NPIC: Hierarchical synthetic image classification using image search and generic features

Fei Wang and Min-Yen Kan*

Department of Computer Science, School of Computing,
National University of Singapore, Singapore, 117543
{wangfei2,kanmy}@comp.nus.edu.sg

Abstract. We introduce NPIC, an image classification system that focuses on synthetic (e.g., non-photographic) images. We use class-specific keywords in an image search engine to create a noisily labeled training corpus of images for each class. NPIC then extracts both content-based image retrieval (CBIR) features and metadata-based textual features for each image for machine learning. We evaluate this approach on three different granularities: 1) natural vs. synthetic, 2) map vs. figure vs. icon vs. cartoon vs. artwork 3) and further subclasses of the map and figure classes. The NPIC framework achieves solid performance (99%, 97% and 85% in cross validation, respectively). We find that visual features provide a significant boost in performance, and that textual and visual features vary in usefulness at the different levels of granularities of classification.

1 Introduction

Images created entirely by digital means are growing in importance. Such synthetic images are an important means for recording and presenting visual information. The accurate classification of these images – such as icons, maps, figures and charts – is increasingly important. With the advent of the web, images are being used not just to communicate content but also for decoration, formatting and alignment. An image classification system can improve image search and retrieval engines and can act an input filter for downstream web processing as well as image understanding systems.

We introduce NPIC, an image classification system that is specifically trained on synthetic images. The implemented system uses semi-supervised machine learning to create its classifier. It does this by first using class-specific keywords to build a corpus of associated images via an image search engine. Textual features are extracted from the filename, comments and URLs of the images and content-based image retrieval features are also extracted. These features are strung together as a single feature vector and fed to a machine learner to learn a model. The resulting system is able to enhance the performance of text-only based image search, as the addition of visual features allows some spurious image matches to be correctly rejected.

A classifier needs ground truth labels to classify against. Existing image classification taxonomies are a good starting point. However, our dataset comes from the web, and in our opinion, a suitable taxonomy of content images available on the web does

* Contact author

not exist. After sampling synthetic images culled from the web, we decided to create our own hierarchy for the classification of web images, loosely based on portions of the Getty Art and Architecture Thesaurus (AAT).

NPIC obtains very good classification accuracy on all three granularities that we have trained the system on. A key point in the analysis of our study shows that although textual features are an immense help to synthetic image classification, their efficacy can be eclipsed by CBIR features at finer granularities.

After reviewing past related work on image classification, we discuss our methodology, including the design for the image hierarchy and how we construct our training data set using the commodity image search engine, Google Image Search. We then inventory both the textual and visual features in Section 4. Finally, we describe our experiments using cross-validation on the training set as well as using another synthetic dataset drawn from the Wikipedia.

2 Related Work

Image classification is a relatively young field of research, with many published systems being created after the year 2000. As of today, although many image categorization systems have been created, most classify against a very general classification scheme. A representative example is [1], who implemented and evaluated a system that performs a two-stage classification of images: first, distinguishing photo-like images from non-photographic ones, followed by a second round in which actual photos are separated from artificial, photo-like images, and non-photographic images are differentiated into presentation slides, scientific posters and comics. The WebSeer system [2] investigates how to classify images into three categories: photographs, portraits and computer-generated drawings. Both schemes are neither exclusive nor exhaustive; many images fall into multiple categories or none. Work has also focused on specific synthetic image classes. [3] and [4] deal only with chart images. These works aim to classify and then extract the data and semantic meaning of several types of charts: such as bar, pie and line charts. Similar to our work, [5]'s system classifies web images found in news sites by their functionality: including classes for story images, advertisements, server host images, icons and logos.

Textual features. Quite a bit of research focuses on the textual features related to an image. [2] and [5] performed classification based on textual features such as the filename, alternate text, hyperlink and text surrounding the image. Both papers deal only with web-accessible images, so hyperlinks are always available to be used. Attempts have also been made to detect and recognize text embedded in images. [6] and [7] use spatial variance and color segmentation techniques to separate text segments from graphics on an image. OCR or similar techniques often can extract the text from regions of the image. Using this technique, [8] detects text on images by examining connected components that satisfy certain criteria. Structure or comment metadata (i.e., MPEG-7) may also provide useful textual features in the future, but currently is not prevalent enough to affect classification performance. Taken altogether, it is probably unsurprising that [9] argues that textual features of images are far more useful in determining which images to return for a search query.

