higher accuracy rate



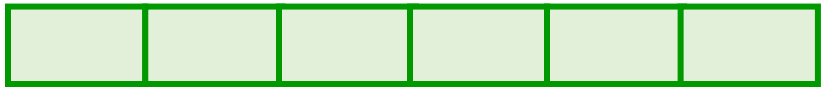

- The dimensionality of a feature will improve accuracy up to a certain point, beyond that it will lower accuracy it will cause overfit

# What Can We Do?

- If we have a 12-dimensional feature vector



  - Feature Selection
  - Dimensionality Reduction



# Feature Selection vs Dimensionality Reduction

## Feature Selection

- Choosing a subset of the previously described features to represent our data
- Filtering irrelevant or redundant features from your database

## Dimensionality Reduction

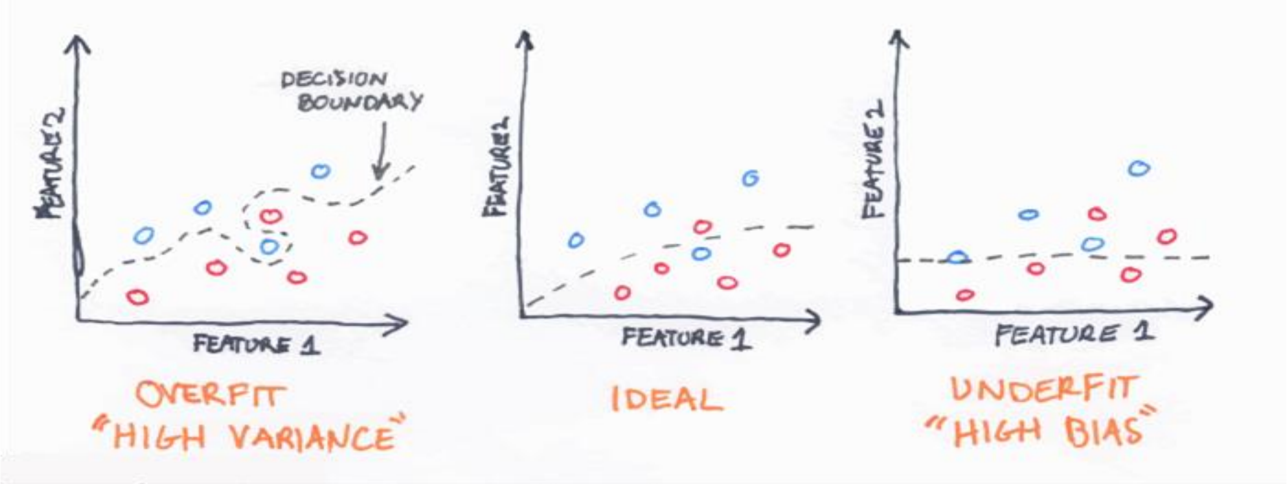
- Choosing a new mathematical representation with which you can describe most of the variance within your data
- Does **not** need to be a subset of previously described features

- Why?
  - Data reduction for fast computation
  - It reduces the complexity of a model and makes it easier to interpret.
  - Obtaining the best classification result
  - It reduces overfitting.

# Overfitting vs Underfitting

- **Overfitting** is a modeling error which occurs when a function is too closely fit to a limited set of data points.
  - Captures the noise of the dataset
  - The feature dimensions are too specific to the dataset

- **Underfitting** occurs when our model has such a low representation power that it cannot model the data
  - When the model cannot achieve good performance even on the training set.




Credit: Christ Albon https://builtin.com/data-science/curse-dimensionality

# Feature Selection

- Sequential Forward Selection (SFS)
  - Bottom-up approach : starting with an empty set, add features until optimal
- Sequential Backward Selection (SBS)
  - Top-down approach : starting with all possible features, remove features until optimal

Once a feature is introduced to the feature set, we can no longer remove it (SFS)
Once a feature is removed from the feature set, we can no longer add it (SBS)

- Sequential Forward Floating Selection (SFFS)
  - Extension of the ‘plus l - take away r’ method (Jain & Zongker, 1997).

# Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS)

**Sequential Forward Selection (SFS)**

- Bottom-up approach : starting with an empty set, add features until optimal
- From n features find the optimal m features (m < n).
  1. Order features from best to worse
  2. Add best features sequentially,
  3. Combine added features and evaluate the criterion (accuracy or other targets)
  4. Repeat 2 and 3 until k = m.

**Sequential Backward Selection (SBS)**

- Top-down approach : starting with all possible features, remove features until optimal
- From n features find the optimal m features (m < n).
  1. Initial set of n features, order features from best to worse
  2. Remove worst features sequentially,
  3. Use remaining features and evaluate the criterion (accuracy or other targets)
  4. Repeat 2 and 3 until k = m.

# Sequential Forward Floating Selection (SFFS)

cari dari berbagai fitur yang dievaluasi, buang yang nilainya paling keicil

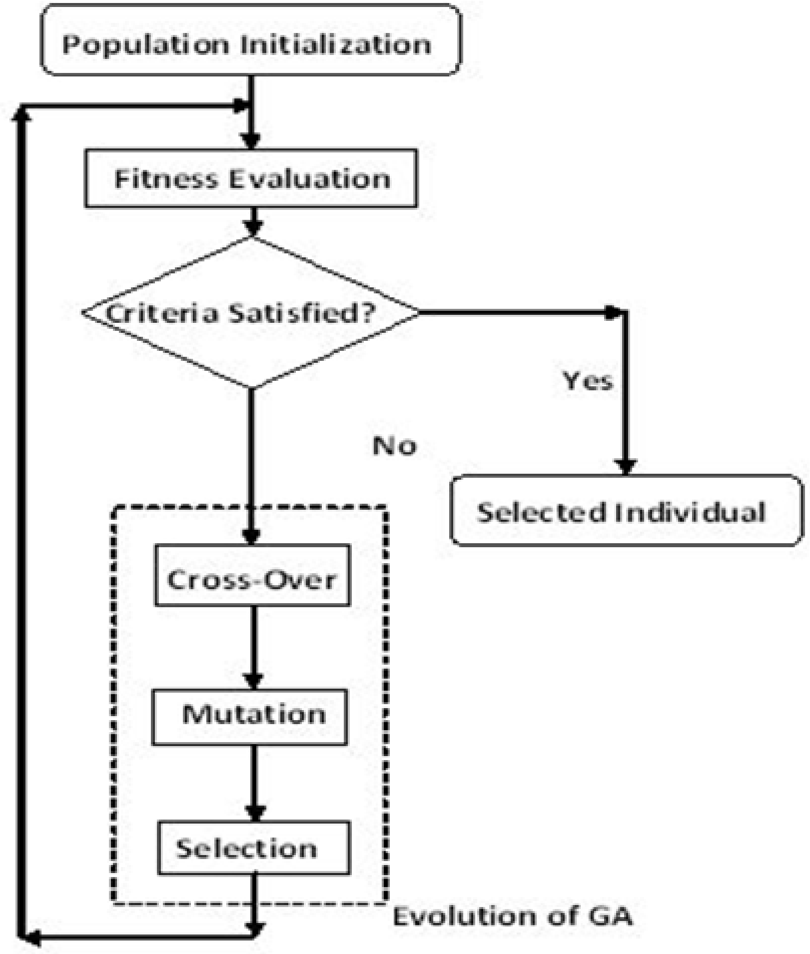
- Extension of the ‘plus l - take away r’ method (Jain & Zongker, 1997).
  - Start with feature set $X = 0$, add two most significant features to $X$.
  - Suppose $m$ features will be selected, then the next steps are:
  - **Step 1 - Inclusion:** If the current subset $X$ has cardinality $m$, stop. Otherwise, add the most significant feature with respect to $X$.
  - **Step 2 - Conditional exclusion:** Find the least significant feature $k$ inX. If it is the feature just added, then keep it and return to step 1. Otherwise, exclude feature $k$
  - **Step 3 - Continuation of conditional exclusion**: Find the least significant feature in $X$. If its removal will leave $X$ with at least two features and the value of the criterion function is greater than the criterion value of the best feature subset of the size found so far, then remove it and repeat step 3. When these two conditions cease to be satisfied, return to step 1.

