Pattern Recognition 45 (2012) 346–362

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A review on automatic image annotation techniques

Dengsheng Zhang*, Md. Monirul Islam, Guojun Lu

Gippsland School of Information Technology, Monash University, Churchill, Vic. 3842, Australia

ARTICLE INFO

Article history:
Received 8 October 2010
Received in revised form
3 May 2011
Accepted 24 May 2011
Available online 31 May 2011

Keywords:
Image retrieval
Machine learning
Semantic gap
Image annotation
Colour
Texture
Shape

ABSTRACT

Nowadays, more and more images are available. However, to find a required image for an ordinary user is a challenging task. Large amount of researches on image retrieval have been carried out in the past two decades. Traditionally, research in this area focuses on content based image retrieval. However, recent research shows that there is a semantic gap between content based image retrieval and image semantics understandable by humans. As a result, research in this area has shifted to bridge the semantic gap between low level image features and high level semantics. The typical method of bridging the semantic gap is through the automatic image annotation (AIA) which extracts semantic features using machine learning techniques. In this paper, we focus on this latest development in image retrieval and provide a comprehensive survey on automatic image annotation. We analyse key aspects of the various AIA methods, including both feature extraction and semantic learning methods. Major methods are discussed and illustrated in details. We report our findings and provide future research directions in the AIA area in the conclusions

1. Introduction

Due to the explosive growth of digital technologies, ever increasing visual data are created and stored. Nowadays, visual data are as common as textual data. There is an urgent need of effective and efficient tool to find visual information on demand. A large amount of research has been carried out on image retrieval (IR) in the last two decades. In general, IR research efforts can be divided into three types of approaches. The first approach is the traditional text based annotation. In this approach, images are annotated manually by humans and images are then retrieved in the same way as text documents [9,10,15,16]. However, it is impractical to annotate a huge amount of images manually. Furthermore, human annotations are usually too subjective and ambiguous. The second type of approach focuses on content based image retrieval (CBIR), where images are automatically indexed and retrieved with low level content features like colour, shape and texture [11–13,41–47]. However, recent research has shown that there is a significant gap between the low level content features and semantic concepts used by humans to interpret images. In addition, it is impractical for general users to use a CBIR system because users are required to provide query images. The third approach of image retrieval is the automatic image annotation (AIA) so that images can be retrieved in the same way as text documents [17–40,115,116]. The main idea of AIA techniques is to automatically learn semantic concept models from large number of image samples, and use the concept models to label new images. Once images are annotated with semantic labels, images can be retrieved by keywords, which is similar to text document retrieval. The key characteristic of AIA is that it offers keyword searching based on image content and it employs the advantages of both the text based annotation and CBIR. There are several surveys on broad CBIR research in literature [2–7,127], and a survey on broad semantic IR techniques is given by Liu et al. [1]. However, none of them gives sufficient attention to AIA which is a new development in IR. In this paper, we focus our review on this emerging trend in IR, so as to complement existing surveys in literature. Specifically, we focus on the two major aspects of AIA, feature extraction and semantic learning/annotation.

The rest of the paper is organised as follows. In Section 2, image segmentation and low level feature extraction are described. In Section 3, various AIA techniques using machine learning are discussed in details. Section 4 summarises and concludes the survey.

2. Feature extraction and image representation

In image classification and retrieval, images are represented using low level features. Because an image is an unstructured array of pixels, the first step in semantic understanding is to extract efficient and effective visual features from these pixels. Appropriate feature representation significantly improves the

* Corresponding author.
E-mail addresses: Dengsheng.Zhang@monash.edu (D. Zhang), Md.Monirul.Islam@monash.edu (M.M. Islam), Guojun.Lu@monash.edu (G. Lu).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.05.013

performance of the semantic learning techniques. While both global and region based image representations are used in the existing image retrieval techniques, the trend is towards using region based features. Region based feature extraction needs prior image segmentation while global features are directly calculated from the whole image. In the following, we first briefly review common image segmentation algorithms used in AIA techniques. Then, various feature extraction techniques will be reviewed in detail.

2.1. Image segmentation

Image segmentation is usually the first step to extract region based image representation. The segmentation algorithm divides images into different components based on feature homogeneity. A number of segmentation approaches exist in the literature, such as grid based, clustering based, contour based, model based, graph based, and region growing based method. This section provides a brief review of segmentation methods commonly used in AIA. For a comprehensive segmentation review, readers are referred to [128].

Because automatic image segmentation is a difficult task, many techniques simplify this task using grid based approach to roughly segment images into blocks [18,20,23,25–27,29,59,67]. Visual features are then extracted from these blocks. Block based approach takes little computations; however, this simple technique does not describe the semantic components in images well. A single block often consists of parts of visually different objects. Furthermore, it is difficult to determine the size of blocks for image representation. Therefore, region features are usually not accurate. If appropriately applied, it can be used in domain specific applications, e.g., medical image classification [14].

Clustering algorithms, like *k*-means, are used to cluster pixels into different groups [8,19,62,63], with each group identifying a region. In most cases, an image is first partitioned into blocks of size 4×4 pixels. Colour and/or texture features are extracted for each block. Then, *k*-means is applied to cluster the block feature vectors. A region is formed with the pixels belonging to blocks of the same cluster. The major issue with this approach is that it needs to predefine the number of segments based on heuristics. An inappropriate choice of *k* may yield poor results. The other issue is that the algorithm assumes data are in spherical clusters so that the mean values are near the cluster centres. This assumption, however, is usually not true.

The main idea of contour based segmentation is to evolve a curve around an object. The evolution stops when the curve coincides with the boundary of an object. Unlike the cluster based segmentation algorithm, contour based segmentation algorithms do not need the prior assumption of the number of clusters [68–70]. The underlying problem in this approach is the dependency on accurate edge detection which is subject to noise. Therefore, it often needs human to define rough boundary outline which makes the approach to be applicable only to specific domain, e.g., image processing tools.

Segmentation algorithms based on statistical models have also been proposed [71–73]. Among them, Blobworld [71] is widely used [60,74,75]. In Blobworld, each pixel is represented by an 8 dimensional feature vector of colour, texture and position. An image pixel is then modelled as a random variable with Gaussian mixture distributions. The number of regions and Gaussian parameters are calculated using *expectation maximisation* (EM) algorithm. Once the model parameters are found, the pixel–region relationship is calculated using the posterior probabilities. The pixel–region relationship is used to determine the image segmentation. One of the major issues with this approach is that the computation is very expensive because the EM is an optimisation algorithm.

Shi and Malik [76] propose a graph based segmentation algorithm known as *normalised cut* (NCut). The NCut method represents an image as a graph where vertices are image pixels and the edge weights represent the feature similarities between pixels. Image segmentation then becomes a graph partitioning problem. The idea is to partition the vertices of the graph into disjoint sets so that the total similarity between different sets is minimised. Each set is regarded as region. As the number of pixels in an image is large, there are exponential numbers of possible partitions of the graph. As a result, it is computationally expensive to find the optimal partition. Tao et al. [77] improves the NCut by pre-segmenting images using mean shift algorithm [72]. Instead of using pixels, the regions of the initial segmentation are used as vertices in the NCut algorithm. Hence, the computational cost is reduced, and the performance is more robust. The basic NCut is based on colour features only. Malik et al. [78] extend it to incorporate texture features.

The widely used JSEG [79] algorithm is a region growing approach. It groups pixels or smaller regions into larger regions. At first, pixel colours of the image are quantised into a number of classes and pixels in the image are replaced with the colour class labels. A class map is formed and region growing is followed on the class map. Pixels with more homogeneous neighbours are assumed to be interior pixels of possible regions. These pixels are selected as candidate seed points and regions are grown around these seed areas. As this method looks for both colour and texture homogeneity, the segmented regions have highly homogeneous characteristics. It has been widely used in image retrieval [37,47,74,80].

Image segmentation is a complex issue and a large research topic. Segmentation performance usually depends on applications. For image retrieval purpose, the region boundary does not have to be accurate as long as the region is homogenous. However, regions from segmentation are usually contaminated with segments from neighbouring regions. This problem can be overcome by a clean-up post-processing [47].

2.2. Colour features

Colour is one of the most important features of images. Colour features are defined subject to a particular colour space or model. A number of colour spaces have been used in literature such as, RGB, LUV, HSV, HMMD [6,81–84].

Once the colour space is specified, colour feature is extracted from images or regions. A number of important colour features have been proposed in the literatures, including colour histogram [11,85], colour moments (CM) [86], *colour coherence vector* (CCV) [87], colour correlogram [41], etc. MPEG-7 [82] also standardizes a number of colour features including *dominant colour descriptor* (DCD), *colour layout descriptor* (CLD), *colour structure descriptor* (CSD), and *scalable colour descriptor* (SCD).

Colour moments are one of the simplest features. They are used in many retrieval systems [20,24,25,56,57,86]. The common moments are mean, standard deviation and skewness. Usually they are calculated for each colour channels (components) separately. Therefore, nine features form the feature vector. These features are useful when they are calculated for region or object. However, the moments are not enough to represent all the colour information of an image.

The colour histogram describes the colour distribution of an image [55–57,86]. It quantises a colour space into different bins and counts the frequency of pixels belonging to each colour bin. This feature is robust to translation and rotation changes. However, a colour histogram does not tell pixels' spatial information. Therefore, visually different images can have similar colour histograms. In addition, the dimension of a histogram is usually very high.

The *colour coherence vector* (CCV) incorporates spatial information into the basic colour histogram. It divides each histogram bin into two components: coherent and non-coherent parts. The coherent component includes those pixels which are spatially connected. The non-coherent component includes those pixels that are isolated. As CCV captures spatial information, it usually performs better than a colour histogram. However, the dimension of a CCV is twice of a conventional histogram.

A colour correlogram is the colour version of *grey level co-occurrence matrix* (GLCM). It characterises the distribution of colour pairs in an image [29,30,41]. A colour correlogram can be treated as a 3D histogram where the first two dimensions represent the colours of any pixel pair and the third dimension is their spatial distance [41]. Thus, in a correlogram, each bin (i, j, k) represents the number of colour pair (i, j) at a distance k. The colour correlogram is calculated for horizontal distance $k=1$. Correlograms for other distances can be similarly calculated. The performance of the colour correlogram is better than both histogram and CCV because it captures both intensity levels and spatial patterns in an image. However, it is much more complex due to high dimensionality and multiple matrix processing.

Among MPEG-7 colour descriptors, the *scalable colour descriptor* (SCD) is a histogram based descriptor. SCD is basically a histogram in HSV colour space [59]. It differs from the conventional histogram by scalability. The scalability is achieved in two ways: (1) reducing the number of colour bins with Haar transform and (2) removing some least significant bits from the quantised (integer) representations of bin values. However, experimental results show that such down scaling significantly affects the retrieval performance [82]. Furthermore, the descriptor does not include any spatial information. Therefore, it has similar problem to the conventional histogram.

The Colour structure descriptor (CSD) is also a histogram based descriptor [88]. The CSD histogram is created by moving a structuring element (e.g., square) throughout the image. Bin i of the histogram indicates how many times the structuring element contains at least one pixel with colour i. If the window is of size 1 pixel, the CSD is an ordinary histogram. The performance of CSD depends on the size and structure of the window, which are difficult to determine. Furthermore, it is computationally more expensive than SCD.

The Dominant colour descriptor (DCD) is also a variation of histogram [24,37]. DCD selects a small number of colours from the highest bins of a histogram. The number of colours (bins) selected as DCD depends on the threshold of bin height. MPEG-7 recommends that 1–8 colours are sufficient to represent a region. Unlike the traditional histogram, the selected colours in DCD are adapted to the region instead of being fixed in the colour space. Thus, the colour representation with DCD is more accurate and compact than the conventional histogram. However, the similarity or distance calculation of two DCDs needs many-to-many matching.

Among the various colour features, colour moments are not sufficient to represent the regions. On the other hand, histogram based descriptors are either too high dimensional or too expensive to compute. DCD is a good balance between the two extreme. It has been shown that DCD is sufficient to represent the colour information of a region [47]. In addition, the feature dimension of DCD is low and the computation is relatively inexpensive. Colour features such as CCV, colour correlogram and CSD are useful for whole image representation, butthey all involve complex computation. Table 1 provides a summary of different colour methods.

Table 1
Contrast of different colour descriptors.

