IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

Local Features and a Two-layer Stacking Architecture for Semantic Concept Detection in Video

Foteini Markatopoulou, Vasileios Mezaris, *Senior Member, IEEE,* Nikiforos Pittaras, and Ioannis Patras, *Senior Member, IEEE*

Abstract—In this work we deal with the problem of extending and using different local descriptors, as well as exploiting concept correlations, towards improved video semantic concept detection. We examine how state-of-the-art binary local descriptors can facilitate concept detection, we propose color extensions of them inspired by previously proposed color extensions of SIFT, and we show that the latter color extension paradigm is generally applicable to both binary and non-binary local descriptors. In order to use them in conjunction with a state-of-the-art feature encoding, we compact the above color extensions using PCA and we compare two alternatives for doing this. Concerning the learning stage of concept detection, we perform a comparative study and propose an improved way of employing stacked models, which capture concept correlations, by using multi-label classification algorithms in the last layer of the stack. We examine and compare the effectiveness of the above algorithms in both semantic video indexing within a large video collection and in the somewhat different problem of individual video annotation with semantic concepts, on the extensive video dataset of the 2013 TRECVID Semantic Indexing Task. Several conclusions are drawn from these experiments on how to improve video semantic concept detection.

Index Terms—Content analysis and indexing, semantic concept detection, concept correlation, stacking, multi-label classification, video feature extraction, binary descriptors, semantic video annotation

1 Introduction

Semantic concept detection in video is the task of assigning one or more labels (semantic concepts) to a video sequence, based on a predefined concept list [1]. This is a very important task for the multimedia analysis field and a significant part of applications such as semantics-based video segmentation and retrieval, complex video event detection and recounting, video hyperlinking ([1], [2], [3], [4], [5]). A typical semantic concept detection system consists of three main modules (Fig. 1): the video decomposition module, where video sequences are segmented into shots and each shot is represented by e.g. one or more characteristic keyframes/images; the feature extraction module, where features (e.g. local image descriptors, motion descriptors) are extracted from the visual information and encoded into a descriptor vector; and finally the learning module, which employs machine learning algorithms, typically Support Vector Machines (SVM) or Logistic Regression (LR), in order to solve the problem of associating descriptor vectors and concept labels. Then, when a new unlabeled video shot arrives, the trained concept detectors will return confidence scores that show the belief of each detector that the corresponding concept appears in the shot. In this typical system, any existing semantic relations among concepts are not taken into account (e.g., the fact that *sun* and *sky* will often appear together in the same video shot). In this work we focus on two directions: Firstly, on feature-based video representation and secondly, on learning algorithms that exploit concept correlations.

On the front of feature extraction for video representation, Scale Invariant Feature Transform (SIFT) [6] and Speeded Up Robust Features (SURF) [7] are probably the two local descriptors that are most-widely used. However, they are non-binary descriptors, which makes them not so suitable for modern applications requiring the transmission of descriptor vectors. For example, when considering a mobile application where pictures are taken with a mobile device and local descriptors from these pictures need to be sent to a server for semantic analysis, then it is very important that the local descriptors are as compact as possible, to minimize transmission requirements [8]. ORB (Oriented FAST and Rotated BRIEF) [9] and BRISK (Binary Robust Invariant Scalable Keypoints) [10] are two binary local descriptors, which were originally proposed for similarity matching between local image patches. We examine ORB and BRISK in the task of video semantic concept detection, and we show that they constitute a viable alternative to the non-binary descriptors currently used in this

- *F. Markatopoulou is with the Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece and with the Queen Mary University of London, Mile end Campus, UK, E14NS. E-mail:markatopoulou@iti.gr,*
- *V. Mezaris and N. Pittaras are with the Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Thermi 57001, Greece. E-mail:{bmezaris, npittaras}@iti.gr,*
- *I. Patras is with the Queen Mary University of London, Mile end Campus, UK, E14NS E-mail: i.patras@qmul.ac.uk.*

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

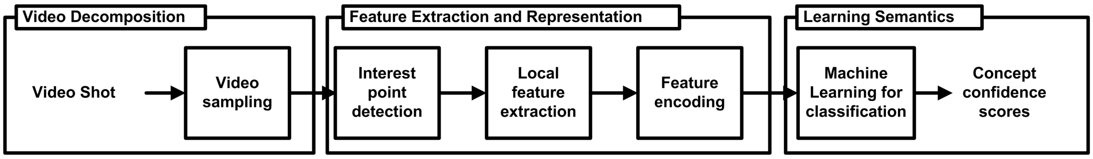

Fig. 1. Block diagram of a typical concept detection system

task, while their compact size and low storage needs make them appealing for mobile applications. Subsequently, inspired by two color extensions of SIFT [11], namely RGB-SIFT and OpponentSIFT, we define the corresponding color extensions for the three other local descriptors considered in this work (SURF, ORB, BRISK), and we show that this relatively straightforward way of introducing color information is in fact a generic methodology that works similarly well for different binary and non-binary local descriptors. In addition, we present a different way of performing Principal Component Analysis (PCA) [12] for feature reduction, which often improves the results of SIFT/SURF/ORB/BRISK color extensions when combined with Vector of Locally Aggregated Descriptors (VLAD) encoding [13].

On the machine learning front, the majority of concept detection systems learn supervised classifiers separately for each semantic concept. However, assigning concepts to video shots is by definition a multi-label classification problem, since multiple concepts may describe a single video shot. The simple process of training each concept detector independently is known as Binary Relevance (BR) transformation and is an elementary way of solving multi-label learning problems. One way of improving this baseline BR approach, is to consider concept correlations. A group of methods in this category follow a stacking architecture (e.g. [14], [15]). The predictions of multiple BR-trained concept detectors form model vectors that are used as a meta-learning training set for a second learning round (mainly by adopting a second round of BR models). In this work we examine the use of elaborate multi-label classification algorithms instead of BR models for the second-layer learning.

Another distinguishing feature of this work is the way that semantic concept detection is evaluated. A closer look at the literature shows that researchers focus on evaluating concept detection in a semantics-based indexing and retrieval setting, i.e. given a concept, measure how well the top retrieved video shots for this concept truly relate to it. However, besides the retrieval problem, another important problem related to semantics-based video manipulation is the annotation problem, i.e. the problem of estimating which concepts best describe a given video shot. We report evaluation results in both directions and compare them.

The rest of this paper is organized as follows: Section 2 reviews related work, focusing on local image descriptors and learning methods that exploit concept correlations. Section 3 examines how two binary descriptors can be used for video concept detection, introduces the color extensions of SURF, ORB and BRISK and discusses a different way of employing PCA for color descriptors. Section 4 presents the proposed stacking architecture for exploiting concept correlations and multi-label learning algorithms that are suitable for instantiating this architecture. Section 5 reports our experiments and results, and finally Section 6 summarizes our main conclusions.

2 Related Work

2.1 Features for Concept Detectors

A variety of visual, textual and audio features can be extracted to represent each piece of visual information; a review of different types of features can be found in [1]. In large-scale video concept detection, typically local image features are utilized, being extracted from representative keyframes or similar 2D image structures [16]. Two of the most popular local descriptors are SIFT [6] and SURF [7]. Both of them extract features that are invariant to rotation, scale and illumination variations, while SURF extraction is somewhat less computationally-demanding (SURF is two times faster than SIFT according to [7]). SIFT and SURF construct vectors of floating-point values (which are often quantized to integers in the range [0,255]). For many modern applications, though, e.g. concept detection on mobile devices, small-sized yet discriminative descriptors are very important in order to extract, store and transmit them efficiently (e.g. send local descriptors to a server for performing concept detection). Binary local descriptors are an attractive alternative to non-binary descriptors such as SIFT and SURF, generating binary strings which can be computed efficiently while also requiring lower storage space. ORB [9], BRISK [10], and FREAK [17] are some examples of binary local descriptors that have been proposed for similarity matching between local image patches. They are all based on calculating the differences between pairs of pixel intensity values within an image patch; what distinguishes them is the pattern they follow in order to perform these pair-wise pixel comparisons. Studies show that ORB [9] and BRISK [10] are among the most accurate binary descriptors for image matching [18]. The possibility of using ORB in image classification was also briefly examined in [19].

