Multimedia Retrieval

Rynson W.H. Lau

CS4185 Multimedia Technologies and Applications

Multimedia Indexing

- A database (or the Internet) may contain a large number of multimedia files, including text, images, videos or a mixture of them.
- Due to the large number, how to efficiently and accurately retrieve the desired files becomes important. This depends largely on the *indexing* and *retrieval* techniques used.
- *Indexing* concerns how to abstract (or index) the multimedia information, while *retrieval* concerns how to measure the similarity between a user query and the indices of the multimedia files.

Multimedia Retrieval

- The effectiveness of a retrieval technique is measured by two major metrics: *recall* and *precision*.
- **Recall** measures the ability of retrieving relevant files from the database. It is defined as the ratio between the number of relevant files retrieved and the total number of relevant files in the database, as:

$$\textbf{\textit{Recall}} = \frac{\text{number of relevant files retrieved}}{\text{total number of relevant files in the database}}$$

- **Precision** measures the accuracy of the retrieval. It is defined as the ratio between the number of relevant files retrieved and the total number of files retrieved, as:

$$\textbf{\textit{Precision}} = \frac{\text{number of relevant files retrieved}}{\text{total number of files retrieved}}$$

- Consider an example of retrieving football images from a dataset of 1000 images.
- Given an input image containing a football, we have a retrieval program that would return 10 best matched images from the database.

Input Image

Retrieved Images

- We assume that there are a total of 100 images (out of the 1000 images) in the database that contain footballs.
- So, we have:

$$\textbf{\textit{Recall}} = \frac{\text{number of images retrieved containing fallballs}}{\text{total number of images in the database containing footballs}}$$

$$= \frac{3}{100} = 0.03$$

(3 football images returned by the program)

(100 football images in the database)

$$\textbf{\textit{Precision}} = \frac{\text{number of images retrieved containing footballs}}{\text{total number of images retrieved}}$$

$$= \frac{3}{10} = 0.3$$

(3 football images returned by the program)

(10 images returned by the program)

Object Detection

- Following the previous example, given an input image, we may sometimes need to know if there is a football and where exactly it is in the image.
- This is referred to as *object detection*.
- An object detection program typically outputs a rectangular bounding box indicating the location of the detected object (i.e., football in our example) in the image.
- The effectiveness of the object detection program is usually measured by a metric: *IoU* (Intersection of Union).

- The IoU is defined as:

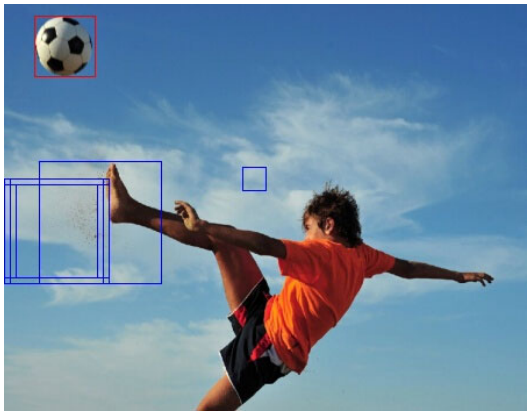
$$IoU = \frac{A \cap B}{A \cup B} = \frac{\text{the overlapping area of A and B}}{\text{the union of the areas of A and B}}$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Input Image