Colour method	Pros	Cons
Histogram	Simple to compute, intuitive	High dimension, no spatial info, sensitive to noise
CM	Compact, robust	Not enough to describe all colours, no spatial info
CCV	Spatial info	High dimension, high computation cost
Correlogram	Spatial info	Very high computation cost, sensitive to noise, rotation and scale
DCD	Compact, robust, perceptual meaning	Need post-processing for spatial info
CSD	Spatial info	Sensitive to noise, rotation and scale
SCD	Compact on need, scalability	No spatial info, less accurate if compact

2.3. Texture features

Texture is another important image feature. While colour is usually a pixel property, texture can only be measured from a group of pixels. Due to its strong discriminative capability, texture feature is widely used in image retrieval and semantic learning techniques. Texture has been well studied in image processing and computer vision area [89]. A number of techniques have been proposed to extract texture features. Based on the domain from which the texture feature is extracted, they can be broadly classified into spatial texture feature extraction methods and spectral texture feature extraction methods. In the following, we describe these techniques.

2.3.1. Spatial texture feature extraction methods

In spatial approach, texture features are extracted by computing the pixel statistics or finding the local pixel structures in original image domain. The spatial texture feature extraction techniques can be further classified as structural, statistical and model based.

Structural techniques describe textures using a set of *texture primitives* (*texon* or *texture elements*) and their placement rules [83,90,91]. Textons are organised into a string descriptor, and syntactical pattern recognition techniques are used to find similarity of two descriptors.

Statistical texture feature characterises texture as a measure of low level statistics of grey level images. The common spatial domain statistical features are moments [6,83], Tamura texture features [46,89,92,111] and features derived from *grey level co-occurrence matrix* (GLCM) [6,58]. Statistical features are compact and robust because they are derived from large support. However, they are not sufficient to describe the large variety of textures.

In model based techniques, texture is interpreted using stochastic (random) or generative models. Model parameters characterize the underlying texture property of the image. Popular texture models are *Markov random field* (MRF) [6,20,91,93,94,96,97], *simultaneous auto-regressive* (SAR) model [8], *fractal dimension* (FD) [91,95], etc. As these models involve optimisation, they are usually computationally expensive.

Spatial texture methods are easy to understand and many of them even have semantics. They do not require regular region shape and can be applied to irregular regions straightforwardly. However, these features are usually sensitive to noise and distortions. Furthermore, many of these methods involve complex search and optimisation processes which have no general solutions. Table 2 summarises different spatial texture methods.

2.3.2. Spectral texture feature extraction techniques

In spectral texture feature extraction techniques, an image is transformed into frequency domain and then feature is calculated from the transformed image. The common spectral techniques include *Fourier transform* (FT) [98,102], *discrete cosine transform*

Table 2
Contrast of different spatial texture methods.

Texture method	Pros	Cons
Texton	Intuitive	Sensitive to noise, rotation and scale, difficult to define textons
GLCM based method	Intuitive, compact, robust	High computation cost, not enough to describe all textures
Tamura	Perceptually meaningful	Too few features
SAR	Compact, robust, rotation invariant	High computation cost, difficult to define pattern size
FD	Compact, perceptually meaningful	High computation cost, sensitive to scale

Table 3
Contrast of different spectral texture methods.

Texture method	Pros	Cons
FT/DCT	Fast computation	Sensitive to scale and rotation
Wavelet	Fast computation, multi-resolution	Sensitive to rotation, limited orientations
Gabor	Multi-scale, multi-orientation, robust	Need rotation normalisation, losing of spectral information due to incomplete cover of spectrum plane [103]
Curvelet	Multi-resolution, multi-orientation, robust	Need rotation normalisation

(DCT) [99], wavelet [8,24,58], and Gabor filters [82,100,101,113]. FT and DCT are very fast to compute but are not scale and rotation invariant. Wavelet is both efficient and robust, but it only captures horizontal and vertical features. Among them, Gabor features are most robust because it captures image features in multi-orientations and multi-scales. Recently, researches on multi-resolution analysis have shown [103] that curvelet features have significant advantages over Gabor features and wavelet features, because curvelet features are more effective in capturing curvilinear properties, like lines and edges [48].

The problem with those spectral methods is that they can only be applied to square regions due to the use of FFT. Most of the existing region based techniques define a region as a set of small blocks of size 4 × 4 pixels and apply spectral transform on those blocks [8], because small blocks are likely homogenous. Features of a region are then calculated as the average features of those blocks. This method has a drawback that the blocks are too small to capture sufficient edge information. To solve this problem, recently Islam et al. [114] propose a texture padding method to transform an irregular texture region to a square texture region. This method also acquires sizable regions to extract meaningful texture features. Table 3 summarises different spectral texture methods.

2.3.3. Summary

Both spatial and spectral features have advantage and disadvantages. Spatial features can be extracted from any shape without losing information and usually have semantic meaning understandable by humans. However, it is difficult to acquire sufficient number of spatial features for image or region representation, and spatial features are usually sensitive to noise. Spectral texture features on the other hand are robust, and they also take less computation because convolution in spatial domain is done as product in frequency domain which is implemented using FFT [100]. However, they do not have the semantic meaning as spatial features usually have. For images or regions with sufficient size, spectral texture features are a desirable choice. However, for small images or regions, especially when the regions are irregular, spatial features should be considered.

2.4. Shape features

Shape is known to be an important cue for human to identify and recognise real world objects. Shape features have been employed for image retrieval in many applications. Zhang and Lu [104] broadly classify shape extraction techniques into two major groups: contour based and region based methods. Contour based methods calculate shape features only from the boundary of the shape, while region based methods extract features from the entire region. Because contour based techniques use only a portion of the region, they are more sensitive to noise than region based techniques, as small changes in the shape significantly affect the shape contour. Therefore, colour image retrieval usually employs region based shape features.

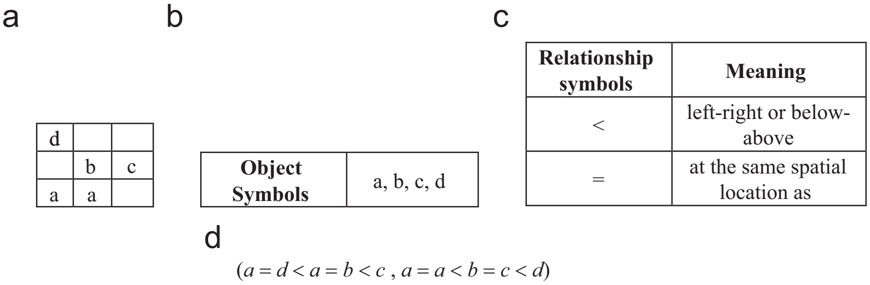
A number of simple region shape descriptors are commonly used in colour image retrieval, including, *area*, *moments*, *circularity*, and *eccentricity*. The area-based descriptor is used in a number of works [19,21,56,63,65]. *Circularity* and *moments* are used in [21,56,65]. Circularity measures the ratio of area to boundary. In [63], *eccentricity* or *elongation* is also used in addition to area. Eccentricity is the ratio of the length of the major axis to that of minor axis. Individual simple shape descriptors are not robust. Therefore, they are normally combined to create a more effective shape descriptor.

More complex shape features are usually used in domain specific applications such as trademark retrieval [105,106], and object classification [63,88,107],where shape is the most important feature. For example, Park et al. [88] use MPEG-7's contour shape descriptor and Liu et al. [107] use Fourier descriptor of shape contour for bird classification.

2.5. Spatial relationship

Spatial relationship tells object location within an image or the relationships between objects. Absolute spatial location of regions are used in [24,56]. Relative locations of regions, such as 'left, right, top, bottom, and centre' with respect to image itself, are used in [63] for ontology based concept learning.

In [108], the spatial relationship between regions is modelled using a 2D string. In a 2D string method, images are projected along the *x*- and *y*-axis. For each projection, the relationship between objects is represented by an array of symbols. The symbols are drawn from two sets: the set of object symbols and the set of relationship symbols, such as 'left/right', 'below/above'. A number of variations of this method have been proposed [109,110]. These approaches differ in the number of relational operators (symbols) and how they define those relations. Fig. 1 shows an example of a 2D string representation. The image in Fig. 1(a) is decomposed into regions (blocks). For simplicity, the block identifiers are used as object symbols. Two relationship symbols '$<$' and '$=$' are used in this case. In horizontal and vertical directions, the symbol '$<$' denotes 'left–right' and 'below–above' relationships, respectively. The symbol '$=$' means the spatial relationship 'at the same spatial location as'. A 2D string takes the form (u, v), where u and v are the relationships of objects in horizontal and vertical directions, respectively. Fig. 1(d) shows the 2D string for the image of Fig. 1(a).



Object Symbols	a, b, c, d

Relationship symbols	Meaning
<	left-right or below-above
=	at the same spatial location as

$(a = d < a = b < c\ ,\ a = a < b = c < d)$

Fig. 1. Illustration of a 2D string: (a) an image decomposed into blocks, (b) object symbols as block names, (c) definitions of relationship symbols, and (d) a 2D string for (a).

The 2D string and its variants can be used as global features for region based representation, provided the objects are well defined by the segmented regions. As segmentation algorithms often divide a single object into different fragments, the 2D string usually does not give accurate representation. In practice, the relative location of regions is usually used [63,112]. In [112], Islam et al. define a distance relationship model for object location. It is assumed that different types of objects are located at different positions in an image. For instance, cloud and bird are usually at the top of an image, people and animal are usually at the centre, water and grass are usually at the bottom etc. Therefore, objects can be differentiated based on their distance to their usual positions. A weight is defined based on the object's distance to its usual position. The weight is combined with other information such as colour, texture and shape to determine the object type.

3. Automatic image annotation techniques

Once images are represented with low level features using either global methods or region methods, higher level semantic can be learned from image samples. Early approach used relevance feedback to learn image semantic from humans [125,126]. However, this approach has similar drawbacks to the traditional manual annotation approach. Therefore, the new trend is towards automatic image annotations. Assuming semantically labelled image samples are collected and represented with low level features, a machine learning algorithm can then be trained using the feature to semantic label matching. Once trained, the algorithm can be used to annotate new image samples. There are generally three types of AIA approaches. The first approach is the single labelling annotation using conventional classification methods. The second approach is the multi-labelling annotation which annotates an image with multiple concepts using the Bayesian methods. The third approach is the web based image annotation which uses metadata to annotate images. In the following, we discuss the three types of approaches in detail.

3.1. Single labelling annotation using binary classification

In this approach, low level features are extracted from image content, and the features are fed directly into a conventional binary classifier which gives a yes or no vote. The output of the classifier is the semantic concept(s) which is used for image annotation. The idea of single labelling is equivalent to collective labelling, that is, instead of labelling images individually, images are first clustered and then labelled collectively. The common machine learning tools include support vector machines (SVM), artificial neural network (ANN), and decision tree (DT). In the following, we review each of these techniques.

3.1.1. Image annotation using support vector machines

Support vector machine (SVM) is a supervised classifier. It has been shown with high effectiveness in high dimensional data classifications, especially when the training dataset is small [118]. SVM can classify both linear and non-linear data due to the use of kernel mapping. The advantage of SVM over other classifiers is that it achieves optimal class boundaries by finding the maximum distance between classes. It has been successfully applied to a number of classification problems, such as text classification, object recognition and image annotation [23,55,57,59,75].

An SVM classifier works by finding a hyperplane from a training set of samples to separate them. Each training sample is represented with a feature vector and a class label. The hyperplane is learned in such a way that it can separate the largest portion of samples of the same class from all other samples. An SVM is basically a binary classifier. However, automatic image classification and annotation needs multiclass classifier. The most common approach is to train a separate SVM for each concept with each SVM generating a probability value. During the testing phase, the decisions from all classifiers are fused to get the final class label of a test image. Fig. 2 shows this process. The complete classifier is a two levels process. The base level consists of multiple binary classifiers and the second level fuses the decisions from the base level classifiers.

Chapelle et al. [55] use the above mentioned basic framework to train 14 SVM classifiers for 14 image level concepts. Images are represented with 4096 dimensional HSV histograms. To train an SVM for a particular concept, training images belonging to that concept are regarded as positive samples while the others are regarded as negative samples. Therefore, each trained classifier can be regarded as a 'one vs. all' classifier. During testing, each classifier generates a probabilistic decision. The class with maximum probability is selected as the concept of the test image. Despite simple histogram based image representation, this basic framework outperforms the *k-nearest neighbour* (*k*-NN) based annotation. Similar approaches are used in [23,75] to learn semantic concepts for image regions. In Shi et al. [75], images are segmented using *k*-means algorithms and 23 SVM classifiers are trained to learn 23 region level concepts. Cusano et al. [23] use SVM to do image segmentation and classification simultaneously. In this approach, an image is partitioned into overlapping tiles which are sampled at fixed interval. Each pixel is covered by a number of tiles. The tiles are classified independently into one of the seven predefined concepts. The concept of the pixel is determined by the majority voting from the classes of its parent tiles. Pixels belonging to the same concept constitute a segment. Thus this approach simultaneously segments images into regions and annotates the segmented regions.

