PES
UNIVERSITY

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

Content Based Visual Information Retrieval

Metadata Searching

- Text-based image retrieval (TBIR) systems are focused on text connected with the images or in relation to the images. For text-based image retrieval, one typically uses metadata to describe images.

- Metadata can be defined as data about data.

- Image metadata can be of different kinds, e.g., tags, keywords, and descriptors of relevance for the image.

- This includes data added by the capturing device, e.g., time/date and GPS coordinates, keywords manually added by individual users to describe the image (tags) or automatic image annotations added by the image retrieval system to simplify search and indexing. The latter is usually referred to as Auto- annotations or linguistic indexing.

Metadata Searching

- Images are first annotated with text and then searched using a text-based approach from traditional database management systems.

- However, since automatically generating descriptive texts for a wide spectrum of images are not feasible, most text-based image retrieval systems require manual annotation of images.

- Obviously, annotating images manually is a cumbersome and expensive task for large image databases and is often subjective, context-sensitive, and incomplete. As a result, it is difficult for the traditional text-based methods to support a variety of task-dependent queries.

Metadata Searching

- The text-based systems are fast as the string matching is computationally less time-consuming process.
- However, it is sometimes difficult to express the whole visual content of images in words and TBIR may end up in producing irrelevant results.
- Manually added tags are keywords added to the image by individual users. In theory, they represent the individuals' natural perception of the image.
- Manually added tags can be very helpful for the retrieval system if available.

Metadata Searching

- One concern that remains in dealing with manually added tags is the subjective nature of human tagging introduces variability, as individual users may possess different perceptions and employ diverse tags to describe the same images.

- For example, an image consisting of grass and flowers might be labeled as either “grass” or “flower” or “nature” by different people.

- There is a high probability of error occurrence during the image-tagging process when the database is large.

Metadata Searching

- As a result, text-based image retrieval cannot achieve high level of efficiency and effectiveness.

- For finding the alternative way of searching and overcoming the limitations imposed by TBIR systems, more intuitive and user-friendly

Metadata Searching

Advantages:

- Precise Retrieval
- Improved Search Relevance
- Facilitates multimodal search

Disadvantages:

- Manual Annotations are time consuming
- Annotations may be subjective
- Not Scalable

References

- “Content-Based Image Retrieval : Ideas, Influences, and Current Trends” - Vipin Tyagi, Springer Nature Singapore Pte Ltd., 2017.

PES UNIVERSITY

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

PES UNIVERSITY

Query by Example

User Perspective

- The user interface is a crucial component of a CBVIR system. Ideally such interface should be simple, easy, friendly, functional, and customizable.
- It should provide integrated browsing, viewing, searching, and querying capabilities in a clear and intuitive way.
- This integration is extremely important, since it is very likely that the user will not always stick to the best match found by the query/search engine.
- More often than not users will want to check the first few best matches, browse through them, preview their contents, refine their query, and eventually retrieve the desired image or video segment.

User Perspective

- Specifying what kind of images a user wishes to retrieve from the database can be done in many ways.
- Several querying mechanisms have been created to help users define their information need.
- Commonly used query formations are as follows:
  - Category Browsing
  - Query by Concept
  - Query by Sketch
  - **Query by Example**

Category Browsing, Query by Concept and Query by Sketch

Category browsing: Category browsing is to browse through the database according to the category of the image.

Query by Concept: Query by concept is to retrieve images according to the conceptual description associated with each image in the database.

Query by Sketch: Query by sketch allows user to draw a sketch of an image with a graphic editing tool provided either by the retrieval system or by some other software. Queries may be formed by drawing several objects with certain properties like color, texture, shape, sizes, and locations. In most cases, a coarse sketch is sufficient, as the query can be refined based on retrieval results.

Query by Example

- Query by example allows the user to formulate a query by providing an example image.
- The system converts the example image into an internal representation of features.
- Images stored in the database with similar features are then searched.

Query by Example

- Query by example can be further classified into:
  Query by external image example, if the query image is not in the database, and
  Query by internal image example, if otherwise.

- For query by internal image, all relationships between images can be pre-computed.

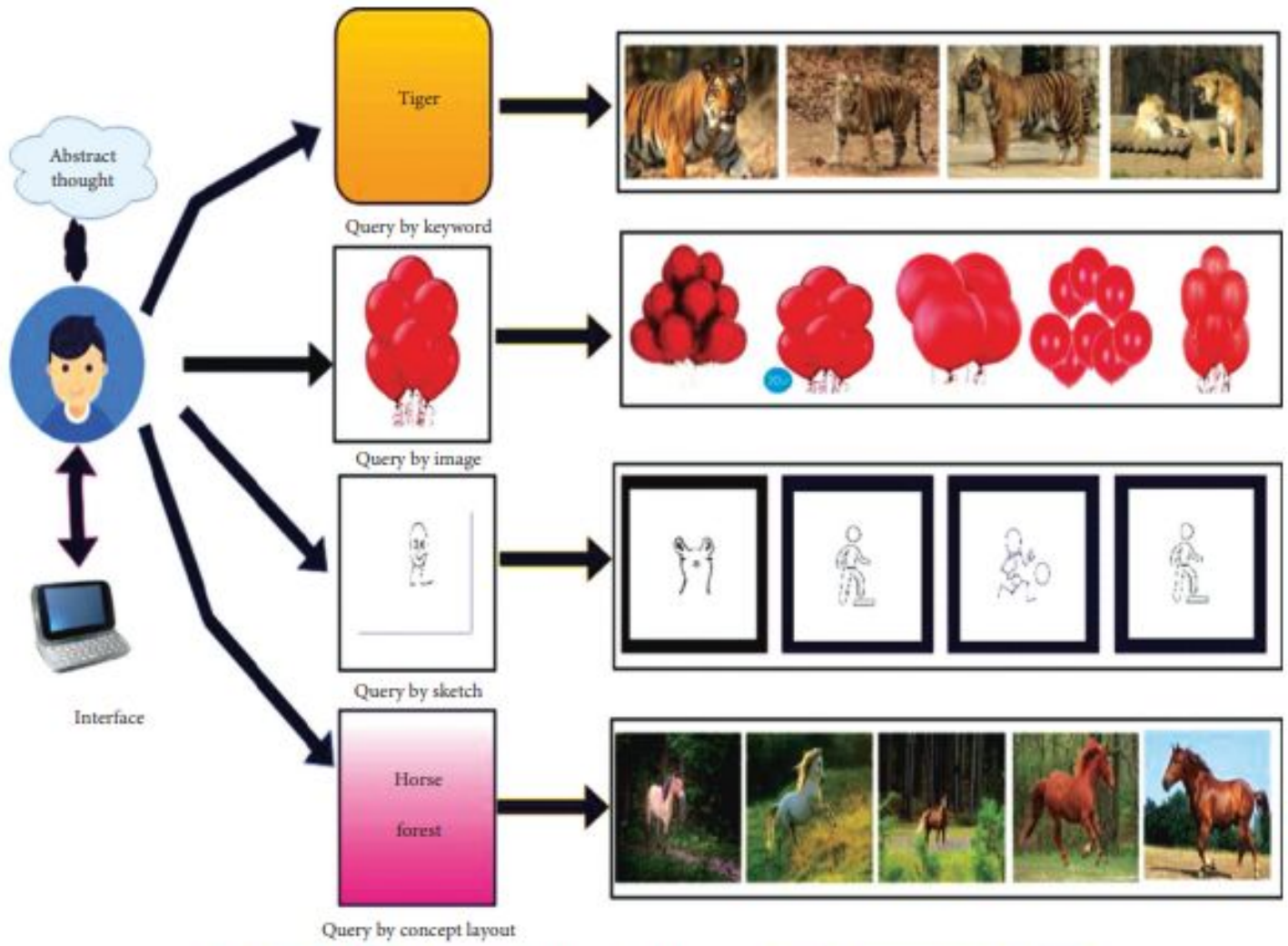
Query by Example

- The main advantage of query by example is that the user is not required to provide an explicit description of the target, which is instead computed by the system.
- It is suitable for applications where the target is an image of the same object or set of objects under different viewing conditions.
- Most of the current systems provide this form of querying.

Query by Example

- Another modification that can be made is by taking a group of examples – **Query by Group Example**
- Query by group example allows user to select multiple images.
- The system will then find the images that best match the common characteristics of the group of examples.
- In this way, a target can be defined more precisely by specifying the relevant feature variations and removing irrelevant variations in the query.
- In addition, group properties can be refined by adding negative examples. Many recently developed systems provide both query by positive and negative examples

Query by Example



References

- “Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review” - Afshan Latif

Semantic Retrieval

- So far we have discussed about low-level features of images and the queries that a user can enter to retrieve relevant images.
- The queries are always of the high-level features such as keywords, text, images, etc.
- However, in general there is no direct link between high-level concepts and low-level features.

Semantic Retrieval

- Extensive experiments on CBIR systems show that low-level contents often fail to describe the high-level semantic concepts in user's mind.

- Therefore, the performance of CBIR is still far from user's expectations. Scientists have mentioned three levels of queries in CBIR.

Semantic Retrieval

- Level 1:
  - Retrieval by primitive features such as color, texture, shape, or the spatial location of image elements.
  - Typical query is a query by example, “find pictures like this.”

- Level 2:
  - Retrieval of objects of given type identified by derived features, with some degree of logical inference.
  - For example, “find a picture of a flower.”

- Level 3:
  - Retrieval by abstract attributes, involving a significant amount of high-level reasoning about the purpose of the objects or scenes depicted. This includes retrieval of named events, of pictures with emotional or religious significance, etc.
  - For example, “find pictures of a joyful crowd.”

Semantic Retrieval

- Levels 2 and 3 together are referred to as semantic image retrieval and the gap between Levels 1 and 2 as the semantic gap.

- Users in Level 1 retrieval are usually required to submit an example image or sketch as query.

- Semantic image retrieval is more convenient for users as it supports query by keywords or by texture.

- Therefore, to support query by high-level concepts, a CBIR system should provide full support in bridging the “semantic gap” between numerical image features and the richness of human semantics

Semantic Retrieval

- The human perception of visual contents is strongly associated to high-level, semantic information about the scene.

- Current Computer Vision techniques work at a lower level (as low as individual pixels).

- CBVIR systems that rely on low-level features only can answer queries such as:
  - Find all images that have 30% of red, 10% of orange and 60% of white pixels, where orange is defined as having a mean value of red = 255, green = 130, and blue = 0.
  - Find all images that have a blue sky above a green grass.
  - Find all images that are rotated versions of this particular image.