The above mentioned non-binary and binary local descriptors are intensity-based: they are applied to grayscale images (e.g. an RGB image is firstly converted to grayscale), and the extracted features are calculated from the pixel intensity values. Two color variants of

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

SIFT, namely RGB-SIFT and OpponentSIFT, that increase the descriptor's discriminative power were proposed in [11]. Methods that consider the color information in order to improve the SURF descriptor have also been proposed. Most of them were examined only on the image matching problem [20], [21], [22], while others, such as OpponentSURF and similar extensions of other descriptors, have also been used for concept detection [23], [24]. In [19], the extraction of ORB from all three color channels of the RGB color space was considered.

For the purpose of visual concept detection, local descriptors extracted from different patches of one image are subsequently aggregated into a global image representation, a process known as feature encoding. The most popular encoding in the last years has been the Bag-of-Words (BoW) [25]. Fisher vector (FV) [26] and VLAD (Vector of Locally Aggregated Descriptors) [13] are two state-of-the-art encodings that significantly outperform the BoW [27] [28]. FV encoding describes the difference between the distribution of features for an image and the distribution fitted to the features of all the training data. VLAD [13] is a fast approximation of FV that performs somewhat worse but is more compact and faster to compute [29], which makes it a good compromise. The two latter encodings are high-dimensional and their dimensionality is affected by the dimensionality of the local descriptors they encode, thus dimensionality reduction approaches such as PCA [12] are widely used for making the image representation more compact prior to learning/classification. Dimensionality reduction can be performed at two stages: local descriptors can be reduced prior to the encoding, and then the final encoding can also be further compacted [29].

2.2 Exploiting Concept Correlations

Associating feature-based image representations with semantic concepts is performed using machine learning algorithms. In order to do this effectively it is useful to take advantage of concept correlation. Concept correlation refers to the relations among concepts within a video shot. By using this information we can refine the predictions derived from multiple independent concept detectors in order to improve their accuracy, a process known as Context Based Concept Fusion (CBCF) [15]. Two main types of methods have been adopted in the literature for this: a) Stacking-based approaches that collect the scores produced by a baseline set of concept detectors and introduce a second learning step in order to refine them, b) Inner-learning approaches that follow a single-step learning process, which jointly considers low-level visual features and concept correlation information [1], [30], [31].

In this work we focus on the first category. Stacking approaches aim to detect dependencies among concepts in the last layer of the stack. One popular group is the BR-based stacking approaches. For example, Discriminative Model Fusion (DMF) [14] obtains concept score predictions from the individual (BR-trained) concept detectors in the first layer, in order to create a *model vector* for each shot. These vectors form a meta-level training set, which is used to train a second layer of BR models. Correlation-Based Pruning of Stacked Binary Relevance Models (BSBRM) [32] extends the previous approach by pruning the predictions of non-correlated concept detectors before the training of each individual classifier of the second-layer BR models. Similarly to DMF, the Baseline CBCF (BCBCF) [15] forms model vectors, in this case using the ground truth annotation, in order to train second-layer BR models. Furthermore, the authors of [15] note that not all concepts can take advantage of CBCF, so their method refines only a subset of them. Another group of stacking approaches are the graph-based ones, which model label correlations explicitly [1]. The Multi-Cue Fusion (MCF) method [33] uses the ground truth annotation to build decision trees that describe the relations among concepts, separately for each concept. Then, the initial concept detection scores are refined by approximating these graphs.

Inner-learning approaches, on the other hand, make use of contextual information from the beginning of the concept learning process. For example, the authors of [30] propose methods that simultaneously learn the relation between visual features and concepts and also the correlations among concepts. However, inner-learning approaches suffer of computational complexity. For example, [30] has complexity at least quadratic to the number of concepts, making it inapplicable to real problems where the number of concepts is large (e.g. hundreds).

In the TRECVID semantic indexing benchmarking activity several teams study label correlations. The Concept Association Network [34], which is a rule-based system, and other systems that aim to take advantage of "imply" and "exclude" relations between concepts [35], [36], are some examples that explicitly study label correlations. However, we did not consider such methods in the present work, because in the TRECVID experiments these methods did not exhibit a significant improvement in the goodness of concept detection, compared to the BR baseline. An extension of the DMF approach, namely *conceptual feedback* (CF), that implicitly captures label correlations and improve the scores derived from multiple independent detectors was proposed [37]. Concept detectors (e.g. a DMF model) are built from the normalized scores of the first-layer independent detectors. The resulting concept detectors are combined with the first layer detectors (e.g. by averaging their predictions) and the new detection scores are again used to build new concept detectors (e.g. a new DMF model) that capture concept correlations. This process can be iterated many times.

Label correlation has also been investigated in the broader multi-label learning domain. In [38], multi-label classification methods, including methods that consider contextual relations, are compared on multimedia data. In [31] such methods are adapted for concept detection.

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

Nevertheless, none of these approaches considers the use of multi-label classification methods as part of a stacking architecture. The latter is the focus of Section 4 of this work, where, extending our previous study on this topic [39], we describe and evaluate the use of such methods for building models in the second-layer of the stacking architecture that learns the correlations among labels.

3 Building Independent Concept Detectors

In this section we present how different local descriptors can be extended and used for building effective independent concept detectors. The detectors can be used as stand alone classifiers or alternatively as part of a stacking architecture. An earlier version of our work on local descriptors also appears in [40].

3.1 Using a Binary Local Descriptor for Concept Detection

ORB [9] and BRISK [10] are two binary local image detectors and descriptors that present similar discriminative power with SIFT and SURF in image matching problems, they have similar properties such as invariance in rotation, scale and illumination, but at the same time are more compact and faster to be computed. A 256-element binary ORB vector requires 256 bits to be stored (similarly a 512-element binary BRISK vector requires 512 bits); in contrast, an integer-quantized 128-element SIFT vector requires 1024 bits. In addition, according to [9] and [10], ORB and BRISK are an order of magnitude faster than SURF to compute, which in turn is faster than SIFT.

There is not a single way for introducing binary descriptors in the visual concept detection pipeline. [19] did so by considering the BoW encoding, and proposed a modified K-means algorithm (the "K-majority" algorithm) for generating the codebook (vocabulary) of BoW, that would result in a binary codebook.

In this work we claim that binary descriptors (ORB, BRISK) can be used for video concept detection in the same way as their non-binary counterparts. Specifically, let us assume that I is a set of images and x_i $i = 1, ..., N$ are ORB or BRISK descriptors extracted from I, where $x_i \in \{0, 1\}^d$. N is the total number of extracted local descriptors and d is their dimension. From these binary descriptors, we generate a floating-point codebook of K visual codewords $w_k \in \mathbb{R}^d$, $k = 1, ..., K$, using a standard K-means. The distances between the binary ORB/BRISK descriptors and the codewords are calculated by the L2 norm. The update of the cluster centres is also performed as in the original K-means (calculating the mean of a set of vectors). We compare these two codebook creation strategies (that of [19] and the one described in this section) in Section 5.

3.2 Color Extensions of Binary and Non-binary Local Descriptors

Based on the good results of two color extensions of SIFT, namely RGB-SIFT and OpponentSIFT [11], we examine the impact of using the same methodology for introducing color information to other descriptors (SURF, ORB, BRISK). Our objective is to examine if this is a methodology that can benefit different local descriptors and is therefore generally applicable.

