Chapter 1
Finding semantically-related videos in closed collections

Foteini Markatopoulou, Markos Zampoglou, Evlampios Apostolidis, Symeon Papadopoulos, Vasileios Mezaris, Ioannis Patras, Ioannis Kompatsiaris

Abstract Modern newsroom tools offer advanced functionality for automatic and semi-automatic content collection from the Web and social media sources to accompany news stories. However, the content collected in this way often tends to be unstructured and may include irrelevant items. An important step in the verification process is to organise this content, both with respect to what it shows, and with respect to its origin. This chapter presents our efforts in this direction, which resulted in two components. One aims to detect semantic concepts in video shots, to help annotation and organization of content collections. We implement a system based on deep learning, featuring a number of advances and adaptations of existing

Foteini Markatopoulou
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: markatopoulou@iti.gr

Markos Zampoglou
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: markzampoglou@iti.gr

Evlampios Apostolidis
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece and School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: apostolid@iti.gr

Symeon Papadopoulos
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: papadop@iti.gr

Vasileios Mezaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: bmezaris@iti.gr

Ioannis Patras
School of Electronic Engineering and Computer Science, Queen Mary University, London, UK, e-mail: i.patras@qmul.ac.uk

Ioannis Kompatsiaris
Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, e-mail: ikom@iti.gr

algorithms to increase performance for the task. The other component aims to detect logos in videos in order to identify their provenance. We present our progress from a keypoint-based detection system to a system based on deep learning. We present the developed methodologies and the evaluation results for both components.

1.1 Problem Definition and Challenge

News events typically give rise to the creation and circulation of User-Generated Content (UGC). This media content, typically in the form of images or videos, spreads in social media and attracts the attention of news professionals and investigators. Such events generate multiple different media items, often published on different platforms, and at different times following the breaking of the event.

In many cases news organizations use automatic or semi-automatic tools to collect such content. These tools crawl the Web and various media sharing platforms and collect potentially related content based on search queries. This leads to the formation of unstructured media collections, which may contain both relevant and irrelevant content. It may include content from different aspects of event, possibly taken at different times and displaying different scenes. It may also include content from different sources, each of which may focus on a different aspect or exhibit different forms of bias.

As a way of assisting the verification process, it is very helpful to organise the collected videos according to what they depict, or based on who published them. This organization step is assumed to take place after the near-duplicate retrieval step (see Chapter 4) which can identify near-duplicates and remove or aggregate them. Consecutively, the semantic-level analysis described in this chapter can allow grouping, comparison, and contrasting, as well as cross-referencing to spot videos that may be altered, misleading, or irrelevant. To this end, we developed two components within InVID that semantically analyse content, the first performing Semantic Video Annotation, and the second Logo Detection. The former analyses videos or video segments, and annotates them with detected concept labels, such as “car”, “dancing”, or “beach”. The second looks for logos in the video, which can reveal the group, agency, or institution sharing (or re-sharing) it, and this in turn can reveal possible biases or intentions behind the posting of the item, as well as allow the investigator to link it to past content published by the same party. In this sense, these two components of InVID cover similar needs from different aspects, offering ways to automatically annotate videos or video segments with semantic tags on their content and origin, allowing more nuanced search within closed collections.

With respect to semantic annotation, video content can be annotated with simple concept labels that may refer to objects (e.g. “car” and “chair”), activities (e.g. “running” and “dancing”), scenes (e.g. “hills” and “beach”), etc. Annotating videos with concepts is a very important task that facilitates many applications such as finding semantically-related videos in video collections, semantics-based video segmentation and retrieval, video event detection, video hyperlinking and concept-based