This will not work in cases where an image to be classified does not come from the web. Reliance on textual features might degrade the system performance when an image is not identifiable by these features, yet is easily associated with a category by the image's visual features.

Visual features. Most systems use simple visual features such as the most prevalent color, width-to-height ratio, image file type, among others. Using additional features from the image itself is the focus of Content Based Image Retrieval (CBIR). CBIR systems have progressively advanced, but practically all systems share a body of features based on the image's color histogram, texture, edge shape, and regions. From these low-level features, higher-level features that may have semantic meanings can be identified and built. For single images, region segmentation [10, 11] or block segmentation [12] is usually done followed by spatial layout based matching of regions or statistical feature extraction [13]. Feature analysis of the same color, salient points [14], texture and line features can then be assessed for individual regions and matched.

While CBIR has undoubtedly improved much over recent years, it remains a technology that has been mostly omitted from standard image search. This is largely due to the fact that searchers would rather type in a textual description to start. Automatic, content-based blind feedback on the top ranked images also does not seem to work, as text-based search followed by CBIR is computationally expensive.

3 Methodology

Given these observations, one architecture for improved image classification incorporates CBIR visual features with textual ones. This captures both the high accuracy and semantic nuances that textual features can garner, but enables classification based purely on visual features when text is not available.

In a nutshell, NPIC performs its task in three steps. Given a taxonomy of image classification, NPIC: 1) Constructs a dataset of sample images each class using traditional image search engines; 2) Extracts both textual and visual features from each sample image to create feature vectors for learning; 3) Builds discriminative models for each set of sibling classes in the hierarchy that originate from a common parent. Images can then be programmatically classified by generating their feature vector representations (step 2), followed by classification against the inferred models.

While this approach can be applied to any classification, we have specifically trained the NPIC system for synthetic images. We address synthetic images specifically as they often carry semantic content and data that are of interest to scholars and as well as the image analysis and digital library community.

An ontology of synthetic images

What is a proper taxonomy of synthetic images? To our knowledge, few classifications of synthetic images exist. In our exploration of related research, only Lienhart and Hartmann's work [1] addressed synthetic images specifically. In their work, synthetic images found on the web are classified into four distinct categories of photo-like images, presentation slides, scientific posters and comics. Another possible classification is the widely-used Getty Art and Architecture thesaurus [15]. The AAT is used mainly by museums and libraries to catalog visual materials. It employs a faceted classification

for objects, materials, activities, styles and periods (among others) and consists of over 133,000 generic terms.

A successful classification scheme must ensure that it can classify most items and that items clearly belong to distinct classes. For us, a successful classification needs to be simple enough such that an ordinary layman can understand and employ the classification scheme without needing specialist knowledge. Given these criteria, we feel neither Lienhart and Hartmann's classification (covers only certain types of web images) nor the Getty AAT schemes (too complex) work well.

Instead, our classification is based on what types of synthetic images a user encounters during her daily computing tasks. Our classification has five broad categories: *maps*, *figures*, *icons*, *cartoons* and *artwork*. We include *icons* as many images on a computer are icons associated with programs or data files. *Artwork* includes work drawings and pictures representing aesthetic images; *figures* include all types of abstract data representations. In our empirical analysis, this classification covers a large portion of important functional image types that users encounter.