The basic approach works well for small number of concepts. The quality of classification degrades with the increase of the number of concepts due to the increase of the noise and class

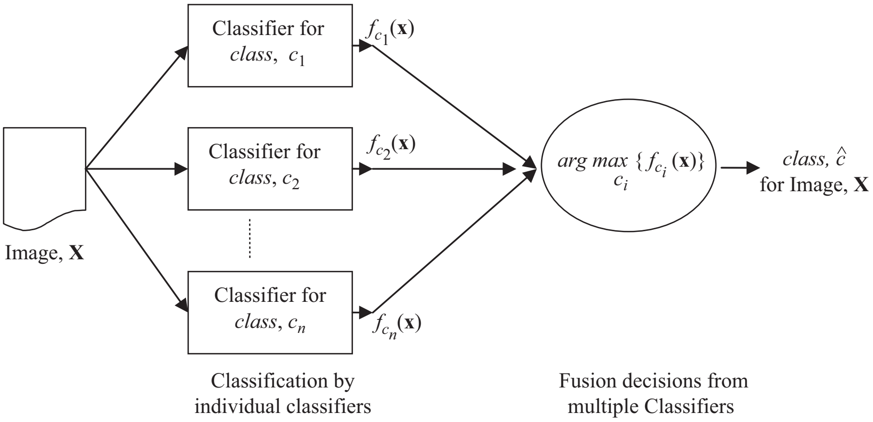

Classifier for class, c_1
Classifier for class, c_2
Classifier for class, c_n
$f_{c_1}(\mathbf{x})$
$f_{c_2}(\mathbf{x})$
$f_{c_n}(\mathbf{x})$
Image, X
$arg\ max_{c_i} \{f_{c_i}(\mathbf{x})\}$
class, $\hat{c}$ for Image, X
Classification by individual classifiers
Fusion decisions from multiple Classifiers

Fig. 2. Multiclass classifier using multiple binary SVM classifiers.

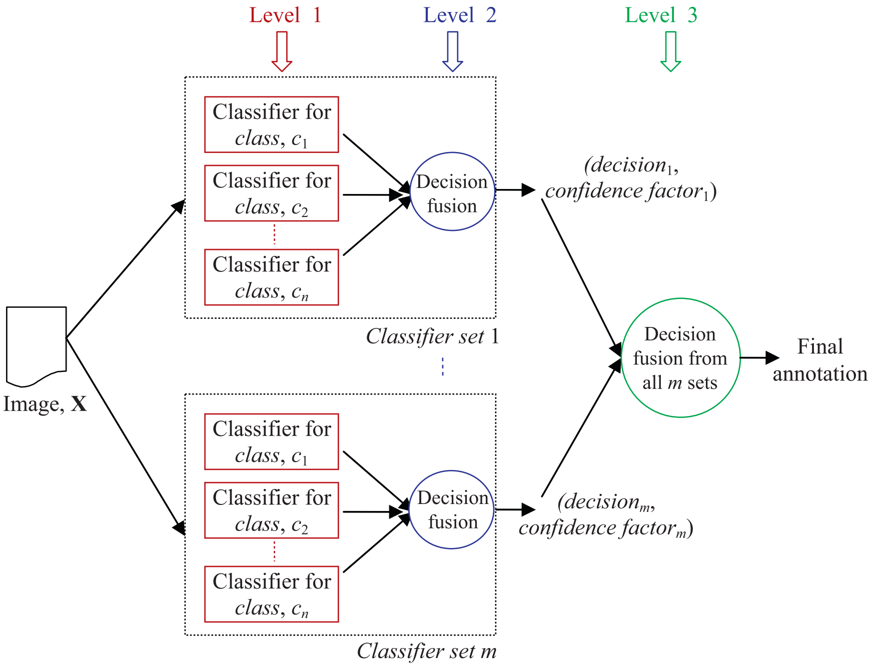

Level 1
Level 2
Level 3
Classifier for class, c_1
Classifier for class, c_2
Classifier for class, c_n
Decision fusion
(decision$_1$, confidence factor$_1$)
Classifier set 1
Image, X
Decision fusion from all m sets
Final annotation
Classifier for class, c_1
Classifier for class, c_2
Classifier for class, c_n
Decision fusion
(decision$_m$, confidence factor$_m$)
Classifier set m

Fig. 3. Image annotation with multiple sets of SVMs.

imbalance in the training data. To be more robust, several approaches use multiple sets of SVM classifiers as shown in Fig. 3. Each set of SVMs is similar to a multiclass classifier shown in Fig. 2. Each set of classifiers independently classifies an input image. The final decision is fused from the decisions of all sets. In the literature, there are a number of works that use this multi-level framework [57,59]. They differ by how the individual classifiers are trained at Level 1 and how the decisions are fused at subsequent levels.

Goh et al. [57] use the above 3-level approach (Fig. 3) to classify images into one of 116 concepts. During training, each set of classifiers is trained using different subset of training samples. At the time of annotation, they normalise the probabilistic outputs of Level 1 classifiers before using them in Level 2 fusion process. During the fusion process at Level 2, they also find a *confidence factor*, in addition to the highest probabilistic decision. The confidence factor is the function of both the highest probabilistic value and the difference between the highest two probabilistic decisions. At Level 3, the confidence factors of same concept are added together. The concept with the maximum cumulative confidence is the final decision. They show that the noisy output from a set of classifiers is compensated using the decisions of other sets of classifiers.

Qi and Han [59] use a similar framework to Goh et al. [57] but fuse the decisions in a different way as shown in Fig. 4. In this work, they use both global and local features in two different sets of SVMs. The same set of training sample is used to train both set of classifiers. The SVM set with global feature representation works in the same way as shown in Fig. 2. For the other set of SVM, each image is represented with the features of the region of interest. Instead of fusing the decisions set by set like Goh et al. [57], Qi and Han [59] fuse them classifier by classifier. Therefore, this approach can compensate the limitations of one type of feature by the other.

SVM has shown considerable performance in learning image annotations. The advantage is that SVM can learn from a small set

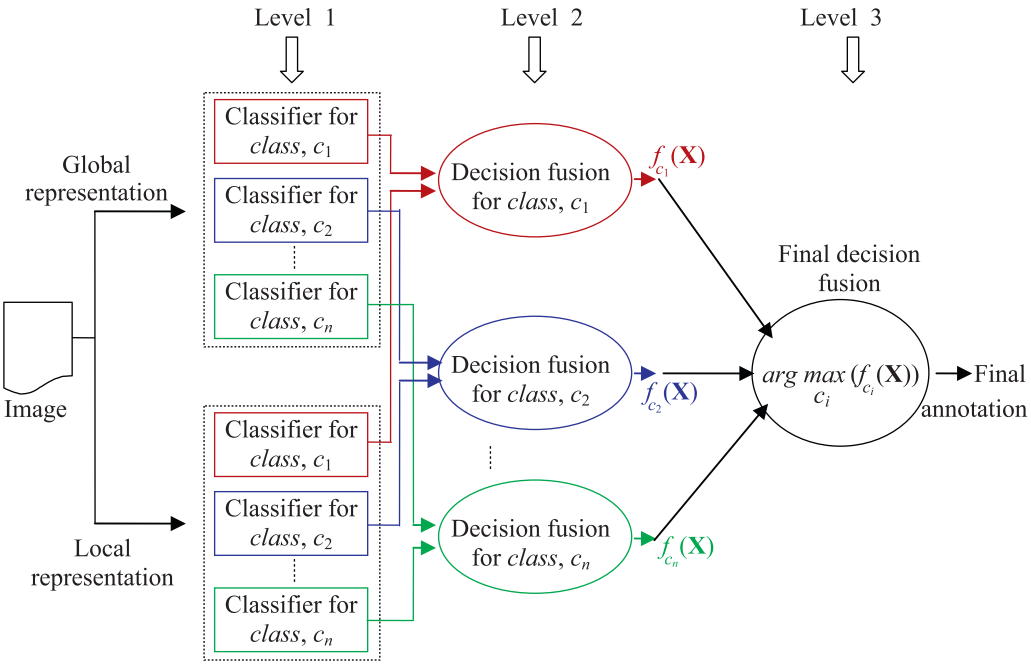

Level 1
Level 2
Level 3
Global representation
Local representation
Image
Classifier for class, c_1
Classifier for class, c_2
Classifier for class, c_n
Classifier for class, c_1
Classifier for class, c_2
Classifier for class, c_n
Decision fusion for class, c_1
Decision fusion for class, c_2
Decision fusion for class, c_n
$f_{c_1}(\mathbf{X})$
$f_{c_2}(\mathbf{X})$
$f_{c_n}(\mathbf{X})$
Final decision fusion
$\arg\max_{c_i}(f_{c_i}(\mathbf{X}))$
Final annotation

Fig. 4. Three levels multiclass classifiers used in [59].

of samples because it needs only the samples (known as 'support vectors') close to the separating hyperplane. However, SVM has class-imbalance problem, which means, it has poor performance on imbalanced data. Unfortunately, class imbalance is a common phenomenon in image data. For example, for a particular semantic, the number of negative samples is often much larger than the number of positive samples. Furthermore, the positive samples are relevant to each other but each negative sample often belongs to separate semantic groups. These problems degrade the quality of the classifiers.

3.1.2. *Image annotation using artificial neural network*

An Artificial neural network (ANN) is a learning network that can learn from examples and can make decision for a new sample. Different from common classifiers which usually learn one class at a time, ANN can learn multiple classes at a time. An ANN consists of multiple layers of interconnected nodes, which are also known as neurons or perceptrons. Therefore, an ANN is also called multilayer perceptron (MLP). The first layer is the input layer which has neurons equal to the dimension of input sample. The number of neurons in the output layer is equal to the number of classes. This means, an ANN can learn multiple classes at a time, although single class ANN is also available [19]. The choice of the number of hidden layers and the number of neurons at each hidden layer are open issues in ANN approaches [119]. These numbers are usually selected empirically. The connecting edges between neurons of different layers are associated with weights. Each neuron works as a processing element and is governed by an activation function which generates output based on the weights of the connecting edges and the outputs of the neurons at the previous layers. During the training, ANN learns the edge weights so that overall learning error is minimised. While classifying a new sample, each output neuron generates a confidence measure and the class corresponding to the maximum measure indicates the decision about the sample.

An ANN can be used both for explicit classification of images, regions or pixels [58,61,119], or implicit assignment of fuzzy decisions on images [62]. Frate et al. [119] use a 4-layer ANN to classify pixels of satellite images into one of the four categories: vegetation, asphalt, building, and bare soil. Based on the optimal experimental performance, they use a network of two hidden layers each consisting of 20 neurons. Kim et al. [61] classifies images into object and non-object images using a 3-layer ANN. Instead of segmenting an image, the centre 25% of the image is used to represent the image content. It assumes that the centre part significantly characterises the entire image. Because of this simplified assumption, the system cannot classify an image properly if the object appears in the other part of the image. A similar assumption is made in Park et al. [58] about the object importance. Park et al. [58] use segmentation algorithm to segment an image into regions and use the largest region at the centre of the image to identify the image. The regions with similar colour distribution of the central region are regarded as foreground (object) regions. The foreground regions are used to extract statistical texture features which are fed to a 3-layer ANN to classify the image into one of 30 concepts. The network consists of 49 neurons in the hidden layer. The drawback of the two approaches is that they may miss important objects from other parts of the image. For example, in a sunset/sunrise image, the sun often appears in the upper corner of the image. Furthermore, object regions may not necessarily be the largest region. In that case, the system will produce incorrect annotation.

