Bootstrapping Visual Categorization With Relevant Negatives

Xirong Li, CeesG. M. Snoek, *Senior Member, IEEE*, Marcel Worring, *Member, IEEE*, Dennis Koelma, and Arnold W. M. Smeulders, *Member, IEEE*

***Abstract*—Learning classifiers for many visual concepts are important for image categorization and retrieval. As a classifier tends to misclassify negative examples which are visually similar to positive ones, inclusion of such misclassified and thus *relevant negatives* should be stressed during learning. User-tagged images are abundant online, but which images are the relevant negatives remains unclear. Sampling negatives at random is the *de facto* standard in the literature. In this paper, we go beyond random sampling by proposing Negative Bootstrap. Given a visual concept and a few positive examples, the new algorithm iteratively finds relevant negatives. Per iteration, we learn from a small proportion of many user-tagged images, yielding an ensemble of meta classifiers. For efficient classification, we introduce Model Compression such that the classification time is independent of the ensemble size. Compared with the state of the art, we obtain relative gains of 14% and 18% on two present-day benchmarks in terms of mean average precision. For concept search in one million images, model compression reduces the search time from over 20 h to approximately 6 min. The effectiveness and efficiency, without the need of manually labeling any negatives, make negative bootstrap appealing for learning better visual concept classifiers.**

***Index Terms*—Model compression, negative bootstrap, relevant negative examples, visual categorization.**

I. Introduction

LABELED examples are crucial to learn visual concept classifiers for image categorization and retrieval. To be more precise, we need positive and negative examples with respect to a specific concept, as shown in Fig. 1. When the number of concepts is large, obtaining labeled examples in an efficient way is essential. Traditionally, labeled examples are annotated by expert annotators. However, expert labeling is labor intensive and time consuming, making well-labeled examples expensive to obtain and consequently their availability is limited.

Much research has been conducted towards inexpensive solutions to acquire positive examples, e.g., from web image search results [1]–[3] or socially tagged data [4]–[6], or by online collaborative annotation [7]–[9]. For instance, Schroff *et al.* [1] train a visual classifier on web image search results of a given concept, and re-rank the search results by the classifier. Though the automated approaches are not comparable to dedicated manual annotation [4], [5], their output provides a good starting point for manual labeling. Deng *et al.* [9] build an ImageNet wherein positive examples of a WordNet concept are obtained by labeling web image search results of the concept using a micro payment service. Compared to traditional expert labeling, the new labeling mechanism yields positive examples for many categories with lower cost. In this paper we assume that positive examples are obtained by (one of) the approaches described above, and focus on obtaining *negative* examples.

Since negative examples of a concept belong to many other concepts, most of expert labeling efforts are dedicated to annotating negatives. One might consider bypassing the negative labeling problem by one-class learning, which creates classifiers using positive examples only [10]. However, because negative examples also bear valuable information, they are important for classification. This has been well observed in Tao *et al.* [11] for interactive image retrieval. Our empirical study shows that visual classifiers trained by one-class learning are inferior to classifiers trained by two-class learning [12]. So labeling negatives remains essential, but methods which can reduce the manual labeling effort are needed.

Obtaining negative examples seems to be trivial, as they are abundant in large photo repositories such as Flickr and Facebook. For a specific concept, say 'whale', due to the relatively sparse occurrence of its positive examples against its negatives in these repositories, sampling a fraction of the data at random already yields a set of genuine negatives. Indeed, random sampling is the *de facto* standard in the literature for obtaining negative examples [1], [3]–[5], [12], [13]. Given the massive amount of potential negatives, training on a small proportion of the data also makes it feasible to learn classifiers on normal computers. Further, conducting random sampling multiple times leads to ensemble learning [14]. This methodology trains multiple meta classifiers on multiple (disjoint) subsets of the data, and combines the meta classifiers to make final decisions. Evidence from the machine learning community shows encouraging results on learning from large data in an ensemble [15]. Also, there are good examples in the multimedia community, leveraging the methodology for learning an ensemble of

Manuscript received March 13, 2012; revised August 10, 2012; accepted September 29, 2012. Date of publication January 09, 2013; date of current version May 13, 2013. This work was supported in part by the Dutch national program COMMIT, the STW SEARCHER Project, the SKL-SDE Project "Research on core techniques of unstructured data management," and NSFC 61103062. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Samson Cheung.

X. Li is with the MOE Key Lab of Data Engineering and Knowledge Engineering, School of Information, Renmin University of China, Beijing 100872, China, and also with the State Key Lab of Software Development Environment, Beijing 100191, China (e-mail: xirong@ruc.edu.cn).

C. G. M. Snoek, M. Worring, and D. Koelma are with the Intelligent Systems Lab Amsterdam, University of Amsterdam, 1098XH Amsterdam, The Netherlands.

A. W. M. Smeulders is with the Intelligent Systems Lab Amsterdam, University of Amsterdam, 1098XH Amsterdam, The Netherlands, and also with the Centrum Wiskunde & Informatica, 1098XG Amsterdam, The Netherlands.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2238523

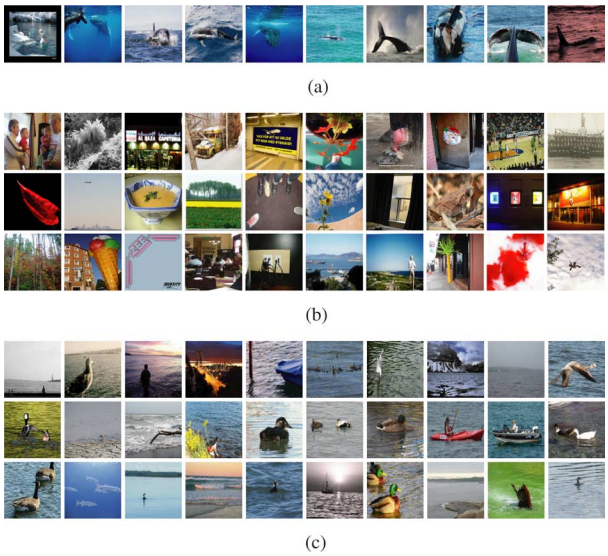

Fig. 1. A positive set and two negative sets of visual concept 'whale'. The negative set (b) is obtained by random sampling from a large set of user-tagged images, while the negative set (c) is sampled from the same set by the proposed algorithm. Compared to (b), our negatives are visually more similar to the positive set (a). Hence, they are more relevant, yielding more accurate visual classifiers for concept search. Note that the relevant negatives are obtained automatically, without the need of manual verification. (a) Positive examples of 'whale'. (b) Negative examples of 'whale' obtained by random sampling. (c) Relevant negatives of 'whale' (this paper).

classifiers for visual search [16], [17]. Ensemble learning with random negative seems attractive for learning visual concepts from abundant negatives.

A classifier tends to misclassify negative examples which are visually similar to positive examples. For learning a visual classifier for the concept 'whale', confusing negatives such as images of birds in water, as illustrated in Fig. 1, should be included during learning. As such relevant negatives are in minority, random sampling alone is not enough for identifying them. In order to go beyond random sampling, the question arises *which examples are relevant negatives?*

In principle, the relevant negatives shall have visual patterns partially overlapping the positive examples. Following this thought, one might try to manually add positive examples of confusing negative classes such as 'bird' or 'ocean' to the negative training data. However, the relevance of a negative example depends on the underlying visual features, kernels, and classifiers, and is not necessarily consistent with what an observer may expect. It is therefore difficult to specify relevant negative classes by hand-crafted rules. One may consider an active learning system [18], [19], asking an annotator to label examples the system considers most useful. Active learning helps reduce human labeling effort as reported for interactive visual search reranking [20], [21] and interactive image annotation [22]. Nonetheless, human interaction is mandatory in an active learning process. In contrast, we aim to acquire relevant negatives in a fully automatic manner.

Although ensemble learning with random negatives exploits more negative data and therefore may include relevant negatives accidently, its performance is bounded by the lack of relevant negatives. Moreover, the execution of all its individual meta classifiers requires a classification time proportional to the size of the ensemble. Recently, Maji *et al.* [23] found that for the histogram intersection kernel, the decision function of an SVM classifier can be well approximated by linear interpolation on a fixed number of precomputed points. This finding makes the classification time independent of the number of support vectors, leading to efficient execution of individual classifiers. Nonetheless, the execution time of the ensemble classifier remains proportional to the number of meta classifiers.

When extending the number of concepts, there will be a more dense division of the data space in terms of semantic classes. As class boundaries need to be better defined, there is a natural need to increase the number of training examples. When adding new negative training examples, one has the option to add random negatives, or to select new relevant negatives. In this paper we plea to go for relevant negatives. In that case, the confinement of the class boundary to just the area of the class and nothing more proves to be as important as the extension of the class boundary by adding new positives. From the above considerations it is clear that random negatives will not help as they are far away from the class boundary. When adding new elements, relevant negatives are indeed superior to random negatives. But the superiority is achieved only when appropriate care is taken to balance classes during training. Therefore, for the purpose of extending concept recognition to many classes, in this paper we make three contributions to visual categorization.

