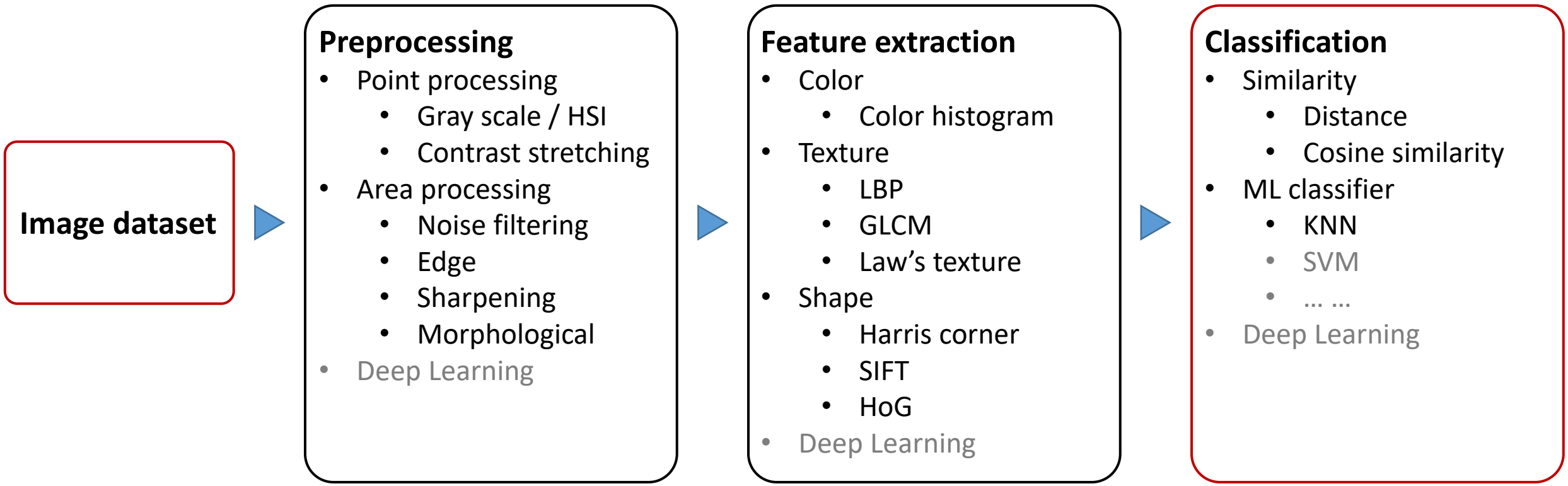
Image Matching System using Early Vision

Longbin Jin

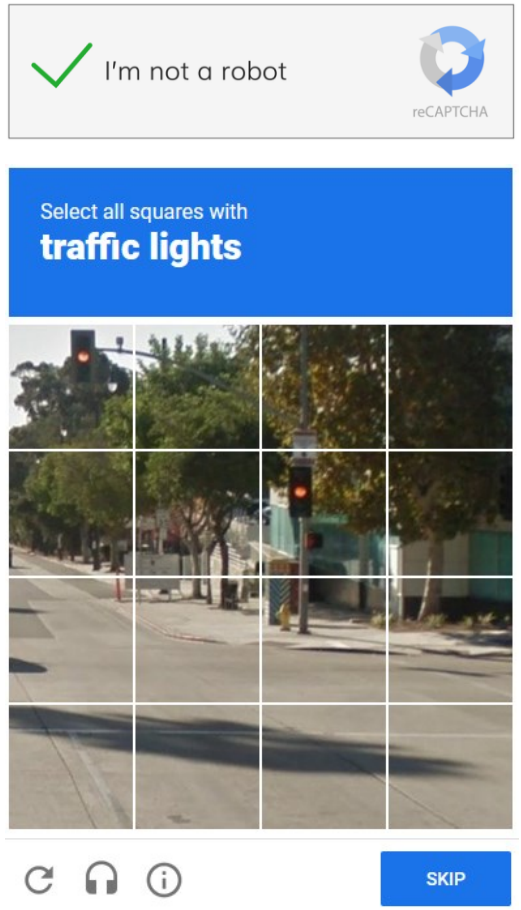
Artificial Inteligence
& Computer Vision
Laboratory