# How Do We Judge the Combination at Each Step?

- Use a Criterion Function
- Filter-based: check at every iteration process (distance or correlation of feature)
  - Check the distance between each combination of feature
  - The best combination is the one with the largest distance (we can use like Bhattacharya distance)
- Wrapper-based: check at every iteration process (the task result: classification or segmentation error)
  - Check the classification error (accuracy)
  - The best combination of feature is the one with the best accuracy

# Genetic Algorithm Based Feature Selection

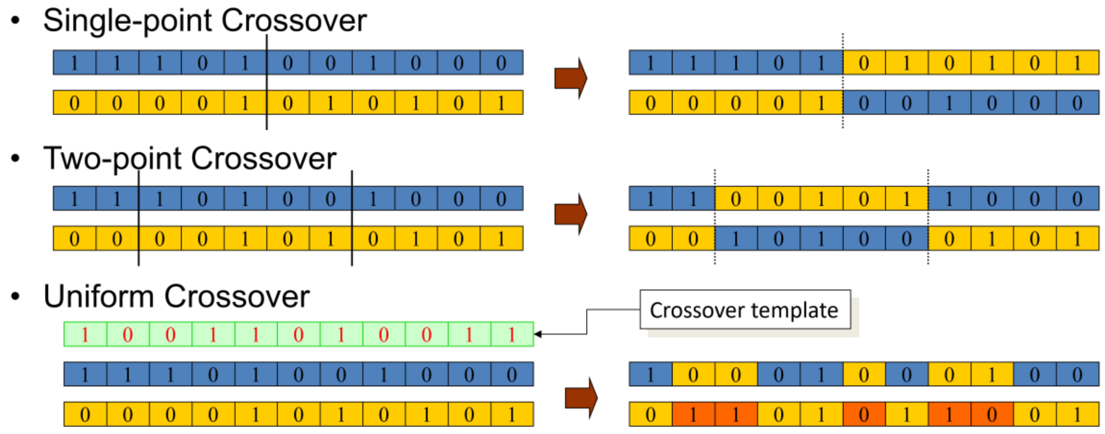
- Genetic algorithm (GAs) is a search algorithm that tries to model natural evolution on biological entities (Goldberg, 1989).
    - The feature vector – modelled as chromosomes
    - Uniform cross over, mutation flips the value of a specific bit, mating/crossover of chromosomes and mutation creates a new generation of chromosomes
    - Fitness function is a function of classification error, select n best individuals at each iteration
    - Finally, a set of bit 1 position indicates the obtained set of features.



# Genetic Algorithm based Feature Selection

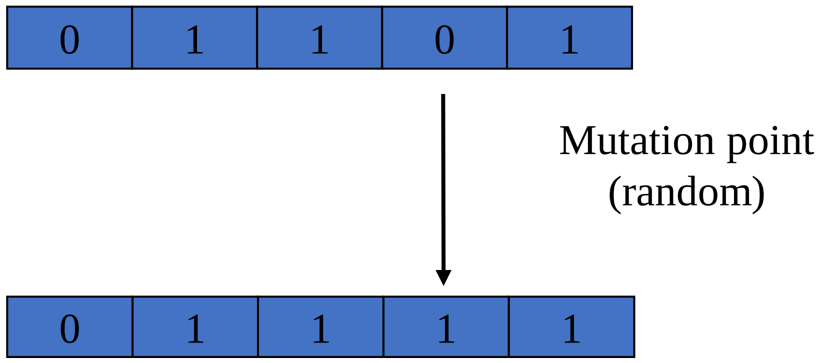
## Crossover

- Tsai-Yang Jea, Dept. of Computer Scie. & Eng., SUNY at Buffalo




## Mutation

- Usually change a single bit in a bit string, with very low probability.

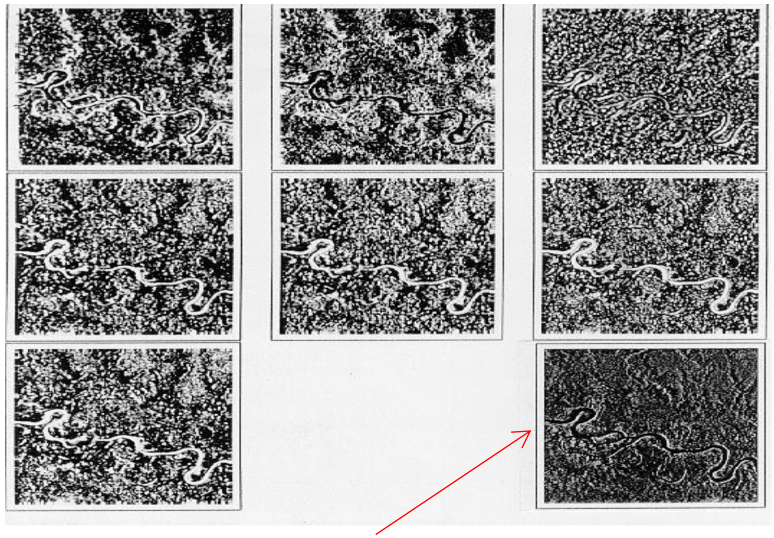


# Manual Feature Selection vs SFFS

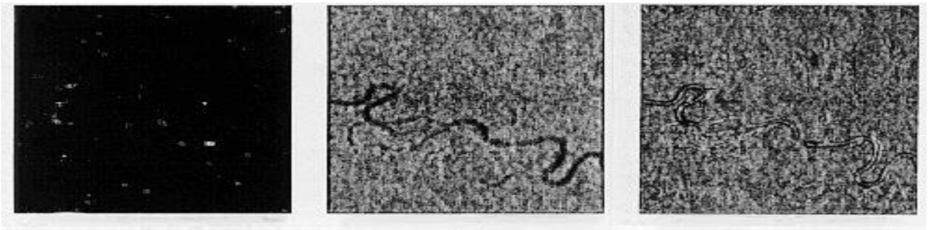
(Source: Murni *et al*., 1994; Murni *et al*., 1999)

- 12 feature images
  (Source: Murni *et al*., 1994; Original image is courtesy of Raimadoya, IPB)

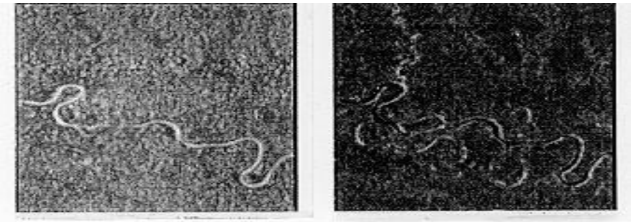
ekstrak fitur dari satelit



Bottom right: The original image of SAR Star-1 X Band of East Kalimantan area; from Left to Right and Top to Bottom are GLCM features: contrast, entropy, correlation, inverse difference moment, homogeneity, maximum probability, and energy image.



From Left to Right are TU features: Black-White Symmetry (has low variance or information), Degree of Direction, and Geometric Symmetry image.



Local Statistic features: Local Mean (Left) and Local Ratio between Standard Deviation and Mean value.

# Manual Feature Selection vs SFFS (2)

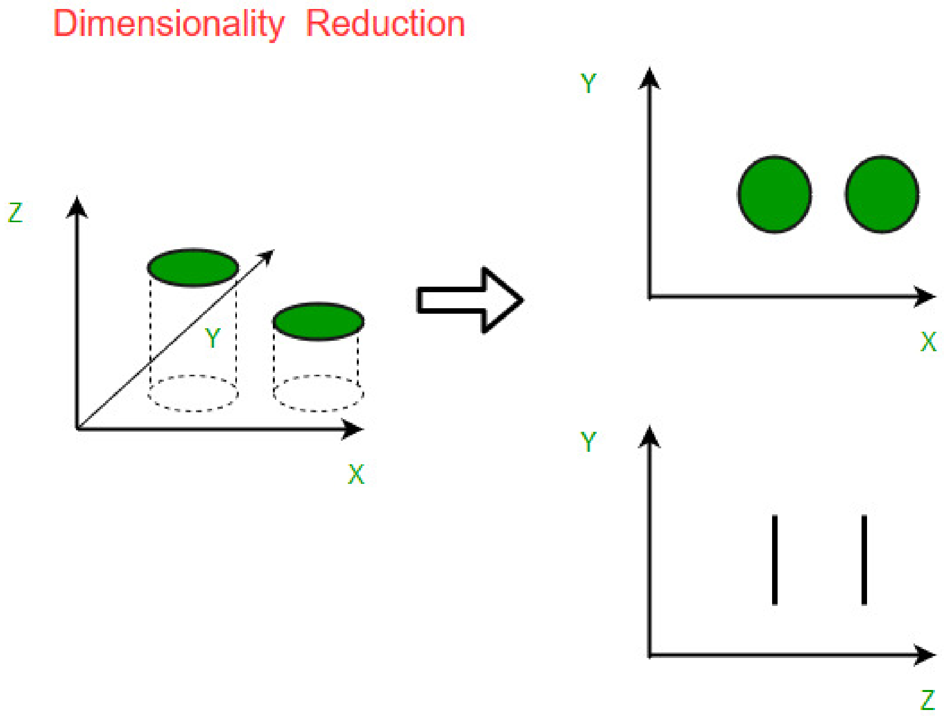
(Source: Murni *et al*., 1994; Murni *et al*., 1999)

