Classification of Iconic Images

Mariia Zrianina, Stephan Kopf

Technical Report TR-2016-001
June 2, 2016

Technical Report TR-2016-001
University of Mannheim
– School of Business Informatics and Mathematics –
Department of Computer Science IV
Seminar Building A5, 6
68159 Mannheim, Germany

Classification of Iconic Images

Mariia Zrianina, Stephan Kopf
Department of Computer Science IV
University of Mannheim, Germany
zrianina@pi4.informatik.uni-mannheim.de, kopf@informatik.uni-mannheim.de

***Abstract*—Iconic images represent an abstract topic and use a presentation that is intuitively understood within a certain cultural context. For example, the abstract topic "global warming" may be represented by a polar bear standing alone on an ice floe. Such images are widely used in media and their automatic classification can help to identify high-level semantic concepts. This paper presents a system for the classification of iconic images. It uses a variation of the Bag of Visual Words approach with enhanced feature descriptors. Our novel color pyramids feature incorporates color information into the classification scheme. It improves the average F1 measure of the classification by** 0.117**. The performance of our system is further evaluated under a variety of parameters.**

Keywords: image classification, semantic image search, iconic images

I. Introduction

A large amount of free multimedia content is available on the Web today. This includes images and videos, but also associated textual descriptions or tags. When searching for multimedia content, image search engines like Google Images or Flickr find a large number of pictures. Most commercial search engines rely heavily on a textual description that surrounds the content, for example on a Web page or that was added manually. These search engines work well, if the topic to be searched for can be labeled with a brief and meaningful description. However, it is not always easy for users to find suitable keywords to be used in the search. This becomes even more challenging when searching for abstract topics like "climate change". Such a search request may be answered by a large variety of multimedia content. Images may show reasons for climate change, e.g., "air pollution" or possible solutions like "wind turbines". Figure 1 shows two example results of such a search query.

In this paper, our aim is to go one step further and search for *iconic images*. In an iconic image, the visualized objects are not relevant on their own, but the complete scene represents a larger, more abstract topic which is understood intuitively within a certain cultural context. An example would be the picture of a polar bear standing on an ice floe. In many western countries such an iconic picture represents global warming, and it has been used in this context for years. Both photographs in Figure 1 are typical examples of iconic images as well. Smokestacks can be considered iconic if they are associated with the topic of "air pollution", which in turn represents the larger theme "climate change".

To specify the term *iconic image*, we refer to the definition of Ponzetto et al. [1]: An iconic image concisely represents an entity that refers to a larger topic, and that is widely used in public communication. Such a topic is identified by media users easily and can trigger a substantial affective, cognitive and/or behavioral reaction.

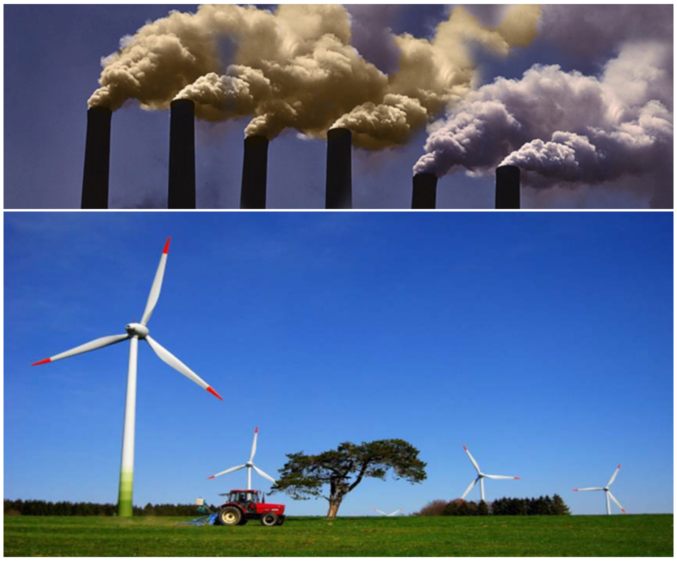


Fig. 1. Two iconic images of climate change. Smokestacks (causal attribution) and wind turbines (proposed solution) [1].

This paper presents a multimedia system that allows the classification of iconic images. The automatic classification of iconic images is a very challenging subject due to the fact that algorithms have great problems in 'understanding' high-level semantic concepts. Many applications benefit like media retargeting [2, 3], saliency detection [4], or video summarization [5, 6] if the semantic content of an image or a video is known. Iconic images are also an important topic for other sciences like literary studies [7, 8]. The goal of such studies is to find out how iconic images are used in different cultures, e.g., how a picture of a polar bear on an ice floe or a smokestack affects people in developing countries. It is also of interest, how the content is used over time and if there is a wear-out effect of certain iconic representations. The manual inspection of iconic images is very time consuming. Our system helps to pre-select candidates of iconic images of different topics and arranges them for manual inspection.

The system also helps to evaluate the quality of different classification algorithms, and supports users in their choice of parameters. The bag of visual words (BoVW) method is used as a starting point. To improve the basic algorithm and to make it more robust for iconic image search, color information in the images is considered additionally. Our novel color pyramid scheme enhances the basic BoVW method by adding feature points associated with a basic color.

The rest of this paper is structured as follows. Section II discusses various methods that extend the basic BoVW algorithm and other approaches of classifying iconic images. The used algorithm as well as the developed color pyramid feature are described in detail in Section III. Sections IV and V present our data set and the experimental results, respectively. Section VI summarizes the paper.

II. Related Work

Much work has been done in the past few years in the areas of image and object classification [9, 10, 11] as well as context based image retrieval [12, 13, 14]. One of the most successful techniques is the Bag of Visual Words (BoVW) approach which has been widely studied in the literature [15, 16].

Several improvements for the general BoVW approach have been proposed. Lazebnik et al. [17] proposed an improvement for the Bag of Visual Words method by using spatial information of visual words. Their approach is based on the pyramid matching scheme described in [18]. The difference to the standard Bag of Visual Words technique is the usage of visual word histograms. Usually, such vectors only represent the frequency of visual words for each image, and they do not include any spatial information. In the modified approach, location is considered as well.

Sato and Katto [19] suggested to modify the standard Bag of Visual Words algorithm applied in the area of object recognition by using Saliency Maps and Seam-Carving [20]. Their idea is based on the observation that many images from object recognition datasets have fairly large background areas. In their approach, such insignificant regions are ignored in order to improve the recognition rate.

Sharma [21] improved the method by Lazebnik et al. [17] by adding saliency information. The computed features are weighted with the corresponding saliency map. The proposed approach combines saliency modeling with the learning of a classifier: In one step, the saliency maps of the positive examples are fixed while the separating hyperplane is being optimized. In the next step, the generic saliency map is being tuned while keeping the hyperplane vector fixed.

The only work in the context of multimedia and computer vision that considers iconic images was proposed by Ponzetto et al. [1]. Their definition of the term “iconic image” is based on the work in [7, 8], as media icons with the focus on hot and sensitive topics. An iconic image refers to an abstract topic, which cannot be depicted directly, but is visualized via related concepts instead. An example is given by [22] with the abstract topic of *global warming* which is depicted by the concrete concept “polar bear on melting ice floe”. Ponzetto et al. propose an approach for the classification of iconic images by reducing the amount of human supervision. The approach consists of five steps. It starts with a human-selected basic set of iconic images along with their caption. Such pictures can be found using Google image search, restricted to Wikipedia and National Geographic Education. In a second step, the dataset is enlarged using a query-by-text approach from the images descriptions. The next two steps are devoted to detecting and filtering outliers from the automatically queried dataset. A picture is only preserved if either both caption and image have references to a person or neither has such references. The final step promotes diversity among images. When the dataset was built, it was checked manually to determine if each image is iconic of not.

