Content-Based Image Retrieval

-

State Of The Art

Léo Vetter

February 22, 2016

Contents

Chapter 1

CBIR Overview

1.1 Introduction

It is often said that an image is worth a thousand words. This idiom refers to the notion that a complex idea can be conveyed with just a single still image. Indeed they are a powerful mean of communication and we interact with them in our everyday life. We use them to freeze a moment and share it with our relatives and friends, to illustrate a written article or to capture a emotion in a painting. Nowadays an increasing number of devices, cameras, satellites, video sensors, allow us to automatically record digital images which are then stored in constantly increasing databases. The phenomenal expansion of the web enabling us to store those images online also contributes to this collection of images in very huge databases. To allow us to take advantage of this increasing amount of data we need to develop tools that efficiently process images and retrieve them in a meaningful fashion. This is the problem addressed by Content-Based Image Retrieval. Formally it can be defined as *any technologies that help to deal with the organization of digital pictures based on their content* [15]. The key idea is that no metadata is involved and only the raw pixels of the images are used in order to infer semantic knowledge. It is a growing field which span over numerous research area such as Information Retrieval, Computer Vision, Machine Learning...

Humans are naturally gifted to deduce informations by just looking a few seconds at an image and it is interesting to understand how we are performing this task in order to be able to reproduce it into a machine. In doing this operation we are usually relying on at least three different factors :

Our sensory abilities Our sensory abilities refer to our ability to acquire visual inputs through our eyes and the process that is made of these inputs by our visual cortex.

The context The context of an image include everything that is not in the image but contribute to its interpretation. It is well known that an image can have different meaning depending on its context.

Our experiences Different experiences makes us respond differently when confronted to an situation. This is valid for the interpretation of an image which will rationally differ between individuals.

In order to successfully retrieve images for an end-user a machine should theoretically be able to incorporate these three factors into its process. As one can imagine such objective is especially challenging for many reasons. First we are far from completely understanding how our brain is processing the visual inputs that it receives. Then the context of an image is not always available, especially when dealing only with raw pixels without the help of additional metadata. Finally individuals with different cultures can have very different interpretation of the same image and retrieve the right images for an user would imply to hame some knowledge about its background or intentions.

1.2 Real World Applications

Even if the process of making good interpretation based on the content of an image is far from being mastered numerous real world applications make already use of Content-Based Image Retrieval technologies. Among this different applications we can already differentiate between those who operate on images with a very broad domain and those who operate on a narrow domain. In the review made by Smeulders and all [52] they define a broad domain as having *"an unlimited and unpredicted variability even for the same semantic meaning"*. Web applications and general visual search engine often process on a broad domain since images can come from many different users and similar objects can have different size, shape or illumination and can even be partially occluded.

The first visual search engine was developed by IBM with their commercial system called QBIC (Query by Image Content) [43]. QBIC enable an user to query a image database with image queries. Many possibilities is given to the user to specify its query. He can either query by sketch, query by specifying specific color, texture and shape or query using an image similar to the ones he was looking for. Depending on the type of the query different similarities measures are then used to retrieve the most similar images. In order to help further the research semi-automatic detection of object or point of interest is also performed when a new image is inserted into the database. Since this first attempt other systems (TinEye, Pinterest, Google Query By Image) have been developed which rely on more advanced techniques. For instance Pinterest is a web and mobile applications where user can upload images of interest through collections known as pinboards. Pinterest then use those images to look for similar images and recommend them to the user. While QBIC was only relying on handcrafted features Pinterest, among other algorithms, rely on features learning and algorithms called Convolutional Neural Network to extract features from images [34].

As opposed to a broad domain a narrow domain is defined as having *a limited and predictable variability in all relevant aspects of its appearance.* When dealing with a narrow domain knowledge of the domain can be efficiently used to design good features detectors. Illustration of a narrow domain can be found in many applications such as medical diagnosis or face recognition. In the medical domain, due in part to the steadily growing rate of image production, Content-Based Image Retrieval can play a key role in many medical applications [42]. It can for example enable the automatic annotation and the classification of medical images. Content-Based Image Retrieval can also be used to help in the clinical decision–making process. Indeed when confronted with a medical case an useful scenario would be to supply the doctor with similar cases by finding other images of the same disease that would allow him to take a decision with more informations at hand. Another application is for instance within the radiology department to develop tools that enable automatic diagnostic of the diseases [35].

Content-Based Image Retrieval techniques might also find useful applications in many more domains. For instance in digital art gallery or museum collection available online they can be used to bring to the user art work similar to the ones he browsed. For law enforcement and criminal investigation they could help investigator to process pictures of security cameras.

1.3 Components of CBIR systems

All typical Content-Based Image Retrieval systems are usually performing common steps to achieve the retrieval of images as illustrated by the figure 1.1. The first steps which might be performed is segmentation which can be useful to isolate meaningful regions from the background for example. As a second step features extraction is performed. The extraction of features can rely of two different methods. Either it is done through handcrafted features that is humans have to design a specific features detector for the problem at hand or it is done through feature learning that is we let the machine learn features thanks to training samples. In the domain of image processing one model have started to show great promises to learn features and is known as Convolutional Neural Network. One of the great challenge of features extraction is to reduce what is known as the *semantic gap* that is the difference between the interpretation of the image and the features that have been computed. Finally one have to interpret these features which is often done through a similarity measure or a machine learning classifier. The following of this work will detail each of these steps.

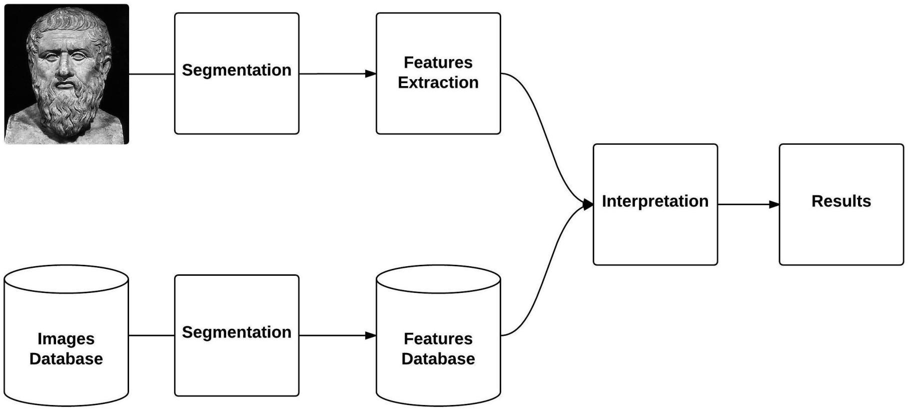

Figure 1.1: Content-Based Image Retrieval System Components

While every Content-Based Image Retrieval system usually perform the operations described above their performance is not limited to them. Below is a review of the main components that typically compose a CBIR systems and improvement in each of these components is required to achieve better systems [24].

