Label Transfer by Measuring Compactness

Robert Varga, and Sergiu Nedevschi, *Member, IEEE,*

Abstract—This paper presents a new automatic image annotation algorithm. First, we introduce a new similarity measure between images: compactness. This uses low level visual descriptors for determining the similarity between two images. Compactness indicates how close test image features lie to training image feature cluster centers. The measure provides the core for a k-nearest neighbor type image annotation method. Afterwards, a formalism for defining different transfer techniques is devised and several label transfer techniques are provided. The method as whole is evaluated on four image annotation benchmarks. The results on these sets validate the accuracy of the approach, which outperforms many state-of-the-art annotation methods. The method presented here requires a simple training process, efficiently combines different feature types and performs better than complex learning algorithms, even in this incipient form. The main contributions of this work are: the usage of compactness as a similarity measure which enables efficient low level feature comparison and an annotation algorithm based on label transfer.

Index Terms—Information search and retrieval, scene analysis, object recognition, automatic image annotation

1 Introduction

Automatic annotation of images in its simplest form aims at labelling images with keywords from a dictionary. This procedure is necessary mainly to enable content based search and a better organization of images. If this can be realised automatically, human users are freed from the painstaking work of parsing thousands of images. This also ensures that the obtained labels are not influenced by the specific taste and inclination of the human annotators.

Today more and more images are stored in image archives and databases both locally on computers and on the Internet. As multimedia data stored grows in size it is becoming more difficult to maintain and organize them. Saving resources such as images in annotated form allows for efficient access and organization. This is the primary reason why automatic image annotation is deemed useful. Labeled images can be retrieved based on their text labels which reflect their content as opposed to providing visual content such as an example image for retrieval.

One of the main problems in this area of research is the difficulty of inferring high level textual labels from low level visual features. This is referred to in the literature as *semantic gap* to emphasize that a bridge must be built between the two. Even though low level visual features such as color, edges and texture can be extracted easily from images it is most difficult to organize and summarize them.

Another important issue in the domain concerns *weak labeling*. Weak labeling means that even though learning algorithms have access to annotated images two factors may prevent learning from these examples. The first one is regarding situations where even though a label is not present, the object or concept represented by it is contained in the image. Such images are considered as negative examples by learning algorithms which is undesirable. The second effect stems from the fact that labels are not linked to special regions of the image instead they are given for the image as whole. Thus it is impossible to determine exactly which bounded area it refers to.

The goal of this paper is to introduce a new approach to image annotation, but not a fully optimized method. In this regard its goal is similar to [1] i.e. to create a new approach on which more complex methods can be built. Currently, global feature vectors are employed for image comparison. This means that a lot of fine detail is lost when transforming the informative but noisy low level features to a global one which represent the image as a whole. Comparing low level features from two images directly is the other extreme where we encounter other problems: noise and high computation time. The proposed approach lies between these two extremes and enjoys the advantages from low level feature descriptiveness while maintaining a low run-time.

The contribution of this work lies mainly in the definition of compactness and using it as a similarity measure between images. Even though compactness is present in many places in the literature in the form of within-cluster sum of squares, in the form proposed here it is much more general. Firstly, because it is defined on two arbitrary sets: the data points and some other set considered to be centers. Secondly, the definition contains a general distance function. Thirdly, the norm-based definition has interesting theoretical properties. Fourthly, and perhaps most importantly, applying this measure in the current context is a totally new idea. We also provide a formalism for defining label transfer

- *R. Varga and S. Nedevschi are with the Computer Science Department, Technical University of Cluj-Napoca, Romania, E-mail: robert.varga@cs.utcluj.ro, sergiu.nedevschi@utcluj.ro*

techniques based on weight function. This formalism permits the mathematical description of several label transfer techniques.

2 Related Work

Research in the domains of object recognition and scene recognition has produced numerous methods for automatic annotation of images/videos. The purpose of this section is not to present the state-of-the-art but rather to put our proposed approach in context. Here, we will categorise these methods into two high-level classes. Even though algorithms from one of the class can differ radically, the underlying approach is the same.

2.1 Keyword-based annotation methods

The first category of methods create models for every label (or keyword, or concept) from the dictionary. We refer to a model as representation of a label. At annotation time the relevant labels are determined using these models and the extracted low level visual features from the test image. In most of the cases the inclusion of a label in the annotation is based on a binary decision. One of the disadvantages of these methods is that annotation with a keyword is based on a single decision and not based on multiple possibilities.

There are various types of models employed in the literature for representing a label: Gaussian Mixture Models[2] characterize a label as a multimodal Gaussian distribution defined on the feature space, Dirichlet Distribution[3] is a latent variable model that models the joint and conditional distribution of of the labels given the image, SVM classifier models[4] are supervised learning models that learn a separating plane between the data points of different classes in the feature space, Bayesian Hierarchical Models[5] are used to infer labels by employing a patch based representation of the input image using a Bayesian inference, Multiresolution Hidden Markov Models[6], [7] learn both spatial and multiresolution relationships between features, Markov Random Fields[8] increase annotation performance by learning spatial relationships between pixels.

Next we describe some methods from this category in more detail. In Supervised Multiclass Learning[2] each label is modelled as a Gaussian Mixture Model. The model corresponding to a specific semantic label is created by applying a hierarchical version of Expectation Maximization(EM) on the visual descriptors from each training image which share the keyword. By substituting descriptor values from a test image into each probability density function - the previously obtained GMMs - one can determine the conditional probability of each concept given the visual descriptors. Annotations are formed by taking the first 5 concepts with highest log-probability. Although this method constructs models in a very efficient manner, the training process is long mainly because of EM.

The works from [4], [14] rely on classifying global image features in the form bag-of-words features for scene recognition. These features constitute a global histogram for an image or a region. Each extracted local feature vector is associated to the closest element from a codebook of local features and the frequency of each center from the codebook constitutes the histogram (i.e. the histogram is the discrete distribution of the local features). This histogram construction method corresponds to average pooling, other alternatives are available such as max-pooling[11], geometric l_p-norm pooling[12], Geometric Consistency Pooling in Superpixels[13]. A classifier (e.g. SVM) is trained using these histograms and the available labels. However, to produce more than one output one must use multiple binary classifiers (one for each keyword to form a multiclass SVM). This makes it necessary to form a training set containing negative examples, images which are not labelled with the respective keyword. The assembling of such a set implies extra work and one can always provide new negative examples, which clearly is a drawback of binary classifiers.

2.2 Retrieval-based annotation methods

The second category of methods are based on the idea that similar images have the same labels. The key point in these methods is to define similarity measure between two images. At annotation time one can retrieve similar images from the database using the similarity measure and use the labels from these images to form the annotation. To emphasize the difference between this category and the former one, we mention that here models are practically constructed for each training image. These methods make use of *label transfer* since labels are passed on from similar images using different strategies to form annotations. Because the proposed approach falls into this category we will present similar approaches from the literature.

A recent publication[1] presents a baseline method for k-nearest neighbor image annotation. The images are represented by different types of global features: color histograms from various color spaces, Gabor and Haar wavelets for texture descriptors. The similarity between images will be the inverse of the distance between the global descriptors of the two images. Different feature types contribute equally in the calculation of the final distance value. Label transfer is then obtained in a greedy manner, giving importance to the first match.

The authors in [9] rely on a large image database of 80 million images from the Web to perform a k-nearest neighbor label transfer. The raw pixel values of 32x32 form the global feature vectors and an adaptive distance function is used as a similarity measure. The large number of learning examples compensate for the usage of only low level features. Another large dataset focusing on scene classification is presented in [10]. The publication evaluates state-of-the-art methods for large-scale scene recognition and makes use of multiple features.