Kuroda and Hagiwara [62] use four different 3-layer ANNs to hierarchically classify image regions. The numbers of neurons used in the hidden layers of the four networks are 30, 10, 20, and 20, respectively. Fig. 5 shows how the first network classifies an image region into one of the three broad categories such as *sky*, *water*, and *earth*. 37 dimensional region features are fed into the input layer. Each node of the output layer corresponds to one of the classes, e.g., *sky*, *water*, and *earth* and produces a likelihood value. The class of the input region is determined by the maximum likelihood value. *Sky* and *earth* regions are further classified into more specific classes using the other two ANNs. For example, a *sky* region is classified into one of five detailed categories: *blue sky*, *cloud*, *sunset*, *night*, and *light*. Similarly,

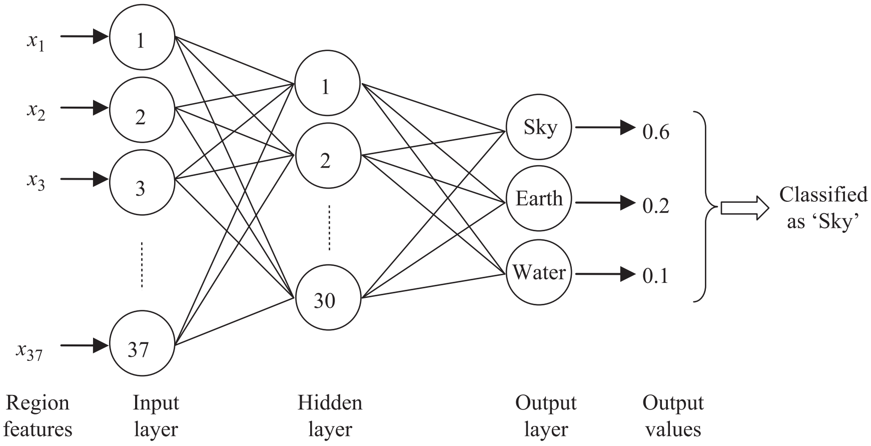


Fig. 5. Classifying a region using ANN.

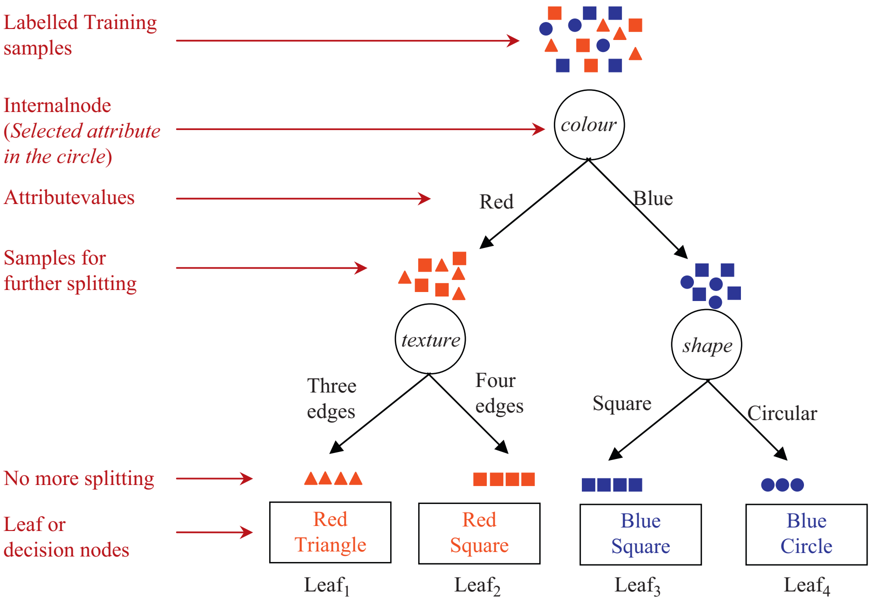


Fig. 6. Decision tree learning.

an *earth* region is classified into one of nine more specific categories. The fourth ANN does not classify any region. Instead, it associates an image with a vector of 18 dimensions: each dimension measures the degree of certain global characteristics of the image, for example, *bright/dark*, *rural/urban*, *busy/plain*, etc.

The neural network has the advantage that the outputs of output layer neurons are determined by the previous layers and the connecting edges. It does not need any other parameter tuning or any assumption about the feature distribution. However, there are several essential issues with ANNs. First, the classification accuracy depends on the number of hidden layers and neurons. Second, in most ANN research works, the numbers of hidden layers and neurons are not justified. Third, the choice of appropriate activation functions for the neurons is also an issue. Fourth, the training (finding the optimal edge weights) takes long time and it can fall into local optima. Fifth, an ANN works like a black box which means that the exact relation between the input and output is not transparent and is difficult to explain [19].

3.1.3. *Image annotation using decision tree*

A decision tree (DT) is a multi-stage decision making or classification tool [120,121]. Depending on the number of decisions made at each internal node of the tree, a DT can be called binary or *n*-ary tree. Different from other classification models whose input–output relationships are difficult to describe, the input–output relationship in a DT can be expressed using human understandable rules, e.g., *if–then* rules.

A DT is trained using a set of labelled training samples. Samples are represented with a number of attributes. During training, a DT is built by recursively dividing the training samples into non-overlapping sets, and every time the samples are divided, the attribute used for the division is discarded. The procedure continues until all samples of a group belonging to the same class or the tree reaches its maximum depth when no attribute remains to separate them. Fig. 6 shows the process. The tree has two types of nodes: internal and leaf node. Each internal node is associated with a decision governed by a certain attribute which divides the training samples most effectively. Each leaf node represents the outcome (class) of the majority samples that follow the path from the root of the tree to the corresponding leaf. The leaf nodes can be expressed with unique if–then–else rules. For example, the decision tee of Fig. 6 can also be expressed using the if–then rules of (1):

$$\begin{cases} \text{if } \textit{colour} = \text{Red and } \textit{texture} = \text{Three edges} & \\ \quad \text{then } \textit{outcome} = \text{Red Triangle} & - \ \text{Leaf}_1 \\ \text{if } \textit{colour} = \text{red and } \textit{texture} = \text{four edges} & \\ \quad \text{then } \textit{outcome} = \text{red square} & - \ \text{leaf}_2 \\ \text{if } \textit{colour} = \text{blue and } \textit{shape} = \text{square} & \\ \quad \text{then } \textit{outcome} = \text{blue square} & - \ \text{leaf}_3 \\ \text{if } \textit{colour} = \text{Blue and } \textit{shape} = \text{circular} & \\ \quad \text{then } \textit{outcome} = \text{blue circle} & - \ \text{leaf}_4 \end{cases} \qquad (1)$$

To label a new sample, the tree is traversed from the root node to a leaf node using the attribute value of the new sample. The decision of the sample is the outcome of the leaf node where the sample reaches. Several DT algorithms are used in the literature, including ID3 [120], C4.5 [121], and CART [122]. These DTs differ by the type of attributes, the attribute selection criteria, the outcome, etc.

ID3 is the simplest DT algorithm that works only with discretized attributes. On the other hand, C4.5 and CART can work with both discretized and continuous attributes. In ID3 and C4.5, the number of children of an internal node is equal to the number of attribute values of the selected attribute at that node. On the contrary, CART always splits an internal node into two children. Therefore, CART usually generates a bigger tree and takes more time than ID3 or C4.5. While ID3 and C4.5 are only used for classification, CART is used for both classification and numerical prediction. The contrast of the 3 major DT algorithms is given in Table 4.

Sethi and Coman [123] use CART to classify outdoor images into four classes. For image representation, they partition each component of HSL colour space into 8 intervals. Each of the 24 ($=3\times 8$) intervals is used as an attribute. Thus, each image is represented with 24 attributes. However, it is found that 9 attributes have been used by the algorithm. This happens because the number of concepts is small and database images have very little variation. Wong and Leung [124] use the DT algorithm to annotate scenery images into 10 classes. Instead of traditional visual features, they use image acquisition parameters, such as aperture, exposure time, focal length, etc. as attributes. As these attributes are continuous valued, they use C4.5 to learn decision rules for mapping these attributes to image semantics.

Apart from single tree based classification, Marée et al. [14,27] use ensemble of multiple DTs for image annotation and classification. Each DT is similar to the classical DT shown in Fig. 6. But, each of them is trained differently. During the splitting of an internal node, an attribute is randomly chosen instead of selecting the best one. The tree built in this way is called a random tree. A set of random trees is generated in this way. During classification, a testing sample is passed through all random trees and a majority vote is used to determine the class label of the image.

Recently, Liu et al. [37] use weighted average of colour and texture features for DT induction. Unlike the above mentioned approaches which can only classify images globally, they use DT to annotate regions of segmented images. One of the major features of this work is the design of both pre-pruning and post-pruning techniques to train a well behaved DT. However, the number of concepts learned in this work is small. Zhang et al. [112] improve Liu et al.'s method using vector quantisation for feature discretization. Furthermore, they learn much larger number of semantic concepts and index images using an inverted file to facilitate text based image retrieval.

Compared to other learning methods, DT is simple to interpret and understand and can learn with small number of samples. It is also robust for incomplete and noisy data [120]. DT usually requires discretized feature values as inputs [37,120]. Though C4.5 and CART can work with continuous features, they perform poorer compared to discrete features [37]. Another issue with DT is that C4.5 [121] and CART [122] are designed for single valued features. They do not work for high dimensional vectored features. Recently, Liu et al. [37] proposed a semantic template based feature discretization for image data. However, this technique is only useful when the underlying feature vectors have the same dimension. Therefore, a more robust feature discretization technique is necessary which can discretize variable dimension feature vectors such as DCD [82].

3.1.4. Summary

In single labelling annotation approaches, once images are classified into different categories, each category is annotated with a concept label such as plane, mammal, building etc. Retrieval of image categories is straightforward by just typing keywords related to the concept labels. The advantage of this type of approach is that the retrieval is efficient as there is no need to do image indexing and expensive online matching as in other IR approaches. The disadvantage of this type of approach is that it does not consider the fact that many images belong to multiple categories. As a result, many relevant images can be missed from the retrieval list if a user does not type the exactly right keyword. One way to alleviate this problem is to label each category with multiple keywords reflecting different themes within the category. Another issue with the single labelling annotation is that images within each category are not ranked, leading to reduced retrieval accuracy.

3.2. Multi-labelling annotation using Bayesian methods

Different from the binary classification approaches in last section, multiple labelling methods annotate an image with multiple semantic concepts/categories. The concept of multi-labelling approach is related to the multi-instance learning, or

Table 4
Contrast of three major DT algorithms.

Algorithm	Attribute type	Split criterion	Tree structure	Pros	Cons
ID3	Discrete	Information gain	Multi-tree	More interpretable rules	Attribute bias Over-fitting, samples in memory
C4.5	Discrete and continuous	Gain ratio	Multi-tree	Attribute balance, no over-fitting, allow missing attribute values	Tall tree, need to test multiple attribute values
CART	Discrete and continuous	Gini coefficient	Binary tree	More balanced tree, numerical prediction	Less interpretable than multi-tree

more specifically, multi-instance multi-label (MIML) learning [129,130]. In MIML, an image is represented with a bag of features or a bag of regions (multiple instances). The image is annotated with a concept label if any of the regions/instances in the bag is associated with the label. As a result, an image is annotated with multiple labels. A typical MIML is achieved using probabilistic tools such as the Bayesian methods [18,20,56,60,65]. The Bayesian methods work by finding the posterior probability that an image belongs to any particular concept, given the observation of certain features from the image or region. This makes it possible to assign an image to multiple concepts and rank images with the same concept according to the probabilities. Given a set of images $\{I_1, I_2, \ldots, I_N\}$ from a set of given semantic classes $\{c_1, c_2, \ldots, c_n\}$, Bayesian models try to determine the posterior probability from the conditional probabilities and the priors. Suppose, an image I is represented by the feature vector $\mathbf{x}$. Given prior probabilities $p(c_i)$ and conditional probability densities $p(\mathbf{x}|c_i)$, the probability of an unknown image I belonging to class c_i is determined by (2):

$$p(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i)p(c_i)}{p(\mathbf{x})} \tag{2}$$

From Eq. (2), it can be seen that a Bayesian framework essentially has four components: one output component $p(c_i|\mathbf{x})$ and three input components: $p(c_i)$, $p(\mathbf{x}|c_i)$, and $p(\mathbf{x})$. Because the distribution $p(\mathbf{x})$ is usually uniform for all classes, the class of image I can be decided using the *maximising a posterior* (MAP) criterion,

$$\hat{c} = \arg\max_{c_i} p(c_i|\mathbf{x}) \approx \arg\max_{c_i}\{p(\mathbf{x}|c_i)p(c_i)\} \tag{3}$$

The critical part of Bayesian annotation is to model the conditional probabilities because prior probabilities $p(c_i)$ can be found by the frequency of samples belonging to concept c_i. The variety of Bayesian models differ by how they model the conditional probabilities $p(\mathbf{x}|c_i)$. There are generally two types of approaches to model the conditional probabilities—non-parametric approach and parametric approach.