1.3.1 Database Management

When an image database is growing very large efficient access methods should be developed in order to speed up the retrieval of images. These methods should act as a filter with only relevant images being retrieved from the storage system and tested for similarity with the query image. Most data structure used today for indexing are spatial access methods and assume that all images features are numeric and represent an image as a point in a multidimensional space. These methods can be classified into three categories : *space partitioning*, *data partitioning* an *distance-based* techniques [52]. Space partitioning techniques (k-d tree, k-d B-tree...) organize the feature space as a tree and with a node standing for a region of the space. When a region contain too much points it is split in subregions whose nodes become the children of the original regions. Data partitioning techniques (R-tree, R+-tree, R*-tree...) associate with each points a region representing the neighborhood of that point and leaf nodes correspond to minimum bounding regions of a set of vectors. Distance-Based index structures (VP-tree, MVP-tree, M-tree...) is applicable to general metric spaces whose primary idea is to pick an example point and divide the rest of the features space into concentric rings around the example. Nevertheless when the features vectors being indexed contain a considerable number of dimensions the performance of the previous indices greatly decrease. This is known as the *curse of dimensionality* phenomenon.

1.3.2 Query Specification

Queries made by the user can take several forms which logically influence on the retrieval process. Query by example where the user provide a image and similarity is computed based on the entire scene is a common way to perform the request Another method is to let the user specify a region of interest (a person or an object for example) and search for images with similar regions. Yet an other way is for the user to draw a sketch of the item or landscape he is interested in. How the query is specified depend of the users of the system. Casual users surely prefer to make its request by using natural language "Find me mountain with snow and with a bird in the sky" or with an image as an example whereas domain expert might want more sophisticated methods such as query by sketch.

1.3.3 Relevance Feedback

As pointed out earlier bringing relevant images to the end user imply to know its point of view. Learning the point of view of the user is more and more achieved through relevant feedback. From a broad perspective a relevance feedback scenario usually has the following scheme. First initial results are returned by the machine then the user provide a judgement on the results being displayed and finally the machine learns from the judgement to retrieve new results and so on. The exact algorithm used for learning is going to depend on whether the user is looking on a particular target item or a class of similar item. Also it is going to differ if the user provide a binary judgement (relevant or not) or a comparative judgement (this result is better that this one). Different relevance feedback algorithms can be found in the survey of Zhou and all [64] or the one of Crucianu and all [12].

Chapter 2

Segmentation

As humans we often attribute the semantic of an image based only on a particular region of this image disregarding the rest of the scene. Therefore it is logic when trying to learn good representation of the image to first select meaningful regions by segmentation of the image. Good segmentation should group into regions pixels with similar characteristic such as color or texture. Below is a review of different techniques that have been investigated to perform segmentation.

2.1 Clustering methods

One way to achieve image segmentation is through clustering method. The most basic case would be to use a simple k-means algorithm. In this case the main steps to perform segmentation are :

1 Choose k clusters (e.g. pixels) either randomly or based on some heuristic

2 Based on a distance measure assign each pixel in the image to the closest cluster

3 Recompute the cluster centers

4 Repeat step 2 and 3 until convergence that is no more pixels change of cluster.

Improvement upon this simple scheme is to use fuzzy clustering. In fuzzy clustering pixels, rather than belonging completely to just one cluster, has a degree of membership. Further improvement is achieved by adding spatial information to the membership function as in the fuzzy c-means clustering proposed by Chuang and all [9]. Indeed their finale membership function take into account features of the pixels and also neighboring information. Hence the probability to belong to a cluster will be higher for a pixel if the neighboring pixels already belong to this cluster. Fuzzy c-Means clustering is also explored by Ahmed and all [1] for segmentation of magnetic resonance imaging.

2.2 Histogram-based methods

The principle of histogram-based methods for image segmentation is first to compute an histogram from the image. Then one or many thresholds are chosen between two peaks of the histogram and pixels of the image are attributed into clusters according to these thresholds. These techniques is for example used by Arifin and all [2] to distinguish between the foreground and background of grayscale image. Refinement of this technique is to recursively apply the histogram computation and the thresholding to subregions as explained in detail in the work of Ohlander and all [45].

2.3 Region-growing methods

In Region-growing methods we start with elementary regions, either all pixels or a subset of pixels, and we iteratively combine these small regions based on a statistical test to decide or not of the merging. A recent algorithm that follow this approach is Statistical Region Merging proposed by Richard Nock and Frank Nielsen [44]. Their algorithm is based on a model of image generation which captures the idea that grouping is an inference problem. It provide a simple merging predicate and a simple ordering in merges. They argue their method can cope with significant noise corruption, handle occlusions, and perform scale-sensitive segmentations. Another method is the seeded region growing method. This method take a set of seeds as input along with the image which correspond to the objects to be segmented. Then the regions are iteratively grown by comparing unallocated neighboring pixels to the regions. One way of comparison could be for example to compare the intensity of a pixel with the average intensity of the region.

2.4 Graph partitioning methods

Graph partitioning methods have been lately the main research direction for segmenting images. These methods see an image as a weighted undirected graph $G = (V, E)$ where each node $v \in V$ correspond to a pixel or a group of pixels and each edge $(i, j) \in E$ is weighted according to the dissimilarity between the two pixels that are linked.

2.4.1 Cut Criterion

In order for the graph image to be partitioned into relevant clusters good cut criterions must be found. A cut in graph theoretic is the partitioning of the graph into two disjoint subset which are judged dissimilar. The degree of dissimilarity is basically computed by doing the sum of the vertices that connect the two subsets :

$$Cut(I, J) = \sum_{i \in I, j \in J} w(i, j)$$

where w is a function used to estimate the similarity between two nodes/pixels. The problem of this metric is that it tends to create clusters composed of a unique node. A popular criterion for finding good clusters is know as normalized cut [50]. To avoid the unique node bias normalized cut suggest to normalize the cut criterion by the total edge weights of all the nodes in the graph. Other criterions are minimal cut [61] (a cut is minimum if we can't find in the graph a cut with smaller weight) or maximum cut (No cut with a biggest cut weight). Yet an other recent variant is the Multiscale Normalized Cuts (NCuts) approach of Cour et al. [11].

2.4.2 Graphical Probabilistic Model

In an image pixels in homogeneous regions often share some properties (they have the same color or the same texture for instance). Markov Random Field is a probabilistic framework that enable to capture such contextual informations. In a Markov Random Field for image segmentation observational field correspond to pixels and the goal is to assign a class label to each pixel. Thus the function the model is trying to maximize is the probability of identifying a label scheme given some features. Roughly the steps involved in the segmentation of an image thanks to a markov random fields are the following :

Features Extraction Features are computed for each pixels.

Initial Probabilities Based on the features extracted initial probabilities of belonging to the class labels are computed.

Parameters estimation Based on training samples parameters statistic (mean and variance) are computed for each label.

Marginal distribution Probabilities of features given a label are computed using Baye's theorem and parameters computed previously.

Class label Probability Taking account its neighborhood probabilities of class labels for each pixel are computed

Iteration Iterate over new prior probabilities and redefine clusters to maximize these probabilities. When probability is maximized and labeling scheme does not change the iterations stop.

Demonstration of segmentation with markov random fields can be found in the work of Won and all [60] or the one of Zhang and all [63] which use the model to segment brain magnetic resonance (MR) images.

As alternative to markov random fields conditional random fields have also been investigated. In the work of Plath and all [48] it is used to perform scale-space segmentation with class assignment.