Image Matching System

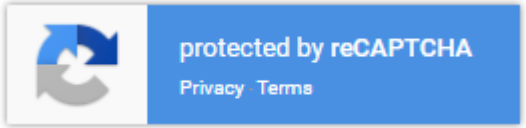


Dataset - reCAPTCHA

reCAPTCHA v1



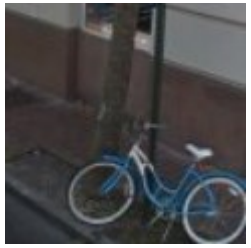
reCAPTCHA v2



reCAPTCHA v3

Dataset

- link
  - git clone https://github.com/folfcoder/recaptcha-dataset.git
- Classes (without mountain)


Bicycle (800)


Bridge (553)


Bus (1229)


Car (3578)


Chimney (56)


Crosswalk (1260)


Hydrant (972)


Motorcycle (101)


Palm (932)


Traffic Light (811)

Dataset

- Remove outlier
  - You don’t have to use all the data to fit your model

Bicycle?

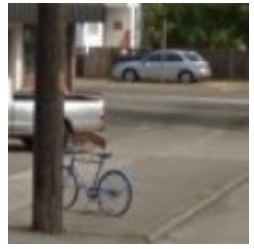
Car?

Traffic light?

- Multi-labels
  - Some images may have multi objects

Crosswalk + Car

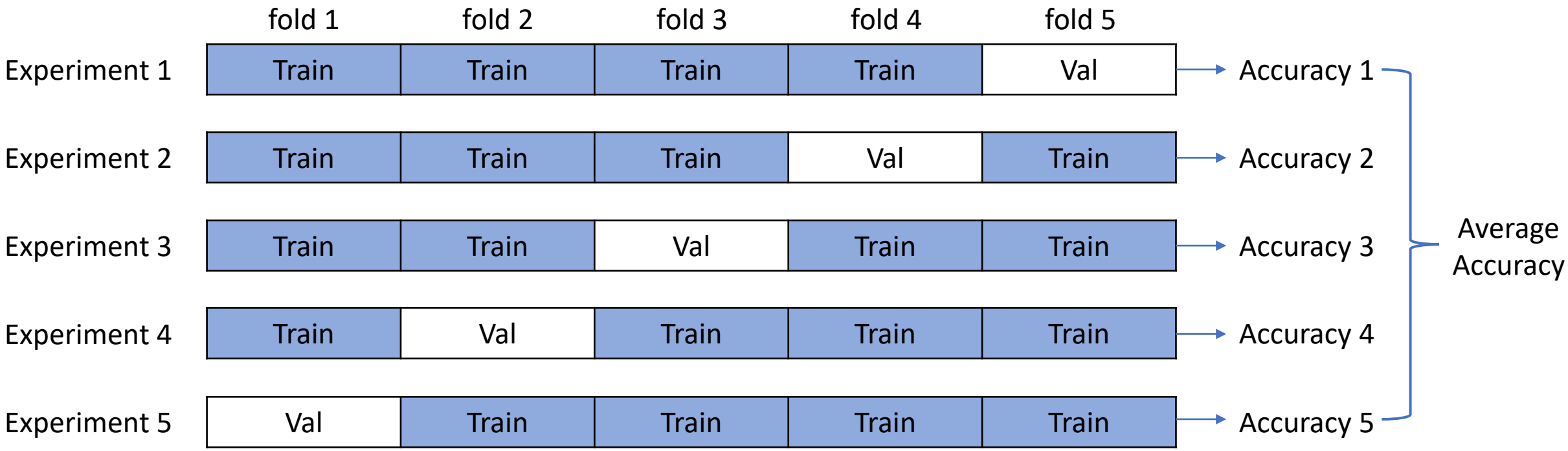
Crosswalk + Car+ Traffic light



Bicycle + Car

Dataset Split

- K-fold cross validation (k=5)