Let d denote the dimension of the original local descriptor (typically, d will be equal to 64 or 128 for SURF, 128 or 256 for ORB and 512 for BRISK). This section summarizes the process of extracting RGB-SURF, RGB-ORB, RGB-BRISK, OpponentSURF, OpponentORB and OpponentBRISK descriptors. An RGB image has three 8-bit channels (for red, green and blue). The original non-color local descriptors are calculated on 8-bit grayscale images, so they first transform the RGB image to grayscale. In contrast to this, our RGB-SURF/ORB/BRISK apply the corresponding original descriptor directly to each of the three R, G, B channels and for each keypoint extract three d-element feature vectors. These are finally concatenated into one $3 \cdot d$-element feature vector, which is the RGB-SURF, RGB-ORB or RGB-BRISK descriptor vector.

Similarly, our OpponentSURF/ORB/BRISK descriptors firstly transform the initial RGB image to the opponent color space [11]. We refer to the transformed channels as O_1, O_2 and O_3. O_3 is the luminance channel, i.e. the one that the original SURF/ORB/BRISK descriptors use, while the other two channels (O_1 and O_2) capture the color information. Following the transformation, a normalization step that converts the ranges of each channel within the [0,255] range is employed, as in [11]. Then, similarly to RGB-SURF/ORB/BRISK, the original SURF, ORB or BRISK descriptor is applied separately to each transformed channel and the final $3 \cdot d$-element feature vector is the concatenation of the three feature vectors extracted from the three channels.

3.3 Reducing the Dimensionality of Local Color Descriptors

State-of-the-art local descriptor encoding methods generate high-dimensional vectors that make difficult the training of machine learning algorithms. For example, while the BoW model generates a k-element feature vector, where k equals to the number of visual words, VLAD encoding generates a $k \cdot l$-element feature vector (where l is the dimension of the local descriptor; in the case of the color extensions of descriptors discussed in the previous section, $l = 3 \cdot d$). Thus, it is common to employ dimensionality reduction before the construction of VLAD vectors, on local descriptors, mainly using PCA [12]. In this section we explain that directly applying PCA to the full vector of color descriptors, as implied from previously published works (e.g. [28]; termed "typical-PCA" in the sequel), is not the only possible solution, and we propose a simple modification of this descriptor dimensionality reduction process that it experimentally shown to improve the concept detection results in several cases.

PCA projects linearly l-dimensional features to a lower-dimensional feature space. Given a matrix A with dimension $l \times n$, where n is the number of observations (i.e., of keyframes in a video training dataset), if we want to perform dimensionality reduction (from l to l') with PCA, the reduced matrix A' will be $A' = E^T \cdot A$, where E is the projection matrix (of dimension $l \times l'$) and T denotes the transpose of a matrix.

PCA aims to find those directions in the data space that present high variance. When PCA is applied directly to the entire vector of one of the color extensions of (binary or non-binary) local descriptors, if one or two of the three color channels of the descriptor exhibit lower diversity than the others, then these risk being under-represented in the reduced dimensionality space. To avoid this, we propose performing PCA separately for each color channel and consider an equal number of principal components from each of them, to create three projection matrices that correspond to each of the three channels (termed "channel-PCA" in the sequel), instead of one projection matrix that corresponds to the complete descriptor vector. The three reduced single-channel descriptor vectors that can be obtained for a color descriptor using the aforementioned projection matrices are finally concatenated in a reduced color-descriptor vector.

4 Stacking for Exploiting Concept Correlations

Having presented our methods for video representation in the previous section, this section deals with learning the mappings between such representations and semantic concepts. Assuming that we first train a set of SVM-based or LR-based independent concept detectors (which is a typical approach in the literature), we propose an improved way of subsequently employing stacked models, by using multi-label classification methods in the last layer of the stack.

4.1 Proposed Stacking Architecture

Let $D_1, ..., D_N$ denote a set of N trained independent concept detectors on N different concepts. Let T denote a validation set of video shots, which will be used for training the second layer of the stacking architecture, and m denote the model vector of a new unlabeled video shot. Fig. 2 summarizes the full pipeline from training the second-layer classifiers to using them for classifying an unlabeled sample when using: (1) the BR stacking architecture (Fig. 2(b),(d)), and (2) the proposed stacking architecture (Fig. 2(c),(e)). Both architectures use exactly the same strategy to create the meta-level training set; the trained BR models ($D_1, ..., D_N$) of the first layer are applied to the validation dataset T and in this way a model vector set M is created, consisting of the scores that each of $D_1, ..., D_N$ has assigned to each video shot of T for every concept (Fig. 2(a)). What distinguishes the two architectures is the way that this meta-learning information is used and therefore the way that the second-layer learning is performed.

During the training phase, the BR stacking architecture builds a new set of BR models ($D'_1, ..., D'_N$). To train each model, a different subset of M that is ground-truth annotated for the corresponding concept C_n that the meta-concept detector D'_n will be trained for, is used (Fig. 2(b)). In contrast, the proposed architecture uses the whole model vector set and ground truth annotation at once in order to train a single multi-label classification model D', instead of separate models $D'_1, ..., D'_N$ (Fig. 2(c)).

During classification, a new unlabeled video shot is given to the first layer BR models ($D_1, ..., D_N$) and a model vector m is returned. Then on the one hand, the BR stacking architecture lets the $D'_1, ..., D'_N$ models to classify m and one score is returned separately from each (Fig. 2(d)). On the other hand, the proposed architecture uses the single trained model D' in order to return a final score vector (Fig. 2(e)).

With respect to learning concept correlations, the BR-based stacking methods learn them only by using the meta-level feature space. However, the learning of each concept is still independent of the learning of the rest of the concepts. The rationale behind us proposing the use of other multi-label learning algorithms in replacement of the BR models at the second layer of the stacking architecture is based on the assumption that if we choose algorithms that explicitly consider label relationships as part of the second-layer training, improved detection can be achieved. Our stacking architecture learns concept correlations in the last layer of the stack both from the outputs of first-layer concept detectors and by modelling correlations directly from the ground-truth annotation of the meta-level training set. This is achieved by instantiating our architecture in our experiments with different second-layer algorithms that model:

- Correlations between pairs of concepts;
- Correlations among sets of more than two concepts;
- Multiple correlations in the neighbourhood of each testing instance.

4.2 Learning Algorithms for Stacking

To model the correlation information described above, we exploit methods from the multi-label learning field [41], [42], [43], [44]. Pairwise methods can consider pairwise relations among labels; similar to the multi-class problem, one versus one models are trained and a voting strategy is adopted in order to decide for the final classification. In this category we choose the Calibrated Label Ranking (CLR) algorithm [41] that combines pairwise and BR learning. Label powerset (LP) methods search for subsets of labels that appear together in the training set and consider each set as a separate class in order

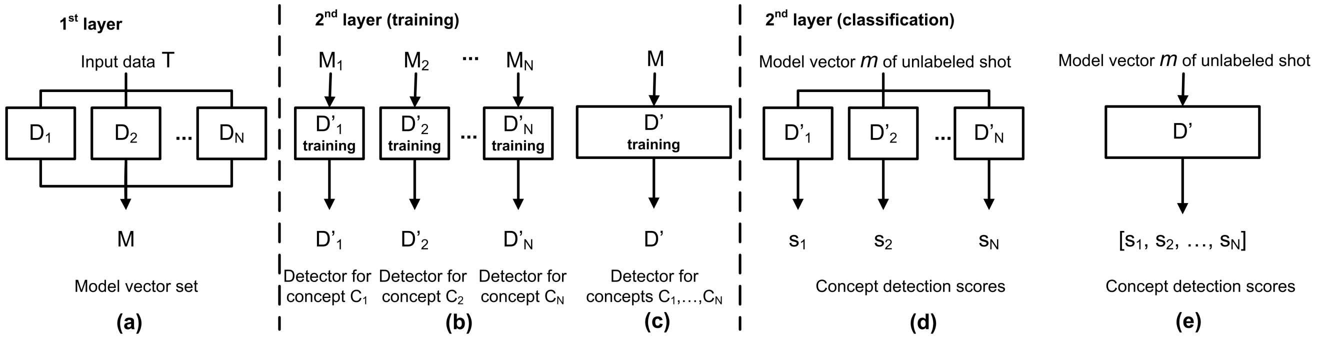

Fig. 2. Comparing BR and the proposed stacking architecture. (a) First layer of a stacking architecture. (b) Training of the second layer of a BR-stacking architecture. (c) Training of the second layer of the proposed stacking architecture. (d) Classification phase of the BR stacking architecture. (e) Classification phase of the proposed architecture.

to solve a multi-class problem. We choose the original LP tranformation [43], as well as the Pruned Problem Transformation algorithm (PPT) [42] that reduces the class imbalance problem by pruning label sets that occur less than l times. Finally, lazy style methods most often use label correlations in the neighbourhood of the tested instance, to infer posterior probabilities. In this direction we choose ML-kNN algorithm [44], which models exactly this information. In selecting the above methods, we took into account the computational complexity of these and other similar methods and tried to avoid using particularly computationally expensive ones.