Chapter 3

Features Extraction

Features in Content-Based Image Retrieval are relevant informations about the underlying image. What makes that a feature is relevant is generally dependent of the problem at hand. Nevertheless one can identify several properties that features should respect in order to be considered good features. Thus features should be invariant to :

Affine transformation (Rotation, Scaling, Translation...). Ideally features computed on an image would be the same whatever the location or the scale of the object in the image.

Distortion As for affine transformation features should be tolerant to small distortion.

We can discern between two approaches when dealing with features extraction. The first one is to rely on handcrafted features that describe elementary characteristics of the image such as the shape, the color or the texture. A major drawback of handcrafted features is their dependence to the application domain which led to another set of techniques called feature learning. Feature learning exploit training dataset to discover useful features from images. Another distinction one can make in the domain of features extraction is between global and local features.

3.1 Global and Local Features

Global features are features which are aggregated from the entire image. More formally global features can be symbolized by :

$$F_j = \sum_{T_j} f \ o \ i(x)$$

where $\sum$ represent an aggregation operation (can be different that sum), T is the partitioning over which the value of F_j is computed, f account for possible weights and i(x) is the image.

As opposed to global features local features are computed by considering only a subpart of the image. Usually for an image a set of features is computed for each pixel using its neighborhood or for non-overlapping block. After this step we usually have a set $X_i, 0 < i < sizeimage$ where X represent the features vector computed at the location i of the image. A further step of summarization can also be performed. For example we might derive a distribution for Xi based on the set. As reported by Datta and all [16] Local features often correspond with more meaningful image components, such as rigid objects and entities, which make association of semantics with image portions straightforward.

3.2 Handcrafted Features

3.2.1 Color Features

An example of global features that has been extensively used is color histogram that is a representation of the distribution of colors in an image. Color histograms can be useful for retrieval as long as the color pattern of interest is representative of an item throughout the dataset. It has the advantages to be robust to translation and rotation transformation. Various distance measure can then be used such as euclidean distance, histogram intersection, cosine or quadratic distances to compute the similarity between images [55] [25] [54]. However color histograms suffer from obvious flaws. Thus color histogram can't be of any help to identify that a red cup and a blue cup actually represent the same object and additionally if the similarity of two images with very different scene but identical color distribution is computed using a color histogram they might be falsely judged similar. To improve color histograms efficiency joint histograms [46], histograms that incorporates additional information other than color, two-dimensional histograms [4], histogram that considers the relation between the pixel pair colors, or correlogram [29], a three dimensional histogram, have been investigated.

Another key issue when dealing with color features is the choice of the color that is been used (RGB, HSV, Lab-Space...) which usually depend on the special need of the application. Two aspect of colors have to be taken account here. The first is that depending on how the scene was taken (viewpoint of the camera, position of the illumination, orientation of the surface...) the color recorded might varies considerably. The second is that the human perception of color greatly changes between individuals. RGB space is one of the most popular color space and assign for each pixel a (R(x), G(x), B(x)) triplet corresponding to the additive primary colors of light (Red, Green, Blue). RGB space is an adequate choice only when there is little variations in the recording. For instance the RGB space would probably be a good choice for art painting but a bad choice for outdoor taken pictures. Indeed a color relatively close in the RGB color space might be perceived as very different from the point of view of an human. In the opponent color space colors are defined according the opponent color axes derived from the RGB values : (R-G, 2B-R-G, R+B+G). It has the advantage to isolate the brightness information on the third axis. Since humans are more sensitive to variations in brightness the two other axis could be downsampled to reduce the memory usage. Other color space have been studied and can be found in the survey of Smeulders and all [52] or the one of Khokher and all [36].

3.2.2 Shape Features

Shape features methods are trying to identify interesting regions in an image like edges or corners and computes features based on these regions. As a primary step scale-space detection is very often performed since it provides the way to detect interesting regions at any scale. A widely used detector is SIFT (Scale-Invariant Feature Transform) published by David Lowe in 1999 [40]. SIFT extract keypoints from an image at different scale-space using Difference of Gaussians and assign an orientation to each of them to achieve invariance to image rotation. Keypoints descriptors are then computed based on their neighborhood that is for each neighborhood a orientation histogram is created. In addition several measures are taken to increase the robustness of the descriptors to changes in illumination. Improvement over SIFT have since been made especially with the Speeded Up Robust Features (SURF) detector [5] which is several times faster than SIFT and claimed by their authors to be more robust against different image transformations. An other descriptor is known as Histogram of oriented gradients. The idea behind histogram of oriented gradients is that object or human can be described by the distribution of oriented gradients. In this techniques the image is divided into cells (grouping of adjacent pixels) of circular or rectangular shape. For each pixels gradient orientation is computed and then for each cell histogram of gradients orientation are deduced. Each pixel in a cell contribute to the histogram depending generally on the magnitude of the gradient. To account for illumination changes and shadowing, gradients in a region (grouping of several cells), are normalized with the average intensity of the region. HOG descriptor is then the concatenated vector of all the normalized histograms. It has been investigated for instance for human detection [14]. Other techniques include harris corner detector [27] or the Hough Transform [3].

3.2.3 Texture Features

Texture can be defined as homogeneous regions in images that do not result only from uniqueness in color but from identical structural arrangements or repetitive patterns within that region. For instance the bark of threes might have different colors between species or depending on the season but they form a same texture. Bricks, parquets or grass are other examples of textures. Detecting texture is important because they often provide strong semantic interpretation.

According to different surveys, Khokher and all [36], Haralick and all [26], texture features can be divided into two main categories : *Structured Approach* and *Statistical Approach*. In the structured approach the image is seen as a set of primitive texels with a regular or repeated pattern. The more widely used statistical approach is based on the distribution of gray level in the image. According to Robert M. Hawlick [26] statistical approach can be divided between height different techniques : autocorrelation functions, optical transforms, digital transforms, textural edgeness, structural elements, spatial gray tone cooccurrence probabilities, gray tone run lengths, and autoregressive models. A explanation for each of these techniques is provided in the survey. Wavelet-based features have also received wide attention. In the work by Minh N. Do and all [17] wavelet features are used in combination with Kullback-Leibler distance for texture retrieval. Gabor filters have also been investigated for texture features [23].

3.3 Shallow methods

Shallow method refer here to techniques used in order to extract meaningful representation using features descriptors described above. Shallow is used in opposition to deep methods that involve several layers of features extraction.

3.3.1 Visual Bag-of-Word

The visual bag-of-words method in computer vision is analogous to the bag-of-words model for document. For a document the bag-of-word is a vector (i.e. histogram) that contain the number of occurrence of words that are defined by a vocabulary. For an image a bag of word is a vector or histogram that contain the number of occurrence of visual words. Visual words correspond to representative vectors computed from local image features. The main steps of the method are :

Local features Extraction For this step different detectors such as harris detector or SIFT can be used and have been effectively investigated.

Encoding in a Codebook From the local features codewords (analogous to words) have to been found that will produce the codebook (analogous to a dictonary). The simplest method is to perform a k-means clustering over the entire features with cluster centers corresponding to codewords. Other methods to cluster the vector space are Square-error partitioning algorithms or Hierarchical techniques.

