Automatic analysis of online image data for law enforcement agencies by concept detection and instance search

Maaike H.T. de Boer, Henri Bouma [1], Maarten C. Kruithof, Frank B. ter Haar,
Noëlle M. Fischer, Laurens K. Hagendoorn, Bart Joosten, Stephan Raaijmakers

TNO, Anna van Buerenplein 1, 2595 DA The Hague, The Netherlands

ABSTRACT

The information available on-line and off-line, from open as well as from private sources, is growing at an exponential rate and places an increasing demand on the limited resources of Law Enforcement Agencies (LEAs). The absence of appropriate tools and techniques to collect, process, and analyze the volumes of complex and heterogeneous data has created a severe information overload. If a solution is not found, the impact on law enforcement will be dramatic, e.g. because important evidence is missed or the investigation time is too long. Furthermore, there is an uneven level of capabilities to deal with the large volumes of complex and heterogeneous data that come from multiple open and private sources at national level across the EU, which hinders cooperation and information sharing. Consequently, there is a pertinent need to develop tools, systems and processes which expedite online investigations. In this paper, we describe a suite of analysis tools to identify and localize generic concepts, instances of objects and logos in images, which constitutes a significant portion of everyday law enforcement data. We describe how incremental learning based on only a few examples and large-scale indexing are addressed in both concept detection and instance search. Our search technology allows querying of the database by visual examples and by keywords. Our tools are packaged in a Docker container to guarantee easy deployment on a system and our tools exploit possibilities provided by open source toolboxes, contributing to the technical autonomy of LEAs.

Keywords: Concept detection, instance search, logo recognition, incremental learning, large scale indexing, open source software, Law Enforcement Agency (LEA).

1. INTRODUCTION

The exponential growth rate of data relevant for police investigations places an increasing demand on the limited resources of Law Enforcement Agencies (LEAs). The absence of appropriate tools and techniques to collect, process, and analyze the volumes of complex and heterogeneous data has created a severe information overload. If a solution is not found, the impact on law enforcement will be dramatic, e.g. because important evidence is missed or the investigation time is too long. Furthermore, there is an uneven level of capabilities to deal with the large volumes of complex and heterogeneous data that come from multiple open and private sources at national level across the EU, which hinders cooperation and information sharing. As such there is a pertinent need to develop tools, systems and processes which expedite online investigations.

The H2020 project ASGARD (http://asgard-project.eu/) develops an analysis system for gathered raw data. The goal is to contribute to LEA technological autonomy and effective use of technology to process massive amounts of raw data. The LEA autonomy is enhanced by using – as much as possible – open-source software to avoid vendor lock-in. Raw data may include text, audio, video and images. The open internet and the dark web contain a huge amount of images, which can be related to various phenomena, such as radicalization or illegal market places. Therefore, scalable and reliable image analysis is quite important.

In this paper, we describe analysis tools to identify and localize generic concepts, instances of objects and logos in images. Concept search allows the retrieval of generic concepts, such as 'car' or 'face' or 'flag'. The concept detector is able to recognize and localize the concepts in the image and this allows users to give one or multiple keywords and the system

[1] henri.bouma@tno.nl; phone +31 888 66 4054; http://www.tno.nl

M.H.T. de Boer, H. Bouma, M.C. Kruithof, F.B. ter Haar, N.M. Fischer, L.K. Hagendoorn, B. Joosten, S. Raaijmakers,
"Automatic analysis of online image data for law enforcement agencies by concept detection and instance search,"
Proc. SPIE, Vol. 10441, (2017). http://dx.doi.org/10.1117/12.2277970

retrieves only the images that contain this keyword. Instance search allows the retrieval of specific instances based on an example image. Instead of the generic search for a concept 'car', instance search allows to search for an image that contains a specific instance of a car, e.g. *my neighbor's blue Toyota*. We also describe how to address incremental learning based on only few examples and large-scale indexing in the concept detection and instance search. The search technology allows querying of the database by visual examples and by keywords. Our tools are packaged in a Docker container to guarantee easy deployment on a system and our tools exploit possibilities provided by open-source toolboxes, to avoid vendor lock-in and to contribute to the technical autonomy of LEAs. We show that our concept detection and localization performs reaches a high precision, even on tiny objects in the image. Our concept-detection approach can recognize all classes with one network, which makes it very efficient. The instance search reaches even higher performance, while only requiring one query example. The instance-search approach uses only six tiles, which makes it fast but it hinders the detection of tiny objects.

The outline of this paper is as follows. Section 2 gives an overview of related work. Section 3 presents our solution for analyzing image data. Section 4 describes our experimental setup and the experimental results. Finally, Section 5 summarizes our conclusions.

2. RELATED WORK

2.1 Concept detection

Concept detection aims to recognize and localize generic concepts. Concepts may include objects, actions, scenes and events. An example of a generic concept is a 'car', which includes vehicles of different brands, different owners and different colors. In concept detection by keyword, words are used to search for a certain concept. For each image, algorithms will automatically produce a set of annotations that consist of a class label (keyword), bounding box and confidence scores.

In the field of object recognition, hand-crafted features have been succeeded by deep convolutional neural networks (D-CNN) without explicit feature construction [15]. Several implementations have been proposed, such as Decaf/Caffe [21][36][45], VGG [79], Overfeat [78] and R-CNN [26]. Typically, the object detectors that perform well on public benchmarks are trained on large collections, e.g., ImageNet [20] or annotated subsets, e.g., the Pascal Visual Object Challenge (VOC) [23] and the ImageNet Large-Scale Recognition Challenge (ILSVRC) [45][75].

In the localization of objects, methods have evolved from scanning window approaches [24] to methods based on so-called *region proposals* e.g., by using objectness [1], multi-scale combinatorial grouping [70], selective search [87] or EdgeBox [98]. These region-proposal methods generate a limited number of candidate locations to reduce the search space [26]. An overview of region-proposal methods can be found in the paper by Hosang et al., [31]. For object detection, one of the most widely used approaches is R-CNN [26], or one of its faster variants in which convolutions are shared over region proposals, such as Spatial Pyramid Pooling (SPPnet) [29] and Fast R-CNN [27]. These methods also have the two distinct phases of region proposals and CNN classification.

Currently, fully CNN-based methods are proposed that directly perform object detection, without a separate hand-crafted phase for region proposal generation [54]. This allows for end-to-end learning and optimization of the system. Early attempts to take the proposal stage out of the loop led to performance degradations [60]. Examples are OverFeat [78], MultiBox [22], R-CNN minus R [49] and YOLO [72]. Different from prior work, these methods predict bounding boxes and class probabilities directly from full images in one evaluation. This direction has led to a new revolution in object detection and recently many novel methods have been proposed, including DenseBox [32], Proposal-Free Network [50], YOLO9000 [71] single-shot detection (SSD) [52], MSC-MultiBox [22], G-CNN [60], DecompNet [63], DeepBox [47], DeepMask [69], Boxsup [17], Region-based FCN [18] and region proposal network (RPN) for Faster R-CNN [73]. Some of these approaches perform detection in one phase [52][72] and others still have a separate phase to propose bounding boxes with a neural network [73]. One of the challenges with these models is runtime performance. In real world systems, both localization and detection should be applied in real time. Often the models with a separate phase for selecting bounding boxes have high performance, but take longer to run that the novel networks with only one phase.