We have developed a complete system that makes it possible to classify iconic images and to evaluate the performance of different classification algorithms [23]. Most of the previous work considers specific objects or scene categories. Only Ponzetto et al. [1] consider the problem of iconic image classification. We also introduce our novel color-based features called color pyramids that significantly improve the classification results on our iconic image dataset.

III. Classification System

Our system implements a variation of the Bag of Visual Words (BoVW) approach. BoVW is analogous to the Bag of Words method that is widely used for text classification [24]. The first step is building a vocabulary, which includes the detection of keypoints, computation of descriptors, and clustering. The next step is the creation of feature vectors and the training of a classifier on the labeled training data. The trained classifier can then be used to label new, unknown images. In the following subsections, the basic BoVW approach is discussed. Our proposed extension that uses color pyramids is discussed in Section III-C. Section III-D then gives some implementation details. The source code [1] of the classification system is available for download under the GNU General Public License.

A. Building the Vocabulary

The construction of a vocabulary of visual words starts with keypoint detection and the computation of descriptors in such obtained points. There are various well-established techniques to perform this task such as the SIFT detector/descriptor [25] and the SURF detector/descriptor [26]. The SIFT descriptor is based on gradients computed around the keypoint and consists of a vector with 128 elements. SURF computes the Haar-wavelet responses and stores a vector with 64 entries. A computationally efficient option for keypoint detection is to use the GRID detector: An image is divided into several cells, and the center of each cell is considered as a point of interest (see Figure 2). The keypoint size equals the size of a cell multiplied by a keypoint scale factor between 0.0 and 1.0.

To build the vocabulary, a set of training images is used. The training images are iconic images for which the larger topic they represent is known. Keypoints are then detected in all training images and corresponding descriptors are computed. The obtained descriptors are clustered using the k-means algorithm. A vocabulary is then defined by a set of computed centroids after the clustering phase. These cluster centroids are called *visual words*. The construction process of the vocabulary is visualized in Figure 3.

In previous works, the vocabulary size varies from hundreds [17] to hundreds of thousands [24] of elements. According to [24], if the vocabulary size is too small, the resulting

[1] Source code of the classification system:
http://ls.wim.uni-mannheim.de/de/pi4/research/projects/iconicimages/

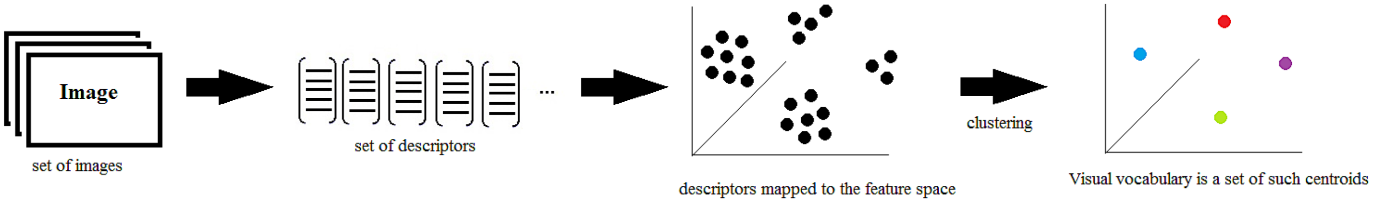


Image
set of images
set of descriptors
...
descriptors mapped to the feature space
clustering
Visual vocabulary is a set of such centroids

Fig. 3. Building a vocabulary. Keypoints and descriptors are computed over the training images. The descriptors are points in a high-dimensional feature space. By using k-means clustering, the descriptors are clustered into a smaller number of clusters whose centroids represent the visual words.

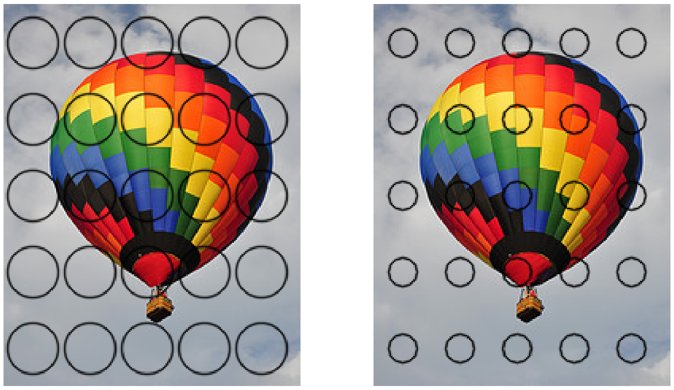


Fig. 2. An example of the GRID detector with a keypoint scale of 1.0 (left) and 0.5 (right).

vocabulary will not be representative of the training images, because different descriptors can be matched to the same visual word. However, if a vocabulary has too many entries, similar descriptors are mapped to different visual words, and the computational costs for clustering and using a classifier increase. Our system allows the user to manually define the vocabulary size. It may vary between 2 and 10000 cluster centroids. For keypoint detection, the SIFT, SURF, and GRID keypoint detectors can be used. They each can be combined with the SIFT or the SURF descriptor.

B. Classification

After the first phase of building the vocabulary is complete, a classifier is chosen and trained. For each image with index i from a training set, a histogram vector v_i of the visual words from the vocabulary is computed as:

$$v_i = (v_{1i}, v_{2i}, ..., v_{Mi}), \tag{1}$$

where M is the size of the used vocabulary, and v_{ji} is the number of times that the visual word j is contained in image i. The computed visual words histograms along with the ground truth image labels serve as input to the classifier. To predict a label for a new image, a visual word histogram for this image is calculated and fed to a trained classifier. We implemented the two classifiers Support Vector Machines (SVM) and Normal Bayes Classifier (NBC) which are described in the following.

Support Vector Machine

A Support Vector Machine (SVM) is a well-known technique for binary classification problems [27]. It learns an optimal separating hyperplane between two classes from training examples.

Let $\{V, Y\}$ be a training set, where $V = (v_1, ..., v_N)$ represents N training samples with each sample consisting of M features. $Y = (y_1, ..., y_N)$ denotes the labels of the samples such that $y_i = -1$ means that the sample belongs to the first class C_1 and $y_i = 1$ means that it belongs to the second class C_2. In our case, one training sample is a histogram of visual words that belongs to one of the training images. Initially, it is assumed that the two classes are linearly separable. This assumption is relaxed later. The goal is to build a linear function $f(v)$ such that

$$f(v_i) > 0 \; \forall \; v_i \in C_1, \quad \text{and} \tag{2}$$

$$f(v_i) < 0 \; \forall \; v_i \in C_2. \tag{3}$$

This is equivalent to finding a linear function such that

$$y_i f(v_i) > 0 \; \forall \; v_i \in V. \tag{4}$$

Multiplying the function f by some positive number yields

$$y_i f(v_i) > 1 \; \forall \; v_i \in V. \tag{5}$$

By making use of the fact that $f(x)$ is a linear function, Equation (5) can be rewritten as

$$y_i(w \cdot v_i + b) > 1 \; \forall \; v_i \in V, \tag{6}$$

where b is a number and w is a vector of coefficients. All hyperplanes that fulfill $w \cdot v + b = \pm 1$ are separating hyperplanes. The distance between the two boundary hyperplanes equals $\frac{2}{||w||}$.