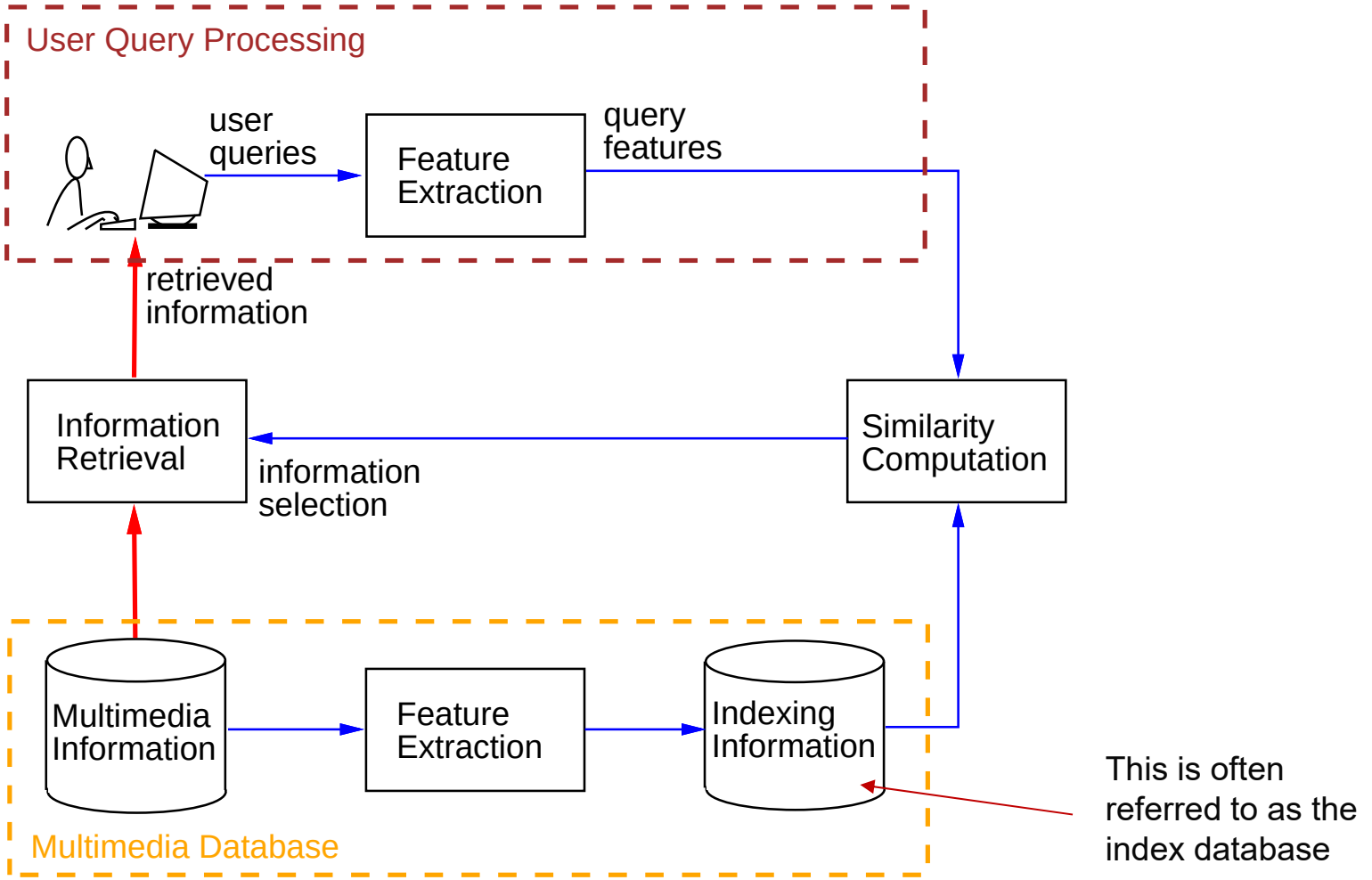
IoU: 0.39

IoU: 0



Object Detection Results

A Typical Retrieval Architecture



- The *Multimedia Database* contains a large number of multimedia files. The system first extracts some features (i.e., representative information) from each multimedia file to form the index of the file.
- An index database is then created containing the indices of all the multimedia files in the database. This is done in advance.
- During runtime, the user inputs a retrieval query for multimedia files. The query can be in the form of a set of words or an input example file.

- The system will then apply a feature extraction process to the input query. This feature extraction process is typically the same as the one applied to the multimedia files.
- This will generate some query features. The similarity computation block (retrieval) will then compare these input query features with those in the index database.
- Finally, the system will return to the user the multimedia files which indices match with the input query features.