Based on the results of the winning teams at the international benchmarks, such as TRECVID [62], ILSVRC [45] and MS-COCO [51], using ensembles of multiple models appears to be the common strategy. The TRECVID-MED evaluation benchmark of 2015 showed top performance of the Vireo-TNO team for Multi-media event detection (MED) [62][95], especially in the cases with zero examples or few (10) examples. This team used 6 kinds of high-level feature concepts

(CNN [45] applied to several datasets, such as ImageNet [20], TrecVid-SIN-2014, MED-2014, MIT-places [97], FCVID [39] and Sports [42]) and one kind of low-level motion feature (Improved dense trajectory [88], which contains HOG [19], HOF and MBH, using PCA and Fisher Vectors [65]). Besides the visual features, also textual [82] and speech features [25] are used.

The TRECVID-SIN evaluation benchmark of 2015 showed top performance of the MediaMill (UVA + QualComm) team for Semantic Indexing (SIN). Mediamill used an approach that was based on a mixture of Inception [85] and VGG [79] networks.

The TRECVID-LOC evaluation benchmark of 2015 showed top performance of the CCNY [91] and MediaMill (UVA + QualComm) team [83] for concept localization (LOC). The CCNY system is based on R-CNN [26]. The candidates are generated with Selective Search and Edge Boxes. Aside from the AlexNet [45], which is employed in the initial version of R-CNN, they also employ another network structure, GoogLeNet [84] by concatenating the features extracted from these two CNN models. A linear SVM classifier is then employed to learn a more discriminate model to classify each region proposal. Inspired by the dense trajectories for action recognition, they propose a region trajectory algorithm to exploit temporal information.

CUImage (Hongkong) is the winner of the object detection tasks (DET and LOC) from the ILSVRC-2016, where they used an ensemble of 6 models. With their earlier work, the same group managed to win the video-task (VID) from the ILSVRC-2015 [41]. A gated bidirectional network (GBD) network [96] with 269 layers is fine-tuned on 200 detection classes. This GBD network passes messages between features from different support regions during both feature learning and feature extraction.

He et al. obtained good results with the residual networks (ResNets) [30]. Best performance on the MS-COCO challenge of 2016 was shown with an implementation of Huang et al. [33], who used an ensemble of five Faster RCNN models jointly trained end-to-end using a pure Tensorflow implementation with differentiable ROI cropping. They used a combination of Inception-Resnet [85] and (stride 8) Resnet-101 [30] base feature extractors.

Within the PASCAL VOC, MSCOCO and ILSVRC challenges, a trend is to use the Faster RCNN model in the ensembles [52]. The SSD model promises comparable, but faster, performance [52].

2.2 Instance search

Historically, instance search was focused on computing SIFT descriptors [55] at sparse interest points. Each of the variable number of descriptors is compared with descriptors in another image to perform matching. To preserve spatial information, a geometric transformation is performed and only matches within a threshold distance are accepted as matches (similar to the robust RANSAC estimator). Matching a variable and large number of descriptors in one image to descriptors in another image takes a lot of memory to store all descriptors and a lot of time to perform the comparison. To allow more efficient implementations for large-scale retrieval, the descriptors are mapped to a fixed length bag of visual words (BoW, sometimes called: bag of features; BoF) with k-means clustering. Similar to the concept recognition, instance search used BoW methods, especially those based on the hand-crafted SIFT descriptor, and later moved towards data-driven CNN based methods [62][96]. In general, the SIFT descriptor [55] transforms an image into a collection of local features. The local features are quantized to visual words using a pre-trained codebook that uses clustering techniques such as K-means. In a Bag of Words approach, a histogram of the visual words is used. Another approach is using a vector of aggregated local descriptors (VLAD) [35]. The centroids in feature space are calculated and the aggregated difference with respect to the centroids is used as the representation of the image. A third method is using a Fisher Vector [65], which uses the Maximum Likelihood estimation to train a Gaussian Mixture Model. The similarity measure between two images can be calculated using class weighting schemes, such as TF-IDF and the inverted index. TFIDF is a product of the frequency of a specific (visual) word in a document or image and the inverse document frequency (occurrence in the other images in the set) [80][81]. The normalized scalar product between the query image and all test images is used to rank the test images. A higher rank means a higher probability that the image contains the instance. The BoW model with the inverted index is similar to the cosine distance between two L2 normalized vectors [96]. Geometric verification, spatial analysis and pseudo-relevance feedback are re-ranking strategies that are often used to increase performance [76].

Zheng et al. [96] describe two types of CNNs: one-pass and multi-pass. In the one-pass CNN model, an image is passed through the network one time. The convolutional filters are viewed as the local feature detectors and the fully connected layers produce the response maps. The multi-pass CNN models use regions or patches generated using similar methods as used in the concept localization and feeds the different regions through the CNN model. The response from a fully connected layer, often fc7, is aggregated to form a compact vector, for example using VLAD or a pooling method. The compact vectors can be compared using metrics similar to those used by SIFT, such as Inverted Index or cosine similarity.

CNN based methods do, however, not necessarily outperform SIFT-based methods [14]. Chandrasekhar et al. give some practical pointers in how to use a CNN and Fisher vectors. According to the survey on instance search by Zheng [96], the best strategy is to use a SIFT-based BoW method with re-ranking and/or a multi-pass CNN method. They point out that in the future the lessons learned from the SIFT method should be used in the CNN methods. Recently, Salvador et al. [76] explored a fine-tuned CNN method with re-ranking and Yan et al. [90] showed that we can exploit the complimentary properties of SIFT and CNNs to obtain higher performance.

The already mentioned TRECVID benchmark also includes a task on instance search (INS). An overview of the task can be found in Awad et al. [2]. The TRECVID evaluation of 2015 showed top performance of the PKU-ICST team for INS [62][64]. In general, nearly all systems use some form of SIFT local descriptors where a large variety of experiments are addressing representation, fusion or efficiency challenges. Most systems also include a CNN component [62]. PKU-ICST team combined different feature types (CNN and BoW based on K-means and SIFT keypoints), feature matching (Cosine distance and multi-bag SVM), keypoint matching and re-ranking based on text matching. WHU-NERCMS, the winning team of 2016, also uses a combination of CNN and BoW based on K-means and SIFT, and adds filters based on face recognition and scenes [89]. Within the interactive instance search, a common strategy is to use object sketches [62][7].

Other tasks address the search for specific buildings, such as Oxford [66] and Paris [67] and specific objects, such as holiday scenes [34], cars or shoes [86]. Tao et al. show that methods that work well on specific buildings, might not work on other objects, such as shoes. Images from buildings are often taken in 2D from one side, whereas shoes can be imaged full-view in 3D. Methods tailored for finding buildings cannot be guaranteed to perform well on the shoes. Tao et al. propose an attribute-based method that works well over all types of instance search tasks.

2.3 Incremental learning

Deep neural networks often need many training examples and long training times to train all layers of the network. However, when the basic network has been trained for multiple classes, addition of a novel concept is possible without retraining all layers of the complete network. This is called incremental learning.