Semantic Retrieval

- In general case, the user is looking for higher-level semantic features of the desired image, such as "a beautiful rose garden", "a batter hitting a baseball", or "an expensive sports car".

- There is no easy or direct mapping between the low-level features and the high-level concepts.

- The distance between these two worlds is normally known as “semantic gap.”

Semantic Retrieval

- Currently there are two ways of minimizing the semantic gap:
  - The first consists of adding as much metadata as possible to the images, which was already discussed and shown to be impractical.
  - The second suggests the use of rich user interaction with relevance feedback combined with learning algorithms to make the system understand and learn the semantic context of a query operation

References

- “Content-Based Image Retrieval : Ideas, Influences, and Current Trends” - Vipin Tyagi, Springer Nature Singapore Pte Ltd., 2017. - Chapter 1

PES UNIVERSITY

Machine Learning Approaches

- The main aspects in designing a CBVIR system are :
    - feature extraction and representation,
    - dimension reduction and multidimensional indexing,
    - extraction of image semantics, and
    - design of user relevance feedback mechanisms.

- A variety of machine learning and deep learning techniques are being employed and developed to improve these systems.

Feature Extraction and Representation

- CBVIR systems should be able to automatically extract visual features that are used to describe the contents of an image or video clip.
- Examples of such features include color, texture, size, shape, and motion information.
- In specific contexts the process of feature extraction can be enhanced and/or adapted to detect other, specialized attributes, such as human faces or objects.
- Because of perception subjectivity, there is no best representation for a given feature.
- The color information, for instance, can be represented using different color models (e.g., RGB, HSV, YCbCr) and mathematical constructs, such as color histograms, color moments, color sets, color coherence vectors, or color correlograms. In a similar way, texture can be represented using co-occurrence matrix , Tamura texture features or Wavelets, to name just a few.

Dimension Reduction and Multidimensional Indexing

- The extracted features are grouped into some suitable data structure or mathematical construct (e.g., a normalized feature vector), and suitable metrics (e.g., Euclidean distance) are used to measure the similarity between an image and any other image.

- One of the techniques commonly used dimension reduction is principal component analysis (PCA).

- It is an optimal technique that linearly maps input data to a coordinate space such that the axes are aligned to maximally reflect the variations in the data.

- The QBIC system uses PCA to reduce a 20-dimensional shape feature vector to two or three dimensions.

Dimension Reduction and Multidimensional Indexing

- In addition to PCA, many researchers have used Karhunen–Loeve (KL) transform to reduce the dimensions of the feature space.

- Although the KL transform has some useful properties such as the ability to locate the most important subspace, the feature properties that are important for identifying the pattern similarity may be destroyed during blind dimensionality reduction.

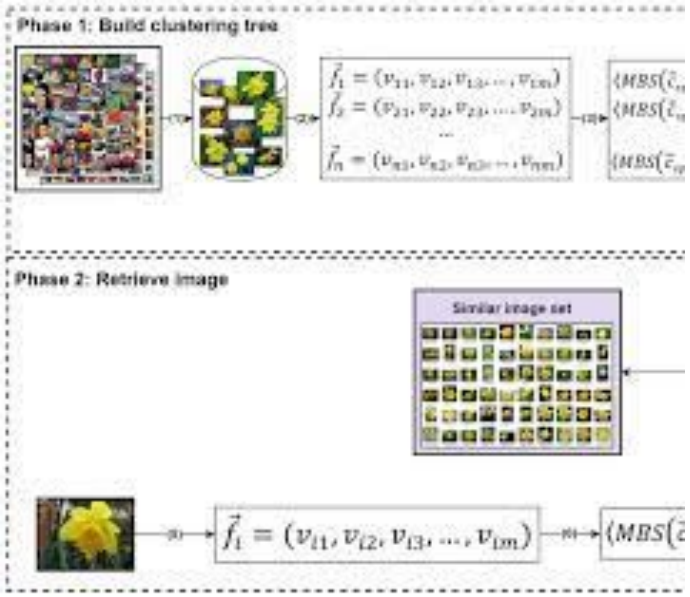
Apart from PCA and KL transformation, neural network has also been demonstrated to be a useful tool for dimension reduction of features

Dimension Reduction and Multidimensional Indexing

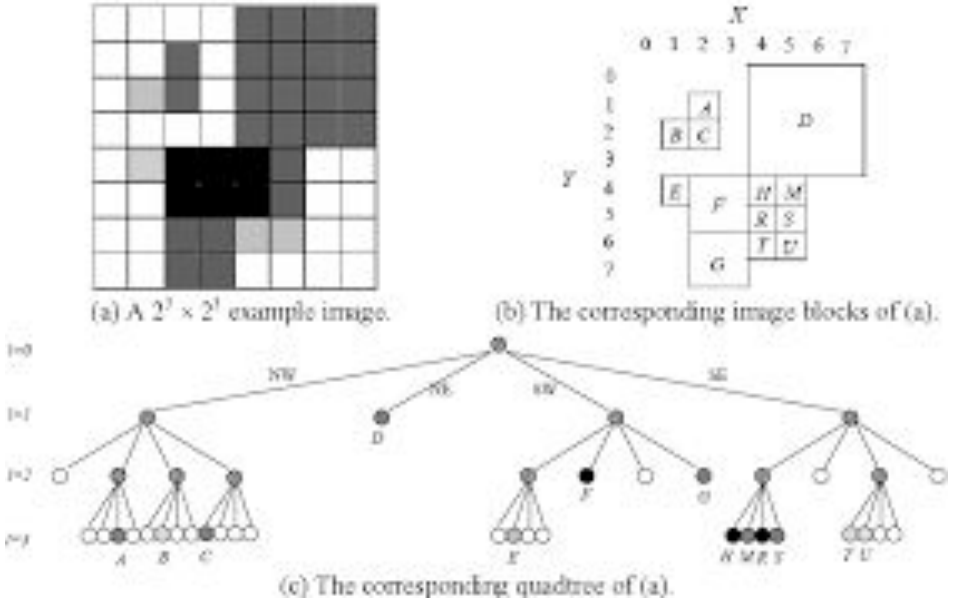
- After dimension reduction, the multidimensional data is indexed.
- A number of approaches have been proposed for this purpose, including R-tree (particularly, R*- tree), linear quad-trees, K-d-B tree, and grid files.
- Most of these multidimensional indexing methods have reasonable performance for a small number of dimensions (up to 20), but explore exponentially with the increasing of the dimensionality and eventually reduce to sequential searching.

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

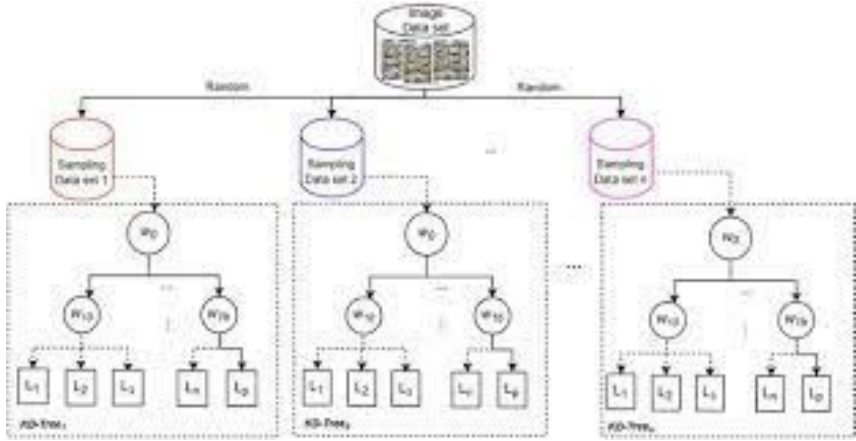
Dimension Reduction and Multidimensional Indexing



R-tree



linear quad-trees



K-d-B tree

Common techniques

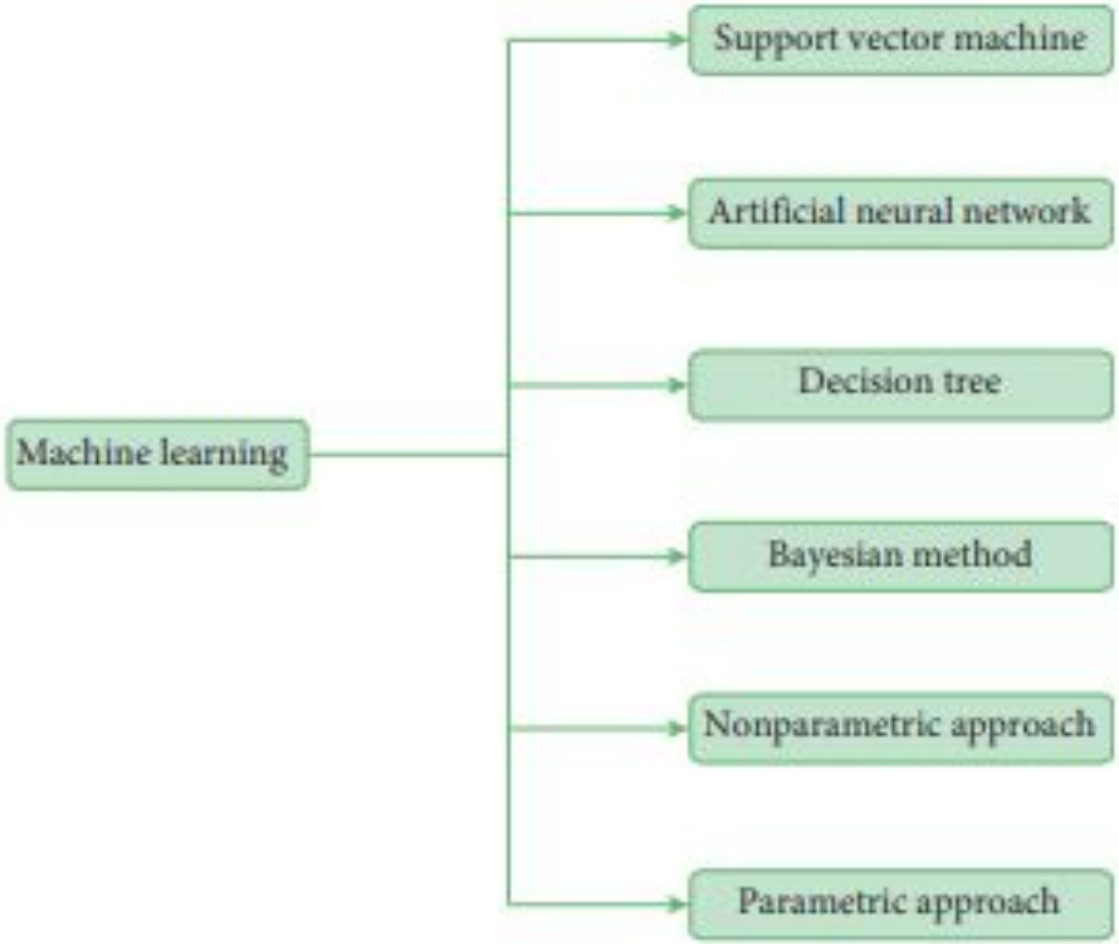


FIGURE 8: An overview of basic machine learning techniques for CBIR [1, 76].