Bag of keypoints construction Once the codebook has been determined we can construct for each image an histogram (called here bag of keypoints) that contain the number of vectors assigned to each codewords (i.e. cluster centers).

More detailed explanation about the bag-of-words model can be found in the paper by Csurka and all [13] that introduced the method in 2004. Since then more encoding methods such as locality-constrained linear encoding, improved Fisher encoding, super vector encoding or kernel codebook encoding have been proposed to improve the model.

3.3.2 Improved Fisher Kernel

The Fisher Kernel was introduced by Jaakkola and Haussler [33]. The Fisher Kernel combines the benefits of generative and discriminative approaches by deriving a kernel from a generative model of the data. In the case of images it consist in characterizing local patch descriptors by its deviation from a Gaussian Mixture Model. Thus the Improved Fisher Kernel extend the BoW representation by including not only statistical counts of visual words but also additional informations about the distribution of descriptors. However in practice results obtained with the Improved Fisher Kernel has not been better that with BoW model [47]. Comparison of these two shallow methods can be found in the work of Chatfield and all [6].

Chapter 4

Convolutional Neural Network

The alternative to handcrafted features that is gaining huge attention in the recent years is the possibility to let the machine learn the best features to represent the image. In the domain of image processing the learning of the features is mainly done through a model called Convolutional Neural Network (CNN). Convolutional neural networks are models inspired by how the brain works and how neurons interact between each other. They are made of successive layers of neurons, each of them aggregating the results from the preceding layer in order to compute more abstract features. The succession of layer is the reason why such algorithms are know as deep learning methods. Convolutional neural network was inspired by the neocognitron proposed by Kunihiko Fukushima in the 1980s [20]. It was the first attempt to mimic the animals visual cortex. Early paper [30] had shown that the cortex of animals was made of different neurons. The neurons located downstream in the visual recognition process are responsible for the extraction of specific patterns within their receptive fields and neurons located upstream in the process aggregate these results and are invariant to specific location of the patterns. Accordingly the neocognitron consist of multiple types of cells, the most important of which are called S-cells and C-cells. The purpose of S-cells is to extract local features which are integrated gradually and classified in the higher layers. However the neocognitron lacked a proper supervised training algorithm. This algorithm was found by Yann LeCun and is known as backpropagation, abbreviation for backward propagation of errors. Backpropagation used in conjunction with an optimization method such as gradient descent has been successfully applied in 1989 to recognize handwritten digit recognition [38]. With the rise of efficient GPU computing a second breakthrough was achieved in 2012 by Geoffrey Hinton and his team which designed a convolutional neural network known as AlexNet [37] that achieved state-of-the-art performance on the Imagenet dataset.

As opposed to handcrafted features convolutional neural networks has the advantages to require very little if not preliminary knowledge of the domain at hand. For instance, when using a Visual Bag-Of-Words representation for images, one has to choose between different detectors which one will be the more efficient for the task. This choice depend obviously of the underlying problem. As for convolutional neural networks the domain knowledge is used to finely tune the parameters of the model but it is also often achieved through trial and errors. Prior knowledge can also be put into the network by designing appropriate connectivity, weights constraints and neuron activation function. In any case it is less intrusive than handcrafted features. Another benefits of convolutional neural network is their ability to learn more discriminative features. Both in the review of Chatfield and all [7] and in the study of Zhang and all [62] they show better performance than traditional shallow methods. Drawbacks of convolutional neural networks include their complexity, the substantial amount of data and the computational resources required to train it effectively.

4.1 Design

The complexity of convolutional neural network reside mostly in their design and in the different layers that the network is made of. The figure 4.1 illustrate a very simple convolutional neural network. It is made of the succession of two convolutional layers with pooling layers and is followed by a fully connected layer. As for the neocognitron the first layers that is the convolutional layers and the pooling layers are responsible for the extraction of local features. Specifically the convolutional layers learn to recognize different patterns and the pooling layers gradually ensure that the network is invariant to the location of these patterns. Fully connected layers enable to map this low level features to more abstract concepts. Below is a detailed description of the different layers one can find in a network.

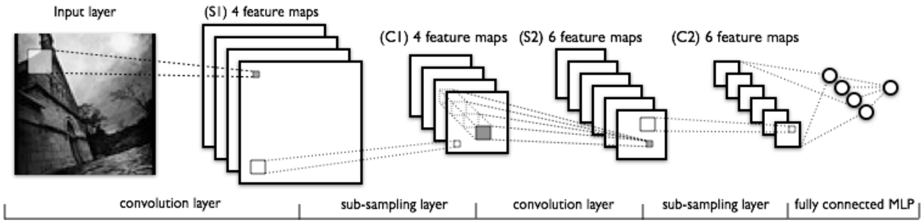

Figure 4.1: A Convolutional Neural Network

4.1.1 Convolutional Layers

Convolutional layers are the core of convolutional neural networks. The figure 4.2 represent a convolutional layer made of four learnable linear filters or kernels. Each perform a convolution on the preceding layer. That is each filter is slid across the width and the height of the preceding layer. The output of the convolution of one filter is called a feature map. Formally we have :

$$h^k_{i\,j} = (W^k * x)_{ij} + b^k$$

where h^k denote the k-th feature map, W^k and b^k the associated filter and bias and x the inputs.

It results of this operations that all neurons in a feature map share the same weights but have different receptive fields that is they look at different regions from the preceding layer. Thus neurons in a same feature map act as replicated features detectors able to detect a pattern at any location. Adjacent neurons in different features map share the same receptive field but the different filters learning different weights each feature map learn to detect different patterns. The exact nature of the patterns detected is hard to know and might not be easily interpretable for humans.

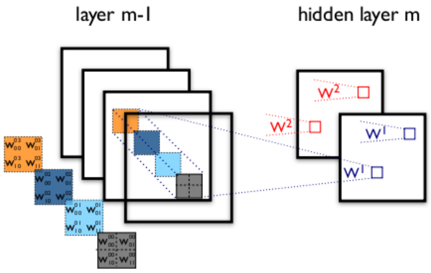
layer m-1
hidden layer m
W^2
W^2
W^1
W^1
W^{03}_{00} W^{03}_{01} W^{03}_{10} W^{03}_{11}
W^{02}_{00} W^{02}_{01} W^{02}_{10} W^{02}_{11}
W^{01}_{00} W^{01}_{01} W^{01}_{10} W^{01}_{11}
W^{00}_{00} W^{00}_{01} W^{00}_{10} W^{00}_{11}

Figure 4.2: A Convolutional Layer

Following the convolution each neuron is the input to an activation function that determine the final output of the layer. Different activation functions can be used such as the binary function, the logistic function, the hyperbolic tangent function but most commonly used activation function is the rectifier function. This function is defined as $f(x) = max(0, x)$ and has been argued to be the more biologically plausible [22].

Using convolutional layers before fully connected layers present many advantages. First it enable to take the spatial structure of the image into account. In fully connected layers neurons are joined to all the neurons of the previous layers thus they treat pixels which are far apart and close together exactly the same way whereas neurons in convolutional layers are only joined to neighboring neurons from the preceding layers. Also it enable to greatly reduce the number of parameters to learn by the network. For example if an image is of size 200*200*3 a single fully connected layer would have to learn $200*200*3 = 120,000$ weights whereas a single convolutional layer with 50 different filters of size 5*5*3 only has to learn $50*5*5*3 = 3750$ weights.