One can take the matching technique one step further by allowing more general metrics such as the Mahalanobis distance. The parameters of the distance metric are obtained using Metric Learning techniques. The results using such method are described in [15], [16], which is one of the currently best performing methods on several benchmarks.

Other approaches for annotating images are considered in [17], where each object is described using attributes. The work [18] focuses on learning object attributes together with object categories. In [19] the authors represent the image as a high-level vector of object detector responses denoted as Object Bank.

3 Compactness based matching

In the following we define the notations used throughout the paper. Let $\mathcal{I} = \{I_1, I_2, ..., I_{Ni}\}$ denote the images constituting the training database, and $\mathcal{L} = \{l_1, l_2, ..., l_M\}$ the vocabulary containing the semantic labels (or keywords, words, concepts, tags). Ground-truth information is represented by associating to every image from $\mathcal{I}$ a set of labels from $\mathcal{L}$: $G = \{(I, L) | I \in \mathcal{I}, L \subset \mathcal{L}\}$. We denote with X the descriptors extracted from a test image, which is a set of descriptor vectors x_i, each having the dimension D. The descriptors from training image I_n are called $X^{(n)} = \{x_i^{(n)} | i = \overline{1, T_n}\}$, where T_n is the number of descriptors extracted from image n. The set of centers extracted from training image n is the set $C^{(n)} = \{c_i^{(n)} | i = \overline{1, K}\}$, the set of all centers is notated with $\mathbf{C} = \cup C^{(n)}$. Note that ultimately these are sets of points in a high dimensional space.

Current annotation methods that are based on image retrieval extract a global feature vector from each image and compare these vectors using a distance function. But is it possible to compare local features? Even though the global representation enables a fast comparison details of the particular image is lost. This is the main reason why it would be better if comparison of local features could be obtained efficiently. It is obvious that comparing each local feature from one image to each from another would be practically infeasible. This is why we propose to represent each training image with a set of relevant descriptor centers. These are obtained using the k-means clustering algorithm[20] applied on all the descriptor vectors extracted from that particular image. The k-means algorithm can be substituted with another method with better performance, but we use it for simplicity. We need to define then a distance - a matching score - between some new data points represented by test image features and a set of centers. This is where compactness comes in.

3.1 Compactness definition

Compactness is a measure that indicates how close data points are to a set of centers. The compactness between a set of points X and the centers C is given by:

$$\mathbf{c}(X, C) = \frac{1}{|X|} \sum_{i=1}^{|X|} \min_j d(x_i, c_j) \tag{1}$$

where $j \in \overline{1, K}$, $|X|$ denotes the cardinality of the set X and $d(x, y)$ is a metric defined on the D dimensional space. The previous definition states that compactness is the sum of the distances of each point from X to the closest point from C. Here X refers to any points in general, and in particular it can be the same as the set of features extracted from image n, $X^{(n)}$ in which case $|X| = T_n$. In our experiments we have found that the L^1 distance performs best in this context compared to the L^2 or the Chi-Square metric.

A less restrictive definition uses L^p norms instead of the distance function. This is useful in practice because it avoids extracting roots and has interesting properties.

$$\mathbf{c}(X, C) = \frac{1}{|X|} \sum_{i=1}^{|X|} \min_j ||x_i - c_j||_p^p \tag{2}$$

Note, when applying k-means on a set of points X the objective function to minimize is exactly the compactness of the centers and the point set X. So the following are equivalent to the L^2 norm compactness applied to the same points from which the clusters centers were obtained: within-cluster sum of squares; the minimum sum of squares; distortion function; potential function (the literature uses a multitude of terms referring to this value).

The compactness is always positive since it is a sum of distances or norms. Also, if we suppose that the points X are characterized by centers C then:

$$\mathbf{c}(X, C) < \mathbf{c}(X, C'), \forall C' \neq C \tag{3}$$

The last inequality(3) holds if k-means truly finds the set of centers that minimize the compactness. This can be ensured by running the algorithm multiple times with different initial center guesses and using different optimized initialization techniques in order to avoid local minimas[21]. More details about the k-means algorithm employed here along with specific parameters are described in section 6.1.

To use compactness as a similarity measure between two images one must first find their representation in some feature space denoted by X and Y respectively. Afterwards, one of the images - consider in this case the second image - is characterized by the cluster centers of the features. At this step we obtain the set C from Y. The similarity is then calculated as the compactness of the features from the first image to the cluster centers of the second image, more precisely $\mathbf{c}(X, C)$. If the asymmetry of this measure is an issue the compactness can be evaluated with the roles of the images swapped, however in practice the training images will be compactly represented by cluster centers and it is much more practical to reuse these.

3.2 Interpretation and motivation

We now investigate what this measure represents. Suppose we are given a set of points X and we want to find their compactness relative to some set of centers C. Consider the following partitioning of X around each $c_k \in C$ (Voronoi partitioning):

$$X_k = \{x \in X | k = argmin_j\{||x - c_j||\}\}, k = \overline{1, |C|} \quad (4)$$

This states that the sets X_k contain all the points that have center c_k as the closest center to them. Clearly the sets X_k are mutually disjoint sets and $\cup X_k = X$. Then the following identity is true for compactness that uses the L^p norm:

$$\begin{aligned} \mathbf{c}(X, C) &= \frac{1}{|X|} \sum_{k=1}^{|C|} |X_k| \mathbf{c}(X_k, c_k) \\ &= \frac{1}{|X|} \sum_{k=1}^{|C|} \sum_{x \in X_k} ||x - c_k||_p^p \quad (5) \end{aligned}$$

$$= \frac{1}{|X|} \sum_{k=1}^{|C|} (\sum_{x \in X_k} ||x - \overline{x_k}||_p^p + |X_k| \cdot ||\overline{x_k} - c_k||_p^p)$$

Where $\overline{x_k}$ are the centers of mass for the points X_k. The decomposition is true because the partitions contain only the closest elements to c_k. Tha last step follows from a well known lemma involing L^p norms:

$$\sum_{x \in X} ||x - c||_p^p = \sum_{x \in X} ||x - x_c||_p^p + |X| \cdot ||c - x_c||_p^p \quad (6)$$

We show this in the L^2 case in 2D, it can be easily extended for any p and any dimensions - we use $p = 1$ and $p = 2$ in this work. Let x_c denote the center of mass of the points x, so $x_c = \frac{1}{|X|} \sum_{x \in X} x$. Consider the translated coordinate system Ouv that has as the origin this center of mass. Also for convenience, rotate the axis such that the point c has the representation $(c_u, 0)$. In this coordinate system we can write for any point from X, $x_i = (u_i, v_i)$:

$$||x_i - c||^2 = (c_u + u_i)^2 + v_i^2 \quad (7)$$

Summing over all the points:

$$\begin{aligned} \sum_i ||x_i - c||^2 &= \sum_i {c_u}^2 + \sum_i (u_i^2 + v_i^2) + 2c_u \sum_i u_i \\ &= \sum_i ||x_i||^2 + |X| {c_u}^2 \quad (8) \end{aligned}$$

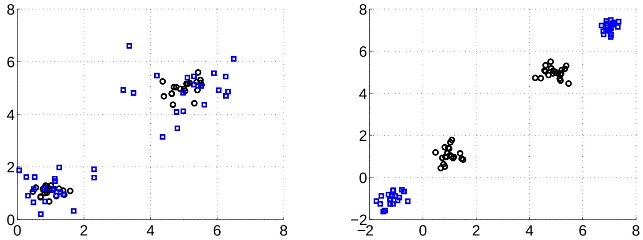
The last step uses the fact that the center of mass is now the origin. If the elements of X_k are considered to be i.i.d. random variables then $\overline{x_k} = E[X_k]$ and $\mathbf{c}(X_i, x_{ck}) = Var[X_k]$ if the compactness uses the Euclidian norm. The identity (5) shows some important characteristics of the compactness measure. It is composed out of two terms: the first term is the variance of the partitions and the second is the distance between the original centers and the centers of the partitions X_k. This means that the compactness will be minimal if the variance is low and the centers are close. Consequently, compactness indicates how tightly the points are situated around the centers. It may happen that $\mathbf{c}(Y, C) < \mathbf{c}(X, C)$ for some test data points Y and training data points X. This is the case when the test points Y have the same centers but a lower variance than X.

