AI VIETNAM
All-in-One Course

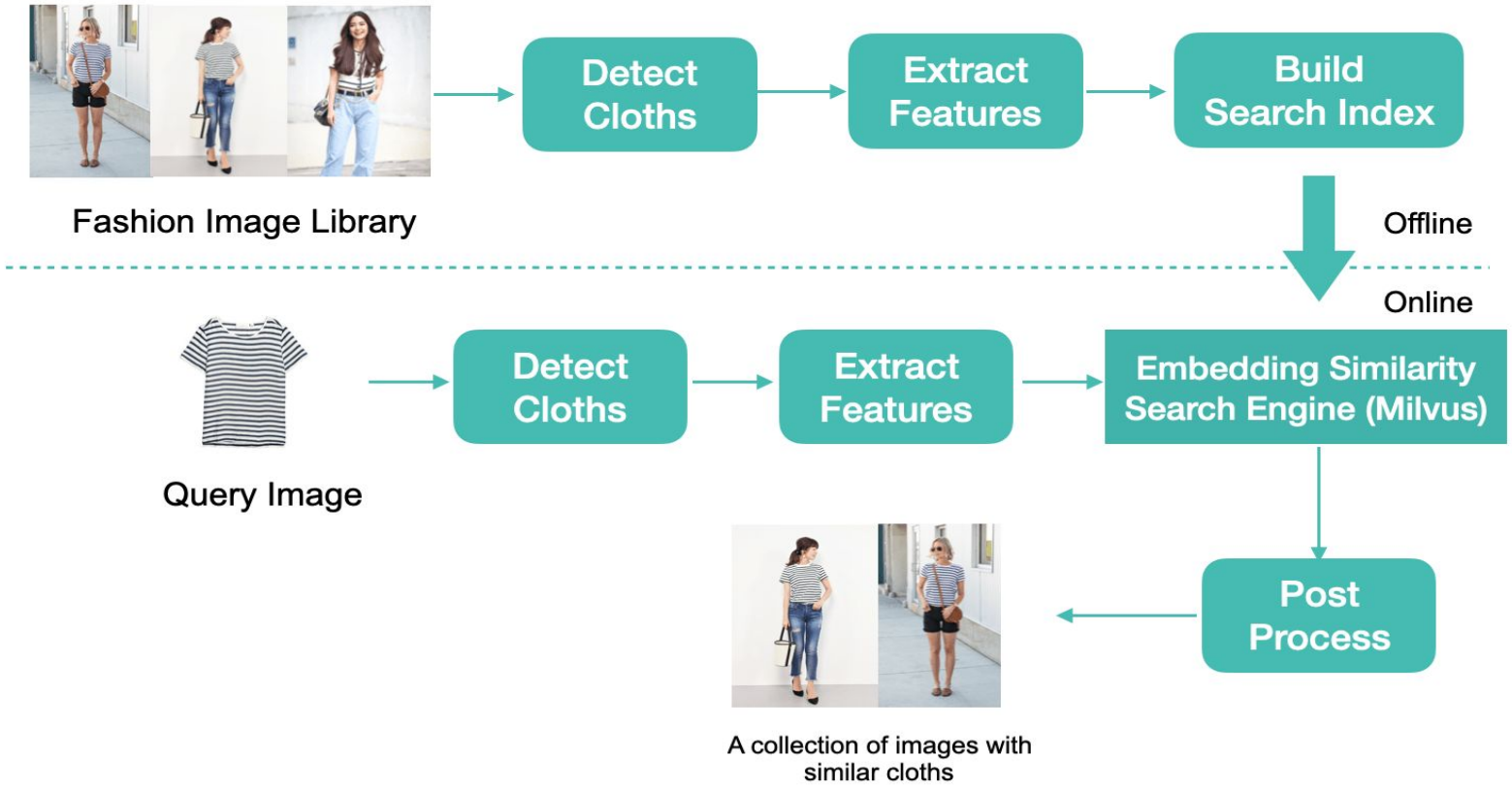
Project: Image Retrieval
(TA Session)

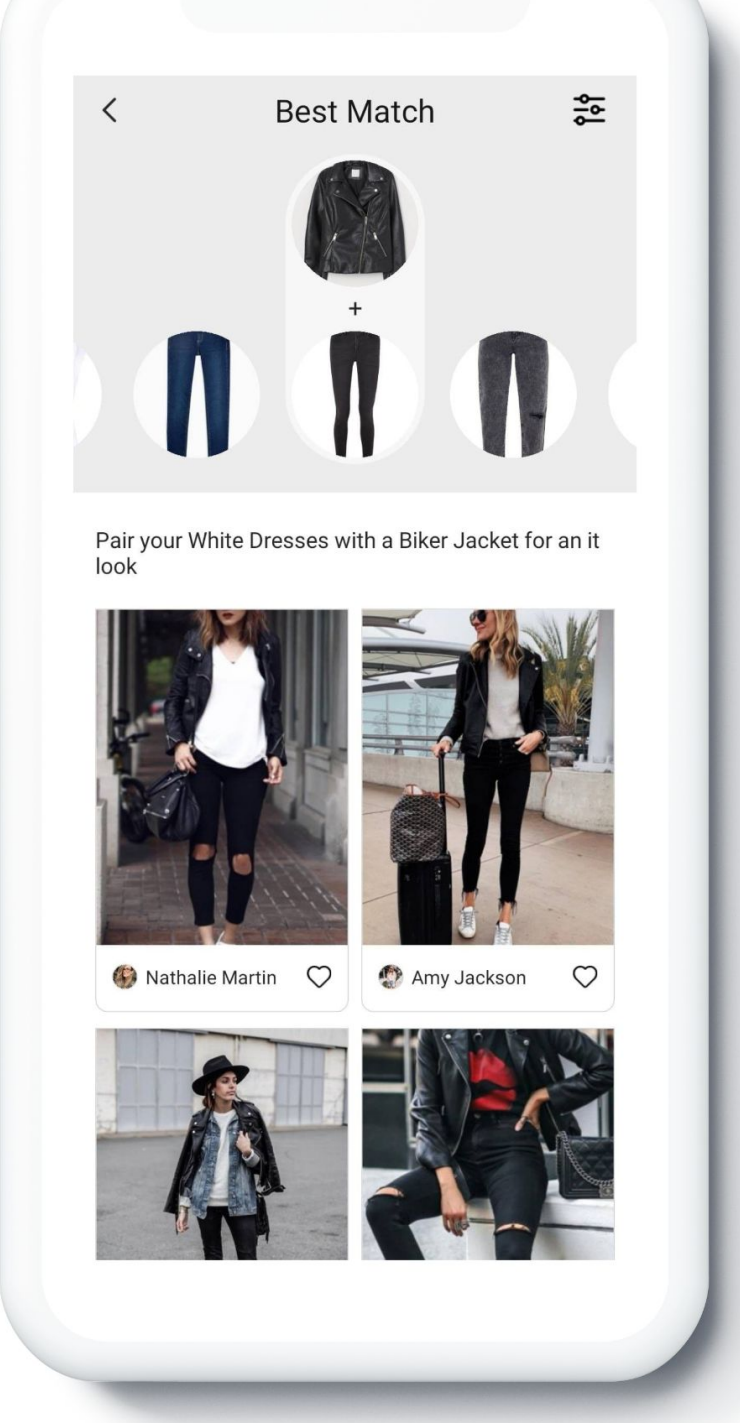
Nguyễn Thọ Anh Khoa
Ph.D. Student

Year 2024

Objective

- ✔ Introduce image retrieval.
- ✔ Discuss similarity measurement techniques.
- ✔ Outline data preparation for image retrieval.
- ✔ Demonstrate a basic image retrieval system.
- ✔ Explore advanced retrieval with deep learning models.
- ✔ Implement image retrieval using vector databases.
- ✔ (Optional) Overview of web crawling for data acquisition.





Outline

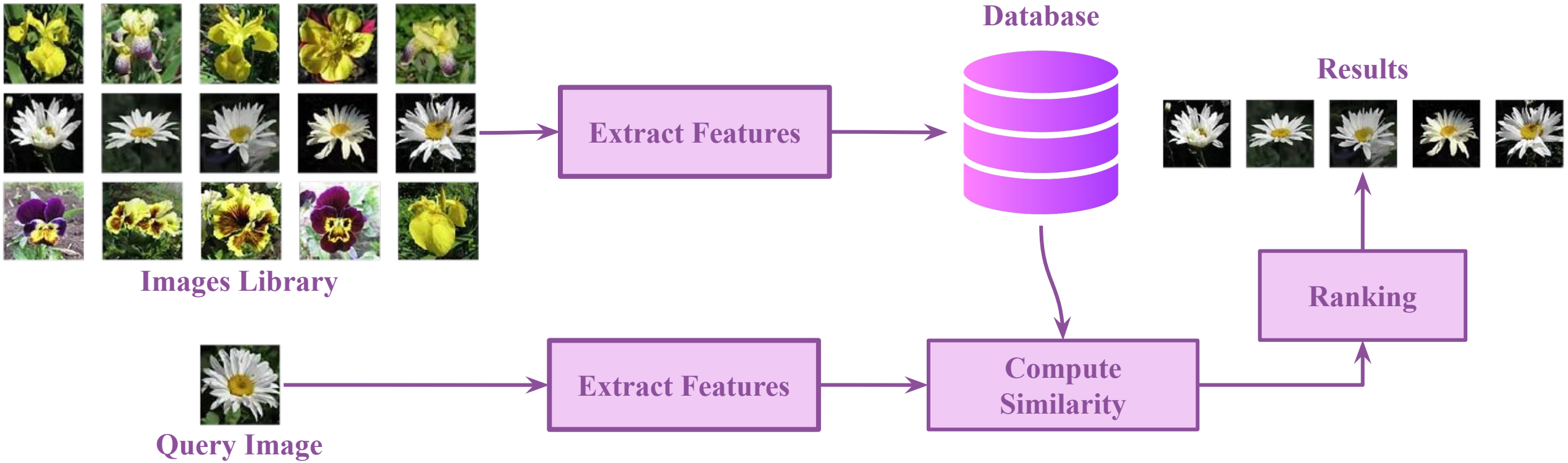
- Introduction to Image Retrieval
- Similarity Measurement Techniques in Image Retrieval
- Data Preparation for Image Retrieval
- Basic Image Retrieval System
- Advanced Image Retrieval Using Pre-trained Deep Learning Model
- Implementing Image Retrieval with Vector Database
- Optional: Data Acquisition through Web Crawling

AI VIETNAM
All-in-One Course

Introduction to Image Retrieval

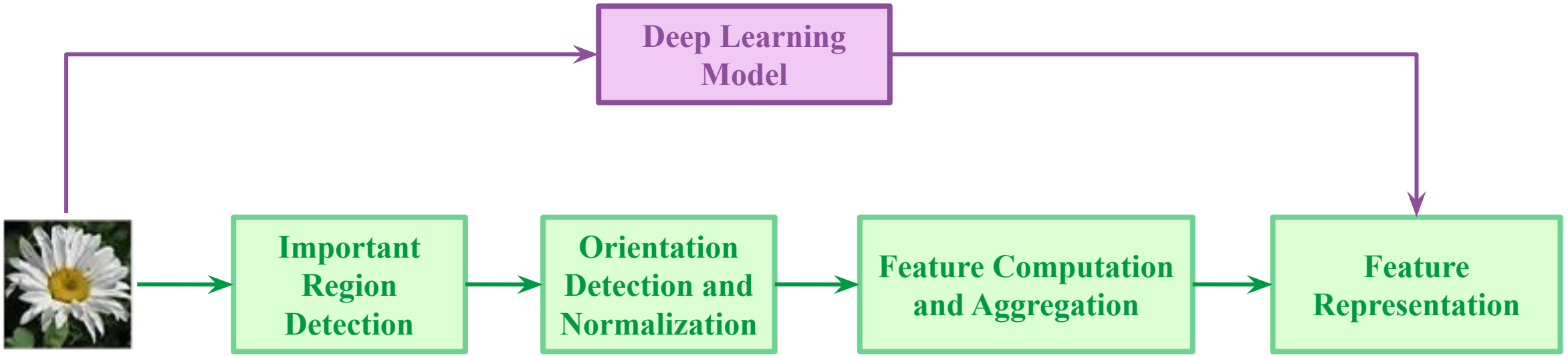
Introduction to Image Retrieval

Definition: The **content based image retrieval** aims to find the **similar images** from a **large scale dataset** against a **query image**. Generally, the similarity between the representative features of the query image and dataset images is used to **rank the images for retrieval**. In early days, various hand designed feature descriptors have been investigated based on the visual cues such as color, texture, shape, etc. that represent the images. However, the deep learning has emerged as a dominating alternative of hand-designed feature engineering from a decade. It learns the features automatically from the data

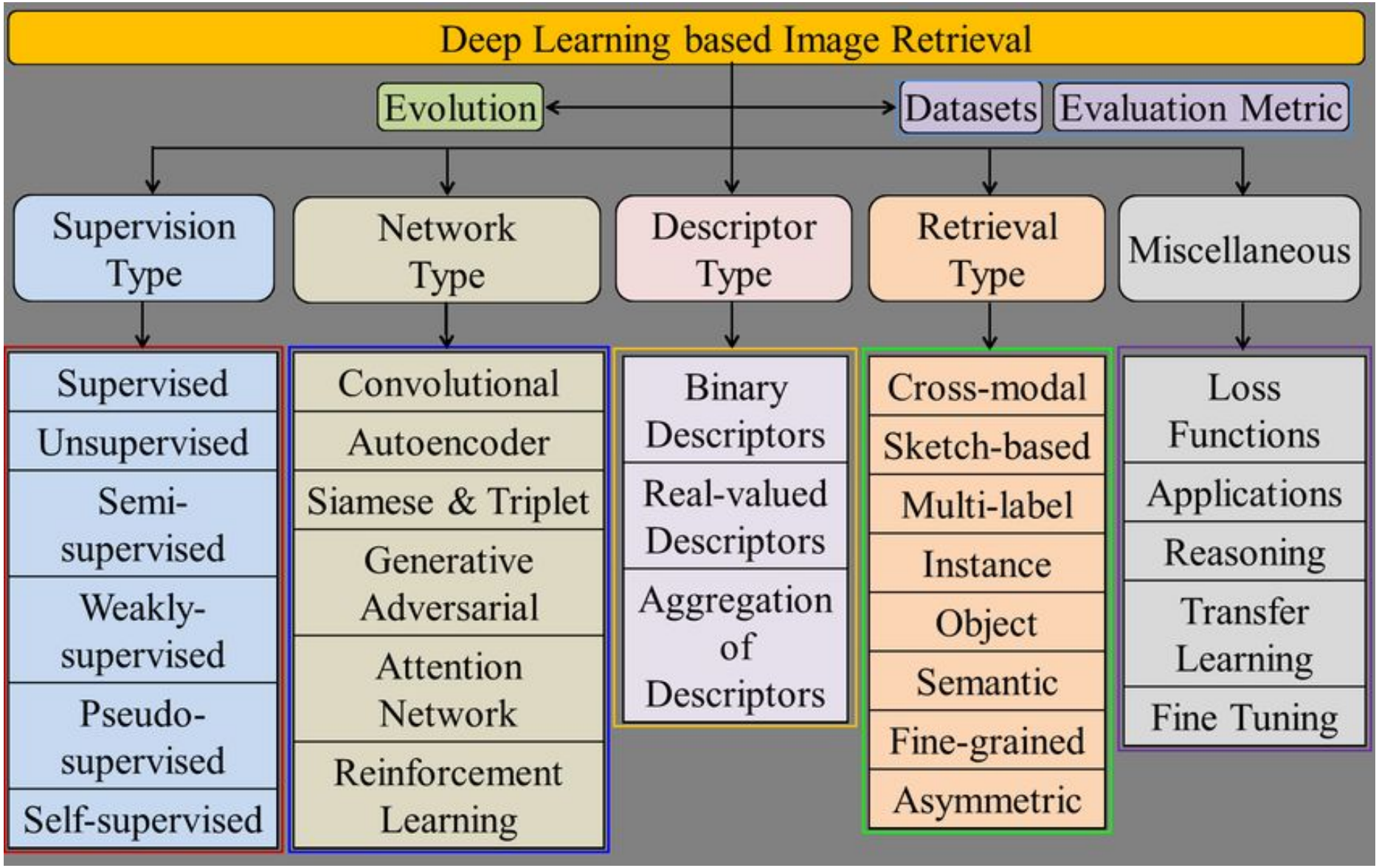


Introduction to Image Retrieval

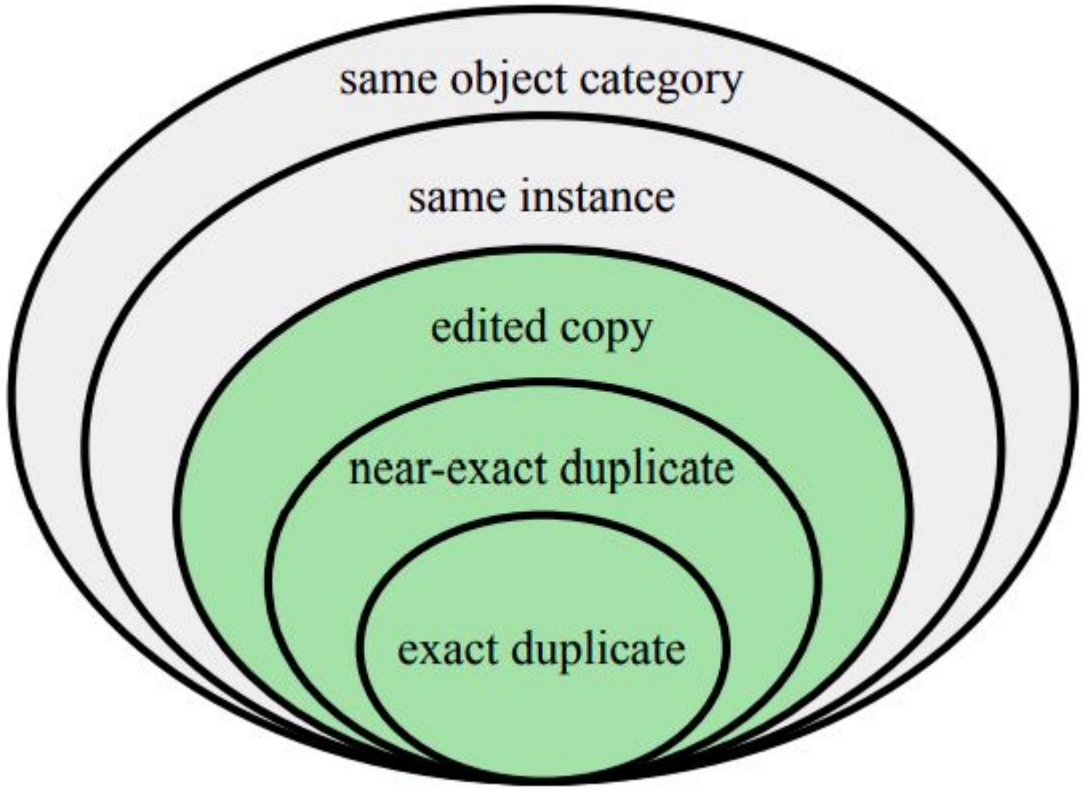
Definition: The **content based image retrieval** aims to find the **similar images** from a **large scale dataset** against a **query image**. Generally, the similarity between the representative features of the query image and dataset images is used to **rank the images for retrieval**. In early days, various hand designed feature descriptors have been investigated based on the visual cues such as color, texture, shape, etc. that represent the images. However, the deep learning has emerged as a dominating alternative of hand-designed feature engineering from a decade. It learns the features automatically from the data

