MFS-Map: efficient context and content combination to annotate images

Alceu Ferraz Costa, Agma Juci Machado Traina, Caetano Traina Jr.
Department of Computer Science
University of São Paulo
{alceufc, agma, caetano}@icmc.usp.br

ABSTRACT

Automatic image annotation provides textual description to images based on content and context information. Since images may present large variability, image annotation methods often employ multiple extractors to represent visual content considering local and global features under different visual aspects. As result, an important aspect of image annotation is the combination of context and content representations. This paper proposes MFS-Map (Multi-Feature Space Map), a novel image annotation method that manages the problem of combining multiple content and context representations when annotating images. The advantage of MFS-Map is that it does not represent visual and textual features by a single large feature vector. Rather, MFS-Map divides the problem into feature subspaces. This approach allows MFS-Map to improve its accuracy by identifying the features relevant for each annotation. We evaluated MFS-Map using two publicly available datasets: MIR Flickr and Image CLEF 2011. MFS-Map obtained both superior precision and faster speed when compared to other widely employed annotation methods.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Image annotation, Image retrieval

1. INTRODUCTION

Research on image retrieval traditionally considers two types of queries: by example, considering the content of the image, and by keywords, considering the context of the image. When querying by example the user provides a sample image and the retrieval system returns a set of similar images regarding a given criterion. However it is often impractical for the user to provide the sample image to express the query. In addition, visual similarity is defined by color, texture and shapes and these low level features present a gap regarding the query semantics.

(a) Annotation "sky".

(b) Annotation "structure".

Figure 1: Pictures from the MIR Flickr dataset with annotations (a) "sky" and (b) "structure". Color information may be correlated to annotation "sky" but will probably be less meaningful to annotation "structure".

In keyword-based queries, images are retrieved employing their textual annotations, as is done with text retrieval. However, images should contain textual annotations, and manual annotation is a very tiresome task that can be impractical for huge numbers of images. In addition, human annotations are subjective and can be ambiguous. Therefore, there is a considerable interest in automatic image annotation (AIA), which annotates images based on their visual content.

Most of the current automatic image annotation methods [16, 1, 13, 15] employ multiple extraction algorithms to analyze images considering local or global features under different visual features. This is necessary because none of existing extraction algorithms is capable of describing the large visual variability of images. However, the use of a large

SAC'14 March 24-28, 2014, Gyeongju, Korea.

number of features results in a problem known as the *dimensionality curse* [8], where the significance and information content of each feature decreases, reducing the annotation accuracy.

Nevertheless, considering images with a given textual annotation, the visual variability tends to be lower. For example, in Figure 1 we have pictures taken from the MIR Flickr dataset [7]. In the MIR Flickr dataset each picture contains manual annotations. In Figure 1(a) and Figure 1(b) we have images with the annotations "sky" and "structures" respectively. Color information, for example, may be appropriate to identify the presence of annotation "sky" because of the prevalence of blue color. For annotation "structures", however, there is no prevalent color. Thus, color may not be an adequate feature.

Therefore, if we knew beforehand which visual features are useful to determine the relevance of a given annotation, we would be able to discard irrelevant features, improving annotation accuracy. However, it is not always clear which features are appropriate to predict the relevance of a given annotation.

In this paper we propose a new image annotation method called MFS-Map (Multi-Feature Space Map). It automatically identifies which features are useful to determine the relevance of each annotation. This is possible because the feature vectors resulting from the extraction algorithms are not concatenated into a single large feature vector. Rather, we divide the features into a number of feature subspaces. This allows us to find relationships between annotations and regions of the subspaces. The useful relationships are automatically selected and represented as rules which are employed to predict annotations for an input non-annotated image.

We evaluated MFS-Map employing two publicly available datasets: MIR Flickr and Image CLEF 2011. We compared our results with widely employed annotation approaches and MFS-Map almost always obtained both significantly superior precision and faster training and testing times.