1) First, we argue that when extending the training set, one should select relevant negatives as the extension to be preferred over random negatives for better classifier performance.
2) Second, for good training under unbalanced classes, we propose Negative Bootstrap, an iterative negative ensemble learning strategy, to select relevant negatives from many user-tagged images, without the need of new annotation.
3) Third, for computational feasibility, we introduce Model Compression, which extends [24] from a single classifier to an ensemble of classifiers, making classification time independent of the number of meta classifiers.

The rest of the paper is organized as follows. Related work is reviewed in Section II. We detail the new negative bootstrap algorithm in Section III. Experiments are set up in Section IV. Result analysis is given in Section V. We conclude the paper in Section VI.

II. Related Work

To automatically create a negative training set for a given concept, the mainstream methods randomly sample a relatively small subset from a large pool of (user-tagged) examples [1], [3]–[5], [12], [13], [16]. The pool may consist of web images with free texts [1], [3] or consumer photos with user provided tags [4], [13]. Apart from the obvious fact that random sampling is simple and easy to use, we attribute its popularity to two reasons. First, except for some over frequent concepts such as 'sky' and 'person', the chance of finding genuine positive examples in a random fraction of the pool is low. False negatives can be further reduced by removing images labeled with the given concept or its semantically related tags [12], [13], [24]. Second, as the possible negatives substantially outnumber the positive

training set, downsampling the negatives bypasses class imbalance which is known to affect classifier learning [17], [25], [26]. If the pool is sufficiently large, one might end up with a set of reliable negatives, but not necessarily the most relevant ones.

When negative examples are selected at random, the performance of individual classifiers varies. According to Breiman's bagging theory [27], such variance can be reduced by model averaging. Hence, both Natsev *et al.* [16] and Tao *et al.* [17] perform random sampling multiple times to create multiple classifiers, and combine them uniformly. While the negative examples vary, the positive examples are fully used because they are relatively rare. To distinguish such a strategy from classical bagging [27] which conducts re-sampling on both positive and negative examples, Tao *et al.* term it asymmetric bagging [17]. Although the robustness of the final classifier is improved by classifier aggregation, the quality of asymmetric bagging is bounded by the lack of relevant negatives. Moreover, since all meta classifiers need to be executed, the classification time is proportional to the amount of meta classifiers. In sum, the lack of relevant negative examples and the computational cost for running all meta classifiers put the effectiveness and efficiency of asymmetric bagging into question.

Obtaining negative examples with manual annotation for free has also been studied in the context of text categorization, e.g., [28]. There, unlabeled examples are inserted into the negative set, if they are most dissimilar to the positives, or predicted as negatives with high confidence by current classifiers. Yan *et al.* [29] reported a similar idea in the context of video retrieval. Though sampling at the bottom probably yields reliable negatives, an intrinsic drawback is that those negatives are already correctly classified, adding them to the training process is not so useful by definition. Indeed, Natsev *et al.* [16] reported that such conservative sampling is inferior to random sampling.

The algorithm we introduce in this paper bears some conceptual resemblance to active learning [18] and AdaBoost [30], as all of them seek useful examples for learning a new classifier. But the new algorithm has a number of characteristics which make it different. There are two notable differences between negative bootstrap and active learning. First, in contrast to active learning which requires human interaction to label examples selected in each round, negative bootstrap selects relevant examples without human interaction. Second, different from active learning wherein the new input data is supposed to be unlabeled and comprised of positive and negative examples, our setting assumes that the new input data contains negatives only. Hence, in active learning, examples the system is most uncertain about, namely closest to the decision boundary [18], are considered informative. Negative bootstrap, by contrast, selects negative examples which are most misclassified, i.e., falling on the positive side and distant from the boundary. Inclusion of such negatives in training pushes towards a tight boundary in the area of the target class, yielding classifiers with better discrimination ability. Compared to AdaBoost which works on fully labeled data, our algorithm takes user-tagged data as its starting point. Unlike AdaBoost, we do not have to maintain the distribution of weights on the entire training data, and we do not need manual labeling of negatives. Therefore, negative bootstrap is more suited for exploiting large datasets.

Our model compression is inspired by Maji *et al.* [23]. The authors accelerate histogram intersection kernel SVMs by introducing a fast kernel computation approximation. The key innovation is that by exploiting the additive property of the histogram intersection kernel, the computation of the SVM decision function is re-expressed as the sum of decision functions with respect to individual feature dimensions. Further, for each dimension, its decision function can be efficiently computed by linear interpolation on a limited set of precomputed decision scores. As a consequence, the test time becomes independent of the number of support vectors. While Maji's algorithm targets at accelerating a single classifier, we aim to compress an ensemble of classifiers such that the classification time will be independent of the amount of meta classifiers. When using the compressed model for finding relevant negatives, the training time is also reduced.

Besides [23] as a method for speeding up classification, we also notice an increasing interest in efficient training of the histogram intersection kernel SVMs [31]–[33]. Since these algorithms work on the meta classifier level while our algorithm works on the ensemble level, they are complementary. We leave the study of their integration for future exploration.

III. Negative Bootstrap

Given a set of unlabeled images, we search for images which contain a specific concept ω by employing a visual classifier of the concept [34]. Let x be an image. Its content-based representation is a d-dimensional feature vector. We will refer to an image and its corresponding feature vector interchangeably, using $x(i)$ to indicate the i-th dimension of the vector. We use $g(x)$ to denote a classifier, which produces a real-valued score of an image being a positive instance of the target concept. In particular, $g(x) > 0$ means the image is classified as positive, and negative otherwise. We need classifiers robust to high dimensional features common in visual classification. To that end, we choose SVMs which can simultaneously minimize the empirical classification error and maximize the functional margin, i.e., the largest distance of the decision boundary to the nearest training data [35]. The maximum-margin property makes SVMs a solid choice for building visual classifiers, as demonstrated by [36]–[39]. The SVM version of $g(x)$ is defined as

$$g(x) = b + \sum_{j=1}^{n} \alpha_j \cdot y_j \cdot \mathcal{K}(x, x_j), \tag{1}$$

where α_j is the positive coefficient of support vector x_j, b is the intercept, $y_j \in \{-1, 1\}$ is a class label, n is the number of support vectors, and $\mathcal{K}$ a kernel function measuring the visual similarity between two examples. A good, but expensive, choice for $\mathcal{K}$ is the χ^2 kernel [37]–[39]. Recently, both Maji *et al.* [23] and Uijlings *et al.* [40] have shown that the histogram intersection kernel is close to the χ^2 kernel in terms of visual classification accuracy, but it is far more efficient. Moreover, Wu [33] proofs that the histogram intersection kernel is a positive definite kernel for non-negative real-valued histograms. Given theoretical and empirical justifications, we prefer this kernel in our study. Given two images x and x', the histogram intersection kernel is defined as

$$\mathcal{K}(x, x') = \sum_{i=1}^{d} \min(x(i), x'(i)). \tag{2}$$

To obtain $g(x)$, we need both positive and negative training data. We assume that positive data are obtained for instance by the approaches described in Section I. As indicated, for negative training data, we aim to harvest them from user-tagged images on the web, but with no need of manual verification. Let $\mathcal{B}_+$ be a positive set, $\mathcal{S}$ a set of user-tagged images independent of $\mathcal{B}_+$, and $\mathcal{B}_-$ a negative set from $\mathcal{S}$. We derive $g(x)$ from $\mathcal{B}_+$ and $\mathcal{B}_-$:

$$g(x) \leftarrow \texttt{learn-classifier}(\mathcal{B}_+, \mathcal{B}_-). \tag{3}$$

Because $\mathcal{B}_+$ usually has a limited number of elements, we make full use of it. The learning process (3) optimizes b and $\{\alpha_j\}$ such that hinge loss is minimized, with the following constraints [41]:

$$\sum_{j=1}^{n} \alpha_j \cdot y_j = 0, \quad 0 \le \alpha_j \le C, \quad j = 1, \ldots, n, \tag{4}$$

where C is the regularization parameter.