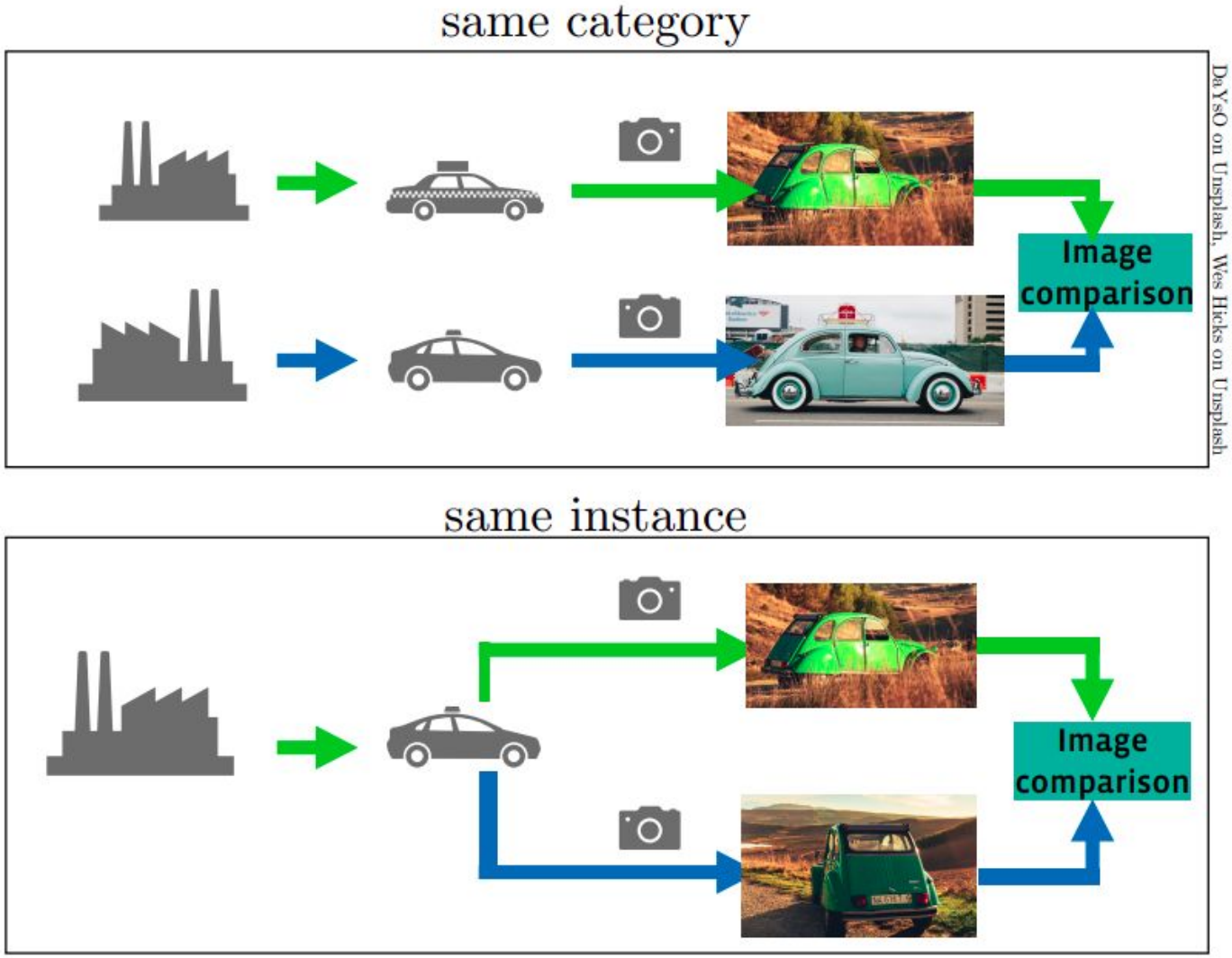


Introduction to Image Retrieval

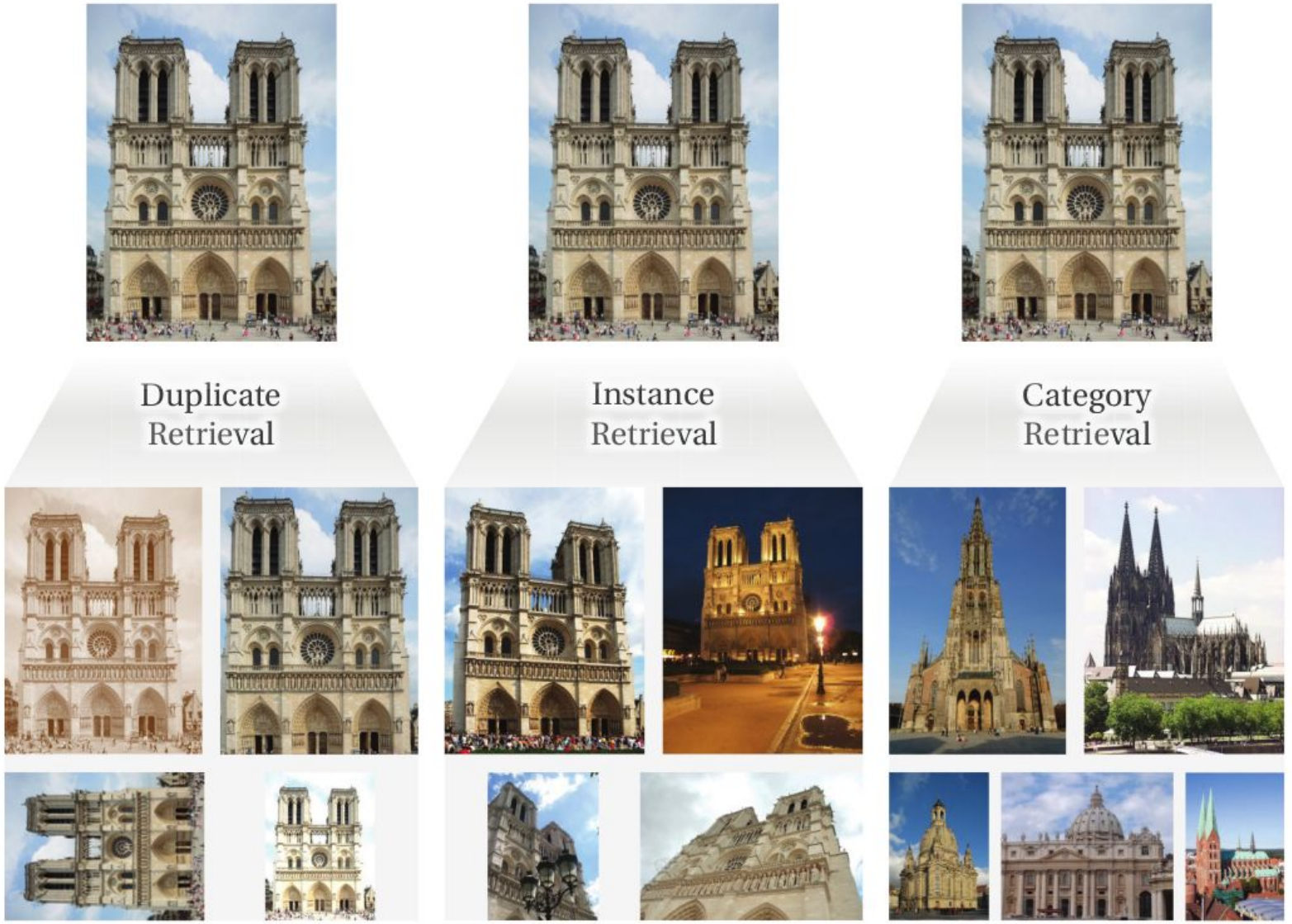


Introduction to Image Retrieval





Introduction to Image Retrieval



AI VIETNAM
All-in-One Course

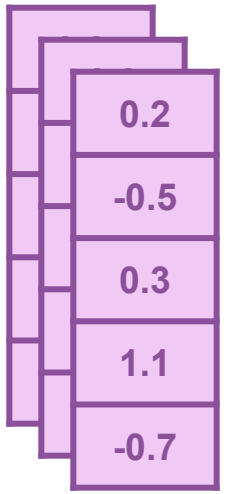
Introduction to Image Retrieval

Project Pipeline

Images Library

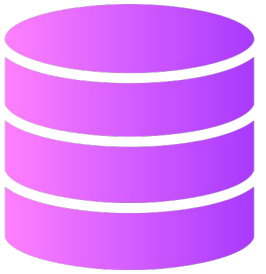
CLIP Model (Pretrained Model)

Feature Vectors



Indexing

(Chroma) Vector Database



Query Image

CLIP Model (Pretrained Model)

Feature Vector

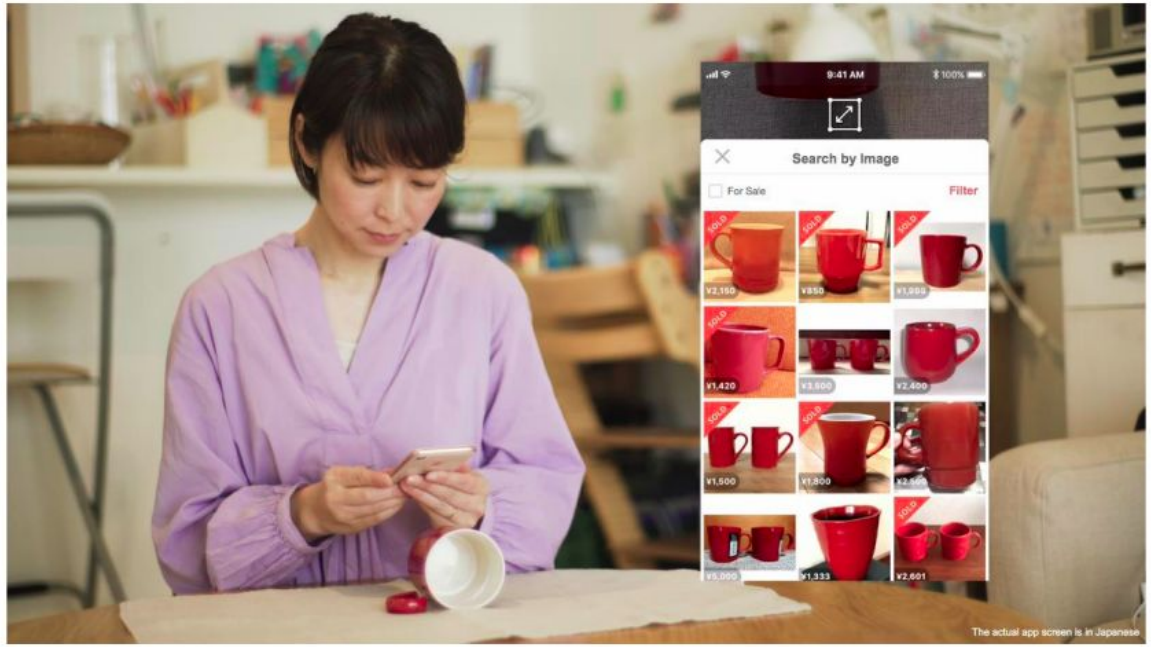
Similarity Computation

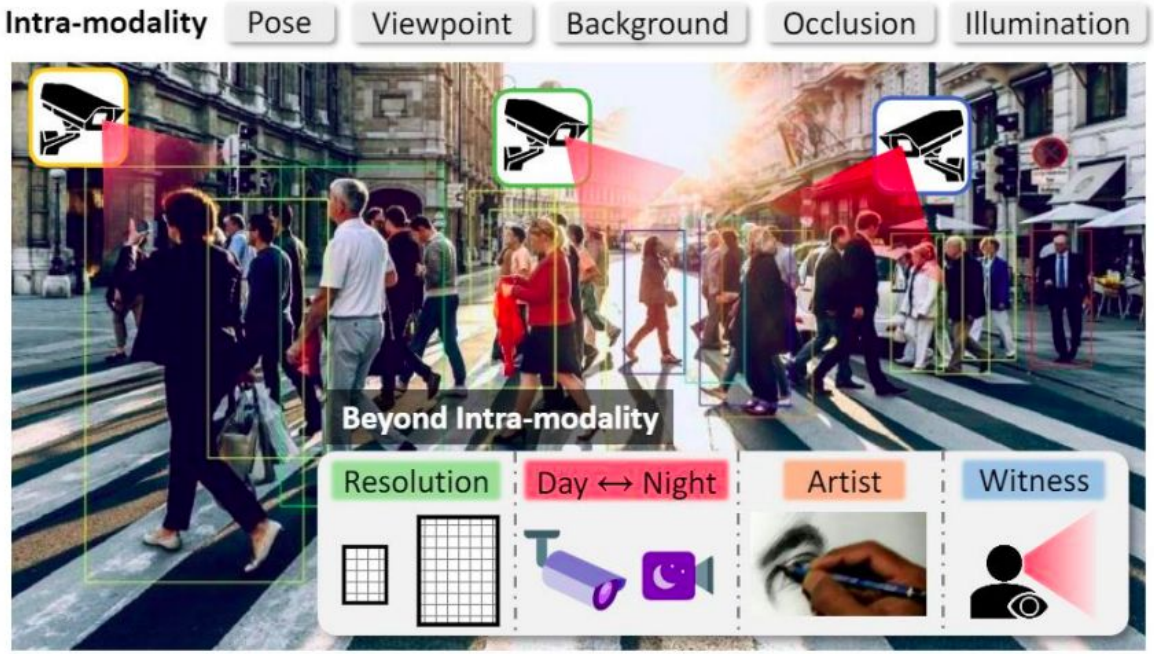
Ranking

Results

Introduction to Image Retrieval

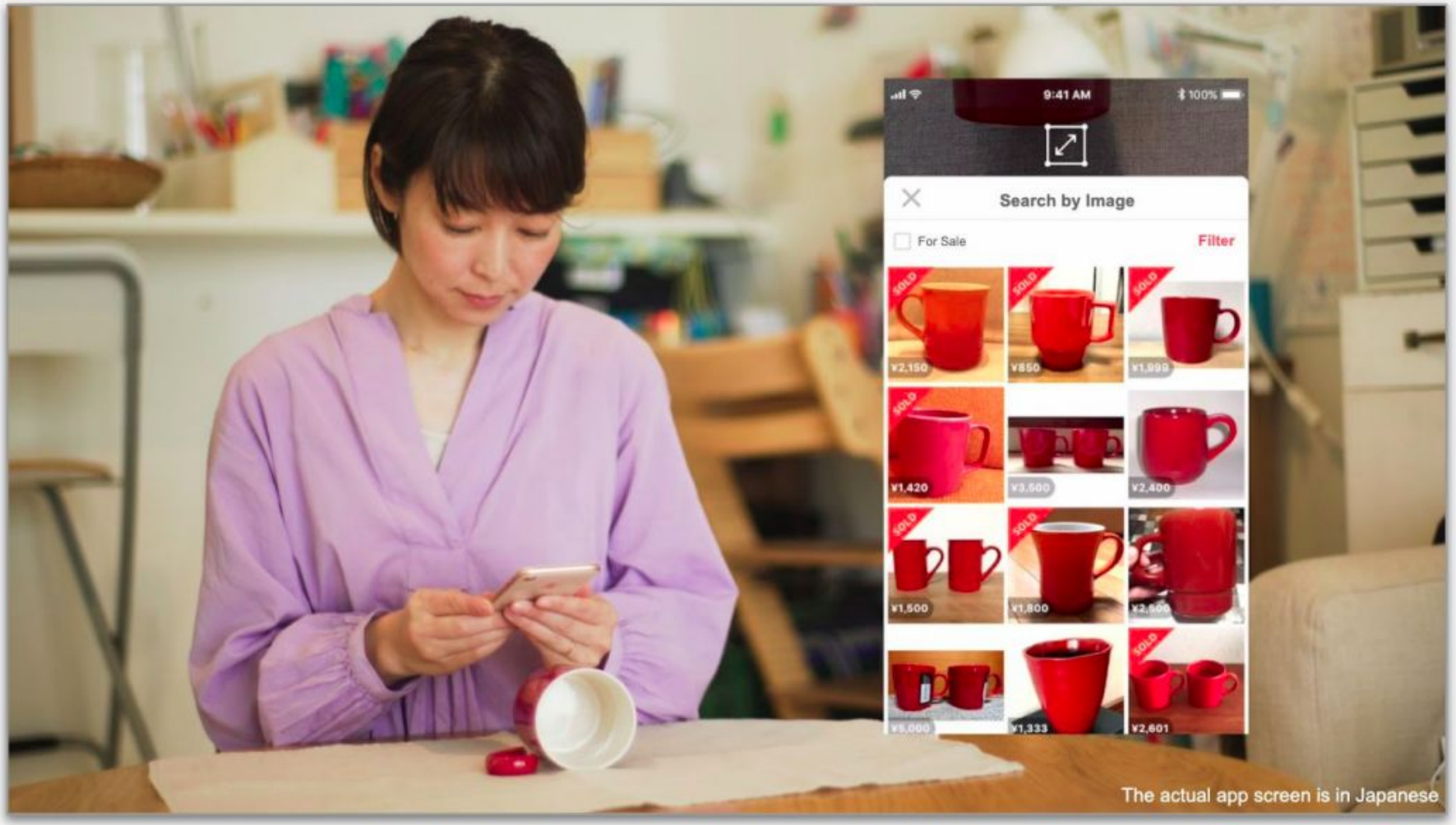
Practical and **large-scale** visual search system for real-world applications
- Discovering visually similar items in online marketplace apps
- Re-identification for pedestrians in a security scenario.





Introduction to Image Retrieval

- Amazon
- eBay
- Asos
- Alibaba
- Taobao
- AliExpress
- Mercari
- etc...



The visual search feature on the Mercari Japan app [Promotional Video]

Introduction to Image Retrieval

How do you query when you look for clothes like this?

- “Plaid” and “Shirt”?
- “Plaid” and “Blouse”?
- “Checks” and “Blouse”?
- “Checks” and “Blouse” and “Frill”?
- “Gingham Plaid” and “Blouse” and “Frill”?
- “Gingham Plaid” and “Blouse” and “Frill” and “Collar”?
- ...

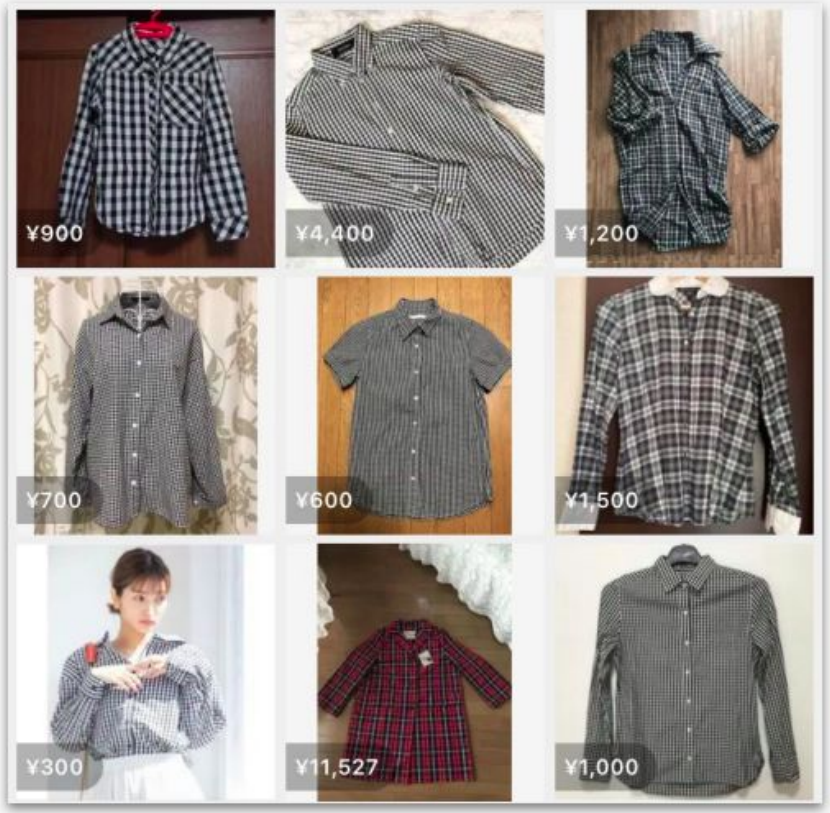
Introduction to Image Retrieval

Visual Search

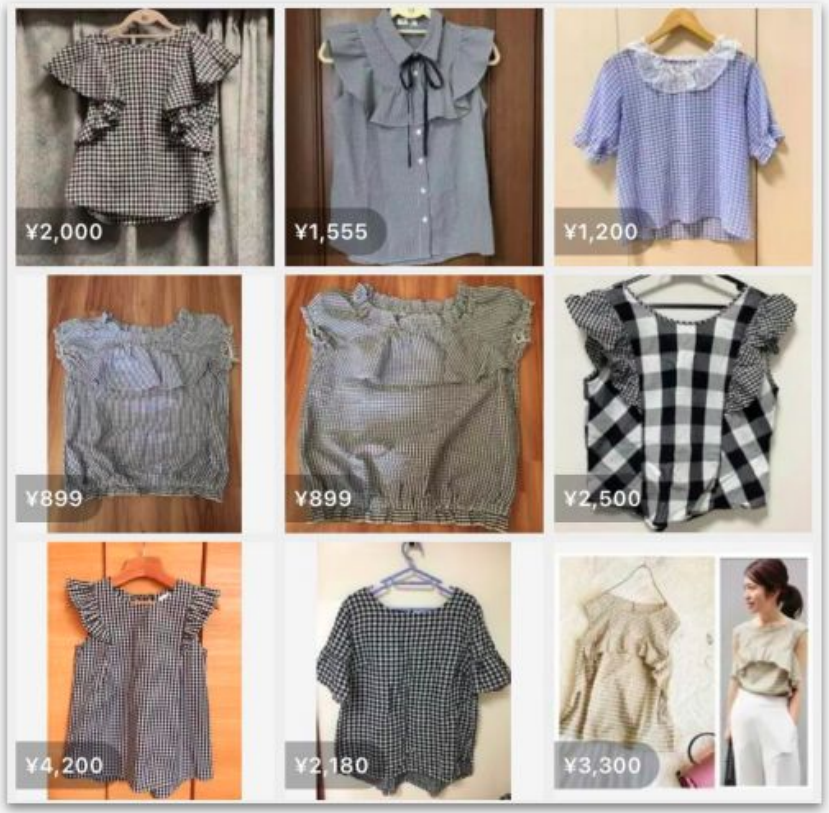
Query:

(c) 2020 Mercari, Inc.