Dataset Split

- K-fold cross validation (k=5)

fold 1	fold 2	fold 3	fold 4	fold 5
Bicycle Bridge	Bus Car	Chimney Crosswalk	Hydrant Motorcycle	Palm Traffic Light

Bicycle (20%) Bridge (20%) Bus (20%) Car (20%) Chimney (20%) Crosswalk (20%) Hydrant (20%) Motorcycle (20%) Palm (20%) Traffic Light (20%)	Bicycle (20%) Bridge (20%) Bus (20%) Car (20%) Chimney (20%) Crosswalk (20%) Hydrant (20%) Motorcycle (20%) Palm (20%) Traffic Light (20%)	Bicycle (20%) Bridge (20%) Bus (20%) Car (20%) Chimney (20%) Crosswalk (20%) Hydrant (20%) Motorcycle (20%) Palm (20%) Traffic Light (20%)	Bicycle (20%) Bridge (20%) Bus (20%) Car (20%) Chimney (20%) Crosswalk (20%) Hydrant (20%) Motorcycle (20%) Palm (20%) Traffic Light (20%)	Bicycle (20%) Bridge (20%) Bus (20%) Car (20%) Chimney (20%) Crosswalk (20%) Hydrant (20%) Motorcycle (20%) Palm (20%) Traffic Light (20%)

Test Data

- Delayed delivery 
  - Test data will be made available at the end of the semester.

- Test data
  - 10 images per class
  - Total 100 images

Evaluation

- Confusion matrix

Total = TP + TN + FP + FN		Predicted Condition: Positive	Predicted Condition: Negative
Actual Condition	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

- TP: You predicted positive and it's true.
- TN: You predicted negative and it's true.
- FP: You predicted positive and it's false.
- FN: You predicted negative and it's false.

Evaluation

- Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision

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

- Recall (Sensitivity)

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

- F1 score (Harmonic mean of precision and recall)

$$\frac{1}{F_1} = (\frac{1}{precision} + \frac{1}{recall})/2$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

Evaluation

- Confusion matrix

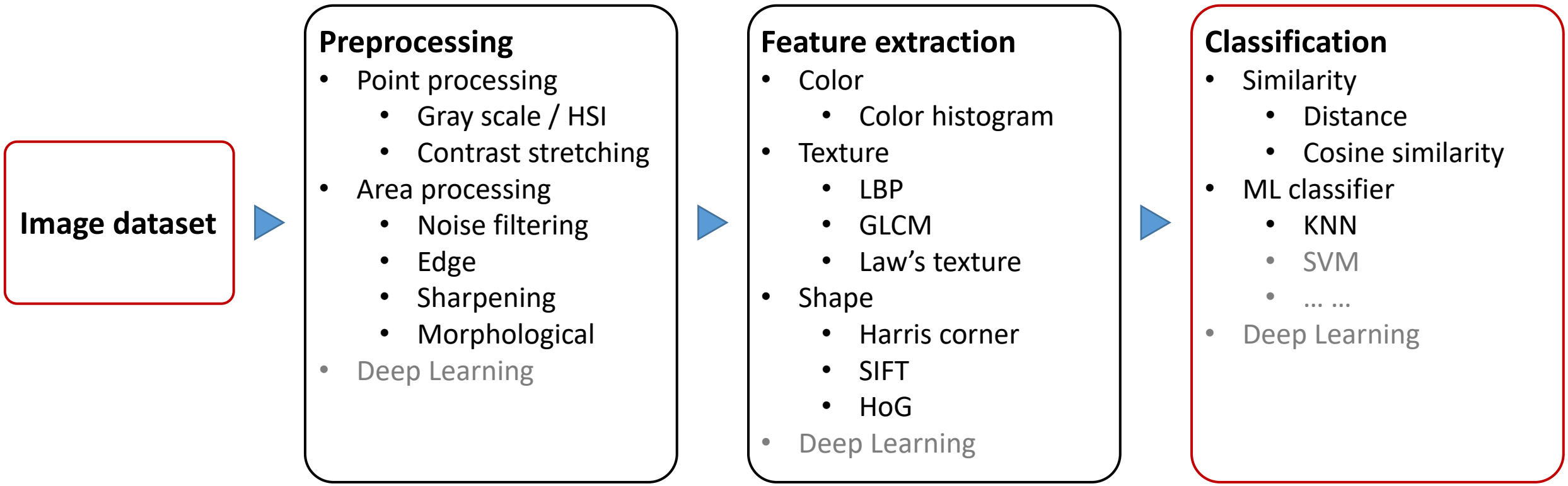
Total = TP + TN + FP + FN		Predicted Condition: Positive	Predicted Condition: Negative	
Actual Condition	Positive	True Positive (TP)	False Negative (FN)	Recall $\frac{TP}{TP + FN}$
	Negative	False Positive (FP)	True Negative (TN)	
		Precision $\frac{TP}{TP + FP}$		Accuracy $\frac{TP + TN}{TP + TN + FP + FN}$