- Manual selection:
  - We can choose: 2 features from GLCM (contrast and inverse difference moment), 2 features from TU (degree of direction, geometry symmetry), and 2 features from statistic (local mean, and local ratio between standard deviation and mean value) as a set of 6 features.
- Correct classification result is 94.5%.

- SFFS
  - The obtained set of features consists of: 3 features from GLCM (contrast, maximum probability, correlation), 1 feature from TU (geometry symmetry), and 2 features from statistic (local mean, and local ratio between standard deviation and mean value)
- Correct classification result is 94.3%.

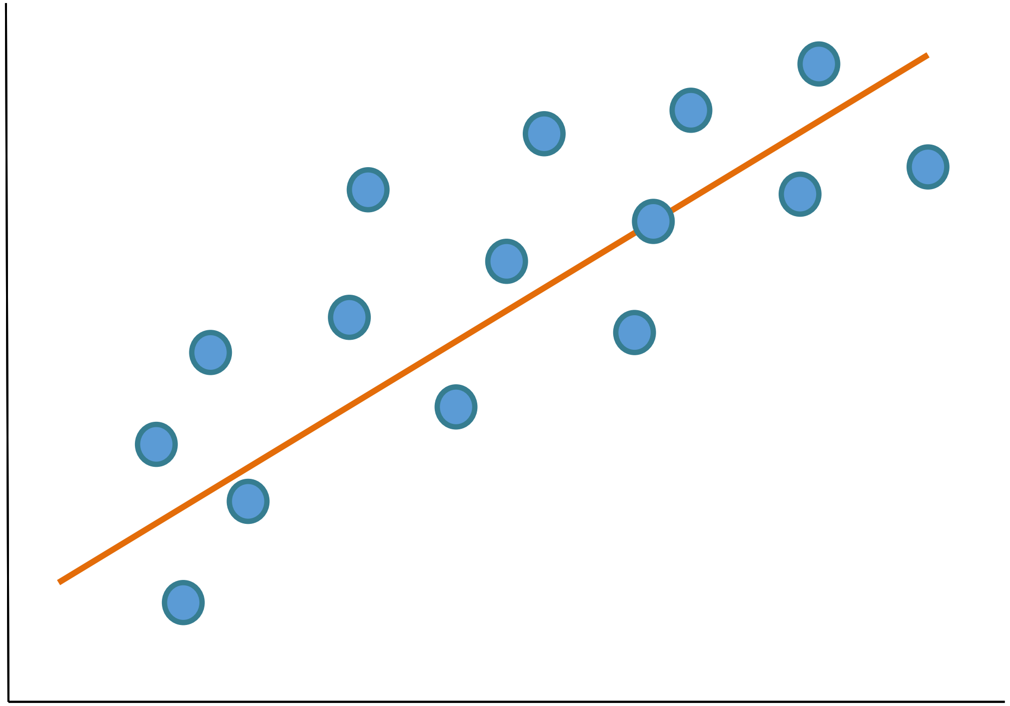
# Dimensionality Reduction

- Your data can be represented by many variables, not all of them are relevant!
- Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining as much of the important information as possible.
- Goal: find a lower dimensional representation of the data matrix



# Principle Component Analysis (PCA)

- PCA combines our input variables in a specific way and orders them by importance
- We can drop the “least important” variables while still retaining the most valuable variable
- Each of the “new” variables after PCA are all independent of one another.
- We utilize eigenvalues and eigenvectors



# Other Dimensionality Reduction Methods

- Linear Discriminant Analysis (LDA)
- Singular Value Decomposition (SVD)
- t-distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)

1) https://machinelearningmastery.com/linear-discriminant-analysis-for-dimensionality-reduction-in-python/
2) https://gregorygundersen.com/blog/2018/12/10/svd/
3) https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
4) https://umap-learn.readthedocs.io/en/latest/

# Hand-Crafted vs Machine Generated Features

## Hand crafted

- The features we have studied **hand-crafted features** that we extract and select to represent our data
- We choose what aspect to be represented (color, texture, shape)
- We extract the correlating feature

## Machine generated

- It is possible to also use **machine-generated features** that we do not determine beforehand, but a machine computes it for us.
- Feature maps from convolutional neural network – to be discussed later in the semester

Stay tuned for Deep Learning topics!

# Feature Fusion

# Complex Features

- Data captured from a sensor system: primary features
  - Primary features are the result of an image acquisition using any sensor.
  - When the primary features could not work well in a recognition process, then secondary, tertiary, etc. features are needed
- Secondary features usually are a result of any transformation of primary features
  - To increase the discrimination of objects in an image.
  - Sometimes one is not enough
- We can have either unimodal multifeatures or multimodal data
  - Unimodal: 1 source/domain, multifeatures: many descriptors
  - Multimodal: multi source/domain, resulting in many descriptors

# Feature Exploration for Prediction of Potential Tuna Fishing Zones

(D. Fitrianah, N. H. Praptono, A. N. Hidayanto, and A. M. Arymurthy, DOI: 10.7763/IJIEE.2015.V5.543)

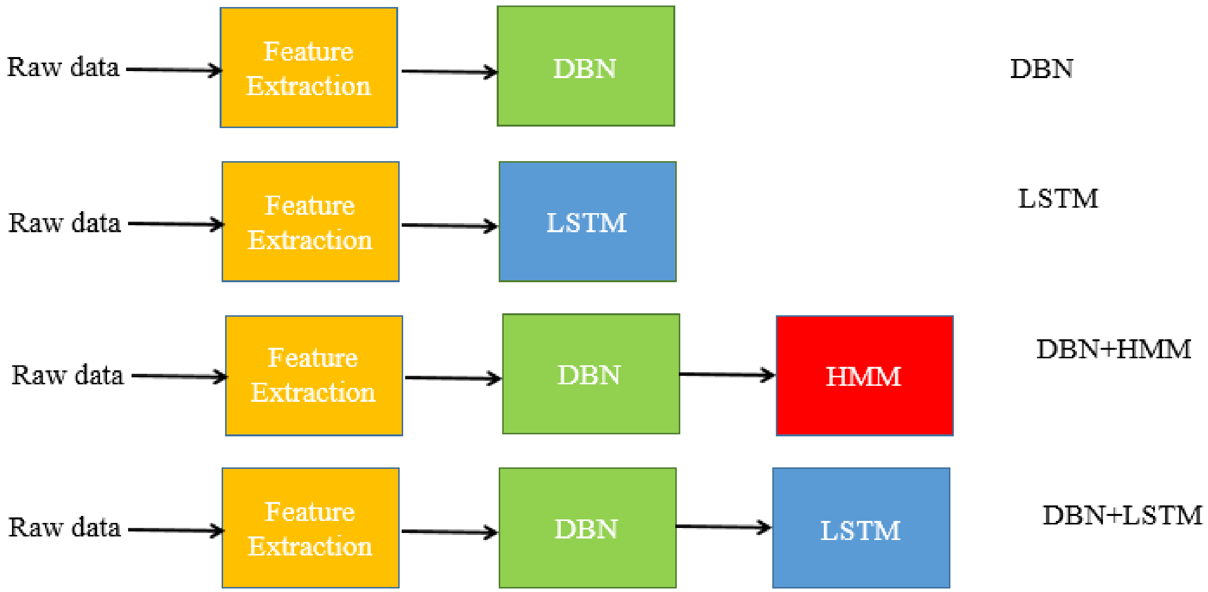
- Features:
  - Usually used features: chlorophyll and sea surface temperature (SST)
  - This research used:  chlorophyll, SST, plus ocean current and salinity
- Classifiers: Naïve Bayes, Decision Tree and SVM