Text-based Search



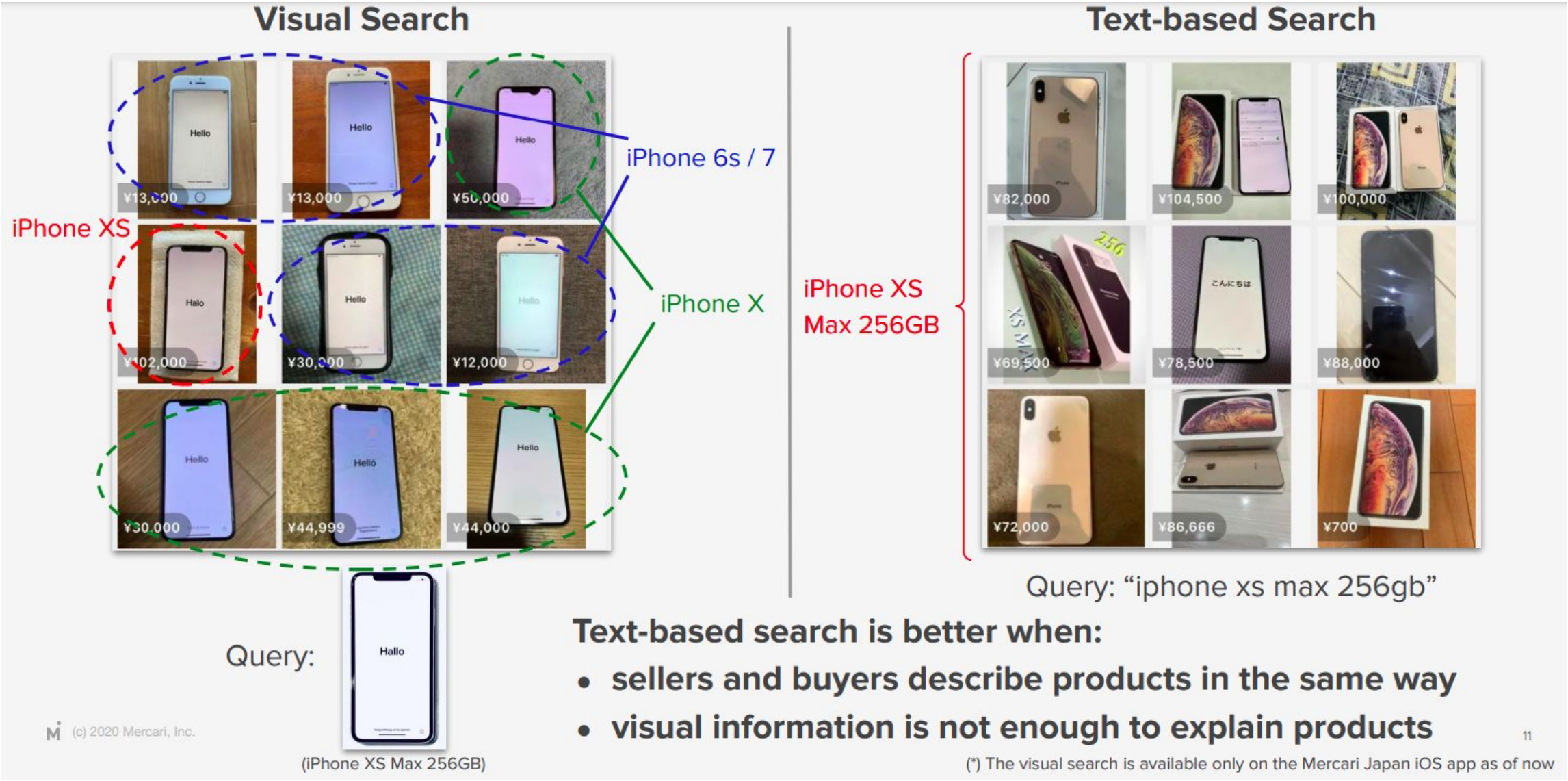
Query: “black white plaid shirt”

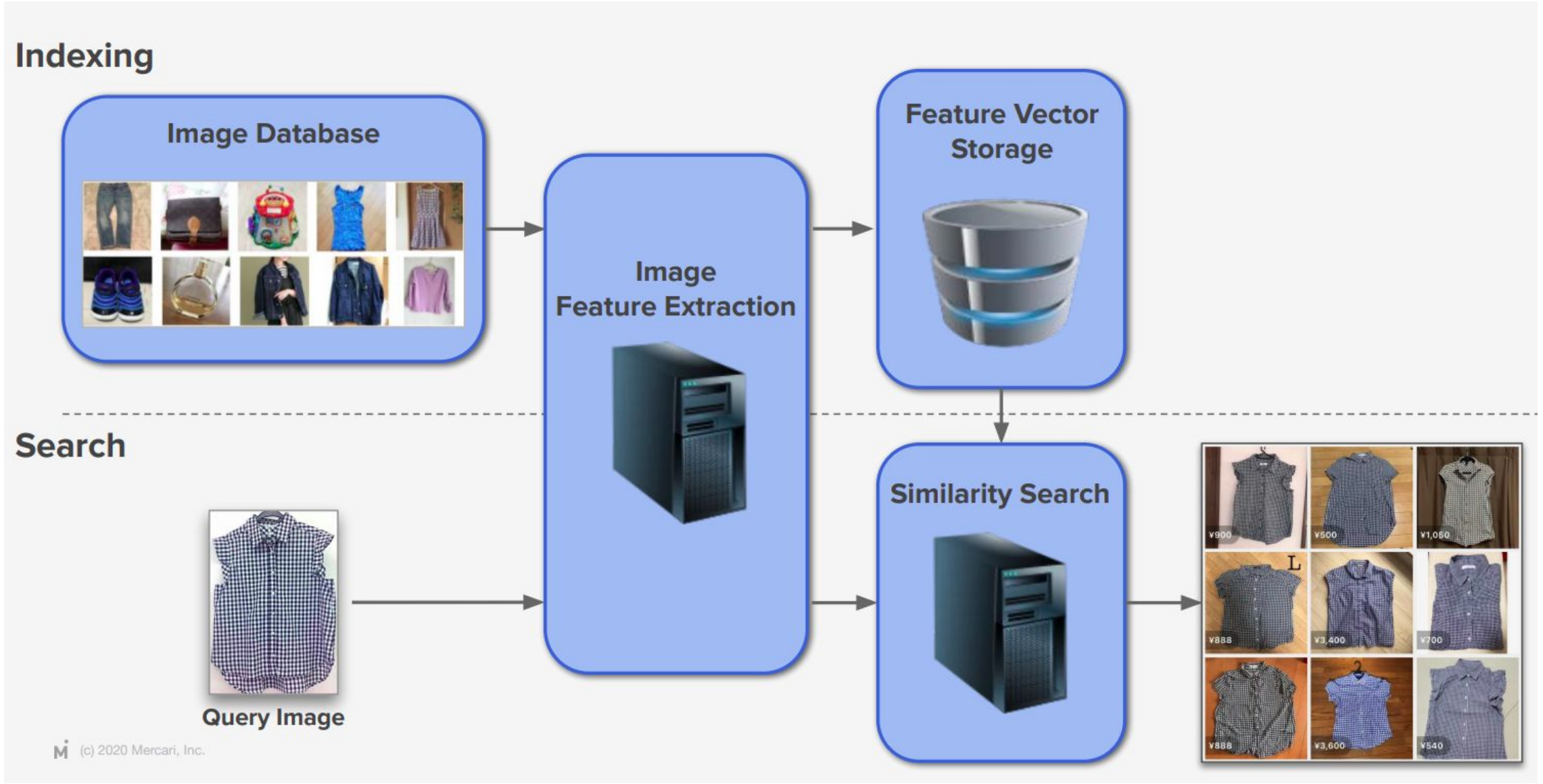


Query: “Gingham Plaid Frill Blouse”

(*) The visual search is available only on the Mercari Japan iOS app as of now

Introduction to Image Retrieval




Indexing
Image Database
Image
Feature Extraction
Feature Vector
Storage
Search
Similarity Search
Query Image
(c) 2020 Mercari, Inc.

ANN search should be used practically

Indexing

Image Database

Image Feature Extraction

Search Index Builder

- Faiss
- nmslib
- Annoy
- etc...

Search Index Storage

Search

Query Image

Similarity Search

(c) 2020 Mercari, Inc.

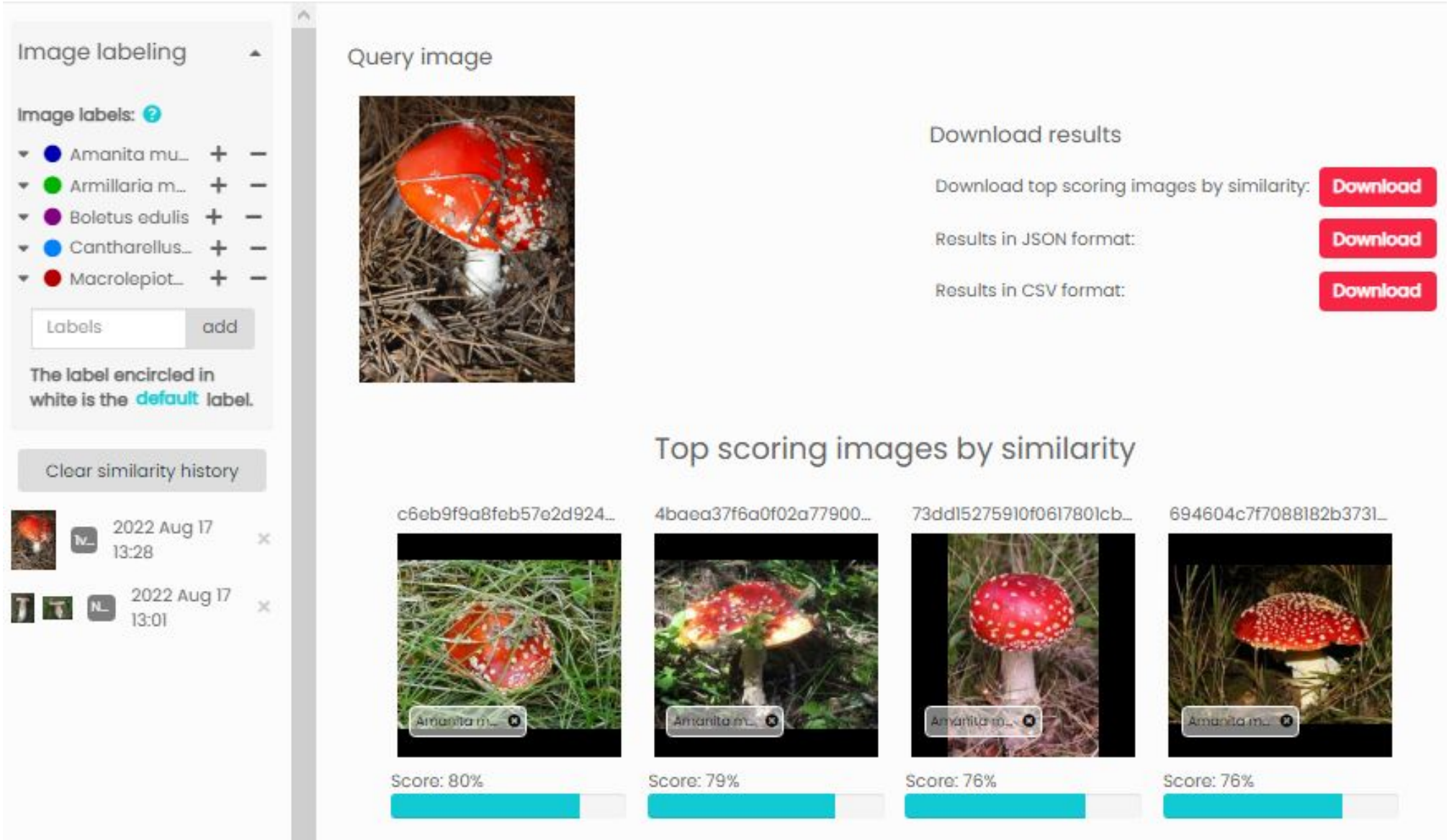


Image labeling
Image labels:
Amanita mu...
Armillaria m...
Boletus edulis
Cantharellus...
Macrolepiot...
Labels
add
The label encircled in white is the default label.
Clear similarity history
2022 Aug 17 13:28
2022 Aug 17 13:01
Query image
Download results
Download top scoring images by similarity:
Download
Results in JSON format:
Download
Results in CSV format:
Download
Top scoring images by similarity
c6eb9f9a8feb57e2d924...
4baea37f6a0f02a77900...
73dd15275910f0617801cb...
694604c7f7088182b3731...
Score: 80%
Score: 79%
Score: 76%
Score: 76%

AI VIETNAM
All-in-One Course

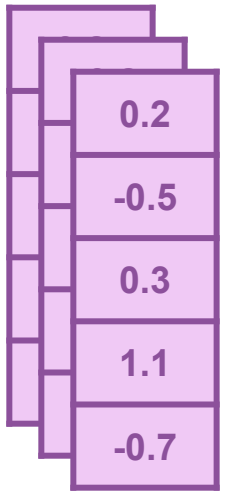
Introduction to Image Retrieval

Summary

Images Library

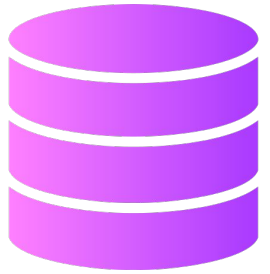
CLIP Model (Pretrained Model)

Feature Vectors



Indexing

(Chroma) Vector Database



Query Image

CLIP Model (Pretrained Model)

Feature Vector

Similarity Computation

Ranking

Results

Introduction to Image Retrieval

Content based image retrieval được định nghĩa là gì?

A. Tạo hình ảnh mới từ các hình ảnh hiện có.
B. Tìm kiếm và lấy hình ảnh từ cơ sở dữ liệu dựa trên nội dung của chúng.
C. Chỉnh sửa màu sắc và độ sáng của hình ảnh.
D. Chuyển đổi hình ảnh sang các định dạng khác.

Similarity Measurement Techniques in Image Retrieval

Similarity Measurement Techniques in Image Retrieval

Distance Metrics and Similarity

Distance Metrics:

- **Definition**: Mathematical measures used to calculate the distance (or similarity) between two elements in a vector space.
- **Purpose**: Evaluate how far apart or how similar two elements are.

Context of Embeddings:

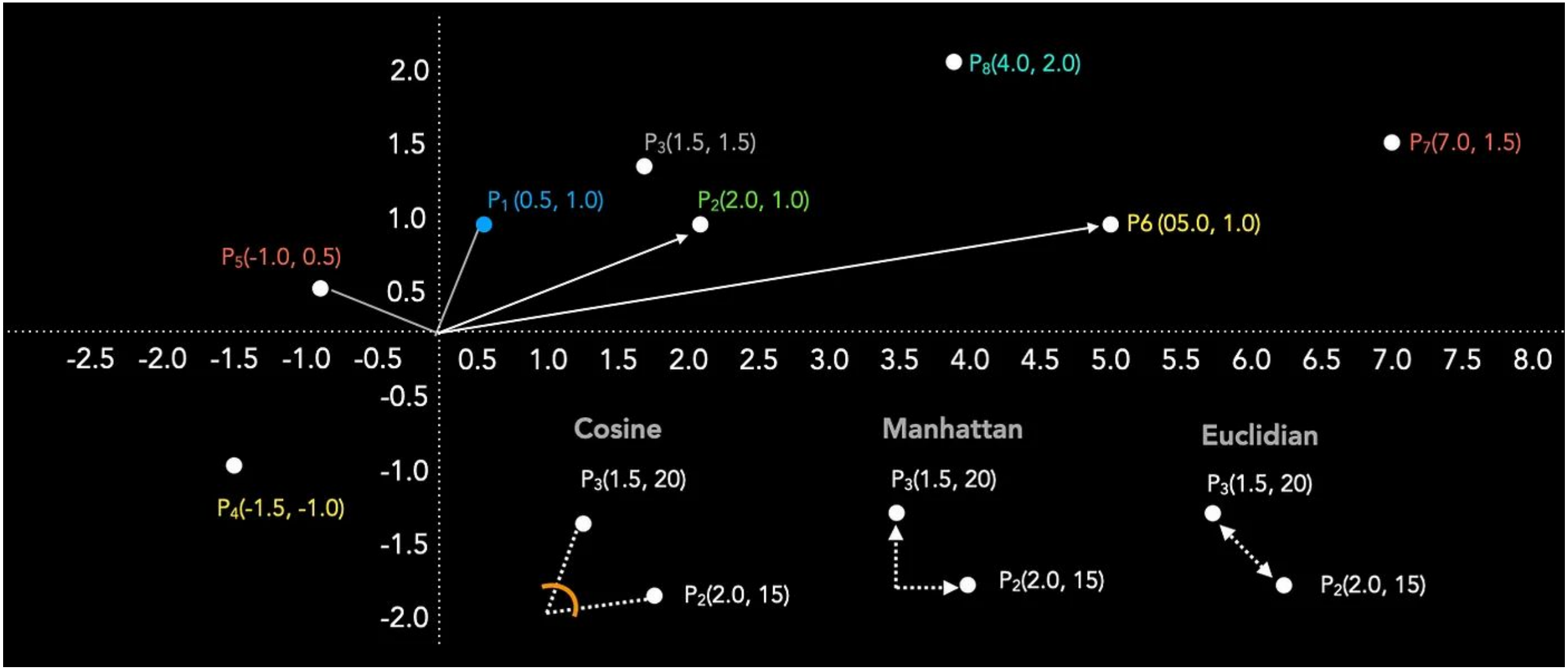
- **Definition**: Embeddings are vector representations of data points.
- **Role of Distance Metrics**: In this context, distance metrics are used to assess the similarity between embeddings.

Similarity Query Search:

- **Function**: Retrieves embeddings that are similar to a given input.
- **Based On**: Uses distance metrics to determine which embeddings are close to the input in the vector space.