Evaluation

- Confusion matrix for Multi-class

		Predicted Condition			
		Bicycle	Bridge	…	Traffic Light
Actual Condition	Bicycle	578	22	…	42
	Bridge	34	363	…	93
	…	…	…	…	…
	Traffic Light	50	87	…	447

Image Matching System



Preprocessing

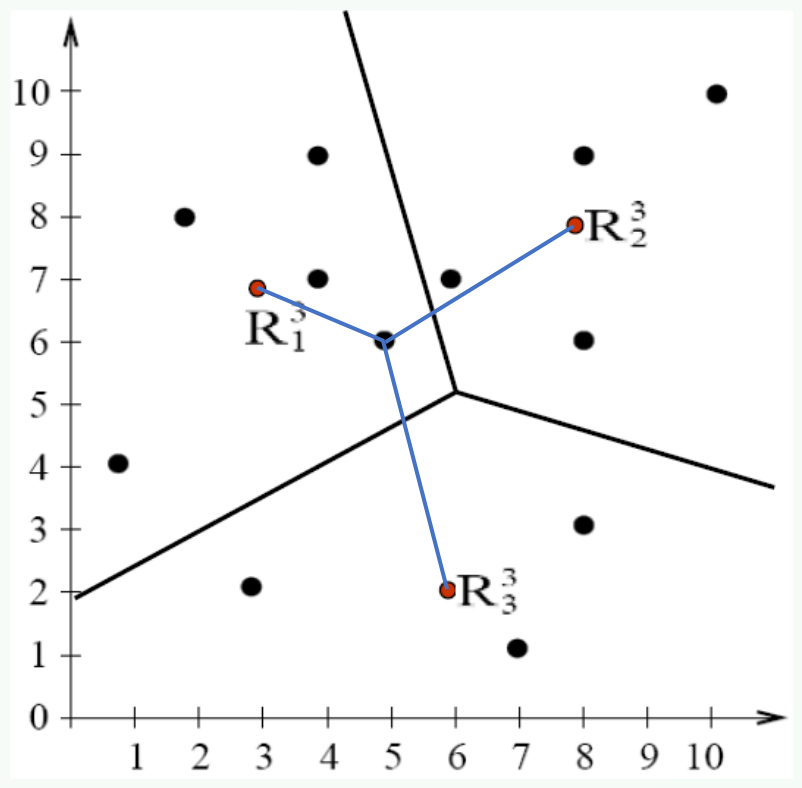
- Point processing
  - HIS
  - gray scale
  - contrast stretching (histogram equalization)
- Area processing
  - noise filtering (mean / gaussian / median)
  - edge (previtt / sobel / canny)
  - sharpening (high-boost filtering)
  - morphological operator (open / close)

Feature Extraction

- Color
  - color / gray scale histogram (3x256-d / 256-d or n-d)
- Texture
  - LBP histogram (256-d or n-d)
  - GLCM (6-d)
    - maximum probability, moments, contrast, homogeneity, entropy, correlation
  - Law's texture (9-d)
    - E5E5, S5S5, R5R5, E5L5, S5L5, R5L5, S5E5, R5E5, R5S5
- Shape
  - Harris
  - SIFT (128-d)
  - HOG