Current CNN methods need many training examples, preferably millions, to train a good model. Some domains do not have such massive amount of examples. Another challenge for CNNs is incremental learning, with new classes becoming available incrementally. Both challenges can be solved by retraining an existing CNN model. Many CNNs trained on natural images appear to learn features similar to Gabor filters and color blobs in the first layer. These first-layer features appear to be general for many datasets and tasks, while the latter layers appear to be more-and-more specific for particular classes [3][4][54]. It was shown that it is possible to flexibly learn novel objects incrementally with only a low number of training samples by freezing many initial layers and retraining a classifier on the outputs of the initial layers [12][26]. This approach has been used in an interactive demonstrator [9][77]. In transfer learning [5][6], a base network is pre-trained on a base dataset, and these learned features are transferred to be further trained on a target dataset. A network that was trained and applied to the same dataset is called a ‘selfer’ network and a network that was trained on one dataset and then applied to another dataset is called a ‘transfer’ network. Transfer of the features works well if the features are general, and not specific for the base task. Yosinski et al. [92] already performed a valuable study about the transferability from a base dataset to a target dataset of features from each layer of the network. Experiments with a Caffe-based network [36] on 1000 ImageNet classes [20][45] and with Keras (which is a high-level deep learning library running on top of Theano or Tensorflow) on the CIFAR-10 dataset [44] showed that the accuracy on the target dataset improves when more target data is used and that when the target dataset is small, it is beneficial to transfer (and freeze) many layers [46]. For a small target dataset, the ‘transfer’ network boosts generalization and it performs much better than the ‘selfer’ network.

Besides retraining a model, several other techniques can be used. The TRECVID-MED evaluation benchmark of 2015 showed top performance of the Vireo-TNO team for Multi-media event detection (MED) [62][95], especially in the cases with zero examples or few (10) examples. In the zero example case, a user query was matched to a pre-trained set of concept detectors. Often, manual matching is used [95], but automatic methods including knowledge based techniques [8], word2vec [10], or other methods from textual retrieval [56] can be used to find the related concept detectors. A state of the art method is to use a linear weighted sum over the relevant concept detectors to provide a score for a video [95], but recently evidence-based pooling methods, inspired by event recounting methods, have been proposed [56]. In the 10 example case, the winning team of 2015 used a combination of the zero example system and the hundred example system, which uses a Chi-Square SVM trained on the concatenated feature vector of the concept detectors for the visual features and a linear SVM trained on the improved dense trajectory feature as the motion feature. Several fusion techniques can be used [11], but average fusion over different modalities seems the best method for few examples learning.

Other strategies in few example learning contain the training of a semantic embedding between words and videos [28][61], attribute based learning [48], using co-occurrence statistics [58], tag propagation [57], training on easy examples first [37] and using pseudo-relevance feedback [38][68].

2.4 Large-scale indexing

Large-scale indexing is deployed to speed-up the retrieval process allowing for rapid retrieval of results when the user performs a query. The images are preprocessed and the intermediate results are stored in a structured way. When the user performs a query, it is not necessary to access the pixels of every image again and to compare the query to every intermediate result. This works similarly to finding a word in the dictionary, where you can jump immediately to the middle of the book when you search for a word that starts with an 'M' without reading every word at the beginning or the end. Large-scale indexing is essential to handle massive amounts of raw data.

The goal of the localization and detection of concepts is to provide a capability to their users, for example law enforcement agencies. In order to make a product that can be used, we would like to search for a concept (either by keyword or with an image) and retrieve the relevant results within seconds. To obtain such a capability, we need large-scale indexing. All images are indexed and stored in a database. The important features and representation of images in the database must be stored as well, to avoid pixel access during rapid retrieval. The CNN has to be pre-trained to allow fast application on a new image. The localization method should be fast to achieve real time search. The CNN, however, needs to be flexible in a way that incremental learning and learning with few examples is possible and the indexing can be changed, for example when a new concept is trained and applied on the images in the database.

For concept detection, Yu et al. [93] propose a semantic search engine that can search through 1 million videos within 1 second on 1 core. They use a pre-computed background pair-wise dot-product matrix, which is a fixed background dataset with pre-computed features to use in training. They represent the images using a product quantized feature representation, which is a compression method. Additionally, a fast re-ranking method is used that is able to retain 80% of their original performance.

In instance search, Chatfield et al. [16] show that it is beneficial to only use the spatial re-ranking on the top images, because it is a costly operation. In the similarity metric only the images that are assigned to the same visual word are used. Pair-wise distance calculations can be fast and are parallelizable. Similar to concept detection, product quantization is used to compress the images and re-ranking or filtering can speed up the process.

In general, Jónsson et al. [40] recently posed some research questions regarding scalability and multimedia analytics, such as how database management techniques can facilitate the increasingly large-scale collections, which query language should be used, how we should support different workloads and how systems can facilitate in interactive search with dynamic data collections. Currently, these are still research questions, but it is important to take into account the indexing possibilities with the current state of the art methods.

3. METHODS

3.1 Concept detection

Our implementation for the identification and localization of concepts in images is based on SSD [52]. The SSD is based on a pretrained (ImageNet) VGG16 model [79] which is combined with bounding-box priors for the localization of the concepts. We used a version of the SSD that was pretrained on the PASCAL VOC dataset and the MS-COCO dataset. We retrained this model on images of the concepts of interest that were scraped from the internet and manually annotated. Because we used a small number of training samples we augmented the samples by randomly adjusting the saturation, brightness and contrast and by randomly flipping the image. To train the network we used the Adam optimizer [43] with a learning rate of $3 \cdot 10^{-5}$ and a batch size of 4. The network was finetuned on a network trained on the PASCAL VOC dataset with the first 10 layers frozen. The implementation uses the Keras deep-learning toolbox with Tensorflow as an engine.

We used incremental learning in concept detection as follows. To facilitate adding a new concept without needing many training samples and a long training time we retrain only the upper part of the SSD network which contains the features that are specific for the different concepts (similar to what was done in [12][46]).

The large-scale indexing and incremental learning are straight-forward for the ‘old’ networks with a clear separation between candidate generation, deep-learning feature computation (e.g., Caffe), and classification (e.g., with SVM) [12]. However, it remains challenging for state-of-the-art end-to-end networks.

3.2 Instance search

Nowadays, everyone is sharing photos and images via social media, and often after editing operations, such as downscaling, cropping, adding text/remarks as overlay, or inclusion in a flyer or collage. The original image content is often manipulated or put out of context. Our instance search aims to find the whereabouts of specific image snippets from such edited content.

The TRECVID instance search challenge proves that a combination of CNN and SIFT features works well for content based image retrieval. In this subsection, we focus on the part that implements the SIFT descriptor in a visual vocabulary for fast instance search. Similar to the work of Perronnin et al. [65] the descriptors are mapped to a Fisher Vector and used as the visual vocabulary. The implementation is fully based on Python with VLFeat and OpenCV support.

The visual vocabulary of an image is computed as follows. First the image is normalized to a diagonal of 500 pixels, after which SIFT keypoints are detected and SIFT descriptors are extracted. The descriptors are converted to Root-SIFT descriptors and mapped to a normalized vector using Principal Component Analysis (PCA) to whiten the data and reduce the descriptor size from 128 to 64. Then a pretrained Gaussian Mixture Model (GMM) based on a K-means clustering (K=128) is used to create a Fisher Vector (length 16,384). To learn the PCA-mapping and GMM for RootSIFT descriptors, a subset of 3200 images were randomly selected from ImageNet [20] and 700k RootSIFT descriptors extracted for the calculations. To each Fisher Vector the signed square root function is applied and then L2-normalized. The final Fisher Vectors are compared using the L2 distance.