3.3 Comparison to two other similarity measures

We now compare compactness to two similarity measures and show its advantages over them. The two alternatives considered are: distances defined on bag-of-words type histogram descriptors, and the probability of data points fitting a Gaussian Mixture Model. Other similarity measures rely on distances defined on raw image pixel values or on histograms. Mutual Information[22], for example, is defined as the difference between the individual entropies and the joint entropy of the two images. Compactness assumes that one of the images is represented succinctly by a set of centers which reduces computation time.

If the bag of words approach is used then every image will be characterized by a histogram which reflects the distribution of the closest prototypes associated to each feature. The prototypes are k-means cluster centers and together they form the dictionary. The disadvantage in this case is that some relevant centers for the current image may not be present in the global dictionary. This prohibits the correct comparing of the images since important centers will be mapped to other centers from the dictionary. Even if all the relevant centers of the image are inside the dictionary, if two images have the same histograms we cannot determine how close or far they are even though there may be significant differences between them.

By studying Figure (1) we can analyze two cases where the histogram representation fails. The 2D training points which are partitioned in two are marked with black circles and were drawn from the same two normal distributions on both figures. The test points corresponding to two partitions are blue squares. In both cases the histogram for the test points will coincide with that of the training points. In the left graph (case a) both the test and the training points have the same centers, but the the test points have a larger variance. In the right graph (case b) the distributions have the same variance but the centers of the partitions are different. In all cases histograms will indicate that the test points are from the same distribution as the training points even though there are significant differences. This drawback is eliminated by the compactness which takes into account both the spread factor - variance - and the displacement between the centers.



(a) Same centers

(b) Displaced centers

Fig. 1: Two cases were the histogram representation fails

Associating every point to the closest center lies at the heart of the compactness. Consider now that the image is characterized by a Gaussian Mixture Model as in [2]. In this case the similarity between it and some feature set X is the probability that the data fits the model:

$$P(X|\pi_k, m_k, \Sigma_k) = \prod_{x \in X} \sum_{k=1}^{K} \pi_k G(x, m_k, \Sigma_k) \tag{9}$$

This compares every x with every center m_k and thus always penalizes good matches, since if a point is close to some m_k it will be far from all the other $K-1$ centers. Regardless of how well the data fits the distribution this penalty to the likelihood is always applied. Another issue with GMM obtained through EM is that on many occasions fixing the number of centers to a constant gives poor distributions since two or more centers are described by only one Gaussian with a covariance matrix of large values. A model with large covariance values will tag as similar a wide range of points which is clearly undesirable. This effect becomes visible if an image is retrieved as similar for many image queries.

4 Label transfer

Label transfer is achieved by constructing a histogram h. Each of its bins corresponds to a concept from $\mathcal{L}$, so $h \in \mathbb{R}^M$. We take into consideration the labels of the best N matches. After histogram construction, the labels with the highest corresponding bin value will be chosen to form the final annotation. Depending on how the histogram bins are incremented, different transfer schemes can be obtained.

The result of the matching procedure can be modelled as a function μ, which returns an ordered list of indexes, based on the compactness between the test image descriptors and each of the training image centers: $\mu(I) =< i_1, i_2, ..., i_N >$, where $\mathbf{c}(X, C^{(i_1)})$ is the minimal compactness, the one with i_2 is the second smallest and so on.

We define a weight function $\omega : \overline{1, N} \rightarrow \mathbb{R}$. Every label from the training image I_{i_n} increments the corresponding bin in the histogram by $\omega(n)$:

$$h = \sum_{n=1}^{N} \sum_{l \in L_n} \omega(n) \delta_l \tag{10}$$

L_n represents the labels from n-th match, more precisely from the training image I_{i_n} from the list $\mu(I)$. These labels are available from ground-truth information G. $\delta_l \in \mathbb{R}^M$ is vector a containing zeros on all positions except at the position uniquely associated to label l where it is one. By changing the expression of the weight function ω different types of transfer techniques can be achieved. In the following we present some particular cases.

4.1 Equal contribution transfer

In this case every match from the list $\mu(I)$ contributes evenly to the histogram. We have:

$$\omega_0(n) = 1, \forall n = \overline{1, N} \tag{11}$$

This is the most elementary type of transfer, it can be viewed as a majority voting scheme. It has the advantage that it eliminates those labels that only appear in a few matches. However, if the matching technique is good, we want to give more importance to the best matches.

4.2 Transfer based on rank

Rank based transfer entails weighing the best matches more and decreasing the weight exponentially based on the rank. In this case the weight function has the following form:

$$\omega_a(n) = 2^{a(1-\frac{k-1}{N-1})}, \forall n = \overline{1, N} \tag{12}$$

The parameter a can be tuned to obtain the best results. Of course the base of the exponent can be any number $b > 1$ or equivalently we can choose a to be $a' log_2 b$. One can see that $\omega_a(1) = 2^a$ and $\omega_a(N) = 1$. Note that weighing the matches equally (case w_0) is a special case of this function where $a = 0$. This is why, at parameter testing these two functions fall into the same category.

Note that this gives importance to matches according to their position regardless of their distance to the test instance. It may well be that all matches are very close, in this case the rank is not relevant.

4.3 JEC type transfer

The technique presented in [1] favours the best match and the rest of the labels are transferred based on their appearance frequencies in the training set. This case corresponds to the following form:

$$\omega_J(1) = 10, \omega_J(n) = 1, \forall n = \overline{2, N} \tag{13}$$

The histogram values will be updated on the last step in order to take into account the label frequencies. It is a greedy technique and thus depends on a good first match.

4.4 Transfer based on distance

To take into account the compactness values $\mathbf{c}_n$ of each of the matches, the following weight function is defined:

$$\omega_d(n) = 2^{b(1-\mathbf{c}_n/\mathbf{c}_1)}, \forall n = \overline{1,N} \tag{14}$$

This is particularly useful where compactness values are relevant, however the first match will always receive the same weight, i.e. because this weight function is relative to $\mathbf{c}_1$ it does not treat the case where even the best match is far away from the test instance.

4.5 Multiple features

Histogram construction using weight functions can be easily extended to the case where we intend to use multiple features. We begin by constructing the histogram normally for the first descriptor type. Then we save the histogram instead of resetting the bins to zero and repeat the process for the matches obtained from the other descriptor types. In this way every descriptor contributes to the final histogram which will provide the annotations.

In this paragragh we will refer to an instance of the algorithm that uses a specific kind of local feature simply as "a method" in order to simplify explanation. In this case the definition for the transfer histogram becomes:

$$h = \sum_m \eta_m \sum_n \sum_{l \in L_{m,n}} \omega(n)\delta_l \tag{15}$$

where the index m refers to the method number, η_m is the weight of the method m and $L_{m,n}$ is the set of labels from the n-th match using method m. In our experiments we have set $\eta_m = 1, \forall m$, i.e. we weigh each feature type equally. Note that the order of applying different methods is irrelevant.