As most images do not come labeled as synthetic or natural, we must include and implement a superordinate classifier to distinguish between *natural* and *synthetic* images for NPIC to be useful. Also, the two classes of *maps* and *figures* can be refined as they are quite general. We use the AAT to refine these two image types. The AAT has classifications for maps based on its form, function, production method, or subject. Based on our analysis, we conflated these schemes to produce a single subordinate classification of five categories: *plans*, *chorographic maps* (i.e., maps of large regions), *relief maps*, *weather maps* and *zip code maps*. Following the same editorial selection of the relevant AAT categories, we construct a categorization of *figures* into seven categories: *block diagrams*, *venn diagrams*, *bar graphs*, *pie charts*, *line graphs*, *tables*, and *illustrations*. Figure 1 shows our resulting classification hierarchy.

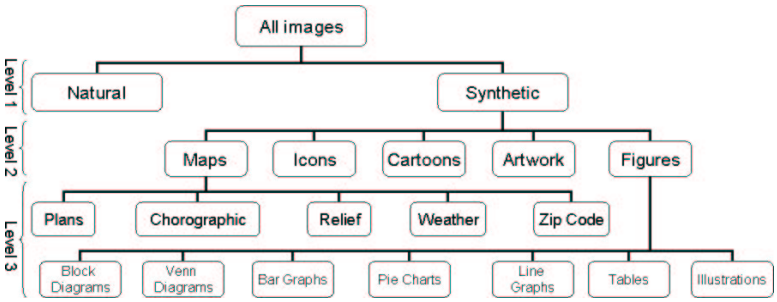

Fig. 1. NPIC's classification hierarchy.

We would like to emphasize that the hierarchy developed here constitutes a working attempt to compile a useable and useful classification to typical end users, and should not be construed as a formal model for synthetic image classification. Other image classes or alternate organizations can be also considered; such alternative classification schemes may work equally well in the NPIC framework.

Automatic corpus collection using image search

Given this classification, NPIC needs to collect labeled image samples to extract features for supervised learning. However, publicly available labeled synthetic datasets do not exist and creating one through manual efforts of annotating and selecting clean images is quite costly. However, as most machine learning algorithms are robust to small amounts of noise in their training data, we opt to create an image dataset by automatic means that may contain small amounts of mislabeled data. NPIC thus relies on the ratio of correctly labeled to mislabeled instances in training.

We do this by employing web image search engines. By searching for keywords that are indicative of the desired image category, we can form a noisy collection of images to use in training (hence our method is semi-supervised, as supervision is equated with image search engine relevance). The returned image dataset from any search is noisy, as image search engines occasionally return false matches. As long as the number of false hits is minimal, the image sets should generate useful training features for classification.

We follow this procedure to build image datasets for each of the image classes in the hierarchy. After associating each image class with a set of representative keywords (as shown in Table 1), we input these terms to Google's Image Search to find matching images. We build this dataset from the bottom up, as sample images from each child class can serve as positive examples for its parent. Given a ranked list of images for a class, we programmatically extract the URLs of the images for the first n hits. To help minimize the skew of the dataset, we extract a balanced corpus for each level (10K, 5K and .6K images for each of the three levels, respectively), balancing the number of images extracted from each keyword. We followed this procedure for all of the categories, except for *icons*, as we had access to a clean collection of icons.

Table 1. Some representative keywords for classes in our image hierarchy.

Level 1	photograph	aerial, birthday, bedroom, central library, concert, face
Level 2	artwork	painting, drawing, artwork
	icons	<*separate icon collections used*>
	cartoons	cartoon, disney, anime, garfield
Level 3	plans	floor, plan, fire escape
	table	data, excel
	illustration	illustration, DNA molecule, engine

4 Features

Once the corpus was collected, each image was processed to extract textual and visual features for training and testing. As our paper does not focus on the feature creation, we only give a brief inventory of the features used in Table 2. These features have been chosen as they have been shown to be useful for image classification (natural as well as synthetic) in past work, as referenced in the final column of the table. We use standard utilities to extract both sets of features: the `identify` utility from the ImageMagick library to extract image metadata from the header; and for visual features, the OpenCV suite of visual detectors and the `xpm` package to examine the raster data.

A short discussion about the features is necessary. Textual features were created by extracting tokens from the filename, extension, and path information from the URL (when available) of the image. For this, simple tokenization was done to create a more

meaningful inventory of features (`garfield_2.jpg` → `garfield 2 jpg`) and to reduce problems with sparse data.