The use of multi-label classification algorithms as the second layer of a stacking architecture has the significant advantage of allowing the representation of the videos using state-of-the-art high dimensional low-level features (for describing the video at the first layer of the stack), as opposed to simpler features used in e.g. [38], [31], while at the same time keeping relatively low the dimensionality of the input to the multi-label classifier of the second layer, thus making the overall concept detection architecture applicable even to large-scale problems.

5 Experimental Results

5.1 Datasets and Evaluation Methodology

Our experiments were performed on the TRECVID 2013 Semantic Indexing (SIN) dataset [45], which consists of a development set and a test set (approx. 800 and 200 hours of internet archive videos, comprising more than 500000 and 112677 shots, respectively). The development set is ground-truth annotated with 346 concepts. We further used the TRECVID 2012 test set (approx. 200 hours; 145634 shots), which is a subset of the 2013 development set, as a validation set to train algorithms for the second layer of the stack. We evaluated all techniques on the 2013 test set, for the 38 concepts for which NIST provided ground truth annotations [45].

As discussed in the Introduction, we want to examine the performance of the different methods both on the video indexing and on the video annotation problem. Based on this, we adopt two evaluation strategies: i) Considering the video indexing problem, given a concept, we measure how well the top retrieved video shots for this concept truly relate to it. ii) Considering the video annotation problem, given a video shot, we measure how well the top retrieved concepts describe it. For the indexing problem we calculated the Mean Extended Inferred Average Precision (MXinfAP) at depth 2000 [46], which is an approximation of the Mean Average Precision (MAP) that has been adopted by TRECVID [45]. For the annotation problem we calculate the Mean Average Precision at depth 3 (MAP@3). In the latter case, our evaluation was performed on shots that are annotated with at least one concept in the ground truth.

5.2 Experimental Setup

For experimenting with all first layer methods, one keyframe was initially extracted for each video shot and was scaled to 320 × 240 pixels prior to feature extraction. For some of our final experiments, we also extracted two visual tomographs [16] from each shot. Regarding feature extraction, we followed the experimental setup of [28] and we used the toolbox that its authors have published. More specifically, we used the dense SIFT descriptor, that accelerates the original SIFT descriptor, in combination with the Pyramid Histogram Of visual Words (PHOW) approach [47]. For SURF, ORB and BRISK we used their implementations included in OpenCV, and further extended these implementations with the corresponding color variants that we introduced in Section 3.2. The same square regions at different scale levels of the PHOW approach were used as the image patches that were described by SURF, ORB and BRISK. We calculated 128-SIFT, 128-SURF, 256-ORB and 512-BRISK grayscale descriptors; then, each color extension of a descriptor resulted in a color descriptor vector three times larger than that of the corresponding original descriptor, as explained in Section 3.2. All the non-binary local descriptors (SIFT, SURF and their color extensions) were compacted to 80 dimensions, using PCA, following the recommendations of [28] and [29]. Since there is no previous research on the influence of dimensionality

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

reduction on binary descriptors when they are used for semantic concept detection, ORB, BRISK and their color extensions were compacted both to 80 and to 256 dimensions (the latter is the original size of ORB), in order to investigate this. All the compacted local descriptors (binary and non-binary) were subsequently aggregated using the VLAD encoding. Similarly with the authors of [28], we divided each image into the same 8 regions using spatial binning and we used sum pooling to combine the encodings from different regions. As a result of the above process, a VLAD vector of 163840 elements for descriptors compacted to 80 dimensions and of 524288 elements for descriptors compacted to 256 dimensions was extracted for each image (by image we mean here either a keyframe or a visual tomograph). These VLAD vectors were compressed into 4000-element vectors by applying a modification of the random projection matrix [48]. The reduced VLAD vectors were L2 normalized according to [28] and served as input to the Logistic Regression (LR) classifiers that we used. Following the *cross validated committees* methodology of [2], we trained five LR classifiers per concept and per local descriptor (SIFT, ORB, RGB-ORB etc.), and combined the output of these five by means of late fusion (averaging). When different descriptors were combined, again late fusion was performed by averaging the classifier output scores. In all cases, the final step of concept detection was to refine the calculated detection scores by employing the re-ranking method of [49].

We instantiate the second layer of the proposed architecture with four different multi-label learning algorithms as described in Section 4.2, and will refer to our framework as P-CLR , P-LP, P-PPT and P-MLkNN when instantiated with CLR [41], LP [43], PPT [42] and ML-kNN [44] respectively. The value of l for P-PPT was set to 30. We compare these instantiations of the proposed framework against BCBCF [15], DMF [14], BSBRM [32], MCF [33] and CF [37]. For BCBCF we use the concept predictions instead of the ground truth in order to form the meta-learning dataset, as this was shown to improve its performance in our experiments; we refer to this method as CBCFpred in the sequel. Regarding the concept selection step we use these parameters: λ = 0.5, θ = 0.6, η = 0.2, γ = the mean of Mutual Information values. For MCF we only use the spatial cue, so temporal weights have been set to zero. The ϕ coefficient threshold, used by BSBRM, was set to 0.09. Finally, for CF we performed two iterations without temporal re-scoring (TRS). We avoided using TRS in order to make this method comparable to the others. For implementing the above techniques, the WEKA [12] and MULAN [50] machine learning libraries were used as the source of single-class and multi-label learning algorithms, respectively.

5.3 Independent Detector Results

We start by assessing the performance of detectors in relation to the indexing problem. In Table 1 we compare the performance of the original grayscale ORB descriptor in concept detection, when used in conjunction with a binary codebook (as in [19]) and a floating-point one (as in Section 3.1). In both cases, VLAD encoding is employed. We can see that the binary codebook proves ineffective; the floating-point one outperforms it by more than 129%. It should be noted that the MXinfAP of random classification is <0.1%, which indicates the difficulty of the problem. Based on this result, in all subsequent experiments with ORB, BRISK and their color extensions a floating-point codebook was used.

In Table 2 we evaluate the different local descriptors and their color extensions considered in this work, as well as combinations of them. First, comparing the original ORB and BRISK descriptors with the non-binary ones (SIFT, SURF), we can see that binary descriptors perform a bit worse than their non-binary counterparts but still reasonably well. This satisfactory performance is achieved despite ORB, BRISK and their extensions being much more compact than SIFT and SURF, as seen in the second column of Table 2. Second, concerning the methodology for introducing color information to local descriptors, we can see that the combination of the original SIFT descriptor and the two known color SIFT variants that we examine ("All SIFT" in Table 2) outperforms the original SIFT descriptor alone by 34.4% (35.3% for channel PCA). The similar combination of the SURF color variants with the original SURF descriptor, is shown in Table 2 to outperform the original SURF by 32.2% (which increases to 32.7% for channel-PCA), and even more pronounced improvements are observed for ORB and BRISK. These results show that this relatively straightforward way for introducing color information is in fact generally applicable to heterogeneous local descriptors.