4.6 Considering appearance frequency

After the histogram has been constructed using one of the weight functions described before, it can be updated by the frequency of each label from the training set. This is the number of times it appeared in the training set. For each non-zero position of the histogram (corresponding to labels that appeared at least once in the matches) we add a value proportional to the frequency (f_l) of the corresponding label:

In this case equation (15) is extended as:

$$h = \sum_m \eta_m \sum_n \sum_{l \in L_{m,n}} \omega(n)\delta_l + \varphi \sum_{l \in \cup L_{m,n}} f_l \delta_l \tag{16}$$

The parameter φ is set in such a way so that $\varphi \max f_l < \min \omega(n)$, i.e. frequency values are secondary to weights. This may seem to reduce the influence of this factor but the goal is to use this information only in uncertain cases, for example when we have two concepts with the same histogram value. JEC type transfer requires the histogram to be constructed using the previously defined equation.

Algorithm 1 Training

```
Ensure: centers from all training images.
for all training images $I_n$ do
    Extract local features $X^{(n)}$
    Apply k-means using $K$ centers to obtain $C^{(n)}$
    Save centers
end for
```

Algorithm 2 Testing

```
Require: Training image centers.
Ensure: Annotations for every test image.
for all test images $I$ do
    Extract local features $X$
    Sample $X$ to get $B$
    for all training image $I_n$ do
        find $\mathbf{c}(B, C^{(n)})$
    end for
    Obtain the first $N$ best matches using $\mu(I)$
    Transfer ground-truth labels from matches to obtain annotation using (10)
end for
```

5 Algorithm description

In this section we provide the high level steps required for the training process and for effective image annotation. Also the effect of different parameters on the execution time is discussed. The training involves the steps described by Algorithm 1.

We have fixed the number of clusters for the k-means algorithm to $K = 20$ for all our experiments based on some preliminary tests. If multiple features will be used for annotation it is necessary to run the training for each feature type. Note that in this way we form the building blocks for more complex methods that use different feature combinations and the training is done only once for each feature type.

In order to annotate an image the following operations from Algorithm 2 are to be executed. If multiple features are used then these operations are performed for each feature type and the equation (15) or (16) is used once at the end to form the transfer histogram.

6 Experimental Results

6.1 Implementation Details

All tests for the Corel5k dataset were run on a machine that has an Intel 2.66 GHz processor with two cores and 2GB RAM. For the larger datasets we performed the tests on the computing grid of the Technical University of Cluj-Napoca[23]. The number of parallel processes was set to 200 or 400. The application was implemented using C/C++. Libraries included were: OpenCV - vision library, vlfeat - for SIFT extraction. K-means implementation provided by OpenCV[24] was used running each time for a maximum of 200 iterations, with tolerance of 10^{-7}, five trials and kmeans++ center initialization by Arthur and Vassilvitskii[21]. The vlfeat library is utilized for dense SIFT extraction[25]. Multithreaded implementation was developed for matching and SIFT extraction in parallel from multiple image channels. The tests on the Corel5k involving high dimensional descriptors (SIFT and different combinations) were run on two threads in parallel.

6.2 Local descriptors

This section contains details about the local descriptors used for testing. A summary of the local descriptors is given in Table 1. Feature extraction strategy employed is dense sampling on a grid with displacements of 2 pixels. Each of the following paragraphs describes a different feature type.

The first feature type we present is a simple color descriptor that is obtained by first resizing the image to maximum a dimension of 64 pixels, while retaining the aspect ratio. This operation performs an averaging and additionally it reduces execution time. Afterwards, the feature vector at each pixel will contain the triplets from RGB, Lab and HSV color spaces. Feature dimension is 9. Despite its simplicity, this descriptor can produce surprisingly good results in this context. The importance and the efficiency of color descriptors is demonstrated by the fact that almost all annotation methods make use of this information. It is natural then to include a local descriptor based on color in our experiments.

A recently published texture descriptor called WLD[26] is also tested. Here we use a local histogram variant of the descriptor. We extract the excitation and gradient orientation values at every pixel and construct histograms of these on 8x8 blocks. These histograms will be the local features. We use the single resolution variant with 8 angle bins, 6 excitation bins and the parameter S (the number of excitation subbins) is set to 1. Final feature dimension is 48. This texture descriptor was evaluated on the Brodatz texture benchmark and has obtained superior results compared to SIFT, Gabor and several other texture descriptors (see [26] for results).

Histogram of Oriented Gradients(HOG) descriptors are extracted both from a grayscale image and from all three channels of the RGB image. The number of angle bins is set to 12. Feature dimension is 12, respectively 36 for color type. This descriptor was successfully applied for pedestrian detection [27] and other objects as well[28].

Discrete Cosine Transform coefficients have been utilized with great success in SML[2]. The coefficients are obtained on a 8x8 region using matrix multiplication and dimension reduction can be obtained easily be considering only the upper left corner of the resultant matrix[29]. We use the descriptors from the 3 channels of the Lab color space.

Scale Invariant Feature Transform(SIFT[30]) features are extracted on a dense grid from the image transformed into the Opponent Color Space. This sampling strategy has been proven to be the most effective in [31]. This descriptor has the largest total dimension from the ones used here. SIFT has properties such as scale and rotation invariance which are very useful for object recognition.

TABLE 1: Local descriptors employed

Descriptor	Type	Dimension
RGB	color	3
Lab	color	3
HSV	color	3
RGB+Lab+HSV	color	9
WLD	texture	48
HOG	texture	12
color HOG	texture	36
DCT	texture	63/192
dense SIFT	texture	128
dense SIFT-OCS	texture	384
Law	texture	10
Gabor	texture	12

Law texture descriptors[32] are obtained by filtering the image with the 16 Law convolution kernels. The result from the outer product of 4 one dimensional kernels and does not include the W kernel. No energy is calculated, but instead these raw values are used. The resulting descriptor is of size 10.

Another texture descriptor is obtained by filtering the image with different Gabor filters. The filters have 4 different orientations and 3 different scales, which is enough to represent 97% of the image energy [33]. The results of the filters form a the feature vector at each pixel. The applications of Gabor filters include mainly texture descriptors [34], [35].

Visual descriptor extraction using dense sampling can yield many feature vectors. If the image has height **h** and width **w** and a sampling strategy with displacement **d** is used then we have: $|X| = \left\lfloor \mathbf{h} \cdot \mathbf{w} / \mathbf{d}^2 \right\rfloor$, where $\lfloor x \rfloor$ indicates the floor function. Using every vector from the set for compactness calculation would be practically inefficient (high execution time). This is why a uniform sampling is applied on the set X and the compactness is calculated on the reduced set B. This is not the same as increasing the grid spacing since it may be that $|X| \neq k^2|B|$, for some natural number k. So let $B = \{b_i = x_{i\Delta} | i\Delta < |X|\}$, i.e. B contains every Δ-th sample from X. In this case $|B| = \lfloor |X|/\Delta \rfloor$ and we refer to the cardinality of the set as the bag size.

6.3 Time complexity analysis

The execution time for annotation is dominated by the matching process. The time complexity of matching using compactness is given by:

$$\mathcal{O}(T \cdot B \cdot K \cdot D)$$

where T is the number of training images; B is the 'bag' size - the number of features selected for compactness calculation; K is the number of clusters for k-means; D is the dimension of the feature vector. This can be optimized by halting distance calculation if the current distance exceeds the minimum distance obtained up to that point. The execution time grows linearly with the feature dimension, this is why it is recommended to apply dimension reduction techniques such as Principal Component Analysis (PCA) on large feature vectors such as SIFT. Matching can be parallelized easily at the highest level (at T) by dividing the training set into groups for each thread. We compare this to a Bag-of-words approach where the histogram construction and the matching costs:

$$\mathcal{O}(F \cdot D \cdot C + T \cdot C)$$

where F is the number of features extracted from an image, C is the number of keywords from the codebook. This shows that the first approach requires more time and is linearly proportional to the number of training instances. However, methods of later type usually require a much larger feature vector (large D). Even though the time complexity is high, relatively low execution time can still be achieved. This is demonstrated by providing the execution times of different cases in the experimental results section.