Approach 1: TF-IDF CNN

PES UNIVERSITY

- Due to most prominent results and with a great performance of the deep convolutional neural network (CNN), a novel term frequency-inverse document frequency (TF-IDF) using as description vector the weighted convolutional word frequencies based on CNN is proposed for CBIR.

- The tf-idf weighting scheme for document retrieval is described as follows: Considering a vocabulary of ***n*** terms, the tf-idf is a **n-dimensional vector** that is calculated as the element-wise product of the tf and idf vectors. That is, a document is represented by a n-dimensional vector, $\boldsymbol{x}_d$, where each of its ***n*** dimensions is given by:

$$[\mathbf{x}_d]_t = F_{tf}(t, d) \times F_{idf}(d, t, D),$$

d is the index of a document,
t is the index of a certain term of the vocabulary of n terms,
D is the corpus,
$\boldsymbol{F_{tf}(t, d)}$ is the function which computes the frequency of term t in the document d, and
$\boldsymbol{F_{idf}(d, t, D)}$ is the function which computes the inverse document frequency

Approach 1: TF-IDF CNN

- The term frequency (***tf***) part produces a ***n***-dimensional vector where each dimension is given by the frequency of occurrence of each term ***t*** in the document ***d***, that is how many times the term ***t*** appears in the document ***d***.
- The inverse document frequency (***idf***) part provides information about the importance of each term, in the sense that terms that appear in many different documents are less informative, and hence of less importance, as compared to those that appear rarely.
- Thus, the idf part produces a ***n***-dimensional vector, ***r***, where each dimension is given by

$$[\mathbf{r}]_t = \log \frac{|\, D \,|}{|\, d \in D : F_{tf}(t, d) \neq 0 \,|},$$

where $|\cdot|$ is the cardinality of a set.

Approach 1: TF-IDF CNN

The filters of the convolutional layers of a pretrained CNN model have been trained to recognize specific visual features in the input image

consider these features as the convolutional words, and the learned filters as the convolutional word detectors

For an input image each filter is activated when it sees a certain visual feature, e.g. edges, blobs, shapes, etc., in the first layers or a visual concept, e.g, face, car, etc., in the last layers, and thus the activations of the filters reveal the degree of presence of the convolutional word that learned during the training procedure

Approach 1: TF-IDF CNN

To translate this to the tf-idf technique, the activations of each filter correspond to the tf part.

Since the activations of each filter output a 2-dimensional activation map of the responses of the filter at every position of the input image, they take the maximum activation value, in order to assign a unique activation value to every filter, considering that the degree of presence of the specific visual pattern is equal to the maximum degree of presence.

This leads to loss of the spatial information that capture the convolutional layers. Thus, the vocabulary size is equal to the number of the learned filters (or the convolutional neurons) of the utilized convolutional layer or the number of the neurons if we obtain the responses of the fully connected layers.

Approach 1: TF-IDF CNN

Since the range of the activation value of a neuron varies through the network layers, we apply a normalization step in order to ensure that the activation value of each neuron of the network lies in the range of [0, 1]. That is, for each image of the dataset, they divide each activation value of the neurons of a certain layer by the maximum activation value of the neurons of this layer over the entire dataset.

Approach 1: TF-IDF CNN

An issue arising on materializing the idf part is that the utilized normalized activations take values between 0 and 1, while in the standard tf-idf technique in text domain a word either exists or not. Thus, we need to investigate when a neuron is considered as activated or not.

The first approach defines a threshold T with value between 0 and 1. If the activations are over this threshold value, the neurons are considered as activated. That is:

$$x_n^i = \begin{cases} 1, & \text{if } a_n^i > T \\ 0, & otherwise \end{cases}$$

Approach 1: TF-IDF CNN

The second approach defines a percentage threshold value for the entire network activation and in each layer we consider as activated the neurons with the maximum activations in a cumulative way until we reach the predefined activation percentage threshold.

$$x_n^i = \begin{cases} 1, & \text{if } a_n^i \text{ in top K\% of activations} \\ 0, & otherwise \end{cases}$$

Approach 1: TF-IDF CNN

Finally, the third approach estimates the importance weight of each neuron based on the standard deviation of the normalized activation values of the specific neuron to the entire dataset. More specifically, the weight w_n of each neuron n is computed as follows:

$$w_n = \sqrt{\sum_i (a_n^i - \mu_n)^2},$$

where µn = i ai n is the mean value of the activations of the neuron n in the entire dataset I , and ai n is the normalized activation of the neuron n, for an image i ∈ I. Thus, the weighted description is given by the following equation:

$$x_n^i = w_n \times a_n^i$$

Approach 1: TF-IDF CNN

- The proposed weighting scheme tries to use the statistical behavior of the neurons in order to estimate its importance.
- That is, a neuron that is consistently activated with similar values throughout the entire dataset, and thus it has a very small standard deviation in the activation value, has limited importance in the retrieval task.
- The standard deviation of the activation resamples the idf value of the neuron.
- On the contrary, a neuron that has a large standard deviation is more informative since it has different behavior across the dataset and thus it can be used for discriminating the images. As previously, the tf term is estimated by the normalized activation of the specific neuron n to the specific image i.

Approach 1: TF-IDF CNN

PES UNIVERSITY

Pyramid-based approach

- In order to partially preserve spatial information from the activations of the convolutional layers they propose the following approach. They divide the activation map into S sections.

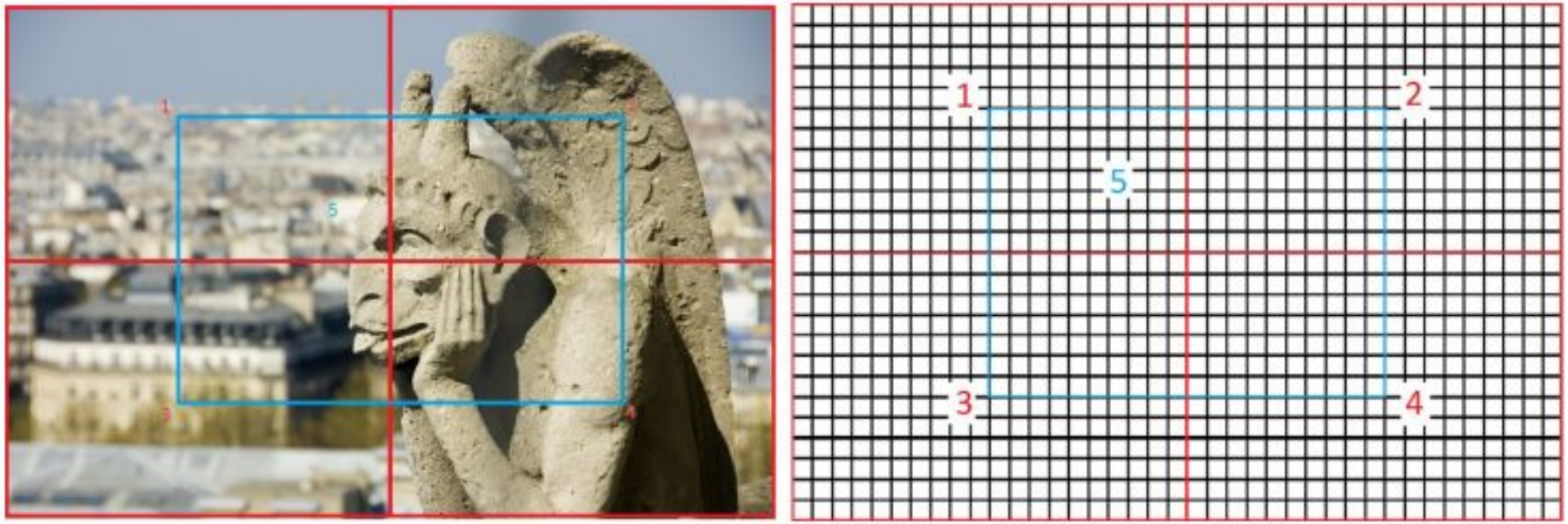
- Then they treat each of the resulting sections as a separate image. That is, we apply the proposed tf-idf approach to each of them. Thus, instead of considering the maximum activation value over the whole image, we consider the maximum activation value over each section.

- This value informs us about the presence of the convolutional words on the corresponding section of the input image.

Approach 1: TF-IDF CNN

- Regarding the idf part, which provides information about the importance of the certain convolutional word, they maintain the same weights calculated for the whole image.
- That is, they extract more than one value for a convolutional neuron for an input image in order to partially preserve the position of the detected convolutional word.
- Hence, they produce S times bigger description for the whole image. This allows us to decide not only whether two images contain the same convolutional words, but also if these words are approximately in the same position on the image.
- It is clear that the more sections lead to more detailed detections on the image, however this also renders the comparison between two images more difficult.

Approach 1: TF-IDF CNN



(a) Five sections in an input image (b) Five sections in the activation map

Fig. 1 Division of the activation maps into sections

Approach 1: TF-IDF CNN

- More specifically, the initial image is divided into five sections (four non overlapping sections whose width and height are equal to the half-width and half-height of the full image, respectively, while the fifth section positioned in the center of the image is of equal dimensions, and has 25% overlap with each of the other four sections), and correspondingly the activation map for a certain neuron is divided into the same sections.

- Then, the maximum activation values for each of the five sections is derived, and the tf-idf scheme is applied. The final representation is derived by concatenating the five representations.

Approach 1: TF-IDF CNN

- Kondylidis *et al* utilize a commonly used CNN model, that is the VGG-16, to apply the proposed tf-idf scheme, and subsequently utilize their recently published fully convolutional model, optimized toward image retrieval in a fully unsupervised fashion.
- They first utilize the VGG 16-layer model, trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 to classify 1,000 ImageNet classes, since it is a commonly used baseline model. The model consists of sixteen trained neural layers; the first thirteen are convolutional and the remaining three are fully connected.