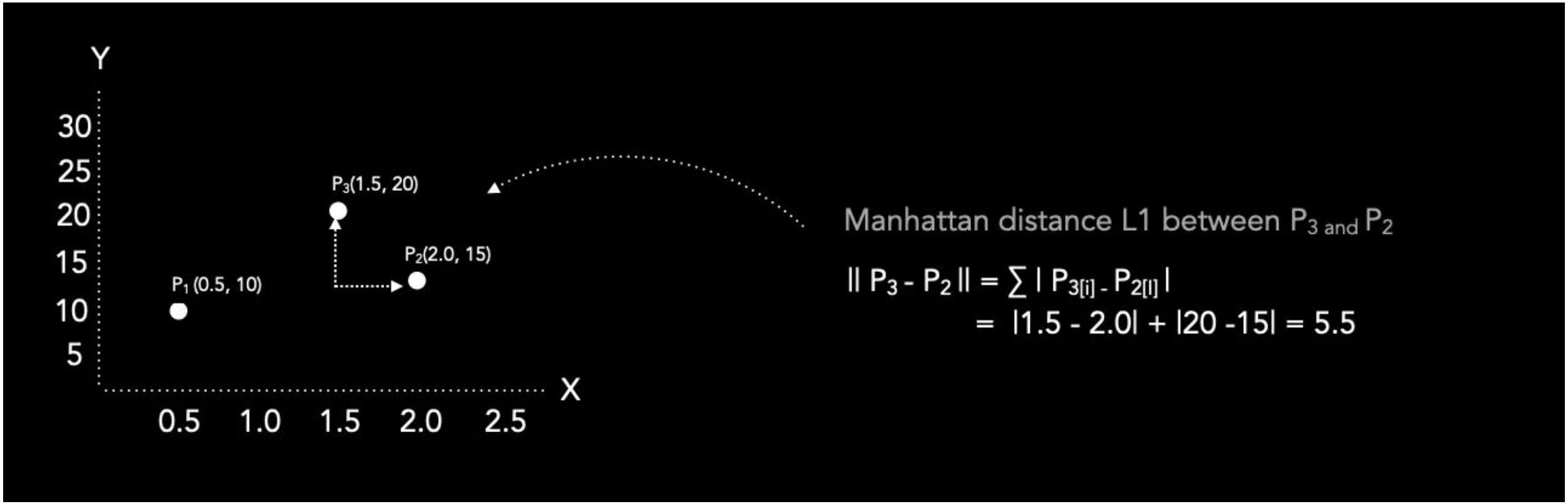
Similarity Measurement Techniques in Image Retrieval

Distance Metrics and Similarity



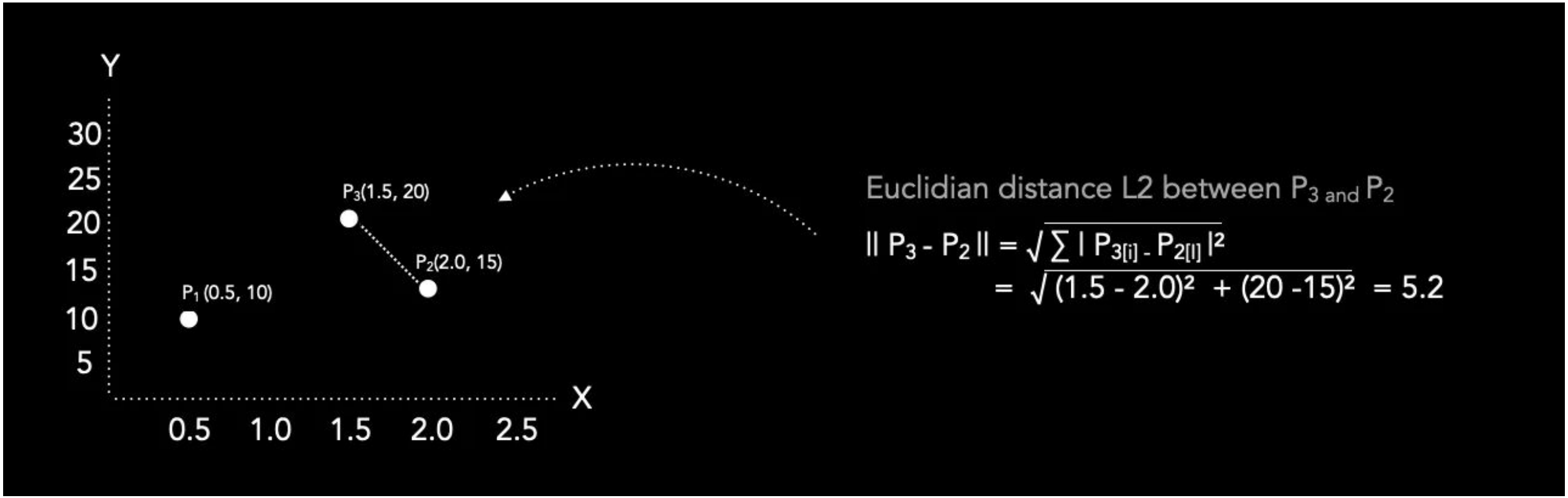
Similarity Measurement Techniques in Image Retrieval

Manhattan Distance L1



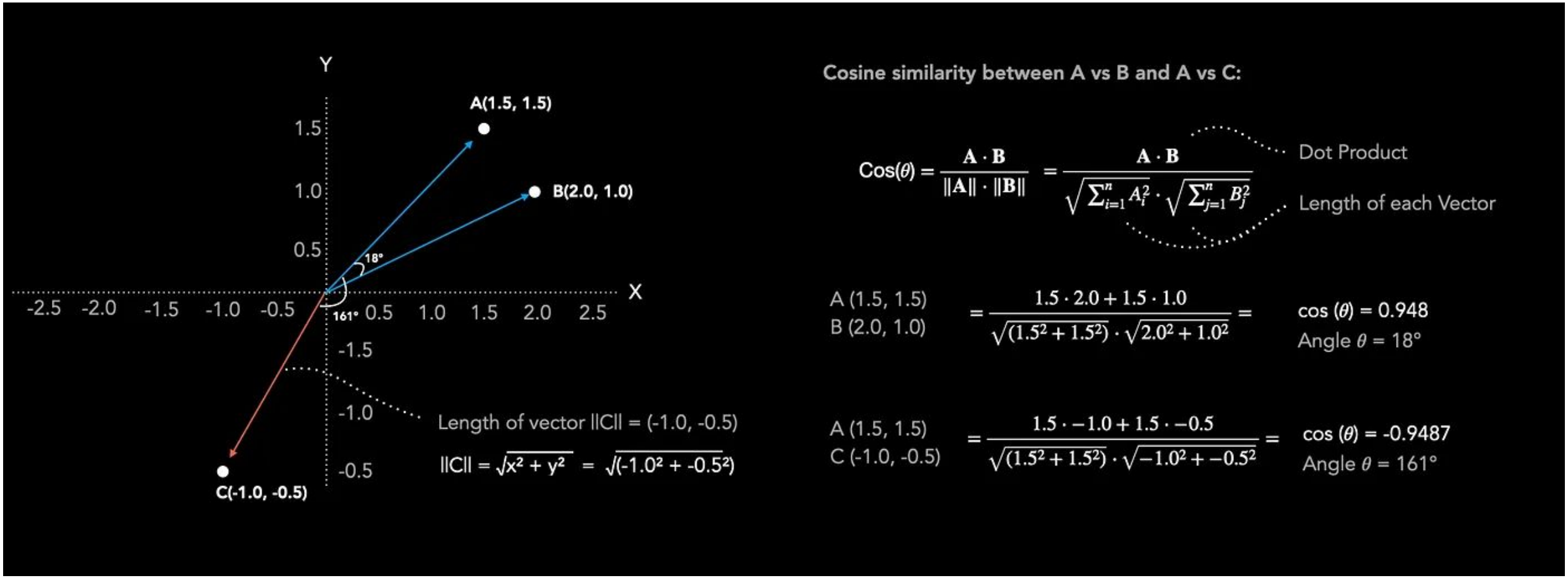
Similarity Measurement Techniques in Image Retrieval

Euclidean Distance L2



Similarity Measurement Techniques in Image Retrieval

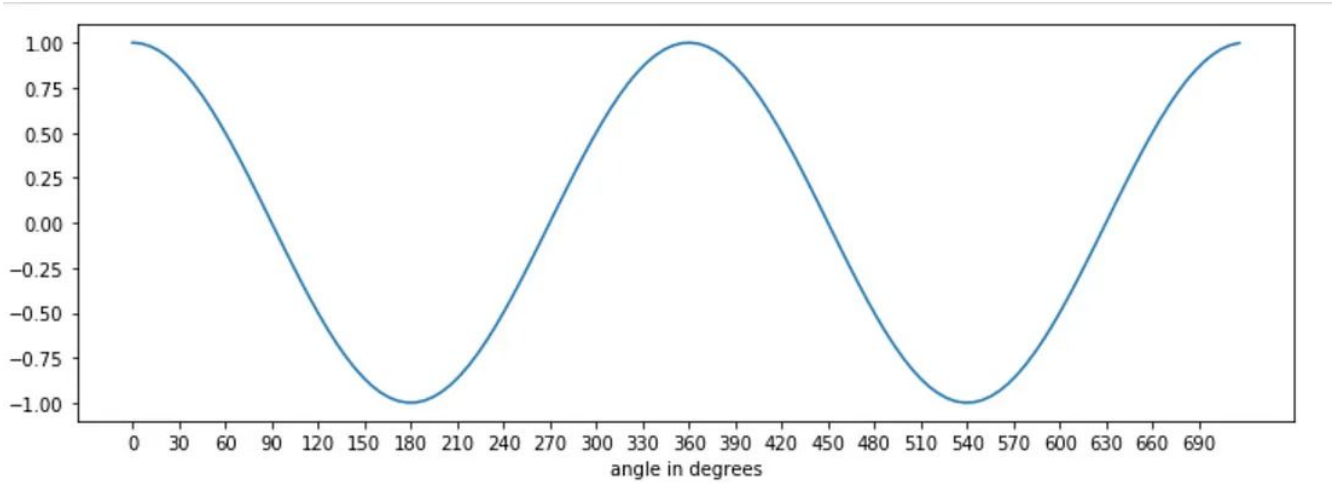
Cosine Similarity

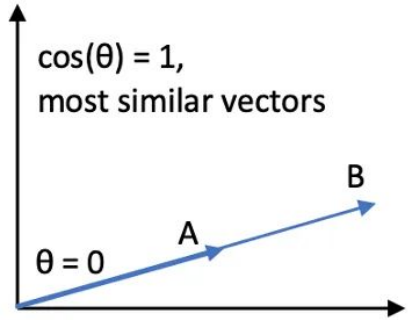


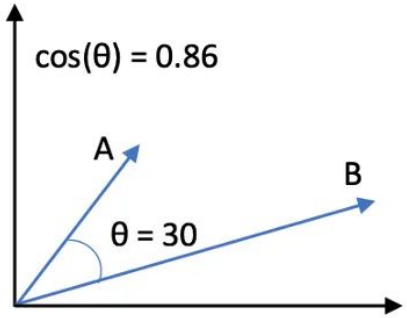
Similarity Measurement Techniques in Image Retrieval

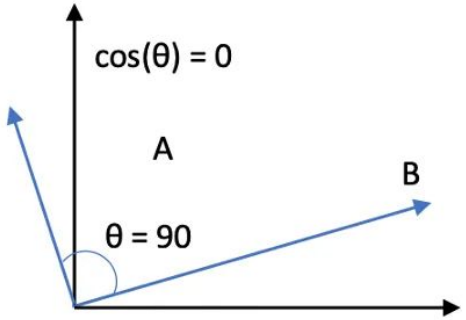
Cosine Similarity

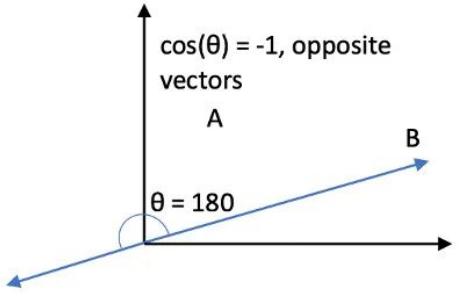
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$





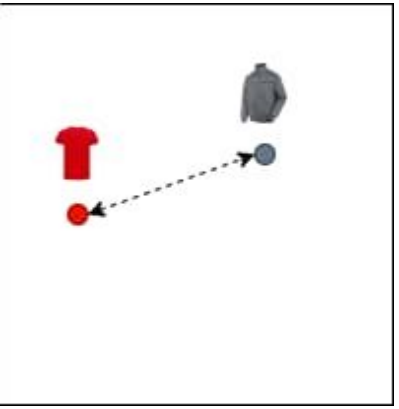






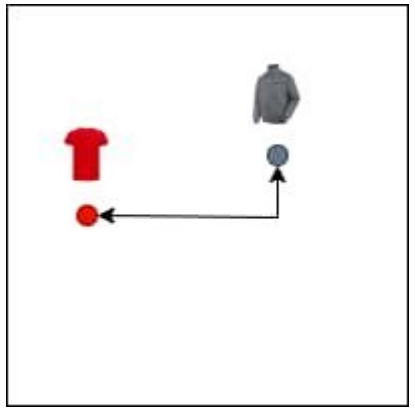
Similarity Measurement Techniques in Image Retrieval

Distance Metrics and Similarity



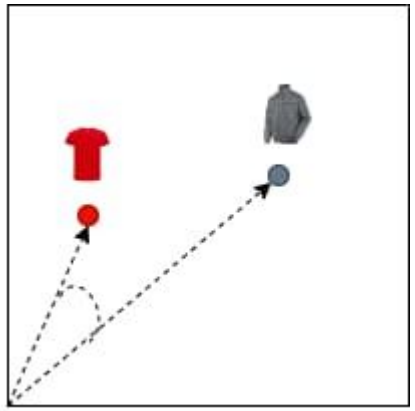
Euclidean Distance:

- **Definition**: Measures the straight-line distance between two points in Euclidean space.
- **Use Case**: Used to determine how far apart two points are in a straight line, considering both the magnitude and direction of the vectors.



Manhattan Distance (L1 Norm):

- **Definition**: Sums the absolute differences of the coordinates of two points.
- **Use Case**: Measures distance by moving along grid lines, often used in contexts where movements are restricted to orthogonal directions.



Cosine Similarity:

- **Definition**: Measures the cosine of the angle between two vector embeddings.
- **Use Case**: Commonly used in text analysis and other domains where the direction of the vector is more important than its magnitude

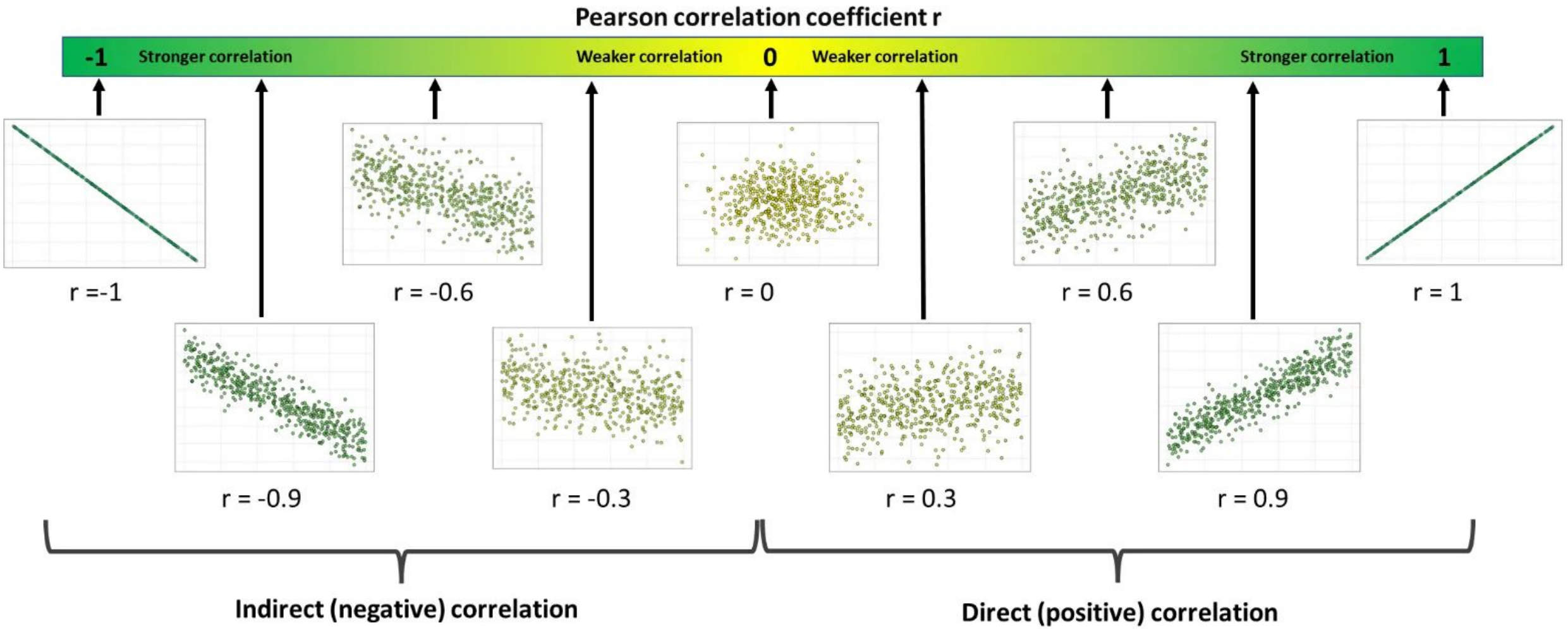
Feature	L1 Distance (Manhattan)	L2 Distance (Euclidean)	Cosine Similarity
Definition	Sum of the absolute differences between the coordinates of the vectors.	Square root of the sum of the squared differences between the coordinates of the vectors.	Measures the cosine of the angle between two non-zero vectors.
Range	$[0,\infty)$	$[0,\infty)$	$[-1,1]$
Usage in Vector Search	Useful when differences in each dimension are important and dimensions are not correlated.	Commonly used due to its straightforward interpretation as the straight-line distance.	Preferred when the magnitude of vectors is less important than their direction.
Sensitivity to Scale	Sensitive; can be affected by scale of dimensions.	Sensitive; can be affected by scale of dimensions.	Not sensitive; invariant to vector scaling.
Impact of Outliers	More robust to outliers compared to L2.	Less robust to outliers due to squaring the differences, making outliers more influential.	Not directly affected by outliers in magnitude, but can be affected by the direction of vectors.
Common Applications	Image processing, text analysis where dimensional independence is key.	Clustering, nearest neighbor searches where Euclidean space properties are desired.	Text similarity, information retrieval, and any application where the direction of vectors is more important than their magnitude.

Similarity Measurement Techniques in Image Retrieval

Absolute value	Interpretation
0.8 to 1	Very strong relationship
0.6 to 0.8	Strong relationship
0.4 to 0.6	Moderate relationship
0.2 to 0.4	Weak relationship
0 to 0.2	Weak or correlation

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sqrt{\sum_{\Sigma_i=1}^{n}\left(X_i - \bar{X}\right)^2}\sqrt{\sum_{\Sigma_i=1}^{n}\left(Y_i - \bar{Y}\right)^2}}$$

Similarity Measurement Techniques in Image Retrieval



Similarity Measurement Techniques in Image Retrieval

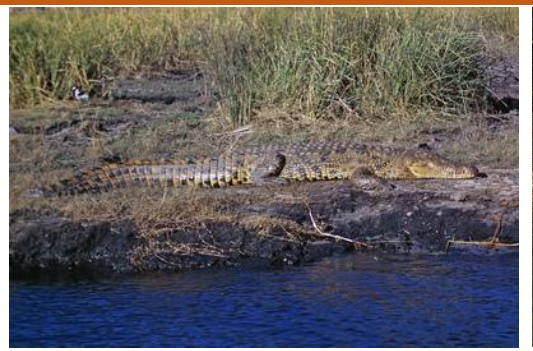
Đối với hai vector có chiều dài bằng nhau, phương pháp nào sau đây được sử dụng để đo sự tương đồng hướng giữa chúng?