6.4 Evaluation protocol

The protocol for evaluation follows that which was already outlined in previous works (such as [2]) in order to enable comparison between methods. The different databases are split into two disjoint sets: training set - used for extracting k-means centers; test set - for the evaluation of the method. No information about the ground-truth labels of the test set are used when generating the automatic annotations. The automatically generated annotations are afterwards compared with the human given ones to obtain metric values.

We label each image with exactly five labels. For each keyword from the dictionary that appeared at least once in the test ground-truth we calculate the precision and recall values. For each label we define the following numbers:

- l_h - the number of times l appears in the test ground-truth;
- l_a - the number of times l was provided in an annotation by the automatic annotation method;
- l_c - the number of correct annotations with the label l.

In this setting the precision is $\mathbf{p} = l_c/l_a$ and the recall is $\mathbf{r} = l_c/l_h$. In order to obtain a global score we find the average precision and recall. These are obtained by averaging the precision and recall values of all the keywords which appear at least once in the test ground-truth. Another metric used is the number of non-zero-recalls. This is calculated as: $\mathbf{nzr} = \sum_{l_{ic}>0} 1$.

Additionally we introduce an indicator rarely used for evaluating annotation methods. The $F1$ score is the harmonic mean of the average precision and the average recall. It enables us to look at a single value for finding the best parameters and makes it easier to compare different annotation methods. By taking the harmonic mean, the score is closer to the lesser value, so a high $F1$ score can only be achieved with both a high precision and high recall.

6.5 Evaluation on Corel5K

We performed extensive testing on this database using different underlying features for the matching method. The structure of this database has been described in the previous works (e.g.[2]). We mention only that the training set has 4500 images and the test set consists of 500 images, the size of the dictionary is 374. The metric values for matching using only one type of feature, as well using multiple features are shown in Table 3 . The numbers next to the feature type indicate the dimension of the descriptor vector.

To clearly show the advantage of using compactness over histogram distances we provide test results on the Corel5k using the same features and transfer method w_J as in [1]. In addition we provide the best results obtained using one of the proposed transfer methods - w_a refers to the rank based exponential weighing with the subscript parameter a having the optimal value. Table (2) shows that in all cases compactness ensures a higher average precision and the same or higher recall. We have used L^1 metric for comparison and not Kullback-Leibler divergence for the Lab colorspace as in [1]. The proposed weighing further improves score values boosting both precision and recall.

To find the best parameters we have performed a grid search varying several parameters in the ranges given below. Test time can be saved because matches are obtained once for each bag size and afterwards different transfer techniques can be applied. The results with the highest $F1$ score are presented in Table (3). As mentioned before, we determined that L^1 distance behaves best in this context for compactness calculation. Parameter ranges used for testing are:

- bag size - $|B| \in \{50k | k \in \overline{1,10}\}$;
- neighbourhood size - $N \in \{5, 10, 15\}$;
- weight function type - w_a, w_J or w_d;
- weight function w_a parameter - $a \in \overline{0,5}$;
- weight function w_d parameter - $b = 300$;
- considering frequency or not - $\varphi \in \{0, 10^{-3}, 2 \cdot 10^{-3}\}$;
- number centers per image - $K = 20$.

The Table 3 contains metric values using different features on the Corel5k benchmark. Entries are ordered based on $F1$ measure that guided us in deciding which method is better. The last column shows the average execution time in seconds for a single image annotation using a single threaded execution on the machine described in section 7.1. ('-' signifies no data available). Execution time is given for the best parameter combination and it depends on bag size. Simple color descriptors

TABLE 2: Comparison using the same feature type

Feature	Precision	Recall	NZR	F1
JEC+RGB	20	23	110	21.39
JEC+Lab	20	25	118	22.22
JEC+HSV	18	21	110	19.38
Comp+RGB+w_J	21.98	24.38	121	23.12
Comp+Lab+w_J	21.29	24.80	123	22.91
Comp+HSV+w_J	19.33	26.94	128	22.51
Comp+RGB+w_5	21.58	26.85	123	23.93
Comp+Lab+w_5	22.34	25.62	123	23.87
Comp+HSV+w_3	21.95	26.68	124	24.09

behaved surprisingly well compared to different texture descriptors. Also the low dimensionality of this feature permits a very low execution time. Color variants of the texture descriptors perform better than gray-scale ones. The lower part of the table contains combinations of features. This confirms that the annotation method successfully combines multiple features and produces better results than using individual features.

We now compare our results with previous state of the art results in Table 4. Each percentage is taken from the indicated reference. Optimal configuration found by our tests is: using color descriptors along with DCT63 and SIFT, with the parameters set to: $K = 20, B = 200, \varphi = 2 \cdot 10^{-3}, N = 5, a = 3$ (last line from Table (3)). We note that Compactness based methods produce similar results to SML when using the same features (see DCT63 and DCT192 in Table 3) but using SIFT proves to be better. By efficiently utilizing multiple features our simple approach outperforms many methods from the literature based on the $F1$ score including MBRM, SML, JEC, ProbSim. MRFA does not provide exactly 5 labels at annotation which helps to achieve higher scores. The better results of TagProp can be explained by the fact that it employs Metric Learning which could also be used in our context to improve results.

6.6 Effect of parameters

Some general remarks can be made about the influence of different parameters on the metric values. We analyse results from Corel5k in detail. It is possible that the behaviour on other databases is different. The effect of each parameter is analysed by fixing the other ones to their optimal values. Multiple cases are considered where necessary.

Increasing the bag size B has moderate effect on score values. This can be studied using Figure (3). Score values increase and oscillate, and in some cases reach a maximum for fairly low values of B. Because bag size linearly influences the execution time lower bag size values such as 100 or 200 can be utilized. This is practical because it achieves faster execution while maintaining near-optimal performance. The oscillating behaviour is due to the errors introduced from sampling.

TABLE 3: Compactness based annotation results using different feature types on Corel5k

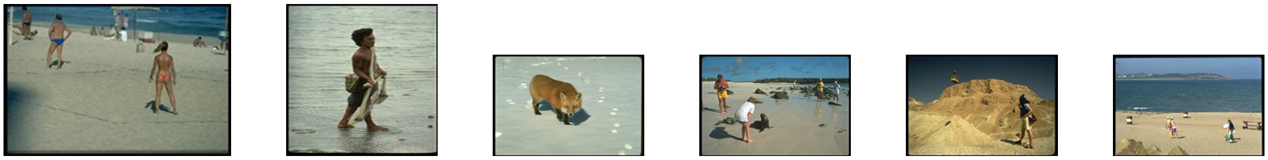
Feature	Precision	Recall	NZR	F1	exec
Gabor(12)	7.46	8.67	76	8.02	-
HOG(9)	11.22	11.57	85	11.40	-
Law(9)	13.82	17.56	105	15.47	-
color HOG(36)	14.36	17.57	109	15.80	-
WLD(48)	16.99	18.42	108	17.67	1.83
SIFT(128)	17.00	24.94	122	20.21	-
CSIFT(256)	19.49	24.51	120	21.72	-
color(9)	22.71	27.06	128	24.69	1.39
DCT(63)	22.32	28.27	129	24.95	0.48
DCT(192)	22.82	29.02	129	25.55	5.28
SIFT-OCS(384)	23.75	31.23	140	26.98	22.58
WLD + color(57)	26.45	27.88	120	27.15	4.4
SIFT + WLD + color(441)	30.19	31.99	131	31.06	18.23
SIFT + DCT63 + color(456)	30.15	32.17	133	31.13	20.62

TABLE 4: Comparison with state-of-the-art Corel5k

Method	Precision	Recall	NZR	F1
MBRM[36]	24	25	137	24.48
SML[2]	23	29	137	25.6
JEC[1]	27	32	139	29.2
ProbSim[37]	25.4	36.5	106	29.7
Compactness	30.15	32.17	133	31.13
MRFA-grid[36]	31	36	172	33.31
TagProp[15]	32.7	42.3	160	36.8

We now study the influence of neighbourhood size N. Better results were obtained using smaller N values. This may be due to the relatively small size of the database, so most of the images have few good matches among the training instances. We have found that $N = 5$ produces best results for individual features and on some occasions $N = 10$ for multiple feature case. Further fine-tuning could involve experiments considering $N \in \{1, 2, 3, 4\}$. This also demonstrates that the matching technique is efficient because the first few matches provide good labels to transfer.