Vectors v_i that belong to such boundary hyperplanes are called *support vectors*. To separate classes better, the distance between such two hyperplanes should be maximized, which means that $||w||$ should be minimized. This leads to an updated goal formulation: Find the minimum of a quadratic functional $0.5(w \cdot w)$. According to the Karush-Kuhn-Tucker conditions [28], this task is equivalent to finding a Lagrangian's saddle point:

$$L(w, b, \lambda) = 0.5(w \cdot w) - \sum_{i=1}^{N} \lambda_i (y_i(w \cdot x_i + b) - 1) \rightarrow \min_{w,b} \max_{\lambda} \tag{7}$$

subject to

$$\forall i = 1, ..., N: \; \lambda_i \geq 0, \; \text{and}$$
$$\lambda_i (y_i(w \cdot v_i + b) - 1) \geq 0$$

From Equation (7), it follows that either $y_i(w \cdot v_i + b) - 1 = 0$ or $\lambda_i = 0$. This can be used to rewrite the Lagrangian (7) into the following form

$$L(w, b, \lambda) \;=\; \sum_{i\,=\,1}^{N} \lambda_i \;-\; 0.5\|\sum_{i\,=\,1}^{N} \lambda_i y_i v_i\|^2 \tag{8}$$

and the goal is to find critical points of $L(w, b, \lambda)$. This problem can be solved by one of the many existing gradient-based optimization methods.

Defining S as the subset of the training data for which members have non-zero Lagrange multipliers λ_i, the optimal hyperplane function is given as

$$f(v) \;=\; \sum_{i\,\in\,S} \lambda_i y_i (v_i \cdot v) \;+\; b. \tag{9}$$

A more detailed description of Support Vector Machines can be found in [29]. If the dataset is not linearly separable, the SVM approach maps the input feature vector $v \;=\; (v_1, ..., v_M) \;\in R^M$ into a feature space with higher dimensionality by using the mapping function $\phi : R^M \rightarrow H$ and finds the optimal hyperplane in this new space where the classes are linearly separable again. The function $K(v, v') = (\phi(v) \cdot \phi(v'))$ is called a kernel function that computes the dot product in the higher dimensional space. The separating function then has the following form:

$$f(v) \;=\; \sum_{i\,\in\,S} \lambda_i y_i K(v_i, v) \;+\; b. \tag{10}$$

Normal Bayes Classifier

The Normal Bayes classifier is a conditional probability model, and it is a special case of the Naive Bayes model [30]. In Naive Bayes classification, an instance under consideration is represented by a feature vector $v \;=\; (v_1, ..., v_M)$ where v_j is a feature and M is the number of features. For each possible class C_k, the probability $p(C_k|v_1, .., v_M)$ of the features belonging to that class is computed. Then, the instance is assigned to the class with the highest probability. Using the Bayes theorem, this probability is calculated as

$$p(C_k|v) \;=\; \frac{p(C_k)p(v|C_k)}{p(v)}\;. \tag{11}$$

According to the Naive Bayes model, each feature is conditionally independent given the class. The numerator can thus be rewritten into:

$$p(C_k|v) \;=\; \frac{p(C_k) \prod_{j=1}^{M} p(v_j|C_k)}{p(v)}\;. \tag{12}$$

Here, the denominator is constant for each set of features and in practice, only the numerator is of interest. The class C_k with the biggest numerator thus gets assigned to the considered instance v.

The conditional probability of one feature v_j under the given class C_k can be calculated as

$$p(v_j|C_k) \;=\; \frac{\#(v_j, C_k) \;+\; 1}{\#(C_k) \;+\; M}, \tag{13}$$

where $\#(v_j, C_k)$ is the number of features of type v_j in the training set of the class C_k, and $\#(C_k)$ denotes the total number of features in the training set of the class C_k. In other words, $p(v_j|C_k)$ is calculated as the relative frequency of occurrence of visual word v_j in the training images belonging to class C_k. Laplace smoothing is used in the calculation to avoid probabilities of zero. An image is then assigned to the class

$$\begin{aligned} C^\star \quad &= \quad arg\ max_k\ log(p(C_k)) \;+ \\ &\sum_{j=1}^{M} log(p(v_j|C_k)). \end{aligned} \tag{14}$$

The Normal Bayes Classifier (NBC), in contrast to the Naive Bayes model, assumes that features are normally distributed. They are not necessarily independent as required by the Naive Bayes Classifier. The NBC algorithm computes the mean vectors for each class along with the co-variance matrix and then uses them to make the prediction [31]. The Bayes decision rule in case of two classes with normal distribution has the following form:

$$\begin{aligned} &\frac{1}{2}(v \;-\; M_1)^T {\Sigma_1}^{-1}(v \;-\; M_1) - \\ &\quad \frac{1}{2}(v \;-\; M_2)^T {\Sigma_2}^{-1}(v \;-\; M_2) \\ &\qquad + \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} \;\underset{C_2}{\overset{C_1}{\gtrless}}\; \ln\frac{p(C_1)}{p(C_2)}, \end{aligned} \tag{15}$$

where v is the observation vector, M_1 and M_2 are mean vectors, Σ_1 and Σ_2 are co-variance matrices.

Classification of an Image

The computed visual word histograms along with image topic labels serve as input to a classifier. To predict a label for a new image, a visual word histogram for this image is calculated and fed to an already trained classifier. In our system, both the NBC and the SVM classifier are implemented. In the case of multi-class classifications there are two main approaches: 'one against all' and 'one against one'. In the first case, a classifier is trained for each class to separate it from all other classes. Considering 'one against one', a classifier is trained for each combination of two classes. This increases the number of classification steps (computation time). Our system uses 'one against one' as default configuration because it shows better results.

C. Color Pyramids Feature

We propose *color pyramids* as a novel feature to enhance the basic BoVW method with color information. Feature descriptors like SIFT or SURF do not consider color information. The idea of color pyramids was motivated by the concept of spatial pyramids as presented by Lazebnik et al. [17]. Instead of dividing an image into spatial sub-regions, it is divided into color sub-regions. If coarse to fine color intervals are used, a hierarchy is created that is similar to spatial pyramids.