Media for Retrieval

There are different types of media for retrieval:

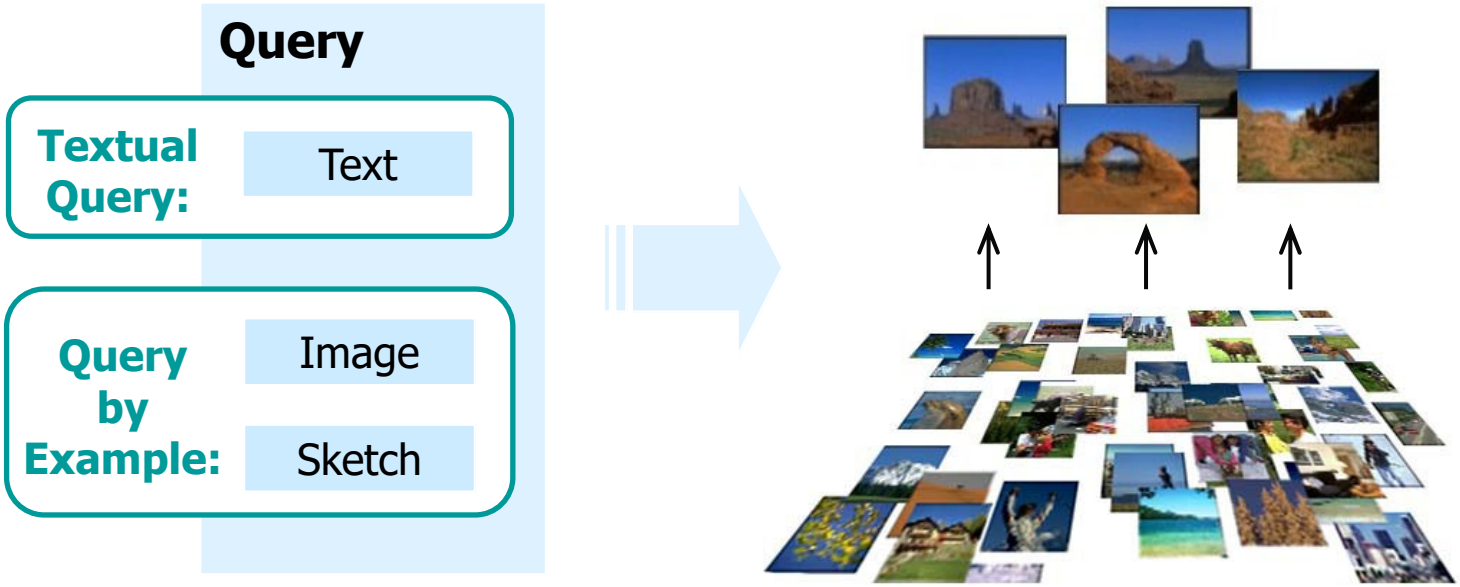
- Audio – 1D data
- Image – 2D data
- Video – 3D data
- 3D geometry models – 3D data
- Motion capture data – 1D or multi-dimensional data

Image Retrieval

Here, we focus on image retrieval.

There are two common techniques for image retrieval:

- Annotation-Based Image Retrieval
- Content-Based Image Retrieval



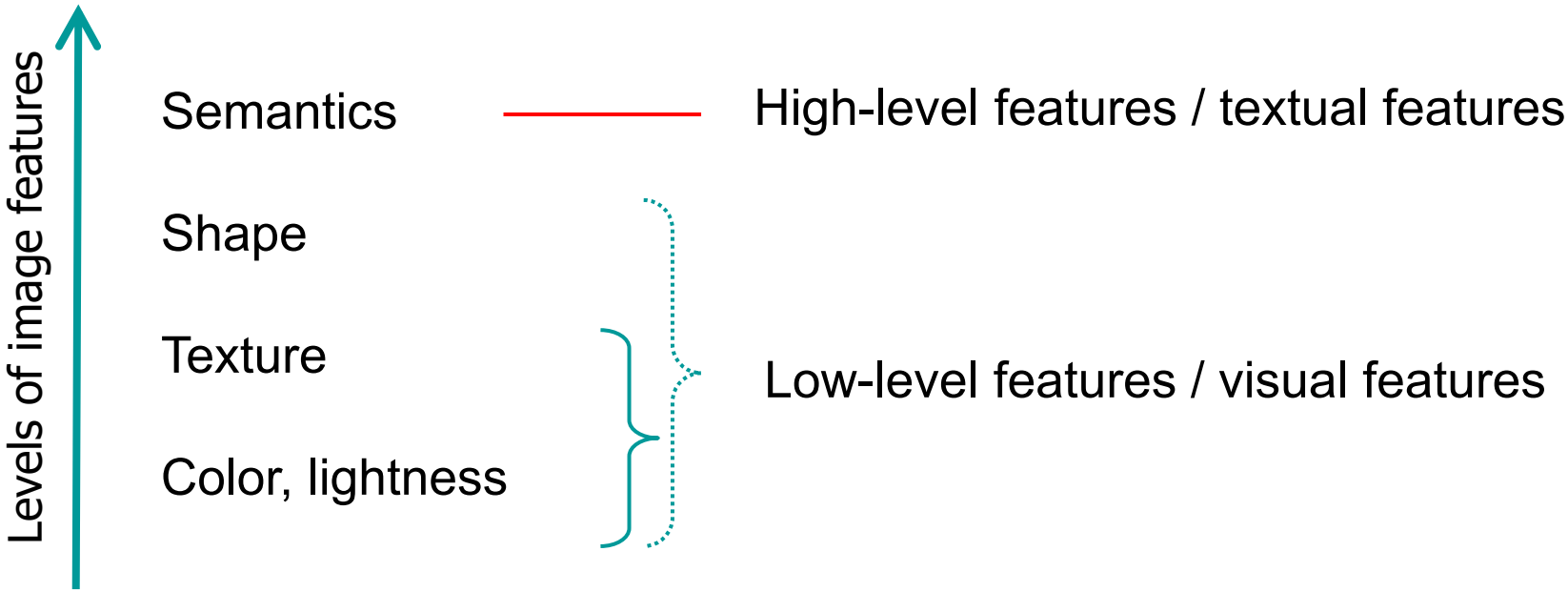
	Annotation-based retrieval	Content-based retrieval
+	Traditional text search methods can be used Search results can match with image semantics	Automatic index construction Less subjective indices
−	Typically requiring manual annotation Not everything can be described in text Manual annotations are subjective	There is a semantic gap Querying by example may not always be convenient to the user

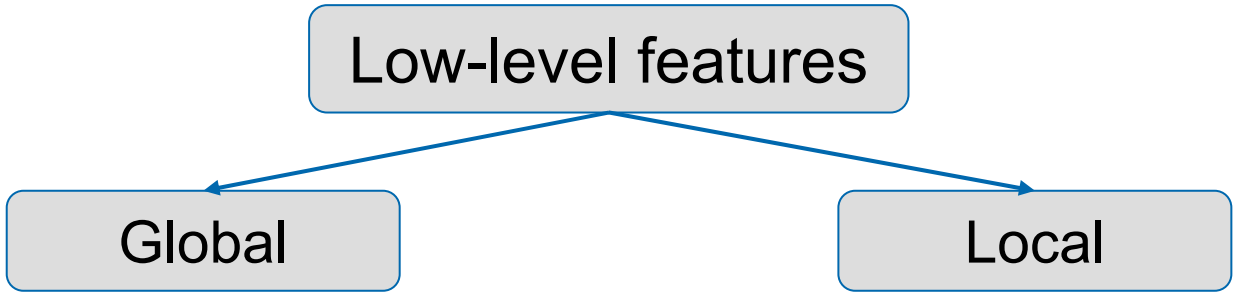
Why do we want content-based image retrieval?

- Huge amount of images in the Internet – not possible to annotate each of them.
- Not everything can be described in text.
- Not everything is described in text.
- Human annotations are subjective.
- Hence, we prefer an automatic, more objective way to describe an image.

Image Features

To describe (or to summarize) an image for content-based retrieval, we need to extract features to represent it. These features can then be used for matching with the features of other images.





Describes the whole image:

- average intensity
- average amount of red
- …

All pixels of the image are considered and processed.

Describes a region of the image:

- average intensity for the left upper part
- average amount of red in the center of the image
- …

Need to first divide the image into segments. Pixels of one segment are processed to extract the features.