- Max-pooling layers follow the second, forth, seventh, tenth, and thirteenth convolutional layers, while the ReLU non-linearity (f (x) = max(0, x)) is applied to every convolutional and fully connected layer, except the last fully connected layer. The output of the last fully connected layer is a distribution over 1000 ImageNet classes. The softmax loss is used during the training. They use the VGG 16 layer model to directly extract the representations from certain layers in order to apply the proposed tf-idf approach.

Approach 1: TF-IDF CNN

- The CNN features have a hierarchical nature. That is, convolutional neurons at the first levels are meant to detect low level local concepts like edges and corners.
- The next levels are able to detect more complicated patterns like basic shapes, which are based on the detections of previous convolutional levels, which can be considered as combination of convolutional words; the mid level local concepts.
- The last convolutional layers are able to detect even more complicated patterns, close to ones that humans detect, like hands, faces or even cars; the high level local concepts.
- Finally, the activations of neurons of the fully connected layers are based on the detection of combinations of patterns and the reason they activate cannot be pointed directly on some pattern on the input image.
- Neurons of fully connected layers produce only one output for the whole image, and they are able to detect high level global concepts.

Approach 1: TF-IDF CNN

PES UNIVERSITY

Fully Unsupervised Model

- The Fully Unsupervised (FU) model, is a recent state-of-the-art fully convolutional model, which is retrained on the modified CaffeNet model, n order to produce more efficient and compact image representations for the retrieval task.

- More specifically, in the Fully Unsupervised (FU) approach, the goal is to amplify the primary retrieval presumption that the relevant image representations are closer to the certain query representation in the feature space. The rationale behind this approach is rooted to the cluster hypothesis which states that documents in the same cluster are likely to satisfy the same information need.

- That is, the pre-trained CNN model is retrained on the given dataset, aiming at maximizing the cosine similarity between each image representation and its n nearest representations, in terms of cosine distance.

Approach 1: TF-IDF CNN

The set of N images to be searched is denoted by $\mathcal{I} = \{\mathbf{I}_i, i = 1, \ldots, N\}$, their corresponding feature representations emerged in the L layer by $\mathcal{X} = \{\mathbf{x}_i, i = 1, \ldots, N\}$, and by μ^i the mean vector of the $n \in \{1, \ldots, N-1\}$ nearest representations to $\mathbf{x}_i$, denoted as $\mathcal{X}^i = \{\mathbf{x}_l^i, l = 1, \ldots, N-1\}$. That is,

$$\mu^i = \frac{1}{n} \sum_{l=1}^{n} \mathbf{x}_l^i$$

The optimization problem to be solved is:

$$\max_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J} = \max_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{\mathbf{x}_i^\mathsf{T} \mu^i}{\|\mathbf{x}_i\| \, \|\mu^i\|}$$

Approach 1: TF-IDF CNN

The above optimization problem is solved using gradient descent. The first-order gradient of the objective function $\mathcal{J}$ is given by:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{x}_i}\left(\sum_{i=1}^{N} \frac{\mathbf{x}_i^{\mathsf{T}} \mu^i}{\|\mathbf{x}_i\| \, \|\mu^i\|}\right) = \frac{\mu^i}{\|\mathbf{x}_i\| \, \|\mu^i\|} - \frac{\mathbf{x}_i^{\mathsf{T}} \mu^i}{\|\mathbf{x}_i\|^3 \, \|\mu^i\|}\mathbf{x}_i$$

The update rule for the v-th iteration for each image can be formulated as:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta\left(\frac{\mu^i}{\|\mathbf{x}_i^{(v)}\| \, \|\mu^i\|} - \frac{\mathbf{x}_i^{(v)\mathsf{T}} \mu^i}{\|\mathbf{x}_i^{(v)}\|^3 \, \|\mu^i\|}\mathbf{x}_i^{(v)}\right), \quad \mathbf{x}_i \in \mathcal{X}$$

A normalization step has been introduced as:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta\|\mathbf{x}_i^{(v)}\| \, \|\mu^i\|\left(\frac{\mu^i}{\|\mathbf{x}_i^{(v)}\| \, \|\mu^i\|} - \frac{\mathbf{x}_i^{(v)\mathsf{T}} \mu^i}{\|\mathbf{x}_i^{(v)}\|^3 \, \|\mu^i\|}\mathbf{x}_i^{(v)}\right), \quad \mathbf{x}_i \in \mathcal{X}$$

Approach 1: TF-IDF CNN

- Hence, using the above representations as targets in the layer of interest, a regression task is formulated for the neural network, which is initialized on the CaffeNet's weights and is trained on the utilized dataset, using back-propagation.

- The Euclidean loss is used during training for the regression task. Thus, the procedure is integrated by feeding the entire dataset into the input layer of the retrained adapted model and obtaining the new representations.

Approach 1: TF-IDF CNN

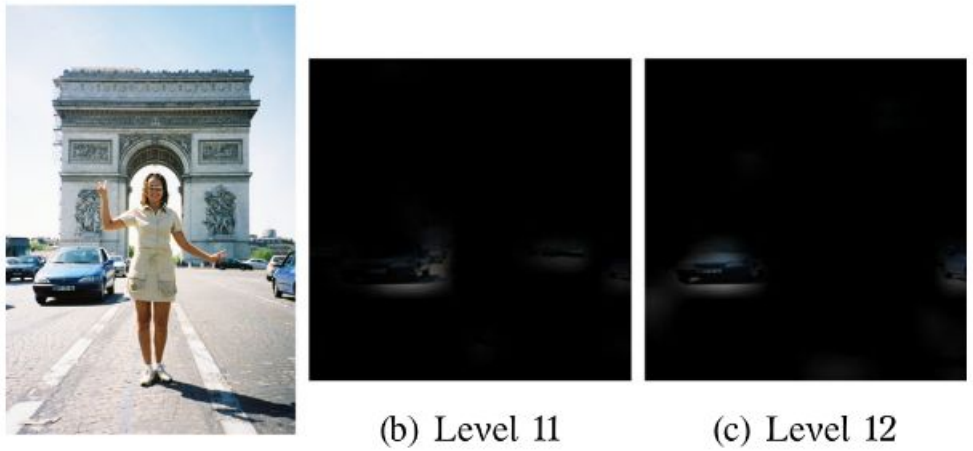
- More specifically, they study the usefulness of the patterns that are detected by the aforementioned layers. Toward this aim, they also propose a novel visualization method, which reveals in which parts of the input image, a convolutional filter was activated, and thus, what patterns has been trained to recognize. The technique of filter visualization is described below.

- Every convolutional neuron takes as input many regions of the input image and outputs an activation value for each of them, forming the activation map.

- The values of the activation maps are normalized as mentioned above in order to belong to the interval [0, 1]. We multiply the RGB values of all pixels of one region with the produced activation value.

- This produces a new image where every region that did not activate the neuron is shaded. We use bicubic interpolation to resize the activation map so that the visual result appears more uniform.

Approach 1: TF-IDF CNN



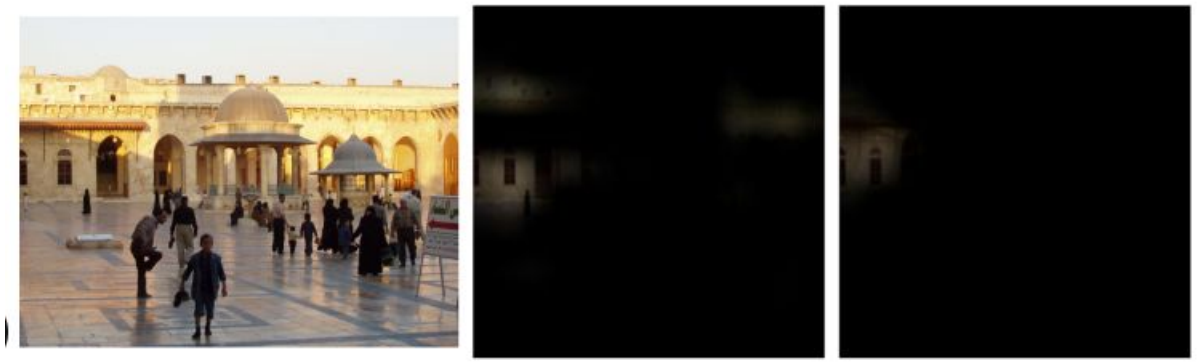
(a) Input Image (b) Level 11 (c) Level 12

Fig. 3 Example of tracing cars

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 13

Fig. 4 Example of tracing human parts for the input image of Fig. 3

Approach 1: TF-IDF CNN



(a) Input Image (b) Level 12 (c) Level 13

Fig. 5 Example of tracing windows

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 12 (e) Level 13

Approach 1: TF-IDF CNN

PES UNIVERSITY

Query Expansion using pseudo relevance feedback

- Query Expansion is a standard, in most cases of negligible cost, technique for accomplishing better retrieval results.
- Concisely, the idea is to re-formulate the initial query, utilizing information deriving from the evaluation of the initial query.
- The majority of CBIR methods include a query expansion step that boosts the retrieval performance.
- On the top of the proposed descriptors they also introduce a simple query expansion method using Pseudo Relevance Feedback.

Approach 1: TF-IDF CNN

That is, they propose to re-issue the top n ranked results from the Initial query as a new query, in order to enhance the original query representation with additional relevant representations, following either the average query expansion scheme or the max one.

The average scheme can be described as follows:

> Let $\mathbf{Q}$ be a certain query image, and $\mathbf{q}$ the corresponding CNN representation using the proposed method. We consider the top k retrieved images and their corresponding CNN representations $\mathbf{x}_j$, $j = 1, \ldots, k$. Then, the new query representation $\mathbf{q}_{new}$ is as follows:

$$\mathbf{q}_{new} = \frac{1}{k+1}\left(\mathbf{q} + \sum_{j=1}^{k} \mathbf{x_j}\right)$$

The max scheme considers as the new query representation the max values of the top k retrieved images in each dimension.