A. Iterative Negative Ensemble Learning

Given a target concept ω and its positive set $\mathcal{B}_+$, we aim to select a set $\mathcal{B}_-$ which contains relevant negatives from $\mathcal{S}$. The relevance of a negative example depends on the classifier. Negative examples which are most misclassified, that is, predicted as positive instances with the largest scores, are most relevant to improve classification. Hence, we shall sort $\mathcal{S}$ in descending order by $g(x)$ and select the top ranked examples to form $\mathcal{B}_-$, formalized as the following condition on $\mathcal{B}_-$:

$$\forall\, x \in \mathcal{B}_-,\ x' \in \mathcal{S} \backslash \mathcal{B}_- \Rightarrow g(x) > 0,\ g(x) > g(x'). \tag{5}$$

In practice (5) is not directly applicable, because the negatives $\mathcal{S}$ are not manually verified, meaning an exhaustive search will incorrectly treat genuine positive examples as relevant negatives. Moreover, the large-scale property of $\mathcal{S}$ makes the search computationally challenging. Downsampling is thus necessary. Due to the random factor in downsampling, negative examples as well as the classifier obtained in a single trial tend to be suboptimal. A common solution is to generate an ensemble of meta classifiers by multiple trials, and average their output, as has been used by Natsev *et al.* [16] for video retrieval and Tao *et al.* [17] for image retrieval. We also construct an ensemble of classifiers, but with the notable distinction that our negative examples are more relevant than the random negatives in [16], [17].

We use T to denote the number of iterations, and $t = 1, \ldots, T$ to index the iterations. Let $G_t(x)$ be the final classifier obtained after t iterations. In the t-th iteration, we conduct a two-stage adaptive sampling to acquire the most relevant negative examples according to $G_{t-1}(x)$, the latest classifier obtained in previous iterations. In the first stage, we randomly sample m examples from $\mathcal{S}$ to form a candidate set $\mathcal{U}_t$, as expressed by

$$\mathcal{U}_t \leftarrow \texttt{random-sampling}(\mathcal{S}, m). \tag{6}$$

To reduce the chance of incorrectly having genuine positives in $\mathcal{U}_t$, we let $m \ll |\mathcal{S}|$. In the second stage, we use $G_{t-1}(x)$ to classify each example in $\mathcal{U}_t$, and obtain $\widetilde{\mathcal{U}}_t$ in which each example is associated with a classification score. We express $\widetilde{\mathcal{U}}_t$ by

$$\widetilde{\mathcal{U}}_t \leftarrow \texttt{classify}(\mathcal{U}_t, G_{t-1}(x)). \tag{7}$$

We sort examples in $\widetilde{\mathcal{U}}_t$ by their scores in descending order and select the top ranked examples to form the relevant negative set, denoted by $\mathcal{B}_-^{(t)}$.

To derive a new classifier given the new negative data $\mathcal{B}_-^{(t)}$, if we simply add $\mathcal{B}_-^{(t)}$ to the existing training data, we will come face to face with the imbalanced data problem as the negatives accumulate. Moreover, the training complexity increases per iteration. Hence, in each iteration, we train a new classifier on $\mathcal{B}_+$ and $\mathcal{B}_-^{(t)}$. To make the positive and negative classes perfectly balanced, we set the number of selected negatives equal to $|\mathcal{B}_+|$, i.e.,

$$\mathcal{B}_-^{(t)} \leftarrow \texttt{select-top}(\widetilde{\mathcal{U}}_t, |\mathcal{B}_+|), \tag{8}$$

where $|\cdot|$ is the cardinality of a set. Subsequently, a new meta classifier $g_t(x)$ is learned from $\mathcal{B}_+$ and $\mathcal{B}_-^{(t)}$ using (3). Because $\mathcal{B}_-^{(t)}$ is composed of negatives most misclassified by the previous classifier, we can safely assume that the new classifier is complementary to its ancestors. Following the regular bootstrap aggregation, we combine the meta classifiers by model averaging:

$$G_t(x) = \frac{1}{t}\sum_{j=1}^{t} g_j(x) = \frac{t-1}{t} G_{t-1}(x) + \frac{1}{t} g_t(x). \tag{9}$$

Since no classifier is available in the first iteration, we acquire $\mathcal{B}_{w-}^{(1)}$ by random sampling.

For obtaining relevant negatives in the t-th iteration, we have to compute (7), which involves running $t-1$ meta classifiers. In total, the number of classifiers to be executed will be $T \cdot (T-1)/2$. Since the number of support vectors in a meta classifier has an order of $|\mathcal{B}_+|$, the time complexity of scoring an image would be $O(|\mathcal{B}_+| \cdot d)$, meaning an order of $O((t-1) \cdot |\mathcal{B}_+| \cdot d \cdot m)$ for computing (7). Moreover, because T iterations results in T meta classifiers and we have to apply all of them, searching for one concept has a time complexity proportional to the number of iterations. Acceleration is thus crucial for both training and testing.

B. Model Compression

We introduce model compression by generalizing the fast intersection kernel algorithm [23] from a single classifier to an ensemble of classifiers. Our compact model classifies an image at a constant time complexity, while simultaneously maintaining the effectiveness of negative bootstrap.

Notice that although we use uniform weights to combine classifiers (9), the weights can be optimized when we have access to extra validation data. To express (9) in a more generic form, let λ_t be a nonnegative weight for a meta classifier $g_t(x)$. Accordingly, we express the ensemble classifier $G_T(x)$ as

$$G_T(x) = \sum_{t=1}^{T} \lambda_t \cdot g_t(x). \tag{10}$$

Substituting (1) and (2) in (10) leads to

$$G_T(x) = \sum_{t=1}^{T} \lambda_t \cdot b_t + \sum_{i=1}^{d} \underbrace{\sum_{t=1}^{T} \sum_{j=1}^{n_t} \lambda_t \cdot \alpha_{t,j} \cdot y_{t,j} \cdot \min(x(i), x_{t,j}(i))}_{\text{decision value per dimension}}. \quad (11)$$

As shown in (11), the decision value is the sum of decision values of each dimension plus the sum of all intercepts from the T meta classifiers. Since the term containing the intercepts is independent of x, it can be easily compressed into a constant.

For a single classifier $g(x)$ as defined in (1), Maji *et al.* construct a function h_i to indicate the decision value computed on the i-th dimension [23]:

$$h_i(z) = \sum_{j=1}^{n} \alpha_j \cdot y_j \cdot \min(z, x_j(i)), \quad (12)$$

where z is a variable. They have proven that for an arbitrary z, $h_i(z)$ can be computed as a linear interpolation on $h_i(x_j(i))$ and $h_i(x_k(i))$, where x_j and x_k are two specific support vectors of $g(x)$. We argue that a similar conclusion holds when multiple classifiers are linearly combined.

To incorporate classifier combination, we first extend (12) to the following form:

$$H_i(z) = \sum_{t=1}^{T} \sum_{j=1}^{n_t} \lambda_t \cdot \alpha_{t,j} \cdot y_{t,j} \cdot \min(z, x_{t,j}(i)). \quad (13)$$

Notice that (13) is exactly the decision value per dimension in (11). Given the T meta classifiers, we use M to denote the number of support vectors in total, i.e., $M = \sum_{t=1}^{T} n_t$. By putting the i-th dimension of these support vectors together, we have a sequence of M elements $\{x_{1,1}(i), \ldots, x_{1,n_1}, \ldots, x_{T,1}, \ldots, x_{T,n_T}(i)\}$. We sort the sequence in ascending order, and use $\bar{x}_j(i)$ to denote the sorted elements, $j = 1, \ldots, M$. For each sorted element, we denote its corresponding model weight, support vector coefficient, and class label as $\bar{\lambda}_j$, $\bar{\alpha}_j$, and $\bar{y}_j$, respectively. The sorting and renaming operations allow us to rewrite (13) as

$$H_i(z) = \sum_{j=1}^{M} \bar{\lambda}_j \cdot \bar{\alpha}_j \cdot \bar{y}_j \cdot \min(z, \bar{x}_j(i)). \quad (14)$$

Taking into account the relative position of z with respect to the interval $[\bar{x}_1(i), \bar{x}_M(i)]$, we can accelerate the computation of (14). With the constraints (4), we have $\sum_{j=1}^{M} \bar{\lambda}_j \cdot \bar{\alpha}_j \cdot \bar{y}_j = 0$. Consequently, if $z \leq \bar{x}_1(i)$, we have $H_i(z) = z \sum_{j=1}^{M} \bar{\lambda}_j \cdot \bar{\alpha}_j \cdot \bar{y}_j = 0$. When $z \geq \bar{x}_M(i)$, due to the min function in (14), we have $H(z) = H(\bar{x}_M(i))$. For any z within the interval, there always exists an integer r_i such that $\bar{x}_{r_i}(i) \leq z \leq \bar{x}_{r_i+1}(i)$. We proof that $H_i(z)$ is a linear interpolation on $H_i(\bar{x}_{r_i}(i))$ and $H_i(\bar{x}_{r_i+1}(i))$ as follows:

$$H_i(z) = \beta_i \cdot H_i(\bar{x}_{r_i}(i)) + (1 - \beta_i) \cdot H_i(\bar{x}_{r_i+1}(i)), \quad (15)$$