2. BACKGROUND AND RELATED WORK

2.1 Feature extraction

Each feature extraction algorithm captures a particular visual aspect from images. For example, RGB and HSV histograms are employed to capture global color content. Gist features [11] are employed to capture global spatial structure content. Local features can be represented, for example, by bag-of-visual-words feature vectors using SIFT [9, 12]. SFTA features can be used to efficiently describe texture information [4]. We refer to feature vectors computed from visual content as *visual features*.

Images often have textual features associated, which can be used by image annotation methods [14]. Textual features can be represented by a bag-of-words feature vector, which consists of a histogram that counts the frequency of each word in the text associated with the images. We refer to feature vectors computed from textual information associated with images as *textual features*.

Since images present large visual variability, image annotation methods employ multiple extractors to analyze images. Two popular approaches to combine the extracted visual and textual features are early-fusion and late-fusion. In early-fusion, the feature vectors are concatenated into a single large feature vector. In late-fusion, the annotation method handles each feature vector separately, returning, for example, a relevancy score based on each individual feature. The final relevancy score is the combination of the output obtained for each feature.

Both early-fusion and late-fusion have limitations. Early-fusion generates feature spaces with a large number of dimensions, worsening the dimensionality curse [8]. In late fusion, the strategy employed to combine the scores may reduce precision or recall. In late fusion, non-relevant features may introduce errors in the decision process [5].

2.2 Related work

Image annotation can be modeled as a multi-label classification problem [2, 10]. In multi-label classification, each object can be associated with a set of labels. In the context of image annotation, the objects are the images and the labels are the annotations.

Cross-training is a widely employed approach for multi-label classification in image annotation [2]. It consists in training a binary classifier for each annotation. The classifiers are employed to predict which annotations are relevant to a non-annotated image. Each classifier is trained by using images that contain their respective annotation as positive examples and the remaining images as negative examples. Annotations are scored based on the probability returned by the respective binary classifier.

Finding classifiers' optimal parameters for cross-training annotation can be extremely time consuming. Most classifiers have a set of user-defined parameters that directly affects accuracy. For instance, in order to train a Support Vector Machine (SVM), it is necessary to set the kernel function type, the kernel function parameters (e.g. the exponent of a radial basis function) and the misclassification cost. In order to obtain acceptable results, it is necessary to find the set of parameters that optimize the classifier performance. This is achieved by cross-validation using different combinations of parameter values.

Another disadvantage of cross-training is that images in the training phase must be divided into positive and negative examples. Since the number of examples in the positive class usually is significantly lower than the number of negative examples, the classification problem is imbalanced, degrading the classification accuracy [6].

Another approach to image annotation consists in using nearest-neighbors methods to annotate images based on annotations of visually similar images [16, 15]. Visual similarity is estimated by calculating the distance between images' feature vectors. Weighted nearest-neighbors models have shown to provide state of the art results in image annotation [1, 13].

A simple annotation model can be formulated by scoring images taking advantage of annotations of its k nearest-neighbors in the training image set. For example, a 1 nearest-neighbor (1-NN) model annotates images with the existing annotations of the most similar image from the training set.

3. THE PROPOSED METHOD

MFS-Map (Multi-Feature Space Map) is an automatic image annotation method that, given a non-annotated image, returns a list of the annotations that best describes its visual content. Each annotation is scored based on its predicted relevance.

In order to predict the annotation relevance, MFS-Map generate rules that describe relationships between annotations and regions in feature spaces. The rules are generated from a training set of images where each image may contain one or more annotations. The rule generation is carried out by the following steps:

1. Extract visual and textual features from the training set of images employing a set of extraction algorithms. Each extraction algorithm yields a feature space where the images are represented by feature vectors;
2. Convert the visual and textual features to *feature items*. Then, represent each image from the training set by a group of items, called itemset, where the items are the images's annotations and its feature items;
3. Generate rules in the format {feature item} → {annotation}.