4.1.2 Pooling Layers

Pooling layers are common between two convolutional layers. They perform a form of non-linear down-sampling. They partition the previous layer into, overlapping or not, rectangular regions and for each region compute a unique output. It enable to get a small amount of translational invariance each time they are used.

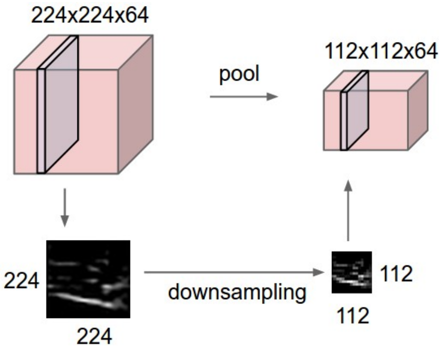

Figure 4.3: A Pooling Layer

Most of the time the function used by pooling layer is max-pooling that compute the maximum of the underlying region but average-pooling (compute the average of the underlying regions) or L2-norm pooling can also be used. By pooling the exact locations of the features are lost but relative locations, which are the most important, are preserved. Pooling layers are useful to progressively reduce the spatial size of the feature maps, thus the number of parameters and prevent overfitting.

4.1.3 Local Response Normalization Layer

Local Response Normalization Layer can be found especially in the AlexNet network [37]. it performs a normalization between neurons from adjacent kernels. The idea is to suppress the activity of a hidden neuron if nearby neurons in adjacent feature maps have stronger activities. Indeed if a pattern is detected with low intensity it becomes not so relevant if strong patterns are detected around. The authors argue that it helps to increase the performance of the network however another study [51] report that performance improvement was rarely achieved by adding local response normalization layer and that it even lead to to increased memory consumption and computation time. More generally this kind of layer is not reused in subsequent works.

4.1.4 Fully ConnectedLayers

The following layers of a convolutional neural network are the ones that we can find in a traditional multi-layer perceptrons that is one or more successions of fully connected layers. Fully Connected Layers are called this way because the neurons of these layers are connected to all the neurons of the layer below. In a convolutional neural network the feature maps of the last convolutional layer is vectorized and fed into fully connected layers. As for convolutional layer an activation function is used. If we suppose that rectifier units are used then we formally have :

$$y_j = max(0, z_j) \quad with \quad z_j = b + \sum_i W_{ij} * x_i$$

where y_j is the output of the j-th neurons of the layer, W is the weights matrix for this layer and b is the associated bias.

4.1.5 Loss Layer

The loss layer specifies how the network penalize the deviation between the predicted and true labels and is obviously the last layer in the network. The layer usually implements the softmax function that take a features vector as input and force the outputs to sum to one so they can represent a probability distribution across discrete mutually exclusive classes :

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

The cost function associated with the softmax function is the cross-entropy function :

$$E = -\sum_j t_j * log(y_j)$$

where t_j is the target value and y_j is the predicted value.

4.2 Training

The training of a network is done through the backprogation algorithm which will be explained shortly. One problem, due to the fact that convolutional neural network are able to learn complex mathematic model, is that they can easily overfit. To prevent this from happening one can either limit the number of hidden layer and the number of units per layer or stop the learning early but several more advanced methods have been proposed that we will discuss thereafter.

4.2.1 Backpropagation and Gradient Descent

Gradient descent is an optimization algorithm to find the local minimum of a function. In the case of machine learning it enable to find the minimum of the cost function by applying the delta rule :

$$W_{ij} = W_{ij} - \alpha * \nabla E(W_{ij})$$

where W_{ij} is the weight between the node i and j, α is the learning rate and $\nabla E(W_{ij})$ is the gradient of the cost function with respect to the weight W_{ij}.

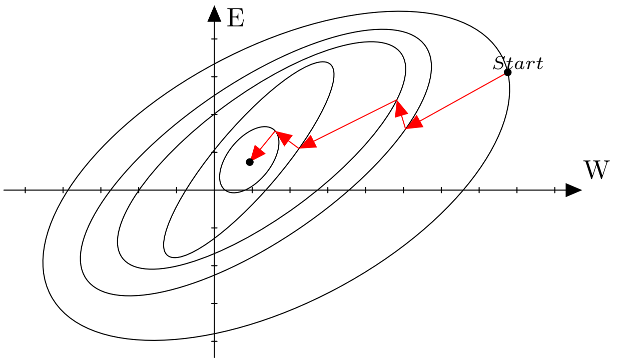

Figure 4.4: Illustration of gradient descent.

The idea is that $\nabla E(W_{ij})$ give us the direction of the steepest increase of the cost function so by iteratively updating the weights in the direction of the negative gradient we should reach the minimum of the cost function as illustrated by the figure 4.4.

The sensitive part with convolutional neural network is to compute the gradients which is achieved thanks to the backpropagation algorithm. Below is a sketch of the algorithm for fully connected layers where we assume that the cost function is the sum of squares :

$$E = \frac{1}{2} * \sum_j (t_j - y_j)^2$$

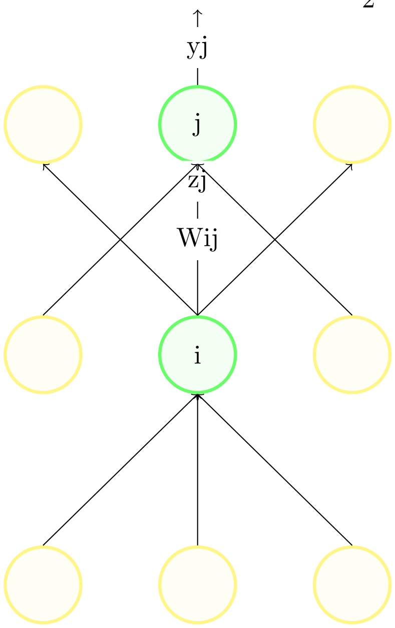

Figure 4.5: Fully Connected Layers

The goal is to find $\frac{\mathrm{d}E}{\mathrm{d}W_{ij}}$ so we can inject it into the delta rule. For this we make use of the chain rule :

$$\frac{\mathrm{d}E}{\mathrm{d}W_{ij}} = \frac{\mathrm{d}y_j}{\mathrm{d}W_{ij}} * \frac{\mathrm{d}E}{\mathrm{d}y_j}$$

$\frac{\mathrm{d}E}{\mathrm{d}y_j}$ can easily be found from the cost function :

$$\frac{\mathrm{d}E}{\mathrm{d}y_j} = -(t_j - y_j)$$

To find $\frac{\mathrm{d}y_j}{\mathrm{d}W_{ij}}$ we once again make use of the chain rule :

$$\frac{\mathrm{d}y_j}{\mathrm{d}W_{ij}} = \frac{\mathrm{d}z_j}{\mathrm{d}W_{ij}} * \frac{\mathrm{d}y_j}{\mathrm{d}z_j}$$

$\frac{\mathrm{d}y_j}{\mathrm{d}z_j}$ will depend on the activation function used. If we assume a logistic activation function where $y_j =$

$\dfrac{1}{1+e^{-z_j}}$ then $\frac{\mathrm{d}y_j}{\mathrm{d}z_j} = y_j(1-y_j)$. Also we easily find $\frac{\mathrm{d}z_j}{\mathrm{d}W_{ij}} = x_i$ so that we finally have :

$$\frac{\mathrm{d}E}{\mathrm{d}W_{ij}} = -x_i * y_j(1-y_j)(t_j - y_j)$$

The final equation can finally be injected into the delta rule enabling the network to be trained. Obviously the above demonstration will depend on the chosen cost function, the kind of layer used and the activation function. Also gradient descent is much of the time used in conjunction with the momentum method that enable faster convergence.