Table 2. Features in NPIC. References indicate past published work using this feature.

Feature	Description	Refs.
Textual Features - via analysis and header metadata		
Filename	Image filename without extension	[2, 9, 5]
File extension	Extension of the file, if any	[2, 9, 5]
Comments	Comments in Image metadata header	*new*
Image URL	URL components of the location of the image on the Web (if applicable)	[2, 9, 5]
Page URL	URL components of the enclosing page of the image	[2, 9, 5]
Visual Features - header information, raster via `XPM`, or shape detection via `OpenCV`		
Height	Image dimensions in pixels	[2, 1, 5]
Width		[2, 1, 5]
X resolution Y resolution	Number of pixels per inch (dpi) along X and Y dimensions	*new*
C_1	Most common color	[5, 2]
C_1 Fraction	Fraction of pixels in the image that have color C_1	[5, 2]
$F1$	Fraction of pixels with the neighbor metric greater than zero	[1]
$F2$	Fraction of pixels with the neighbor metric greater than 1/4 of the maximum	[1]
F2/F1	The ratio of F2 to F1	[1]
L1 distance	L1 $= \sum(\|h_i - k_i\|)$, where $H = \{h_i\}$ is the image histogram, and $K = \{k_i\}$ represents the average histogram in each category	[16]
L2 distance	L2 $= (\sum \|h_i - k_i\|^2)^{1/2}$	[16]
L-∞ distance	L-∞ $= (\sum(\|h_i - k_i\|)^{100})^{1/100}$, a large value of 100 is chosen to represent infinity	[16]
Jeffrey divergence distance	$\sum((h_i log(h_i/m_i) + k_i log(k_i/m_i))$, where $m_i = (h_i + k_i)/2$	[16]
Chi^2 distance	$\sum((h_i - m_i)^2/m_i)$ where $m_i = (h_i + k_i)/2$	[16]
Quadratic distance	$d_A(H, K) = \sqrt{(\mathbf{h} - \mathbf{k})^T \mathbf{A}(\mathbf{h} - \mathbf{k})}$, where **h** and **k** are vectors that list every entry in **H** and **K**. Cross-bin information is incorporated via a similarity matrix **A** = $[a_{ij}]$ where a_{ij} denotes similarity between bins i and j.	[16]
EMD	Earth Movers Distance: $EMD(P, Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij}}$	[17][16]
Rectangles	2 features: Number of rectangles whose sides are parallel to the image frame, fraction of entire image occupied by rectangles	[17]
Circles	Number of circles with certain radius	*new*
Corners	Number of corners found on the image	*new*
Lines	5 features: Number of horizontal, vertical and slanted lines; average line length and average line gradient	*new*

We have chosen to use many color features for visual features as they are relatively straightforward to calculate given raster data. We follow the literature and use both the HSV and RGB color spaces for analysis. For neighbor metrics, we create features using the standard RGB and HSV color spaces, as well as reduced HS and H spaces. Color histogram features are calculated using a simplified 9-bit RGB color space. This is done by first obtaining an average histogram over all training samples in a class. Then for each testing image, we calculate the difference between the class average and the image's histogram. A total of n features are generated, where n is the number of classes in the classifier (e.g., 5 for the second level). A number of different distance measures are used: Minkowski-form (L1, L2 and L-∞), Jeffrey Divergence, Chi-square, Quadratic distances as well as EMD.

For the rectangle, line, circle and corner detection features, we need to specific settings for the spatial and scaling constraints of the detectors. Using a *laissez faire* approach, we use a wide variety of parameter settings to create different features and forward these to the learner to decide which group of parameter settings should be used.

5 Evaluation

There are two questions that we would like to answer with our evaluation: 1) how well does NPIC perform? 2) how do the different textual and visual features interact to achieve its performance?