Feature items are computed from the images' extracted feature vectors and are defined as follows:

DEFINITION 1. *A feature item f_i is either a centroid label generated from visual features or a word label generated from bag of words features.*

The MFS-Map is described as follows. Sections 3.1 and 3.2 describe how MFS-Map generates feature items from visual and textual features. Section 3.3 presents the algorithm employed to obtain rules from the itemset representations of the image (step 3). Finally, section 3.4 describes MFS-Map's annotation algorithm.

3.1 Visual features

MFS-Map obtains cluster centroids from visual features applying k-means clustering in the feature spaces obtained by each extraction algorithms. A naive way to obtain the cluster centroids would be to run k-means by setting the number of clusters to the number of annotations present in the training set. This approach assumes that each annotation can be represented by a single cluster. However, practical scenarios tend to be more complex. For example, each annotation may be better represented by more than one cluster.

MFS-Map improves clustering quality by using annotations of the training images to solve the clustering problem for each annotation separately. Thus, the clustering algorithm for each feature space is:

1. For each annotation a:
   (a) Insert in the set S_a every feature vector from the training set that contain the annotation a;
   (b) Apply k-means to S_a and insert the resulting centroids $\{c_1, c_2, \cdots, c_k\}$ into the set of centroids $\mathbf{C}$;
2. Return $\mathbf{C}$.

In order to obtain the feature items from the images, MFS-Map finds for each feature space the centroid nearest to the image feature vector and adds it to the image itemset representation. The centroid labels represent regions of the feature spaces obtained by the extraction algorithms where the images are similar with respect to a visual aspect (e.g. texture, color or shape).

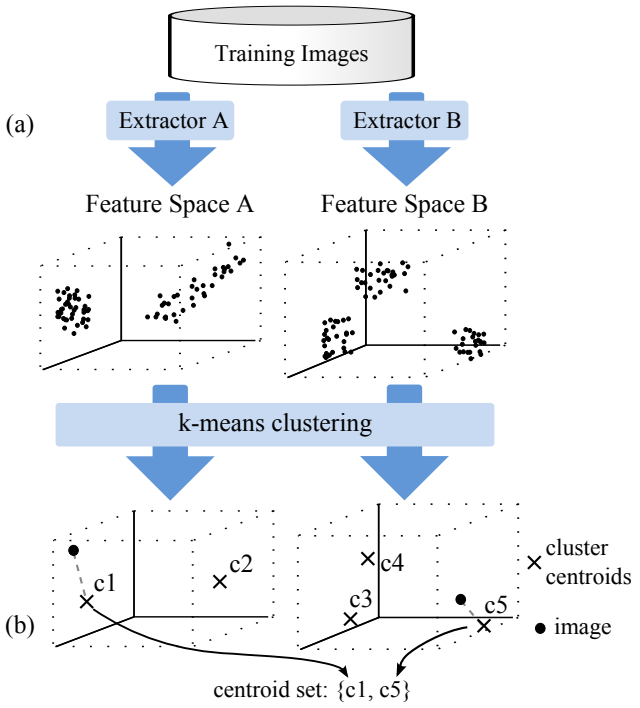

Figure 2: Centroid labels extraction in MFS-Map. (a) feature vector representations are extracted from the images. (b) resulting centroids obtained by applying k-means in feature spaces A and B separately. The resulting set of centroid labels for an image is obtained by finding the nearest centroid in each feature space.

Fig. 2 illustrates how MFS-Map extracts centroid labels from images for a scenario in which two extraction algorithms are employed. In Fig. 2(a), the extraction algorithms extract feature vector representations of the images. In this example, the feature spaces are represented as three-dimensional spaces and each feature vector is represented by a point. Fig. 2(b) shows the resulting centroids obtained by applying k-means in feature space A and B separately. For feature space A two centroids were obtained (c_1, and c_2) and for feature space B three centroids were obtained (c_3, c_4 and c_5). The resulting set of centroids of an image is obtained by finding the nearest centroid in feature spaces A (c_1) and B (c_5).