TABLE I
THE NEGATIVE BOOTSTRAP ALGORITHM

Input: Positive examples $\mathcal{B}_+$, User-tagged images $\mathcal{S}$
Output: Compressed visual classifier $G_T(x)$

```
1  B_-^(1) ← random-sampling(S, |B_+|)
2  g_1(x) ← learn-classifier(B_+, B_-^(1))
3  G_1(x) ← compress-models({g_1(x)})
4  for t := 2 to T do
5     U_t ← random-sampling(S, m)
6     Ũ_t ← classify(U_t, G_{t-1}(x))
7     B_-^(t) ← select-top(Ũ_t, |B_+|)
8     g_t(x) ← learn-classifier(B_+, B_-^(t))
9     G_t(x) ← compress-models({g_1(x), ..., g_t(x)})
10 end
```

where $\beta_i = (\bar{x}_{r_i+1}(i) - z / \bar{x}_{r_i+1}(i) - \bar{x}_{r_i}(i))$. Hence, if we have $H_i(\bar{x}_1(i)), \ldots, H_i(\bar{x}_M(i))$ precomputed, $H_i(z)$ can be efficiently computed. Substituting (15) into (11), we obtain the decision value of the ensemble classifier as the sum of linear interpolations on a set of precomputed functions,

$$G_T(x) = b_0 + \sum_{i=1}^{d} \Big(\beta_i \cdot H_i(\bar{x}_{r_i}(i)) + (1 - \beta_i) \cdot H_i(\bar{x}_{r_i+1}(i)) \Big), \quad (16)$$

where

$$b_0 = \sum_{t=1}^{T} \lambda_t \cdot b_t,$$

and

$$\beta_i = \min\left(1, \max\left(0, \frac{\bar{x}_{r_i+1}(i) - x(i)}{\bar{x}_{r_i+1}(i) - \bar{x}_{r_i}(i)}\right)\right).$$

For computing β_i, we use the min and the max functions to cope with $x(i)$ oustside the interval $[\bar{x}_1(i), \bar{x}_M(i)]$.

Thus far, $G_T(x)$ has not been compressed. We still have to sort M elements for each dimension, and conduct binary search to locate r_i for (15). To compress the ensemble classifier and to bypass the binary search, we adopt the strategy from Maji *et al.* [23] and uniformly divide the interval $[\bar{x}_1(i), \bar{x}_M(i)]$ into q segments. With such an approximation, the linear interpolations are conducted by looking up a real-valued table with size of $d \times q$, instead of $d \times M$. It is in this manner that we compute the decision value for each dimension in constant time, independent of the number of meta classifiers and their support vectors. As a consequence, we reduce the complexity of scoring an example from $O(T \cdot |B_+| \cdot d)$ to $O(d)$. Now, computing (7) has an order of $O(d \cdot m)$ only.

We summarize the negative bootstrap algorithm in Table I. The algorithm has a number of desirable properties. By iteratively mining relevant negatives, we obtain visual classifiers with better discrimination ability. With model compression, we not only speed up the training process, but also make the time complexity of visual concept search independent of the number of meta classifiers.

IV. Experimental Setup

A. Data Sets

1) Pseudo Negative Examples: As an instantiation of $\mathcal{S}$, we use a set of 3.5M user-tagged images[1] randomly sampled from Flickr in our previous work [6]. Since batched-tagged pictures are often visually redundant, we remove them beforehand. We also exclude images whose social tags contain no visual concepts, as determining the negativeness of these examples is difficult. To that end, we create a vocabulary of 5K visual concepts by taking the intersection between the ImageNet vocabulary [9] and social tags used by over a hundred users. By removing images having no tags corresponding to the vocabulary, we obtain an $\mathcal{S}$ consisting of 610K images.

We evaluate negative bootstrap on two present-day benchmark sets [42], [43] which provide ground truth annotations for a diverse set of visual concepts including objects such as 'bicycle' and 'horse', scenes such as 'beach' and 'cityscape', and events such as 'dancing' and 'swimming'. Further, to test the effectiveness of the new algorithm for visual concept search in larger data, we create our third test set of one million images.

2) Two Benchmark Sets: VOC08-devel[2] [42] and NUS-WIDE[3] [43]. Both sets were collected from Flickr, with manually verified annotations for 20 and 81 visual concepts, respectively. The VOC08-devel set consists of the following two distinct subsets: VOC08train with 2,111 images and VOC08val with 2,221 images. We take the positive set $\mathcal{B}_+$ from VOC08train, and use VOC08val for testing. As the original NUS-WIDE training and testing sets were divided at random, they have many batch-tagged images from the same users, introducing a dependency between the two sets. To avoid such a dependency, we exclude batch-tagged images and images whose social tags do not overlap with the 81 concepts, resulting in a set of 128,097 images. We then divide this set into two subsets in terms of the Flickr DateUploaded property. Images in the resulting NUSpast (64,048 images) were uploaded before NUSfuture (64,049 images). This division improves generalizations of our findings to unseen data. We use NUSpast as another source of $\mathcal{B}_+$, and NUSfuture as our second test set.

3) A Test Set of 1M Images: This set consists of one million images collected from Flickr in a random fashion, independent of the training data and the other test data. Less than 0.2% of the 1M set appears in VOC08 and NUS-WIDE and 0.6% in $\mathcal{S}$. Removing the overlapped part does not affect the result.

B. Experiments

1) Experiment 1. Negative Bootstrap vs. State of the Art: We compare negative bootstrap with the following two state of the art algorithms, both of which rely on a form of random sampling to obtain negative examples: pure random sampling [1], [4], [5], [12], [13], and asymmetric bagging [16], [17]. For a fair comparison, whenever applicable we will make the three algorithms share the same input and parameters. As the number of negative examples and the number of iterations are the two parameters shared by the three algorithms, we let all the algorithms sample negative examples from the same pool, use the same amount of negatives to train a meta classifier, and run the same number of iterations. This experimental protocol allows us to conclude which algorithm yields the most relevant negatives.

By choosing $|\mathcal{B}_+|$ from $\{20, 100, 500\}$, we compare the algorithms given varying amounts of positive examples available. For concepts that have positive examples less than required, we use all the available positive examples.

To study whether adding relevant negatives is as important as adding new positives, we implement a strategy which keeps the negative set fixed, but randomly samples positives from the entire positive training data in each iteration to construct new meta classifiers. For a fair comparison between this 'sampling positives' strategy, asymmetric bagging, and negative bootstrap, we let them have the same 20 random positives and 20 random negatives as their starting point, and the same amount of positives and negatives hereafter.

2) Experiment 2. The Influence of Model Compression: By comparing negative bootstrap with and without model compression, we study the influence of model compression on both effectiveness and efficiency. The amount of positive examples for each concept is set to be 100. We report training and testing time in seconds, which are averaged over concepts.

3) Experiment 3. Negative Bootstrap for Concept Search in Large Data: Given the classifiers trained in Experiment 1 by negative bootstrap and asymmetric bagging respectively, we apply them for concept search in the 1M test set. We take the intersection between VOC and NUS-WIDE concepts as the query concepts: 'airplane', 'bird', 'boat', 'car', 'cat', 'cow', 'dog', 'horse', 'person', and 'train'. As there is no ground-truth available for the 1M set, we manually check for genuine positives in the top ranked images. To reduce the manual annotation effort and potential labeling bias towards certain runs, we employ a pooling mechanism similar to the TRECVid benchmark [44]. For each run, we put its top 20 ranked images into a common pool without indicating their origin. For a given concept, we label an image as positive if the concept is (partially) visible in the image. Artificial correspondences such as drawings, toys, and statues are labeled as negative.

Notice that for image search by visual classifiers, we deliberately treat the 1M test set as unlabeled. But because the test images are already associated with user tags, this allows us to compare the two algorithms further in a reranking scenario: applying the classifiers to rerank tag-based image search results. For the reranking experiment, we construct the initial tag-based search results using search-by-tag, which sorts images labeled with a target concept in descending order according to the time they were uploaded.

4) Experiment 4. Qualitative Analysis of Relevant Negatives: To gain a more intuitive understanding of negative bootstrap, we examine negatives which are most misclassified in each iteration. Recall that the negatives are associated with user tags. In order to quickly see which negative classes are most relevant to a given concept, we employ tag clouds to visualize the distribution of user tags in the selected negatives.