Approach 1: TF-IDF CNN

PES UNIVERSITY

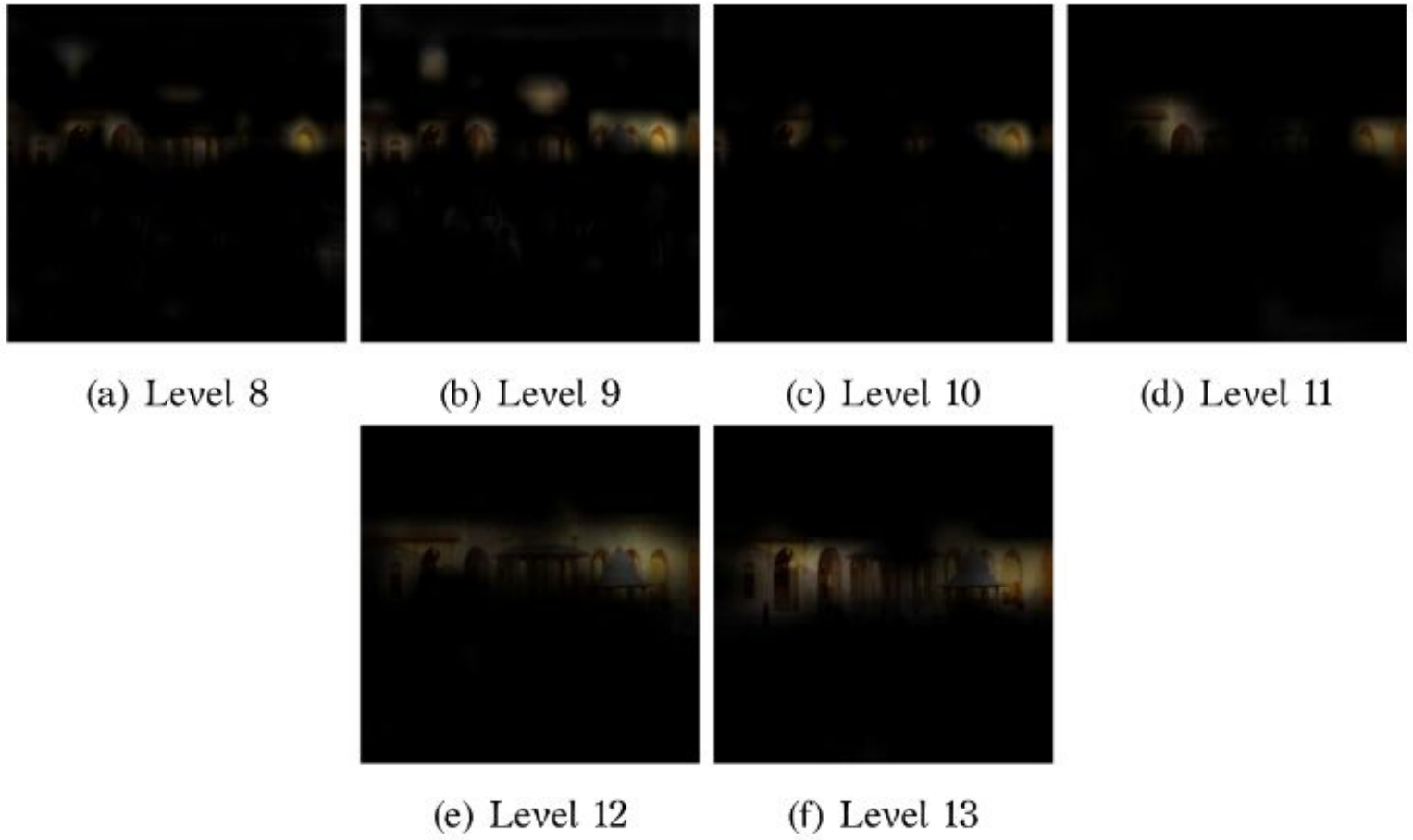
Throughout this work they use **mean Average Precision (mAP)**, and top-N score in order
to evaluate the performance of the proposed method. The definitions of the above metrics
follow below:

Mean average precision is the mean value of the Average Precision (AP) of all the queries. The definition of AP for the i-th query is formulated as follows:

$$AP_i = \frac{1}{Q_i} \sum_{n=1}^{N} \frac{R_i^n}{n} t_n^i,$$

where Q_i is the total number of relevant images for the i-th query, N is the total number of images of the search set, R_i^n is the number of relevant retrieved images within the n top results; t_n^i is an indicator function with $t_n^i = 1$ if the n-th retrieved image is relevant to the i-the query, and $t_n^i = 0$ otherwise.

Approach 1: TF-IDF CNN



(a) Level 8 (b) Level 9 (c) Level 10 (d) Level 11

(e) Level 12 (f) Level 13

Fig. 7 Example of tracing arch like patterns for the input image of Fig. 5

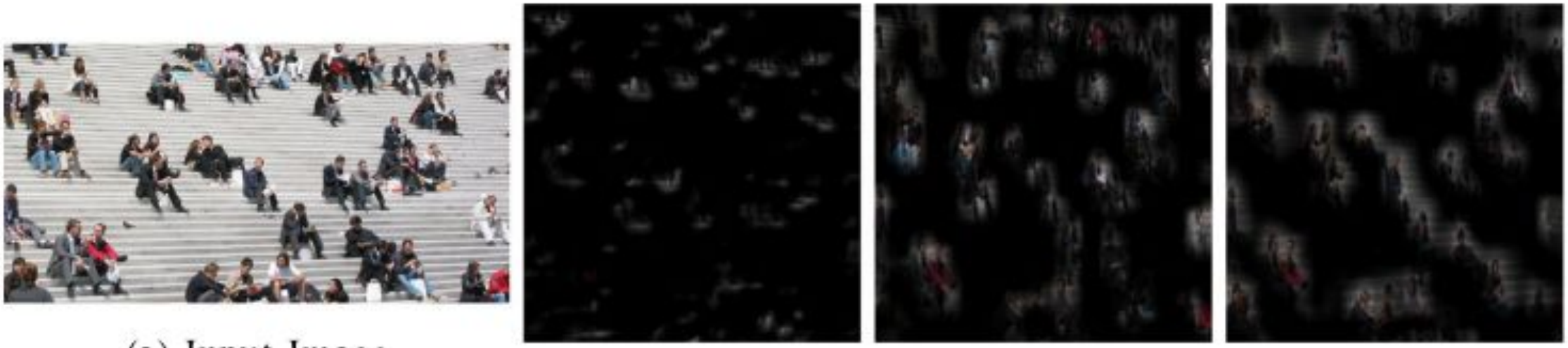
Approach 1: TF-IDF CNN

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 12 (e) Level 13

Fig. 8 Example of tracing domes for the input image of Fig. 5

Top-N score refers to the average number of same-object images, within the top-N ranked images.

Approach 1: TF-IDF CNN



(a) Input Image (b) Level 7 (legs) (c) Level 9 (bodies) (d) Level 9 (heads)

Fig. 9 Example of tracing Human parts

Approach 1: TF-IDF CNN

PES UNIVERSITY

Datasets

Inria Holidays : consists of 991 images divided into 500 classes, and 500 discrete queries. Each class in the search set consists of between 1 and 12 images. Some images of the dataset are not in a natural orientation. We measure the retrieval performance in terms of mAP.

Oxford 5k : consists of 5062 images collected from Flickr by searching for particular Oxford landmarks. The collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. Images are assigned one of four possible queries: Good, Ok, Junk and Absent. Good and ok images are considered as positive examples, absent as negative examples while junk images as null examples. Following the standard evaluation protocol they measure the retrieval performance in mAP.

Approach 1: TF-IDF CNN

Paris 6k : similar to Oxford Buildings dataset, consists of 6392 images (20 of the 6412 provided images are corrupted) collected from Flickr by searching for particular Paris landmarks and provides 55 queries. The retrieval performance is also measured in terms of mAP, using both the full queries and cropped queries versions of the dataset.

UKBench: contains 10200 images of objects divided into 2550 classes. Each class consists of 4 images. All 10200 images are used as queries. The performance is reported as Top-4 Score, which is a number between 0 and 4.

Approach 1: TF-IDF CNN

- They apply the proposed tf-idf approach in the VGG model, on certain layers and they also experiment with combinations of different layers, in order to show that the proposed scheme can improve the baseline results of directly extracted feature representations using a commonly used CNN pre-trained model.

- Additionally, in order to partially preserve spatial information from the convolutional layers, in the VGG case, they apply the aforementioned method of division into five sections in the last convolutional layer.

Approach 1: TF-IDF CNN

- Subsequently, they apply the proposed tf-idf based representation scheme on the FU retrained model, in order to show that the proposed method is applicable to different network architectures, and also can be combined with other state-of-the-art CNN-based works, improving their performance even more.
- In this case, they perform experiments utilizing the last two optimized convolutional layers, where the max pooling operator outputs 384-dimensional, and 256-dimensional feature representations respectively. All results are obtained using the **cosine distance**.

- In the following they abbreviate the proposed tf-idf scheme utilizing the VGG model as TF-IDF(VGG), and correspondingly TF-IDF(FU) the proposed tf-idf scheme on the FU optimized model.

Approach 1: TF-IDF CNN

Feature representation	Oxford 5k cropped	Oxford 5k full	Paris 6k cropped	Paris 6k full
VGG baseline	0.411	0.475	0.358	0.676
TF-IDF(VGG)	0.473	0.543	0.699	0.705
TF-IDF(VGG) & QE	0.52	0.563	0.757	0.746

Feature representation	Inria holidays	UKBench
VGG baseline	0.772	3.184
TF-IDF(VGG)	0.8	3.668
TF-IDF(VGG) & QE	–	3.779

Tables 1 and 2 illustrate the experimental results utilizing the VGG-16 model, for the Oxford 5k and the Paris 6k datasets, and for the Inria Holidays and the UKBench, respectively.

They compare the proposed method against the baseline VGG utilizing the same layers, and extracting directly the feature representations from them, using the common max-pooling operation in the case of the convolutional layers

Approach 1: TF-IDF CNN

- As they can observe the proposed method can achieve notably improved results against the baseline, in all the used datasets, validating our claim that the tf-idf weighting method can successfully applied in the deep CNNs enhancing the information captured from the CNN layers. Furthermore they observed that the query expansion gives another boost in the performance.

- We should also note that the query expansion cannot be applied to the Inria Holidays dataset, since in many cases there is only one relevant to the query sample in the dataset.

Approach 1: TF-IDF CNN

Feature representation	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k	UKBench
FU baseline	0.5509	0.5302	0.8107	0.7468	3.8094
TF-IDF(FU)	0.5742	0.5476	0.8292	0.7481	3.8421

Subsequently, in Table 3, they provide the experimental results for the UKBench, Paris 6k, and Oxford 5k datasets, for the tf-idf scheme on the FU retrained model.

