Image Landmark Recognition with Hierarchical K-Means Tree

Magdalena Rischka, Stefan Conrad

Institute of Computer Science
Heinrich-Heine-University Düsseldorf
D-40225 Düsseldorf, Germany
{rischka, conrad}@cs.uni-duesseldorf.de

Abstract: Today's giant-sized image databases require content-based techniques to handle the exploration of image content on a large scale. A special part of image content retrieval is the domain of landmark recognition in images as it constitutes a basis for a lot of interesting applications on web images, personal image collections and mobile devices. We build an automatic landmark recognition system for images using the Bag-of-Words model in combination with the Hierarchical K-Means index structure. Our experiments on a test set of landmark and non-landmark images with a recognition engine supporting 900 landmarks show that large visual dictionaries of size about 1M achieve the best recognition results.

1 Introduction

Today's giant-sized image databases on the World Wide Web, in personal households and on mobile devices pose great challenges for the user: to manage and use these images in a meaningful way it is necessary to know the images' contents. The contents of web images embedded in web pages can be exploited somehow by analysing the surrounding web text, however images captured by digital cameras or other mobile devices usually include only technical metadata which do not reveal the contents of the image (except GPS data). In these cases the user has to annotate the images with meaningful concepts. Manual content exploration is time-consuming, in such large-scale scenarios even intractable, therefore automatic content-based solutions are required. We focus on the exploration of personal image collections and assume that images have no metadata available. A large amount of the personal collections' images are photos shot in the photographer's vacations and trips showing (prominent) places and landmarks. We address the domain of landmark recognition in images as this topic offers several advantages regarding applications in personal image collections, as well as in the field of the World Wide Web and on mobile devices: the annotation is a basis for a search or can be used as a suggestion for a photo description to the user, the identification of locations by the recognition of landmarks can be used to summarize personal image collections by offering an overview of places the photographer visited. The application of mobile landmark recognition enables tourists to look up sights in real-time to obtain information on them.

Several systems for automatic landmark recognition have been proposed [GBQG09, AKTS10, ZZS$^+$09, PZ08, CBHK09]. Each work addresses different aspects of landmark recognition, the creation of the landmark database, the general problem definition, the initial situation referring metadata (for example if the GPS data of test images is used) and the techniques used, however all of them offer a content-based approach for recognition. We outline these related work with focus on the image representation (features and visual dictionary size in case of Bag-of-Words model or index structures), recognition technique (classifier and/or index structure), dataset (esp. number of landmarks supported) and specifics used. [CBHK09] recognizes images using a Bayesian and a SVM classifier on SIFT-based Bag-of-Words image representation (created with K-Means clustering) with the vocabulary size 1K. The dataset contains 33M self-downloaded Flickr photos and the number of landmarks supported in the evaluation is 10, 25 and 50. [AKTS10] also creates a dataset from Flickr images and then derives scene maps of landmarks which are retrieved with a kd-tree index. The image representation bases on SURF features and a visual dictionary of size 75K. The authors of [GBQG09] create a database from geo-tagged Flickr photos and Wikipedia. The recognition is performed on object-level and with the aid of an approximate K-Means index on SIFT features which constitutes a dictionary of size 500K. [PZ08] uses SIFT features and a Bag-of-Words approach with a dictionary size of 1M. For the dictionary again approximate K-Means index structure is used. The evaluation is performed on *The Oxford Buildings Dataset*[1] (11 landmarks). [ZZS$^+$09] creates the database by crawling travel guide websites (5312 landmarks) and then builds a matching graph out of the feature matches of the images. Images are represented using 118-dim Gabor wavelet texture features with PCA. For retrieval a kd-tree is used, however the size of the underlying dictionary remains unknown. Allmost all presented systems use SIFT features (or the like) with a Bag-of-Words image representation (based on flat K-Means) or a feature index structure like the kd-tree or approximate K-Means. Unfortunately each proposed system is evaluated on different datasets (with different number of landmarks) and different dictionary sizes are used, thus the comparison between methods is difficult. There exists some public databases for landmark recognition, however they are not always suitable for the own problem definition. For example the public datasets *The Paris Dataset*[2] and *The Oxford Buildings Dataset* contain a low number of landmarks (about 12 and 17), which can not be used when developing a landmark engine which has to support thousands of landmarks. Our own experiments [RC14] on different classifiers, dictionary sizes (from 500 to 8K, derived from flat K-Means clustering) and different number of landmarks (45, 300, 600, 900) show that the kNN classifier outperforms the other ones (like SVM) in a large-scale scenario, when the number of landmarks supported increases. Furthermore increasing the dictionary sizes increases the performance. Large (flat) K-Means dictionaries however lead to inefficient recognition times, thus the access to the features/visual words have to be supported by efficient quantizer/index structures. [PJA10] analyses some structured and unstructured quantizer, the families of Locality Sensitive Hashing (LSH) functions as well as the K-Means and the Hierarchical K-Means on SIFT descriptors extracted from a subset of the *INRIA Holidays dataset*[3]. These experiments show that unstructured