We also compare the performance of each binary descriptor when it is reduced to 256 and to 80 dimensions. Reducing ORB and its color variants to 80 dimensions and combining them performs better than reducing them to 256 dimensions (both when applying typical- and channel-PCA). On the other hand, reducing BRISK and its two color variants to 256 dimensions and combining them gave the best results (in combination with channel-PCA).

To analyse the influence of PCA on the vectors of local color descriptors, in Table 2 we also compare the channel-PCA of Section 3.3 with the typical approach of applying PCA directly on the entire color descriptor vector. In both cases PCA was applied before the VLAD encoding, and in applying channel-PCA we kept the same number of principal components from each color channel (e.g. for RGB-SIFT, which is reduced to $l' = 80$ using typical-PCA, we set $p_1 = p_2 = 27$ for the first two channels and $p_3 = 26$ for the third color channel; $p_1+p_2+p_3 = l'$). According to the relative improvement figures reported in the fifth column of Table 2 (i.e., for the indexing problem), performing the proposed channel-PCA in most cases improves the concept detection re-

TABLE 1
Performance (MXinfAP, %, and MAP@3, %) for ORB, when the binary codebook proposed in [19] and when a floating-point codebook is used. In parenthesis we show the relative improvement w.r.t. the binary codebook.

	MXinfAP (indexing)		MAP@3 (annotation)	
Descriptor	Binary codebook [19]	Floating-point codebook	Binary codebook [19]	Floating-point codebook
ORB	4.52	10.36 (+129.2%)	66.85	71.05 (+6.3%)

TABLE 2
Performance (MXinfAP, %, and MAP@3, %) for the different descriptors and their combinations, when typical and channel-PCA is used for dimensionality reduction. In parenthesis we show the relative improvement w.r.t. the corresponding original grayscale local descriptor for each of the SIFT, SURF, ORB, BRISK color variants.

		MXinfAP (indexing)			MAP@3 (annotation)		
Descriptor	Descriptor size in bits	Keyframes, typical-PCA	Keyframes, channel-PCA	Boost(%) w.r.t typical-PCA	Keyframes, typical-PCA	Keyframes, channel-PCA	Boost(%) w.r.t typical-PCA
SIFT	1024	14.22	14.22	-	74.32	74.32	-
RGB-SIFT	3072	14.97 (+5.3%)	14.5 (+2.0%)	-3.1%	74.67 (+0.5%)	74.07 (-0.3%)	-0.8%
OpponentSIFT	3072	14.23 (+0.1%)	14.34 (+0.8%)	+0.8%	74.54 (+0.3%)	74.53 (+0.3%)	0.0%
All SIFT (SIFTx3)	**-**	**19.11 (+34.4%)**	**19.24 (+35.3%)**	**+0.7%**	**76.47 (+2.9%)**	**76.38 (+2.8%)**	**-0.1%**
SURF	1024	14.68	14.68	-	74.25	74.25	-
RGB-SURF	3072	15.71 (+7.0%)	15.99 (+8.9%)	+1.8%	74.58 (+0.4%)	74.83 (+0.8%)	+0.3%
OpponentSURF	3072	14.7 (+0.1%)	15.26 (+4.0%)	+3.8%	73.85 (-0.5%)	74.07 (-0.2%)	+0.3%
All SURF (SURFx3)	**-**	**19.4 (+32.2%)**	**19.48 (+32.7%)**	**+0.4%**	**75.89 (+2.2%)**	**76.12 (+2.5%)**	**0.3%**
ORB 256 (no PCA)	256	10.36	10.36	-	71.05	71.05	-
RGB-ORB 256	768	13.02 (+25.7%)	13.58 (+31.1%)	+4.3%	72.86 (+2.6%)	73.21 (+3.0%)	+0.5%
OpponentORB 256	768	12.61 (+21.7%)	12.73 (+22.9%)	+1.0%	72.66 (+2.3%)	72.46 (+2.0%)	-0.3%
All ORB 256	**-**	**16.58 (+60.0%)**	**16.8 (+62.2%)**	**+1.3%**	**74.32 (+4.6%)**	**74.20 (+4.4%)**	**-0.2%**
ORB 80	256	11.43	11.43	-	72.02	72.02	-
RGB-ORB 80	768	13.79 (+20.6%)	13.48 (+17.9%)	-2.2%	73.20 (+1.6%)	72.96 (+1.3%)	-0.3%
OpponentORB 80	768	12.81 (+12.1%)	12.57 (+10.0%)	-1.9%	72.56 (+0.7%)	72.01 (0.0%)	-0.8%
All ORB 80 (ORBx3)	**-**	**17.48 (+52.9%)**	**17.17 (+50.2%)**	**-1.8%**	**74.64 (+3.6%)**	**74.58 (+3.6%)**	**-0.1%**
BRISK 256	512	11.43	11.43	-	72.36	72.36	-
RGB-BRISK 256	1536	11.78 (+3.1%)	12 (+5.0%)	+1.9%	72.74 (+0.5%)	72.67 (+0.4%)	-0.1%
OpponentBRISK 256	1536	11.68 (+2.2%)	11.96 (+4.6%)	+2.4%	72.42 (+0.1%)	72.35 (0.0%)	-0.1%
All BRISK 256 (BRISKx3)	**-**	**16.4 (+43.5%)**	**16.47 (+44.1%)**	**+0.4%**	**74.56 (+3.0%)**	**74.58 (+3.1%)**	**0.0%**
BRISK 80	512	10.73	10.73	-	71.79	71.79	-
RGB-BRISK 80	1536	12.21 (+13.8%)	11.6 (+8.1%)	-5.0%	72.70 (+1.3%)	72.29 (+0.7%)	-0.6%
OpponentBRISK 80	1536	11.05 (+3.0%)	11.15 (+3.9%)	+0.9%	72.10 (+0.4%)	71.49 (-0.4%)	-0.9%
All BRISK 80	**-**	**16.43 (+53.1%)**	**15.95 (+48.6%)**	**-2.9%**	**74.51 (+3.8%)**	**74.39 (3.6%)**	**-0.2%**

TABLE 3
Performance (MXinfAP, % ; MAP@3, %) of pairs and triplets of the best combinations of Table 2 descriptors (SIFTx3 channel-PCA, SURFx3 channel-PCA, ORBx3 typical-PCA, BRISKx3 channel-PCA).

(a) Descriptor pairs	+SURFx3	+ORBx3	+BRISKx3
SIFTx3	**22.4**; 76.64	21.31; **76.81**	20.71; 76.53
SURFx3		21.6; 76.43	21.13; 76.68
ORBx3			19.08; 75.34

(b) Descriptor triplets	+ORBx3	+BRISKx3
SIFTx3+SURFx3	**22.9**; 77.29	22.52; **77.39**
SIFTx3+ORBx3		21.5; 76.61
SURFx3+ORBx3		21.76; 76.56

TABLE 4
Performance (MXinfAP, %, and MAP@3, %) for the best combinations of local descriptors (SIFTx3 channel-PCA, SURFx3 channel-PCA, ORBx3 typical-PCA, BRISKx3 channel-PCA). (a) When features are extracted only from keyframes, (b) when horizontal and vertical tomographs [16] are also examined.

	MXinfAP (indexing)			MAP@3 (annotation)		
Descriptor	(a) Keyframes	(b) Keyframes+ Tomographs	Boost (%) w.r.t (a)	(a) Keyframes	(b) Keyframes+ Tomographs	Boost (%) w.r.t (a)
SIFTx3	19.24	20.28	+5.4%	76.38	76.30	-0.1%
SURFx3	19.48	19.74	+1.3%	76.12	75.98	-0.2%
BRISKx3	16.47	19.08	+15.8%	74.58	75.26	+0.9%
ORBx3	17.48	19.24	+10.1%	74.64	75.16	+0.7%
SIFTx3+SURFx3+ORBx3	22.9	24.57	+7.3%	77.29	77.79	+0.7%

sults, compared to the typical-PCA alternative, without introducing any additional computational overhead.