Approach 1: TF-IDF CNN

Table 4 mAP and Top-4 Score for different idf computation approaches (FU retrained model)

Idf approach	UKBench	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k
No idf	3.8094	0.5509	0.5302	0.8107	0.7468
Threshold value	3.8336	0.5658	**0.5476**	0.8270	0.7461
Percentage threshold	3.8294	**0.5742**	0.5475	**0.8292**	**0.7481**
Statistical idf	**3.8421**	0.5636	0.5415	0.8287	0.73

- In Table 4 they present the Top-4 Score and mAP for the different idf computation approaches in the UKBench, Oxford and Paris datasets, utilizing the FU retrained model.
- The best results are printed in bold. It can be observed that the proposed tf-idf schemes outperform the baseline without tf-idf in all the utilized datasets.

Table 5 Comparison against other unsupervised methods

Method	Dim	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k	UKBench
CVLAD* [47]	64k	0.514	–	–	–	3.62
VLAD* [1]	128	0.448	–	–	–	–
T-embedding* [15]	512	0.528	0.461	–	–	–
Fine-residual VLAD [24]	256	–	–	–	–	3.43
BOW* [17]	200k	0.364	–	0.46	–	2.81
SPoC [2]	256	**0.589**	**0.578**	–	–	3.65
Multi-layer [45]	4k	0.567	–	–	–	3.243
MAC* [39]	256	0.522	–	–	–	–
Small Memory Footprint Regimes [33]	256	0.533	0.489	0.67	–	3.368
TF-IDF(FU)	640	0.5742	0.5476	**0.8292**	**0.7481**	**3.8425**

- Table 5 provides a comparison of the proposed tf-idf scheme on the FU optimized model, against other CNN-based as well as hand-crafted techniques for CBIR. The best results are printed in bold.
- Since the proposed method does not utilize supervised learning, they only compare it with unsupervised methods.

PES UNIVERSITY

Conclusion

- Experimental results on four challenging image retrieval datasets demonstrated the improved performance of the proposed approach.
- It should also be noted that the proposed approach can be easily combined with more sophisticated approaches that have been recently proposed to give a new perspective on treating convolutional image retrieval.
- To this aim, they also conducted experiments utilized our fully unsupervised model toward image retrieval enhancing even more the retrieval performance, leading also to state-of-the-art results against other unsupervised approaches.

References

- “Content-Based Visual Information Retrieval”, Oge Marques and Borko Furht
- “Exploiting tf-idf in deep Convolutional Neural Networks for Content Based Image Retrieval” - Nikolaos Kondylidis, Maria Tzelepi, Anastasios Tefas.
- “Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review” - Afshan Latif

Approach 2: Hashing Based Retrieval

- Hashing-based retrieval methods refer to techniques that use hash functions to map data into a fixed-size hash code or hash value. These methods are commonly used in information retrieval systems to efficiently store and retrieve data, especially in scenarios where the dataset is large.
- The purpose of hashing is to convert high-dimensional images to low-dimensional binary codes while preserving the similarity relation of the images in the original space.
- Hashing allows for rapid data lookup and retrieval by mapping data items to hash codes and organising them in hash tables or other data structures.

Approach 2: Hashing Based Retrieval

- To handle large-scale image data, hashing-based retrieval methods have become attractive because hashing can encode the high-dimensional data into compact binary codes while maintaining similarity among neighbors, leading to significant gains in both computation and storage.

Approach 2: Hashing Based Retrieval

Based on whether employing semantic information, hashing methods can be classified into two groups:

Unsupervised Hashing:

- The primary goal of unsupervised hashing is to explore the intrinsic structure of the data, maintain the similarity among neighbours without any semantic information and represent it as compact binary codes without relying on any external semantic information.
- Unsupervised hashing methods typically leverage unsupervised learning algorithms to map data points into binary codes. These methods often focus on preserving the pairwise similarities or distances between data points during the hashing process.

Approach 2: Hashing Based Retrieval

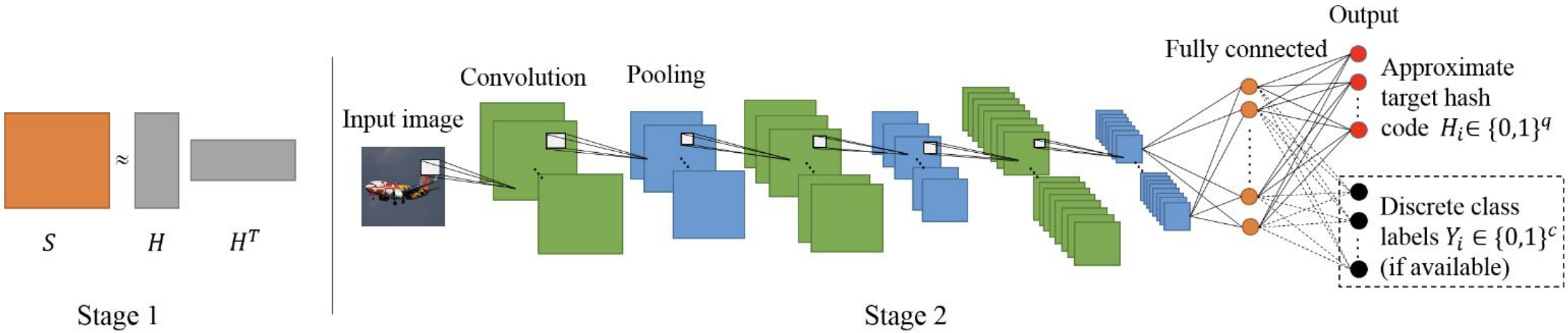
- Unsupervised hashing is useful when the data lacks explicit semantic labels or when semantic information is not readily available.

Supervised Hashing:

- Supervised hashing incorporates semantic information (Eg: class labels , user-defined similarities) during the learning process to generate binary codes.
- Supervised hashing methods utilize labeled data to learn a mapping that preserves the intrinsic structure but also respects the semantic relationships among data points. This often involves using techniques like deep neural networks or other machine learning models trained with supervision.
- Supervised hashing is particularly useful when semantic information is available and can enhance the quality of binary codes.

Approach 2: Hashing Based Retrieval

Supervised Hashing example:



Reference: [Supervised Hashing for Image Retrieval via Image Representation Learning - R Xia et al]

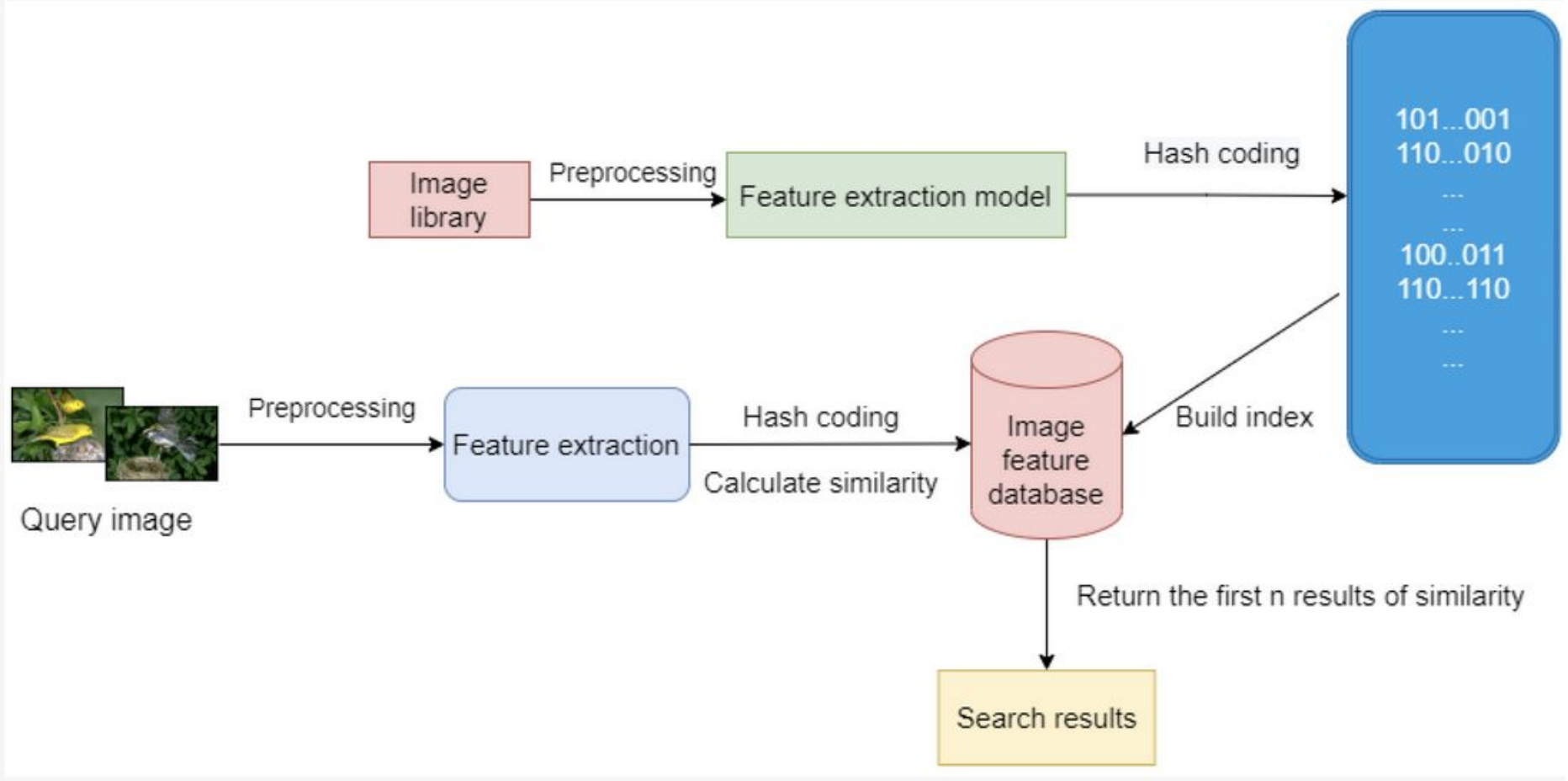
Approach 2: Hashing Based Retrieval

- In stage 1, the pairwise similarity matrix S is decomposed into a product HH^T ,where H is a matrix of approximate target hash codes.
- In stage 2, we use a convolutional network to learn the feature representation for the images as well as a set of hash functions.
- The network consists of three convolution-pooling layers, a fully connected layer and an output layer.
- The output layer can be simply constructed with the learned hash codes in H (the red nodes). If the image tags are available in training, one can add them in the output layer (the black nodes) so as to help to learn a better shared representation of the images.
- By inputting an test image to the trained network, one can obtain the desired hash code from the values of the red nodes in the output layer.