3.2 Bag-of-words features

MFS-Map's strategy to obtain itemset representation from bag-of-words features is different from the strategy employed for visual features. For a bag-of-words feature MFS-Map assigns a label for each possible textual word. If the image textual representation contains the word, then the corresponding word label will be included into the image's itemset. The word labels can be analogous to the centroid labels employed to represent visual features in section 3.1. Since generating items from bag-of-words features does not require clustering, its computation is more efficient. Additionally, we apply this same strategy for visual features represented by the bag-of-visual-words approach (e.g. SIFT).

3.3 Rule generation

In the rule generation phase, MFS-Map takes as input the itemset representations of the training images and finds a set of rules that are employed to predict annotation relevance. The itemsets are composed of centroid labels, word labels and annotations of the training images, as discussed in sections 3.1 and 3.2. Before presenting the rule generation algorithm, let us define the confidence conf($\{f_i, a_j\}$) of a feature item f_i and annotation a_j pair.

DEFINITION 2. *The confidence conf($\{f_i, a_j\}$) of a feature item f_i and annotation a_j pair is given by:*

$$conf(\{f_i, a_j\}) = freq(\{f_i, a_j\})/freq(\{f_i\}), \quad (1)$$

where freq($\{f_i, a_j\}$) is the number of times that the pair feature item f_i and annotation a_j occurs in the same itemset and freq($\{f_i\}$) is the number of times f_i appears in an itemset.

The rule generation algorithm calculates a confidence value for each pair $\{f_i, a_j\}$. The confidence value of a pair $\{f_i, a_j\}$ is an estimate of the usefulness of the feature item f_i in predicting the relevance of annotation a_j.

The rules are generated from all pairs $\{f_i, a_j\}$ whose confidence value is higher than a minimum confidence threshold, denoted by *minConf*. The format of a rule is $\{f_i\} \rightarrow \{a_j\}$, where f_i is defined as the antecedent and a_j is defined as the consequent of the rule.

The rule generation algorithm requires a single pass over the set of itemsets to count the frequency of pairs of feature items and annotation and the frequency of each feature item. Since the number of feature items and annotation pairs occurring is usually significantly lower than the number of all possible pairs, the pair frequency is stored using a hash table data structure in order to reduce memory usage.

3.4 Annotation relevance prediction

The annotation phase of MFS-Map takes as input the set of mined rules and predicts annotations for an input non-annotated image. Each annotation returned is scored by its predicted relevance. The first part of the annotation phase consists of extracting the itemset representation of the input image applying the procedures described in sections 3.1 and 3.2. For non-annotated images, the itemset representation does not contain annotations and thus, it is composed by feature items only (i.e. centroids or words). Additionally, this procedure requires the extraction of feature vectors from the input image using the same set of extraction algorithms applied to the training phase. Since centroids were already computed during the training phase, no clustering is required for visual features during the annotation phase.

In the next step of the annotation phase, MFS-Map selects from the set of mined rules all rules whose antecedent contains an item that is also in the itemset of the input image. The relevance score of an annotation a_i is given by the mean confidence of all selected rules whose consequent contains a_i. For example, if the itemset representation of the input image was $\{f_2, f_4\}$ and the set of mined rules were:

1. $\{f_1\} \rightarrow \{$'animal'$\}$(conf = 0.67)
2. $\{f_4\} \rightarrow \{$'sunset$\}$(conf = 0.82)
3. $\{f_2\} \rightarrow \{$'city'$\}$(conf = 0.72)
4. $\{f_2\} \rightarrow \{$'sunset'$\}$(conf = 0.93)

then the rules 2, 3 and 4 would be selected because their antecedent contains either the feature item f_2 or f_4. Rule 1, however, would not be selected because feature item f_1 is not present in the itemset representation of the input image.

The relevance score of the annotation 'sunset', considering the selected rules, is the average of the confidence of the rules that contains the annotation 'sunset' in its consequent, that is, rules 2 and 4 which results in a relevance score of $(0.82 + 0.93)/2 = 0.875$. For annotations that do not appear in the consequent of any selected rules, such as 'animal' in the example, the relevance score is zero.