Example 1: Comparing two images using pixel-by-pixel differences (low-level)

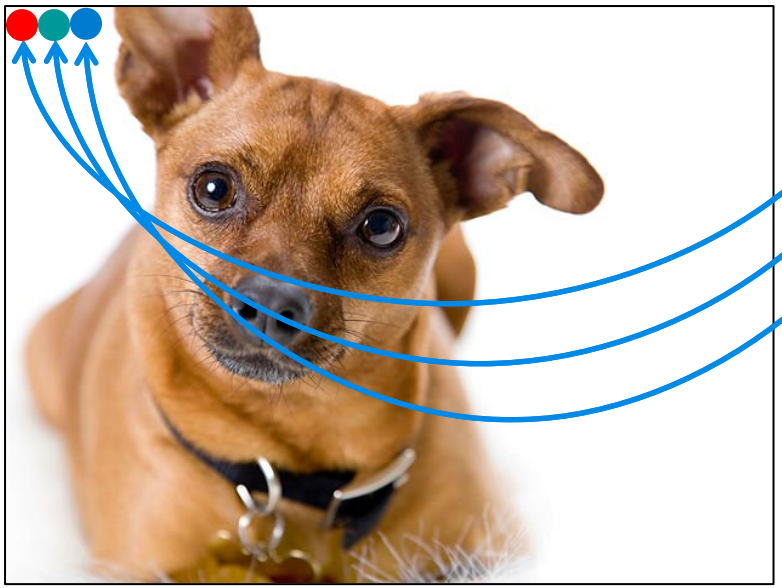


Image P

Image Q

$$Pixelwise\ Similarity = \sum_{i=0}^{All\ pixels} |P_i - Q_i|$$

Image P

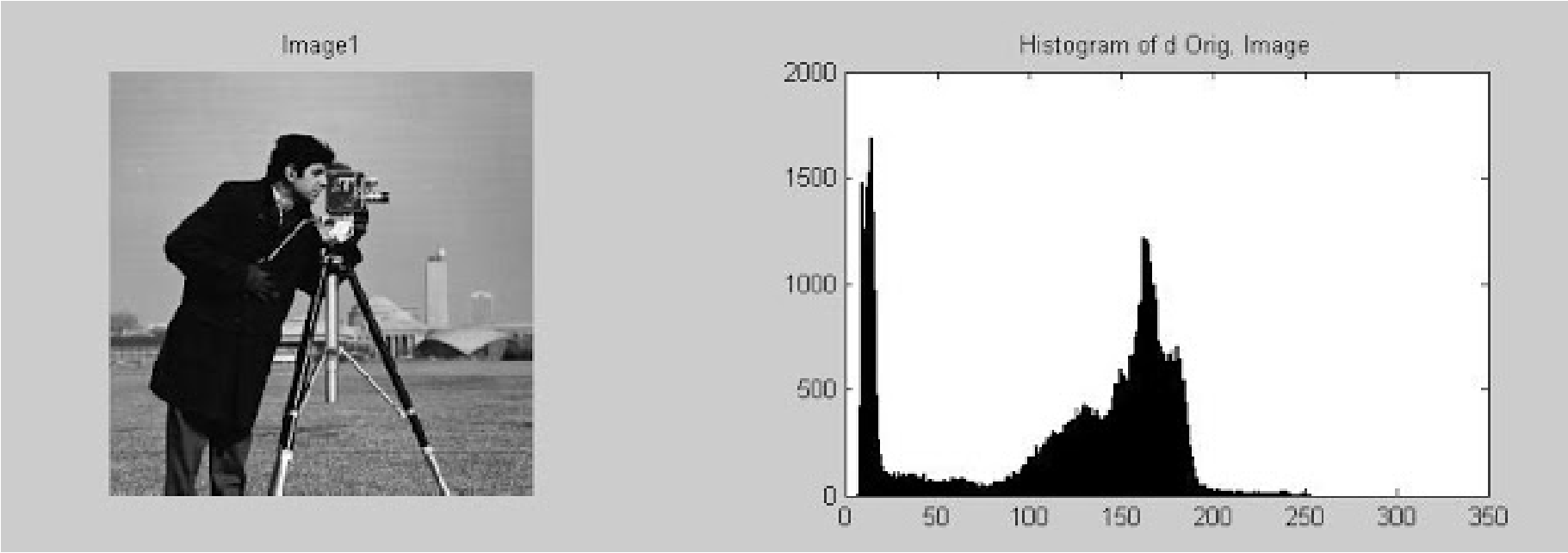
Image Q

$$Pixelwise\ Similarity = \sum_{i=0}^{All\ pixels} |P_i - Q_i|$$

This method does not work on these two images!