C. Implementations

To train visual classifiers, we use the popular bag of keypoints plus SVMs pipeline [37], [38], [45]. To extract bag of key-

[1]http://staff.science.uva.nl/~xirong/tagrel/

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/

[3]http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

points features, we choose dense sampling to locate keypoints and the SIFT as keypoint descriptors [37]. We create a codebook of 1,024 bins by running K-means clustering on SIFT descriptors extracted from a holdout set of random Flickr images. With the descriptors quantized by the codebook, each image is represented by a 1,024-dimensional l_1-normalized dense-SIFT histogram. We train two-class SVMs using LIBSVM [41]. For the parameter C, we empirically find that setting it to be 1 is a good choice.

To study the influence of different parameters on the effectiveness of negative bootstrap, we vary the size of the candidate set $\mathcal{U}_t$ by setting m to be fivefold, tenfold, and fifteenfold of $|\mathcal{B}_+|$, respectively. We choose the number of segments q from $\{50, 100\}$, and set the number of iterations T to 50. Our experiments show that the algorithm is robust to the parameter changes. Hence, unless specified, we use the following setting: $m = 10 \times |\mathcal{B}_+|$, $q = 100$, and $T = 50$.

For the implementation of asymmetric bagging, we follow [17] but use the full feature space rather than random subspaces, as studying random subspaces is beyond the scope of this paper. In the t-th iteration, random sampling selects $\mathcal{B}_-^{(t)}$ at random to train a classifier, while asymmetric bagging uniformly combines the classifier and $t-1$ classifiers generated in the previous rounds.

To reduce the chance of incorrectly selecting genuine positives for a given concept w, we remove images labeled with w or its semantically related tags [24]. We observe that if an image is labeled with visual concepts, but not labeled with w or its semantically related tags, the image tends to be a negative example of w. We implement the set of related tags as the union of childnodes of w in WordNet [46] and tags closest to w according to their Normalized Google Distance [47]. Notice that the tag reasoning is conducted in the tag space, rather than in the visual feature space where categorization is performed. Hence we obtain reliable negatives, amongst which we expect sufficient samples that are relevant for training classifiers.

1) Evaluation Criteria: We adopt Average Precision (AP), a common choice for evaluating visual search engines [36], [44]. As AP measures the ranking quality of the entire list while a user may be interested in the top ranked results, we report precision at 20 (P20) in addition to AP. The overall performance is averaged over the concepts.

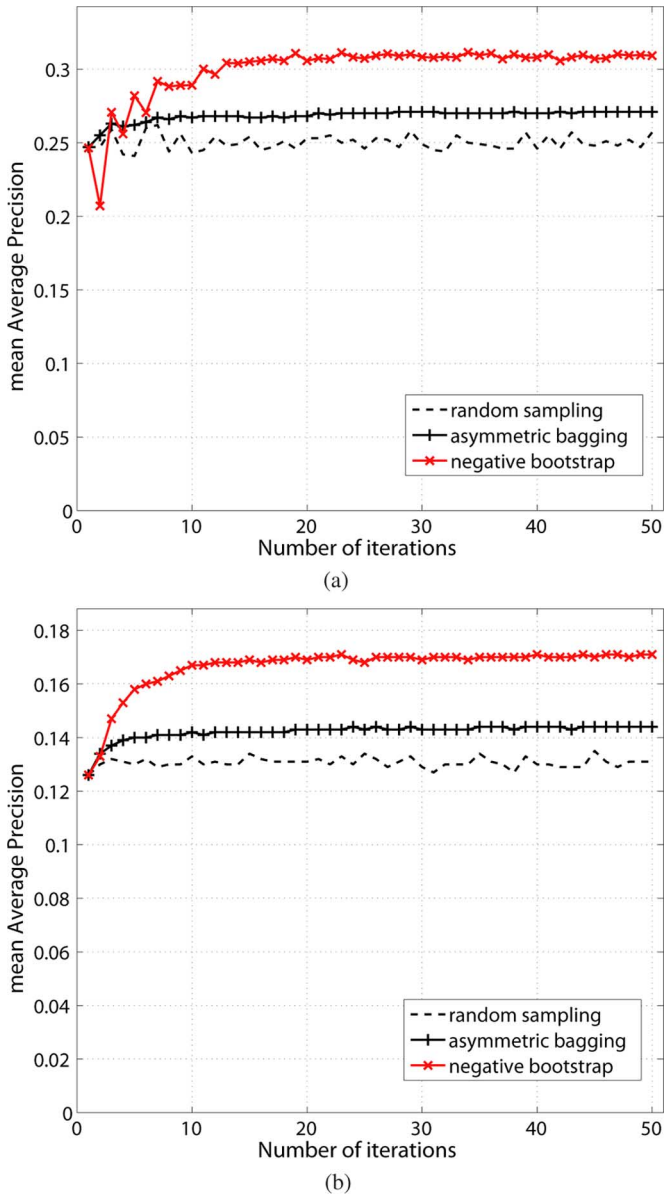

Fig. 2. Experiment 1. Negative bootstrap vs. state of the art. The number of positive training examples per concept is 100. Negative bootstrap outperforms both random sampling and asymmetric bagging. (a) Test set: VOC08val. (b) Test set: NUSfuture.

V. Results

A. Experiment 1. Negative Bootstrap Vs. State of the Art

As shown in Fig. 2, negative bootstrap compares favorably to random sampling and asymmetric bagging. In the first round, as no classifier is available, all algorithms start with the same negative set $\mathcal{B}_-^{(1)}$, and consequently produce the same classifier $G_1(x)$. Afterwards, while the baselines continue selecting random negatives, our algorithm starts to search for the most relevant negatives. The performance of random sampling varies due to the random factor in sampling. Asymmetric bagging reduces such variance by combining meta classifiers. Moreover, as the meta classifiers are trained on distinct negative sets, the combined classifier is more suited for test data of diverse content. Because of this, asymmetric bagging yields larger improvements on NUSfuture than VOC08val. The random sampling runs has an averaged mAP of 0.262 on VOC08val and 0.131 on NUSfuture. In comparison, asymmetric bagging obtains a relative improvement of 3% on VOC08val and 10% on NUSfuture in terms of mAP. Compared to asymmetric bagging, negative bootstrap obtains a relative gain of 14% on VOC08val and 18% on NUSfuture. Recall that all the three algorithms use the same positive data. The results allow us to conclude that negatives found by negative bootstrap are more relevant than randomly sampled negatives for learning visual concepts.

Further, to reveal whether the improvement is merely contributed by very few concepts, we make a concept-by-concept comparison, as shown in Fig. 3. Given the same amount of positive training data $\mathcal{B}_+$, negative bootstrap outperforms asymmetric bagging. More positive training data results in larger

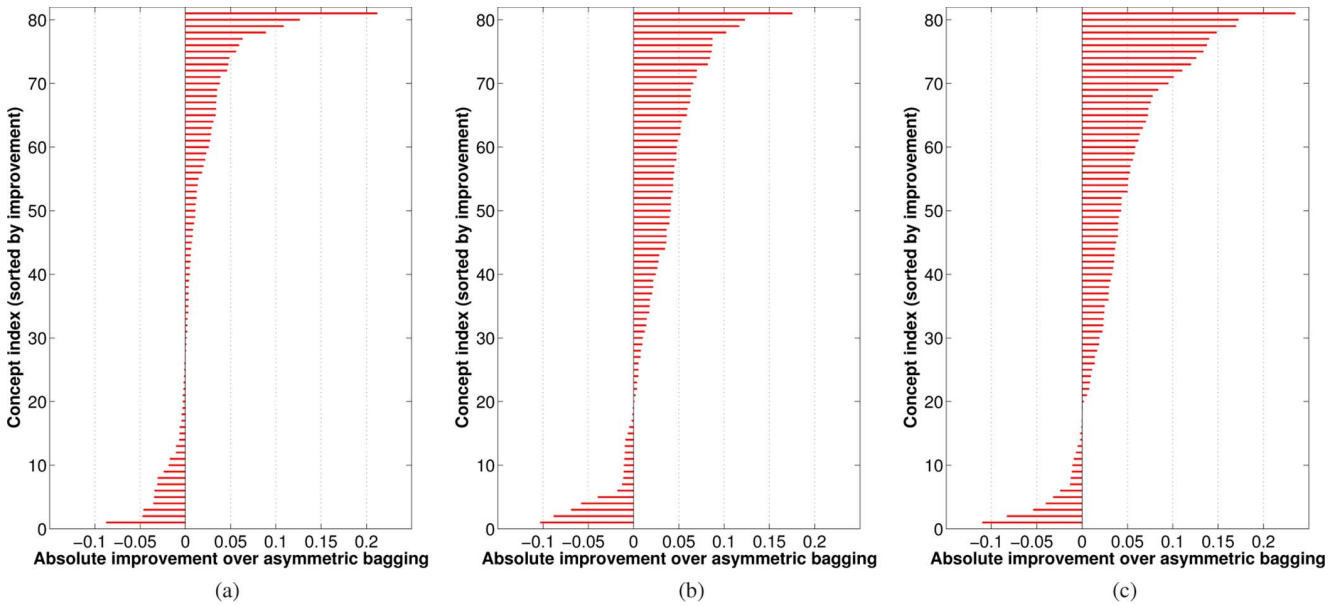

Fig. 3. Negative bootstrap vs. asymmetric bagging: a concept-by-concept comparison. We report results on the 81 concepts in NUSfuture, measured in terms of average precision. Falling at the left side of the reference line $x = 0$ means asymmetric bagging is better, while falling at the right side means negative bootstrap is better. Given varying amounts of positive training data, (a) 20 positive training examples, (b) 100 positives, and (c) 500 positives, negative bootstrap outperforms asymmetric bagging for all settings. Moreover, more positive training data results in larger improvements. (a) 20 positive examples. (b) 100 positive examples. (c) 500 positive examples.