Approach 2: Hashing Based Retrieval

PES UNIVERSITY

Architecture



Approach 2: Hashing Based Retrieval

1. **Feature Extraction:**
    - Identify the features or characteristics in the image that are relevant to the detection task. These could be keypoints, descriptors, or any other distinctive features.
2. **Hashing Function:**
    - Design a hashing function that can encode the extracted features into a fixed-size hash code. The goal is to represent the essential information about the features in a compact form.
    - Traditional hash functions, like random projections or cryptographic hashes, are often manually designed, but ML allows for the automatic learning of hash functions from data.

Approach 2: Hashing Based Retrieval

- ML models, particularly deep learning architectures, can be trained to directly generate hash codes that capture intricate patterns and semantic similarities within the data.
- For instance, deep hashing models employ neural networks to map input data into a hash space where similar items yield similar hash codes.
- Supervised hashing leverages labeled data to guide the learning process, while unsupervised hashing methods, such as autoencoders or GANs, focus on learning hash codes without explicit labels.

3. **Database Construction:**
   - Create a database of hash codes for known objects or features. This involves processing a set of training images containing the target objects or features, extracting their features, and generating hash codes.

Approach 2: Hashing Based Retrieval

4. **Encoding the Query Image:**
   - Apply the same feature extraction process to the query image. Use the hashing function to generate a hash code for the features in the query image.
5. **Hash Code Comparison:**
   - Compare the hash code of the query image with the hash codes stored in the database. The goal is to find the closest matches based on hash code similarity.
   - One of the most common methods to do so is using Hamming distance. The Hamming distance measures the number of positions at which the corresponding bits are different.

Approach 2: Hashing Based Retrieval

The Hamming distance between two integers is the number of positions at which the corresponding bits are different. For example 20 (10100 in binary) and 17 (10001 in binary), the Hamming distance is 2 as there are two differing bits.

1	0	**1**	0	**0**

1	0	**0**	0	**1**

Approach 2: Hashing Based Retrieval

PES UNIVERSITY

Hashing Methods:

A. Data-Independent Methods

- The hash function design of the data-independent hashing method is independent of the data, and the hash function is generally generated by means of random mapping.
- The most typical representative method is the Locality-Sensitive Hashing (LSH) . The principle of locality-sensitive hashing is to map samples with high similarity in the original space to the same hash bucket with higher probability, which ensures that the hash codes of the neighbor samples in the original space can be as close as possible after hash mapping.

Approach 2: Hashing Based Retrieval

- The random mapping matrix is independent of the data and is determined by the probability distribution.
- However, in large-scale data retrieval, because of the hash collision problem, a longer hash code is needed to ensure the precision of the retrieval, which brings additional time and computational overhead, and leads to a decrease in the recall rate.

B. **Spectral Hashing (SH)**

- Spectral hashing is a type of single-modal hashing technique designed to maintain the neighborhood relationships of data points in the original feature space even after they have been hashed which ensures that the hash code obtained by the mapping is compact and rich in information.

Approach 2: Hashing Based Retrieval

The algorithm works as follows:

1. **Principal Component Analysis (PCA)**: Find the principal components of the data using PCA.
2. **Calculate eigenfunctions**: Calculate the k smallest single-dimensional analytical eigenfunctions of Lp using a rectangular approximation along every PCA direction. This is done by evaluating the eigenvalue equation to obtain an eigenvector solution
3. **Fit a multidimensional rectangle**: The aspect ratio of this multidimensional rectangle determines the code using a simple formula

Reference:
https://people.csail.mit.edu/torralba/publications/spectralhashing.pdf

Approach 2: Hashing Based Retrieval

PES UNIVERSITY

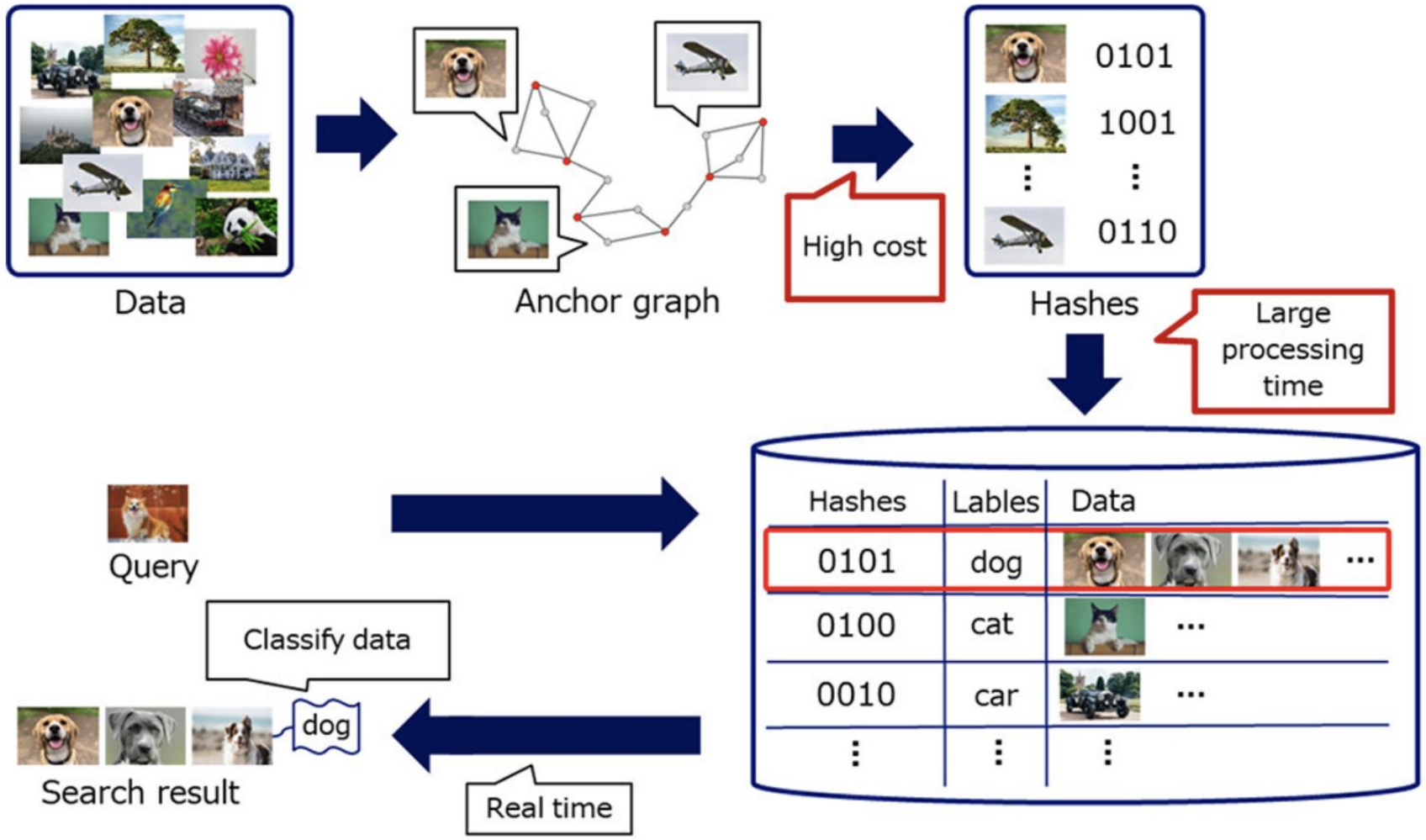
C. Anchor Graph Hashing (AGH)

- It is a single-modal graph-based hashing method whose principle is similar to the spectral hashing, but it does not require the precondition of uniform data distribution.
- The method first selects the points in the dataset (generally using the K-means clustering) as an anchor point to approximate the similarity matrix, and converts the similarity between any two data sample points into a sample-anchor relationship.
- The anchor graph hashing method is based on graph analysis and has strong scalability. It also uses a double hash function to generate a multidimensional hash code to solve the problem of uneven information content in the feature vector generated by the original data.

Approach 2: Hashing Based Retrieval

PES UNIVERSITY

AGH Architecture



Approach 2: Hashing Based Retrieval

D. Cross-View Hashing (CVH)

- The Cross-View Hashing is a multi-modal extension of the spectral hashing.
- The basic idea is to learn the hash function by minimizing the weighted average Hamming distance of different modalities and use the generalized eigenvalue solution method to obtain the minimum value.
- CVH can be applied to the data retrieval of multiple modalities, but the differences between modalities are not fully considered, hence the retrieval performance is limited.
- CVH is an unsupervised method. It uses graphs to describe the similarity between modalities. It can also use the label information to solve the similarity matrix and convert it into a supervised method.

Approach 2: Hashing Based Retrieval

PES UNIVERSITY

Advantages:

- Fast retrieval
- Space efficient
- Scalable
- Suitable for parallelism

Disadvantages:

- Collision Risks
- Sensitivity to small changes: Even a slight modification in an item can result in a significantly different hash code, impacting the effectiveness of similarity search.
- Some hash functions lack semantic relationships and can limit their ability to capture complex relationships.