A. L1 distance
B. L2 distance
C. Cosine similarity
D. Euclidean distance

Data Preparation for Image Retrieval

Data Preparation for Image Retrieval

Dataset



```
data
├── test
│   ├── African_crocodile
│   ├── ambulance
│   ├── American_egret
│   ├── apiary
│   ├── bannister
│   └── ...
└── train
    ├── African_crocodile
    ├── ambulance
    ├── American_egret
    ├── apiary
    ├── bannister
    └── ...
```

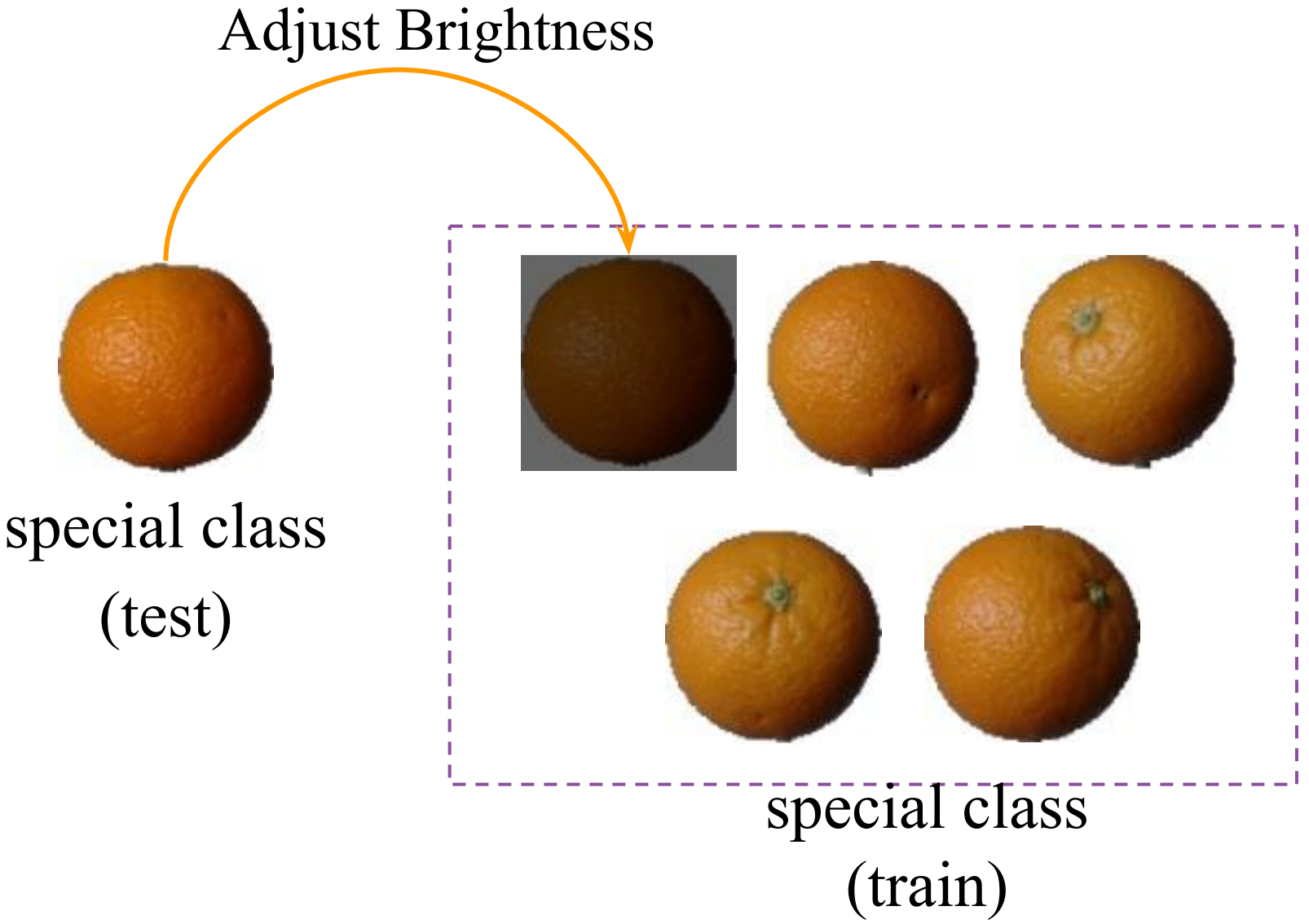
Subset of **ImageNet1K**

- Categories: 59 Classes + 1 special class
- Train: 9 samples/class (5 samples for special class)
- Test: 1 sample/class (1 samples for special class)

Data Preparation for Image Retrieval

Dataset

```
data
├── test
│   ├── African_crocodile
│   ├── ambulance
│   ├── American_egret
│   ├── apiary
│   ├── bannister
│   └── ...
└── train
    ├── African_crocodile
    ├── ambulance
    ├── American_egret
    ├── apiary
    ├── bannister
    └── ...
```



Data Preparation for Image Retrieval

Import Library/Modules

```python
import os
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
```

```
['African_crocodile', 'American_egret', 'Doberman', 'Lakeland_terrier',
'Orange_easy', 'Rhodesian_ridgeback', 'ambulance', 'apiary', 'bannister',
'barn_spider', 'basketball', 'black_widow', 'brain_coral', 'bullet_train',
'can_opener', 'capuchin', 'car_mirror', 'carousel', 'castle', 'china_cabinet',
'cliff_dwelling', 'comic_book', 'conch', 'cornet', 'dugong',
'electric_locomotive', 'flamingo', 'flatworm', 'goldfish', 'grocery_store',
'guillotine', 'half_track', 'hen-of-the-woods', 'horizontal_bar', 'killer_whale',
'kit_fox', 'knee_pad', 'lion', 'loudspeaker', 'lynx', 'magpie', 'meat_loaf',
'mixing_bowl', 'parachute', 'pizza', 'ram', 'red-backed_sandpiper',
'ruffed_grouse', 'safety_pin', 'scabbard', 'screen', 'steam_locomotive',
'steel_arch_bridge', 'theater_curtain', 'traffic_light', 'triceratops',
'vine_snake', 'warplane', 'white_stork', 'yawl']
```

```python
ROOT = 'data'
CLASS_NAME = sorted(list(os.listdir(f'{ROOT}/train')))
```

```python
def read_image_from_path(path, size):
    im = Image.open(path).convert('RGB').resize(size)
    return np.array(im)
```

All images must have same size

Data Preparation for Image Retrieval

```
5  def folder_to_images(folder, size):
6      list_dir = [folder + '/' + name for name in os.listdir(folder)]
7      images_np = np.zeros(shape=(len(list_dir), *size, 3))
8      images_path = []
9      for i, path in enumerate(list_dir):
10         images_np[i] = read_image_from_path(path, size)
11         images_path.append(path)
12     images_path = np.array(images_path)
13     return images_np, images_path
```

"List all file paths for images belonging to a specific class

Load all images belonging to a specific class and return them along with their corresponding file paths

Basic Image Retrieval System

AI VIETNAM
All-in-One Course

Introduction to Image Retrieval

Summary

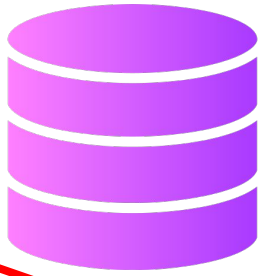
Images Library

CLIP Model (Pretrained Model)

Feature Vectors

Indexing

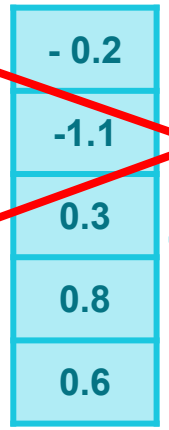
(Chroma) Vector Database



Query Image

CLIP Model (Pretrained Model)

Feature Vector



Similarity Computation

Ranking

Results

Basic Image Retrieval System

Manhattan Distance L1

Images Folders

folder_to_images

Results

Query Image

read_image_from_path

absolute_difference

Ranking

Basic Image Retrieval System

Manhattan Distance L1

$$L1(\vec{a}, \vec{b}) = \sum_{i=1}^{N} |a - b|$$

```python
def absolute_difference(query, data):
    axis_batch_size = tuple(range(1,len(data.shape)))
    return np.sum(np.abs(data - query), axis=axis_batch_size)
```

Purpose: This function calculates the sum of absolute differences between a given query and each element in an array `X`, along all dimensions except the first one.

Parameters:

- `query`: A value or array to compare against.
- `X`: An array containing values to compare with `query`.

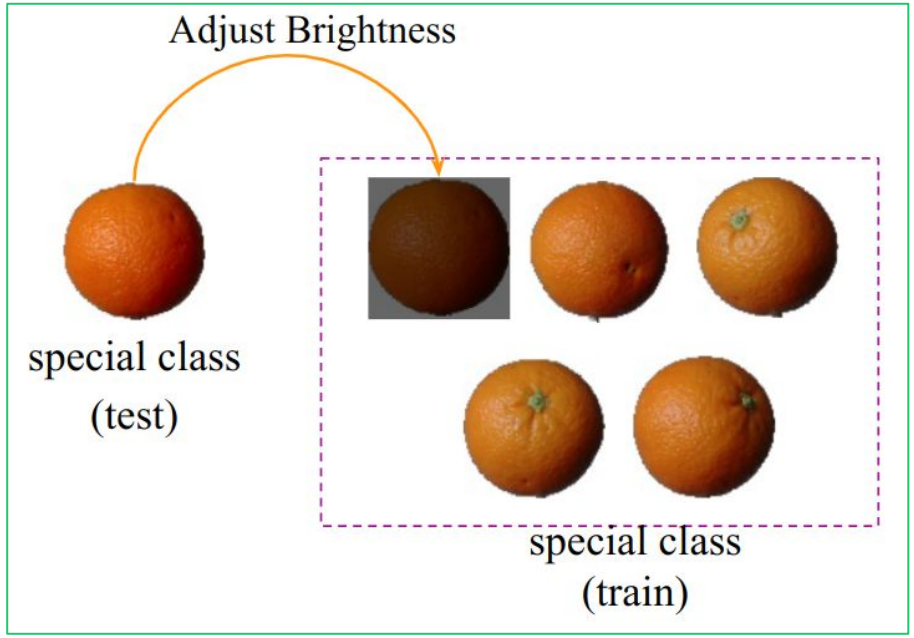
Basic Image Retrieval System

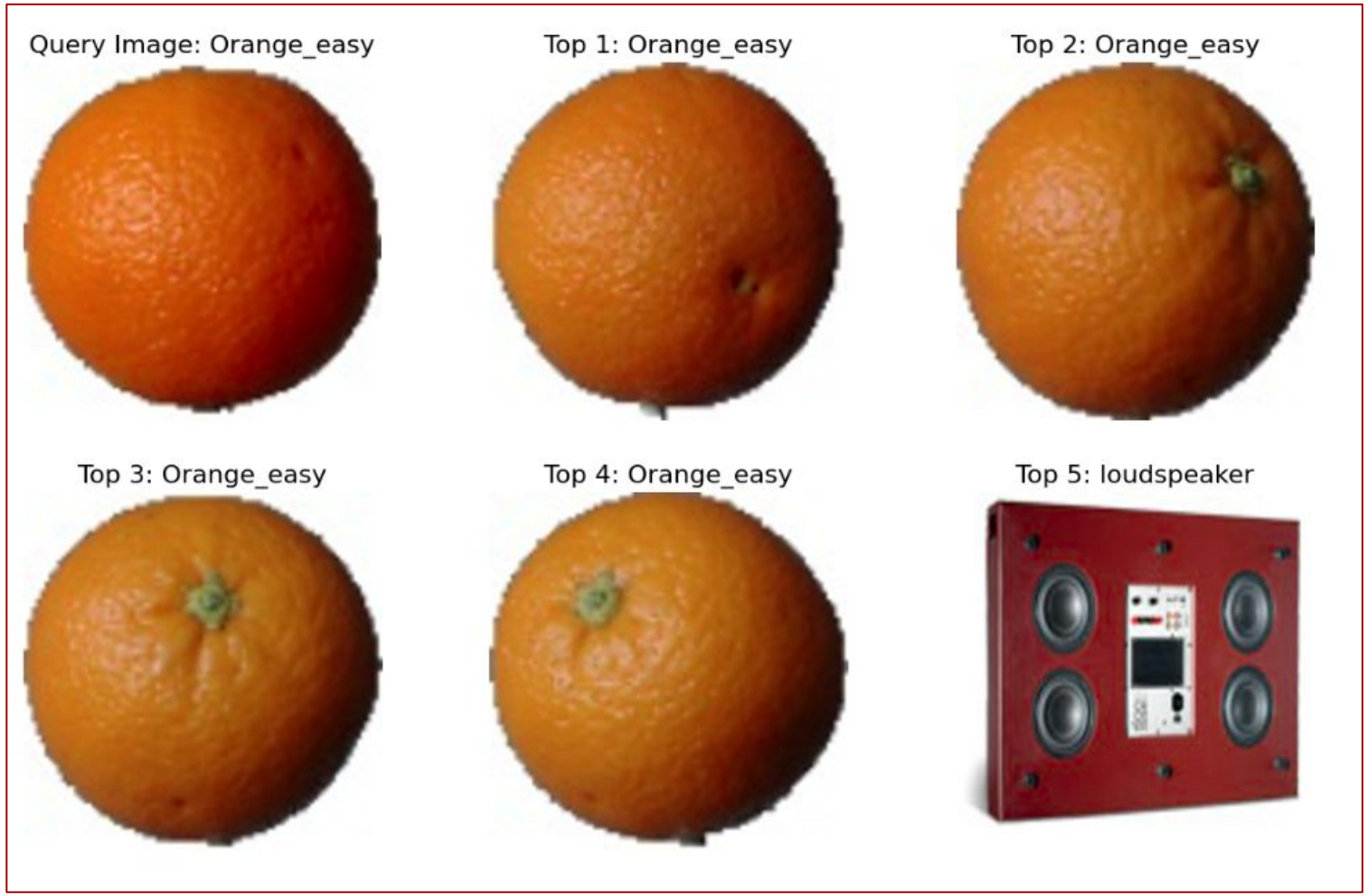
Manhattan Distance L1

```python
def get_l1_score(root_img_path, query_path, size):
    query = read_image_from_path(query_path, size)
    ls_path_score = []
    for folder in os.listdir(root_img_path):
        if folder in CLASS_NAME:
            path = root_img_path + folder
            images_np, images_path = folder_to_images(path, size) # mang numpy nhieu anh,
                paths
            rates = absolute_difference(query, images_np)
            ls_path_score.extend(list(zip(images_path, rates)))
    return query, ls_path_score
```

Basic Image Retrieval System

Manhattan Distance L1





Basic Image Retrieval System

Manhattan Distance L1

Query Image: African_crocodile

Top 1: barn_spider

Top 2: American_egret

Top 3: ambulance

Top 4: kit_fox

Top 5: flatworm

Basic Image Retrieval System

Euclidean Distance L2

Images Folders

folder_to_images

Results

Query Image

read_image_from_path

mean_square_difference

Ranking

Basic Image Retrieval System

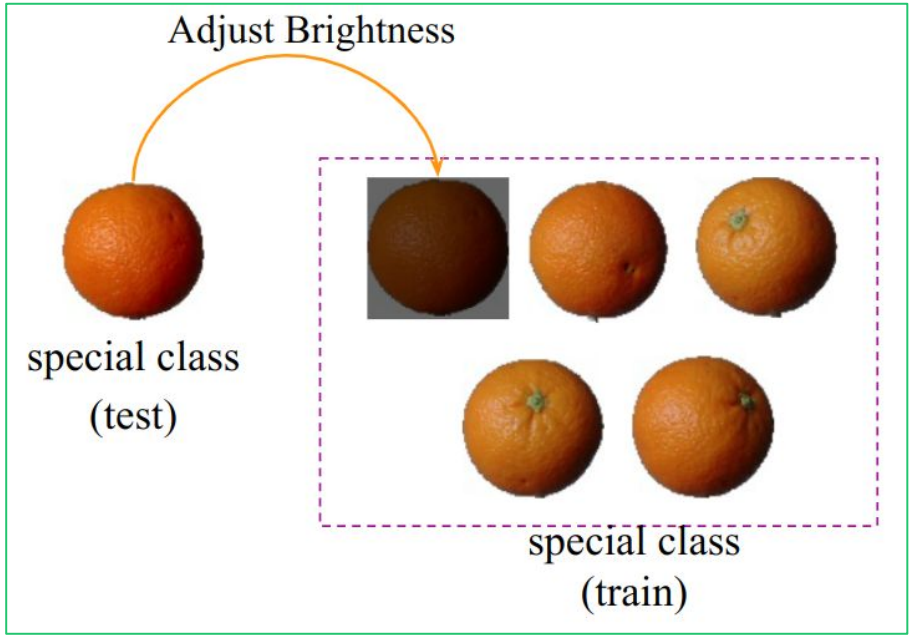
Euclidean Distance L2

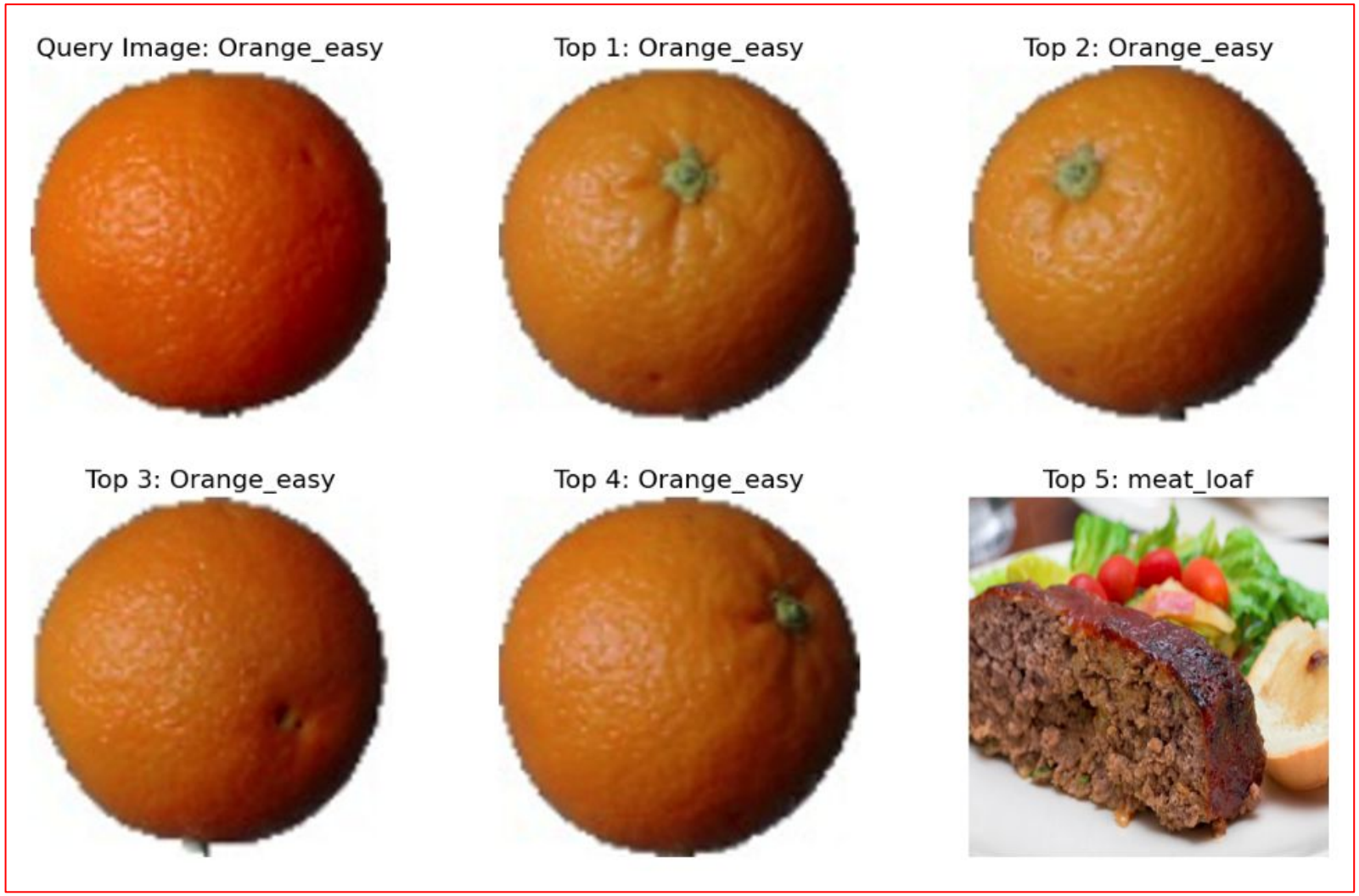
$$L2(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}$$

```python
def mean_square_difference(query, data):
    axis_batch_size = tuple(range(1,len(data.shape)))
    return np.mean((data - query)**2, axis=axis_batch_size)
```