The first step to compute the color pyramids feature is to convert the input image into the HSV color space, and to discard everything but the hue channel. Then, keypoints and

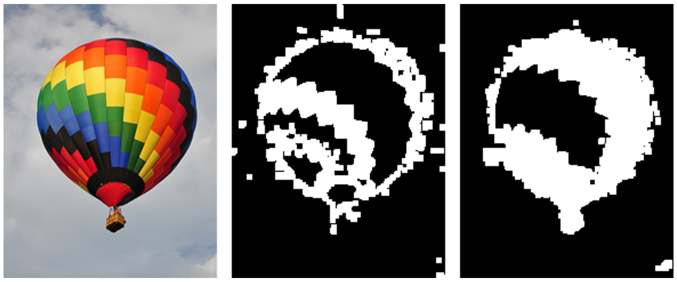


Fig. 4. An example of color masks. Original image (left), color mask with hue value 120 (center), and color mask with hue value 240 (right). Color masks are computed with a range of 20 and blurred with a Gaussian filter of size 5 × 5.

descriptors are calculated in the input image. L evenly spaced values c_k from the hue channel are selected and L color masks M_k are calculated that contain a range r of colors around each value. The color mask M_k is defined as

$$M_k(x,y) = \begin{cases} 255, & I(x,y) \in [c_k - r, c_k + r], \\ 0, & \text{otherwise} \end{cases} \qquad (16)$$

where I denotes the source image and (x, y) is the pixel coordinate. Optionally, color masks can be smoothed to reduce noise. Figure 4 shows an example of two computed color masks. White means that a pixel's color is within the color range of the mask, and black means that it is not.

Next, a complete histogram of visual word vectors v_0 is computed by using all keypoints along with their descriptors. For each created color mask $M_k, k \in 1, ..., L$, all keypoints that lie on black pixels in the mask are filtered out. By using only the remaining keypoints and their descriptors, a histogram of visual word vectors v_k is computed that is specific to the considered color mask M_k. All vectors v_k are then concatenated into one large visual word histogram vector $(v_0, v_1, ..., v_L)$ that represents an image. The structure of this vector now also captures the color information. The vector is then used as feature in the BoVW method. Figure 5 shows two histogram vectors of the same image. One contains all visual words and the other one contains only those in a specific color mask.

D. Implementation

We implemented a complete system for iconic image classification. This source code of the system[2] is available under the GNU public license and may be reused or extended. We used the C++ OpenCV library and the QT framework for our implementation. The SVM implementation is based on the LibSVM library. Our system supports a simple graphical user interface where the functionality is divided into six categories (see the tabs in Figure 6).

The first step is to build a vocabulary. The system supports SIFT, SURF, and GRID as keypoints detectors. The keypoint descriptors can be computed using either SIFT or SURF. Optionally, saliency maps can be used as discussed in Section II, and the vocabulary size can be set to values between 2 and 10000. After specifying the location of the input images, the system computes the vocabulary and stores it in the '.yml' format.

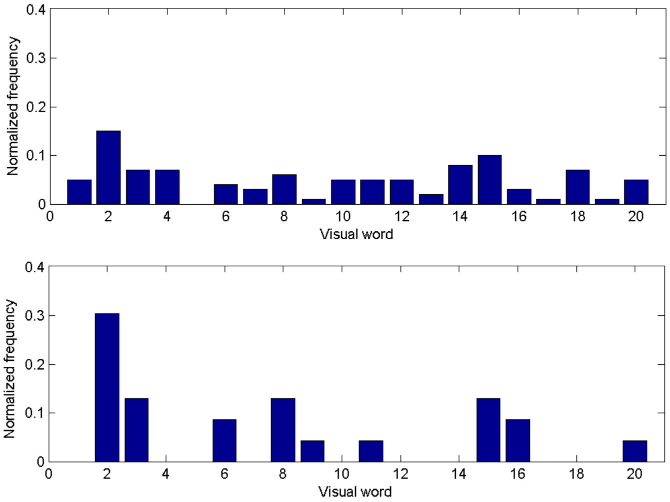


Fig. 5. Top: Histogram vectors without using a mask. Bottom: Histogram vectors when using the first mask with a hue value of 120. The horizontal axes label all 20 visual words from the dictionary used in the example. The vertical axes represent the normalized number of visual words found in the image.

The next step is the training of a classifier. SVM or NBC may be chosen as a classifier. Our SVM implementation supports four different kernels:

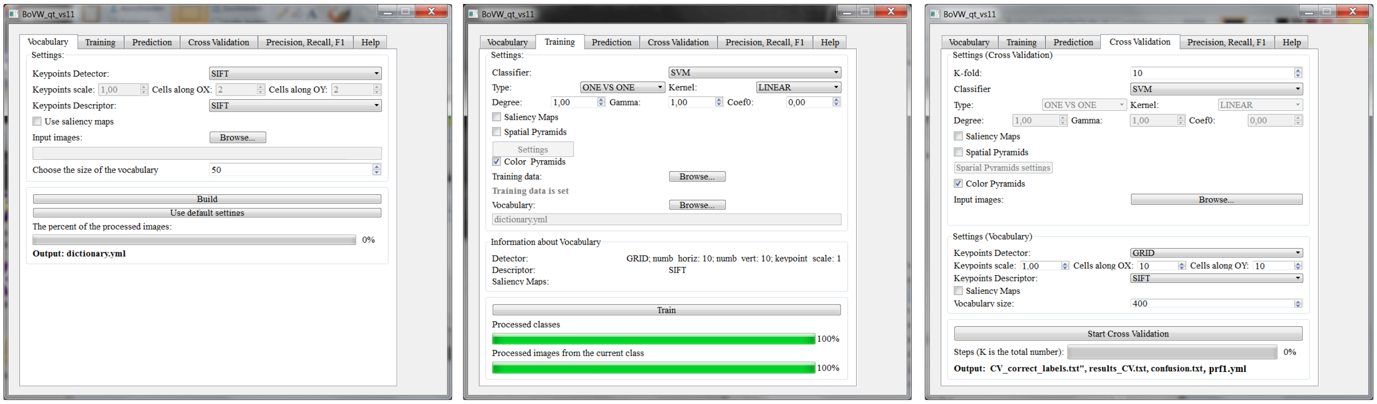
- Linear: $K(x,y) = (x,y)$,
- Polynomial: $K(x,y) = (\gamma(x,y) + coef0)^{degree}, \gamma > 0$,
- Radial Basis: $K(x,y) = e^{-\gamma||x-y||^2}, \gamma > 0$, and
- Sigmoid: $K(x,y) = \tanh(\gamma(x,y) + coef0)$.

Advanced options like saliency maps, spatial pyramids, and Color Pyramids can be selected. Additional settings for spatial pyramids such as grid type (horizontal or standard) and a grid level (between one and four) can also be specified.

The third tab is used to predict class labels for unknown input images. The vocabulary must be available and the training must be done before using the prediction. Again, advanced functions can be selected with corresponding check boxes. This functionality is relevant for users who do not wish to train a classifier on new labeled iconic images, but want to use an already trained classifier on a collection of unknown images. The prediction results are stored in a text file.

The last two tabs are used for cross validation and quality measurements. Cross validation allows the automatic separation of an annotated dataset into a number of equal blocks. Using several iterations, each block becomes the testing data without label information while all other blocks are used for training. A default value of 10 blocks is pre-defined. The predicted testing data is compared to the ground truth, and aggregated results (precision, recall, and F1) are stored in a text file. A confusion matrix as depicted in Figure 10 is also computed automatically.

[2] The system is available at: http://removed.due.to.anonymous.review.