Image datasets. We tested NPIC's performance on two datasets of image data. The first is the original corpus of 15,600 images that was obtained by automatically downloading pictures from Google Image Search. The second corpus consists of a subset of 1,300 images (200, 500 and 600 images for levels 1, 2 and 3, respectively) retrieved from the Wikipedia Commons. The Wikipedia Commons is a license-free repository of media files free for anyone to use in any way. These datasets are available from our NPIC website, to facilitate further research in the field[1]. These datasets are entirely independent of each other.

Procedure. After obtaining the datasets, each dataset was hand-labeled by the first author (for evaluation only – we rely on the assigned labels from the keyword search in training). For the Google dataset, we performed five-fold cross validation; that is, we used 4/5ths of the data to train a model and 1/5 for testing, and repeated this process five times and averaging the performance. For the Wikipedia dataset, the entire Google dataset was used for training a model, and tested on the Wikipedia set. A boosted decision list learner, BoosTexter [18], was used as the machine learner, as its inferred rules are easy to interpret. The learner was asked to do 300 rounds of boosting (i.e., 300 serial rules inferred) for each classifier. The rules also easily lend themselves to an analysis of which features are helpful. For succinctness, Table 3 shows only the resulting accuracy; precision, recall and F_1 are intentionally omitted.

We observe several trends from the results. First, accuracy increases as we go from the specific Level 3 classifiers towards the Level 1 classifier. This is expected, as the Level 3 classifiers are more fine-grained and are harder, 5- or 7-way decision problems. Second, accuracy on the Wikipedia dataset is lower across the board. Specifically, the textual features are less helpful than the visual ones. This is partially due to the fact that URLs are not available in this dataset and that the filenames are not nearly as indicative of the class as in the Google dataset (after all, filenames are partial evidence for relevance in Google's image search, used to construct the dataset). The visual features show roughly the same performance on both data sets. As such, we feel that the test on the Wikipedia dataset is more realistic and representative of what would be encountered in practice. Third, maps are harder to classify than figures, as the figure subcategories have notably different visual features that are captured by the OpenCV detectors. Fourth, icons do extremely well, as their extension in Windows is a fixed `.ico` and we start with a clean corpus, unlike any of the other sets. Finally, although

[1] `http://wing.comp.nus.edu.sg/npic/`

Table 3. Performance of NPIC on the two datasets, with different feature sets.

Level	Class	Average C.V. accuracy (Google)			Testing accuracy (Wikipedia)		
		Text (T)	Visual (V)	V + T	T	V	V + T
Level 1	Synthetic	99.4%	95.9%	99.9%	94%	93%	95%
	Natural	99.7%	93.5%	99.9%	90%	92%	94%
	Total	99.6%	94.7%	99.9%	92%	92.5%	94.5%
Level 2	Map	94.3%	87.6%	98.5%	78%	77%	86%
	Figure	90.5%	82.9%	98.7%	74%	78%	90%
	Icon	100.0%	77.6%	100.0%	95%	91%	96%
	Cartoon	89.2%	73.6%	97.6%	69%	84%	81%
	Artwork	92.5%	67.0%	93.2%	73%	74%	79%
	Total	93.3%	77.7%	97.6%	77.8%	80.8%	83.4%
Level 3	Block diagram	84%	86%	84%	72%	82%	86%
(Figure)	Venn diagram	88%	86%	90%	70%	88%	90%
	Bar graph	84%	78%	82%	78%	78%	74%
	Pie chart	82%	86%	90%	80%	86%	86%
	Line graph	80%	78%	80%	66%	74%	76%
	Table	78%	68%	82%	72%	72%	76%
	Illustration	82%	78%	82%	74%	80%	82%
	Total	82.6%	80.0%	84.3%	73.1%	79.9%	81.4%
Level 3	Plan map	86%	76%	86%	82%	78%	84%
(Map)	Chorographic map	86%	80%	88%	78%	82%	82%
	Relief map	90%	68%	84%	70%	70%	72%
	Weather map	84%	64%	84%	74%	66%	72%
	Zip code map	96%	72%	92%	88%	72%	86%
	Total	88.4%	72.0%	86.8%	78.4%	73.6%	79.2%

the performance is not directly comparable with prior reported results (as the problem specifications and datasets differ), the NPIC classifiers show similar performance. The advantage here is that NPIC system uses a set of very general, coarse features that are inexpensive to compute and applicable to a wide range of problems. Classifiers aimed at specific tasks (c.f., [19]) are bound to do better in their stated problem domain.