| Feature Combination | Naïve Bayes | | Decision Tree | | SVM | |
|---|---|---|---|---|---|---|
| | average Acc (%) | average $k$ | average Acc (%) | average $k$ | average Acc (%) | average $k$ |
| f1: (Baseline) Chlorophyll, SST | 57.44 | 0.1490 | 58.91 | 0.1780 | 56.74 | 0.1348 |
| f2: Chlorophyll, SST, Ocean Current | 65.28 | 0.3054 | 64.16 | 0.2831 | 66.29 | 0.3253 |
| f3: Chlorophyll, SST, Salinity | 62.62 | 0.2522 | 82.07 | 0.6412 | 63.55 | 0.2712 |
| f4: Chlorophyll, SST, Salinity, Ocean Current | **69.03** | **0.3807** | **82.32** | **0.6462** | **68.30** | **0.3658** |

# Combining Generative and Discriminative Neural Nets for Sleep Stages Classification

(Endang P. Giri, M. Ivan Fanany, and Aniati M. Arymurthy, https://arxiv.org/abs/1610.01741)

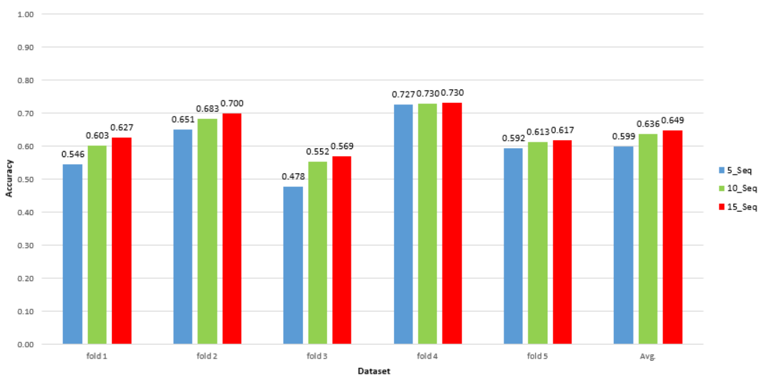
- A novel classification model for predicting sleep stages with a high accuracy
- Methods: the **generative** capability of **DBN** (Deep Belief Network) and smart time sequence **discriminative** capability of **LSTM** (Long Short-Term Memory).
- Comparison: DBN, LSTM, DBN+HMM (Hidden Markov Model), and DBN+LSTM



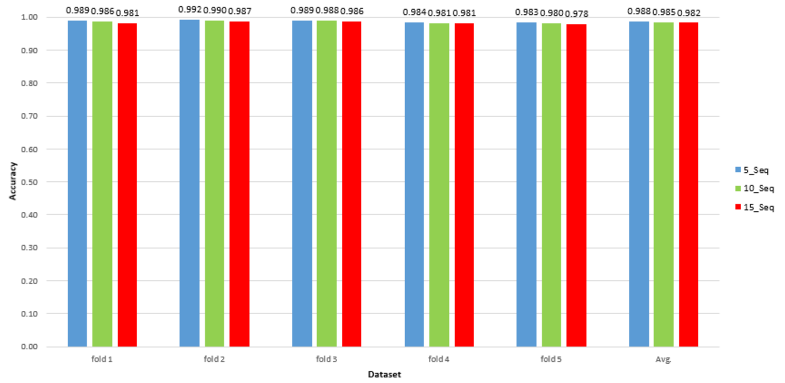
# Combining Generative and Discriminative Neural Nets for Sleep Stages Classification (2)

(Endang P. Giri, M. Ivan Fanany, and Aniati M. Arymurthy, https://arxiv.org/abs/1610.01741)

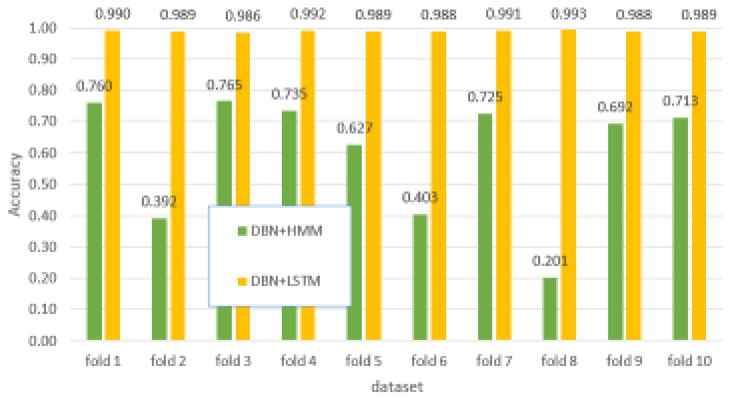
- Sleep stages are classified per a particular length of signal (epoch of 30 seconds)
- Sleep stages could be the stage of Wake (W), stage 1 (N1), stage 2 (N2), stage 3 (N3) and REM (R)
- Sequence of sleep stages could be W, N1, N2, N3, N2, R, N2, N3, etc.




LSTM accuracy for each fold




DBN+LSTM accuracy for each fold




Accuracy DBN+HMM and DBN+LSTM

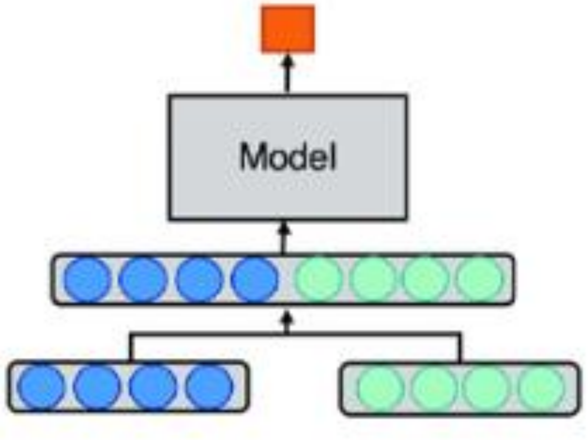
# Multimodal Data Examples

| | |
|---|---|
| Prediction of Alzheimer's disease based on magnetic resonance imaging and positron emission tomography (images) that are performed multiple times on one patient within specified periods of time (time series). Patient demographics and genetic data are also taken into account (tabular). | El-Sappagh, et al. 2020. doi: 10.1016/j.neucom.2020.05.087. |
| Question answering based on images containing some textual data. | Singh , et al. CVF Conference on Computer Vision and Pattern Recognition. 2019. |
| Outfit/movie recommender systems. Movies are recommended based on plot (text), poster (image), liked and disliked movies, and cast (graphs). Outfits are chosen based on product features in images and text descriptions | Rychalska et al. arXiv. 20202006.09979; Laenen and Moens, 2020; doi: 10.1016/j.ipm.2020.102316; Salah, et al.; 2020. |

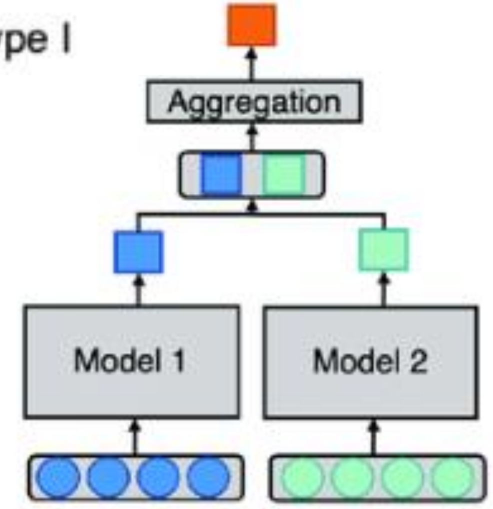
# Multimodal Feature Fusion

Boulachia, et al. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. 2021.