Settings:

Keypoints Detector: SIFT

Keypoints scale: 1,00 Cells along OX: 2 Cells along OY: 2

Keypoints Descriptor: SIFT

☐ Use saliency maps

Input images:

Choose the size of the vocabulary 50

The percent of the processed images: 0%

Output: dictionary.yml

Settings:

Classifier: SVM

Type: ONE VS ONE Kernel: LINEAR

Degree: 1,00 Gamma: 1,00 Coef0: 0,00

☐ Saliency Maps

☐ Spatial Pyramids

☑ Color Pyramids

Training data:

Training data is set

Vocabulary:

dictionary.yml

Information about Vocabulary

Detector: GRID; numb horiz: 10; numb vert: 10; keypoint scale: 1

Descriptor: SIFT

Saliency Maps:

Processed classes 100%

Processed images from the current class 100%

Settings (Cross Validation)

K-fold: 10

Classifier SVM

Type: ONE VS ONE Kernel: LINEAR

Degree: 1,00 Gamma: 1,00 Coef0: 0,00

☐ Saliency Maps

☐ Spatial Pyramids

☑ Color Pyramids

Input images:

Settings (Vocabulary)

Keypoints Detector: GRID

Keypoints scale: 1.00 Cells along OX: 10 Cells along OY: 10

Keypoints Descriptor: SIFT

☐ Saliency Maps

Vocabulary size: 400

Steps (K is the total number): 0%

Output: CV_correct_labels.txt", results_CV.txt, confusion.txt, prf1.yml

Fig. 6. GUI of the iconic image classification system. Creating the vocabulary (left), training the classifier (middle), and validation (right).

Category	Class label	Global topic	Image size (avg.)
Mushrooms	9	Biodiversity	194 x 226
Reef	12	Biodiversity	185 x 232
Summer Forest	6	Biodiversity	178 x 224
Cattle	3	Agriculture	175 x 235
Wheat	15	Agriculture	174 x 229
Tractor	14	Agriculture	171 x 238
Air Balloons	2	Air	190 x 219
Clouds	4	Air	165 x 238
Plane	11	Air	170 x 238
Elephants	5	Africa Nature	176 x 234
Lions	8	Africa Nature	183 x 228
Giraffe	7	Africa Nature	198 x 215
Aurora	1	North Nature	161 x 236
Owls	10	North Nature	208 x 212
Seal	13	North Nature	170 x 236

TABLE I
OVERVIEW OF THE USED DATASET.

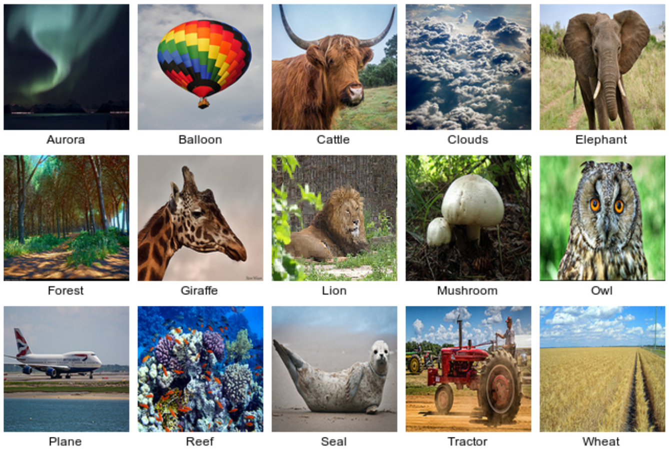


Fig. 7. Image examples from the used dataset.

IV. Dataset

The iconic image dataset was generated based on the pipeline described in [1]. The seed images along with their keywords for each topic were chosen based on Google image search as well as on Google image search restricted to the National Geographic and Wikipedia encyclopedias. Afterwards, based on the requests with the collected keywords which represents names for the used categories, dataset images were gathered from Flickr. Topics that have less than one hundred pictures were not selected. The remaining images were filtered manually based on their correspondence to their topic.

The idea behind iconic image classification is that each global topic includes several more narrow sub-categories. For example, the categories mushroom, reef, or summer forest refer to the larger topic of biodiversity. By classifying images from such sub-categories, it is possible to distinguish iconic images from the global topics. If a category that is assigned to an image is incorrect, even though the larger topic is correct, it is still considered as an error.

The created dataset consists of fifteen categories with 100 images in each. Each of the categories belongs to one of the five global topics. They are biodiversity, agriculture, air, africa nature and north nature. Table I presents the chosen global topics along with their sub-categories, class labels, and an average image size. There are 1500 images in total. Figure 7 shows one example image for each of the categories.

V. Experimental Results

This section presents selected results from the classification of iconic images. In the first part, we focus on our proposed technique that uses color pyramids and compare it to the basic BoVW approach. The color pyramids technique is orthogonal to other existing methods that improve the BoVW approach. It can be combined with spatial pyramids or saliency maps for example. The second part discusses parameter settings and findings when combining saliency maps, spatial or color pyramids. The computational effort of our system is evaluated at the end of this Section.

A. Results for color pyramids

To evaluate the improvement of classification accuracy when using color pyramids, the settings described in Table II were applied. Multiple SVMs were combined in a "one against one" approach to achieve multi-class classification. There is one SVM for each pair of classes, and a label for a new entity is assigned in a maximum voting process. To detect keypoints, the GRID detector was used. Each image was divided into 10×10 regions, and the keypoint size was set to be equal to the minimum side of a cell. To implement the color pyramids

Variable Name	Value
Vocabulary Settings	
Keypoints detector	GRID
Keypoints scale	1.0
Number of cells	10 x 10
Keypoints descriptor	SIFT
Vocabulary size	150
Training Settings	
Classifier	SVM
Type	One vs. One
Kernel	linear
Color pyramids	**{false, true}**
Cross-Validation Settings	
K-fold	10

TABLE II
COLOR PYRAMIDS TEST SETTINGS.

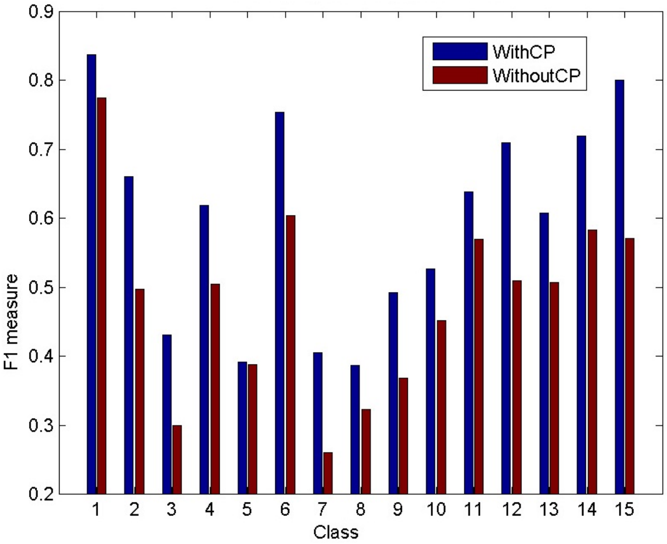


Fig. 8. Achieved F1 measure for each class with and without using color pyramids (CP).

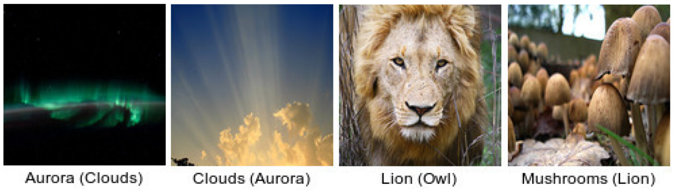


Fig. 9. An example of false negative classification. Under each image, its actual class label is written along with the assigned wrong class label in parentheses.