Given that image search primarily employs textual features, are the improvements by incorporating visual features significant? We compared the textual versus the combined feature judgments using Student's 2-tailed T-test. Our findings indicate a significant ($p < .05$) for both Level 2 classifiers but not the Level 1 or 3 classifiers. We believe the reason for this is simply because there are too few images for the Level 3 classifiers (600 for both Level 3 classifiers) and for the Level 1 Wikipedia classifier (1000).

To assess the efficacy of the feature sets, we explore the resulting classifiers. Table 4 shows the first 100 features used by each of the four inferred models (with repetitions omitted). We see that individual words (each a separate feature) constitute a large fraction the useful features in the Level 1 and 2 classifiers, but a smaller fraction of Level 3 features (validating our earlier claim). We also see that the color histogram distance measures play a larger role in the fined-grained classifiers, and that no one distance measure is best: they all seem to be used by the classifier for discriminating in different instances. Finally, our OpenCV features have been effective for the classes we suspect:

circles are used in the *figure* classifier and vertical/slanted lines in the *map* classifier (perhaps for deciding between building plans vs. natural region maps).

For the OpenCV detectors, the learner found optimal settings through cross-validation separation. For the circle detector, a diameter setting of $d = 0.3 \times min(height, width)$ performed best, as lower settings of d would find many spurious results; the rectangle detector was set to detect only ones parallel to the image frame.

Table 4. Salient features found in the BoosTexter models.

Level	Textual Features	Visual Features
Level 1	jpeg smsu co jennifer friends azoft stylefest gif map painting pie shtml a search drawing areas iconfan serials paris freeyellow online ru tv sponsors sponsors k12 eastburtonhouse	Quantum, C_1, F1/D2, Magick, L-∞, C_1 Fraction, Colors, Height, Background$_H$
Level 2	map painting artwork drawing ico cartoon venn graph diagram disney pie anime garfield maps physics www directory chemistry comics com world artwork art maths archie chem. page street au image tintin gifs sg city hein edu books chinese asp sun moaa gov nr 278 nice chart assembled ga, region	Width, F1/D3, #slantedLines, F2/D1, Quad$_{artwork}$ #HorizontalLines, Quad$_{icon}$, Height, AvgLineLength, Background$_H$, averageLineGradient, F1/D4, F1/D1, Size, F1-F2/D1, EMD$_{diagram}$, JD$_{artwork}$, EMD$_{artwork}$, X-resolution
Level 3 (Figure)	block pie venn bar table diagram data archives illustration 2 barograph none gov charts htm cty us edu venndiagram fag articles hisoftware en cfm 0805rettable pubs	#SlantedLines, #circles, Chi$^2_{block}$, AvgLineGradient, #HorizontalLines, Width, X-resolution, Size, Chi$^2_{diagram}$, #VerticalLines, Colors, AvgLineLength, EMD$_{block}$, Background, Y-resolution, EMD$_{pieChart}$, JD$_{pieChart}$, L1$_{barGraph}$, L2$_{graph}$, EMD$_{block}$, Height, Quad$_{block}$, L-∞_{block}
Level 3 (Map)	weather plan relief noaa gov weather country map us maps com plan wunderground leone scbtvsworld graphics planning wr php province map asp files ca	#SlantedLines, EMD$_{region}$, #HorizontalLines, AvgGradient, #Corners, L1$_{relief}$, EMD$_{relief}$, AvgLineLength, EMD$_{weather}$, L1$_{zipAreaCode}$, L-∞_{relief}, EMD$_{weather}$, L2$_{zipAreaCode}$, JD$_{plan}$, QuadDist$_{zipAreaCode}$, #VerticalLines, Height, EMD$_{plan}$, FractionOccupiedByRectangles, L-$\infty_{weather}$

6 Conclusion

We have introduced NPIC, a system specifically trained for synthetic image classification. This system is fully automated and distinguishes between natural vs. synthetic images, and types synthetic images into five classes, of which *maps* and *figures* are further subdivided. We obtain the image datasets by standard text-based image search using keywords highly correlated with each class. This noisily labeled corpus serves as training data, making our classification scheme semi-supervised. In all cases, performance of the classifiers increases when simple color and geometric shape detection features (specifically for particular synthetic image classes) are added. A key result is that visual features make a stronger contribution than the textual ones when fine grained classification is needed.