4.2.2 L1 and L2 regularization

L1 and L2 regularization are a common way of preventing the overfitting of the network. They both consist to introduce a extra term in the cost function that is going to penalize large weights. The L2 regularization is the most commonly used and is also known as *weight decay.* The extra term in this case is the sum of the squares of all the weights in the network scaled by a factor. So the new error function is :

$$E' = E + \frac{\lambda}{2} * \sum_{ij} W^2_{i\,j}$$

The effect of such a regularization is to make the network prefers to learn small weights and allow for large weight only if they considerably improve the cost function. Promoting small weight has been proven to be effective to reduce overfitting.

4.2.3 batch normalization

Batch normalization [32] potentially helps the training in two ways : faster learning and higher overall accuracy. When dealing with machine learning normalization of the data is often performed before the training process in order to make the data comparable across features. One problem with neural network is that during the training process the weights of the network fluctuates and that inputs of upper layers are affected by the weights of the precedents layers. Therefore the distribution of inputs for upper level can end up being distorted. This problem is referred by the authors as *internal covariate shift.* Their solution is to normalize the inputs data of each layers for each training mini-batch.

4.2.4 Dropout

A typical way to reduce overfitting when doing machine learning is to combine the results of several models. However with large neural network training many models can be very tedious due to the amount of training data required, the computational resources... Dropout is a technique proposed by Geoffrey Hinton and al [28] [53] to prevent large network from overfitting and is equivalent to combine the result of many models. The term dropout refers to dropping out temporarily neurons along with their incoming and outgoing connections from the network. Each neuron is assigned a fixed probability p independent of each other and for each training case each neuron is retained with that probability p. Hence for each training case a different network will be trained. It prompts neurons to not make up for the errors made by the other neurons but to make good predictions on their own. At test time the outgoing weight of each neuron is multiplied by the probability p assigned to it which is a way to approximate the the predictions of the different models that have been trained.

4.3 Research Trends

As mentioned earlier the first convolutional neural network to show great performance was AlexNet [37]. The overall architecture of AlexNet was composed of five convolutional layers and three fully connected layers. Three max-pooling layers was also used as well as layers they called Local Response Normalization Layers. Since AlexNet many other attempts to improve the performance of CNN have been made which mainly focus on varying the number of layers used as well as the size of the kernel filters. Simonyan and all [51] for instance have fixed all the parameters of the network other than the depth of the network (the number of layers) and have shown that the deeper the network the better the performance achieved. However a bigger size usually means more parameters which can become problematic regarding the use of computational resources, the time of training as well as the overfitting. This problem is addressed by Szegedy and all [56] whose the study focus on the efficiency of convolutional neural network by using sparsely connected architectures. Apart from increasing the depth and the width of convolutional neural networks other methods have been explored, some of them being presented below.

4.3.1 Convolutional Network Architecture

MCDNN MCDNN stands for Multi-Column Deep Neural Network and is an implementation of a neural network proposed by Ciresan and all [10]. Each column of their network is actually a convolutional neural network and the final classification is made by aggregating the results of these networks. Different training strategies ensure that each column produce a different result.

Network In Nework Lin and all argue in their study [39] that the level of abstraction obtained with linear filters is low. Therefore they substitute linear filters in the convolutional layers by what they call micro neural network and are actually multilayer perceptron. Also, fully connected layers often being responsible for overfitting, they propose a strategy called global average pooling to replace them. The idea is for the last convolutional layer to have as many feature maps that the number of category on which the network is trained. For each feature map the average is computed and the resulting vector is fed directly into the softmax layer. The final network is tested on four benchmark datasets (CIFAR-10, CIFAR-100, SVHN and MNIST) and achieved the best performances on all of them.

R-CNN In this work by Girshick and all [21] convolutional neural network is used for object recognition. Their main contribution is to show that convolutional neural network can be used successfully not only for image classification but also for object recognition. They use the following approach : first for an image they generate category-independent region proposals through selective search, then for each region they compute a features vector using a CNN and finally they classify each regions with category-specific linear SVMs. An other contribution is to show that a first training on a large dataset (ILSVRC) followed by a domain-specific fine-tuning on a small dataset is an effective way to train a network. Since their system combine region proposals with CNNs, they called their method R-CNN: Regions with CNN features. It has improved previous performance on the object recognition challenge of Pascal Voc.

4.3.2 Supervised Pre-training

Besides testing different architectures several studies have investigated how well features extracted from convolutional neural networks pre-trained on a large dataset such as ImageNet can be used to discriminate between unseen classes of other datasets. In the work of Donahue and all [18] they show by different experiments that features extracted from a pre-trained model actually generalize well to unseen classes. The pre-trained model follow the architecture of the AlexNet model [37] and is trained on the ILSVRC-2012 dataset. Their first experiment is to extract features from a new dataset called SUN-397 then run a dimensionality reduction algorithm (t-SNE algorithm) to find a 2-d representation. By then plotting the points which are colored depending on their semantic category we can observe good clustering. The second experiment is made on the Caltech-101 dataset. Again they extract features thanks to their pre-trained model and use different classifiers (SVM and logistic regression) to discriminate between objects. They show that the performance obtained with the SVM classifier outperform methods with traditional hand-engineered image features. In others experiments, domain adaptation, subcategory recognition and scene recognition they show again that features from a pre-trained deep model outperform traditional hand-engineered image features. Supervised pre-training is also explored by Wan and all [59]. As for Donahue and all they pre-train a AlexNet model with the ImageNet dataset and test its performance on other datasets agains visual bag-of-words and GIST features. In this case features from the network frequently, but not always, outperform traditional features. They also experiment to retrain the network on the new datasets that is they start a new training on the new dataset but with the weights initialized with the values learned from the pre-training. In this case the model always outperform traditional features.

Chapter 5

Features Interpretation

Once features have been extracted by the way of some techniques described in the previous chapter one need to interpret these features in order to achieve end applications. What I mean by interpret is that based on the features one of the following questions, depending on the problem at hand, has to be answered : "Can we identify an object in the Image ?", "What is the class of the image ?", "Are these two image similar ?" . . .

5.1 Similarity Measure

An obvious objective in CBIR is to assess the similarity of a query image with images in database. Different similarity measure have been investigated for this. Also algorithms that are able to learn a similarity measure from training data have been developed. The choice of the right metric to use is of course dependent of the problem at hand but also of the technique that was used to extract the features. Indeed depending on which techniques was used the representation of the image will differ. In the survey of Datta and all [16] they call these different representations signatures and they discern between three types of signature :

Single Vector The image is represented as a single features vector. It is for instance the result of using a global shape descriptor.