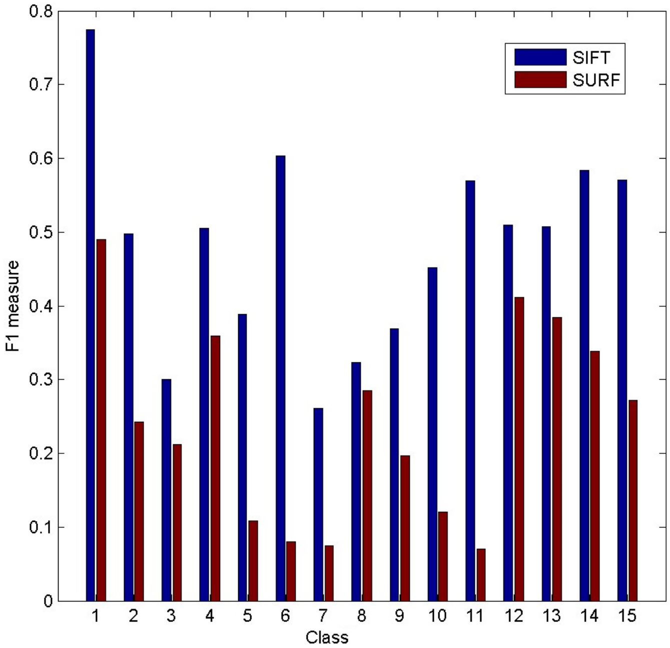


Fig. 11. F1 values achieved when using the SIFT and the SURF descriptors as features.

method, ten different color masks were computed with the hue values equally distributed between 0 and 180. The range was set to 20 to create slightly overlapping color masks. Each mask was then smoothed with a Gaussian filter of size 9.

Figure 8 shows the F1 measure for each category. Using color pyramids as features leads to better results. The biggest increase in F1 measure is 0.144 for class 7 (giraffe). There is no significant benefit (0.0039) in the case of class 5 which depicts gray or dark brown elephants. On average, the use of color pyramids improves the F1 measure by 0.1176.

When comparing the confusion matrices (see Figure 10), a decrease of type 1 and 2 errors can be seen when using color pyramids. For instance, class 1 (aurora) is partially misclassified as class 4 (clouds). This error drops significantly when colors are also considered. A possible reason is that the sky in the aurora images usually contains green colors that are unlike the blue sky in the cloud pictures. *Wheat* pictures where the major color is beige, are not misclassified as *cattle* (green grass), *forest* (green color), *mushroom* (beige, green, yellow, red colors) or *owl* (brown, white, green colors) as much anymore. The *balloon* label is less often assigned to images from the categories *cattle*, *elephant*, *giraffe* or *lion*. *Forest* pictures are less often confused with pictures of *lion*, *owl*, *elephant*, and *wheat*. Without considering colors, *mushroom* images are more often predicted as *reef* images. Figure 9 shows examples of false classifications.

B. Results for different configurations

We measured the F1 value for various parameter settings in our system. In our experiments, we found that GRID outperforms both SIFT and SURF as keypoint detector. For this reason, we always use GRID as keypoint detector in the following experiments.

To evaluate the SIFT and SURF descriptors, the settings presented in Table II were used, except always without considering color pyramids. Figure 11 shows that the SIFT approach reaches higher F1 measure values for each class. The largest difference in the results occurs in class 6 (forest) and class 11 (plane). The smallest difference is less than 4% for the category 8 (lion). This leads to the conclusion that the SIFT descriptor outperforms the SURF descriptor for the use in our iconic image dataset.

The next experiment evaluates difference choices for the vocabulary size. The same configuration as before is used again. Figure 12 presents the calculated F1 values when changing the vocabulary size from 50 to 500 in steps of 50. No consistent behavior between the classes could be observed. Some classes like 7 (giraffe) or 14 (tractor) clearly benefit from using a larger vocabulary. On the other hand, the F1 measure

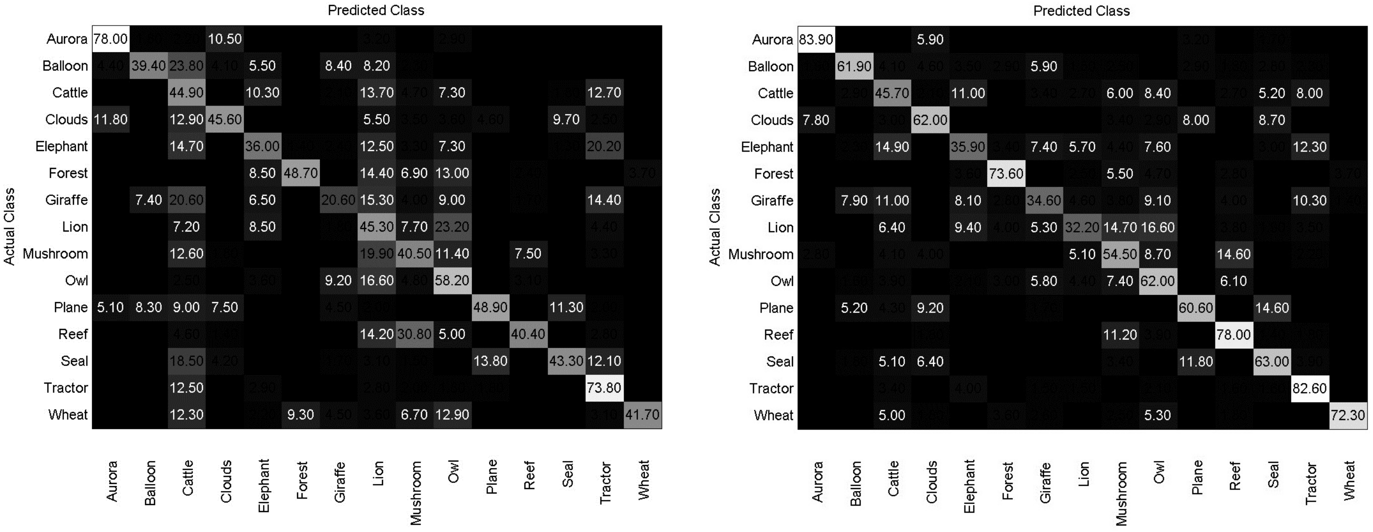


Fig. 10. Confusion matrices comparing the standard BoVW method with the SIFT descriptor and the advanced method using color pyramids.

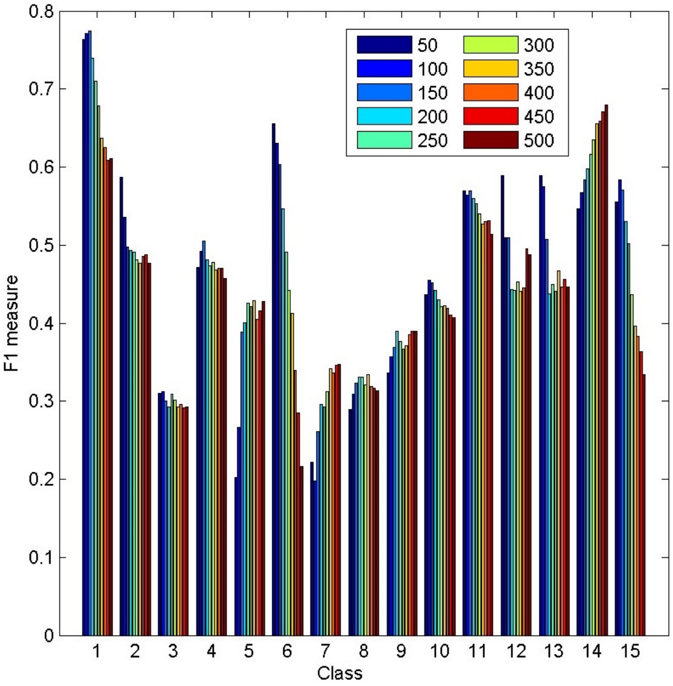


Fig. 12. F1 measure values when changing the vocabulary size. Each color corresponds to one vocabulary size.