3.2.1. Non-parametric approach

In this approach, the conditional probabilities are calculated without any prior assumption about the distribution of the image features. Rather, the actual feature distribution is learned from the features of the training samples using certain statistics. In practice, the image features are first quantised into clusters using a certain clustering algorithm. Next, the continuous features are replaced by the cluster centroids. This process discretizes the image feature space. The conditional probabilities for each class are calculated by finding the frequency of samples belonging to that class. For example, if the closest centroid of feature vector $\mathbf{x}$ is $\mathbf{x}_j$, the $p(\mathbf{x}|c)$ in Eq. (2) can be calculated as,

$$p(\mathbf{x}|c) \approx p(\mathbf{x}_j|c) = \frac{\text{No. of samples in } \mathbf{x}_j \text{ which are from concept, } c}{\text{Total no. of samples from concept, } c} \tag{4}$$

The complete annotation process of this approach is shown in Fig. 7. Given a new image, its features are extracted and compared with cluster centres. The closest cluster centres are selected. The conditional probability models corresponding to the selected cluster centres are then used to calculate the posterior probabilities. The MAP criterion of (6) is then used to annotate the new image.

This model is used in [20,60]. Vailaya et al. [20] use this model for vacation image classification. Instead of segmentation, they directly cluster global image features to calculate the conditional probabilities. In [60], training images are segmented into regions and regions are then clustered. For instance, an image I is segmented into regions and represented with their closest centroids, $\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\ldots,\mathbf{x}_n$. The conditional probability $p(I|c)$ is calculated as,

$$p(I|c) = p(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\ldots\mathbf{x}_n|c) = \prod_{k=1}^{n} p(\mathbf{x}_k|c) \tag{5}$$

In the work by Mori et al. [18], training images are divided into blocks and blocks are clustered. As shown in Fig. 8, each block inherits all the annotations of its parent images and each cluster is a collection of concepts from all the blocks in it. The posterior probability $p(c|\mathbf{x}_j)$ is modelled as the *co-occurrence* of word c within cluster X_j,

$$p(c|\mathbf{x}_j) = \frac{\text{Total no. of annotation } c \text{ inherited into cluster } X_j}{\text{Total no. of all annotation words in } X_j} \tag{6}$$

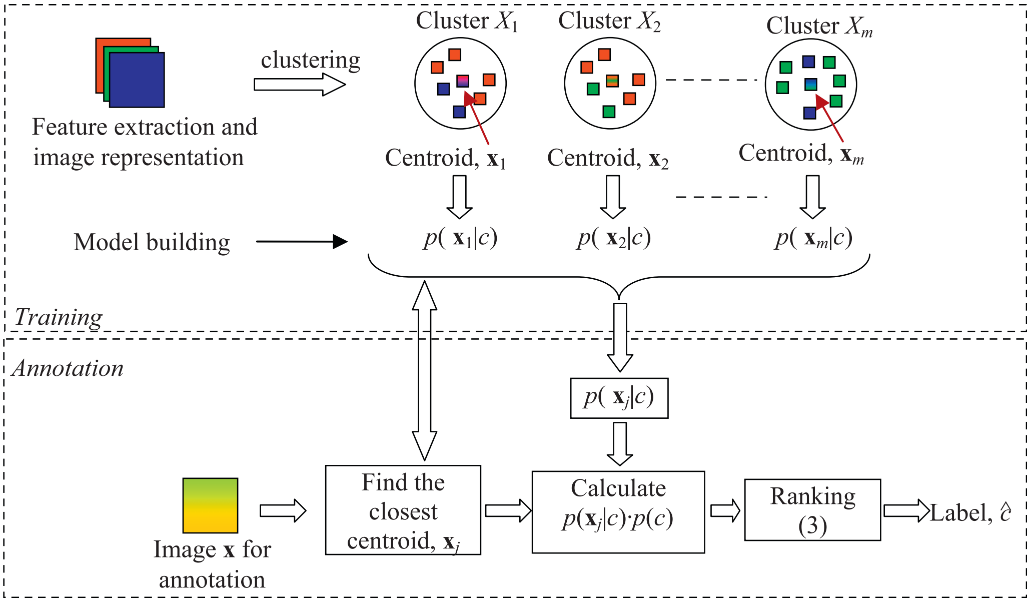


Fig. 7. The general Bayesian annotation model.

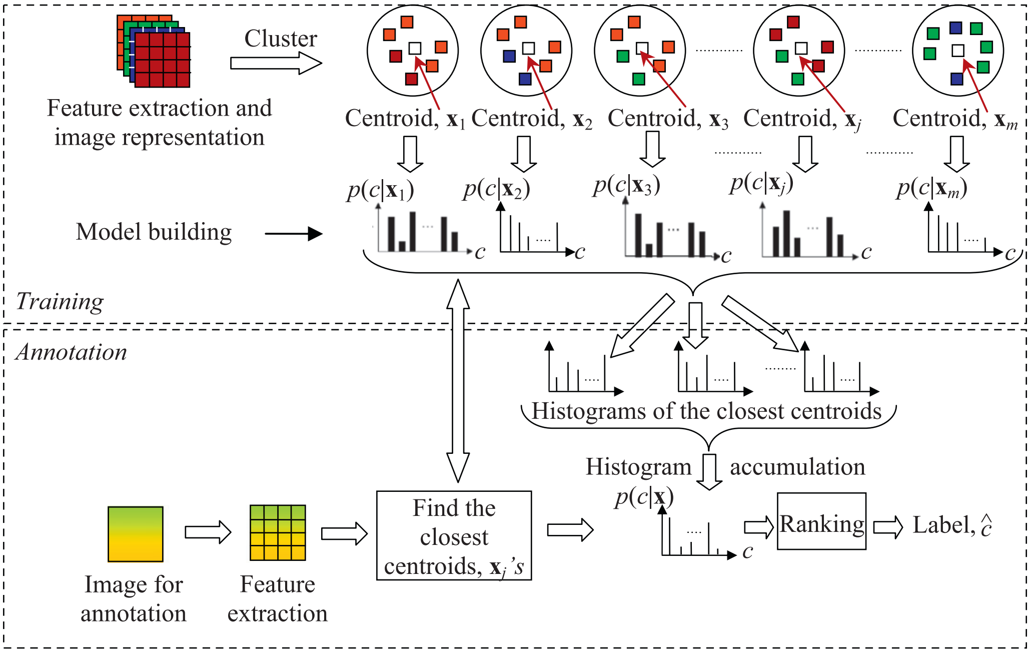


Fig. 8. The word co-occurrence model [18].

The posterior probability of each concept c_i is computed for cluster X_j. As a result, a *concept histogram* is created for each cluster centroid $\mathbf{x}_j$. During annotation, an unknown image is partitioned into blocks. For each block in the unknown image, the centroid with the closest feature to the block is selected. The histograms of the selected centroids are accumulated. The concepts with top likelihood values (highest bins in the accumulated histogram) are used as the annotations of the testing image. The advantage of this model is its simplicity. However, the idea of labelling regions/blocks using image concepts is flawed. For example, an image can have a multiple objects, like grass, horse, sky, water, etc. It is not appropriate for a sky block to inherit grass or horse, etc. This type of errors can propagate, making the system incapable of doing correct annotation. The other issue is that blocks are inaccurate to represent images.

In the above approaches, the clusters are used to compute the conditional probabilities which are then used for image annotations. Duygulu et al. [21], however, attempt to label the clusters by a learning process. Regions can then be annotated by relating to their closest clusters. They first cluster the regions from all training images. Regions are represented by the index of the closest centroid (blob). Next, they associate each blob with a word in the vocabulary by maximising the joint probability of associating each of the instances in the blob with the word in each of the images. As this is a one-to-one match between word and blob, it is called the *translation model*. Basically, they formulate an optimisation problem to learn the probability of a word c given a blob $\mathbf{x}_j$, e.g., $p(c|\mathbf{x}_j)$. The maximisation process is done by EM algorithm, making the computation very expensive. During annotation, regions of a testing image are represented by the closest centroids (blobs) and consequently, the annotation of each region is determined using the translation probabilities.

Different from translation model, Jeon et al. [65] assign words to entire images instead of specific blobs. They model the posterior probability $p(c|I)$ as the joint probability of words and blobs, which they call *cross media relevance model* (CMRM). This is in contrast to the blob to word translation model. Suppose, a testing image I is represented by the set of blobs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$. $p(c|I)$ is then approximated as $p(c, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ which is calculated from the training set T,

$$p(c,\mathbf{x}_1,\ldots,\mathbf{x}_m)=\sum_{J\in T}p(J)p(c,\mathbf{x}_1,\ldots,\mathbf{x}_m|J)=\sum_{J\in T}p(J)p(c|J)\prod_{i=1}^{m}p(\mathbf{x}_i|J) \quad (7)$$

where, J is a training image in the set T. The prior probabilities $p(J)$ are kept uniform for the entire training set. To calculate $p(c|J)$ and $p(\mathbf{x}_i|J)$, they build a word model and a blob model for each individual training image J. A global word model and a global blob model are built for the entire training set. $p(c|J)$ and $p(\mathbf{x}_i|J)$ are then calculated using the interpolation between the individual models and the overall models as follows:

$$p(c|J)=(1-\alpha_J)\frac{\#(c,J)}{|J|}+\alpha_J\frac{\#(c,T)}{|T|}$$

$$p(\mathbf{x}_i|J)=(1-\beta_J)\frac{\#(\mathbf{x}_i,J)}{|J|}+\beta_J\frac{\#(\mathbf{x}_i,T)}{|T|} \quad (8)$$

where α_J and β_J are the interpolation parameters. #(c, J) is the number of times concept c appears in J and #($\mathbf{x}_i$, J) denotes the number of times blob $\mathbf{x}_i$ appears in J. #(c, T) and #($\mathbf{x}_i$, T) are similarly defined for the entire training set T. $|J|$ is the aggregate count of all concept words and blobs in J and $|T|$ is the size of the training set. The model has better annotation accuracy than the translation model [21]. However, the performance is subject to the selection of appropriate interpolation parameters.

Yavlinsky et al. [92] also use a non-parametric approach to model the conditional probabilities. Different from the above mentioned approaches, they learn an estimation of the actual feature distribution using the traditional kernel smoothing method. In practice, they use two types of kernels to estimate the feature distribution. These are Gaussian kernel and EMD kernel. In the case of Gaussian kernel, they use either 24 or 108 dimensional colour and texture features for image representations. In the case of EMD kernel, they use region based image representations where regions are created by applying simple k-means clustering to image pixels. To calculate the conditional probability $p(\mathbf{x}|c)$ for an unknown image $\mathbf{x}$, the kernel functions are averaged over all training images in the training set.

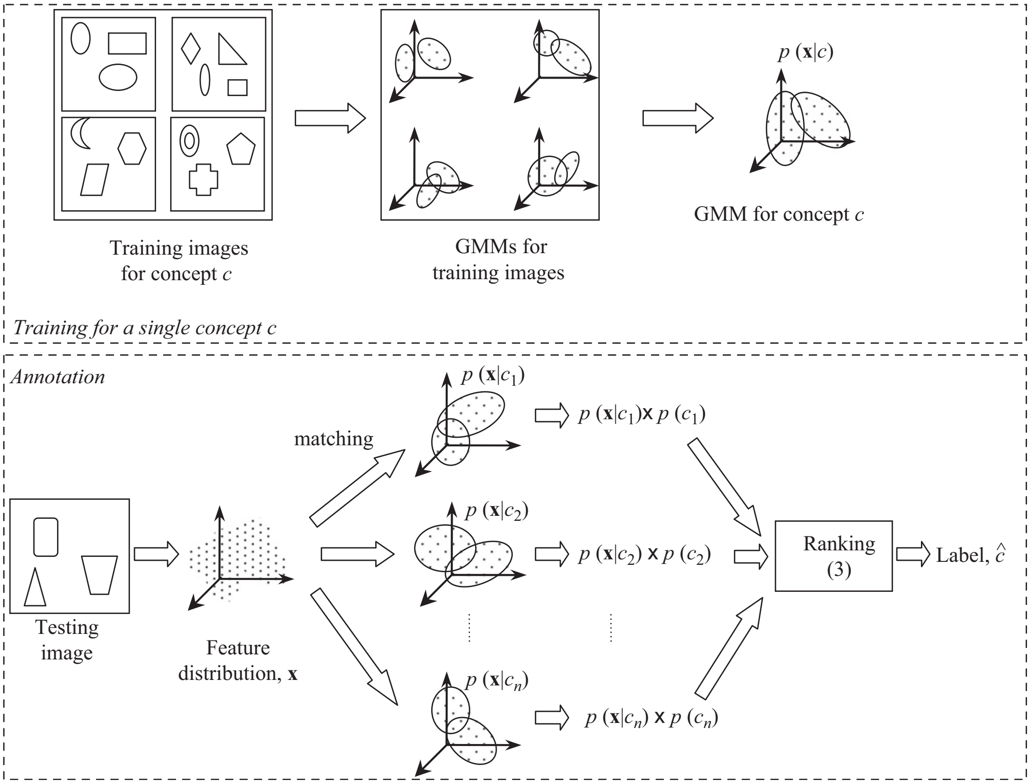


Fig. 9. Conditional probability modelling and image annotation using hierarchical GMMs [26].