Figure (4) contains metric values using different transfer techniques. We have associated $a = -2$ to JEC-type transfer and $a = -1$ to distance-based transfer. We have found that in almost all cases weighing based on rank performs best. In some cases we obtain better results with ω_d or ω_J, but the general recommendation is ω_a. The scores with $a = 0$ are almost always lower than the optimal scores obtained using $a = 3$ or $a = 4$. Overall tendency here suggests to accord significantly more importance to the best match. To provide a more encompassing overview Table (6) shows score values



(a) Test image (b) Match 1 (c) Match 2 (d) Match 3 (e) Match 4 (f) Match 5

Fig. 2: Sample matches and annotation from Corel5k - Predicted labels for test image a): **water, beach, tree, people, sand**. Showing only best five matches based on SIFT features. Note that incorrect labels from match 2 (confusion between sand and snow) get filtered out because of the transfer technique.

TABLE 5: Sample annotations using color+DCT63+SIFT from Corel5k

Prediction	people swimmers pool water athlete	stone pillar temple people sculpture	people outside museum dance tree	cars tracks formula wall straightaway	sky mountain tree snow sky
Ground-truth	people pool swimmers water	pillar temple sculpture stone	tree people dance outside	cars formula tracks wall	clouds mountain sky snow

using only RGB features, every row corresponds to a constant bag size and every column contains a different transfer technique.

If we consider frequency information it can increase overall performance ($F1$ score). However, this almost always entails an increase in precision and a decrease in recall and NZR values. The reason for this is that favouring the more frequent terms reduces the chance to annotate with rare labels. The influence of the frequency was tested using 3 different values: zero influence, minimal influence setting $\varphi = max f_l$ as suggested, and twice the previous value. The second case give higher $F1$ score in general.

Computation time varies in accordance with the time complexity formulas derived in section 5. It is linear with respect to feature dimension and also with respect to bag size. These two parameters can control the execution time, modifying the bag size has only minor negative effects on annotation performance. Even though the IAPR-TC12 and ESP-game datasets are much larger annotation time still remains fairly low due to the optimizations mentioned (halting calculation when distance exceeds the current N-th maximum).

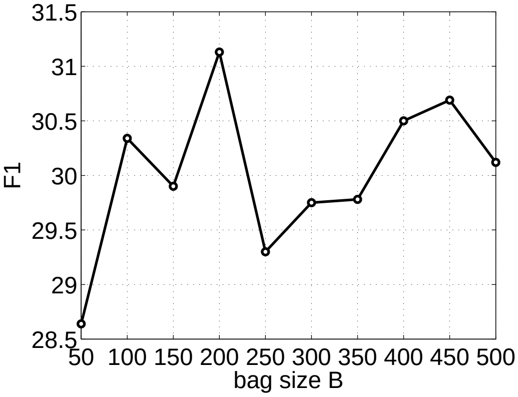
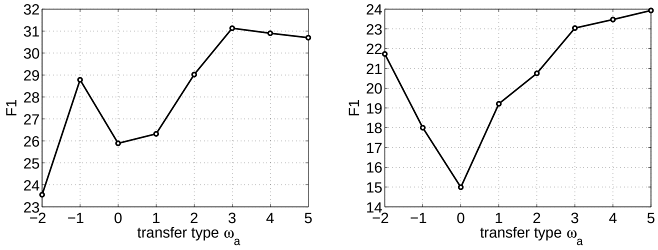


Fig. 3: The influence of bag size - color+DCT63+SIFT



(a) multiple features (b) RGB only

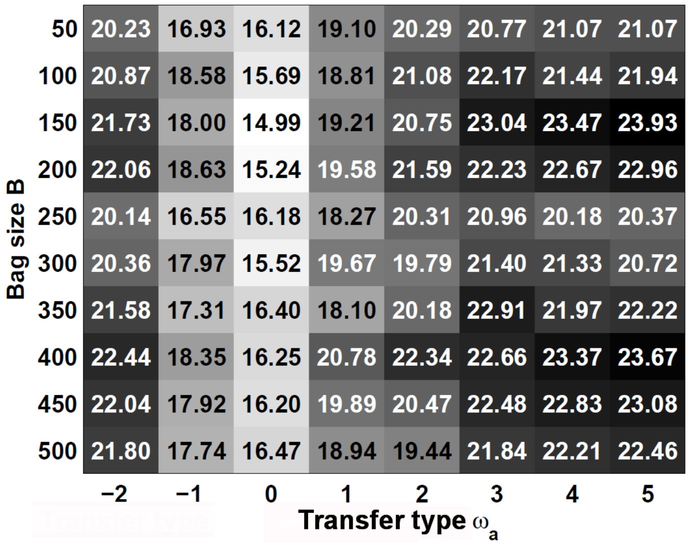
Fig. 4: The influence of transfer type

6.7 Evaluation on IAPR-TC12

This image collection consists of 20,000 still natural images taken from locations around the world and comprising an assorted cross-section of still natural images[38]. The same images are used from the IAPR-TC12 database as those in [1] in order to compare results in a correct manner. This database is larger, the training set numbers 17825 images and the test set contains 1980 images with 291 labels. The image annotations and test/training split is obtained from the files located at the web-page [1].

The metric values are calculated using all the labels from the ground-truth. This is the right way to obtain the number of correct labels however recall values will be lower. This is so because we only provide 5 labels,

[1] Makadia annotation files



Bag size B \ Transfer type ω_a	−2	−1	0	1	2	3	4	5
50	20.23	16.93	16.12	19.10	20.29	20.77	21.07	21.07
100	20.87	18.58	15.69	18.81	21.08	22.17	21.44	21.94
150	21.73	18.00	14.99	19.21	20.75	23.04	23.47	23.93
200	22.06	18.63	15.24	19.58	21.59	22.23	22.67	22.96
250	20.14	16.55	16.18	18.27	20.31	20.96	20.18	20.37
300	20.36	17.97	15.52	19.67	19.79	21.40	21.33	20.72
350	21.58	17.31	16.40	18.10	20.18	22.91	21.97	22.22
400	22.44	18.35	16.25	20.78	22.34	22.66	23.37	23.67
450	22.04	17.92	16.20	19.89	20.47	22.48	22.83	23.08
500	21.80	17.74	16.47	18.94	19.44	21.84	22.21	22.46

TABLE 6: The influence of parameters on F1 score

TABLE 7: Compactness based annotation results using different feature types on IAPR-TC12

Feature	Precision	Recall	NZR	F1	exec
color(9)	23.89	23.63	216	23.76	2.0
DCT(63)	25.24	24.64	225	24.94	6.3
SIFT(384)	31.82	32.45	245	32.13	17.0
SIFT+DCT+color	42.9	22.6	228	29.6	44.0

TABLE 8: Comparison with state-of-the-art IAPR-TC12

Method	Precision	Recall	NZR	F1
MBRM[39]	24	23	223	23.48
JEC[1]	28	29	250	28.49
Compactness	42.9	22.6	228	29.6
TagProp[15]	46.0	35.2	266	39.88

and in cases where in the ground-truth there are more than 5, we inevitably end up marking some labels as not recalled.