[1] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/

[2] http://www.robots.ox.ac.uk/∼vgg/data/parisbuildings/

[3] http://lear.inrialpes.fr/people/jegou/data.php

quantizer like the K-Means and the Hierarchical K-Means outperform LSH functions, as they adapt better to the distribution of the features. We develop an automatic content-based image landmark recognition system with a SIFT-based Bag-of-Words image representation in combination with the Hierarchical K-Means (HKM)[NS06] tree index structure and evaluate our system on a large-scale dataset which supports 900 landmarks. Our contribution is to analyse in how far an increasing dictionary up to 3M can improve the recognition performance and whether the HKM tree parameters branching factor and height or only the resulting number of leaf nodes (dictionary size) have an impact on the recognition. Furthermore we analyse the kNN voting distribution of landmark and non-landmark images and based on this we propose a simple and efficient verification filter to decide whether an image contains a landmark or not. In literature this verification is usually done with expensive model fitting algorithms like RANSAC. The remainder of this paper is organized as follows: section 2 introduces into the landmark recognition problem and presents our implemented landmark recognition system with its steps. In section 3 we evaluate our system comparing it to a baseline approach. Finally (section 4) we summarize our results and discuss future work.

2 Automatic Landmark Recognition System

The Landmark Recognition Problem A *landmark* is a physical object, created by man or by nature, with a high recognition value. Usually a landmark is of remarkable size and it is located on a fixed position of the earth. Examples of landmarks are buildings, monuments, statues, parks, mountains and other structures and places. Due to their recognition value, landmarks often serve as geographical points for navigation and localisation. A landmark recognition system has to conduct the following task automatically:

Definition 1 (Landmark Recognition Task)**.** *Given a set of L landmarks $\mathcal{L} = \{l_1, ..., l_L\}$ and an image i which contains the landmarks $\mathcal{G}_i \subseteq \mathcal{L}$. The task for a landmark recognition system is to assign a set of landmarks $\mathcal{P}_i \subseteq \mathcal{L}$ to the image i in such a way that $\mathcal{P}_i = \mathcal{G}_i$.*

$\mathcal{G}_i$ is the *groundtruth set* and $\mathcal{P}_i$ the *prediction set* of the (test) image i. Please note that $\mathcal{G}_i = \emptyset$ and $\mathcal{P}_i = \emptyset$ are possible. The landmark recognition task is a multi-label classification problem including a decision refusal.