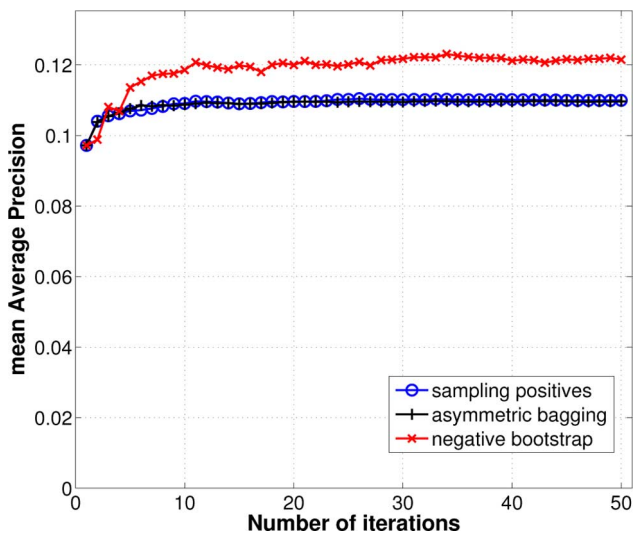

Fig. 4. Comparing different strategies for adding new training examples. Meta classifiers in each iteration are trained on 20 positives and 20 negatives. Testset: NUSfuture. Compared to adding new positive examples by 'sampling positives' and adding random negative examples by 'asymmetric bagging', adding relevant negatives by 'negative bootstrap' is most effective.

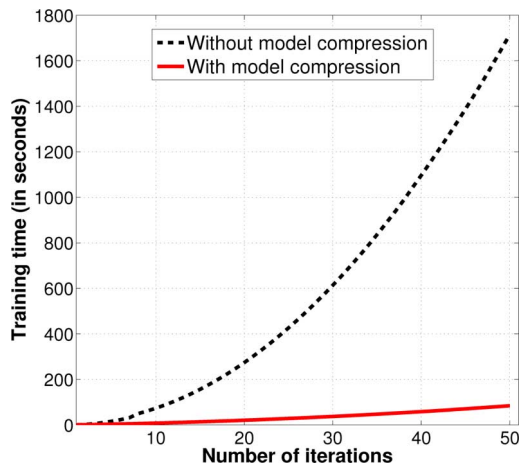

Fig. 5. Model compression accelerates negative bootstrap. Training time is measured throughout the negative bootstrap process, including meta classifier training, negative example selection, and model compression.

improvements. When only 20 positive examples are used for training, for 55 out of the 81 concepts in NUSfuture, negative bootstrap beats asymmetric bagging. When $|\mathcal{B}_+|$ increases to 100 and 500, the number of winning concepts increases to 62 and 64, respectively. These results show the viability of negative bootstrap.

The improvement over asymmetric bagging is obtained at the price of increasing training time. While it takes asymmetric bagging 18 seconds to train 50 meta classifiers, negative bootstrap with model compression costs 77 seconds.

TABLE II
EXPERIMENT 2. THE INFLUENCE OF MODEL COMPRESSION. THE PERFORMANCE OF NEGATIVE BOOTSTRAP *WITH* AND *WITHOUT* MODEL COMPRESSION ON THE TWO BENCHMARK SETS

	VOC08val		NUSfuture	
Metric	*without*	*with*	*without*	*with*
mean Average Precision	0.306	0.304	0.171	0.171
Precision at 20	0.503	0.512	0.436	0.443
Test time (seconds)	190	0.6	5,664	18

To further justify the necessity of classifier ensemble, we pool the negatives obtained from each iteration to learn just one classifier. To cope with class imbalance, the positive and negative classes are assigned with different cost factors according to the

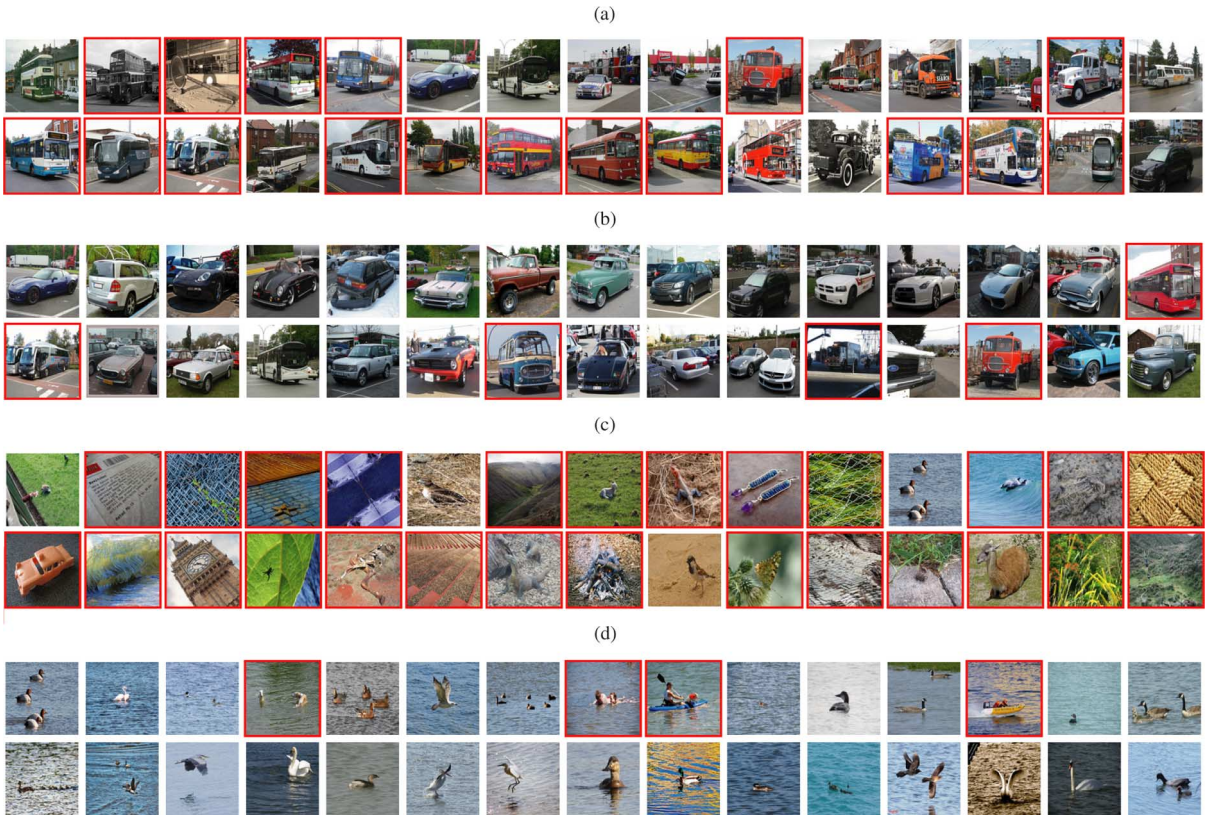

Fig. 6. Search for visual concepts in one million images by visual classifiers. The top 30 results are shown. Notice that the results are obtained using the visual classifiers alone, without taking user tags into account. A red border indicates a false positive result. Best viewed in color. (a) Searching for 'car' by asymmetric bagging. (b) Searching for 'car' by negative bootstrap. (c) Searching for 'bird' by asymmetric bagging. (d) Searching for 'bird' by negative bootstrap.