- Early fusion: Multimodal data fused at input level before any model
- Intermediate fusion: Process each data into a latent representation, then input to model
- Late fusion: Each data processed separately, then the outputs are fused at decision level
- Hybrid fusion: a combination of the above three




Early Fusion – Type I

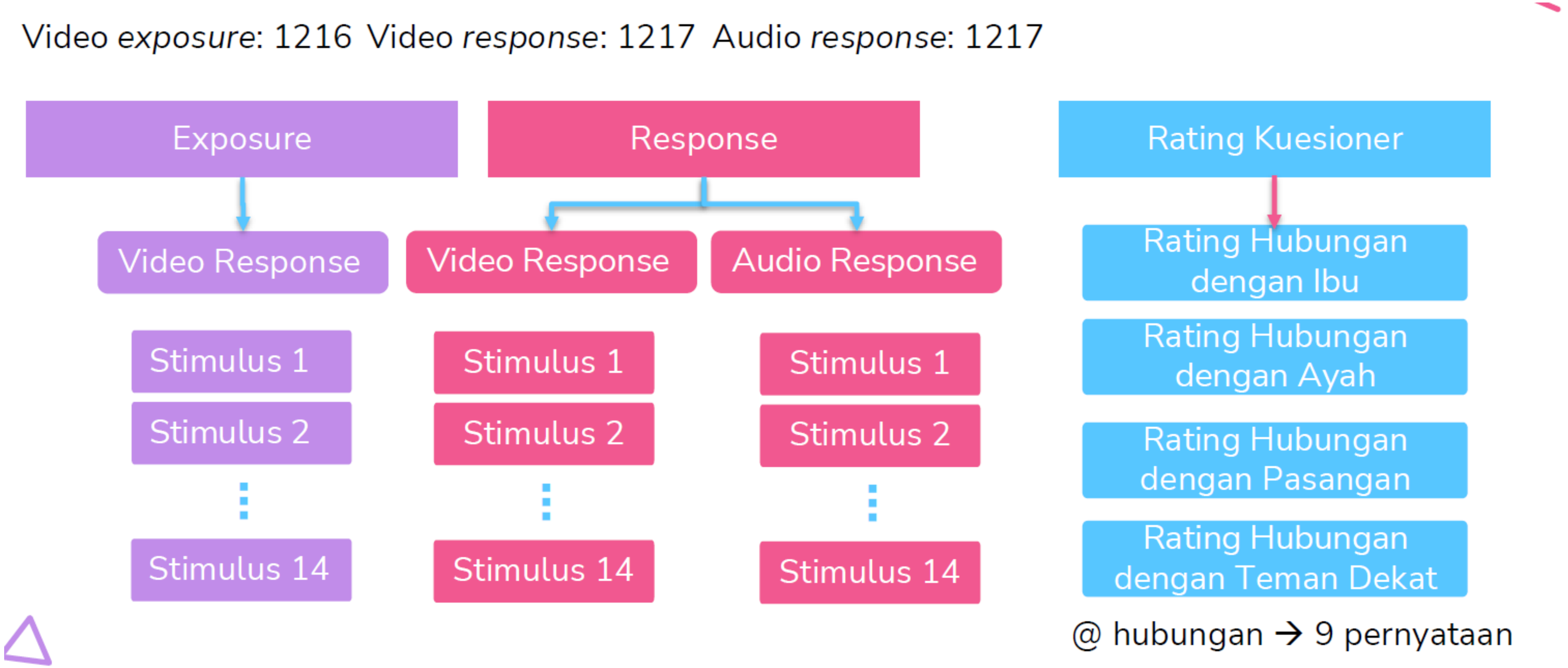



Late Fusion

# Attach-SwiNet: Multimodal Attachment Style Classification Model Based on Non-Verbal Signals

Maghfira, T. N., Basaruddin, T., Alfa Krisnadhi, A., and Redatin Retno Pudjiati, S. (2024).

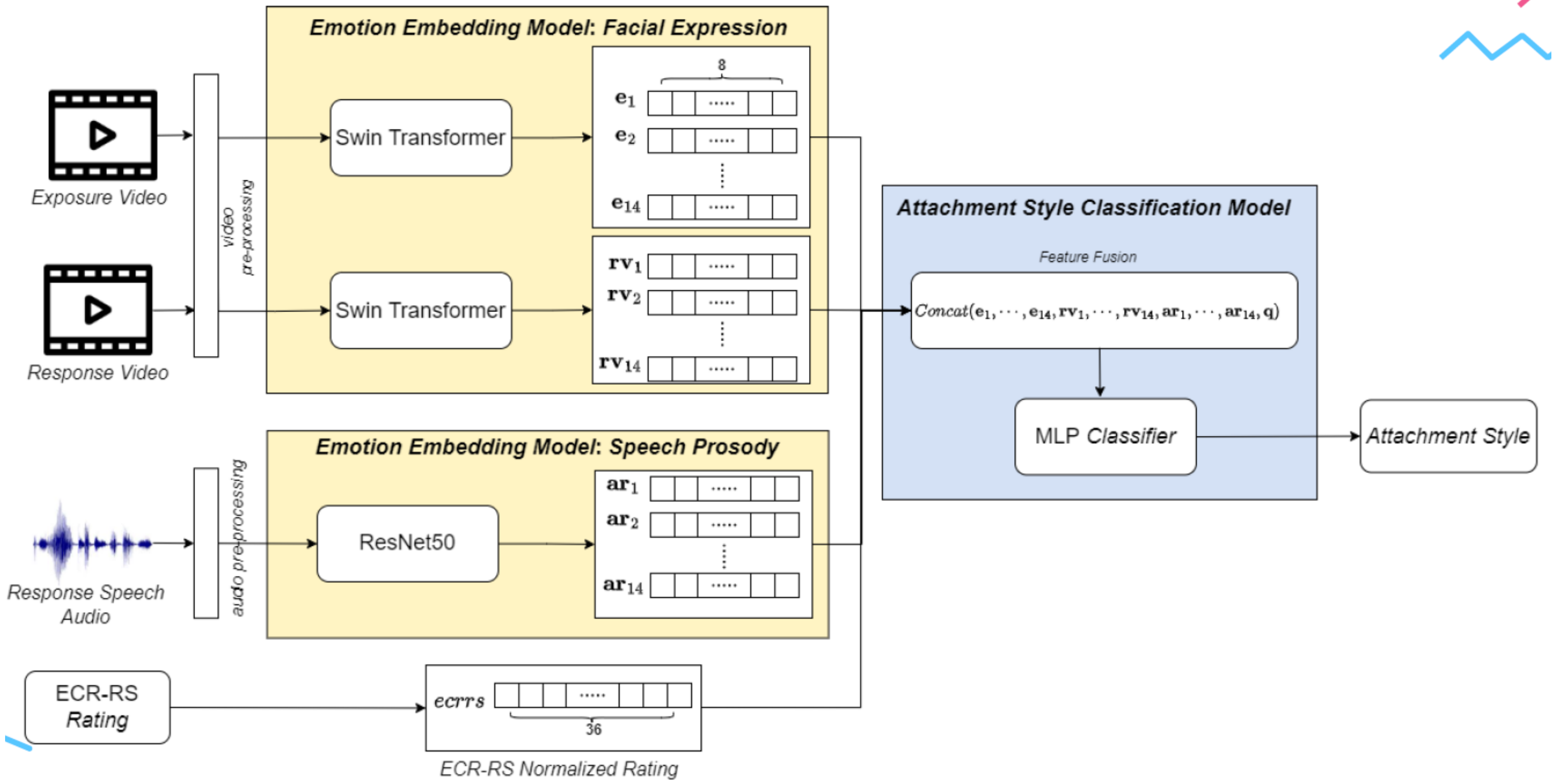
- Attachment style classification: video and audio recording plus questionnaire



# Attach-SwiNet: Multimodal Attachment Style Classification Model Based on Non-Verbal Signals (2)

Maghfira, T. N., Basaruddin, T., Alfa Krisnadhi, A., and Redatin Retno Pudjiati, S. (2024).

- Early fusion:



# Attach-SwiNet: Multimodal Attachment Style Classification Model Based on Non-Verbal Signals (3)

Maghfira, T. N., Basaruddin, T., Alfa Krisnadhi, A., and Redatin Retno Pudjiati, S. (2024).