The Design of the Landmark Recognition System Our automatic landmark recognition engine bases on a single-label classification approach with the Bag-of-Words image representation and the Hierarchical K-Means tree index structure. This single-label classification approach can be extended to multi-label one (as defined in the Landmark Recognition Task) by applying a postprocessing step which analyses if locally adjacent (ref. city) landmarks (of the classified landmark) are available in the image. This step is not object of this work. Our system consists of two stages: the training stage, in which the recognition engine is learned, and a recognition stage, in which the landmarks in the (test) image are recognized. Figure 1 shows the overall design of our system. The training phase (red path in the figure) begins with the feature extraction step, in which for each training image SIFT features are extracted. Next the HKM tree is created. For this purpose we select a subset of all SIFT descriptors from the training set and based on this subset we create the HKM tree. The created HKM tree bases only on the SIFT descriptors, still uncoupled from the corresponding training images. For later classification purposes we have to feed the HKM tree with information on the training images. Before the tree feeding step we determine the image representation using the created HKM tree. In the next step the HKM tree is fed with the training images using the information from the training image representation. The resulting HKM tree is ready to use for the recognition phase. In the recognition stage (green path) the test image has to pass the same steps of feature extraction and image representation as the training images. Then the test image is classified to one landmark. In the verification step the classification result is verified, i.e. we check whether the recognized landmark is really available in the image. In this step an assignment of a landmark label to a non-landmark test image should be rejected. The following subsections present each step of the system in detail.

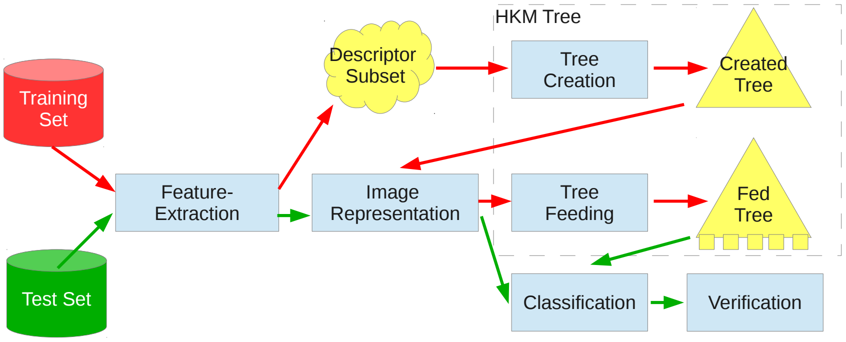

Figure 1: The design of the landmark recognition system

Feature-Extraction We describe images with the local features *Scale Invariant Feature Transform* (SIFT) [Low04]. To extract these features the SIFT algorithm detects stable points in the image and then describes the (small) surrounding area around each point by a histogram of gradients. Finally the SIFT representation of an image i is a set of local SIFT points p: SIFT$(i) = \{p_1, ..., p_P \mid p = (x, y, s, d)\}$ with x, y are the coordinates of the stable point p in the image, s is the scale influencing the size of the surrounding area and d is the 128-dimensional descriptor which is the histogram of gradients.

Image Representation For the image representation we use the Bag-of-Words (BoW) approach. The idea behind BoW is to aggregate local features to one global descriptor and thus to avoid the expensive comparison of images by matching local descriptors against each other. The BoW descriptor bases on a dictionary of visual words which usually is obtained by partitioning the descriptor-space. Then each partition is represented by an instance of this partition, usually the center of the partition, which is called the *visual word*. A simple and most used method to partition is the use of the K-Means clustering algorithm. However flat K-Means do not scale well with an increasing dictionary size: already dictionary sizes of 10K lead to long recognition times. Applying the HKM tree enables us to use large-sized dictionaries with over 1M words. To create the image representation of a training or test image each SIFT descriptor of the image is looked up in the tree to get the leaf as a visual word. Then the relative frequency of each leaf/visual word in the image is determined. The final image representation is a set of visual words with the corresponding relative frequency.

HKM Tree *General Information* The HKM tree is a tree-based index structure introduced by [NS06]. We use this index structure to maintain the large 128-dimensional SIFT descriptor data of the training images with references to the corresponding training images. The use of an index structure enables fast look-ups of SIFT descriptors and thus an efficient classification of test images at the recognition stage. The HKM tree is created by recursively dividing the data space into disjoint regions using the clustering algorithm K-Means.

Tree Creation For HKM tree creation we take a large-sized and representative subset of all SIFT descriptors of the underlying training set as the root data. For a fixed value K we cluster the root data with K-Means into K clusters. For each resulting cluster consisting of the data part belonging to this cluster we again cluster this data into K clusters. This step is repeated until the data to be clustered reaches a size of $\leq 2K$ data points. This procedure automatically builds a tree of a fixed branching factor K (for each level of the tree) and a tree height h_c (c stands for *created*) determined by the longest clustering path. Each node in the tree corresponds to a data cluster which is representated by a centroid. The tree leaves correspond to the final data partitions.