TABLE III
EXPERIMENT 3. NEGATIVE BOOTSTRAP FOR VISUAL CONCEPT SEARCH IN LARGE DATA, MEASURED IN TERMS OF PRECISION AT 20. IN THE COMPARISON BETWEEN NEGATIVE BOOTSTRAP AND ASYMMETRIC BAGGING, THE WINNER IS INDICATED BY A GRAY CELL

		Rerank by visual classifier		**Search by visual classifier**	
Concept	**Search by tag**	Asymmetric Bagging	*Negative Bootstrap*	Asymmetric Bagging	*Negative Bootstrap*
airplane	0.750	**1.000**	0.950	0.750	0.750
bird	1.000	0.950	0.950	0.150	**0.800**
boat	0.850	**1.000**	0.950	0.800	**0.950**
car	0.600	0.850	**1.000**	0.500	**0.900**
cat	0.650	0.900	**1.000**	0.200	**0.650**
cow	0.550	1.000	1.000	0.150	0.150
dog	0.900	1.000	1.000	0.250	**0.450**
horse	0.800	0.950	**1.000**	0.350	**0.650**
person	0.800	1.000	1.000	**1.000**	0.850
train	0.550	0.950	0.950	0.400	**0.850**
MEAN	0.745	0.960	**0.980**	0.455	**0.700**

reciprocal of their distribution in the training data. The single classifier with an mAP of 0.258 on VOC08val and 0.151 on NUSfuture is less effective than the ensemble.

As shown in Fig. 4, purely expanding the positive data is less effective than adding relevant negatives for defining a proper class boundary. As random negatives are often distant from the boundary, their gain is also relatively limited. It is clear from Fig. 4 that relevant negatives are the best, and require no new annotation. Negative bootstrap is thus attractive towards expansion to more and more concepts.

B. Experiment 2. The Influence of Model Compression

As shown in Fig. 5, the training time of negative bootstrap without model compression grows quadratically with respect to the number of iterations. In contrast, model compression reduces the training time from 1,736 seconds to 77 seconds. Using the compressed model, we also achieve faster concept search. As shown in Table II, to search for a specific concept on NUSfuture, applying the 50 meta classifiers takes 5,664 seconds. With model compression, we finish the search process in 18 seconds

Fig. 7. The 80 most relevant negative examples for a specific concept, found by negative bootstrap. By visualizing the distribution of user tags in the selected negatives as a tag cloud, we see which negative classes are most relevant to a given concept. (a) Relevant negatives of 'bear'. (b) Relevant negatives of 'car'. (c) Relevant negatives of 'window'. (d) Relevant negatives of 'moon'.

approximately. The advantage becomes more clear when we deal with larger collections. The result verifies the efficiency of model compression.

As shown in Table II and Fig. 2, negative bootstrap with model compression, reaching an mAP of 0.304 on VOC08val and 0.171 on NUSfuture, is as effective as negative bootstrap without model compression. For a more comprehensive comparison, we perform a paired t-test on NUSfuture, and obtain a p value of 0.90 for mAP and 0.38 for P20. Since the value is much larger than the standard 0.05 significance level, the performance difference between the two runs are not statistically significant. The results allow us to conclude that model compression substantially accelerates the negative bootstrap process, meanwhile the effectiveness of negative bootstrap is maintained.

Concerning the impact of different parameters on negative bootstrap, reducing q from 100 to 50 has a negligible impact on the performance. Because the memory footprint of compressed models is proportional to q, this result is attractive when we want to cache many classifiers into memory. The choice of m mainly affects the first few iterations. For instance, the performance curves with $m = 5 \times |\mathcal{B}_+|1$ dip at $T = 2$ (data not shown). This is because $g_2(x, \omega)$ is derived from $\mathcal{B}_-^{(2)}$, which are the most misclassified negatives by $G_1(x, \omega)$, and thus very distinct from generic negatives. Consequently, $g_2(x, \omega)$ is less effective for classifying generic negatives. Nevertheless, as subsequent clas-

sifiers are designed to be complementary to their ancestors, such ineffectiveness is tentative and resolved by adaptive sampling. We observe that the performance curves converge after 20 iterations. We recommend the following parameters: $m = 10 \times |\mathcal{B}_+|$, $q = 50$, and $T = 20$.

C. Experiment 3. Negative Bootstrap for Concept Search in Large Data

As shown in Table III, when user tags are available, the SearchByTag run has a P20 of 0.745. When reranking the tag-based search results by visual classifiers, both asymmetric bagging and negative bootstrap improve the performance, making P20 close to 1. In such a reranking scenario, negative bootstrap is slightly better than asymmetric bagging. When no user tags are given, as in a typical scenario of visual concept search in unlabeled data, classifiers trained on the relevant negatives are more accurate than classifiers trained by asymmetric bagging, with an absolute improvement of 0.245. For a more intuitive comparison, we show some image search results in Fig. 6. Because we continuously select the most relevant negatives, the ensemble classifier imposes a more tight boundary around the positive examples. As a consequence, classifiers trained on relevant negatives are more discriminative. In addition, with the compressed models on our machine it only takes approximately 6 minutes to scan 1M images per query, which would be over 20 hours without model compression.

D. Experiment 4. Qualitative Analysis of Relevant Negatives

As we use the Dense-SIFT feature, negative examples visually close to the positives in terms of their visual context are recognized as the most relevant negatives for training (see Fig. 7). For the concept 'car' as shown in Fig. 7(b), one might expect images of 'bus' in the relevant negative set as the two concepts often appear in a similar visual context, e.g., a street scene. Because the two concepts have a high co-occurrence in user tagging, 'bus' is treated as semantically related to 'car' by the tag reasoning strategy. Consequently, images labeled with 'bus' are automatically excluded by this strategy. As an alternative, negative bootstrap automatically finds 'firetruck' as the most relevant class. Since 'truck' and 'bus' have similar visual patterns, the 'car' classifier with 'firetruck' as negatives can still distinguish 'car' from 'bus'. The results show the merit of negative bootstrap, even when the selected negative classes might not be the first option in terms of a human's perception. For some concepts such as 'window', 'person', and 'grass', we find that asymmetric bagging performs better. This is largely due to the fact that these concepts are frequently present in the background and they are less labeled by user tagging. Because there is no manual verification in the negative bootstrap process, some genuine positives are incorrectly included in the negative set, as shown in Fig. 7(c). Nevertheless, for the majority of the concepts in consideration, we observe improvements. In addition, we have manually checked the error rate of the selected negatives for the 20 VOC concepts as shown in Fig. 8. Reliable negatives can be obtained at an averaged error rate of 0.042. In sum, the qualitative and quantitative results further demonstrate the effectiveness of negative bootstrap in finding relevant negatives for learning visual classifiers.

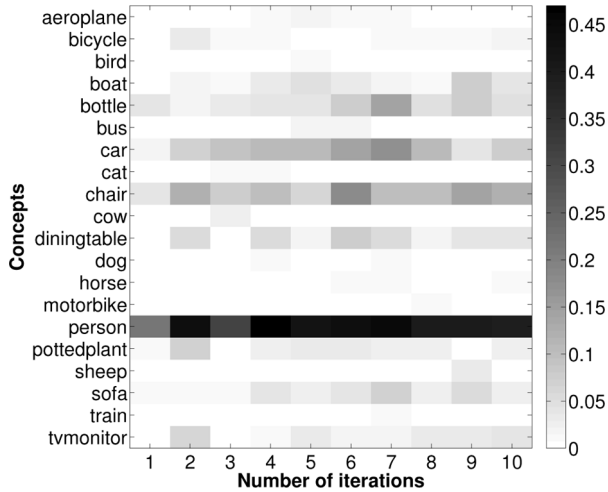

Fig. 8. Error rate of negative training examples selected by negative bootstrap. The error rate is the proportion of (manually checked) genuine positives in the negative training data. Despite the absence of manual verification, for the majority of concepts, reliable negatives can be obtained at an averaged error rate of 0.042.

VI. Summary and Conclusions

Given widely available user-tagged images online, in this paper we study which images are relevant negatives for learning visual concept classifiers. To that end, we propose Negative Bootstrap. Given a specific concept and a few positive examples, the new algorithm combines random sampling and adaptive selection to iteratively find relevant negatives. To address the inefficiency in applying ensemble classifiers, we introduce Model Compression to compress an ensemble of histogram intersection kernel SVMs. Consequently, the prediction time is independent of the size of the ensemble. To justify our proposals, we exploit 610K user-tagged images as pseudo negative examples, and conduct visual concept search experiments on two popular benchmark sets and a third test set of one million Flickr images.

The experimental results allow us to draw a number of conclusions. First, relevant negatives can be selected from those negatives which have the highest probability of being misclassified, but with no need of actually labeling any negative examples. Compared to classifiers trained on randomly sampled negative examples, classifiers derived from such relevant negative have better discrimination ability. Compared to asymmetric bagging, the new algorithm obtains a relative gain of 14% and 18% in terms of mean average precision on the two benchmarks. Second, the confinement of the class boundary to just the area of the class and nothing more by adding relevant negatives proves to be as important as the extension of the class boundary by adding new positives. Without introducing annotation overhead, relevant negatives are key to expansion to more classes. Third, model compression substantially accelerates training and testing, while at the same time the effectiveness of negative bootstrap is maintained. For visual concept search in the 1M set, model compression reduces the search time from tens of hours to just a few minutes.