Set of Vectors The image is represented as a set of vectors and is usually the result of features computed on different regions obtained by segmentation of the image.

Histogram The image is represented as an histogram. This can be the result of the shallow methods introduced previously.

5.1.1 Fixed Measure

Basic metrics for computing the similarity between two features vectors include the minkowski distance or the cosine similarity. However when computed based on low-level features such metrics often fail to concur the distance computed and the semantic similarity. An attractive measure for measuring the distance between two probability distribution is the Earth Mover Distance (EMD) introduce by Rubner and all in 2000 [49]. The EMD is based on the minimal cost that must be paid to transform one distribution into the other and matches perceptual similarity better than other distances. Another metric is the Kullback–Leibler Distance used for instance by Do and all [17] to compute the similarity from waveket-based texture features.

5.1.2 Similarity Learning

Rather that using a fixed measure learning another approach is to rely on machine learning to learn a similarity measure. Three common setups exist :

Regression similarity learning In this approach pairs of image are given (I_i^1, I_i^2) with their associated measure of similarity y_i and the goal is to learn a function that approximate $f(I_i^1, I_i^2) = y_i$.

Classification similarity learning In this approach pairs of image are given (I_i^1, I_i^2) with binary labels $y_i \in \{0, 1\}$. The goal is to learn a classifier able to judge new pairs of images.

Ranking similarity learning This time training samples consist to triplet (I_i, I_i^+, I_i^-) where I_i is know to be more similar to I_i^+ than to I_i^-.

An example of ranking similarity learning for large scale image learning is the OASIS algorithm [8] that model the similarity function as a bilinear form : Given two images $I_1 and I_2$ the similarity is measured through the form $I_1 * W * I_2$ where matrix W is not required to be positive or symmetric.

5.2 Ontology

Object Ontology can be used to map low-level features to high level entities. An illustration of an object ontology is show by the figure 5.1. With this example a concept such as sky could be defined as having a blue color, a upper position and an uniform texture. This way by extracting color, texture and other features from an images different regions can be mapped to concepts. Object ontology is used by Hyvonen and all [31] to access pictures of the promotional events of the university of Helsinki. Top-level ontological categories represent concepts such as Persons, Events, Places and pictures are mapped to theses entities when inserted into the database. In the work by Mezaris and all [41] the images are first segmented then color and shape descriptors are extracted enabling the mapping of each regions to a corresponding concept.

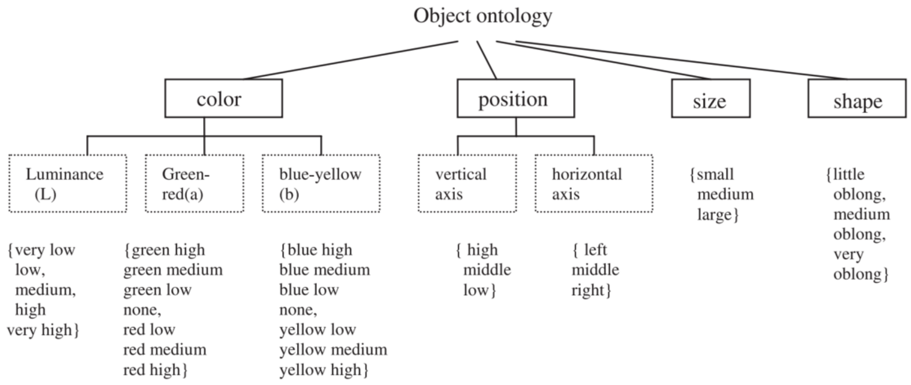
Object ontology
color
position
size
shape
Luminance (L)
Green-red(a)
blue-yellow (b)
vertical axis
horizontal axis
{small medium large}
{little oblong, medium oblong, very oblong}
{very low low, medium, high very high}
{green high green medium green low none, red low red medium red high}
{blue high blue medium blue low none, yellow low yellow medium yellow high}
{ high middle low}
{ left middle right}

Figure 5.1: Object Ontology

5.3 Machine Learning

A last approach to associate features with high-level concept is to rely on supervised learning methods. A good candidate which has proven to show strong performance is the support vector machine (SVM) classifier. SVM has originally been designed for binary classification so that we need to train multiple models if we want to achieve multi-classification. The idea behind SVM is to find a hyperplane that best divide the features belonging to two separate classes. Among the possible hyperplane a common one is the *optimal separating plane* which maximize the distance between the hyper-plane and the nearest data point of each class. SVM has often been used in conjunction with the Visual Bag-Of-Word representation to classify images. This is the case for example for the work of Csurka and all [13] or the one of Lowe and all [40]. Another common classifier is the naive bayes classifier. It relies on the baye's theorem so that prior probabilities and class-conditional densities for the features have to be computed from the training sample. In the work of Vailaya and all [57] they use first and second order moments in the LUV color space as color features and MSAR texture features. Then they perform vector quantization with the learning vector quantization (LVQ) algorithm and the result is fed as input to the naive bayes classifier. This scheme is used to classify indoor / outdoor and city / landscape image. Other classifier that have been investigate are neural network and decision trees.

Chapter 6

Evaluation of CBIR System

In order to evaluate between the different systems that have been proposed for Content-Based Image Retrieval researcher need common images databases with trustworthy ground truth and well defined metrics. We will review here some of the well established dataset used for this purpose as well as the different tasks that are evaluated.

6.1 Datasets

6.1.1 Pascal Voc

The Pascal Visual Object Classes (VOC) Challenge consists of two components : a publicly available dataset and an annual competition. The dataset consist of annotated consumer photographs collected from the flickr photo-sharing website. In total 500,000 images were retrieved from flickr and divided between 20 classes. For each of the 20 classes images were retrieved by querying flickr with a number of related keywords and randomly choosing an image among the 100,000 first results. The process was repeated until sufficient images were collected. In order to evaluate the detection challenge a bounding box was further added for every object in the target set of object classes.

6.1.2 Caldech

The goal of the Caldech dataset is to provide with relevant images for performing multi-class object detection. Caldech 101 dataset provide pictures of object belonging to 101 categories with most of the categories having 50 images. Thus the resulting training is relatively small compared to other datasets. Each image contains only a single object. A common criticism of this dataset, is that the images are largely without clutter, variation in pose is limited, and the images have been manually aligned to reduce the variability in appearance. Caldech 256 correct some disadvantages of the previous dataset by more than doubling the number of class and introducing a large number of clutter images.

6.1.3 ImageNet

ImageNet is an image database organized according to the WordNet hierarchy that is each meaningful concept is possibly described by multiple words or word phrases and is called a *synonym set* or *synset*. ImageNet is comprised of more of 100,000 synsets with on average 1000 images for each synset. The ImageNet dataset has been created especially for deep learning methods that need huge amount of training data.

6.2 Tasks Evaluated

6.2.1 Image Classification

One common task is to discern between different classes which one correspond to a given image. It is called Image Classification and is often relying on the presence or not of a specific object in the image such as a car, a plane, a bicycle and so on. Performing such task can be useful to answer to a query of the type "Find pictures with a red car". To achieve Image Classification a commonly used approach is to compute local features of the image, summarize them into an histogram which is given as input to a classifier [19]. This approach is know as bag-of-visual-word in analogy with the bag-of-words (BOW) used for text representation. Within the approach different features extractors (SIFT descriptor, Harris descriptor...) and different classifiers (SVM, Earth Mover's Distance kernel...)have been investigated [13] [62] [40]. New trends also perform classification using Convolutional Neural Network who achieved best performance on several benchmarks especially on the ImageNet database.