Tree Feeding In the tree feeding step the relation between the SIFT descriptors and the corresponding training images is established by registering the training images in the leaves of the created HKM tree. To register a training image we take its image representation which is a set of visual words (leaves) with their relative frequencies in this image. The training image is registered in all occurring visual words/leaves of the tree with its relative frequency. Finally each leaf of the HKM tree contains a list of all training images which contain this leaf. To influence the number of leaves of the HKM tree, thus the size of the visual dictionary, we set a height parameter h_f (f stands for *fed*) for the tree which cuts the original created tree and limits its height.

Classification After the test image passed the feature extraction and the image representation step, it is classified to a landmark. For each visited leaf/visual word in the image representation step the training images registered in these leaves are taken for similarity calculation. The similarity between the test image and a training image is determined by the histogram intersection of the relative frequency values of the common leaves. The kNN classifier considers the k nearest training images of the test image. Each of the k nearest training images votes for the landmark it *belongs* to. Thus we can have a maximum of k candidate landmarks for the classification, whereas each candidate landmark can be voted by a maximum of k training images. The number of training images which vote for the finally classified landmark is the *voting score* v. In this work the kNN parameter k is set to 5 as a result of preliminary experiments. We choose a kNN classifier instead of other well-known classifiers because the kNN has shown classification superiority in large-scale scenarios referring a large number of classes [DBLL10, RC14].

Verification To decide whether a test image contains the classified landmark, we apply the following simple filter based on the kNN classifier:

Definition 2 (Simple kNN Filter)**.** *Given the test image i which has been classified to landmark $\mathcal{P}_i = \{l\}$ with a kNN voting score of v. The classification result of image i is reliable if the voting score v is equal to or exceeds a threshold value $t \in \{1, ..., k\}$, i.e.*

$$i \rightarrow \begin{cases} \emptyset & \textit{if } v < t \\ \mathcal{P}_i & \textit{if } v \geq t \end{cases} \tag{1}$$

3 Evaluation

Evaluation Dataset For the evaluation we use a self-provided dataset, as there are no public datasets for landmark recognition available which support large number of landmarks. We gathered 900 landmark terms (from 449 cities and 228 countries) from several websites which list landmarks from all over the world, including the website of [ZZS^{+}09][4]. To get images for the training and test sets, we queried the Google image search engine with each landmark term (specified by its region - city or country) and then downloaded the results from the original source. From the downloaded image set we derived three training sets (A, B, C) and a test set. Each training set supports (the same) 900 landmark terms. The test set supports only a subset of the 900 landmark terms: 45 landmark terms (well- and lesser-known landmarks from Europe) with always 20 images per landmark, resulting in 900 *landmark images*. We have choosen the test images manually to ensure their correctness of containing the corresponding landmark and to create a challenging test set: the test images show the landmark in their canonical views, under different perspective changes, distortions and lighting conditions (also at night) as well as indoor shootings and parts of the landmark. To simulate a realistic test set we extended the test set with *non-landmark images*, i.e. images which do not contain any landmark. For this we took a subset of the *Flickr 100k dataset*[5]. All test images have been proofed to be (visually) disjoint from the images of the training sets.