The logic behind this approach is that two images with a similar object (e.g. a bottle of pills) have a large number of features that correspond, have a similar distribution with respect to the K-means clustering, and consequently have a similar Fisher Vector. For instance search, we assume that the query is an image snippet for which most content matters. The database, on the other hand, contains images of which only a subpart matches the query. To cope with this, a snippet selection strategy is applied to each image that is added to the database. After the snippet selection each snippet is normalized in size (diagonal = 500 pixels) and a Fisher Vector (FV) is created. The URL, the snippet’s bounding box and the FVs are stored. While querying the database the query-FV is compared to each snippet-FV of a database image and the best match is returned. This way, instance search can retrieve relevant subparts of images in the database. The current implementation of the snippet selection is straight forward; it considers the whole image as a snippet, the four quadrants, and one snippet in the center. Alternatively, an object-proposal method, such as RegionLets, is considered.

Large-scale indexing can be implemented in the following way. The fixed-length representation (such as Bag of Words or Fisher Vector) first ranks the references that have the largest number of similar local features with the query. An inverted index on the reference descriptions exploits the sparsity of the representation and allows fast walkthrough in the database.

4. EXPERIMENTS AND RESULTS

4.1 Datasets

To evaluate our methods, the VOC2012 [23] and the FlickrLogos [74] datasets were used. Both datasets have a wide range of images and manual segmentations for more than two thousand objects. The VOC2012 consists of 17,125 images with 2913 annotated images with segmented objects and 14,212 images without objects. The FlickrLogos dataset has 8240 images with 2240 annotated images with segmented objects and 6000 images without these objects. This dataset has 32 logo classes and 70 images for each class. For most experiments, only the annotated images were used for performance estimation, unless indicated otherwise.

To investigate the performance in relation to the object size, the results are divided over the categories based on image coverage percentages of [0, 2.5, 5, 7.5, 10, 20, 30, 40, 50, 60, 100]. The distribution of the samples over the categories in the FlickrLogos dataset is shown in Figure 1. The coverage of these segments varies between 2% up to 99% coverage of the original image (30% on average), thus providing a wide range of snippet sizes ideal for a break-down analysis of our methods. Example queries in the VOC set are shown in Figure 2.

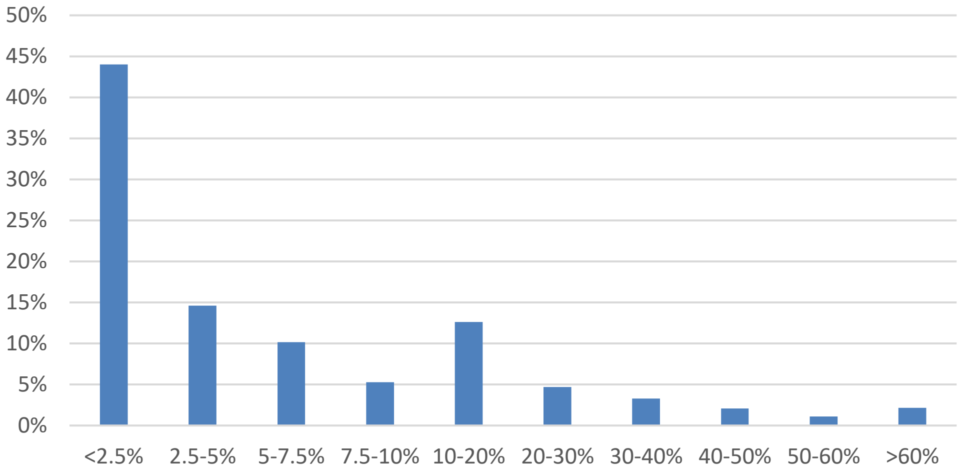

Figure 1. Histogram of samples in FlickrLogos over size categories.

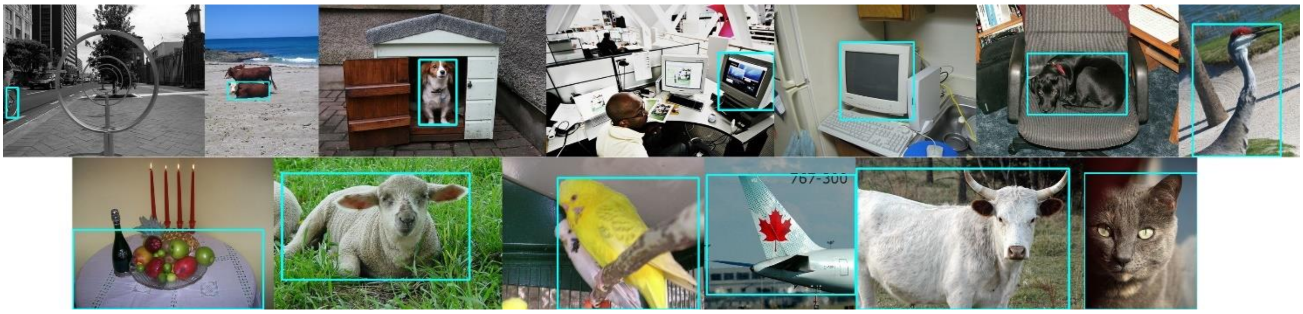

Figure 2. Example queries of the VOC set with an increasing coverage of the query snippet with respect to its original.

4.2 Concept detection

The SSD network is known to localize objects quite well, the model trained on the PASCAL VOC 2007 and 2012 and the MS-COCO datasets is known to have a Mean Average Precision (MAP) score of 79.6% [52]. The SSD network is applied to the FlickrLogos dataset [74], to estimate the performance when using a limited amount of training data. To avoid overtraining on a limited set of data, 3 layers were frozen. We used 5-fold cross validation to estimate the performance of the detector. So, for each logo class, we used 56 example images for training, aiming to retrieve the 14 test images in the database of 2184 images. For concept detection and localization this is an extremely low number of training examples. Although the method localizes the logo in the image, we only evaluate the capability to recognize a class without assessing the quality of the localization. Applying the object detector to an image (inference) is very efficient, because it can recognize all different classes with the same network. The Top-1 and Top-5 scores are reported in Figure 3. Despite the low number of training samples, the overall Top-1 and Top-5 precision scores are 68% and 87% respectively. Note that the performance for larger objects (>40% coverage) is slightly worse than for small objects. This is caused by the lack of training data for larger objects (Figure 1).

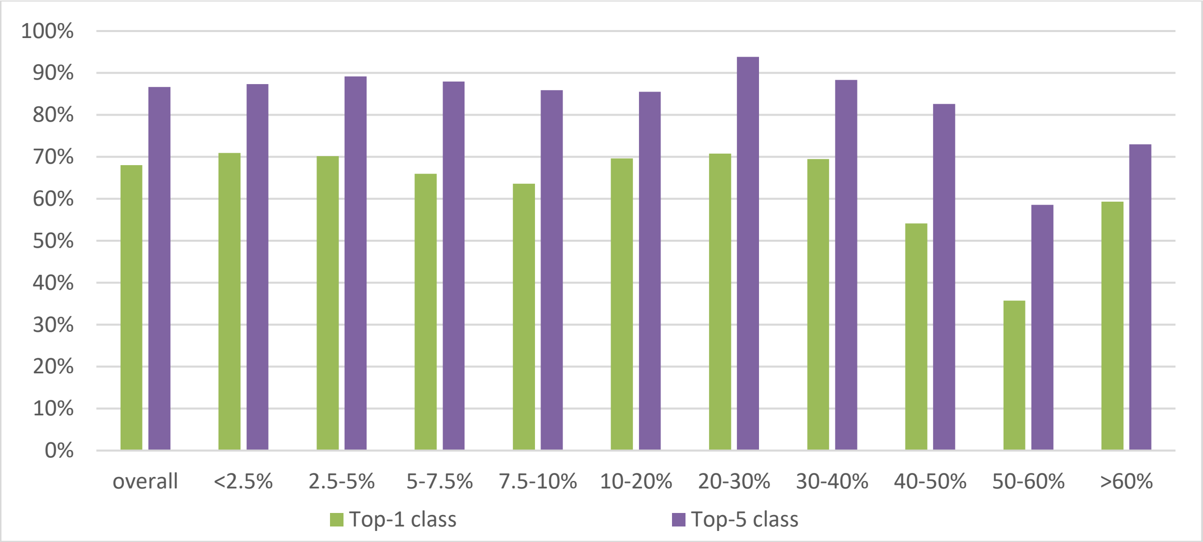

Figure 3. Precision results in the top-1 and top-5 for concept detection on FlickrLogos for different size categories.