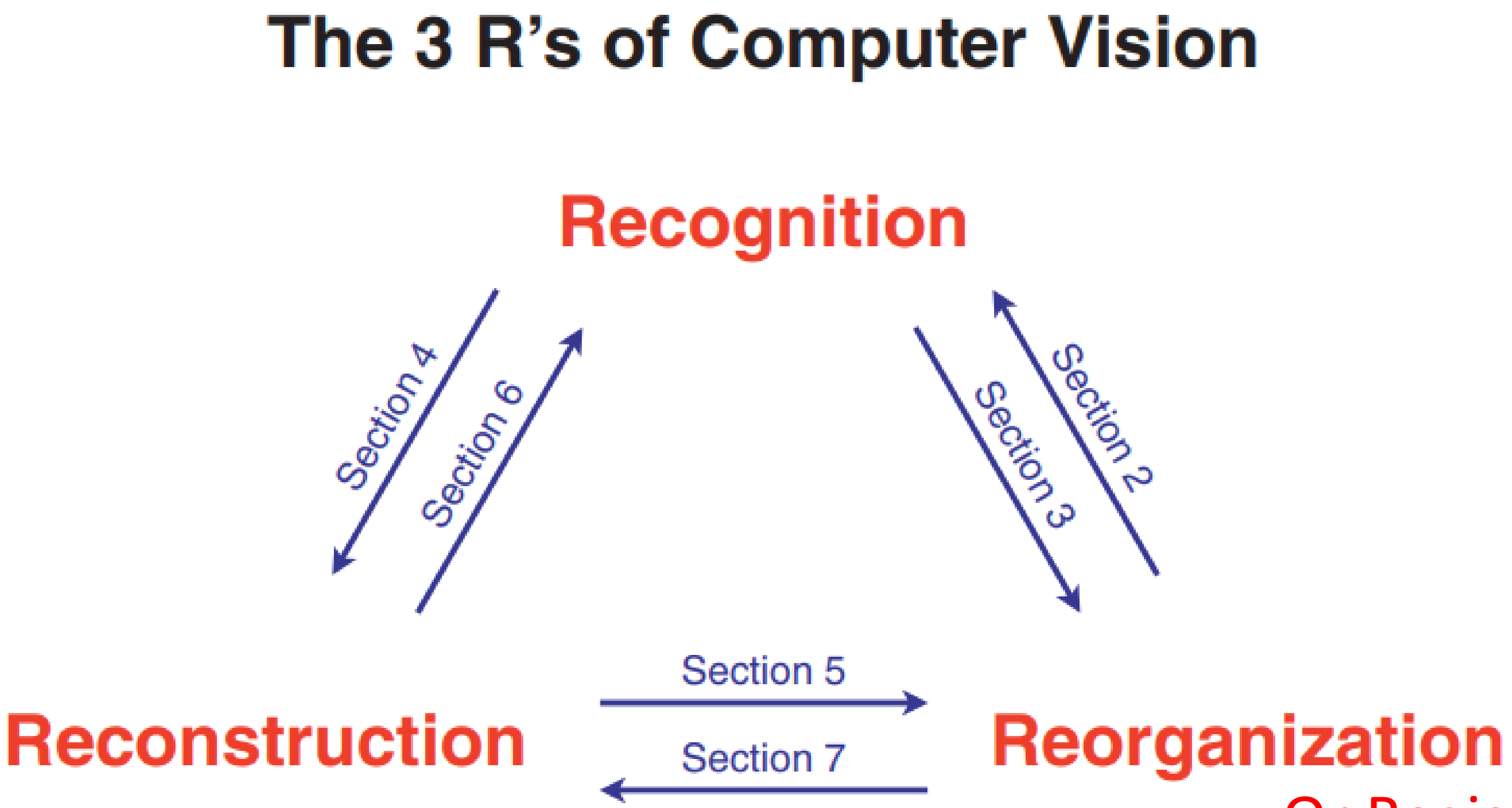
- All modalities show the best result

| Model | Presisi | *Recall* | *F1* |
|---|---|---|---|
| VQ | 0.7463 | 0.8571 | 0.7943 |
| AQ | 0.7475 | 0.8929 | 0.8007 |
| EVA | 0.7231 | 0.8929 | 0.7944 |
| EVQ | 0.7358 | 0.9286 | 0.8191 |
| EAQ | 0.7289 | 0.9464 | 0.8180 |
| VAQ | 0.8578 | 0.7321 | 0.7672 |
| Semua Modalitas | 0.7735 | 0.9286 | 0.8392 |

# Recognition: Image Classification

# The 3 R's of Computer Vision

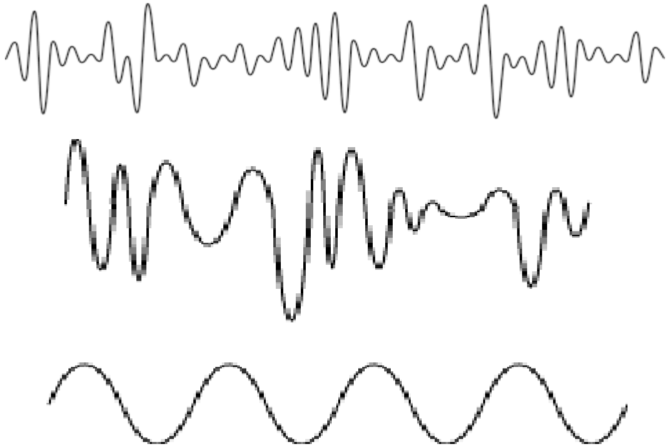
Malik, Jitendra, et al. "The three R's of computer vision: Recognition, reconstruction and reorganization." *Pattern Recognition Letters* 72 (2016): 4-14.



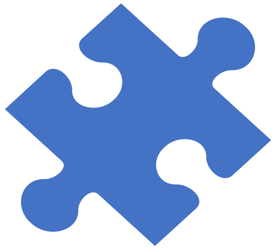

Or Registration

# Pattern Recognition

- What is a pattern?
  - A regularity
  - An arrangement of descriptors
  - For example: sound waves, fingerprints, facial expressions, etc



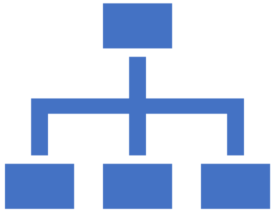
# Finding the Pattern Classes



**Clustering (unsupervised):**

Grouping a set of objects so that objects in the same **cluster** are more similar to each other than to those in other clusters

→ Pattern clusters



**Classification (supervised):**

Assigning an input image one label / class from a fixed set of categories.

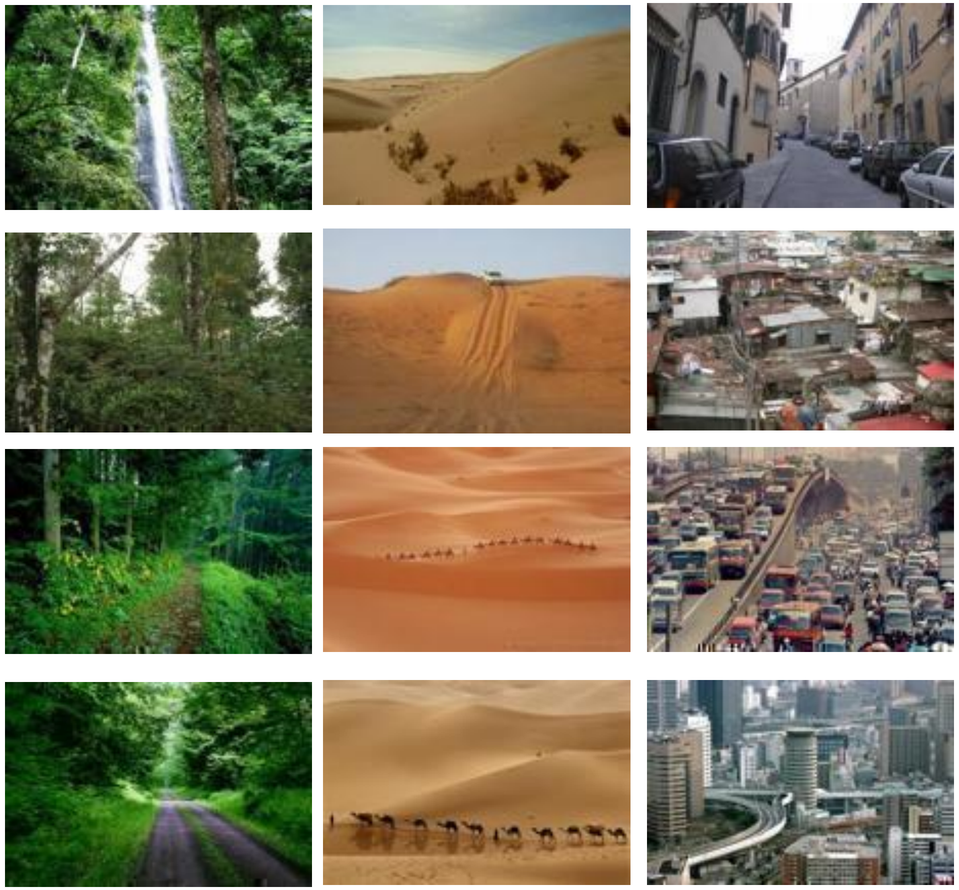
→ Pattern classes

# Image Classification

- Assigning an input image one label from a fixed set of categories:
  - Categories: Forest, desert, city