We provide some sample annotations for this dataset in Table 9. Notice that a lot of images have much more labels than 5. The results for this database (Table 7) again indicate that fairly good results can be obtained using simple color descriptors. However, SIFT features outperform other features mostly by reaching an $F1$ score of 32.13. It can be seen that the combination of different features is more successful on this database. Average precision value increases with 11%. Note that combining features results in lower recall and higher precision values. This is natural since more features provide more ”opinions” about the correct label and the consensus tends to reflect the truth.

The comparison made in Table 8 shows that Compactness obtains much better precision than MBRM and JEC (by 15%). Recall and NZR values are lower, but we mention here that using JEC-type transfer similar values were obtained as in [1].

TABLE 10: Compactness based annotation results using different feature types on ESP-game

Feature	Precision	Recall	NZR	F1	exec
Law-color(30)	16.36	16.07	217	16.22	2.32
WLD(48)	19.65	17.33	228	18.42	3.61
color(9)	19.73	19.28	230	19.50	1.56
DCT(63)	21.49	20.50	236	20.99	7.92
SIFT(384)	22.75	20.42	230	21.52	35.13
WLD+color	31.07	19.78	227	24.17	11.63
SIFT+DCT+color	34.67	21.29	233	26.38	39.45

TABLE 11: Comparison with state-of-the-art ESP-game

Method	Precision	Recall	NZR	F1
JEC[1]	22	25	224	23.4
Compactness	34.67	21.29	233	26.38
TagProp[15]	39.2	27.4	239	32.25

6.8 Evaluation on ESP game

This dataset is the result of an experiment involving collaborative human annotation. The subset of pictures used is the same as in[1]. More exactly: 19659 training images, 2185 test images, annotated with 269 different labels. An advantage of this set is that it is a result of an agreement between multiple annotators, so annotations are not biased by individual preference.

Table 10 contains results using a limited set of features and their combination. Five sample annotations are provided in Table 12. In this case WLD texture descriptor and color descriptors collaborate well. This may be so because in this set texture can discriminate instances better than in previous datasets. To enable comparison with the existing methods we summarize other results in Table 11.

6.9 Evaluation on NUS-WIDE

NUS-WIDE[40] is a large image dataset consisting of 269,648 images and associated tags from Flickr. This was created by a research team from the National University of Singapore, who also provide tags for 81 concepts. It is suitable for testing label transfer annotation algorithms. We have obtained this dataset by downloading the images using the provided URLs, however 36515 images are either missing or are blank, which can be detrimental for annotation precision.

We have carried out experiments using the proposed color descriptor and we have compared the obtained results with the NUS-wide Lab histogram based k-NN annotation baseline [40]. The only difference between the two methods is the distance calculation. In the first case we have used compactness and in the second case the L1 distance between global Lab histograms as in [40]. We could not directly use the feature vectors provided with the dataset because they are global feature vectors and compactness operates on local features, but the underlying feature type is the same. For every test image we

TABLE 9: Sample annotations using color+DCT63+SIFT from IAPR-TC12

Prediction	view river jungle middle cloud	pool people woman tree man	building front ornament trouser jacket	bike country helmet side short	sky mountain cloud desert bush
Ground-truth	cloud hill jungle middle palm range river view	chair man people pool woman	building column front jacket ornament person trouser	bike cap country cycling cyclist hand helmet jersey racing road short side	cloud desert mountain shrub sky

TABLE 12: Sample annotations using SIFT from ESP-game

Prediction	people sky crowd tree blue	man black dog grass tree	coin gold round circle money	sky blue people tower building	old man shirt glasses book
Ground-truth	crowd man people pole sky tree	black dog grass green guy man run shoes white	circle coin gold old round square	blue building people sky tower	book glasses green hand man old shirt

generate 5 labels. If the ground truth information specifies n labels we evaluate the performance on the first m labels, where m = min(5; n). We present the annotation performance in Figure 5. It is given in terms of precision for each concept and in terms of mean average precision (MAP). Concepts with more training examples - such as clouds, person, sky - have a significantly higher precision value for both methods. The k-NN based method has more concepts with non-zero precision and performs better for some concepts with more training examples. However, for most concepts compactness provides a higher precision. The MAP obtained with compactness is of 6.21 in comparison with 4.8 corresponding to the k-NN based classification algorithm.

7 Conclusions

In this paper we have presented a new technique for matching images. This can be employed in a nearest neighbour image annotation method. Several transfer techniques have been proposed and analysed.

In the experimental section we have provided metric values on four benchmarks to validate the presented method. This demonstrates that compactness outperforms the histogram distance. Furthermore the proposed transfer technique improves score values. The annotation method using multiple features does better than most the state-of-the-art algorithms. We stress that our goal was to show that compactness can be considered a useful alternative to image matching and not to provide a complete algorithm. This would entail careful feature selection, balancing the weights of each feature, changing the distance functions.

We enumerate the advantages of the presented approach:

- conceptually simple;
- simple and fast training process;
- flexible - can easily work with different underlying low level image descriptors;
- can efficiently combine different feature types (e.g. color and texture);
- does not need segmented images;
- does not need negative examples for training;
- robust - even with untuned parameters provides good results;
- competes with and outperforms complex learning algorithms.

As a drawback, we mention that the matching phase is more time consuming than some currently used methods (such as distance between global histograms from the bag-of-words approach) and is linearly dependent on the database size, like any k-NN matching annotation algorithm. However, we have shown by indicating execution times that even so, annotation time is well within acceptable ranges. This is why this approach can be utilized to provide good quality annotations in 5-10 seconds on the machine described in Section 7.1.

Contributions that result from the presented research:

- original idea to use compactness as a ”distance” measure between images, that enables us to effectively compare local descriptors;
- providing a formalism for defining label transfer

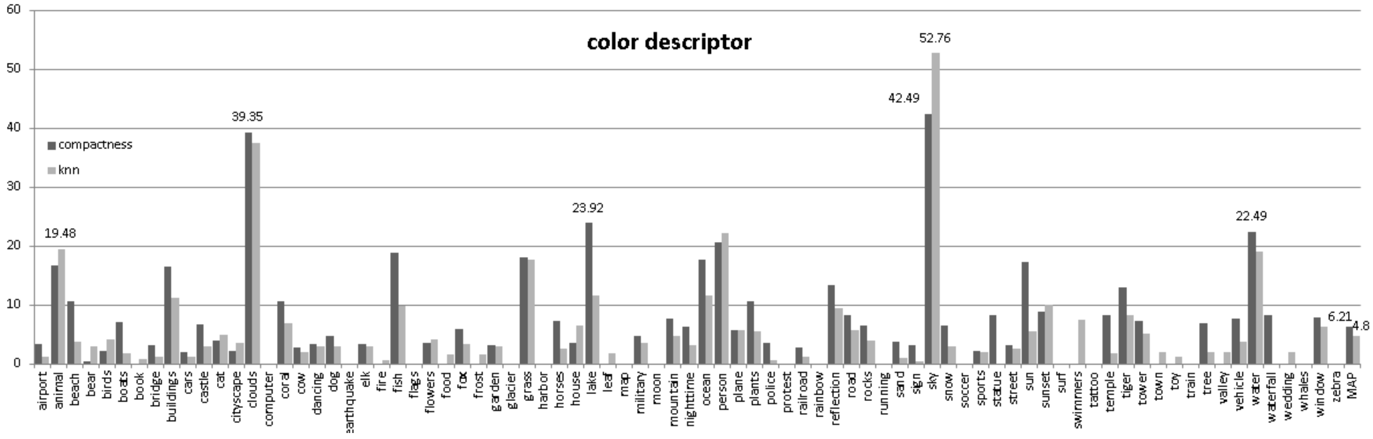


Fig. 5: Precision values for each concept and MAP on the NUS-WIDE dataset

techniques;

- devising and testing of elementary transfer types;
- validation and result analysis on 4 different datasets that proves the efficiency of the method.