4. EXPERIMENTS

We evaluated our proposed method MFS-Map for the task of image annotations using two publicly available datasets: *MIR Flickr* [7] and *Image CLEF 2011*. Both datasets are employed by many works in automatic image annotation [13, 17, 14] and are composed of images downloaded from the Flickr site. The MIR Flickr and Image CLEF 2011 datasets contains, respectively, 25,000 and 18,000 images.

Each image in both datasets is manually annotated. The number of different possible annotations for MIR Flickr is 25 and for Image CLEF is 98. The annotations describe depicted objects (e.g. "cars", "dog", "flower") and scene description (e.g. " sunset", "indoor", "night"). Additionally, each image also has tags assigned by Flickr users. The tags are a valuable resource, but they contain noise, since not all tags are relevant to the image visual content. Because of the noise in the tags, performance evaluation is based on manual annotations and tags are used as textual features. Figure 3 shows a sample image from the MIR Flickr dataset, its user tags and manual annotations.

user tags: crane, gru, sunset, hdr, tramonto, cielo, sky, ray, raggi, light, luci, chdk

manual annotations: clouds, sky, structures, sunset

Figure 3: Sample image from the MIR Flickr dataset, its user tags and manual annotations.

To quantify the performance, we employed two metrics: the average precision (AP) and the break-even point precision (BEP). Both AP and BEP are computed for each annotation but can be averaged to provide a single measurement. AP and BEP results were obtained by ten-fold cross-validation with ten repetitions.

4.1 Feature extraction

The visual feature extractors employed in the experiments were the following:

RGB and HSV histograms: Histograms computed by requantizing each color channel of the RGB and HSV color spaces to 7 bins yielding two $7^3 = 343$ dimensional feature vectors.

HSV Histogram with layout information: Feature vector computed by dividing the input image into 3 horizontal stripes of the same height and computing a local

Table 1: Average precision (AP) and break even precision (BEP) obtained by each annotation method for the MIR-Flickr and Image CLEF 2011 datasets under different feature configurations. T corresponds to results obtained using only textual features (Flickr tags). V corresponds to results obtained using only visual features. V+T corresponds to results using both textual and visual features. Standard deviation is indicated between parentheses.

		T		V		V+T	
Dataset	**Method**	**AP (%)**	**BEP (%)**	**AP (%)**	**BEP (%)**	**AP (%)**	**BEP (%)**
MIR Flickr	**MFS-Map**	**54.0 (0.69)**	**47.3 (0.66)**	**41.1 (0.42)**	**39.3 (0.42)**	**55.1 (0.66)**	**52.0 (0.37)**
	EF-SVM	29.9 (0.26)	26.9 (0.35)	28.5 (0.89)	25.0 (1.28)	29.9 (0.26)	27.0 (0.44)
	LF-SVM	29.9 (0.27)	26.9 (0.35)	29.9 (0.73)	27.0 (0.86)	29.9 (0.31)	26.9 (0.47)
	1-NN	30.8 (2.92)	28.3 (4.36)	35.1 (0.43)	37.7 (0.53)	30.9 (3.19)	28.5 (4.80)
Image CLEF	**MFS-Map**	**48.1 (0.27)**	**45.5 (0.35)**	**42.9 (0.33)**	40.2 (0.34)	**49.4 (0.58)**	**47.6 (0.59)**
	EF-SVM	37.7 (0.22)	37.2 (0.22)	36.8 (0.59)	35.7 (0.73)	37.8 (0.27)	37.2 (0.32)
	LF-SVM	37.8 (0.21)	37.2 (0.22)	37.7 (0.18)	37.2 (0.19)	37.8 (0.33)	37.2 (0.31)
	1-NN	38.2 (1.36)	37.9 (2.10)	40.2 (0.29)	**41.6 (0.53)**	38.1 (0.98)	37.7 (1.59)

HSV Histogram. The local HSV histogram is computed by re-quantizing each color channel to 5 bins yielding a $3 \times 5^3 = 375$ dimensional feature vector.