According to Table 2, for each local descriptor, the combination with its color variants that presents the highest MXinfAP is the following: SIFTx3 with channel-PCA, SURFx3 with channel-PCA, ORBx3 with typical-PCA, BRISKx3 with channel-PCA. In Table 3 we further combine the above to examine how heterogeneous descriptors would work in concert. We can see from the results that the performance increases when pairs of local descriptors (including their color extensions) are combined (i.e., SIFTx3+SURFx3, SIFTx3+ORBx3, SIFTx3+BRISKx3 etc.), which shows a complementarity in the information that the different local descriptors

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

capture. The performance further increases when triplets of different descriptors are employed, with the best combination being SIFTx3+SURFx3+ORBx3. Combining all four considered local descriptors and their color variants did not show in our experiments to further improve the latter results.

In Table 4 we improve selected results of Tables 2 and 3 by additionally exploiting the literature technique of video tomographs [16] (for simplicity, these tomographs are described using only SIFT and its two color extensions). The results of Table 4 indicate that introducing temporal information (through tomographs) can give an additional 7.3% relative improvement to the best results reported in Table 3 (MXinfAP increased from 22.9 to 24.57).

Concerning the performance of independent detectors with respect to the annotation problem, for which results are also presented in Tables 1, 2, 3 and 4, similar conclusions can be reached regarding the usefulness of ORB and BRISK, and how color information is introduced to SURF, ORB and BRISK. Concerning channel-PCA, in this case it does not seem to affect the system's performance: the differences between detectors that use the typical-PCA and channel-PCA are marginal. Another important observation is that in all the above tables a significant improvement of the MXinfAP (i.e., of the indexing problem results) does not lead to a correspondingly significant improvement of results on the annotation problem.

5.4 Results of Exploiting Concept Correlations

In Table 5 we report results of the proposed stacking architecture and compare with other methods that exploit concept correlations. As first layer of the stack we use the best-performing independent detectors of Table 4 (i.e., the last line of Table 4, fusing keyframes and tomographs). We start the analysis with the upper part of Table 5, where we used the output of such detectors for 346 concepts.

In relation to the indexing problem (Table 5:(a),(b)), we observe that the second layer concept detectors alone do not perform so well; in many cases they are not able to outperform the independent first layer detectors (baseline). However, when the concept detectors of the two layers are combined (Table 5:(b)), i.e. the second layer concept detection scores are averaged with the initial scores of the first layer, the accuracy of most methods is improved. More specifically, P-LP outperforms all the compared methods, reaching a MXinfAP of 25.6. LP considers each subset of labels (label sets) presented in the training set as a class of a multi-class problem, which seems to be helpful for the stacking architecture. PPT models correlations on a similar manner, however, it prunes away label sets that occur less times than a threshold. Modelling different kinds of correlations (e.g. by using ML-kNN, CLR) exhibits moderate to low performance. To investigate the statistical significance of the difference of each method from the baseline we used a two-tailed pair-wise sign test [51] and found that only differences between P-LP and the baseline are significant (at 1% significance level).

In relation to the annotation problem (Table 5:(c),(d)) the results show again the effectiveness of the proposed stacking architecture when combined with P-LP, reaching a MAP@3 of 80.88 and improving the baseline results by 4.0%. In this problem also P-MLkNN presents good results, reaching top performance when combined with the detectors of the first layer. Also, for P-LP the relative boost of MXinfAP with respect to the baseline is of the same order of magnitude as the relative boost of MAP@3 (which, as we recall from Section 5.3, is not the case when examining independent concept detectors).

To assess the influence of the number of input detectors in the second layer we also performed experiments where the predictions of a reduced set of 60 concept detectors (the 60 concepts that NIST pre-selected for the TRECVID SIN 2013 task [45]) is used for constructing the meta-level dataset (Table: 5:(II)). Results show that usually a larger input space (detectors for 346 concepts instead of 60) is better, increasing both MXinfAP and MAP@3.

To investigate the importance of stacking-based methods separately for each concept, we closely examine the four best-performing methods of column (b) in Table 5:(I). Fig. 3 shows the difference of each method from the baseline. We observe that the majority of concepts exhibit improved results when any of the second-layer methods is used. The most concepts benefit from the use of P-LP (29 of the 38 concepts), while the number of concepts that benefit from DMF, BSBRM and CF, compared to the baseline, is 25, 21, and 25 respectively. One concept (6:animal) consistently presents a great improvement when concept correlations are considered, while there are 3 concepts (5:anchorperson, 59:hand and 100:running) that are negatively affected regardless of the employed stacking method.

Finally, we take a look at the execution times that each method requires (Table 5:(e)). One could argue that the proposed architecture that uses multi-label learning methods requires considerably more time than the typical BR-stacking one. However, we should note here that extracting one model vector from one video shot, using the first-layer detectors for 346 concepts requires approximately 3.2 minutes in our experiments, which is about three orders of magnitude slower than the slowest of the second-layer methods. As a result of the inevitable computational complexity of the first layer of the stack, the execution time differences between all the second-layer methods that are reported in Table 5 can be considered negligible. This is in sharp contrast to building a multi-label classifier directly from the low-level visual features of video shots, where the high requirements for memory space and computation time that the latter methods exhibit make their application to our dataset practically infeasible.

Specifically, the computational complexity of BR, CLR,

TABLE 5
Performance, (MXinfAP (%), MAP@3 (%) and CPU time), for the methods compared on the TRECVID 2013 dataset. The meta-learning feature space for the second layer of the stacking architecture is constructed using detection scores for (I) 346 concepts and (II) a reduced set of 60 concepts. CPU times refer to mean training (in minutes) for all concepts, and application of the trained second-layer detectors on one shot of the test set (in milliseconds). Columns (a) and (c) show the results of the second layer detectors only. Columns (b) and (d) show the results after combining the output of first and second layer detectors, by means of arithmetic mean. “Baseline” denotes the output of the independent concept detectors that constitute the first layer of the stacking architecture (i.e. the best detectors reported in Table 4). In parenthesis we show the relative improvement w.r.t. the baseline.

Method	MXinfAP (indexing)		MAP@3 (annotation)		
	(a) 2nd layer	(b) 1st and 2nd layer combination	(c) 2nd layer	(d) 1st and 2nd layer combination	(e) Mean Exec. Time Training/Testing
Baseline	24.57	24.57	77.79	77.79	N/A
	(I) Using the output of 346 concepts’ detectors for meta-learning				
DMF [14]	23.97 (-2.4%)	25.38 (+3.3%)	78.71 (+1.2%)	79.12 (+1.7%)	27.62/0.61
BSBRM [32]	24.7 (+0.5%)	24.95 (+1.5%)	79.31 (+2.0%)	79.06 (+1.6%)	1.02/0.08
MCF [33]	24.33 (-1.0%)	24.53 (-0.2%)	76.14 (-2.1%)	77.31 (-0.6%)	1140.98/0.22
CBCFpred [15]	24.32(-1.0%)	24.56 (0%)	78.95 (1.5%)	78.39 (0.8%)	26.84/0.27
CF [37]	23.34 (-5.0%)	25.27 (+2.8%)	78.13 (+0.4%)	78.81 (+1.3%)	55.24/1.22
P-CLR	14.01 (-43.0%)	24.52 (-0.2%)	79.17 (+1.8%)	79.26 (+1.9%)	49.40/9.85
P-LP	**25.23 (+2.7%)**	**25.6 (+4.2%)**	**80.88 (+4.0%)**	79.06 (+1.6%)	549.40/24.93
P-PPT	23.8 (-3.1%)	24.94 (+1.5%)	79.39 (+2.1%)	78.3 (+0.7%)	392.49/0.03
P-MLkNN	19.38 (-21.1%)	24.56 (0.0%)	77.55 (-0.3%)	**79.64 (+2.4%)**	607.40/273.80
	(II) Using the output of a subset of the 346 concepts’ detectors (60 concepts) for meta-learning				
DMF [14]	24.32 (-1.0%)	25.04 (+1.9%)	79.47 (+2.2%)	79.19 (+1.8%)	2.64/0.30
BSBRM [32]	24.71 (+0.6%)	24.96 (+1.6%)	79.82 (+2.6%)	79.26 (+1.9%)	0.65/0.08
MCF [33]	**24.85 (+1.1%)**	24.74 (+0.7%)	77.84 (+0.1%)	77.88 (+0.1%)	466.69/0.18
CBCFpred [15]	15.66 (-36.3%)	22.41 (-8.8%)	79.58 (+2.3%)	79.01 (+1.6%)	2.42/0.25
CF [37]	24.8 (+0.9%)	25.18 (+2.5%)	79.02 (+1.6%)	79.04 (+1.6%)	5.28/0.60
P-CLR	16.16 (-34.2%)	24.44 (-0.5%)	78.85 (+1.4%)	79.12 (+1.7%)	6.32/5.82
P-LP	23.85 (-2.9%)	**25.28 (+2.9%)**	**80.22 (+3.1%)**	79.04 (+1.6%)	208.9/41.43
P-PPT	24.12 (-1.8%)	24.96 (+1.6%)	79.6 (+2.3%)	78.45 (+0.8%)	90.13/0.31
P-MLkNN	22.21 (-9.6%)	24.94 (+1.5%)	77.68 (-0.1%)	**79.42 (+2.1%)**	167.40/72.54