Evaluation Measures To evaluate the performance of the system we use example-based measures which judge the recognition quality for one test image and then we report the average over all test images. In the general problem of multi-label classification we apply example-based Precision and Recall which for an image i with its groundtruth set $\mathcal{G}_i$ and prediction set $\mathcal{P}_i$ are defined as follows:

$$\text{Precision}(i) = \frac{|\mathcal{G}_i \cap \mathcal{P}_i|}{|\mathcal{P}_i|} \qquad \text{Recall}(i) = \frac{|\mathcal{G}_i \cap \mathcal{P}_i|}{|\mathcal{G}_i|} \tag{2}$$

The special cases of Precision and Recall for $\mathcal{G}_i = \emptyset$ and/or $\mathcal{P}_i = \emptyset$ are defined in figure 2. For the restriction to single-label classification, i.e. $|\mathcal{G}_i| = |\mathcal{P}_i| = 1$ Recall and Precision

[4] http://mingzhao.name/landmark/landmark_html/demo_files/1000_landmarks.html

[5] http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/flickr100k.html

have the same value, thus we report only one of them (Recall). Then the average Recall over all test images reports the amount of test images classified correctly, i.e. a landmark image is classified to the correct landmark, a non-landmark image is detected correctly as non-landmark image.

Experiments on the Baseline System In the first experiment we present the results of our baseline approach described in [RC14]. The baseline is a single-label classification approach which uses SIFT features and the Bag-of-Words image representation on a visual dictionary created by (flat) K-Means clustering. For the classifier used, we tested on the 5NN and on an SVM. For the 5NN the baseline system can be seen as a specific case of the HKM tree-based system, in which the HKM tree is obtained by one clustering process resulting in a tree of height 2 (root data level and clustered data level). The Bag-of-Words model has one parameter which is the visual dictionary size. We examine this approach on the following five visual dictionary sizes: 500, 1K, 2K, 4K and 8K. Here larger dictionaries are not considered due to long recognition times using the (flat) K-Means algorithm. Figure 3 shows the recognition results (Recall) of the landmark images (non-landmark images of the test set are not considered) for the classifier 5NN and SVM, before applying the verification step, and depending on the visual dictionary size. Values reported are averages over the three training sets A,B,C. The results reveal that the larger a visual dictionary the better are the classification results. From a dictionary size of 4K on, the 5NN outperforms the SVM and the behaviour of both plots indicate that this can still hold for larger dictionaries. The best recognition result with a Recall value of 0.43 is achieved with a dictionary size of 8K. The trend suggests, that for even larger dictionaries (larger than 8K) the Recall value furthermore grows. This assumption is examined in the next experiment.

K	20	25	15	20	25	15	30	25	30	20	15
h_f	4	4	5	5	5	6	5	6	6	7	8
max	160K	390K	759K	3,2M	9,7M	11M	24M	244M	729M	1,2B	2,5B
real	8T	15T	50T	160T	390T	757T	809T	1M	828T	2,5M	1,4M

Table 1: Maximum ($max = K^{(h_f)}$) and real number of leaves/dictionary sizes of the created HKM trees depending on the branching factor K and the height h_f

Experiments on the HKM Tree-based System This experiment evaluates the HKM tree-based approach on small up to very large dictionaries. To create the HKM tree we take a (representative) subset of 10M SIFT descriptors from the corresponding training set and create trees for the following branching factors K: 15, 20, 25, 30. Then we learn trees (tree feeding step) for different height values, for the height h_c (original created tree) and for smaller heights h_f (tree cuts). Table 1 shows the maximum and the real number of leaves of the trees created depending on the branching factor K and the tree height h_f. Figure 4 shows the recognition results of the test set depending on the parameter visual dictionary size/real number of leaves (x-axis) and the threshold t of the verification step (plots, one color for a fixed threshold). The Recall values reported are averages over the three training sets A,B,C. The diagram on the left of figure 4 shows the 'global' recognition value which is the average over all test images (landmark and non-landmark images). The diagram on

$\mathcal{G}_i$	$\mathcal{P}_i$	Precision	Recall
$\emptyset$	$\emptyset$	1	1
$\emptyset$	$\neq \emptyset$	0	0
$\neq \emptyset$	$\emptyset$	0	0

Figure 2: Definition of Precision and Recall for combinations of $\mathcal{G}_i = \emptyset$ and $\mathcal{P}_i = \emptyset$