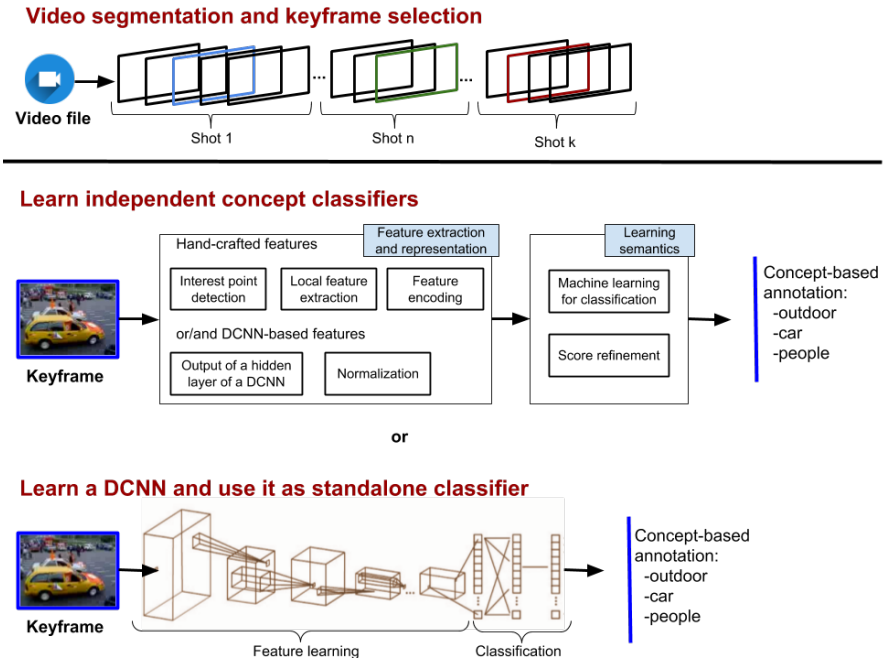

Fig. 1.1: Video concept annotation pipelines: After temporal video segmentation, e.g. using automatic video shot detection and extracting one representative keyframe per shot, the upper part shows a typical concept-based video annotation pipeline that is based on hand-crafted or DCNN-based features and supervised classifiers trained separately for each concept. The lower part is based on features that can be learned directly from the raw keyframe pixels using a DCNN, and subsequently using the DCNN as standalone classifier to perform the final class label prediction.

video search [68, 44, 24, 63, 74, 75, 39, 21, 23]. Concept-based video search refers to the retrieval of video fragments (e.g. keyframes) that present specific simple concept labels from large-scale video collections. Thus, within InVID, the task entails creating a module that will be able to reliably annotate videos by taking their keyframes and detecting any known concepts found within them.

With respect to logo detection, the ability to annotate videos with respect to their provenance can be an important part of verification. Knowing who first produced the video or is responsible for its dissemination can help determine potential bias in the content or form an initial impression about its trustworthiness. Furthermore, identifying the source of the video can help establish contact, in order to ask permissions or verify the authenticity of content. Even when no direct contact with the content creator or owner is possible, determining the content's origin can provide important context for the verification process. Since many times content tends to be reproduced not by sharing but by re-uploading, it is commonly hard to find the

original source. However, in this process of tracing the video, logos can play an important role, provided they can be identified.

While many logos, –and especially the ones belonging to the most popular news channels– are well known, especially among news professionals, there exist many organizations which are not so easy to identify, whether less well-known channels, or unofficial groups such as paramilitary organizations or independent journalist groups. There exist more than 27,000 TV broadcast stations in the world according to the CIA World Factbook[1], and a very large –and hard to estimate– number of paramilitary groups. Those cases are aggravated by the large numbers of such logos that a professional might have to memorise, and a certain degree of instability which leads to groups merging or splitting (this is the case with militant groups in the Syrian Civil War, for example). As a result, identifying one logo among the multitude of possible candidates is very challenging for human investigators (Fig. 1.2). In those cases, automatically identifying the logo and providing information about its owner can significantly speed up the analysis and verification process.

Fig. 1.2: Top: Two video frames with easily identifiable news channel sources; Bottom: Two video frames where the logos cannot be easily identified

It is important to note that, in cases where we have to deal with videos consisting of multiple shots, each shot should be treated independently, since it may contain different logos and entirely different semantic concepts. Thus, both components are aimed to operate at the shot level, after the videos have been analyzed by the video shot fragmentation component of the InVID platform.

[1] https://www.nationsencyclopedia.com/WorldStats/CIA-Television-broadcast-stations.html, accessed 08 April 2019.

1.2 Semantic Video Annotation

1.2.1 Related Work

To deal with concept-based video search, concept-based video annotation methods have been developed that automatically annotate video-fragments, e.g. keyframes extracted from video shots, with semantic labels (concepts), chosen from a predefined concept list [68]. A typical concept-based video annotation system mainly follows the process presented in Fig. 1.1. A video is initially segmented into meaningful fragments, called shots; each shot is represented by e.g. one or more characteristic keyframes. Then, several hand-crafted or DCNN-based (Deep Convolutional Neural Network) features are extracted from the generated representation of each shot; e.g. visual features from the extracted keyframes, and audio and textual features from the audio representation of the shot. Given a ground-truth annotated video training set, supervised machine learning algorithms are then used to train concept classifiers independently for each concept, using the extracted features and ground-truth annotations. The trained classifiers can subsequently be applied to an unlabelled video shot, following feature extraction, and return a set of confidence scores for the appearance of the different concepts in the shot. A recent trend in video annotation is to learn features directly from the raw keyframe pixels using DCNNs. DCNNs consist of many layers of feature extractors, and are thus able to model more complex structures in comparison to handcrafted representations. DCNN layers can learn different types of features without requiring feature engineering, in contrast to the hand-crafted features that are designed by humans to capture specific properties of video frames, e.g. edges and corners. DCNNs can be used both as standalone classifiers (Fig. 1.1, bottom), i.e. unlabelled keyframes are passed through a pre-trained DCNN that performs the final class label prediction directly, using typically a softmax or a hinge loss layer [64, 31], and also as extractors for video keyframe features (Fig. 1.1, top), i.e. the output of a hidden layer of the pre-trained DCNN is used as a global keyframe representation [64]. This latter type of features is referred to as DCNN-based, and in that case DCNN features are used to train binary classifiers (e.g. SVMs) separately for each concept.