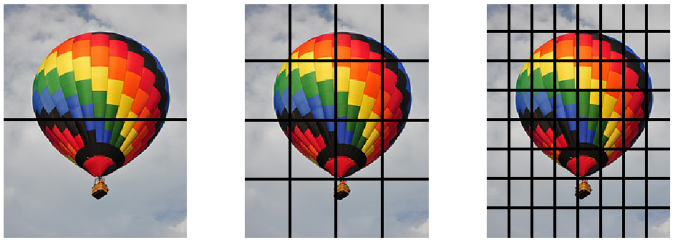


Fig. 13. Spatial pyramids with a horizontal grid (level 1) as well as regular grids of level 2 and 3.

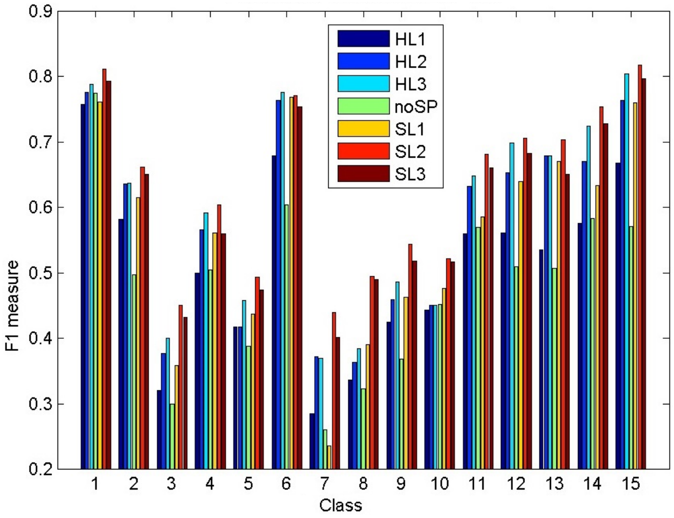


Fig. 14. Depicted F1 measure when using spatial pyramids. HLi specifies the horizontal grid of level i, while SL is the regular grid. The green bars show the results when no spatial pyramids are used.

drops significantly for classes 1 (aurora), 6 (forest), and 15 (wheat) when increasing the vocabulary size. A vocabulary size of 150 appears to be a good compromise for the iconic image dataset, so it is set as the default value.

The last experiment considers spatial pyramids of different levels. Figure 13 shows examples of a horizontal grid as well as two regular grids. Using spatial pyramids with any grid improves the F1 measure for all classes (see Figure 14). The regular grid yields much more reliable results compared to the horizontal grid. For instance, SL1 outperforms HL1 in 14/15 cases, for level two, SL2 is better in 15/15 cases, and for level three, it is better in 10/15 cases. If too many grids are used (in our case level 3), the regions become too small and the F1 measure value drops. Using a standard level two grid yields the highest F1 value for all classes except for class 6 (forest). Our obtained results show that spatial pyramids also improve the classification results of our iconic image dataset.

C. Computational Effort

This subsection compares the runtime of the standard and the advanced BoVW methods. For this experiment, the dataset

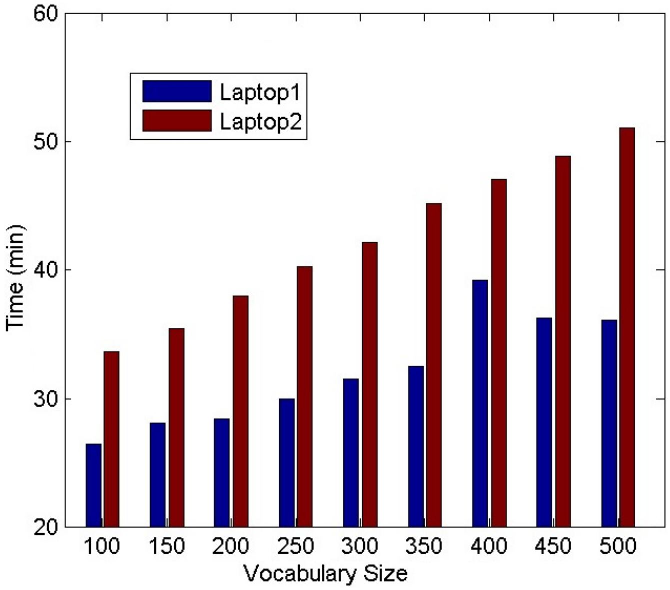


Fig. 15. Effect of the vocabulary size on the runtime.

was limited to 1000 pictures in total. All the other test settings are defined in Table II. We used two laptops for this evaluation[3] that are comparable to standard workplace computers. All time measurements were carried out five times on each machine.

We first compare the runtime of the different descriptors when training the vocabulary. When the SIFT descriptor is used, the runtime is approximately 28 minutes (35 minutes for laptop 2). In comparison, SURF requires only 6 minutes (8 minutes for laptop 2). This is an expected result since SURF descriptors are designed to be computationally efficient.

The effect of changing the vocabulary size is analyzed next. In general, the computation time increases with larger vocabulary sizes (see Figure 15). This is mainly caused by increasing the number of clusters in the k-means method. The outlier at 400 clusters on laptop 1 was probably caused by additional processes that were executed in the background.

Figure 16 compares the runtime of the advanced algorithms. As expected, the runtime of the spatial pyramids increases with higher levels. The difference in performance between using no spatial pyramids ("no" in the figure) and using a standard level three grid (SL3) reaches 70 minutes for laptop 1 and more than 90 minutes using laptop 2. The performance depends on the number of spatial regions where keypoints are considered. To apply the spatial pyramids feature with a certain level, all previous levels are used as well. The number of cells increases as 2^l in the number of levels l for the horizontal grid and 4^l for the standard grid. This explains why with each level, the computation time increases significantly. With the current setting, the runtime of the color pyramids is between that of the spatial pyramids with levels 2 and 3. The speed of the color pyramids depends on the number of color masks used. It decreases if less than 10 masks are used or if the overlap between the color masks is reduced.