6.2.2 Object Detection

Object Detection consist to assess if an object is present in a given image and to identify its location. A very widespread method to achieve Object Detection is to use a sliding window on the image. Features are computed from the window and given to a classifier to compute evaluate if the object is present. The window is slid throughout the image at different scale and for each scale and location the classifier is applied [58] [14].

6.2.3 Image Similarity

Image Similarity purpose is to assess if images, commonly stored in a database, are similar to a query image. The most similar images can then be returned in answer to the query. Image similarity is typically performed by reverse search engines. The process to achieve Image Similarity is first to extract features from the images. Then a similarity measure is used to compute the similarity between the images. Similarity measure can be computed with distance metrics such as euclidean distance or with more advanced techniques relying on machine learning to learn a similarity function [8].

6.2.4 Multi-class Image Segmentation and Labeling

Multi-class Image Segmentation and Labeling consist in assigning to each pixel a class label. First step here is usually to perform a segmentation on the image and to aggregate similar pixels into groups. Secondly a label is chosen for each group. Probabilistic graphical models have been successfully applied for this kind of task.

6.3 Evaluation Metrics

6.3.1 Precision and Recall

Probably the most common evaluation measures used in information retrieval are precision and recall. Precision is the fraction of retrieved items that are relevant to the query :

$$precision = \frac{|relevantdocuments \cap retrieveddocuments|}{retrieveddocuments}$$

Recall is the fraction of relevant items that are retrieved :

$$recall = \frac{|relevantdocuments \cap retrieveddocuments|}{relevantdocuments}$$

Precision and recall are often presented through a precision versus recall graph. Based on these two metrics several other have been derived that bring additional informations and make up for their inadequacy in some cases. For instance when performing retrieval with huge quantity of documents recall is not any more a relevant metric as the query might have thousand of relevant documents.

Precision at k Precision at rank k correspond to the number of relevant results in the first k documents.

Average Precision For one query average precision is the average of the precision computed over an interval rank.

Mean Average Precision For a set of queries mean average precision is the mean of the average precision scores for each query.

F-score The F-score is a weighted average of the precision and recall defined as $2 * \frac{precison * recall}{precision + recall}$ where 1 is the best value and 0 the worst.

6.3.2 Top-k Error

The Top-k error rate is useful to evaluate classification tasks. It is defined as the fraction of items where the correct label is not among the k labels considered the most probable by the model. In the Imagenet classification challenge the top-1 and the top-5 error rates are used as benchmarks to compare the different submissions.

6.3.3 Confusion Matrix

To evaluate the performance of a classification system a confusion matrix, also known as a contingency table, is often used. The name *confusion* come from the fact that the matrix enable us to easily check if the system is confusing different classes. A confusion matrix compute the rate of true positive and true negative also know as sensitivity and specificity. For a binary classification test sensitivity measure the proportion of positives that are correctly identified as such while specificity measure the proportion of negatives that are correctly identified as such.

Bibliography

[1] Mohamed N Ahmed, Sameh M Yamany, Nevin Mohamed, Aly A Farag, and Thomas Moriarty. A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *Medical Imaging, IEEE Transactions on*, 21(3):193–199, 2002.

[2] Agus Zainal Arifin and Akira Asano. Image thresholding by histogram segmentation using discriminant analysis. In *Proceedings of Indonesia–Japan Joint Scientific Symposium*, pages 169–174, 2004.

[3] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[4] EA Bashkov and NS Kostyukova. Effectiveness estimation of image retrieval by 2d color histogram. *Journal of Automation and Information Sciences*, 8(11):74–80, 2006.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[6] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, volume 2, page 8, 2011.

[7] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[8] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11:1109–1135, 2010.

[9] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen. Fuzzy c-means clustering with spatial information for image segmentation. *computerized medical imaging and graphics*, 30(1):9–15, 2006.

[10] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[11] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[12] Michel Crucianu, Marin Ferecatu, and Nozha Boujemaa. Relevance feedback for image retrieval: a short survey. *Report of the DELOS2 European Network of Excellence (FP6)*, 2004.

[13] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[15] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40, 2008.

[16] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):5, 2008.

[17] Minh N Do and Martin Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *Image Processing, IEEE Transactions on*, 11(2):146–158, 2002.

[18] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[19] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[20] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[22] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[23] Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on gabor filters. *Image Processing, IEEE Transactions on*, 11(10):1160–1167, 2002.

[24] Venkat N Gudivada and Vijay V Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[25] James Hafner, Harpreet S Sawhney, Will Equitz, Myron Flickner, and Wayne Niblack. Efficient color histogram indexing for quadratic form distance functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(7):729–736, 1995.

[26] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[27] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[28] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[29] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3):245–268, 1999.

[30] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[31] Eero Hyvönen, Samppa Saarela, Avril Styrman, and Kim Viljanen. Ontology-based image retrieval. In *WWW (Posters)*, 2003.

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[33] Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[34] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jeff Donahue, and Sarah Tavel. Visual search at pinterest. *arXiv preprint arXiv:1505.07647*, 2015.

[35] Charles E Kahn Jr. Artificial intelligence in radiology: decision support systems. *Radiographics*, 14(4):849–861, 1994.

[36] Amandeep Khokher and Rajneesh Talwar. Content-based image retrieval: Feature extraction techniques and applications. In *Conference proceedings*, 2012.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[38] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[39] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[40] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[41] V Mezaris, I Kompatsiaris, and MG Strintzis. Ontologies for object-based image retrieval. In *Proc. Workshop Image Analysis For Multimedia Interactive Services*, pages 96–101, 2003.

[42] Henning Müller, Nicolas Michoux, David Bandon, and Antoine Geissbuhler. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 73(1):1–23, 2004.

[43] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: Querying images by content using color, texture, and shape. *Storage and Retrieval for Image and Video Databases*, 1908, 1993.

[44] Richard Nock and Frank Nielsen. Statistical region merging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1452–1458, 2004.

[45] Ron Ohlander, Keith Price, and D Raj Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8(3):313–333, 1978.

[46] Greg Pass and Ramin Zabih. Comparing images using joint histograms. *Multimedia systems*, 7(3):234–240, 1999.

[47] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[48] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 817–824. ACM, 2009.

[49] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[50] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[54] Markus A Stricker and Markus Orengo. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pages 381–392. International Society for Optics and Photonics, 1995.

[55] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[57] Aditya Vailaya, Mário AT Figueiredo, Anil K Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, 2001.

[58] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[59] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM International Conference on Multimedia*, pages 157–166. ACM, 2014.

[60] Chee Sun Won and Haluk Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP: Graphical Models and Image Processing*, 54(4):308–328, 1992.

[61] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, 1993.

[62] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

[63] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *Medical Imaging, IEEE Transactions on*, 20(1):45–57, 2001.

[64] Xiang Sean Zhou and Thomas S Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.