3.2.2. *Parametric approach*

In this approach, the feature space is assumed to follow a certain type of known continuous distribution. Therefore, the conditional probability $p(\mathbf{x}|c)$ is modelled using this feature distribution. The general process is similar to that shown in Fig. 7. Features or regions are first clustered and quantised; the conditional probability model is then built for each cluster (or blob). The conditional probability $p(\mathbf{x}|c)$ is usually modelled as a multivariate Gaussian distribution as shown in Eq. (9):

$$p(\mathbf{x}|c) = \frac{1}{\sqrt{2^d \times \pi^d \times |\boldsymbol{\Sigma}|}} e^{-(\mathbf{x}-\bar{\mathbf{x}})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\bar{\mathbf{x}})} \tag{9}$$

where, d is the feature dimension, $\bar{\mathbf{x}}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix computed from the training feature vectors belonging to concept c.

Yang et al. [56] use the above equation to model the conditional probabilities in a region based approach. Training images are segmented into regions and the regions are grouped into sets based on their annotations. Let, X_c be the set of regions extracted from images belonging to concept c. It is assumed that region features in X_c follow the Gaussian distribution. During annotation, a testing image I is segmented into regions $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$. The conditional probability $p(I|c)$ is then calculated as following, which is similar to Eq. (5),

$$\begin{aligned} p(I|c) &= p(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\ldots,\mathbf{x}_n|c) = \prod_{k=1}^{n} p(\mathbf{x}_k|c) \\ &= \prod_{k=1}^{n} \left(\frac{1}{\sqrt{2^d \pi^d |\boldsymbol{\Sigma}_k|}} e^{-(\mathbf{x}_k-\bar{\mathbf{x}}_c)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_k-\bar{\mathbf{x}}_c)} \right) \end{aligned} \tag{10}$$

where multivariate mean $\bar{\mathbf{x}}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$ are learned from the regions of X_c. Once the conditional probabilities are determined, the same MAP criterion of (3) is used to determine the annotation of the testing image.

Both Li and Wang [33] and Carneiro et al. [26] learn the conditional probability models concept by concept and then use the models to annotate unknown images. Li and Wang [33] first break down training images in each concept into regions which are represented using LUV colours and wavelet texture features. They then cluster regions into clusters which they call prototypes. For each prototype, a Gaussian model is learned. Finally, a *Gaussian Mixture Model* (GMM) is built for each concept by averaging the Gaussian models of individual prototypes within the concept. To annotate an unknown image, its region features are extracted and the posterior probability of the image belonging to a concept is computed based on the concept GMM model. The drawback of this method is that parameter estimation for the Gaussian models is complex.

Different from Li et al.'s approach, Carneiro et al. [26] do not segment images into regions. Instead, they assume that image features follow certain Gaussian distributions and directly learn a GMM for each training image within a concept using *expectation maximisation* (EM) algorithm. This is equivalent to a simultaneous segmentation and model learning process. They then build the concept GMM by averaging the individual GMMs within the concept. In the annotation stage, a GMM is learned for the unknown image and the GMM is then matched with each concept model. The concepts with the best match are selected as the annotations for the unknown image. Fig. 9 shows this process. Similar to Li et al.'s method, the drawback is that the estimation of the GMM models is complex due to the use of EM optimisation method.

There are several issues with the parametric approaches. First, these approaches usually assume a particular feature distribution which may not be true. Second, the model parameters are usually learned by optimisation methods, in most cases it may not converge; and the computation of the optimisation process is very expensive. Third, for those concept by concept annotation models [26,33], the model for each concept is learned independently and the correlations between concepts are not captured.

3.2.3. Summary

In multi-labelling annotation, once the models are learned, the annotation process is pretty much similar to the online matching and ranking in the traditional CBIR. Therefore, every time a user types a keyword, the entire annotation process is repeated online in order to test all images against the keyword [26] and it may take unacceptably long time. Alternatively, images in database are annotated offline and are then indexed with an inverted file. Images are then retrieved in the same way as text documents [112]. However, in either case, expensive online matching is used. This is a key drawback compared with the single labelling annotation.

3.3. Image annotation incorporating metadata

The WWW is a rich source of both imagery and text information. Web images often come with text descriptions, URL, HTML code, etc. The web information can be used for image annotation and retrieval. A number of techniques have been developed for annotating web images, most of them integrating both metadata and visual features for accurate image annotation [49–54,66]. Therefore, these methods can be called *hybrid methods*.

Cai et al. [51] proposed a two level annotation and clustering mechanism: textual clustering for semantic annotation and visual clustering for re-organisation of images within each semantic category. Images from web pages are first represented using three types of features: textual features (derived from surrounding text), link graph (derived from three complex hyperlink matrices) and visual features (derived from colour moments on local Fourier transform). The textual features and link graph are used to cluster images into semantic category which is equivalent to annotation. However, images within each of the semantic categories may not be perceptually similar. Therefore, they apply a second level of clustering on each of the semantic categories to re-organise the images into clusters based on visual features. The major issue with this method is that the textual features especially the link graph features are not reliable, as shown in existing image search engines.

Wang et al. [31] also propose an automatic system that annotates images using both web description and content features. The system needs at least one correct initial keyword and one example image to initiate the process. The keyword is used to search the web to find images and their web descriptions. Thereafter, 36 dimensional colour correlogram is used to select a number of top ranked images similar to the example image. The web descriptions of the selected images are clustered using a special text clustering algorithm. Each cluster is scored either by its size or by the average number of words. The words in the top scored clusters are used for annotations. The advantage in this approach is that it does not need any training samples. However, the performance is subject to the quality of the description of the images, which is not reliable in the WWW.

As annotations from text description can be noisy, these annotations need refinement. This is especially needed for web based image annotation, because each image is usually annotated with multiple words which may not be related to each other. In a refinement stage, it preserves the annotations which are strongly correlated and rejects those which are not so strongly related to each other. Jin et al. [64] use WordNet [117] for annotation refinement. WordNet is an online lexicon where more than 150 K words are hierarchically organised. The words in WordNet maintain 'is a kind of' or 'is a part of' relationships which are used to find similarity between words. After getting the annotations of an image using any existing method, Jin et al. [64] use WordNet to calculate the total similarity of each word to other words of the annotation set. If the similarity is below certain threshold, it is discarded. The principle is that the words, which are very similar to each other, must belong to the same higher level semantic. They should be preserved and others should be removed. Suppose an image is annotated with four words: Tiger, Tree, Bush and South Asia. Heuristically, the first three are regarded as strongly correlated to each other and they are part of a higher level semantic 'tiger in forest'. However, 'South Asia' has a weak relationship to the other three words; therefore, the phrase is discarded from the annotation.

The disadvantage of Jin et al. [64] is that it depends on WordNet and thus, this approach cannot refine the annotations which do not appear in WordNet. Wang et al. [29,32] improve the refinement process by removing the dependency on WordNet. Instead of using WordNet, Wang et al. [29] calculate the similarity between two words as the normalised frequency of images annotated by both words. In [32], they calculate this similarity as the normalised sum of content-similarities between the candidate image and the images annotated by both words. The similarity values are used to find strongly correlated annotation words.

Another important issue is to define appropriate level of semantics. This is especially important in web image applications because the noise in web texts is common and it is difficult to understand which annotations should be used. For example, for an image with kangaroos, the concept 'Australia' is more abstract than the concepts 'Kangaroo'. Therefore, it is necessary to identify the concepts which have smaller semantic gap (like the concept, 'Kangaroo').

Lu et al. [34] propose a system that models the *semantic gap* and rank a set of candidate annotation words according to their increased semantic gap. For example, in an image with annotations of 'green', 'brown', 'grass', 'bush', 'kangaroo', and 'Australia', the terms 'green' and 'brown' would have smaller semantic gap from low level features. 'Kangaroo', 'grass', and 'bush' would be the next level semantics and have higher semantic gap than the annotations 'green' and 'brown' from the low level features. The annotation 'Australia' has the highest semantic gap from the low level features. Lu et al. use web images to develop a sorted list of lexicon (annotating words or concepts) based semantic gap, and

Table 5
Contrast of different annotation methods.

Annotation method	Pros	Cons
SVM	Small sample, optimal class boundary, non-linear classification	Single labelling, one class per time, expensive trial and run, sensitive to noisy data, prone to over-fitting
ANN	Multiclass outputs, non-linear classification, robust to noisy data, suitable for complex problem	Single labelling, sub-optimal, expensive training, complex and black box classification
DT	Intuitive, semantic rules, multiclass outputs, fast, allow missing values, handle both categorical and numerical values	Single labelling, sub-optimal, need pruning, can be unstable
Non-parametric	Multi-labelling, model free, fast	Large number of parameters, large sample, sensitive to noisy data
Parametric	Multi-labelling, small sample, good approximation of unknown distribution	Predefined distribution, expensive training, approximated boundary
Metadata	Use of both textual and visual features	Difficult to relate visual features with textual features, difficult textual feature extraction

they first cluster a short listed web images based on their textual and visual similarity. The available words of each cluster are ranked based on text ranking technique. The rankings of words from all the clusters are fused together to generate a final ranking. Existing annotation approaches can use this ranking to decide which concepts should be learned because the top ranked words in the list are easier to learn than the bottom ones. The problem is that the ranking needs to be learned every time the database changes. Table 5 summarises the different type of annotation methods.

4. Discussions and conclusions

We have made a comprehensive review on the state of the art AIA techniques in literature. We have focused on two major aspects of AIA: feature extraction and semantic learning. In terms of features, several types of features have been used in AIA. These features can be extracted either locally or globally. Global methods compute a single set of features from the entire image like a colour histogram or moments. As natural images are not homogenous, this single set of features may not be meaningful unless they are applied in domain specific applications. Local methods divide images into regions or blocks, a set of features is computed for each of the regions. As a result, an image is represented as a bag of features. Bag of features can represent images at object level and provides spatial information. However, region features may not be accurate due to the usually unsupervised segmentation. For AIA, supervised segmentation may be considered. It is interesting to find a few works in the literature which integrate segmentation and learning in a single process. It has also been found that some colour descriptors like correlogram, CSD and CCV capture both colour and texture features. However, they are usually applied to entire image. Texture descriptors like edge histogram, co-occurrence matrix, directionality and spectral methods capture both texture and shape features.

In terms of semantic learning, there are several approaches including the traditional binary classification methods, multiple labelling methods and metadata methods. Traditional binary classification may sound attractive and straightforward but it overlooks the fact that an image can usually be included into more than one category. Furthermore, it lacks a mechanism of ranking images according to their similarity to the classified categories. Multiple labelling is a more reasonable approach, because it assigns an image to several categories and assigns an image to a category with a confidence value which assists image ranking. All the above approaches attempt to learn higher level semantics from low level features or visual features alone. It appears natural that metadata should be used to overcome the limitation of visual features if it is available. However, analysing metadata is a complex matter and becomes another issue in image annotation.

Overall, AIA is a very challenging research area. There are several major issues in AIA research. The first issue is high dimensional feature analysis. Currently, all existing features have limitations of describing images and none of existing features is powerful enough to represent the large variety of images in nature. Common practice is to combine several types of features to represent as many images as possible. However, the processing and analysing of high dimensional image features is a very complex issue. Due to the 'curse of dimensionality', the performance of classifiers degrades dramatically when feature dimension is too high. Therefore, features need to be further mined to select the right number of features and right features for annotation. The recent advance in subspace research offers promising solution in this regard.

The second issue is how to build an effective annotation model. Most existing AIA models are learned from low level image features. However, due to the 'combinatorial explosion' of required image to build an annotation model, the number of sample images is not large enough to train an accurate model. Therefore, textual information or metadata should be employed to improve annotation accuracy. However, very often, metadata is either not accurate or not adequate. How to integrate both low level visual information and high level textual information into a coherent annotation model is a challenging issue. The number of hybrid annotation methods discussed in Section 3.3 may provide some clues to addressing this issue.