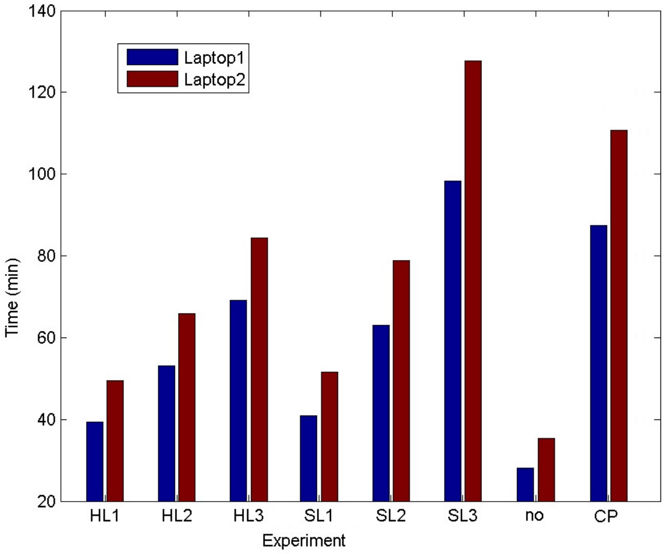


Fig. 16. Runtime of the advanced algorithms: spatial pyramids (HL and SL), no pyramids (no), and color pyramids (CP).

VI. Conclusion

We presented a system for iconic image classification. Our system may be used by scientists of literary studies for analyzing and understanding the use of iconic images in the Web. In addition, it makes it possible to easily evaluate different Bag of Visual Words algorithms and compare classification results. As a novel feature, we proposed color pyramids that enhance the standard Bag of Visual Words method with color information. They make it possible to distinguish between similar textures like grass or wheat by considering their colors. Using this feature increases the average F1 measure for all classes of iconic images by 0.117. We also analyzed the basic Bag of Visual Words method in detail and varied all parameters including the vocabulary size as well as several keypoint detectors and descriptors. Both the source code of the system is available for download.

The decision of whether an image is iconic or not is still mainly made by a human observer. Familiarity with the global topic and the image context play an important role here. As future work, we would like to develop algorithms that generally answer the question about iconicity in multimedia documents. To achieve this goal, a combined analysis of text and image search will be required.

VII. Acknowledgements

The sample images shown in this paper have been provided by Flickr users under creative commons licenses. Thanks to Billy Idle (Aurora), Len Radin (Balloon), Martin Liebermann (Cattle), Farrukh (Clouds), Brittany H. (Elephant), Moyan Brenn (Forest), Steve Wilson (Giraffe), Finn Frode (Lion), Ryan Steele (Mushroom), Edgar Barany C (Owl), Bill Damon (Plane), thinkpanama (Reef), Northwest Power and Conservation Council (Seal), Mobilus In Mobili (Tractor), and Rae Allen (Wheat) for providing the pictures.

[3] Intel Core i7-2670QM (2.20 GHz) and Intel Core i7-720QM (1.60 GHz), both with 4GB RAM and Windows 7 OS.

References

[1] S. P. Ponzetto, H. Wessler, L. Weiland, S. Kopf, W. Effelsberg, and H. Stuckenschmidt, "Automatic classification of iconic images based on a multimodal model," in *Bridging the Gap between Here and There - Combining Multimodal Analysis from International Perspectives : Interdisciplinary Conference*, Bremen, Germany, 2014.

[2] J. Kiess, B. Guthier, S. Kopf, and W. Effelsberg, "Seamcrop: Changing the size and aspect ratio of videos," in *Proc. of Workshop on Mobile Video (MoVid)*, 2012, pp. 13–18.

[3] S. Kopf, T. Haenselmann, J. Kiess, B. Guthier, and W. Effelsberg, "Algorithms for video retargeting," *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 819–861, 2011.

[4] T. Dittrich, S. Kopf, P. Schaber, B. Guthier, and W. Effelsberg, "Saliency detection for stereoscopic video," in *Proc. of ACM Multimedia Systems Conference (MMSYS)*, 2013, pp. 12–23.

[5] S. Kopf, T. Haenselmann, and W. Effelsberg, "Automatic generation of summaries for the web," in *SPIE 5307, Storage and Retrieval Methods and Applications for Multimedia*, 2004, pp. 417–428.

[6] S. Kopf, T. Haenselmann, D. Farin, and W. Effelsberg, "Automatic generation of video summaries for historical films," in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 3, June 2004, pp. 2067–2070.

[7] D. D. Perlmutter and G. L. Wagner, "The anatomy of a photojournalistic icon: Marginalization of dissent in the selection and framing of 'a death in genoa'," *Visual Communication*, vol. 3, no. 1, Feb. 2004.

[8] B. Drechsel, "The berlin wall from a visual perspective: comments on the construction of a political media icon," *Visual Communication*, vol. 9, no. 1, pp. 3–24, 2010.

[9] S. Kopf, T. Haenselmann, and W. Effelsberg, "Shape-based posture and gesture recognition in videos," in *SPIE 5682, Storage and Retrieval Methods and Applications for Multimedia*, 2005, pp. 114–124.

[10] S. Richter, G. Kuehne, and O. Schuster, "Contour-based classification of video objects," in *SPIE 4315, Storage and Retrieval for Media Databases*, 2001, pp. 608–618.

[11] S. Kopf, T. Haenselmann, and W. Effelsberg, "Robust character recognition in low-resolution images and videos," School of Business Informatics and Mathematics, University of Mannheim, Germany, Tech. Rep., April 2005.

[12] U. Altintakan and A. Yazici, "Towards effective image classification using class-specific codebooks and distinctive local features," *Multimedia, IEEE Transactions on*, vol. 17, no. 3, pp. 323–332, March 2015.

[13] S. Suchitra and S. Chitrakala, "A survey on scalable image indexing and searching," in *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on*, July 2013, pp. 1–5.

[14] G. Rafiee, S. Dlay, and W. Woo, "A review of content-based image retrieval," in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, July 2010, pp. 775–779.

[15] J. Mukherjee, J. Mukhopadhyay, and P. Mitra, "A survey on image retrieval performance of different bag of visual words indexing techniques," in *Students' Technology Symposium (TechSym), 2014 IEEE*, Feb 2014, pp. 99–104.

[16] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, "Generating descriptive visual words and visual phrases for large-scale image applications," *Image Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2664–2677, Sept 2011.

[17] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[18] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *IEEE International Conference on Computer Vision*, Beijing, China, October 2005.

[19] M. Sato and J. Katto, "Performance improvement of generic object recognition by using seam carving and saliency map," in *The International Workshop on Advanced Image Technology (IWAIT)*, 2010.

[20] J. Kiess, S. Kopf, B. Guthier, and W. Effelsberg, "Seam carving with improved edge preservation," in *SPIE 7542, Multimedia on Mobile Devices*, 2010, pp. 1–11.

[21] G. Sharma, "Discriminative spatial saliency for image classification," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 3506–3513.

[22] S. O'Neill and N. Smith, "Climate change and visual imagery," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 5, no. 1, pp. 73–87, 2014.

[23] S. Kopf, M. Zrianina, B. Guthier, L. Weiland, P. Schaber, S. Ponzetto, and W. Effelsberg, "Enhancing bag of visual words with color information for iconic image classification," in *Proceedings of 20th International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV)*. CSREA Press, July 2016.

[24] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 197–206.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision ECCV 2006*. Springer, 2006, pp. 404–417.

[27] A. Tzotsos, "A support vector machine approach for object based image analysis," in *Proceedings of the 1st International Conference on Object-based Image Analysis*, 2006, pp. 10–50.

[28] H. Kuhn and A. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, J. Neyman, Ed. University of California Press, Berkeley, California, 1951, pp. 481–492.

[29] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 1995.

[30] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1. Prague, 2004, pp. 1–2.

[31] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed., ser. Computer science and scientific computing. Boston: Academic Press, 1990.