Basic Image Retrieval System

Euclidean Distance L2





Basic Image Retrieval System

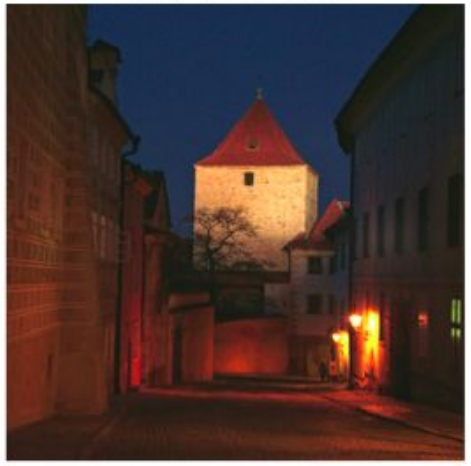
Euclidean Distance L2

Query Image: African_crocodile

Top 1: American_egret

Top 2: kit_fox

Top 3: castle



Top 4: vine_snake

Top 5: barn_spider

Basic Image Retrieval System

Cosine Similarity

Images Folders

folder_to_images

Results

Query Image

read_image_from_path

cosine_similarity

Ranking

Basic Image Retrieval System

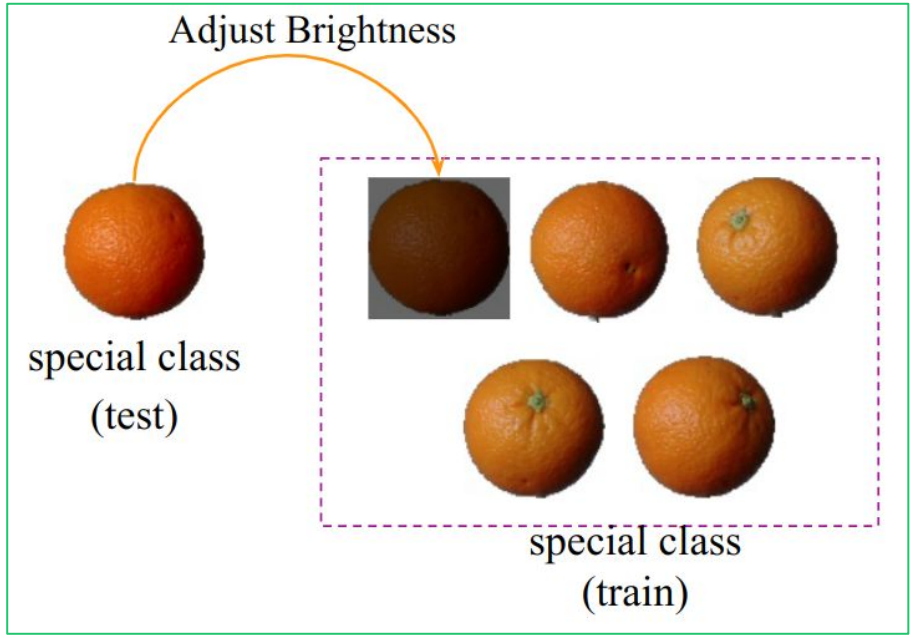
Cosine Similarity

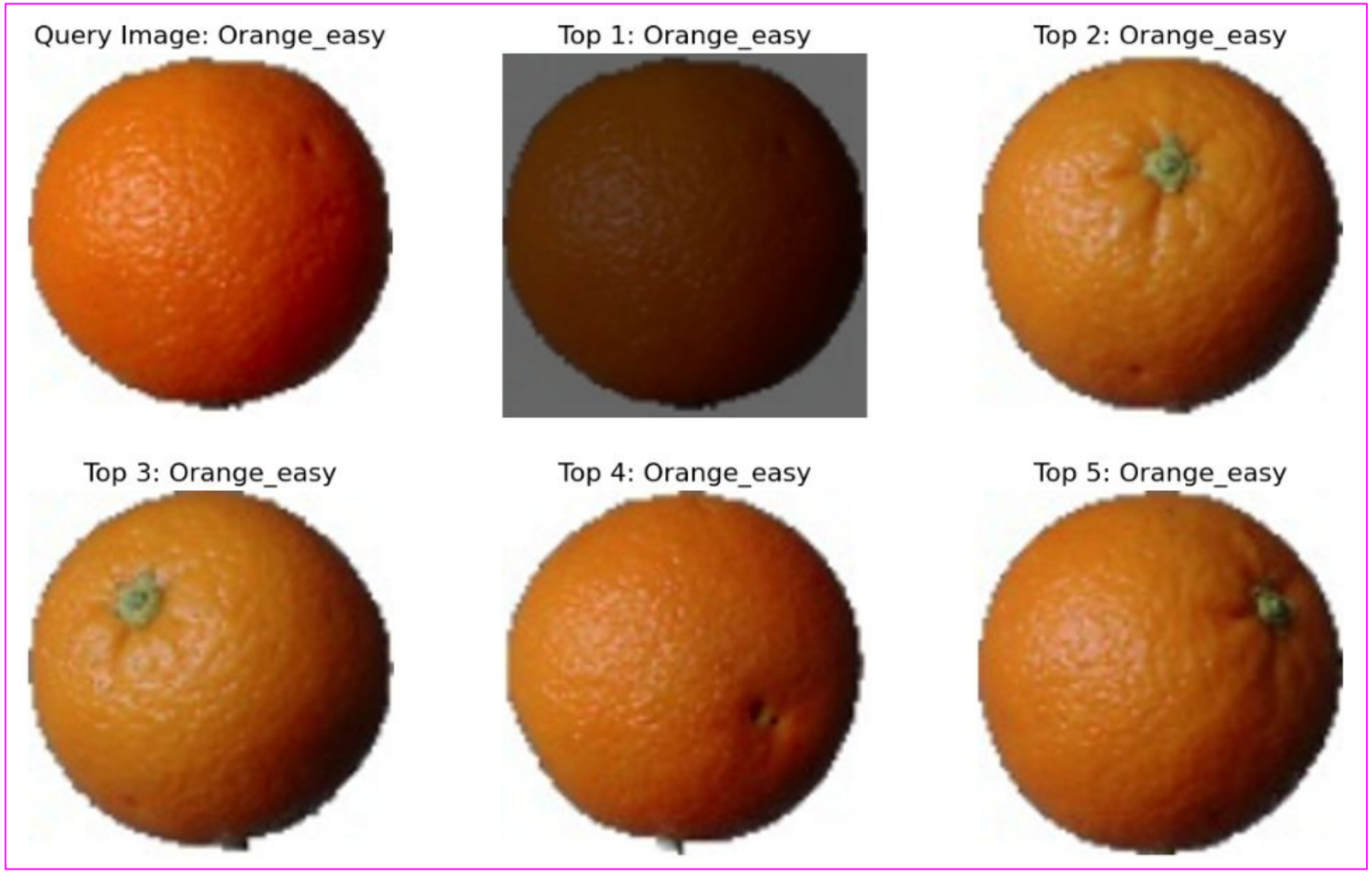
$$cosine\_similarity(\vec{a}, \vec{b}) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{\sum_{i=1}^{N} a_i^2}\sqrt{\sum_{i=1}^{N} b_i^2}}$$

```
1  def cosine_similarity(query, data):
2      axis_batch_size = tuple(range(1,len(data.shape)))
3      query_norm = np.sqrt(np.sum(query**2))
4      data_norm = np.sqrt(np.sum(data**2, axis=axis_batch_size))
5      return np.sum(data * query, axis=axis_batch_size) / (query_norm*data_norm + np.finfo(float
           ).eps)
```

Basic Image Retrieval System

Cosine Similarity





Basic Image Retrieval System

Cosine Similarity

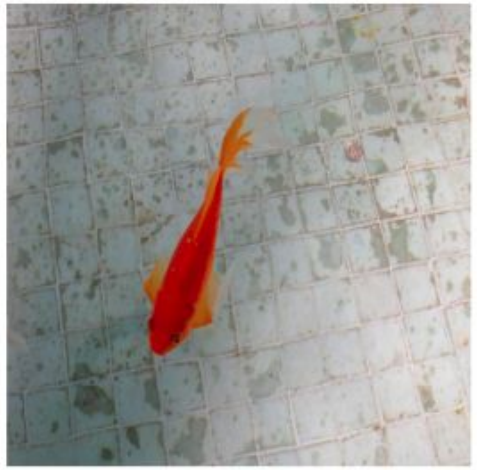
Query Image: African_crocodile

Top 1: black_widow

Top 2: bannister

Top 3: flamingo

Top 4: goldfish



Top 5: black_widow

Basic Image Retrieval System

Correlation Coefficient

Images Folders

folder_to_images

Results

Query Image

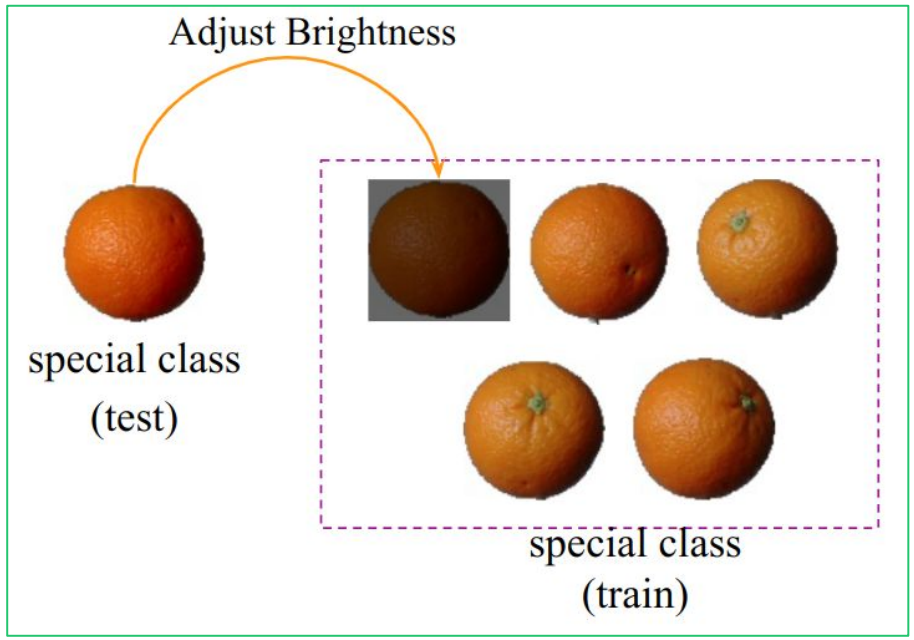
read_image_from_path

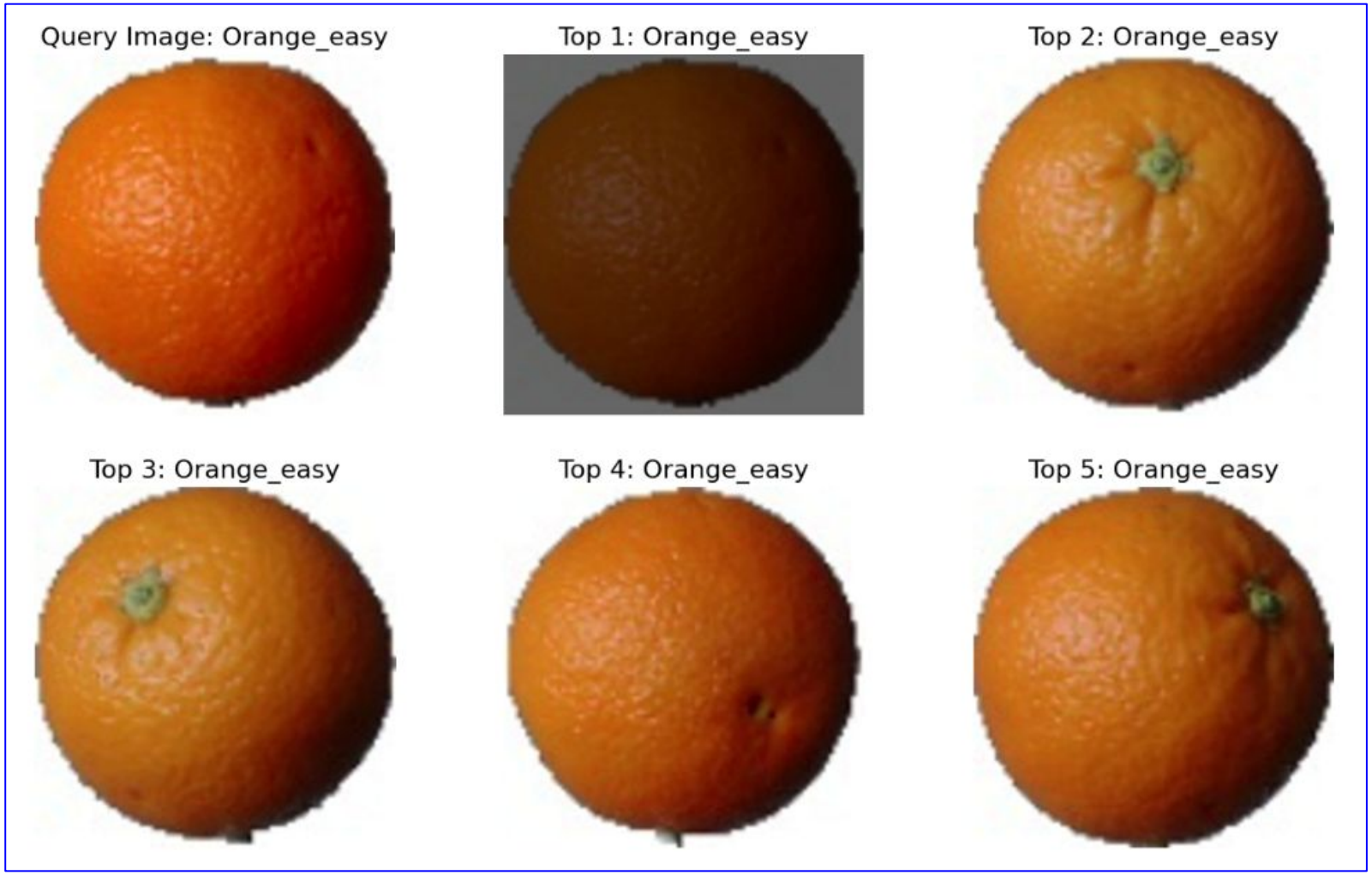
correlation_coefficient

Ranking

Basic Image Retrieval System

Correlation Coefficient





Basic Image Retrieval System

Correlation Coefficient

Query Image: African_crocodile

Top 1: bannister

Top 2: black_widow

Top 3: magpie

Top 4: flamingo

Top 5: barn_spider

Basic Image Retrieval System

Trong bài toán truy vấn hình ảnh, khoảng cách L2 (Euclidean distance) trực tiếp trên các giá trị pixel của ảnh thường được sử dụng để:

A. Đo lường sự khác biệt về hướng giữa các vector pixel của hình ảnh
B. Tìm kiếm các hình ảnh có hướng vector pixel tương tự nhau
C. Đo lường sự khác biệt về độ lớn giữa các giá trị pixel của hai hình ảnh
D. Đo lường sự khác biệt tuyệt đối giữa các vector pixel của hình ảnh

Advanced Image Retrieval Using Pre-trained Deep Learning Model

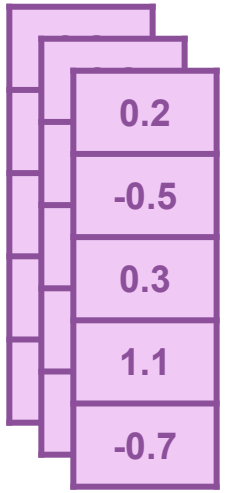
Advanced Image Retrieval Using Pre-trained Deep Learning Model

Summary

Images Library

CLIP Model (Pretrained Model)

Feature Vectors



Indexing

(Chroma) Vector Database

Query Image

CLIP Model (Pretrained Model)

Feature Vector

Similarity Computation

Ranking

Results

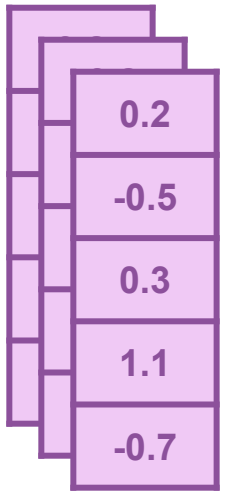
Advanced Image Retrieval Using Pre-trained Deep Learning Model

Summary

Images Library

CLIP Model (Pretrained Model)

Feature Vectors



Query Image

CLIP Model (Pretrained Model)