The third issue is that currently annotation and ranking are done online simultaneously in the multiple labelling annotation approaches. This is not efficient for image retrieval. The alternative is to do the annotation offline as in the single labelling approach and separate the ranking from annotation, that is, images are first annotated with a concept/category and ranking is also done offline after annotation. Once the images are annotated and ranked offline, retrieval is instant.

The fourth issue is how to rank images within each of the categories resulted from the single labelling techniques, so as to improve retrieval accuracy. Since images within each category show certain distribution pattern, a Gaussian mixture model followed by an MAP ranking offers a practical solution.

The fifth issue is the lack of standard vocabulary and taxonomy for annotation. At this moment, arbitrary vocabularies are used in AIA literature. It is not known how images should be categorised. A hierarchical modelling of image semantics is needed to categorise images properly. A hierarchical taxonomy not only standardizes the annotation vocabulary but also allows step by step annotation which is more practical.

Finally, there is no commonly acceptable image database for AIA training and evaluation. All AIA methods require a large number of pre-labelled image samples for training the model. At this moment, different AIA methods use different image datasets for training and evaluation, making it difficult to compare the performance. The database issue is closely related to the taxonomy issue. If a standard taxonomy of image semantics is available, a standard database can also be created accordingly.

All these issues point to future research directions in AIA area.

Acknowledgement

The authors are grateful for the constructive and valuable comments made by the many expert reviewers.

References

[1] Y. Liu, D. Zhang, G. Lu, A survey of content-based image retrieval with high-level semantics, Pattern Recognition 40 (1) (2007) 262–282.

[2] M.S. Lew, N. Sebe, C. Djeraba, R. Jain, Content-based multimedia information retrieval: state of the art and challenges, ACM Transactions on Multimedia Computing, Communications and Applications 2 (1) (2006) 1–19.

[3] N. Vasconcelos, From pixels to semantic spaces: advances in content-based image retrieval, Computer 40 (7) (2007) 20–26.

[4] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences and trends of the new age, ACM Computing Surveys 40 (2) (April 2008).

[5] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE PAMI 22 (12) (2000) 1349–1380.

[6] F. Long, H.J. Zhang, D.D. Feng, Fundamentals of content-based image retrieval, in: D.D. Feng, W.C. Siuandg, H.J. Zhan (Eds.), Multimedia Information Retrieval and Management, Springer, 2003.

[7] Y. Rui, T.S. Huang, S.F. Chang, Image retrieval: current techniques, promising directions and open issues 10 (1999) 39–62Journal of Visual Communication and Image Representation 10 (1999) 39–62.

[8] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picture libraries, IEEE PAMI 23 (9) (2001) 947–963.

[9] H. Tamura, N. Yokoya, Image database systems: a survey, Pattern Recognition 17 (1) (1984) 29–43.

[10] S.K. Chang, A. Hsu, Image information systems: where do we go from here? IEEE Transactions on Knowledge and Data Engineering 5 (5) (1992) 431–442.

[11] A.K. Jain, A. Vailaya, Image retrieval using colour and shape, Pattern Recognition 29 (8) (1996) 1233–1244.

[12] N. Vasconcelos, Minimum probability of error image retrieval, IEEE Transactions on Signal Processing 52 (8) (2004) 2322–2336.

[13] N. Rasiwasia, N. Vasconcelos, P.J. Moreno, Query by semantic example, in: Proceedings of the Fifth International Conference on Image and Video Retrieval, Lecture Notes in Computer Science, vol. 4071, Springer, Berlin, 2006, pp. 51–60.

[14] R. Marée, M. Dumont, P. Geurts, L. Wehenkel, Random subwindows and randomized trees for image retrieval, classification and annotation, in: Proceedings of the 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) and Sixth European Conference on Computational Biology, 2007.

[15] Yahoo Home Page: 〈http://www.Yahoo.Com〉.

[16] Google Home Page: 〈http://www.Google.Com〉.

[17] M. Szummer, R.W. Picard, Indoor–outdoor image classification, in: Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Databases, January 1998.

[18] Y. Mori, H. Takahashi, R. Oka, Image-to-word transformation based on dividing and vector quantizing images with words, in: Proceedings of the Seventh ACM International Conference on Multimedia, ACM Press, 1999.

[19] C.P. Town, D. Sinclair, Content-Based image retrieval using semantic visual categories, Technical Report, No. MV01-211, Society for Manufacturing Engineers, 2001.

[20] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, H.J. Zhang, Image classification for content-based indexing, IEEE Transactions on Image Processing 10 (1) (2001) 117–130.

[21] P. Duygulu, K. Barnard, J.F.G.D. Freitas, D.A. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary, in: Proceedings of the Seventh European Conference on Computer Vision, Springer, 2002, pp. 349–354.

[22] E. Chang, K. Goh, G. Sychay, G. Wu, Cbsa: content-based soft annotation for multimodal image retrieval using Bayes point machines, IEEE Transactions on Circuits Systems and Video Technology 13 (1) (2003) 26–38.

[23] C. Cusano, G. Ciocca, R. Schettini, Image annotation using SVM, in: Proceedings of the Internet Imaging IV, vol. 5304, SPIE, 2004.

[24] J. Fan, Y. Gao, H. Luo, G. Xu, Automatic image annotation by using concept-sensitive salient objects for image content representation, in: Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom, 2004, pp. 61–368.

[25] S.L. Feng, R. Manmatha, V. Lavrenko, Multiple Bernoulli relevance models for image and video annotation, in: Proceedings of the CVPR04, 2004, pp. 1002–1009.

[26] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE PAMI 29 (3) (2007) 394–410.

[27] R. Marée, P. Geurts, J. Piater, L. Wehenkel, Random subwindows for robust image classification, in: Proceedings of the CVPR05, vol. 1, 2005, pp. 34–40.

[28] J. Li, J.Z. Wang, Real-time computerized annotation of pictures, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, Santa Barbara, CA, USA, 2006, pp. 911–920.

[29] C. Wang, F. Jing, L. Zhang, H.-J. Zhang, Image annotation refinement using random walk with restarts, in: Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA, 2006, pp. 647–650.

[30] C. Wang, F. Jing, L. Zhang, H.-J. Zhang, Scalable search-based image annotation of personal images, in: Proceedings of the Eighth ACM International Workshop on Multimedia Information Retrieval, Santa Barbara, California, USA, 2006, pp. 269–278.

[31] X.-J. Wang, F.J.L. Zhang, W.-Y. Ma, Annosearch: image auto-annotation by search, in: Proceedings of the CVPR06, vol. 2, 2006, pp. 1483–1490.

[32] C. Wang, F. Jing, L. Zhang, H.-J. Zhang, Content-based image annotation refinement, in: Proceedings of the CVPR07n, 2007, pp. 1–8.

[33] J. Li, J.Z. Wang, Real-time computerized annotation of pictures, IEEE PAMI 30 (6) (2008) 985–1002.

[34] Y. Lu, L. Zhang, Q. Tian, W.-Y. Ma, What are the high-level concepts with small semantic gaps, in: Proceedings of the CVPR08, June 2008, pp. 1–8.

[35] C. Wang, L. Zhang, H.-J. Zhang, Learning to reduce the semantic gap in web image retrieval and annotation, in: Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, 2008, pp. 355–362.

[36] C. Wang, L. Zhang, H.-J. Zhang, Scalable Markov model-based image annotation, in :Proceedings of the International Conference on Content-Based Image and Video Retrieval, Niagara Falls, Canada, 2008, pp. 113–118.

[37] Y. Liu, D. Zhang, G. Lu, Region-based image retrieval with high-level semantics using decision tree learning, Pattern Recognition 41 (8) (2008) 2554–2570.

[38] J. Li, J.Z. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach, IEEE PAMI 25 (9) (2003) 1075–1088.

[39] W. Shao, Automatic Annotation of Digital Photos, Master's Thesis, University of Wollongong, August 2007.

[40] J. Zobel, A. Moffat, Inverted files for text search engines, ACM Computing Surveys 38 (2) (July 2006) 1–56.

[41] J. Huang, S. Kuamr, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using colour correlogram, in: Proceedings of the CVPR97, 1997, pp. 762–765.

[42] S.K. Chang, C.W. Yan, D.C. Dimitroff, T. Arndt, An intelligent image database system, IEEE Transactions on Software Engineering 14 (5) (May, 1988) 681–688.

[43] J. Dowe, Content-based retrieval in multimedia imaging, in: Proceedings of the SPIE Storage and Retrieval for Image and Video Database, 1993.

[44] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003.

[45] N.-C. Yang, W.-H. Chang, C.M. Kuo, T.-H. Li, A fast MPEG-7 dominant colour extraction with new similarity measure for image retrieval, Journal of Visual Communication and Image Retrieval 19 (2008) 92–105.

[46] M.M. Islam, D. Zhang, G. Lu, A geometric method to compute directionality features for texture images, in: Proceedings of the International Conference on Multimedia and Expo, Hannover, Germany, June 23–26, 2008, pp. 1521–1524.

[47] M.M. Islam, D. Zhang, G. Lu, Automatic categorization of image regions using dominant colour based vector quantization, in: Proceedings of the Digital Image Computing: Techniques and Applications, Canberra, Australia, December 1–3, 2008, pp. 191–198.

[48] J. Starck, E.J. Candès, D.L. Donoho, The curvelet transform for image denoising, IEEE Transactions on Image Processing 11 (6) (2002) 670–684.

[49] R. Yan, A. Natsev, M. Campbell, A learning-based hybrid tagging and browsing approach for efficient manual image annotation, in: Proceedings of the CVPR08, Alaska, USA, June 23–28, 2008, pp. 1–8.

[50] J. Jeon, R. Manmatha, Automatic image annotation of news images with large vocabularies and low quality training data, in: Proceedings of the ACM Multimedia, 2004.

[51] D. Cai, X. He, Z. Li, W.-Y. Ma, J.-R. Wen, Hierarchical clustering of www image search results using visual, textual and link information, in: Proceedings of the ACM International Conference on Multimedia, 2004, pp. 952–959.

[52] X.-J. Wang, W.-Y. Ma, Q.-C. He, X. Li, Grouping web image search result, in: Proceedings of the 12th ACM Conference on Multimedia, New York, NY, USA, 2004, pp. 436–439.

[53] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, W.-Y. Ma, Igroup: web image search results clustering, in: Proceedings of the 14th ACM Conference on Multimedia, Santa Barbara, CA, USA, 2006, pp. 377–384.

[54] R. Marée, P. Geurts, J. Piater, L. Wehenkel, A generic approach for image classification based on decision tree ensembles and local sub-windows, in: Proceedings of the Sixth Asian Conference on Computer Vision, vol. 2, 2004, pp. 860–865.

[55] O. Chapelle, P. Haffner, V.N. Vapnik, Support vector machines for histogram-based image classification, IEEE Transactions on Neural Networks 10 (1999) 1055–1064.

[56] C. Yang, M. Dong, F. Fotouhi, Image content annotation using Bayesian framework and complement components analysis, in: Proceedings of the IEEE International Conference on Image Processing, Geneva, Italy, 2005.

[57] K.-S. Goh, E.Y. Chang, B. Li, Using one-class and two-class SVMs for multiclass image annotation, IEEE Transactions on Knowledge and Data Engineering 17 (10) (2005) 1333–1346.

[58] S.B. Park, J.W. Lee, S.K. Kim, Content-based image classification using a neural network, Pattern Recognition Letters 25 (2004) 287–300.

[59] X. Qi, Y. Han, Incorporating multiple SVMs for automatic image annotation, Pattern Recognition 40 (2) (2007) 728–741.

[60] S. Rui, W. Jin, T.-S. Chua, A novel approach to auto image annotation based on pairwise constrained clustering and semi-naïve Bayesian model, in: Proceedings of the 11th International Conference on Multimedia Modelling, 2005, pp. 322–327.

[61] S. Kim, S. Park, M. Kim, Image classification into object/non-object classes, in: Proceedings of the International Conference on Image and Video Retrieval, Dublin, Ireland, 2004, pp. 393–400.

[62] K. Kuroda, M. Hagiwara, An image retrieval system by impression words and specific object names—Iris, Neurocomputing 43 (2002) 259–276.

[63] V. Mezaris, I. Kompatsiaris, M.G. Strintzis, An ontology approach to object-based image retrieval, in: Proceedings of the International Conference on Image Processing, 2003, pp. 511–514.

[64] Y. Jin, L. Khan, L. Wang, M. Awad, Image annotations by combining multiple evidence and Wordnet, in: Proceedings of the ACM Multimedia, Singapore, 2005.