4.3 Instance search

Our instance-search method aims to find the locations of specific image snippets in an image. For our application, we need to find a specific snippet of an object in a database of images. We used the datasets of FlickrLogos and VOC2012 to estimate the performance of our method. To build up the feature database for fast image retrieval, each image is converted to six snippets normalized in size and the calculated RootSIFT features for each snippet are converted to a Fisher Vector. In the feature database each image is represented by six FVs of which the best matching FV is returned when querying the database. This conversion to FVs makes it very efficient to query a database; only a single multiplication of a vector to a matrix is required.

For each of the 2913 mask images of the VOC2012, the bounding box around the first segment (the red segment) was calculated and applied as a cropping region for its corresponding color image. It is expected that snippets with a large coverage match better. For the FlickrLogos we select the bounding box of the largest component from the mask image and create query-snippets with that. The majority of these snippets cover less than 14% of the original image, which has a large impact on the overall results. In our evaluation, we aim to retrieve for each query-snippet the original uncropped image at Top-1, although for the FlickrLogos the confusion is expected to by higher because there are 70 images with the same logo.

For evaluation, we query the dataset with a snippet and return the rank of the original image (the query was selected from), and report the precision percentage in the Top-1 and in the Top-5. To investigate the performance in comparison to the coverage of snippet versus original, the results are also divided over the categories based on coverage percentages of [0, 2.5, 5, 7.5, 10, 20, 30, 40, 50, 60, 100]. With more focus on the low coverage queries, we get better insight in the performance for these difficult queries. Figure 2 shows example queries of the VOC set with increasing coverage.

The precision results of the experiment are shown in Figure 4, with the overall precision in the first columns. In contrast to the concept-detection method, tiny snippets are almost impossible to find, but at 7.5% coverage already half of the queries has the Top-1 result. From 20% coverage and more each query is already within the Top-5. Some typical results of our method are shown in Figure 5. As expected, performance on the FlickrLogos dataset is lower, because logos that belong to the same class can be very similar.

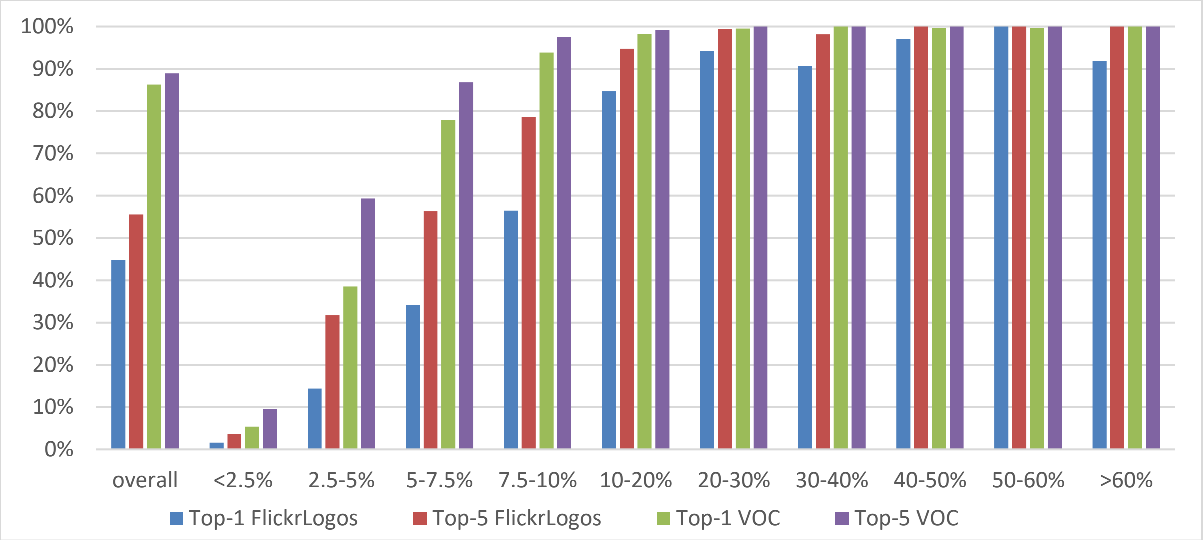

Figure 4. Precision results in the top-1 and top-5 of the instance search on FlickrLogos and VOC2012.

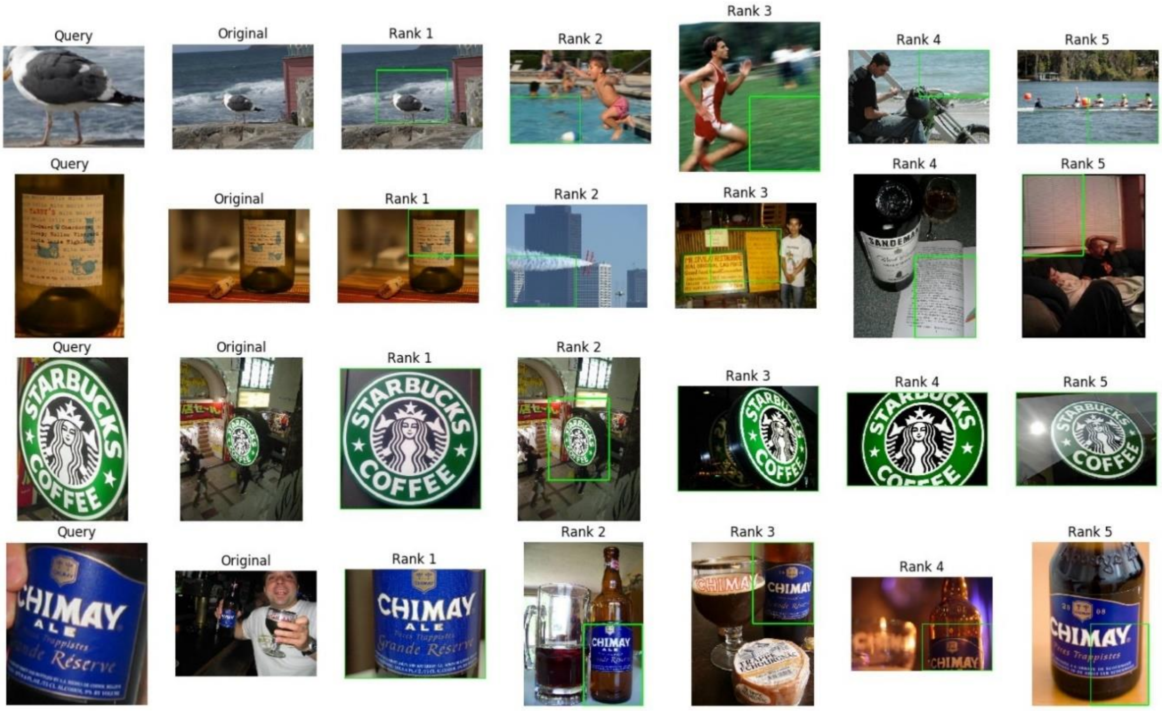

Figure 5. Typical results of our instance-search method on VOC (top) and FlickrLogos (bottom). Two correct results, a Rank-2 result and a failure according to our goal to find the original image the snippet was selected from.