Feature Vector

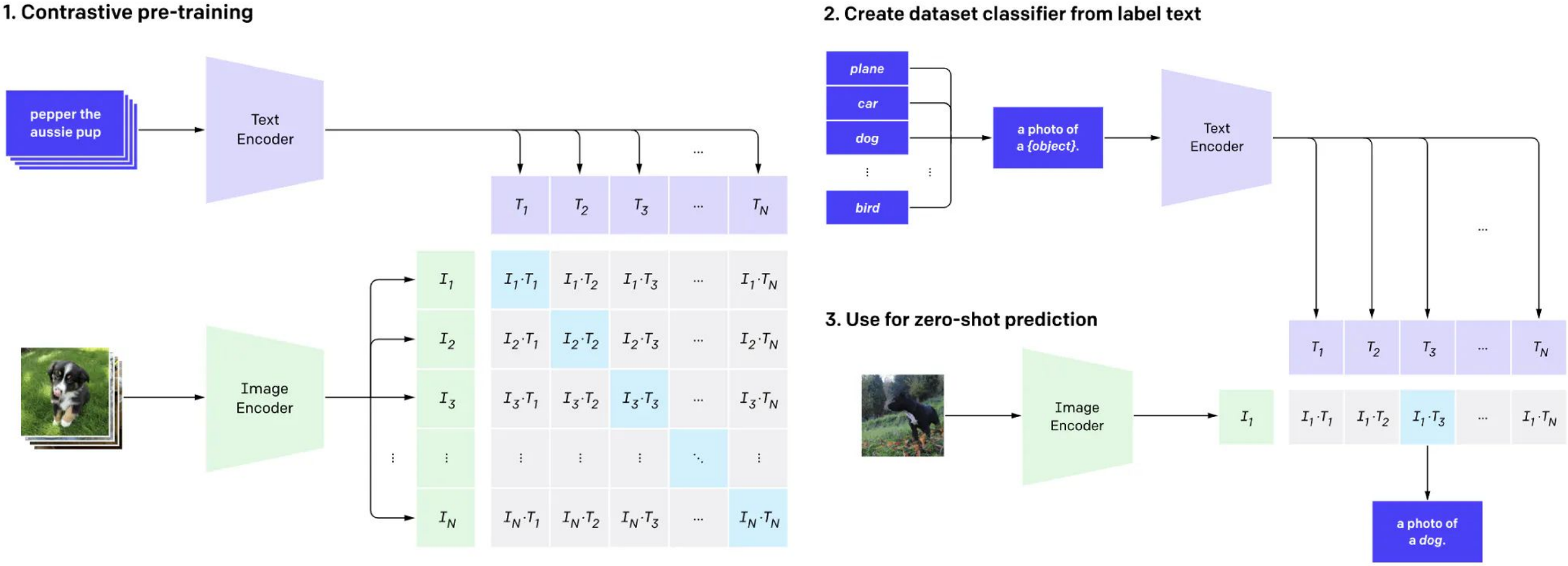
Similarity Computation

Ranking

Results

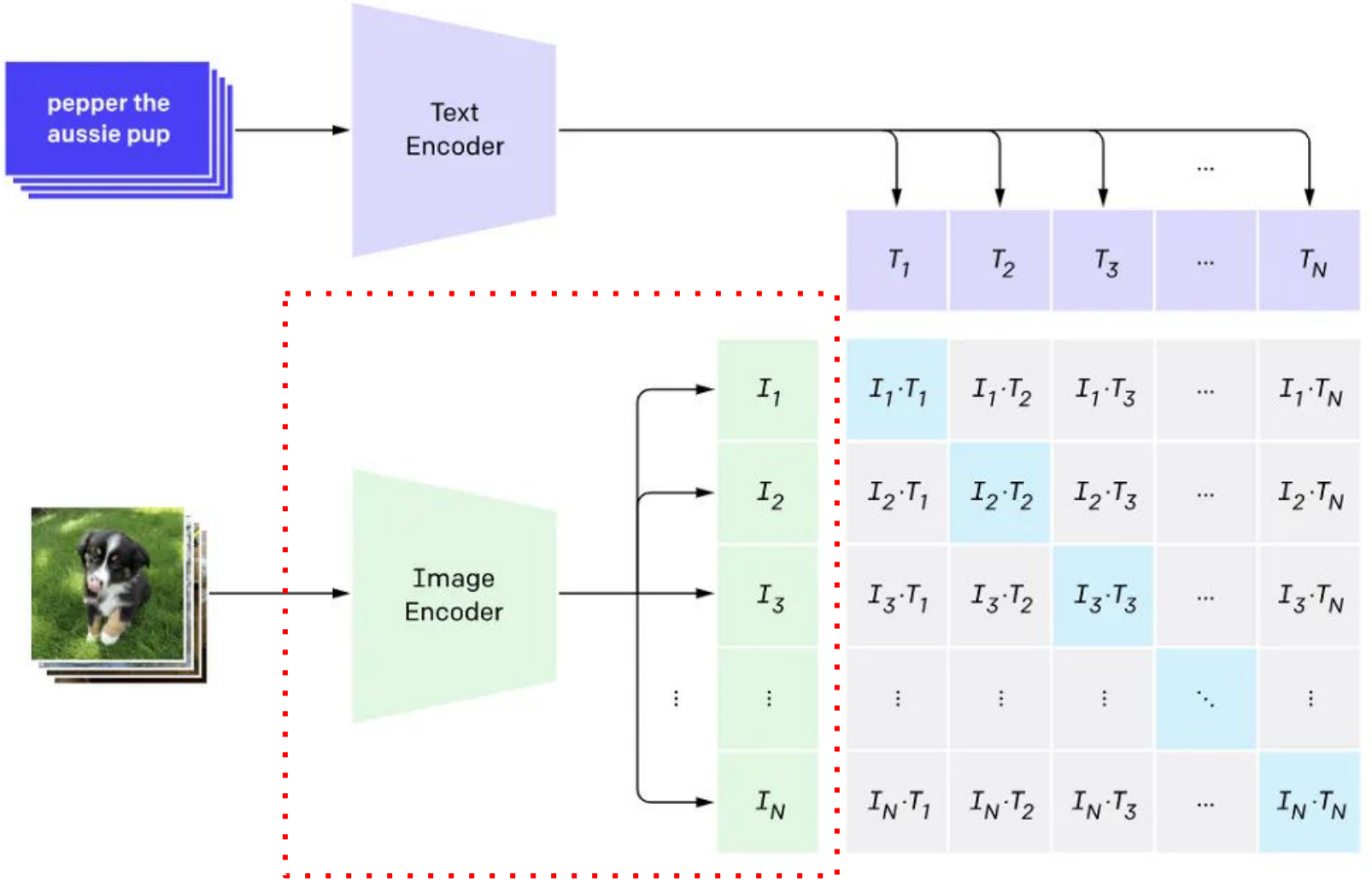
Advanced Image Retrieval Using Pre-trained Deep Learning Model

How does the CLIP algorithm work?



Advanced Image Retrieval Using Pre-trained Deep Learning Model

Get Embeddings with CLIP Model



Advanced Image Retrieval Using Pre-trained Deep Learning Model

Get Embeddings with CLIP Model

```
pip install chromadb
pip install open-clip-torch
```

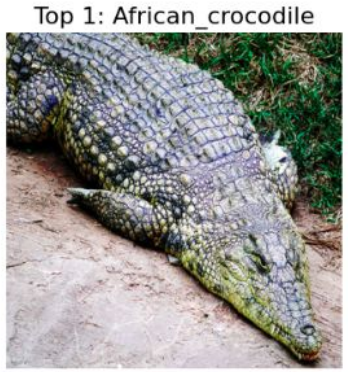
```
from chromadb.utils.embedding_functions import OpenCLIPEmbeddingFunction
```

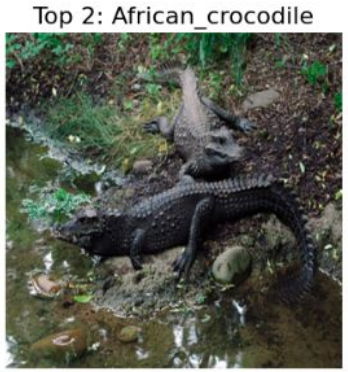
```
embedding_function = OpenCLIPEmbeddingFunction()

def get_single_image_embedding(image):
    embedding = embedding_function._encode_image(image=image)
    return np.array(embedding)
```

Basic Image Retrieval System

Manhattan Distance L1





Basic Image Retrieval System

Euclidean Distance L2

Query Image: Orange_easy

Top 1: Orange_easy

Top 2: Orange_easy

Top 3: Orange_easy

Top 4: Orange_easy

Top 5: Orange_easy

Query Image: African_crocodile

Top 1: African_crocodile

Top 2: African_crocodile

Top 3: African_crocodile

Top 4: African_crocodile

Top 5: African_crocodile

Basic Image Retrieval System

Cosine Similarity

Query Image: Orange_easy

Top 1: Orange_easy

Top 2: Orange_easy

Top 3: Orange_easy

Top 4: Orange_easy

Top 5: Orange_easy

Query Image: African_crocodile

Top 1: African_crocodile

Top 2: African_crocodile

Top 3: African_crocodile

Top 4: African_crocodile

Top 5: African_crocodile

Basic Image Retrieval System

Correlation Coefficient

Query Image: Orange_easy

Top 1: Orange_easy

Top 2: Orange_easy

Top 3: Orange_easy

Top 4: Orange_easy

Top 5: Orange_easy

Query Image: African_crocodile

Top 1: African_crocodile

Top 2: African_crocodile

Top 3: African_crocodile

Top 4: African_crocodile

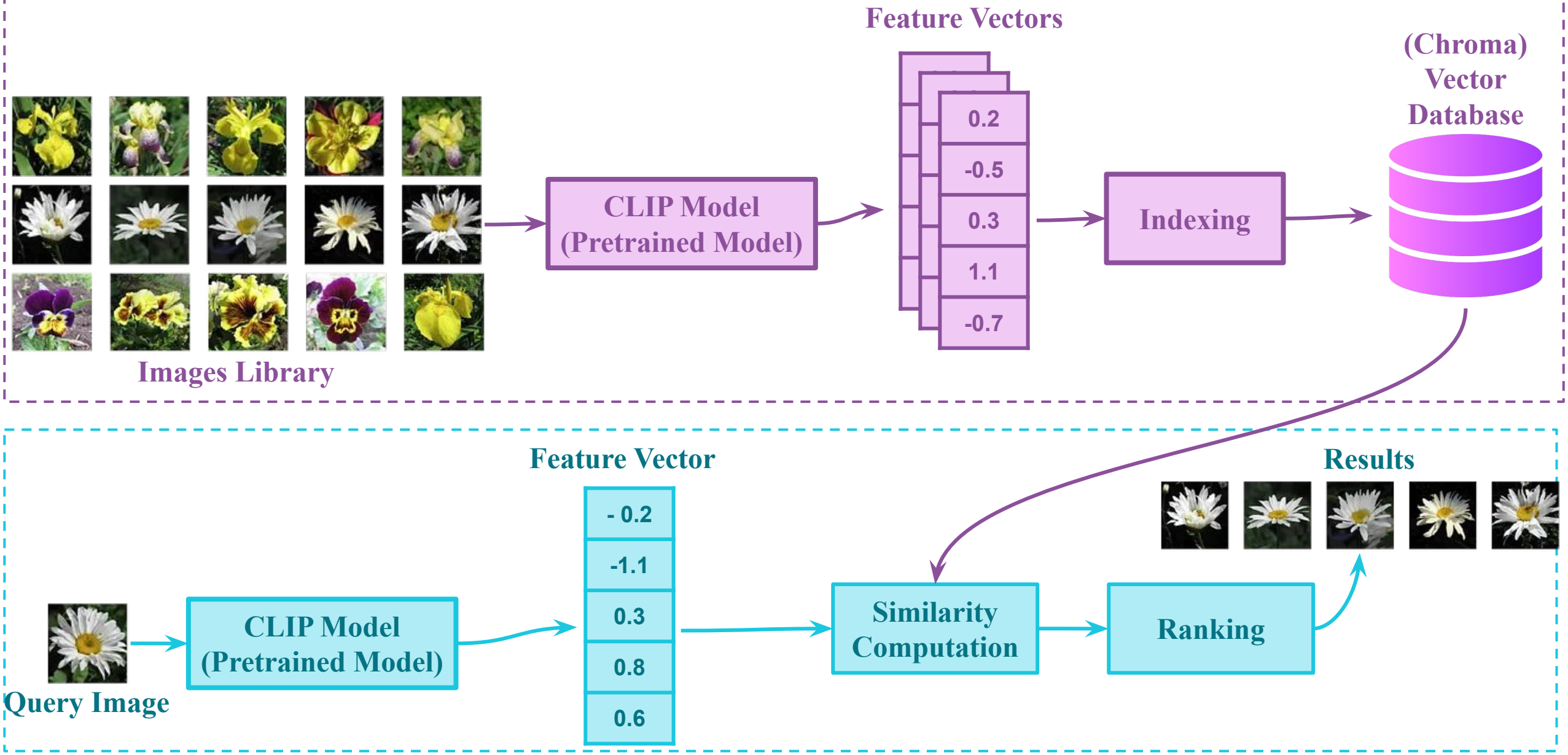
Top 5: African_crocodile


QUIZ
QUIZ
QUIZ
QUIZ

Implementing Image Retrieval with Vector Database

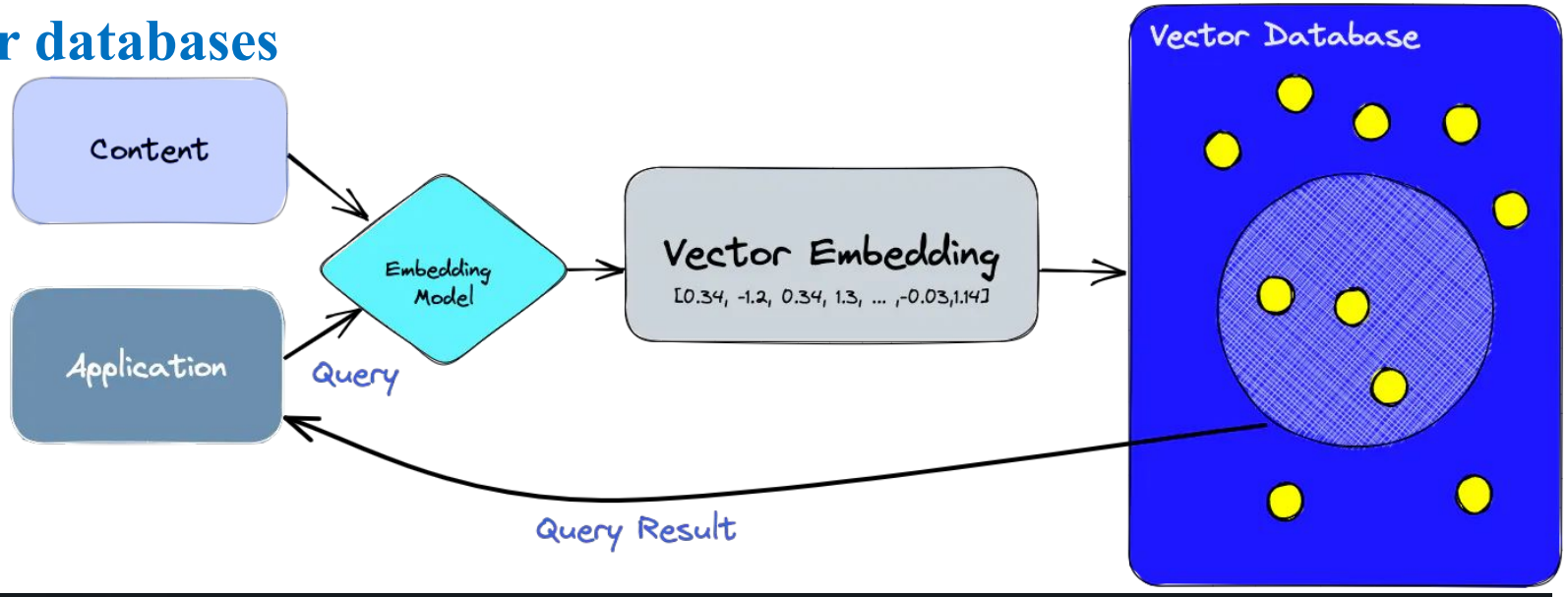
AI VIETNAM
All-in-One Course

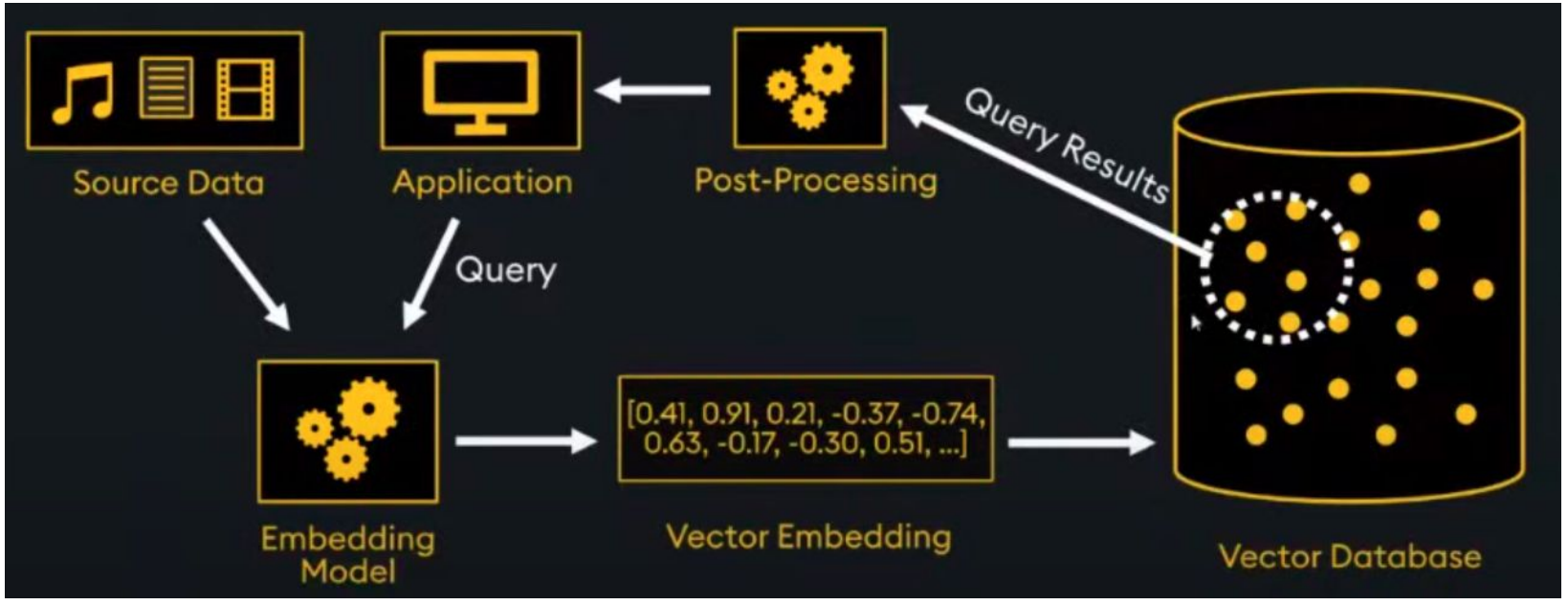
Implementing Image Retrieval with Vector Database



Implementing Image Retrieval with Vector Database

Overview of vector databases



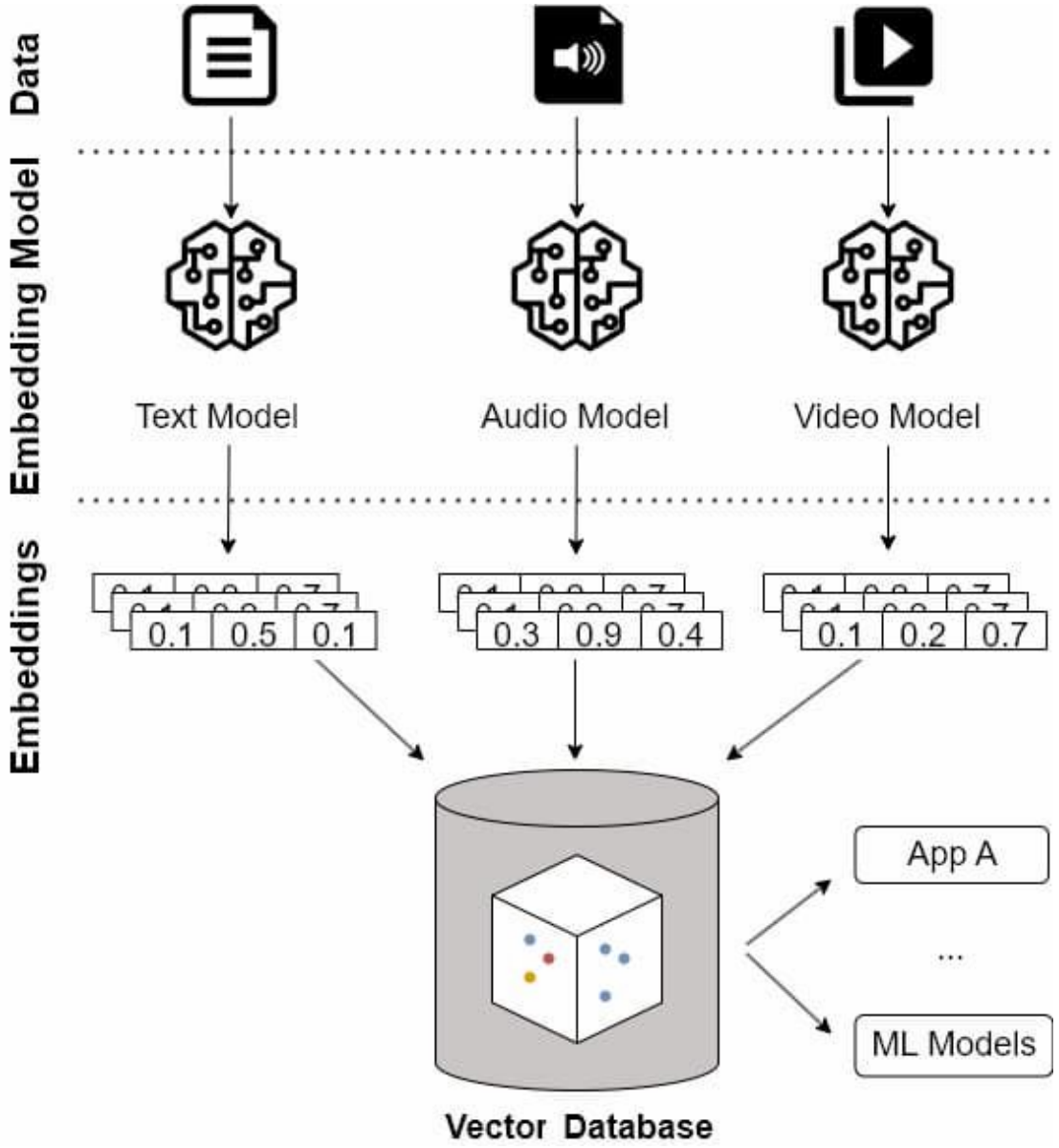


Implementing Image Retrieval with Vector Databases

Overview of vector databases

A **vector database** is a specialized database for storing, searching, and managing information as vectors, which are the numerical representation of objects in a high-dimensional space (e.g., documents, text, images, videos, audio) that capture certain features of the object itself. This numerical representation is called a **vector embedding**

- Designed for scenarios where understanding the context, similarity, or pattern is **more important** than matching exact values
- Data is organized by leveraging mathematics of vectors, and principles of geometry



Implementing Image Retrieval with Vector Databases

Overview of vector databases

Vector Embeddings:

- Created using ML models.
- Translate semantic and qualitative values into numerical representations.
- Different ML models for different data types (text, audio, image).