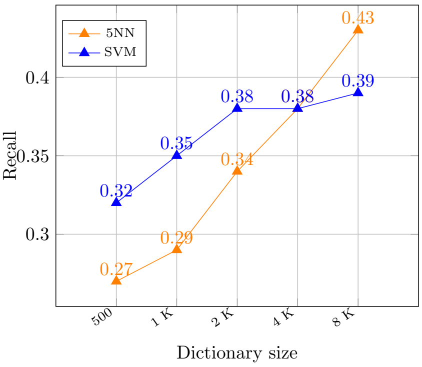

Figure 3: Recognition results of the baseline approach on landmark images

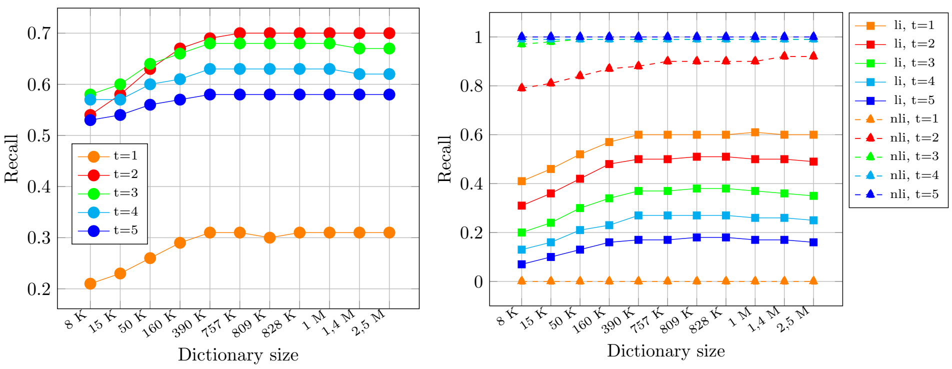

Figure 4: Recognition results of the HKM tree-based approach. Left diagram: on all test images. Right diagram: on landmark (li) and non-landmark images (nli) separately.

the right shows the recognition value for landmark and non-landmark images separately (as averages over all landmark images and all non-landmark images, respectively). The results confirm the assumption derived from the baseline system that larger dictionaries achieve better recognition quality. Up to a dictionary size of 390K the Recall values rise continuously when enlarging the visual dictionary. From a dictionary size of 390K on the recognition value for all test images (left diagram) and for all landmark images (right diagram) levels out and reach the best Recall value of 0.7 and 0.6 on dictionaries of size 800K to 1M for a threshold $t = 2$ and $t = 1$, respectively. Comparing the Recall value of 0.6 ($t = 1$) on landmark images with the best result of the baseline system, we achieve a recognition improvement of 17% by enlarging the dictionary. We can also see that the Recall value for the landmark images, thus for all test images, begins to fall slightly from a dictionary size of 1,4M. However the Recall value of the non-landmark images rises uninterrupted with an increasing dictionary size (for $t > 1$). Please note that for $t = 1$ (equals to no verification step), all test images are assigned to one landmark, thus all non-

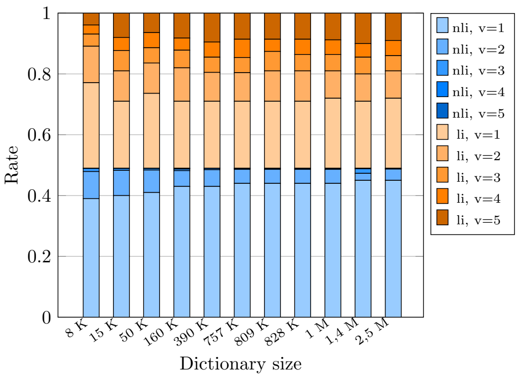

Figure 5: Distribution of the voting score v for $v \in \{1, 2, 3, 4, 5\}$ on landmark (li) and non-landmark images (nli).