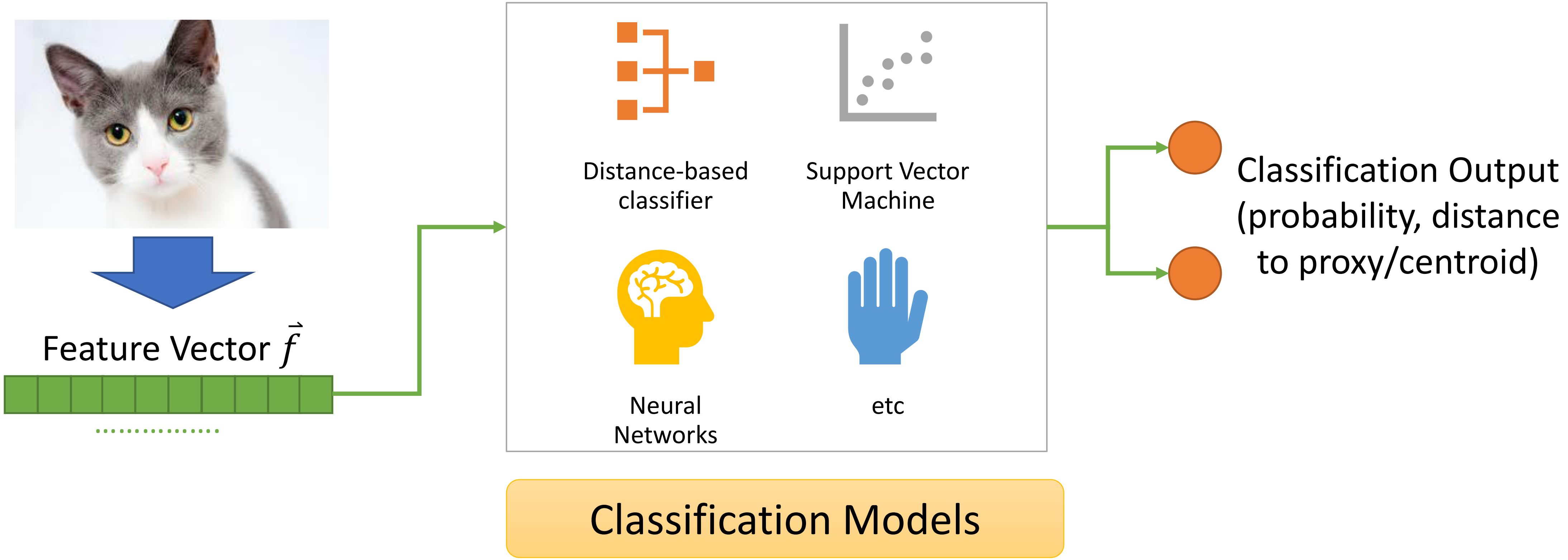
We need labels!!!

- Then any next data point (image) can be classified to the closest "label"
  - Plus/minus



# Machine Learning for Image Classification

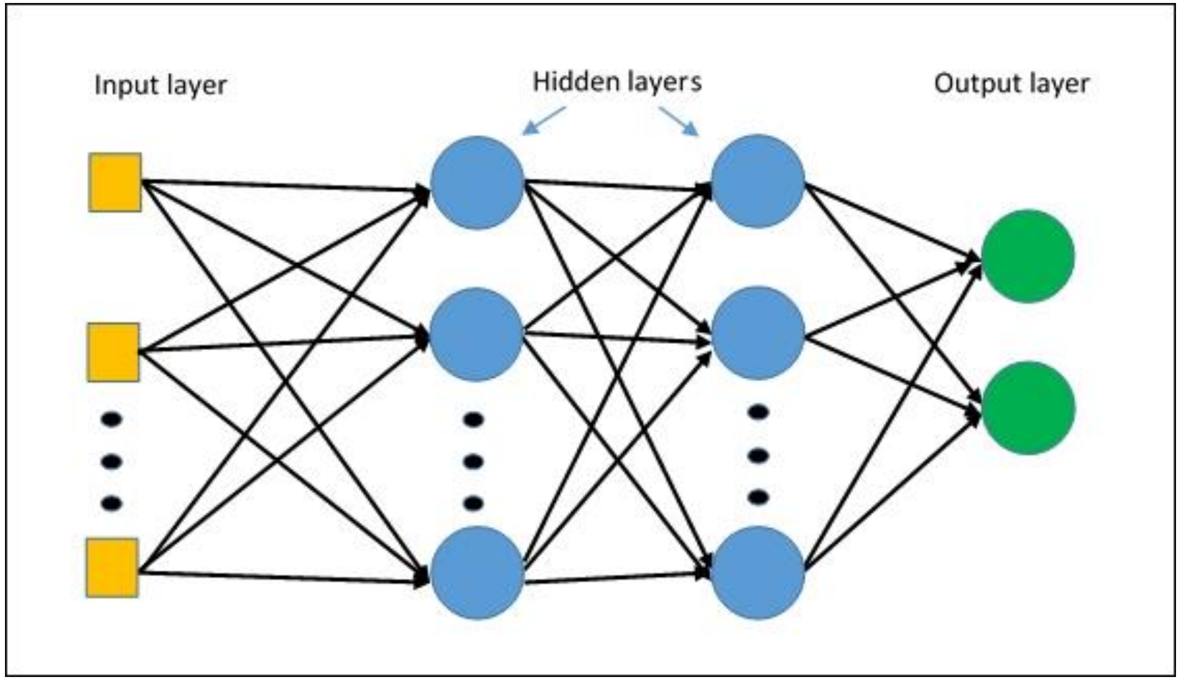
- Cat or dog?



# Some Classification Methods

| | |
|---|---|
| K-nearest Neighbors classifier | Distance-based classifier<br>Lazy learners, doesn’t build models explicitly |
| Support Vector Machine | Determines hyperplanes to separate data per classes |
| Neural Networks | Follows the biology of neurons in human brains<br>Will be covered in more detail in lead up to CNN |
| etc | |

# Neural Networks: Multilayer Perceptron




- MLP is a logistic regression classifier where the input is transformed using a learnt non-linear transformation
- Input is the feature vector
- Weights on each connector
- Activation function at each node

Stay tuned for deep networks topic!

# Performance Evaluation

Accuracy

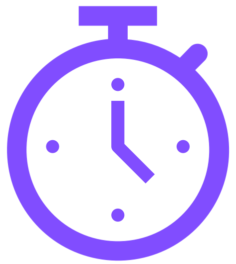
Speed

Robustness

Scalability

Interpretability

# Speed and Robustness



## Speed

Some methods require shorter computation times than others

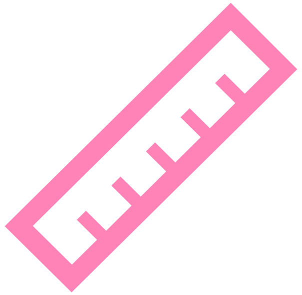
Why is it important?

## Robustness

The performance of a classifier depends on the training set.

A classification method is robust if the performance does not vary significantly as the training set varies.