In case we accept any image containing the queried logo, which makes it a class recognition instead of instance search, the Top-1 and Top-5 results improve significantly. These results are shown in Figure 6. The results distinguish between the *exact same logo* on top and *any logo from the same class* on top. For the application of logo class recognition we achieve 87% Top-1 and 93% Top-5 for the 2240 logo-queries on the dataset of FlickrLogos (Figure 6). The difference in performance on the subset or the complete dataset is minimal (Table 1).

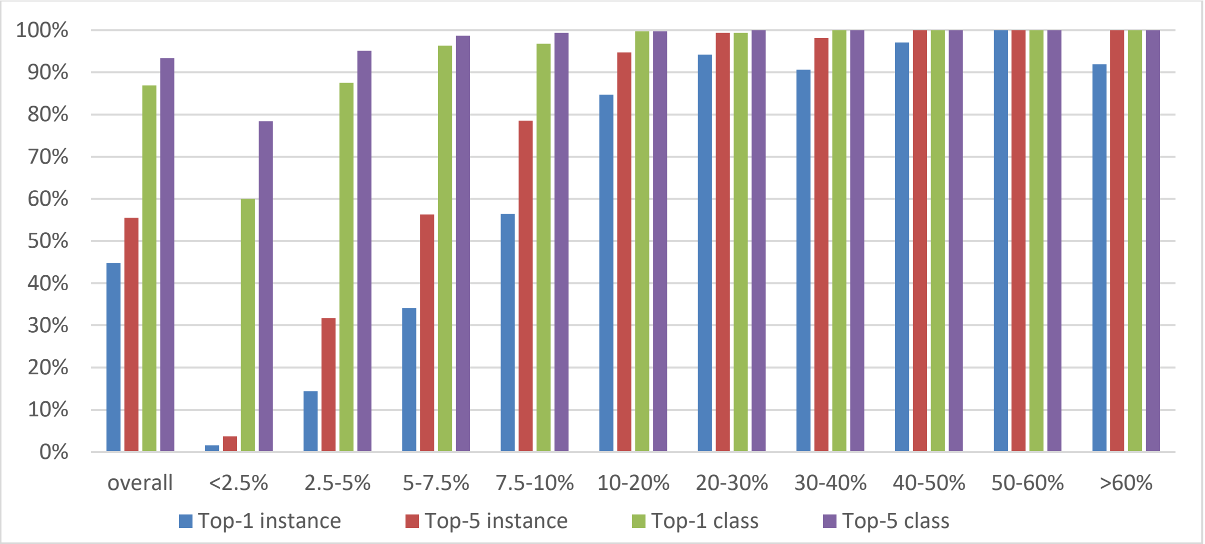

Figure 6. Precision results in the Top-1 and Top-5 for instance search and class recognition on FlickrLogos based on the instance-search method.

Table 1: Overall precision (%) in the Top-1 and Top-5 of instance search on the subset or complete dataset.

	FlickrLogos		**VOC2012**	
	Subset 2240 x 2240	**Complete 2240 x 8240**	**Subset 2913x2913**	**Complete 2913x17125**
Top-1 Instance	45	45	86	85
Top-5 Instance	56	55	89	87
Top-1 Class	87	85	-	-
Top-5 Class	93	92	-	-

In our experiments, we have shown that our method can be successfully applied to our instance-search task and also for class recognition, e.g. to retrieve the copies and edited images that correspond to a specific snippet. The strengths of our instance-search method are:

- No model-learning required; only one query image and a GMM (<1MB) are required for a large set of SIFT descriptors.
- No sliding window or object segmentation required; the strength of SIFT feature detection converted to a FV for six basic tiles is sufficient for our retrieval task.
- No direct feature matching; instead of directly matching the SIFT descriptors each image is converted to FVs of fixed length, which enables fast and large-scale image retrieval suitable for hierarchical search operations.

We also observe that instance-search the rapid approach with six tiles of 25% and larger are not optimal for objects with tiny image coverage below 10%.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we described analysis tools to identify and localize generic concepts, instances of objects and logos in images. These tools can be used by LEAs to retrieve relevant content in images. We also described how incremental learning based on only few examples and the large-scale indexing can be deployed in concept detection or instance search. The search technology allows querying of the database by visual examples and by keywords. Our concept detection and localization performs reaches a Top-5 precision of 87%, even on tiny objects in the image. Our concept-detection approach can recognize all classes with one network, which makes it very efficient. The instance search reaches even higher performance (Top-5 precision of 93%), while only requiring one query example. The instance-search approach uses only six tiles, which makes it fast but it hinders the detection of tiny objects.

Future work may include the following. For large scale indexing we are still looking into the possibilities regarding the deep learning frameworks that are available for concept localization. One possible solution would be to store the features of the top layer of the layers that are frozen during incremental learning. This would speed up training but would not necessarily decrease the storage size, since the network features sometimes become larger than the image at certain layers in the network. For the faster R-CNN, the usage of the frozen layer features would be difficult since the network uses the raw pixels in one of the upper layers.

ACKNOWLEDGEMENT

The ASGARD ("Analysis System for Gathered Raw Data") project has received funding from the EU H2020 Secure Societies program (FCT-01-2015) under grant agreement number 700381. The authors want to thank the project partners for their contributions and especially the French Alternative Energies and Atomic Energy Commission (CEA) for their collaboration in the image-analysis task. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

REFERENCES

[1] Alexe, B., Deselaers, T., Ferrari, V., "Measuring the objectness of image windows," IEEE Trans. PAMI 34(11), 2189-2202 (2012).

[2] Awad, G., et al., "Instance search retrospective with focus on TRECVID," International Journal of Multimedia Information Retrieval 6(1), 1-29 (2017).

[3] Azizpour, H., Razavian, A., Sullivan, J., Maki, A., Carlsson, S., "From generic to specific deep representations for visual recognition," IEEE CVPR, 36-45 (2015).

[4] Azizpour, H., Sharif-Razavian, e.a., "Factors of transferability for a generic ConvNet representation," IEEE Trans. PAMI, 38(9), 1790-1802 (2016).

[5] Bengio, Y., "Deep learning of representations for unsupervised and transfer learning," JMLR Proc. Unsupervised and Transfer Learning 27, 17-36 (2012).

[6] Bengio, Y., Bastien, F., Bergeron, A., et al., "Deep learners benefit more from out-of-distribution examples," JMLR Proc. AISTATS, 164–172 (2011).

[7] Bhattacharjee, S.D., Yuan, J., Tan, Y.P. and Duan, L.Y., "Query-adaptive small object search using object proposals and shape-aware descriptors," IEEE Trans. Multimedia 18(4), 726-737 (2016).

[8] Boer, M. de, Schutte, K. and Kraaij, W., "Knowledge based query expansion in complex multimedia event detection," Multimedia Tools and Applications 75(15), 9025-9043 (2016).