Dimensionality Reduction

- K-means for dimension reduction
  - n-dim -> k-dim
  1. Apply the K-means algorithm
  2. Calculate the distance of each data point from each cluster center
  3. We got k-dim features
     - Distance of point from cluster 1
     - Distance of point from cluster 2
     - …
     - Distance of point from cluster k



Similarity

- Distance

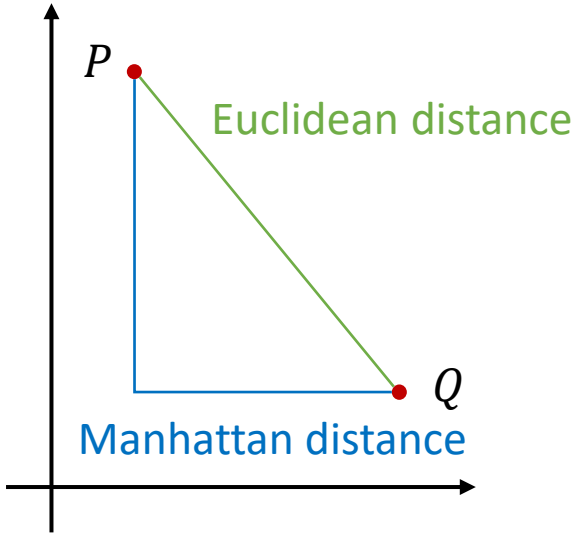
$$P = [p_1, p_2, \dots, p_n], \qquad Q = [q_1, q_2, \dots, q_n]$$

  - Manhattan distance (L1 norm)

$$d(P, Q) = \sum_{i=1}^{n} (|p_i - q_i|) = |p_1 - q_1| + \cdots + |p_n - q_n|$$

  - Euclidean distance (L2 norm)

$$d(P, Q) = (\sum_{i=1}^{n} (|p_i - q_i|^2))^{1/2} = \sqrt{|p_1 - q_1|^2 + \cdots + |p_n - q_n|^2}$$