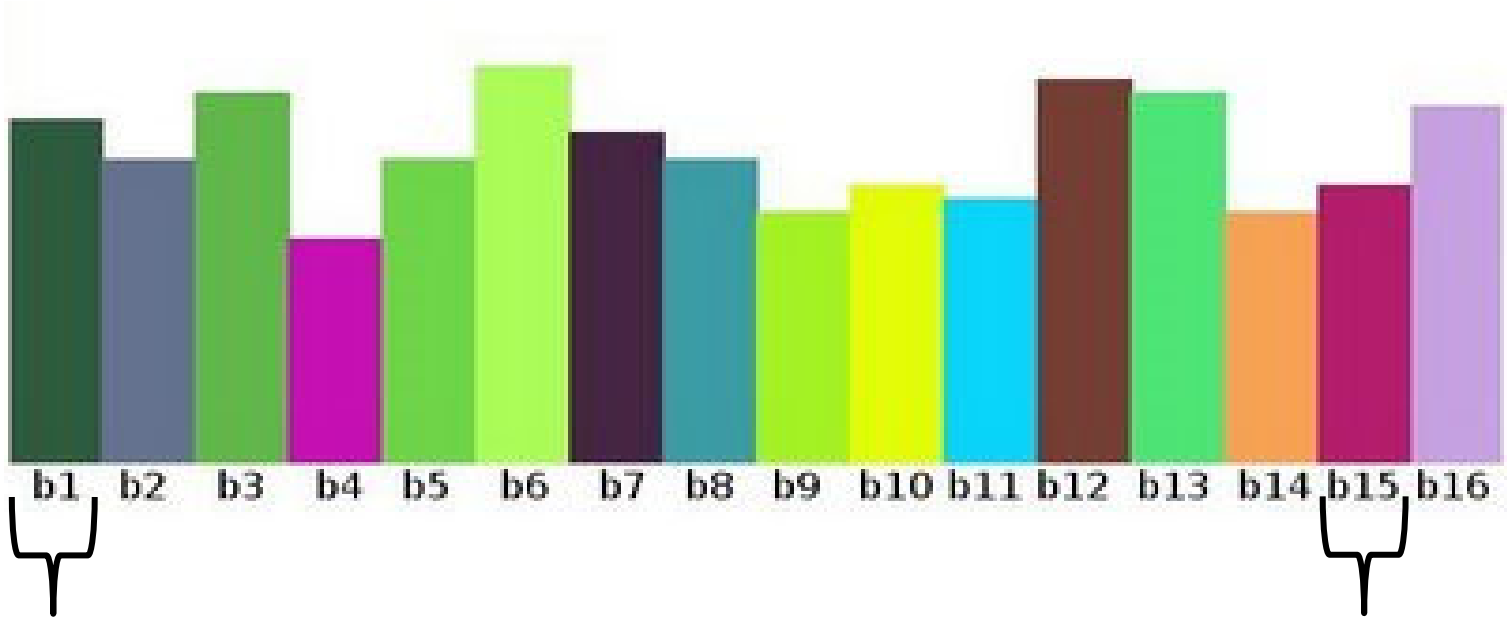
Example 2: Comparing two images using histograms (low-level)

- The histogram of a greyscale image.



a histogram with 256 values (a 256-bin histogram)

- Quantized histogram



16-bin histogram

- The histograms of a color image:

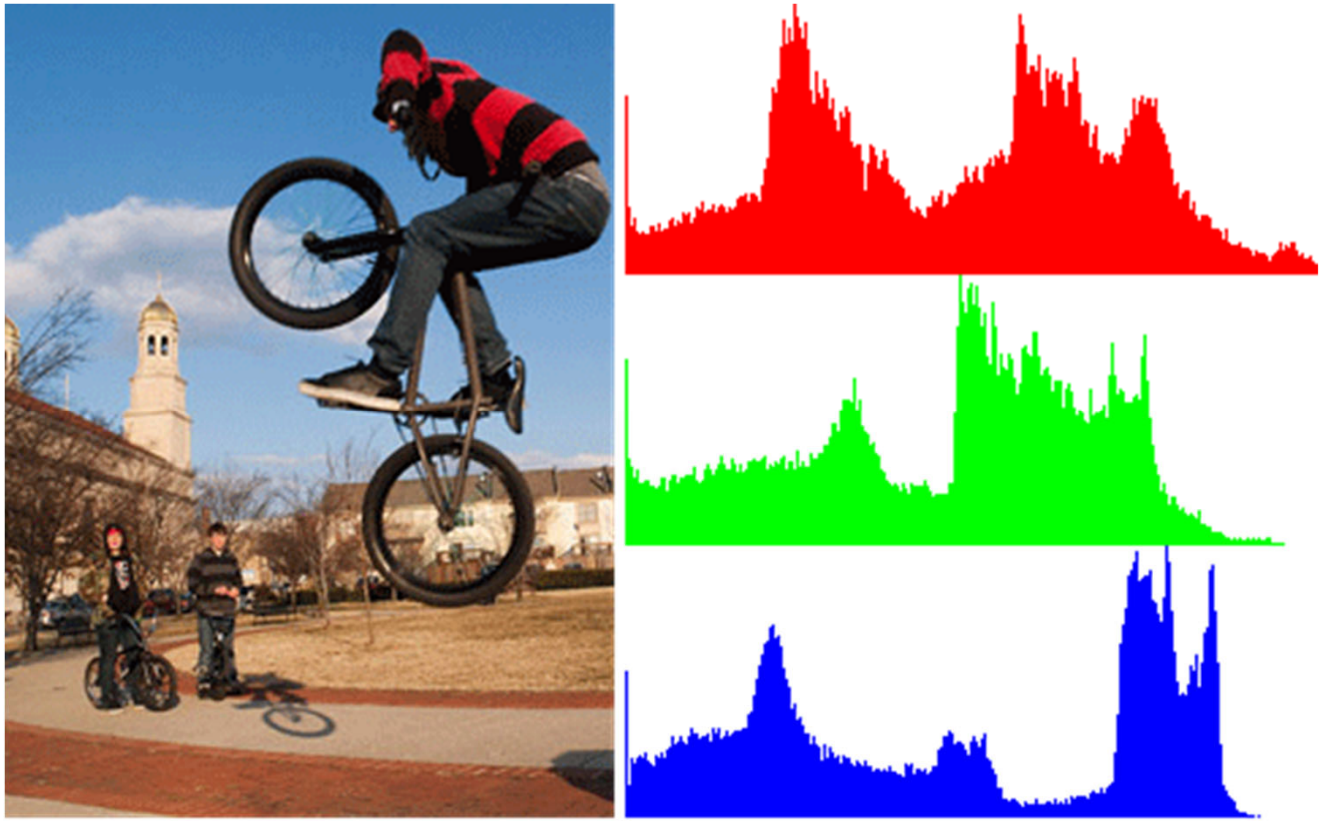


Image P

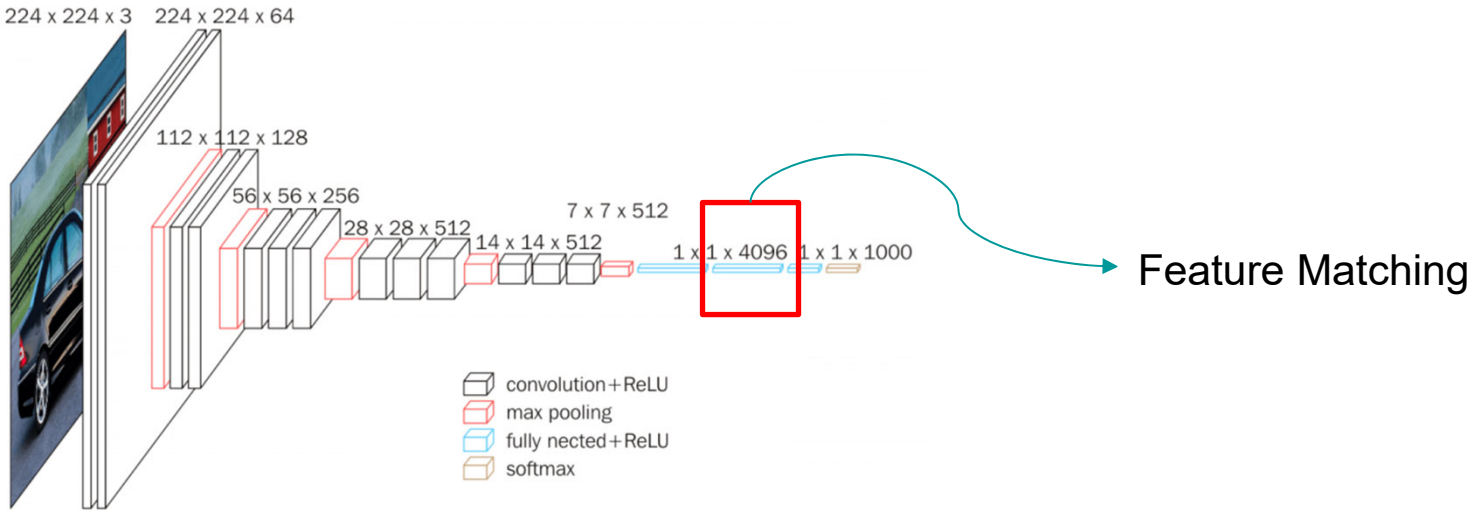
Image Q

The histograms of these two images are the same!

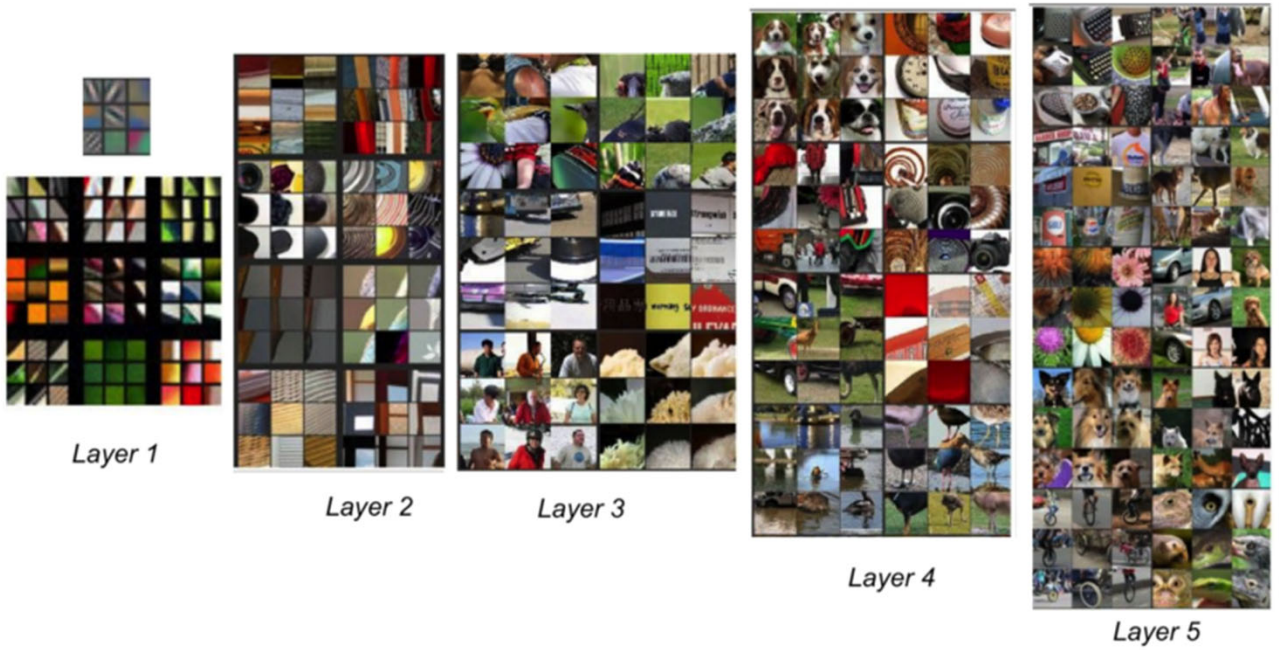
$$Histogram\ Similarity = \sum_{i=0}^{All\ bins} |P\_H_i - Q\_H_i|$$

Example 3: Comparing two images using high-level Semantics

- In the past, it was difficult to automatically obtain high-level features/semantics.
- With deep-learning models, it is now possible to extract high-level features from images directly for matching.



- A deep-learning model is typically composed of many layers. Each layer attempts to extract some features from the input image.



The above diagram shows an example. Starting from the first layer, it learns simple features such as edges. In the next layer, it learns features that are made up of edges such as shapes. In the following layer, it learns features made up of shapes such as smaller components of objects. As we go deeper, it learns more and more complex structures that cover larger extents.