References

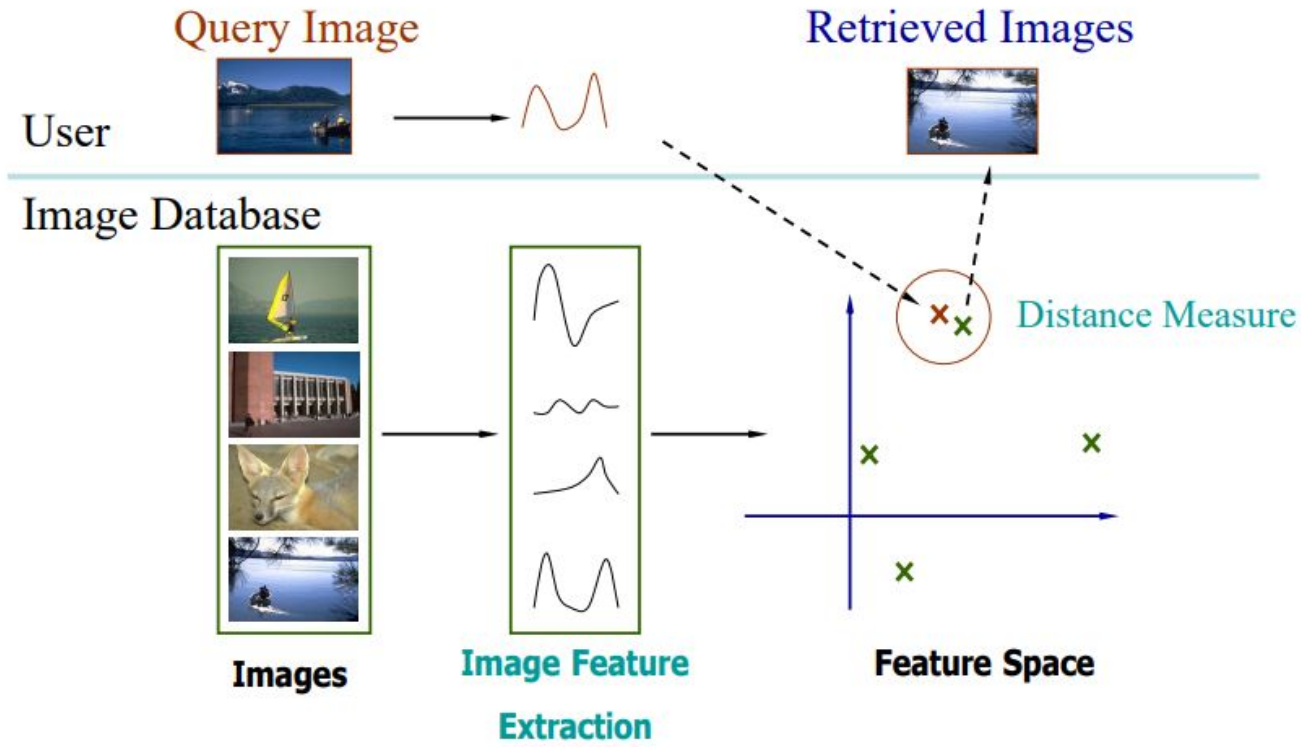
- https://cdn.aaai.org/ojs/8952/8952-13-12480-1-2-20201228.pdf
- W. Cao, W. Feng, Q. Lin, G. Cao and Z. He, "A Review of Hashing Methods for Multimodal Retrieval," in IEEE Access, vol. 8, pp. 15377-15391, 2020

PES UNIVERSITY

Content Comparison using Image Distance Measures

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image.

Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly.



Minkowski-form Distance

Minkowski-form is the most widely used distance measure due to its computational simplicity. It is also known as L_P norm; where P represents the order of distance. Consider two vectors M and N in $\mathbb{R}^k$; where k is the dimensionality of Euclidean space, then Minkowski-form distance L_P is:

$$L_P(M,N) = \left(\sum_{i=1}^{k} |m_i - n_i|^P\right)^{1/P}$$

Minkowski-form distance of order $P \geq 1$ is a metric. On the contrary, it is a non-metric for $P < 1$ due to the violation of triangle inequality. The most commonly used distances in this family are L_1 distance and L_2 distance, also known as Manhattan distance and Euclidean distance, respectively.

Euclidean Distance

Euclidean distance is a straight-line distance between two corresponding elements of a feature vector. It is a bin-by-bin distance, also called as Pythagoras distance.

$$L_2(M,N) = \sqrt{\left(\sum_{i=1}^{k} |m_i - n_i|^2\right)}$$

Additionally, it is the only Lp distance that is invariant to orthogonal transformation. Euclidean distance is applicable in various areas of image processing such as clustering, classification, retrieval, and also in distance transformation or mapping.

Distance transform is a process of converting a binary digital image into another image using some distance function. Each pixel in converted image has a value corresponding to the distance to the nearest pixel in image.

Euclidean Distance

Applications that use Euclidean distance:

MARS system used Euclidean distance to compute the similarity between **texture** features

Netra used Euclidean distance for **color** and **shape** feature, and

Blobworld used Euclidean distance for **texture** and **shape** feature

Manhattan Distance

Apart from Euclidean distance, another popular distance metric of L_P family is Manhattan distance, also known as taxi-cab distance or city block distance.

$$L_1(M,N) = \sum_{i=1}^{k} |m_i - n_i|$$

It is widely used in many image retrieval and color indexing-based approaches especially for texture features.

Chebyshev Distance

Lastly, another useful member of L_P distance family is L_∞ distance, also known as Chebyshev distance or chessboard distance. Chebyshev metric computes the distance between two feature vectors by taking the maximum of their element differences along any coordinate dimension. In two-dimensional space, it is equivalent to the minimum number of moves a king needed to travel between any two squares on the chessboard.

$$L_\infty(M,N) = \max_{i=1}^{k}(|m_i - n_i|) = \lim_{P\to\infty}\left(\sum_{i=1}^{k}|m_i - n_i|^P\right)^{1/P}$$

L∞ distance to compute the similarity between texture images

Chi-square Statistic

Chi-square statistic is a weighted Euclidean distance which is used to measure the corresponding elements of two feature vectors. This distance measure derived its name from chi-square test statistics which is used to test fitness between a distribution and observed frequencies. It is defined as:

$$\chi^2 = \frac{1}{2}\sum_{i=1}^{k}\frac{(m_i - n_i)^2}{m_i + n_i}$$

It is widely used to compute the difference between two histograms where the difference between the large bins is less important than the difference between the small bins

Histogram Intersection Distance

PES UNIVERSITY

Histogram intersection distance was originally proposed by for comparison of color histograms. This approach is robust to occlusion, image resolution, variation, varying viewpoint, and distraction in the background of the object. It is defined as:

$$\text{Histogram intersection distance } D_{\cap}(M,S) = \sum_{i=1}^{n} \min(M_i, S_i)$$

where M and S are two histograms with n bins. The outcome of histogram intersection is the number of pixels from the model image that has corresponding pixels of the same color in the sample image. The intersection result can be normalized by dividing the distance value by number of pixel in model histogram. Then, the distance value is:

$$D_{\cap}(M,S) = 1 - \frac{\sum_{i=1}^{n} \min(M_i, S_i)}{\sum_{i=1}^{n} M_i}$$

Mahalanobis Distance

Mahalanobis distance is used to compute the distance between a given feature vector and a distribution.

$$D_M(v) = \sqrt{(v-m)^T C^{-1}(v-m)}$$

where v is the feature vector of form $v = (v_1, v_2, v_3, \ldots v_k)$, m represents the mean row vector, C is the covariance matrix and T indicates the transpose operation. It is unit-less and scale-invariant distance metric and takes into account the correlations of the dataset.

However, for certain high-dimensional data, the computation of Mahalanobis distance is quite expensive due to the covariance matrix calculation.

Quadratic Form Distance

Quadratic form (QF) distance is used in color-based image retrieval.

two N-dimensional distributions $x, y \in \mathbb{R}^n$, quadratic form distance is defined as:

$$\mathrm{QF}(\mathrm{x}, \mathrm{y}) = \sqrt{(x-y)^{\mathrm{T}} A (x-y)}$$

where A is bin similarity matrix that stores the cross-bin information in form of matrix elements a_{ij}. Each element in similarity matrix tries to capture the perceptual similarity between the features represented by bins i and j. Computation of a_{ij} usually depends on the ground distance (d_{ij}). One such interpretation of a_{ij} is:

$$a_{ij} = 1 - \frac{d_{ij}}{d_{\max}}$$

d_{ij} could be the Euclidean distance between bin i and j, and $d_{\max} = \max_{ij}(d_{ij})$. Generally, the QF distance is not a metric, but for certain choice of similarity matrix A, it is indeed a metric.

Quadratic Form Distance

Quadratic form distance has been used in many retrieval systems for color histogram-based image retrieval. It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross-similarity between colors.

CBVIR - Examples

PES UNIVERSITY

QBIC

QBIC (Query By Image Content) was developed by IBM Almaden Research Center. Its framework and techniques have influenced many later systems. QBIC supports queries based on example images, user-constructed sketches, and selected colors and texture patterns. In its most recent version, it allows text-based keyword search to be combined with content based similarity search. The online QBIC demo can be found at: http://wwwqbic.almaden.ibm.com

CBVIR - Examples

Photobook

Photobook is a set of interactive tools for browsing and searching images developed at MIT Media Lab. Photobook consists of three sub-books, from which shape, texture, and face features are extracted respectively. Users can query the system based on features from each of the three sub-blocks. Additional information about Photobook can be found at: http://www.white.media.mit.edu/vismod/demos/photobook/index.html.

CBVIR - Examples

PES UNIVERSITY

Netra

Netra is a prototype CBVIR system developed in the UCSB Alexandria Digital Library (ADL) project. It uses color, shape, texture, and spatial location information in the segmented image regions to search and retrieve similar images from the database. An online demo is available at http://vivaldi.ece.ucsb.edu/Netra/.

A new version of Netra, Netra 2, which emphasizes the group's latest work on color image segmentation and local color feature, is available at http://maya.ece.ucsb.edu/Netra/index2.html.

CBVIR - Examples

PES UNIVERSITY

MARS

MARS (Multimedia Analysis and Retrieval System) was originally developed at University of Illinois at Urbana-Champaign. The main focus of MARS is not on finding a single “best” feature representation, but rather on how to organize the various visual features into a meaningful retrieval architecture, which can dynamically adapt to different applications and different users. MARS formally proposes a relevance feedback architecture in Image Retrieval and integrates such technique at various levels during retrieval, including query vector refinement, automatic matching tool selection, and automatic feature adaptation. More information about MARS can be obtained at: http://www-db.ics.uci.edu/pages/ research/mars.shtml.

CBVIR - Examples

PES UNIVERSITY

Blobworld

Blobworld is a CBVIR system developed at U.C. Berkeley. The program automatically segments an image into regions, which roughly correspond to object or parts of objects allowing users to query for photographs or images based on the objects they contain. Their approach is useful in finding specific objects and not, as they put it, “stuff” as most systems which concentrate only on “low level” features with little regard for the spatial organization of those features. It allows for both textual and content-based searching. This system is also useful in its feedback to the user, in that it shows the internal representation of the submitted image and the query results. Thus, unlike some of the other systems, which allow for color histogram similarity metrics, which can be adjusted, this can help the user understand why they are getting certain results.

References

- “Content-Based Image Retrieval : Ideas, Influences, and Current Trends” - Vipin Tyagi, Springer Nature Singapore Pte Ltd., 2017. - Chapter 1 & Chapter 4