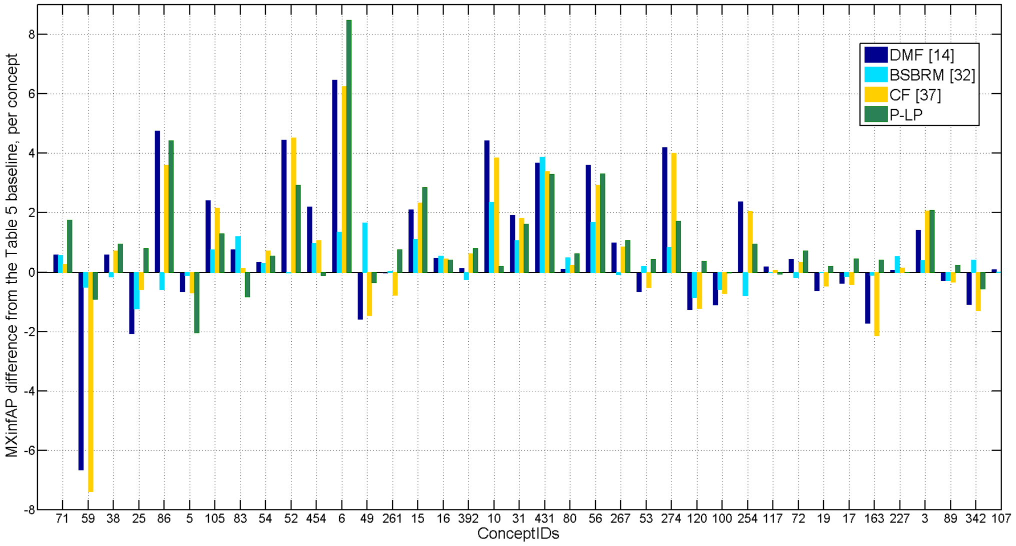

Fig. 3. Differences of selected second layer method from the baseline per concept with respect to the indexing problem when a meta-learning set of 346 concepts is used. Concepts ordered according to their frequency in the test set (in descending order). Concepts on the far right side of the chart (most infrequent concepts) seem to be the least affected, either positively or negatively, by the second-layer learning.

LP and PPT when used in a single-layer architecture depends on the complexity of the base classifier, in our case the Logistic Regression, and on the parameters of the learning problem. Given that the training dataset used in this work consists of more than 500.000 training examples, and each training example (video shot) is represented by a 4000-element low-level feature vector for each visual descriptor, the BR algorithm, which is the simplest one, would build N models for N concepts; CLR, the next least complex algorithm, would build N

IEEE Trans. on Emerging Topics in Computing, vol. 3, no. 2, pp. 193-204, June 2015.
© 2015 IEEE. DOI: 10.1109/TETC.2015.2418714. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7073626

BR-models and $N * (N-1)/2$ one-against-one models. LP and PPT, would build a multi-class model, with the number of classes being equal to the number of distinct label sets in the training set (after pruning, in the case of PPT); this is in order of N^2 in our dataset. Finally ML-kNN would compare each training example with all other (500.000) available examples; in all these cases, the 4000-element low-level feature vectors would be employed. Taking into consideration the dimensionality of these feature vectors, using any such multi-label learning method in a single-layer architecture would require several orders of magnitude more computations compared to the BR alternative that we employ as the first layer in our proposed stacking architecture. In addition to this, typically, multi-label learning algorithms require the full training set to be loaded on memory at once (e.g. [50]), which would be practically unfeasible in a single-layer setting, given the dimensionality of the low-level feature vectors. We conclude that the two major obstacles of using multi-label classification algorithms in a one-layer architecture are the high memory space and computation time requirements, and this finding further stresses the merit of our proposed multi-label stacking architecture.

6 Conclusions

In this work we first dealt with video frame description and representation for concept detection. We showed that two binary local descriptor (ORB, BRISK) can perform reasonably well compared to their state-of-the-art non-binary counterparts in the video semantic concept detection task. We subsequently showed that a methodology previously used for defining two color variants of SIFT is a generic one that is also applicable to other binary and non-binary local descriptors. We also proposed a different way of employing PCA for dimensionality reduction of color descriptors that are used in combination with VLAD (channel-PCA). A second major direction of this work was to take advantage of concept correlation information for building better detectors. For this we proposed an alternative way of employing the stacking architecture, using multi-label learning algorithms in the last level of the stack. We showed that using the proposed architecture in combination with the Label Powerset (LP) method represents an attractive solution. Furthermore, this paper compared concept detection approaches on two different experimental settings: video indexing and annotation. In relation to this comparison, the message that this work aims to pass is that the usual evaluation of concept detection results in a retrieval-based problem setting is not sufficient for assessing the goodness of concept detectors in the context of the annotation problem, and we experimentally underline the importance of reporting evaluation results in both these directions.

Acknowledgments

This work was supported by the EC under contracts FP7-287911 LinkedTV and FP7-600826 ForgetIT.

References

[1] C. G. M. Snoek and M. Worring, "Concept-Based Video Retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2009.

[2] F. Markatopoulou et al., "ITI-CERTH participation to TRECVID 2013," in *TRECVID 2013 Workshop*, Gaithersburg, MD, USA, 2013.

[3] V. Mezaris, P. Sidiropoulos, A. Dimou, and I. Kompatsiaris, "On the use of visual soft semantics for video temporal decomposition to scenes," in *IEEE Int. Conf. on Semantic Computing (ICSC)*, 2010, pp. 141–148.

[4] V. Mezaris, P. Sidiropoulos, and I. Kompatsiaris, "Improving interactive video retrieval by exploiting automatically-extracted video structural semantics," in *IEEE Int. Conf. on Semantic Computing (ICSC)*, 2011, pp. 224–227.

[5] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, "A joint content-event model for event-centric multimedia indexing," in *IEEE Int. Conf. on Semantic Computing (ICSC)*, 2010, pp. 79–84.

[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*, ser. LNCS. Springer, 2006, vol. 3951, pp. 404–417.

[8] D. M. Chen, M. Makar, A. F. de Araújo, and B. Girod, "Inter-frame coding of global image signatures for mobile augmented reality," in *DCC*, 2014, pp. 33–42.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE Int. Conf. on Computer Vision*, 2011, pp. 2564–2571.

[10] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *IEEE Int. Conf. ICCV 2011*, 2011, pp. 2548–2555.