Future work will involve experimenting with different feature types and their various combinations in order to obtain optimal results. Different implementation ideas for matching execution time reduction are under consideration. New weight function types for transfer are also under research. Another variant of the algorithm would be to use GMMs to represent images instead of k-means centers. It would also be desirable to find distance functions that weigh important features more.

Acknowledgements

This work was supported by the Romanian Ministry of Education and Research, through the project ”SmartCo-Drive”, code PN II PCCA 2011 3.2-0742, from 03.07.2012 (2012-2015). The authors would like to thank Kobus Barnard for providing the Corel5K data set. We thank A. Makadia for making public the annotations and training/testing splits employed in his work for the IAPR-TC and ESP datasets.

References

[1] A. Makadia, V. Pavlovic, and S. Kumar, “A new baseline for image annotation,” in *ECCV*, 2008, pp. III: 316–329.

[2] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, Mar. 2007.

[3] Blei, D. M., Jordan, and M. I., “Modeling annotated data,” in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. Multimedia information retrieval, 2003, pp. 127–134.

[4] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR*, 2006, pp. II: 2169–2178.

[5] F. F. Li and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” in *CVPR*, 2005, pp. II: 524–531.

[6] J. Li and J. Z. Wang, “Real-time computerized annotation of pictures,” *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985–1002, Jun. 2008.

[7] J. Li and J. Z. Wang, “Automatic linguistic indexing of pictures by a statistical modeling approach,” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 25, no. 9, pp. 1075–1088, 2003.

[8] A. Llorente, R. Manmatha, and S. M. Rüger, “Image retrieval using markov random fields and global image features,” in *CIVR*, S. Li, X. Gao, and N. Sebe, Eds. ACM, 2010, pp. 243–250.

[9] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[10] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “SUN database: Large-scale scene recognition from abbey to zoo,” in *CVPR*. IEEE, 2010, pp. 3485–3492.

[11] T. Serre, L. Wolf, and T. Poggio, “Object recognition with features inspired by visual cortex,” in *CVPR*, 2005, pp. II: 994–1000.

[12] J. Feng, B. Ni, Q. Tian, and S. Yan, “Geometric ℓp-norm feature pooling for image classification,” in *CVPR*. IEEE, 2011, pp. 2697–2704.

[13] L. Cao, R. Ji, Y. Gao, Y. Yang, and Q. Tian, “Weakly supervised sparse coding with geometric consistency pooling,” in *CVPR*. IEEE, 2012, pp. 3578–3585.

[14] H. Dunlop, “Scene classification of images and video via semantic segmentation,” in *CVPR Workshop on Perceptual Organization in Computer Vision*, 2010.

[15] M. Guillaumin, T. Mensink, J. J. Verbeek, and C. Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in *ICCV*. IEEE, 2009, pp. 309–316.

[16] J. Verbeek, M. Guillaumin, T. Mensink, and C. Schmid, “Image annotation with tagprop on the MIRFLICKR set,” in *11th ACM International Conference on Multimedia Information Retrieval*, 2010.

[17] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, “Describing objects by their attributes,” in *CVPR*, 2009, pp. 1778–1785.

[18] S. J. Hwang, F. Sha, and K. Grauman, “Sharing features between objects and their attributes,” in *CVPR*. IEEE, 2011, pp. 1761–1768.

[19] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li, “Object bank: A high-level image representation for scene classification semantic feature sparsification,” in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*. Curran Associates, Inc, 2010, pp. 1378–1386.

[20] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, pp. 128–137, 1982.

[21] Arthur and Vassilvitskii, “k-means++: The advantages of careful seeding,” in *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 2007.

[22] P. A. Viola and W. M. Wells, “Alignment by maximization of mutual information,” *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, Sep. 1997.

[23] E. Cebuc, A. Suciu, K. Marton, S. Dolha, and L. Muresan, “Implementation of cryptographic algorithms on a grid infrastructure,” in *aqtr, vol. 2, pp.1-6, 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2010.

[24] G. R. Bradski and A. Kaehler, *Learning OpenCV - computer vision with the OpenCV library: software that sees*. O'Reilly, 2008.

[25] A. Vedaldi and B. Fulkerson, "Vlfeat – an open and portable library of computer vision algorithms," in *Proceedings of the 18th annual ACM international conference on Multimedia*, 2010.

[26] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "Wld: A robust local image descriptor," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2010.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. I: 886–893.

[28] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[29] A. Watson, "Image compression using the discrete cosine transform," *Mathematica Journal*, vol. 4, no. 1, pp. 81–88, Winter 1994.

[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[31] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 32, no. 9, pp. 1582–1596, 2010.

[32] K. I. Laws, *Textured image segmentation (PhD dissertation)*. Dept. of Engineering, Univ. Southern California, 1980, lAWS80.

[33] R. Sandler and M. Lindenbaum, "Gabor filter analysis for texture segmentation," in *Workshop on Perceptual Organization in Computer Vision*, 2006, p. 178.

[34] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[35] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on gabor filters," *IEEE Trans. Image Processing*, vol. 11, no. 10, pp. 1160–1167, Oct. 2002.

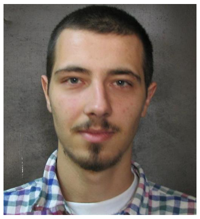
[36] Y. Xiang, X. D. Zhou, T. S. Chua, and C. W. Ngo, "A revisit of generative model for automatic image annotation using markov random fields," in *CVPR*, 2009, pp. 1153–1160.

[37] A. R. Chunsheng Fang, "Probsim-annotation: A novel image annotation algorithm using a probability-based similarity measure," in *20th Midwest Artificial Intelligence and Cognitive Science Conference, Fort Wayne, Indiana*, 2009.

[38] M. Grubinger, P. Clough, H. Muller, and T. Deselaers, "The iapr benchmark: A new evaluation resource for visual information systems," in *International Conference on Language Resources and Evaluation*, Genoa, Italy, May 2006.

[39] S. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *CVPR (2)*, 2004, pp. 1002–1009.

[40] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: a real-world web image database from national university of singapore," in *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR 2009, Santorini Island, Greece, July 8-10, 2009*. ACM, 2009.



Robert Varga received his M.S. degree in Computer Science and B.S. degree in Automation from the Technical University of Cluj-Napoca, Romania in 2012 and 2010, respectively. He is currently a PhD. student at the same university. Main research interests include image processing, pattern recognition, automatic image annotation, object detection, pedestrian detection.

Sergiu Nedevschi (M'99) received the M.S. and PhD degrees in Electrical Engineering from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 1975 and 1993, respectively. From 1976 to 1983, he was with the Research Institute for Computer Technologies, Cluj-Napoca, as researcher. In 1998, he was appointed Professor in computer science and founded the Image Processing and Pattern Recognition Research Laboratory at the TUCN. He is currently vice-rector at TUCN. He has published more than 200 scientific papers and has edited over ten volumes, including books and conference proceedings. His research interests include Image Processing, Pattern Recognition, Computer Vision, Intelligent Vehicles, Signal Processing, and Computer Architecture.