As the results suggest, when combined with the latest achievements in obtaining positive examples, negative bootstrap brings learning thousands of visual concepts with good

discrimination ability within reach. What is more, model compression facilitates learning visual concepts on demand by classifier ensembles. Negative bootstrap opens up interesting avenues for future research.

REFERENCES

[1] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 754–766, Apr. 2011.

[2] K. Yanai and K. Barnard, "Probabilistic web image gathering," in *Proc. ACM MIR*, 2005, pp. 57–64.

[3] Y. Liu, D. Xu, I. Tsang, and J. Luo, "Textual query of personal photos facilitated by large-scale web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1022–1036, May 2011.

[4] L. Kennedy, S.-F. Chang, and I. Kozintsev, "To search or to label?: Predicting the performance of search-based automatic image classifiers," in *Proc. ACM MIR*, 2006, pp. 249–258.

[5] A. Ulges, C. Schulze, M. Koch, and T. Breuel, "Learning automatic concept detectors from online video," *Comput. Vis. Image Underst.*, vol. 114, no. 4, pp. 429–438, Apr. 2010.

[6] X. Li, C. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1310–1322, Nov. 2009.

[7] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 157–173, May 2008.

[8] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proc. SIGCHI*, 2004, pp. 319–326.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[10] D. Tax, "One-class classification," Ph.D. dissertation, Dept. Intell. Syst., Delft Univ. of Technology, Delft, The Netherlands, 2001.

[11] D. Tao, X. Li, and S. Maybank, "Negative samples analysis in relevance feedback," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 568–580, Apr. 2007.

[12] X. Li and C. Snoek, "Visual categorization with negative examples for free," in *Proc. ACM Multimedia*, 2009, pp. 661–664.

[13] S. Zhu, G. Wang, C.-W. Ngo, and Y.-G. Jiang, "On the sampling of web images for learning visual concept classifiers," in *Proc. CIVR*, 2010, pp. 50–57.

[14] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010.

[15] N. Chawla, L. Hall, K. Bowyer, and W. Kegelmeyer, "Learning ensembles from bites: A scalable and accurate approach," *J. Mach. Learn. Res.*, vol. 5, pp. 421–451, Dec. 2004.

[16] A. Natsev, M. Naphade, and J. Tešić, "Learning the semantics of multimedia queries and concepts from a small number of examples," in *Proc. ACM Multimedia*, 2005, pp. 598–607.

[17] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[18] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM Multimedia*, 2001, pp. 107–118.

[19] M. Wang and X.-S. Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, pp. 10:1–10:21, Feb. 2011.

[20] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.

[21] X. Tian, D. Tao, and Y. Rui, "Sparse transfer learning for interactive video search reranking," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 8, no. 3, pp. 26:1–26:19, Aug. 2012.

[22] J. Tang, Z.-J. Zha, D. Tao, and T.-S. Chua, "Semantic-gap-oriented active learning for multilabel image annotation," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2354–2360, Apr. 2012.

[23] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. CVPR*, 2008, pp. 1–8.

[24] X. Li, C. Snoek, M. Worring, and A. Smeulders, "Social negative bootstrapping for visual categorization," in *Proc. ICMR*, 2011, pp. 12–19.

[25] J. Yuan, J. Li, and B. Zhang, "Learning concepts from large scale imbalanced data sets using support cluster machines," in *Proc. ACM Multimedia*, 2006, pp. 441–450.

[26] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory under-sampling for class-imbalance learning," in *Proc. ICDM*, 2006, pp. 965–969.

[27] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[28] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. ICDM*, 2003, pp. 179–186.

[29] R. Yan, A. Hauptmann, and R. Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proc. ACM Multimedia*, 2003, pp. 343–346.

[30] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[31] S. Maji and A. Berg, "Max-margin additive classifiers for detection," in *Proc. ICCV*, 2009, pp. 40–47.

[32] G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from Flickr groups using fast kernel machines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2177–2188, Nov. 2012.

[33] J. Wu, "A fast dual method for HIK SVM learning," in *Proc. ECCV*, 2010, pp. 552–565.

[34] C. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retr.*, vol. 2, no. 4, pp. 215–322, 2009.

[35] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 2000.

[36] M. Everingham, L. van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

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

[38] Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. Hauptmann, "Representations of keypoint-based semantic concept detection: A comprehensive study," *IEEE Trans. Multimedia*, vol. 12, no. 1, pp. 42–53, Jan. 2010.

[39] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, Jun. 2007.

[40] J. Uijlings, A. Smeulders, and R. Scha, "Real-time visual concept classification," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 665–681, Nov. 2010.

[41] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[42] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2008 Results." [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html

[43] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. CIVR*, 2009, pp. 48–55.

[44] A. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proc. ACM MIR*, 2006, pp. 321–330.

[45] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop Stat. Learning Comput. Vis.*, 2004, pp. 1–22.

[46] , C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT, 1998.

[47] R. Cilibrasi and P. Vitanyi, "The Google similarity distance," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 370–383, Mar. 2004.

Xirong Li received the M.Sc. and B.Sc. degrees from the Tsinghua University, Beijing, China, in 2005 and 2007, respectively, and the Ph.D. degree from the University of Amsterdam, Amsterdam, The Netherlands, in 2012, all in computer science.

He is currently an Assistant Professor in the MOE Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China. His research interest is visual search in varied context.

Dr. Li received the 2012 IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award, the Best Paper Award of the CIVR 2010, and Best Paper Nominee of the ICMR 2012.

Cees G. M. Snoek received the M.Sc. degree in business information systems and the Ph.D. degree in computer science from the University of Amsterdam, Amsterdam, The Netherlands, in 2000 and 2005, respectively.

He is currently an Assistant Professor in the Intelligent Systems Lab at the University of Amsterdam. He was a visiting scientist at Carnegie Mellon University, Pittsburgh, PA (2003) and at the University of California, Berkeley, CA (2010–2011). His research interest is video and image search.

Dr. Snoek is the lead researcher of the MediaMill Semantic Video Search Engine, which is a consistent top performer in the yearly NIST TRECVID evaluations. He is a co-initiator and co-organizer of the VideOlympics, co-chair of the SPIE Multimedia Content Access conference, and member of the editorial boards for *IEEE Multimedia* and IEEE TRANSACTIONS ON MULTIMEDIA. Cees is recipient of an NWO Veni award (2008), a Fulbright Junior Scholarship (2010), an NWO Vidi award (2012), and the Netherlands Prize for ICT Research (2012). All for research excellence. Several of his Ph.D. students have won best paper awards, including the IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award.

Marcel Worring received the M.Sc. degree (honors) in computer science from the VU Amsterdam, Amsterdam, The Netherlands, in 1988 and the Ph.D. degree in computer science from the University of Amsterdam in 1993.

He is currently an Associate Professor in the Informatics Institute of the University of Amsterdam. His research focus is multimedia analytics, the integration of multimedia analysis, multimedia mining, information visualization, and multimedia interaction into a coherent framework yielding more than its constituent components. He has published over 100 scientific papers covering a broad range of topics from low-level image and video analysis up to multimedia analytics.

Dr. Worring was co-chair of the 2007 ACM International Conference on Image and Video Retrieval in Amsterdam, co-initiator and organizer of the VideOlympics, program chair for the ICMR 2013, and the ACM Multimedia 2013. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and of the *Pattern Analysis and Applications* journal.

Dennis Koelma received the M.Sc. and Ph.D. degrees in computer science from the University of Amsterdam in 1989 and 1996, respectively. The subject of his thesis is "A software environment for image interpretation".

He is currently a senior scientific programmer at the UvA. His research interests include image and video processing, software architectures, parallel programming, databases, graphical user interfaces, and visual information systems. He is the lead designer and developer of Impala: a software architecture for accessing the content of digital images and video. The software serves as a platform for consolidating software resulting from ISIS research. It has been licensed by the UvA spin-off Euvision where he has a part-time affiliation.

Arnold W. M. Smeulders is at the national research institute CWI in Amsterdam leading COMMIT, a nation-wide, very large public-private research program. He is also chair of IPN, the national policy committee for research in computer science. He is with the ISIS group at the University of Amsterdam for research in the theory and practice of visual search. He is co-owner of Euvision Technologies BV, a company spun off from the UvA.

Prof. Smeulders is an Associate Editor of the *IJCV*. He was recipient of a Fulbright fellowship at Yale University, and visiting professor in Hong Kong, Tuskuba, Modena and Cagliari. He is fellow of the International Association of Pattern Recognition, and honorary member of the Dutch Society for Pattern Recognition. He was general chairman of IEEE and ACM conferences on Multimedia.