Vector Databases:

- Suitable for high concurrency and large data volumes.
- Support embedding associations with object metadata.

Benefits:

- Makes vector databases integrable, versatile, and interpretable.
- Similar querying capabilities to traditional databases.

Metadata Association:

- Can include structured and object definitions.
- Enables sophisticated querying and filtering.
- Facilitates better management and integration with data architectures.

Query Processor

Interface

Queries

Attribute | k-NN | ANN | Hybrid

Data Manip. | Range | Batch | Multi.

Operators

Ins | Up | Del | Sort / Top-K | Table Scan

Embed | Proj | Idx Scan | Hybrid Scan

Query Optimizer | Query Executor

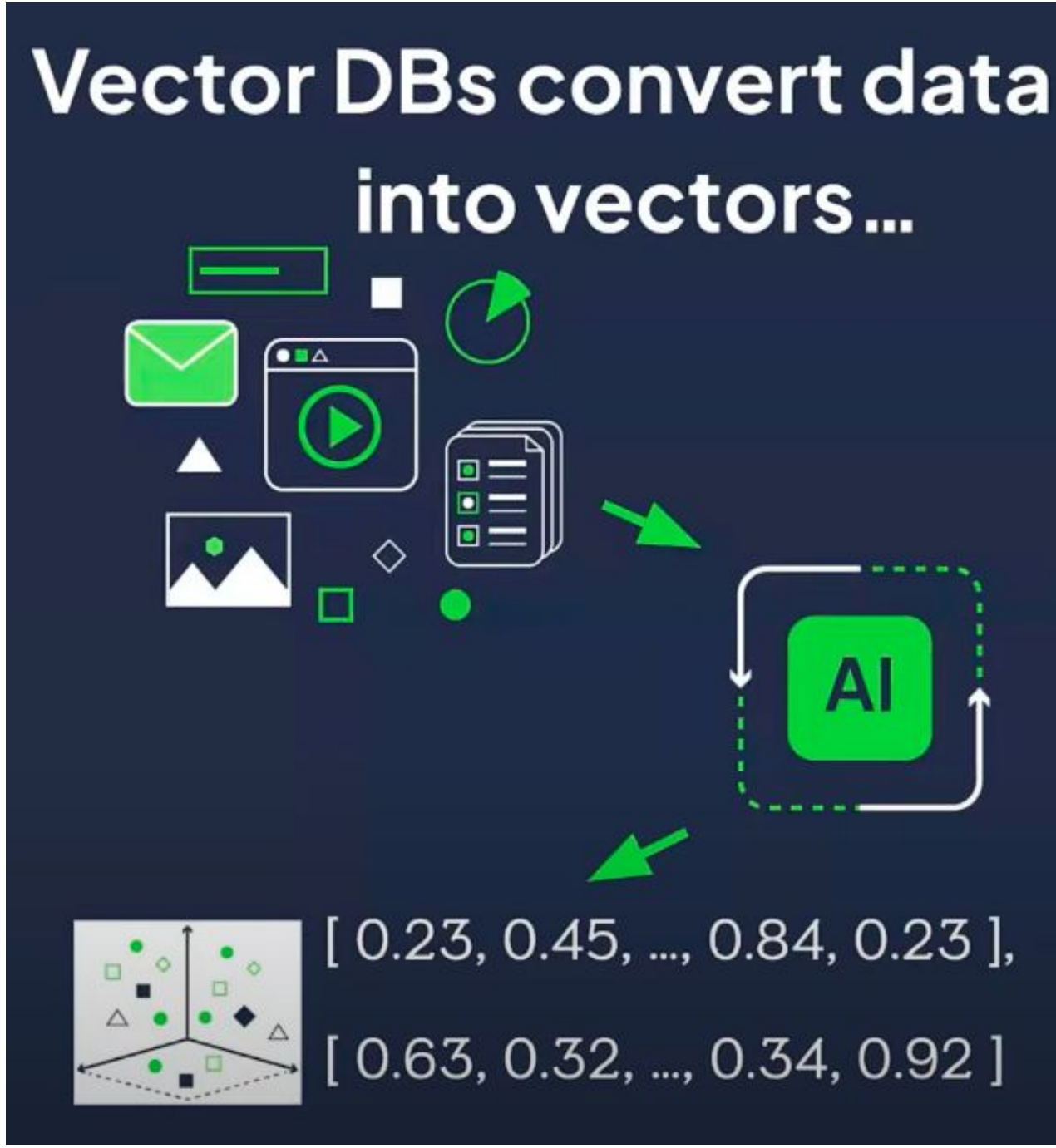
Storage Manager

Search Indexes

LSH | IVF | *k*-d | RP | KGraph | EFANNA | NSG

L2H | FLANN | ANNOY | FANNG | HNSW | NGT

Vector Storage



Images: EXIF metadata

- Added by camera
  - Or post-processing software
- JPEG or HEIC

General | Exif | GPS | TIFF

Altitude	184,47 m (605,2 ft)
Altitude Reference	above sea level
Destination Bearing	21,327
Destination Bearing Refere...	True direction
Horizontal Positioning Error	17,226
Image Direction	21,327
Image Direction Reference	True north
Latitude	44° 43′ 38,772″ N
Longitude	5° 1′ 17,07″ E
Speed	0
Speed Reference	Kilometers per hour

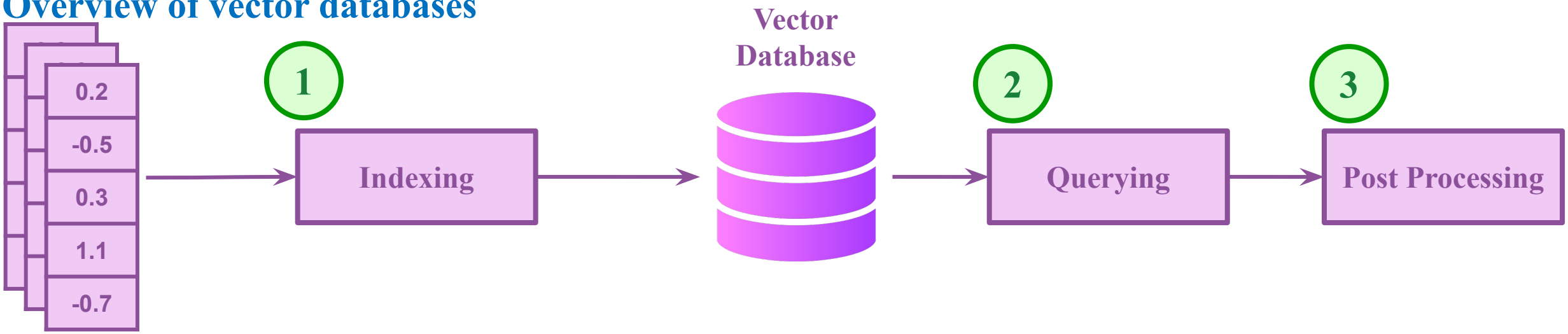
General | Exif | GPS | TIFF

Aperture Value	2
Brightness Value	1,082
Color Space	Uncalibrated
Components Configurati...	1, 2, 3, 0
CompositeImage	2
Date Time Digitized	9 Aug 2023 at 21:15:18
Date Time Original	9 Aug 2023 at 21:15:18
Digital Zoom Ratio	1,217
Exif Version	2.3.2
Exposure Bias Value	0
Exposure Mode	Auto exposure
Exposure Program	Normal program
Exposure Time	1/50
Flash	Off, did not fire
FlashPix Version	1.0
FNumber	2
Focal Length	6
Focal Length In 35mm Fi...	63
Photographic Sensitivity...	400
Lens Make	Apple
Lens Model	iPhone 11 Pro back trip...
Lens Specification	1,54, 6, 1,8, 2,4
Metering Mode	Pattern
Time Zone for Modificati...	+02:00
Time Zone for Digitized...	+02:00
Time Zone for Original D...	+02:00
Pixel X Dimension	4 032
Pixel Y Dimension	3 024
Scene Capture Type	Standard
Scene Type	A directly photograph...
Sensing Method	One-chip color area se...
Shutter Speed Value	1/50
Subject Area	2 022, 1 515, 2 329, 1 395
Sub-second Time Digitiz...	527
Sub-second Time Original	527
White Balance	Auto white balance

This is the village I live in!

Implementing Image Retrieval with Vector Database

Overview of vector databases



1. Indexing: The vector database indexes vectors using an algorithm such as PQ, LSH, or HNSW. This step maps the vectors to a data structure that will enable faster searching.

2. Querying: The vector database compares the indexed query vector to the indexed vectors in the dataset to find the nearest neighbors (applying a similarity metric used by that index)

3. Post Processing: In some cases, the vector database retrieves the final nearest neighbors from the dataset and post-processes them to return the final results. This step can include re-ranking the nearest neighbors using a different similarity measure.

Implementing Image Retrieval with Vector Databases

Overview of vector databases

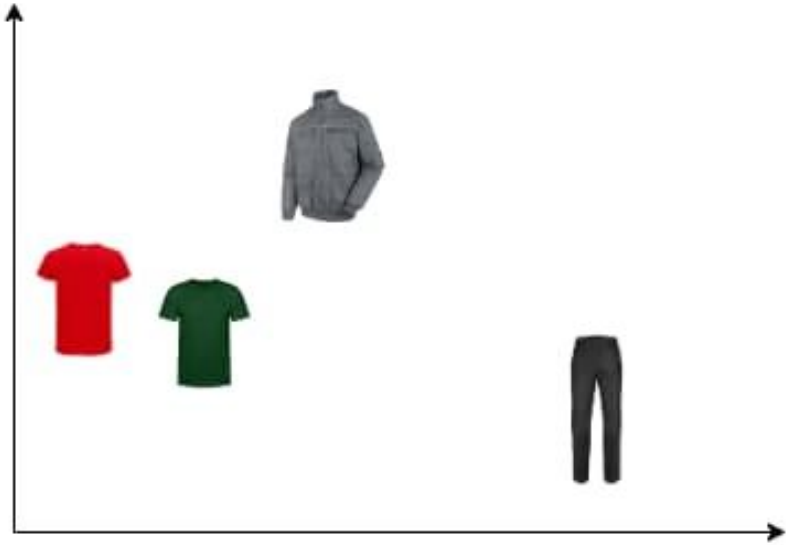
Embedding representation

0.46	-0.30	1.20	0.65

Array of embeddings

0.45	0.32
0.46	0.30
0.70	0.88
2	1.45

Dimensional map

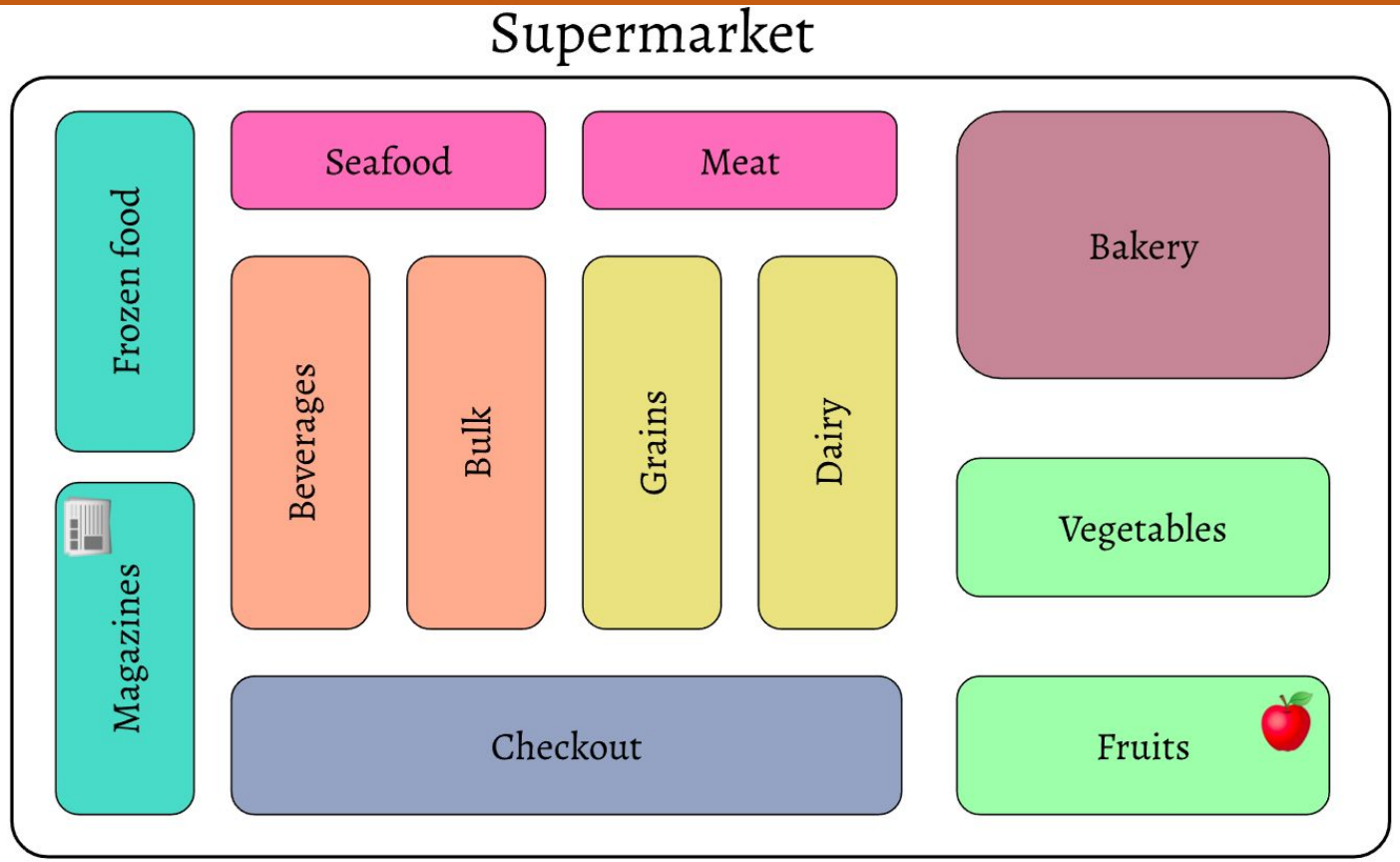


The number of **dimensions** in embeddings is **crucial** because each dimension represents a feature of the object, captured as a numerical value. While more dimensions can **enhance accuracy**, they also require **more computational resources**, including memory, latency, and cost.

- T-shirts are close to each other due to being the same product with different colors.
- The jacket is near the T-shirts because they share attributes like sleeves and a collar.
- Jeans are the furthest to the right, as they do not share attributes with the other products.

Implementing Image Retrieval with Vector Databases

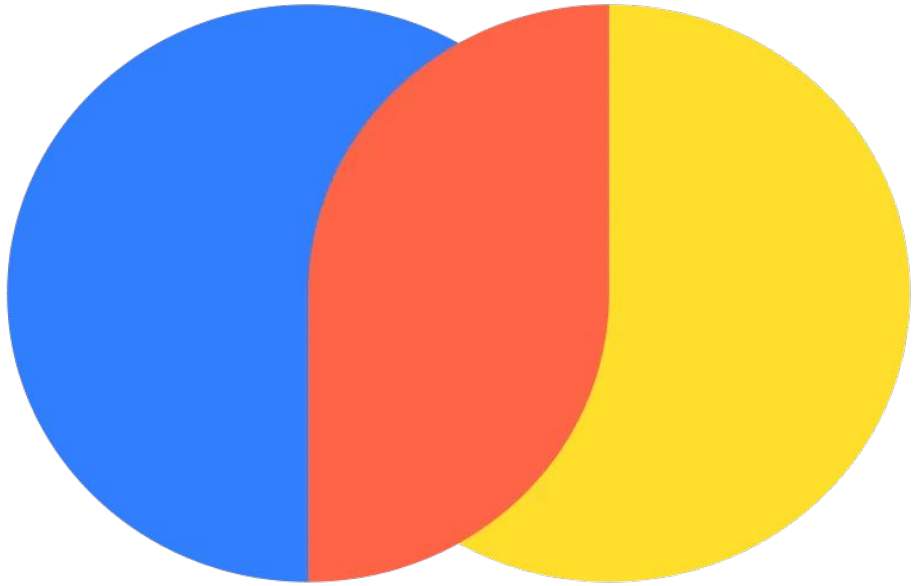
Vector Indexes



- **Definition**: Specialized data structures for efficiently storing, organizing, and querying high-dimensional vector embeddings.
- **Purpose**: Provide fast and cost-effective search queries.
- **Examples**: Approximate nearest neighbor, Inverted index, Locality-sensitive hashing

Implementing Image Retrieval with Vector Databases

Chroma Vector Database



Embeddings made easy

Chroma has all the tools you need to use embeddings

Simple

As easy as `pip install`, use in a notebook in 5 seconds

```
> pip install chromadb
```

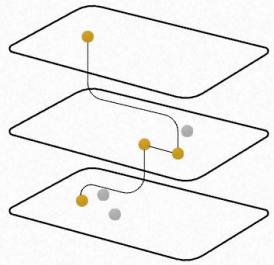
```
import chromadb

client = chromadb.Client()

c = client.create_collection("test")

# add embeddings and documents
c.add(...)

# get back similar ones
c.query(...)
```

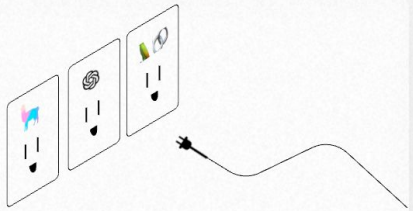


Feature-rich

Search, filtering, and more

JavaScript Client

`npm install chromadb` and it ships with @types

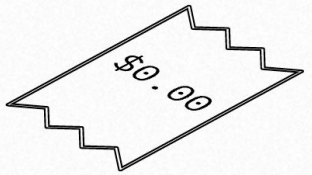


Integrations

Plugs right in to LangChain, LlamaIndex, OpenAI and others

Free

Apache 2.0 and open source



Implementing Image Retrieval with Vector Databases

Vector Indexes

Euclidean Distance L2

Cosine Similarity

Basic Image Retrieval System

Khi sử dụng vector database trong bài toán truy vấn hình ảnh, việc lưu trữ và truy xuất các vector đặc trưng của hình ảnh cho phép:

A. So sánh trực tiếp các giá trị pixel của hình ảnh
B. Tăng tốc quá trình tìm kiếm các hình ảnh tương tự bằng cách so sánh các vector đặc trưng thay vì các hình ảnh gốc
C. Đo lường sự khác biệt về độ lớn giữa các hình ảnh gốc
D. Phân loại các hình ảnh gốc dựa trên kích thước của chúng

Summary

- ✔ Introduce the concept of image retrieval and its applications.
- ✔ Discuss various similarity measurement techniques used in image retrieval.
- ✔ Outline the process of data preparation for image retrieval tasks.
- ✔ Demonstrate how to build a basic image retrieval system.
- ✔ Explore advanced image retrieval methods using pre-trained deep learning models.
- ✔ Explain the implementation of image retrieval with vector databases.
- ✔ (Optional) Provide an overview of data acquisition through web crawling.