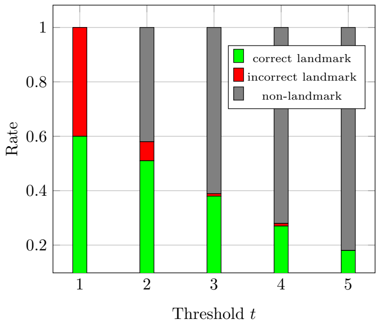

Figure 6: Amount of *correct*, *incorrect* and as *non-landmark* recognized landmark images for a dictionary of size 828K depending on the threshold t.

landmark images are recognized incorrectly and get a Recall value of 0. Due to this fact, the recognition for $t = 1$ on all test images results in a low Recall value. Figure 5 shows the distribution of the voting score v among the whole test set, represented for landmark and non-landmark images separately. The largest amount of test images (concerns both, landmark as non-landmark images) has a voting score $v = 1$. For non-landmark images this amount grows with an increasing dictionary size and is a good indicator for the prediction of non-landmark images. Landmark images are stronger represented with voting scores of $v \geq 2$ than non-landmark images, unfortunately about half of the landmark images still have a voting score of 1. The best trade-off between landmark and non-landmark images according to the global Recall value (left diagram of figure 4) is achieved with $t = 2$. Figure 6 presents the amount of correct, incorrect and as non-landmark recognized landmark images for a dictionary of size 828K depending on the threshold t. The highest rate of correct recognition as well as incorrect recognition is achieved with $t = 1$. Despite the lower amount of correctly recognized landmark images at threshold $t = 2$, this result can be more satisfying for a user than the one at $t = 1$ because the amount of landmark images assigned to the wrong landmark is radically smaller, although simultaneously the amount of landmark images which are predicted as non-landmark is high, too. In a user's perception an incorrect classification (wrong landmark assignment) can be worse than the non-detection (non-landmark prediction) of landmark images. To answer the question on the influence of the branching factor K and the tree height, in figure 4 we can see that the recognition results depend on the real visual dictionary size than on the parameters branching factor and tree height on their own.

4 Summary and Future Work

We developed a CBIR system which bases on the Bag-of-Words approach combined with the HKM tree. Our experiments on different dictionary sizes show that a visual dictionary with a size of about 1M achieves the best recognition results with a Recall value of 0.6 on landmark images and 0.7 on all test images. This improves the baseline approach (dictionary size 8K) by almost 17%. For future work it is interesting to compare other well-known index structures like the kd-tree and the approximate K-Means to see which index structure is most appropriate for the problem of large-scale landmark recognition.

References

[AKTS10] Y. S. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *Proc. of the 18th Int. Conf. on Multimedia 2010, Firenze, Italy*, pages 153–162, 2010.

[CBHK09] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg. Mapping the world's photos. In *Proc. of the 18th Int. Conf. on World Wide Web, WWW 2009, Madrid, Spain*, pages 761–770, 2009.

[DBLL10] J. Deng, A. C. Berg, K. Li, and F. F. Li. What Does Classifying More Than 10, 000 Image Categories Tell Us? In *Computer Vision - ECCV 2010 - 11th European Conf. on Computer Vision, Heraklion, Crete, Greece, Proc., Part V*, pages 71–84, 2010.

[GBQG09] S. Gammeter, L. Bossard, T. Quack, and L. J. Van Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *IEEE 12th Int. Conf. on Computer Vision, ICCV 2009, Kyoto, Japan*, pages 614–621, 2009.

[Low04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[NS06] D. Nistér and H. Stewénius. Scalable Recognition with a Vocabulary Tree. In *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2006), New York, NY, USA*, pages 2161–2168, 2006.

[PJA10] L. Paulevé, H. Jégou, and L. Amsaleg. Locality Sensitive Hashing: A Comparison of Hash Function Types and Querying Mechanisms. *Pattern Recogn. Lett.*, 31(11):1348–1358, August 2010.

[PZ08] J. Philbin and A. Zisserman. Object Mining Using a Matching Graph on Very Large Image Collections. In *Sixth Indian Conf. on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India*, pages 738–745, 2008.

[RC14] M. Rischka and S. Conrad. Landmark Recognition: State-of-the-Art Methods in a Large-Scale Scenario. In *Proc. of the 16th LWA Workshops: KDML, IR and FGWM, Aachen, Germany.*, pages 10–17, 2014.

[ZZS$^+$09] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2009), Miami, Florida, USA*, pages 1085–1092, 2009.