[9] Boer, M. de, Brandt, P., Sappelli, M., Daniele, L.M., Schutte, K., Kraaij, W., "Query interpretation - an application of semiotics in image retrieval," Int. J. Adv. Software 3/4(8), 435-449 (2015).

[10] Boer, M.H.T. de, Lu, Y., Zhang, H., Schutte, K., Ngo, C., Kraaij, W., "Semantic reasoning in zero example video event retrieval," ACM Trans. on Multimedia Comp. Comm. and Appl., (2017).

[11] Boer, M. de, Schutte, K., Zhang, H., e.a., "Blind late fusion in multimedia event retrieval," Int. J. of Multimedia Information Retrieval 5(4), 203-217, (2016).

[12] Bouma, H., Eendebak, P., Schutte, K., Azzopardi, G., Burghouts, G., "Incremental concept learning with few training examples and hierarchical classification," Proc. SPIE 9652, (2015).

[13] Bouma, H., Azzopardi, G., Spitters, M., et al., "TNO at TRECVID 2013: Multimedia event detection and instance search," Proc. TRECVID, (2013).

[14] Chandrasekhar, V., Lin, J., Morère, O., Goh, H., Veillard, A., "A practical guide to CNNS and Fisher Vectors for image instance retrieval," Signal Processing, 128, 426-439 (2016).

[15] Chatfield, K., Simonyan, K., Vedaldi, A. et al., "Return of the devil in the details: delving deep into convolutional nets," BMVC, (2014).

[16] Chatfield, K., Arandjelović, R., Parkhi, O., Zisserman, A., "On-the-fly learning for visual search of large-scale image and video datasets," Int. J. Multimedia Information Retrieval 4(2), 75-93 (2015).

[17] Dai, J., He, K. and Sun, J., "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," IEEE ICCV, 1635-1643 (2015).

[18] Dai, J., Li, Y., He, K., Sun, J., "R-FCN: Object detection via region-based fully convolutional networks," Advances in Neural Information Processing Systems, 379–387 (2016).

[19] Dalal, N., Triggs, B., "Histograms of oriented gradients for human detection". Proc. IEEE CVPR, 886-893 (2005)

[20] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., "Imagenet: a large-scale hierarchical image database," IEEE CVPR, 248–255 (2009).

[21] Donahue, J., Jia, Y., Vinyals, O., et al., "Decaf: A deep convolutional activation feature for generic visual recognition," ICML, (2014).

[22] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D., "Scalable object detection using deep neural networks," IEEE CVPR, 2155–2162 (2014).

[23] Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A., "The PASCAL visual object classes (VOC) challenge," IJCV 88, 303–338 (2010).

[24] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan. D., "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010).

[25] Gauvain, J., Lamel, L., Adda, G., "The LIMSI broadcast news transcription system," Speech communication 37(1), 89-108 (2002).

[26] Girshick, R., Donahue, J., e.a., "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE CVPR, 580–587 (2014).

[27] Girshick, R. "Fast R-CNN, " Proc. of the Int. Conf. on Computer Vision (ICCV), 1440–1448 (2015).

[28] Habibian, A., Mensink, T., & Snoek, C. G., "Videostory: A new multimedia embedding for few-example recognition and translation of events", Proc. ACM Int. Conf. Multimedia, 17-26 (2014).

[29] He, K., Zhang, X., Ren, S., Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition," ECCV, 346–361 (2014).

[30] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition," IEEE CVPR, 770–778 (2016).

[31] Hosang, J., Beneson, R., e.a., "What makes for effective detection proposals," IEEE Trans. on pattern analysis and machine intelligence 38(4), 814-830 (2016).

[32] Huang, L., Yang, Y., Deng, Y., Yu, Y., "Densebox: Unifying landmark localization with end to end object detection," arXiv:1509.04874, (2015).

[33] Huang, J., Rathod, V., Sun, Chen, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," IEEE CVPR, (2017).

[34] Jégou, H., Douze, M., Schmid, C., "Hamming embedding and weak geometric consistency for large scale image search," ECCV, 304-317 (2008).

[35] Jégou, H., Douze, M., Schmid, C., Pérez, I., "Aggregating local descriptors into a compact image representation," Proc. IEEE CVPR, 3304–3311 (2010).

[36] Jia, Y., Shelhamer, E., Donahue, J., et al., "Caffe: convolutional architecture for fast feature embedding," Proc. ACM Multimedia, 675–678 (2014).

[37] Jiang, L., Meng, D., Mitamura, T., Hauptmann, A. G., "Easy samples first: Self-paced reranking for zero-example multimedia search," Proc. ACM Int. Conf. on Multimedia, 547-556 (2014).

[38] Jiang, L., Mitamura, T., Yu, S., Hauptmann, A., "Zero-example event search using multimodal pseudo relevance feedback," Int. Conf. Multimedia Retrieval, (2014).

[39] Jiang, Y., Wu, Z., Wang, J., Xue, X., Chang, S. F., "Exploiting feature and class relationships in video categorization with regularized deep neural networks," IEEE Trans. PAMI, (2017).

[40] Jónsson, B., Worring, M., Zahálka, J., Rudinac, S., Amsaleg, L., "Ten research questions for scalable multimedia analytics," Int. Conf. on Multimedia Modeling, 290-302 (2016).

[41] Kang, K., Li, H., Yan, J., Zeng, X., "T-CNN: tubelets with convolutional neural networks for object detection from videos," IEEE CVPR, 817–825 (2016).

[42] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., "Large-scale video classification with convolutional neural networks," CVPR, 1725-1732 (2014).

[43] Kingma, D., Ba, J., "Adam: a method for stochastic optimization," ICLR, (2015).

[44] Krizhevsky, A., Hinton, G., "Convolutional deep belief networks on CIFAR-10," Tech. Report Univ Toronto, (2010).

[45] Krizhevsky, A., Sutskever, I., and Hinton, G., “ImageNet classification with deep convolutional neural networks,” NIPS, (2012).

[46] Kruithof, M., Bouma, H., Fischer, N., Schutte, K., “Object recognition using deep convolutional neural networks with complete transfer and partial frozen layers,” Proc. SPIE 9995, (2016).

[47] Kuo, W., Hariharan, B., Malik, J., “DeepBox: Learning objectness with convolutional networks,” Proc. IEEE ICCV, 2479-2487 (2015).

[48] Lampert, C. H., Nickisch, H., & Harmeling, S., “Learning to detect unseen object classes by between-class attribute transfer,” IEEE CVPR, 951-958 (2009).

[49] Lenc, K., Vedaldi, A., “R-CNN minus R,” CoRR 1506-06981, (2015).

[50] Liang, X., Wei, Y., Shen, X., and others, “Proposal-free network for instance-level object segmentation,” arXiv:1509.02636,, (2015).

[51] Lin, T., Maire, M., Belongie, S., Hays, J., et al., “Microsoft COCO: Common objects in context,” ECCV, 740-755 (2014).

[52] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C., “SSD: Single shot multibox detector,” ECCV, 21-37 (2016).

[53] Long, M., Cao, Y., Wang, J., Jordan, M., “Learning transferable features with deep adaptation networks,” ICML, 97-105 (2015).

[54] Long, J., Shelhamer, E., Darrell, T., “Fully convolutional networks for semantic segmentation,” CVPR, 3431–3440 (2015).

[55] Lowe, D., “Distinctive image features from scale-invariant keypoints,” IJCV 60(2), 91–110 (2004).