[65] J. Jeon, V. Lavrenko, R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, in: Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 119–126.

[66] Y. Liu, D. Zhang, G. Lu, Sieve-search images effectively through visual elimination, Lecture Notes in Computer Science 4577 (2007) 381–390.

[67] J.-H. Lim, Q. Tian, P. Mulhem, Home photo content modelling for personalized event based retrieval, IEEE Multimedia 10 (4) (2003) 28–37.

[68] S.C. Zhu, A. Yuille, Region competition: unifying snakes, region growing and Bayes/mdl for multiband image segmentation, IEEE PAMI (1996) 884–900.

[69] A. Chakraborty, J. Duncan, Game-theoretic integration for image segmentation, IEEE PAMI 21 (1) (1999) 12–30.

[70] T.F. Chan, L.A. Vese, Active contours without edges, IEEE Transactions on Image Processing 10 (2) (2001) 266–277.

[71] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using expectation–maximization and its application to image querying, IEEE PAMI 24 (8) (2002) 1026–1038.

[72] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE PAMI 24 (5) (2002) 603–619.

[73] Z. Tu, S.-C. Zhu, Image segmentation by data-driven Markov chain Monte Carlo, IEEE PAMI 24 (5) (2002) 657–673.

[74] H. Feng, T.S. Chua, A bootstrapping approach to annotating large image collection, in: Proceedings of the Workshop on Multimedia Information Retrieval in ACM Multimedia, pp. 55–62, 2003.

[75] R. Shi, H. Feng, T.S. Chua, C.H. Lee, An adaptive image content representation and segmentation approach to automatic image annotation, in: Proceedings of the International Conference on Image and Video Retrieval, 2004, pp. 545–554.

[76] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE PAMI 22 (8) (2000) 888–905.

[77] W. Tao, H. Jin, Y. Zhang, Colour image segmentation based on mean shift and normalized cuts, IEEE Transactions on Systems, Man and Cybernetics 37 (5) (2007) 1382–1389.

[78] J. Malik, S. Belongie, T.K. Leung, J. Shi, Contour and texture analysis for image segmentation, International Journal of Computer Vision 43 (1) (2001) 7–27.

[79] Y. Deng, B.S. Manjunath, Unsupervised segmentation of colour-texture regions in images and video, IEEE PAMI 23 (8) (2001) 800–810.

[80] F. Jing, M. Li, H.-J. Zhang, B. Zhang, An efficient and effective region-based image retrieval framework, IEEE Transactions on Image Processing 13 (5) (May, 2004) 699–709.

[81] J.D. Foley, A.V. Dam, S.K. Feiner, J.F. Hughes, Computer Graphics: Principles and Practice, second ed., Addison-Wesley, 1997.

[82] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7: Multimedia Content Description Language, John Wiley & Sons Ltd., 2002.

[83] R.C. Gonzalez, R.E. Woods, Digital Image Processing, third ed., Prentice-Hall, 2007.

[84] P.L. Stanchev, D. Green Jr., B. Dimitrov, High level colour similarity retrieval, International Journal of Information Theories and Applications 10 (3) (2003) 363–369.

[85] M.J. Swain, D.H. Ballard, Colour indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[86] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, IEEE Computer 28 (9) (September, 1995) 23–32.

[87] G. Pass, R. Zabith, Histogram refinement for content-based image retrieval, in: Proceedings of the IEEE Workshop on Applications of Computer Vision, 1996, pp. 96–102.

[88] K.-W. Park, J.-W. Jeong, D.-H. Lee, Olybia: ontology-based automatic image annotation system using semantic inference rules, in: Proceedings of the 12th International Conference on Database Systems for Advanced Applications, Bangkok, Thailand, Lecture Notes in Computer Science, Springer, Berlin, 2007, pp. 485–496.

[89] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Transactions on Systems, Man and Cybernetics 8 (6) (1978) 460–473.

[90] K.S. Fu, Syntactic Pattern Recognition and Applications, Prentice-Hall, New Jersey, 1982.

[91] A. Materka, M. Strzelecki, Texture Analysis Methods—A Review, Institute of Electronics, Technical University of Lodz, Brussels, , 1998 COST B 11.

[92] A. Yavlinsky, E. Schofield, S. Rüger, Automated image annotation using global features and robust nonparametric density estimation, in: Proceedings of the International Conference on Image and Video Retrieval, Singapore, 2005, pp. 507–517.

[93] G.R. Cross, A.K. Jain, Markov random field texture models, IEEE PAMI 5 (1) (1983) 25–39.

[94] M. Tuceryan, A.K. Jain, Texture analysis, in: C.H. Chen, L.F. Pau, P.S.P. Wang (Eds.), The Handbook of Pattern Recognition and Computer Vision, second ed.,World Scientific Publishing Co., 1998, pp. 207–248.

[95] B.B. Chaudhuri, N. Sarkar, Texture segmentation using fractal dimension, IEEE PAMI 17 (1) (1995) 72–77.

[96] F. Liu, R.W. Picard, Periodicity, directionality and randomness—world features for image modeling and retrieval, IEEE PAMI 18 (7) (1996) 722–733.

[97] J. Luo, A. Savakis, Indoor vs. outdoor classification of consumer photographs using low-level and semantic features, in: Proceedings of the International Conference on Image Processing, 2001, pp. 745–748.

[98] K.-L. Lee, L.-H. Chen, An efficient computation method for the texture browsing descriptor of MPEG-7, Image and Vision Computing 23 (2005) 479–489.

[99] Z. Lu, S. Li, H. Burkhardt, A content-based image retrieval scheme in jpeg compressed domain, International Journal of Innovative Computing, Information and Control 2 (4) (2006) 831–839.

[100] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of large image data, IEEE PAMI 18 (8) (1996) 837–842.

[101] R. Zhang, Z.M. Zhang, M. Li, W.-Y. Ma, H.-J. Zhang, A probabilistic semantic model for image annotation and multi-modal image retrieval, in: Proceedings of the 10th International Conference on Computer Vision, vol. 1, 2005, pp. 846–851.

[102] N. Hervé, N. Boujemaa, Image annotation: which approach for realistic databases? in: Proceedings of the Sixth ACM International Conference on Image and Video Retrieval, Amsterdam, Netherlands, 2007, pp. 170–177.

[103] I.J. Sumana, M.M. Islam, D. Zhang, G. Lu, Content based image retrieval using curvelet transform, in: Proceedings of the International Workshop on Multimedia Signal Processing, Cairns, Australia, October 8–10, 2008, pp. 11–16.

[104] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (1) (2004) 1–19.

[105] Z. Hong, Q. Jiang, Hybrid content-based trademark retrieval using region and contour features, in: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications—Workshops, March 25–28, 2008, pp. 1163–1168.

[106] W.H. Leung, T. Chen, Trademark retrieval using contour-skeleton stroke classification, in: Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 2, August 26–29, 2002, pp. 517–520.

[107] Y. Liu, J. Zhang, D. Tjondronegoro, S. Geve, A shape ontology framework for bird classification, in: Proceedings of the Ninth Conference on Digital Image Computing Techniques and Applications, 2007, pp. 478–484.

[108] S.K. Chang, E. Jungert, A spatial knowledge structure for image information systems using symbolic projections, in: Proceedings of the Fall Joint Computer Conference, 1986, pp. 79–86.

[109] A.J.T. Lee, H.-P. Chiu, 2D z-string: a new spatial knowledge representation for image databases, Pattern Recognition Letters 24 (16) (December 2003) 3015–3026.

[110] Y.I. Chang, B.Y. Yang, W.H. Yeh, A bit-pattern-based matrix strategy for efficient iconic indexing of symbolic pictures, Pattern Recognition Letters 24 (1–3) (January 2003) 537–545.

[111] X.-J. He, Y. Zhang, T.-M. Lok, M.R. Lyu, A new feature of uniformity of image texture directions coinciding with the human eyes perception, in: Lecture Notes in Artificial Intelligence, vol. 3614, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 727–730.

[112] D. Zhang, M.M. Islam, G. Lu, J. Hou, Semantic image retrieval using region based inverted file, in: Proceedings of the Digital Image Computing: Techniques and Applications (DICTA09), Melbourne, Australia, 1–3 December 2009.

[113] D. Zhang, A. Wong, M. Indrawan, G. Lu, Content-based image retrieval using Gabor texture features, in: Proceedings of the First IEEE Pacific-Rim Conference on Multimedia, Sydney, Australia, December 2000, pp. 392–395.

[114] M.M. Islam, D. Zhang, G. Lu, Region based colour image retrieval using curvelet transform, in: Proceedings of the Asian Conference on Computer Vision (ACCV09), Xi'an, China, 2009, pp. 242–249.

[115] O. Cavus, S. Aksoy, Semantic scene classification for image annotation and retrieval, in: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic and Statistical Pattern Recognition, Orlando, Florida, 2008, pp. 402–410.

[116] R. Zhao, W.I. Grosky, From feature to semantics: some preliminary results, in: Proceedings of the International Conference on Multimedia and Expo, 2000, pp. 679–682.

[117] G.A. Miller, Wordnet: a lexical database for English, Communications of the ACM 38 (11) (1995) 39–41.

[118] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[119] F.D. Frate, F. Pacifici, G. Schiavon, C. Solimini, Use of neural networks for automatic classification from high-resolution images, IEEE Transactions on Geoscience and Remote Sensing 45 (4) (April 2007) 800–809.

[120] J.R. Quinlan, Induction of decision trees, Springer Machine Leaning (1986) 81–106.

[121] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, California, USA, 1993.

[122] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Chapman & Hall (Wadsworth, Inc.), Monterey, California, USA, 1984.

[123] I.K. Sethi, I.L. Coman, Mining association rules between low-level image features and high-level concepts, SPIE Data Mining and Knowledge Discovery 3 (2001) 279–290.

[124] R.C.F. Wong, C.H.C. Leung, Automatic semantic annotation of real-world web images, IEEE PAMI 30 (11) (2008) 1933–1944.

[125] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Transactions on Circuits and Video Technology 8 (5) (1998) 644–655.

[126] D. Tao, X. Tang, X. Li, Xindong Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image

retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 28 (7) (July 2006) 1,088–1,099.

[127] S. Antani, R. Kasturi, R. Jain, A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video, Pattern Recognition 35 (4) (2002) 945–965.

[128] V. Dey, Y. Zhang, M. Zhong, A review on image segmentation techniques with remote sensing perspective, in Proceedings of the International Society for Photogrammetry and Remote Sensing Symposium (ISPRS10), vol. XXXVIII (Part 7A), Austria, July 5–7, 2010.

[129] Z. Zhou, M. Zhang, Multi-instance multi-label learning with application to scene classification, in: Proceedings of the Advances in Neural Information Processing Systems, vol. 19 (NIPS'06), 2006.

[130] Z. Zha, et al., Joint multi-label multi-instance learning for image classification, in: Proceedings of the CVPR 2008.

Dengsheng Zhang obtained his Ph.D. in Computing in 2002 from Monash University, Australia. He has also completed the Graduate Certificate of Higher Education and the Ph.D. supervision accreditation certificate in Monash University. He joined Monash University after he completed his Ph.D. in 2002. He is currently a senior lecturer in Gippsland School of Information Technology at Monash University, Australia. Dr. Zhang has over 15 years research experience in multimedia information processing and retrieval areas, and has published over 70 refereed international journal and conference papers in his career. The number of citations on his publications is near 2000 based on Google Scholar. He has supervised a number of PhDs and masters to completion. His main research interests include pattern recognition, image, audio and video information processing, and retrieval. He has won the 2009 Faculty of Information Technology Early Career Researcher Award for his outstanding research achievement.

Md. Monirul Islam received his B.Sc. and M.Sc. in Computer Science and Engineering (CSE) from Bangladesh University of Engineering and Technology (BUET) in 2001 and 2004, respectively. He obtained his Ph.D. from Monash University in 2009. He is currently an associate professor at BUET.

Guojun Lu is currently a Professor and Associate Dean of Faculty of Information Technology, Monash University. He has held positions at Loughborough University, National University of Singapore, and Deakin University, after he obtained his Ph.D. in 1990 from Loughborough University and B.Eng. in 1984 from Nanjing Institute of Technology (now South East University). Guojun's main research interests are in multimedia communications and multimedia information indexing and retrieval. He has published over 130 refereed journal and conference papers in these areas and wrote two books Communication and Computing for Distributed Multimedia Systems (Artech House 1996), and Multimedia Database Management Systems (Artech House 1999).