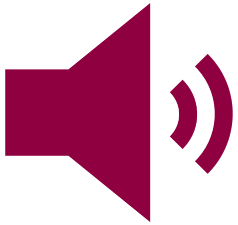
# Scalability and Interpretability



## Scalability

The scalability of a classifier refers to its ability to learn from infinitely larger datasets.

Also related to its computation speed.



## Interpretability

Classification aims to predict classes - **but -** classification analysis can be done to interpret and understand the data as well as predict.

The rules and results should be simple and easily understood by domain expert

# Accuracy

- Tested with a **testing set!**
  - Set of data input without labels
  - Let the model assign labels based on its training
  - The true labels are known to compare it with the results

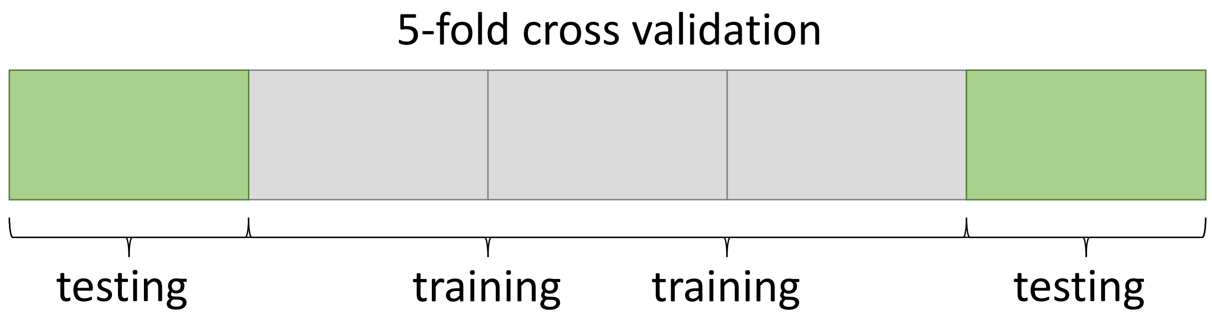
Data in the testing set must be separate from data in the training set!

- Accuracy / recognition rate is commonly measured in percentage
  - The percentage of the test set that is recognized correctly

$$Accuracy = \frac{no\ of\ correctly\ classified\ test\ data}{total\ no\ of\ test\ data} \times 100\%$$

# Cross-Validation

- From a single set of data → separated into smaller sections
- K-folds
  - Separate data into k sections




  - At each fold evaluate the results (e.g the accuracy)
  - Take the average

# Accuracy Interpretation

- What does it measure?
  - How well the model knows / understands the intended label of a data point
  - Only gives us how much is correct / incorrect
- In some instances, we need to know the **type** of error committed

| | |
|---|---|
| **True Negatives (TN)**<br>Observation is negative, and predicted to be negative. | **False Negatives (FN)**<br>Observation is positive, but is predicted negative. |
| **False Positives (FP)**<br>Observation is negative, but is predicted positive. | **True Positives (TP)**<br>Observation is positive, and predicted to be positive. |

# Confusion Matrix

- For a binary classification

| Predicted Values | True condition: Positive | True condition: Negative |
|---|---|---|
| Positive | TP | FP |
| Negative | FN | TN |

$$Recall: \frac{TP}{TP + FN}$$

$$Precision: \frac{TP}{TP + FP}$$

$$Accuracy: \frac{TP + FN}{Total}$$

- For multi-class classification

| True class \ Predicted class | airplane | automobile | bird | cat | deer | dog | frog | horse | ship | truck |
|---|---|---|---|---|---|---|---|---|---|---|
| airplane | 923 | 4 | 21 | 8 | 4 | 1 | 5 | 5 | 23 | 6 |
| automobile | 5 | 972 | 2 | | | | | 1 | 5 | 15 |
| bird | 26 | 2 | 892 | 30 | 13 | 8 | 17 | 5 | 4 | 3 |
| cat | 12 | 4 | 32 | 826 | 24 | 48 | 30 | 12 | 5 | 7 |
| deer | 5 | 1 | 28 | 24 | 898 | 13 | 14 | 14 | 2 | 1 |
| dog | 7 | 2 | 28 | 111 | 18 | 801 | 13 | 17 | | 3 |
| frog | 5 | | 16 | 27 | 3 | 4 | 943 | 1 | 1 | |
| horse | 9 | 1 | 14 | 13 | 22 | 17 | 3 | 915 | 2 | 4 |
| ship | 37 | 10 | 4 | 4 | | 1 | 2 | 1 | 931 | 10 |
| truck | 20 | 39 | 3 | 3 | | | 2 | 1 | 9 | 923 |

# Training Classifiers

- How do you teach a child to know that this is a phone?

- You show them samples

Training Data

# Training Classifiers (2)

- How do they understand what it is?

  - They take note of the features
- How do we ensure they understood correctly??

Give appreciation when they give correct recognition

Give punishment when they give incorrect recognition

Reinforcement - Supervised learning

# Training Data

- How much data do we need?
  - **Just enough**
    - Depends on the complexity of the classification and model
  - Too much data can cause overfitting → Delete some
  - Too little data can not train the model

Data Augmentation

- Which models benefit from data augmentation?
  - **Non-linear models**
  - Such as neural networks and deep networks

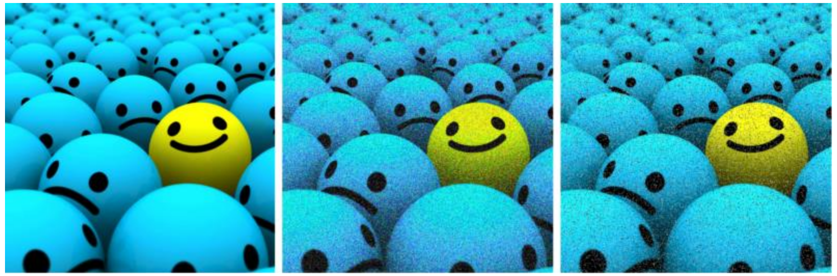
# Data Augmentation for Images

- Crop

- Translation

- Dilation / Scale

- Addition of noise



# Data Augmentation for Images (2)

- Flips

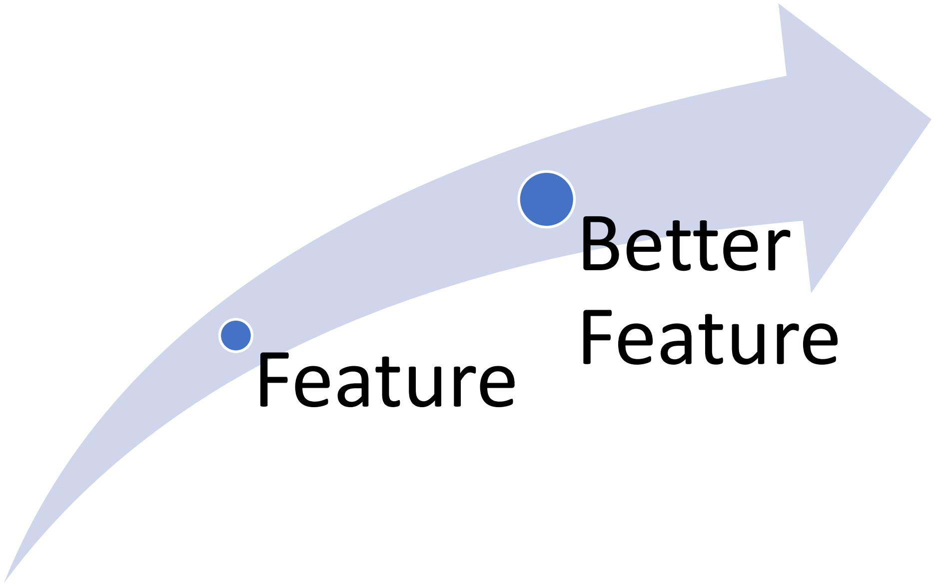
- Rotations

- Color manipulation

  *** Only if the feature used for classification is not color!

# Feature Engineering

- Feature engineering is the process of using **knowledge of the data** to **create** features that make machine learning algorithms work better.
- Feature engineering is an **art** as well as a science
- Data scientists often spend 70% of their time in the **data preparation** phase before modeling.



# Recognition: Classification