[56] Lu, Y. J., Zhang, H., de Boer, M., & Ngo, C. W., “Event detection with zero example: select the right and suppress the wrong concepts,” Proc. 2016 ACM on Int. Conf. on Multimedia Retrieval, 127-134 (2016).

[57] Mazloom, M., Li, X., & Snoek, C. G., “Few-example video event retrieval using tag propagation,” Proc. of Int. Conf. on Multimedia Retrieval, 459 (2014).

[58] Mensink, T., Gavves, E., Snoek, C. G., “Costa: Co-occurrence statistics for zero-shot classification,” Proc. IEEE CVPR, 2441-2448 (2014).

[59] Muja, M., Lowe, D., “Flann, fast library for approximate nearest neighbors,” Int. Conf. Computer Vision Theory and Applications VISAPP, (2009).

[60] Najibi, M., Rastegari, M., Davis, L., “G-CNN: an iterative grid-based object detector,” IEEE CVPR, 2369-2377 (2016).

[61] Norouzi, M. Mikolov, T., Bengio, S., and others, “Zero-shot learning by convex combination of semantic embeddings,” ICLR, (2014).

[62] Over, P., Awad, G., Fiscus, J., et al., “TRECVID 2015 – an overview of the goals tasks data evaluation mechanisms and metrics,” Proc. TrecVid, (2015).

[63] Park, E., Berg, A. C., “Learning to decompose for object detection and instance segmentation,” ICLR workshop, (2016).

[64] Peng, Y., et al., “PKU-ICST at TRECVID 2015: Instance Search Task”, Proc. TRECVID, (2015).

[65] Perronnin, F.,Liu, Y., Sánchez, J., Poirier, H., “Large-scale image retrieval with compressed fisher vectors,” IEEE CVPR, 3384–3391 (2010).

[66] Philbin, J. , Chum, O., Isard, M., Sivic, J., Zisserman, A., “Object retrieval with large vocabularies and fast spatial matching,” IEEE CVPR, 1-8 (2007).

[67] Philbin, J. , Chum, O., Isard, M., Sivic, J., Zisserman, A., “Lost in quantization: Improving particular object retrieval in large scale image databases,” IEEE CVPR, 1-8 (2008).

[68] Pingen, G., de Boer, M., Aly, R., “Rocchio-based relevance feedback in video event retrieval,” Int. Conf. on Multimedia Modeling, 318-330 (2017).

[69] Pinheiro, P., Collobert, R., Dollar, P., “Learning to segment object candidates,” Adv. NIPS, 1990-1998 (2015).

[70] Pont-Tuset, J., Arbelaez, P., Barron, J.T., Marques, F. and Malik, J., “Multiscale combinatorial grouping for image segmentation and object proposal generation,” IEEE Trans. PAMI 39, 128-140 (2017).

[71] Redmon, J., Farhadi, A., “YOLO9000: Better, Faster, Stronger,” arXiv:1612.08242, (2016).

[72] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., “You only look once: unified real-time object detection,” IEEE CVPR, 779–788 (2016).

[73] Ren, S., He, K., Girshick, R., Sun, J., “Faster R-CNN: Towards real-time object detection with region proposal networks,” Advances in NIPS, 91-99 (2015).

[74] Romberg, S., Pueyo, L., Lienhart, R., Zwol, R. van., “Scalable logo recognition in real-world images,” ACM ICMR, (2011).

[75] Russakovsky, O., Deng, J., Su, H., Krause, J., et al., “ImageNet large scale visual recognition challenge,” IJCV 115(3), 211-252 (2015).

[76] Salvador, A., Giró-i-Nieto, X., Marqués, F., & Satoh, S., “Faster R-CNN features for instance search,” IEEE CVPR workshop, (2016).

[77] Schutte, K., Bouma, H., Schavemaker, J., et al., “Interactive detection of incrementally learned concepts in images with ranking and semantic query interpretation,” IEEE Content-Based Multimedia Indexing CBMI, (2015).

[78] Sermanet, P., Eigen, D., Zhang, X., et al., “Overfeat: integrated recognition, localization and detection using convolutional networks,” ICLR, (2014).

[79] Simonyan, K., Zisserman, A., “Very deep convolutional networks for large-scale image recognition,” ICLR, (2015).

[80] Sivic, J., Zisserman, A., “Video Google: A text retrieval approach to object matching in videos,” ICCV, 2 (1470), 1470-1477 (2003).

[81] Sivic, J., Zisserman, A., “Video Google: Efficient visual search of videos”. In Toward category-level object recognition, Springer, 127-144 (2006).

[82] Smith, R., “An overview of the Tesseract OCR engine,” IEEE Int. Conf. Document Analysis and Recognition, 629 – 633 (2007).

[83] Snoek, C., Cappallo, S., Fontijne, D., et al., “Qualcomm Research and UvA at TRECVID 2015: Recognizing concepts, objects and events,” Proc. TRECVID, (2015).

[84] Szegedy, C., Liu, W., Jia, Y., et al., “Going deeper with convolutions,” IEEE CVPR, 1-9 (2015).

[85] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” arXiv:1602.07261, (2016).

[86] Tao, R., Smeulders, A., e.a., “Generic instance search and re-identification from one example via attributes and categories,” arXiv:1605.07104, (2016).

[87] Uijlings, J., Sande, K., Gevers, T., Smeulders, A., “Selective search for object recognition,” IJCV 104(2), 154-171 (2013).

[88] Wang, H., Schmid, C., “Action recognition with improved trajectories,” IEEE ICCV, 3551-3558 (2013).

[89] Wang, Z., Yang, Y., Guan, S., Han, C., Lan, J., Shao, R., Wang, J., Liang, C., “WHU-NERCMS at TRECVID2016: Instance Search Task,” Proc. TRECVID, (2016).

[90] Yan, K., Wang, Y., Liang, D., Huang, T., Tian, Y., “CNN vs. SIFT for image retrieval: alternative or complementary?,” Proc. ACM Multimedia Conf., 407-411 (2016).

[91] Ye, Y., Rong, X., Yang, X., Tian, Y., “CNNY at TRECVID 2015: Video semantic concept localization,” Proc. TRECVID, (2015).

[92] Yosinski, J., Clune, J., Bengio, Y., Lipson, H., “How transferable are features in deep neural networks?,” NIPS, 3320–3328 (2014).

[93] Yu, S. I., Jiang, L., Xu, Z., Yang, Y., Hauptmann, A., “Content-based video search over 1 million videos with 1 core in 1 second,” Proc. ACM Int. Conf. on Multimedia Retrieval, 419-426 (2015).

[94] Zeng, X., Ouyang, W., Yang, B., “Gated bi-directional CNN for object detection,” Proc. ECCV, 354-369 (2016).

[95] Zhang, H., Lu, Y., Boer, M. de, Haar, F. ter, e.a., “VIREO-TNO at TRECVID 2015: Multimedia event detection,” Proc. TRECVID, (2015).

[96] Zheng, L., Yang, Y., Tian, Q., “SIFT meets CNN: a decade survey of instance retrieval,” IEEE Trans. PAMI, (2017).

[97] Zhou, B., Khosla, A., Lapedriza, A., e.a., “Places: An image database for deep scene understanding,” arXiv:1610.02055, (2016).

[98] Zitnick, C., Dollar, P., “Edge boxes: Locating object proposals from edges,” ECCV, 391-405 (2014).