[11] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[12] I. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[13] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE Int. Conf. on CVRP 2010*, SF, CA, 2010, pp. 3304–3311.

[14] J. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *2003 Int. Conf. on Multimedia and Expo. (ICME)*. NY: IEEE, 2003, pp. 445–448.

[15] W. Jiang, S.-F. Chang, and A. C. Loui, "Active context-based concept fusion with partial user labels," in *IEEE Int. Conf. on Image Processing*. NY: IEEE, 2006.

[16] P. Sidiropoulos, V. Mezaris, and I. Kompatsiaris, "Video tomographs and a base detector selection strategy for improving large-scale video concept detection," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1251–1264, 2014.

[17] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Int. Conf. CVPR 2012*, 2012, pp. 510–517.

[18] A. Canclini et al., "Evaluation of low-complexity visual feature detectors and descriptors," in *18th Int. Conf. on Digital Signal Processing (DSP), 2013*, 2013, pp. 1–7.

[19] C. Grana, D. Borghesani, M. Manfredi, and R. Cucchiara, "A fast approach for integrating orb descriptors in the bag of words model," in *SPIE*, vol. 8667, 2013, pp. 866 709–866 709–8.

[20] J. Fu et al., "C-surf: Colored speeded up robust features," in *Trustworthy Computing and Services*, ser. Communications in Computer and Information Science, Y. Yuan, X. Wu, and Y. Lu, Eds. Springer, 2013, vol. 320, pp. 203–210.

[21] P. Fan, A. Men, M. Chen, and B. Yang, "Color-SURF: A surf descriptor with local kernel color histograms," in *IEEE Int. Conf. on Network Infrastructure and Digital Content*, 2009, pp. 726–730.

[22] D. Chu and A. Smeulders, "Color invariant surf in discriminative object tracking," in *Trends and Topics in Computer Vision*, ser. LNCS. Springer, 2012, vol. 6554, pp. 62–75.

[23] S. Strat, A. Benoit, P. Lambert, and A. Caplier, "Retina enhanced surf descriptors for spatio-temporal concept detec-

tion," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 443–469, 2014.

[24] S. T. Strat, A. Benoit, and P. Lambert, "Retina enhanced bag of words descriptors for video classification," in *22nd Europ. Signal Processing Conf. (EUSIPCO)*, 2014, pp. 1307–1311.

[25] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," *Pattern Recognition*, vol. 35, pp. 1675–1686, 2002.

[26] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," in *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, ser. Communications in Computer and Information Science, P. Richard and J. Braz, Eds. Springer Berlin, 2011, vol. 229, pp. 28–42.

[27] K. E. A. Van de Sande, C. G. M. Snoek, and A. W. M. Smeulders, "Fisher and vlad with flair," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[28] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *British Machine Vision Conference*. British Machine Vision Association, 2011, pp. 76.1–76.12.

[29] H. Jegou et al., "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[30] G.-J. Qi et al., "Correlative multi-label video annotation," in *15th Int. Conf. on Multimedia*. NY: ACM, 2007, pp. 17–26.

[31] M. Wang, X. Zhou, and T.-S. Chua, "Automatic image annotation via local multi-label classification," in *Int. Conf. on Content-based image and video retrieval - CIVR '08*. NY: ACM, 2008, pp. 17–26.

[32] G. Tsoumakas et al., "Correlation-Based Pruning of Stacked Binary Relevance Models for Multi-Label learning," in *ECML/PKDD 2009 Workshop on Learning from Multi-Label Data (MLD'09)*. Berlin: Springer-Verlag, 2009, pp. 101–116.

[33] M.-F. Weng and Y.-Y. Chuang, "Cross-Domain Multicue Fusion for Concept-Based Video Indexing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1927–1941, 2012.

[34] T. Meng et al., "Florida International University and University of Miami TRECVID 2013," in *TRECVID 2013 Workshop*, Gaithersburg, MD, USA, 2013.

[35] S.-I. Yu et al., "Informedia @TRECVID 2012," in *TRECVID 2012 Workshop*, Gaithersburg, MD, USA, 2012.

[36] F. Wang, Z. Sun, D. Zhang, and C. Ngo, "Semantic Indexing and Multimedia Event Detection: ECNU at TRECVID 2012," in *TRECVID 2012 Workshop*, Gaithersburg, MD, USA, 2012.

[37] A. Hamadi, P. Mulhem, and G. Quenot, "Conceptual feedback for semantic multimedia indexing," in *11th Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, 2013, pp. 53–58.

[38] G. Nasierding and A. Z. Kouzani, "Empirical Study of Multi-label Classification Methods for Image Annotation and Retrieval," in *2010 Int. Conf. on Digital Image Computing: Techniques and Applications*. China: IEEE, 2010, pp. 617–622.

[39] F. Markatopoulou, V. Mezaris, and I. Kompatsiaris, "A comparative study on the use of multi-label classification techniques for concept-based video indexing and annotation," in *MultiMedia Modeling*, ser. LNCS, vol. 8325. Springer, 2014, pp. 1–12.

[40] F. Markatopoulou et al., "A study on the use of a binary local descriptor and color extensions of local descriptors for video concept detection," in *21st Int. Conf. on MultiMedia Modeling*, Sydney, 2015.

[41] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[42] J. Read, "A pruned problem transformation method for multi-label classification," in *2008 New Zealand Computer Science Research Student Conference (NZCSRS)*, New Zealand, 2008.

[43] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*. Berlin: Springer, 2010, pp. 667–686.

[44] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[45] P. Over et al., "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[46] E. Yilmaz, E. Kanoulas, and J. A. Aslam, "A simple and efficient sampling method for estimating ap and ndcg," in *31st ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*. USA: ACM, 2008, pp. 603–610.

[47] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *IEEE Int. Conf. ICCV 2007*, Rio de Janeiro, 2007, pp. 1–8.

[48] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. NY: ACM, 2001, pp. 245–250.

[49] B. Safadi and G. Quénot, "Re-ranking by local re-scoring for video indexing and retrieval," in *20th ACM Int. Conf. on Information and Knowledge Management*. NY: ACM, 2011, pp. 2081–2084.

[50] G. Tsoumakas, E. Spyromitros-xioufis, J. Vilcek, and I. Vlahavas, "MULAN : A Java Library for Multi-Label Learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[51] H. M. Blanken, A. P. de Vries, H. E. Blok, and L. Feng, *Multimedia Retrieval*. NY: Springer Berlin Heidelberg, 2005.

Foteini Markatopoulou received her BSc in Informatics from Aristotle University of Thessaloniki in 2011 and her MSc in Advanced Computing: Machine Learning and Data Mining from University of Bristol. Her research is in the area of Machine Learning and Pattern recognition with application in Multimedia Analysis. She is a PhD student in Queen Mary University of London and works as a research assistant at the Information Technologies Institute (ITI) of the Centre of Research & Technology Hellas (CERTH).

Vasileios Mezaris received the BSc and PhD in Electrical and Computer Engineering from the Aristotle University of Thessaloniki in 2001 and 2005, respectively. He is a Senior Researcher (Researcher B) at the Information Technologies Institute (ITI) of the Centre for Research of Technology Hellas (CERTH). His research interests include image and video analysis, retrieval, event detection, machine learning for multimedia analysis. He is an Associate Editor for the IEEE Trans. on Multimedia and a Senior Member of the IEEE.

Nikiforos Pittaras received his BSc in Computer Science from the Computer Science & Engineering Department, University of Ioannina. His research focuses on features for multimedia concept detection. He is a research assistant at the Information Technologies Institute (ITI) of the Centre of Research & Technology Hellas (CERTH).

Ioannis Patras received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Crete, Heraklion, Greece, in 1994 and 1997, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, Delft (TU Delft), The Netherlands, in 2001. He is a Senior Lecturer in computer vision with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. His current research interests are in the area of computer vision and pattern recognition, with emphasis on the analysis of human motion. He is an Associate Editor of the Image and Vision Computing Journal.