Similarity

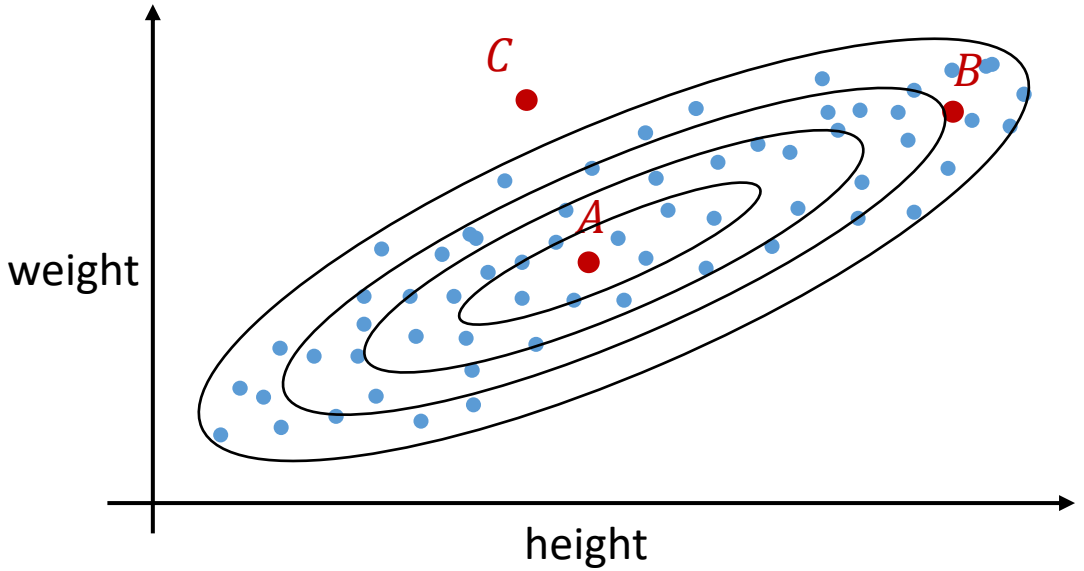
- Distance

$$P = [p_1, p_2, \dots, p_n], \qquad Q = [q_1, q_2, \dots, q_n]$$

  - Mahalanobis Distance

$$d(P, Q) = \sqrt{(P - Q)\Sigma^{-1}(P - Q)^T}$$

$$\Sigma = \frac{1}{n} X^T X$$

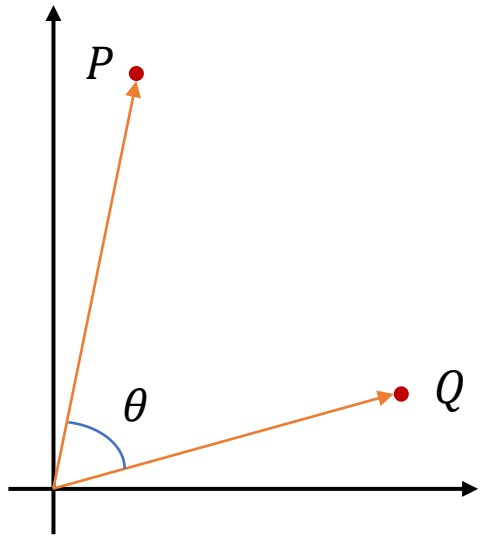


Similarity

- Cosine Similarity

$$P = [p_1, p_2, \dots, p_n], \qquad Q = [q_1, q_2, \dots, q_n]$$

$$sim(P, Q) = \cos(\theta) = \frac{P \cdot Q}{\|P\|\|Q\|}$$



Similarity

- Extract feature from test image
- Find the most similar feature from train data
    1. Bus
    2. Car
    3. Car
    4. Bus
    5. Bus
    6. Bus
- Top-1 class: Bus, Top-3 class: Car, Top-6: Bus

Test image

K-Nearest Neighbors (KNN)

- Lazy model
  - Non-parametric supervised learning
  - it is helpful to choose *k* to be an odd number
- Find the optimal parameter k using train data and validation data

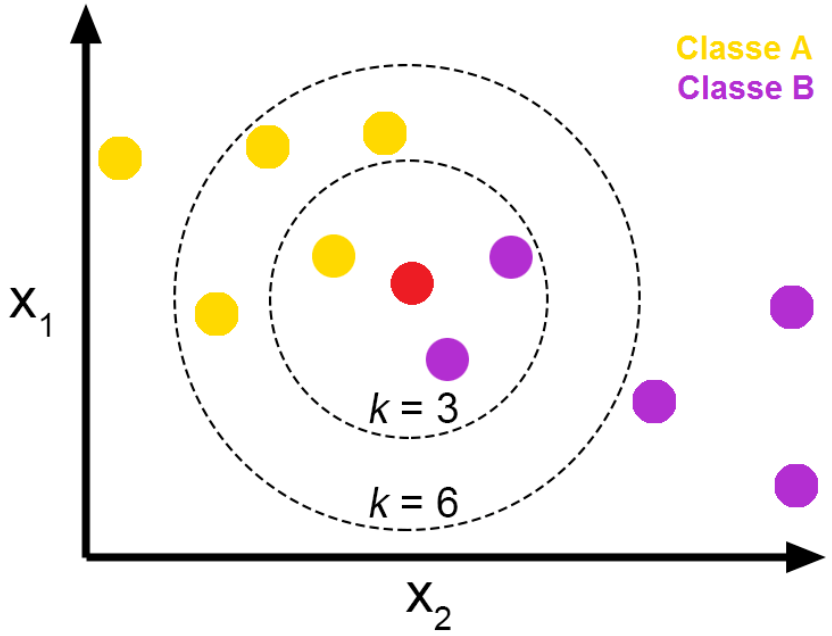


Image Matching System