NPIC is based on a general framework that relies on the scale of image search engines to sift away noise from the training data. Such a framework could be extended to natural image classification, where much of image retrieval research is centered on. We expect to further improve NPIC in the future by 1) using the relevance ranking of the images from search engine in weighting examples for training, and 2) exploring how to

find keywords automatically for training data acquisition. We plan to achieve the latter using mutual information which can provide a list of statistically correlated modifiers for a base keyword. We have already done a detailed error analysis on the dataset, and have additional features in mind that may help to improve performance.

References

1. Lienhart, R., Hartmann, A.: Classifying images on the web automatically. Journal of Electronic Imaging **11** (2002)
2. Swain, M.J., Frankel, C., Athitsos, V.: Webseer: An image search engine for the world wide web. In: International Conference on Computer Vision and Pattern Recognition. (1997)
3. Huang, W., Tan, C.L., Loew, W.K.: Model-based chart image recognition. In: Proceedings of the International Workshop on Graphics Recognition (GREC). (2003) 87–99
4. Carberry, S., Elzer, S., Green, N., McCoy, K., Chester, D.: Extending document summarization to information graphics. In: Proceedings of the ACL-04 Workshop. (2004) 3–9
5. Hu, J., Bagga, A.: Functionality-based web image categorization. In: Proceedings of WWW 2003. (2003)
6. Cao, R., Tan, C.L.: Separation of overlapping text from graphics. In: Proc. of the 6th International Conference on Document Analysis and Recognition (ICDAR '01). (2001) 44
7. Zhong, Y., Karu, K., Jain, A.K.: Locating text in complex color images. Pattern Recognition **29** (1995) 1523–1535
8. Zhou, J., Lopresti, D.: Extracting text from www images. In: Proc. of the 4th International Conference on Document Analysis and Recognition. Volume 1. (1997) 248 – 252
9. Munson, E., Tsymbalenko, Y.: To search for images on the web, look at the text, then look at the images. In: Proc. of the 1st international workshop on Web Document Analysis (WDA '01). (2001)
10. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1026–1038
11. Ma, W.Y., Manjunath, B.: NaTra: A toolbox for navigating large image databases. In: Proc. IEEE International Conference on Image Processing. (1997) 568–71
12. Smith, J.R., Chang, S.F.: Quad-tree segmentation for texture-based image query. In: Proceedings of the 2nd Annual ACM Multimedia Conference, San Francisco, CA (1994)
13. Wang, J.Z., Li, J., Chan, D., Wiederhold, G.: Semantics-sensitive retrieval for digital picture libraries. D-Lib Magazine **5** (1999)
14. Tian, Q., Sebe, N., Lew, M.S., Loupias, E., Huang, T.S.: Image retireval using wavelet-based salient points. J. of Electronic Imaging **10** (2001) 835–849
15. Getty Institute: Art and architecture thesaurus (2006) http://www.getty.edu/research/ conducting_research/vocabularies/aat/.
16. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40** (2000) 99–121
17. Qin, L., Charikar, M., Li, K.: Image similarity search with compact data structures. In: Proc. of the 13th ACM conference on Information and knowledge management, Washington, D.C. (2004)
18. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning **39** (2000) 135–168
19. Ng, T.T., Chang, S.F., Hsu, J., Xie, L., Tsui, M.P.: Physics-motivated features for distinguishing photographic images and computer graphics. In: Proc. of the ACM Int'l Conf. on Multimedia, Singapore (2005) 239–248