SIFT: 100 bin bag-of-visual-words histogram computed by extracting local SIFT features using a dense multi-scale grid for sampling.

Gist: 512 dimensional feature vector computed using the Gist descriptor by resizing the image size to 256×256 pixels and using 8 orientations per scale.

SFTA: 21 dimensional feature vector computed by applying the SFTA texture descriptor.

Gist and SFTA feature vectors were normalized to the range 0.0 to 1.0. The remaining feature vectors were L_1 normalized.

We also employed the user tags associated to each Flickr image as textual features represented by a bag-of-words feature vector. We first selected user tags whose minimum frequency in the dataset is 25 and built binary feature vectors where each entry corresponds to a user tag. If the tag is present in the image metadata, the corresponding entry takes the value 1, otherwise the entry takes the value 0. In our experiments, we refer to features obtained by image extractors as visual features and bag-of-words features obtained from user tags as textual features.

4.2 Annotation methods

We compared MFS-Map performance to the three following methods: (i) cross-training using late-fusion (**LF-SVM**); (ii) cross-training using early-fusion (**EF-SVM**) and (iii) a nearest neighbor model that annotates images with the annotations of the most similar image from the training set (**1-NN**).

For MFS-Map we set the confidence parameter to 0.5, which we found out to provide the best average performance. Additionally, for textual features and SIFT features, which are represented by the bag-of-words approach, we configured MFS-Map to generate items from textual or visual words as described in Section 3.2.

For cross-training we employed SVM classifiers with radial basis function (RBF) kernels. In order to find the best SVM parameters (SVM cost and RBF kernel degree) we employed cross-validation before each binary SVM was trained. An important aspect of cross-training is that the classifiers must provide class membership probabilities for each prediction. Since SVMs do not output class membership probabilities, we mapped SVM scores to probabilities by learning a regression model using 20% of the training data.

For SIFT and Flickr tags we employed the cosine distance to measure dissimilarity. For all other features we used the Euclidean distance. Additionally, for MFS-Map, if a word or visual word was present in the feature vector, we added a corresponding item into the itemset representation of the image.

All annotation methods were implemented in C++. For cross-training annotation (LF-SVM and EF-SVM) we employed LibSVM [3].

4.3 Annotation precision

In this section we describe the experiments performed to evaluate MFS-Map precision for the task of annotating images. Table 1 shows the mean AP and BEP of each annotation method for different feature configurations: T, V and T+V. Column T shows the results obtained using only textual features (Flickr tags). Column V shows the results obtained using only visual features. Column V+T shows the results obtained using both textual and visual features. Standard deviation is indicated between parentheses.

We compared AP and BEP values for each combination of feature configuration and dataset using a two-tailed t-Student test with p=0.05 and the best results are indicated in bold. For both MIR Flickr and Image CLEF, MFS-Map obtained the best results for the three feature configuration (T, V and T+V) with one exception. 1-NN obtained the best BEP for Image CLEF using visual features.

The combination of visual and textual features (V+T) improved MFS-Map precision by a small but statistically significant value with p=0.03. However, for the other methods, the V+T precision was equal or inferior to the values obtained with only visual or only textual features. This indicates that MFS-Map performs well also when different feature modalities (visual and textual) are combined.

4.4 Training and annotation time

Figure 4 shows a plot of training plus annotation time (total time) versus average precision for each method. Times were obtained using a random sample of 20% of the MIR Flickr dataset and 40% of the Image CLEF dataset. The experiments were executed in a computer with an Intel i7 2.66GHz processor, 8GB RAM running Windows 64-bit OS. AP values may differ from section 4.3 because of the sampling process. For the two datasets, MFS-Map was the fastest method. Additionally, as was shown in section 4.3, MFS-Map also obtained the highest average precision.

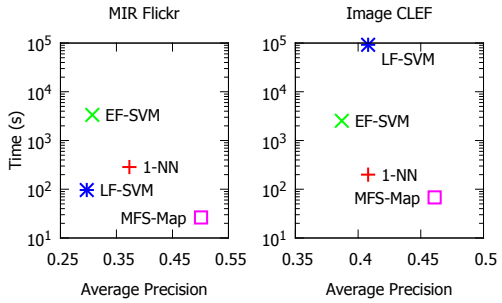

Figure 4: Average precision (AP) versus training and annotation time for each annotation method. Our proposed method, MFS-Map, is indicated by the black × symbol.

1-NN does not have a training phase and the total time corresponds to its annotation (test) time only. However, for all other methods, training accounted for at least 95% of total time. MFS-Map training time was at least 48 times faster for the Image CLEF dataset and 3.6 times faster for MIR Flickr dataset when compared to the other cross-training approaches (EF-SVM and LF-SVM). EF-SVM and LF-SVM larger training time can be attributed to the need to train a separated classifier for each annotation. Thus, when the number of different annotations is higher - Image CLEF has 98 annotation while MIR Flickr has 25 - the cross-training time is also larger.

5. CONCLUSIONS

In this paper we proposed MFS-Map, a novel automatic image annotation method. We compared our results to cross-training approaches (early-fusion and late-fusion) and a nearest neighbor model. MFS-Map obtained both superior precision and faster training and annotation (testing) times.

An important aspect of MFS-Map is that it efficiently combines the images' visual content and textual context to improve annotation precision. Additionally, the feature vectors resulting from the extraction algorithms are not concatenated into a large single feature vector. Rather, MFS-Map divides the problem, handling each feature space separately. This allows MFS-Map's rule generation algorithm to discard rules that present weak relationship between features and annotation. This is important in annotation problems, since a particular feature may be useful to annotate an image but may introduce noise for other images.

6. ACKNOWLEDGMENTS

This research has been supported by FAPESP (São Paulo State Research Foundation), CNPq (Brazilian National Research Council) and CAPES (Brazilian Coordination for Improvement of Higher Level Personnel).

7. REFERENCES

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *CVPR*, pages 1–8, June 2008.

[2] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[3] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2(3):27:1—-27:27, 2011.

[4] A. F. Costa, G. Humpire-Mamani, and A. J. M. Traina. An Efficient Algorithm for Fractal Analysis of Textures. In *SIBGRAPI*, pages 39–46, 2012.

[5] A. Depeursinge and H. Müller. Fusion techniques for combining textual and visual information retrieval. In *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*, chapter 6, pages 95–114. Springer Berlin Heidelberg, 2010.

[6] H. He and E. A. Garcia. Learning from Imbalanced Data. *IEEE Trans. on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[7] M. J. Huiskes and M. S. Lew. The MIR Flickr Retrieval Evaluation. In *ICMR*, pages 39–43. ACM, 2008.

[8] H. P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1):1–58, 2009.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] H. Müller, P. Clough, T. Deselaers, and B. Caputo. *ImageCLEF Experimental Evaluation in Visual Information Retrieval*. Springer Berlin Heidelberg, 2010.

[11] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[12] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[13] J. Verbeek, M. Guillaumin, T. Mensink, and C. Schmid. Image annotation with tagprop on the MIRFLICKR set. In *MIR*, pages 537–546. ACM, 2010.

[14] G. Wang, D. Hoiem, and D. Forsyth. Building text features for object image classification. In *ICPR*, pages 1367–1374, 2009.

[15] D. Zhang, M. M. Islam, and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2011.

[16] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In *CVPR*, volume 2, pages 2126–2136. IEEE, 2006.

[17] A. Znaidia, A. Shabou, A. Popescu, H. le Borgne, and C. Hudelot. Multimodal Feature Generation Framework for Semantic Image Classification. In *ICMR*, pages 38:1—-38:8, 2012.