While significant progress has been made during the last years in the task of video annotation and retrieval, it continues to be a difficult and challenging task. This is due to the diversity in form and appearance exhibited by the majority of semantic concepts and the difficulty to express them using a finite number of representations. The system needs to learn a practically limitless number of different patterns that characterise the different concepts (e.g. landscapes, faces, actions). As a result, generality is an important property that a concept-based video annotation system should present in order to generalise its performance across many different heterogeneous concepts. Finally, computational requirements are another major challenge. The large number of concepts that a video annotation system should learn is computationally expensive requiring lightweight and fast methods. Finally, there are by far more labelled datasets available that contain still images than datasets

extracted from video keyframes. Typically classifiers are trained on the former still image datasets and applied on video datasets, which is a suboptimal practice.

It has been shown that combining many different features for the same concept, instead of using a single feature, improves concept annotation accuracy. However, which subsets of features to use for the needs of a specific task, and which classification scheme to follow, is a challenging problem that will affect the accuracy and computational complexity of the complete concept-based video annotation system. Other methods also improve the overall video annotation accuracy by looking for existing semantic relations e.g. concept correlations. As discussed above the dominant approach for performing concept-based video annotation is to train DCNNs whereby concepts share features within the architectures up to the very last layer, and then branch off to T different classification branches (using typically one layer), where T is the number of concepts [49]. However, in this way, the implicit feature-level relations between concepts, e.g. the way in which concepts such as a *car* and *motorcycle* share lower-level features modelling things like their wheels, are not directly considered. Also, in such architectures, the relations or inter-dependencies of concepts at a semantic level, i.e. the fact that two specific concepts may often appear together or, inversely, the presence of the one may exclude the other, are also not directly taken into consideration. In this chapter we will refer to methods that have been proposed for exploiting in a more elaborate way one of these two different types of concept relations. Then, in Section 1.2.2 we will present a more advanced method that jointly exploits visual- and semantic-level concept relations in a unified DCNN architecture.

1.2.1.1 Supervised Learning Using Deep Networks

Concept-based video annotation is a multi-label classification (MLC) problem (one keyframe may be annotated with more than one semantic concepts). One way to solve this problem is to treat it as multiple independent binary classification problems where for each concept a model can be learned to distinguish keyframes where the concept appears from those where the concept does not appear. Given feature-based keyframe representations that have been extracted from different keyframes and also the ground-truth annotations for each keyframe (i.e. the concepts presented) any supervised machine learning algorithm that solves classification problems can be used in order to learn the relations between the low-level image representations and the high-level semantic concepts.

We can distinguish two main categories of visual features: hand-crafted features and features based on Deep Convolutional Networks (DCNN-based). With respect to hand-crafted features, binary (ORB [56]) and non-binary (SIFT [35], SURF [5]) local descriptors, as well as color extensions of them [60] have been examined for concept-based video annotation. Local descriptors are aggregated into global image representations by employing feature encoding techniques such as Fisher Vector (FV) [10] and VLAD [29]. With respect to DCNN-based features, one or more hidden layers of a pre-trained DCNN are typically used as a global keyframe repre-

sentation [64]. Several DCNN software libraries are available in the literature, e.g. Caffe [30], MatConvNet, TensorFlow [1] and different DCNN architectures have been proposed, e.g. AlexNet [31], VGGNet [64], GoogLeNet [72], ResNeXt [81], ResNet [26]. DCNN-based descriptors present high discriminative power and generally outperform local descriptors [59], [66].

The most commonly used machine learning algorithms are Support Vector Machines (SVM), Logistic Regression (LR) and Random Forests (RF). A recent trend in video annotation is to learn features directly from the raw keyframe pixels using DCNNs. DCNNs were derived from simple neural networks so here we will briefly explain how neural networks and subsequently deep networks work. Neural networks consist of artificial neurons that have learnable weights and biases. Neurons are connected to each other, each neuron receives some inputs from other neurons, and outputs a new signal, i.e. a value, that can be used to activate or deactivate other neurons connected to its output. Pairs of neurons are assigned with weights that represent their connection relation. In order to calculate the output value of a neuron, i.e. its activation, we calculate the weighted sum of the activations of all neurons that are fed into it. This sum is subsequently given as input to an activation function that outputs the final neuron's activation value. In a DCNN, neurons are arranged in layers with each neuron in a single layer being connected to all or a subset of neurons in the previous layer. The connections go only from lower to top layers and this is why DCNNs are also referred as feed forward networks. In a concept-based video annotation task a DCNN consists of an input layer, a number of intermediate layers, a.k.a. hidden layers, and the output layer. The input layer takes a keyframe, it forward propagates it to the hidden layers and based on the neurons that are activated, the keyframe's class labels are finally triggered in the output layer that consists of as many neurons as the number of concepts that the network aims to learn. A deep network has millions of parameters and for this reason a large set of inputs is needed to train the network without overfitting on the data. In addition, during training a loss function is used (e.g. hinge loss, softmax) in order to measure how well the network's output fits the real ground-truth values. Then, randomly selected keyframes are provided to it and the network's weights are adjusted based on the output that is returned in order to reduce the value of the loss function. To update the weights the popular technique of back-propagation is used. A few years before, training networks with many hidden layers was computationally infeasible. However, the great success on the development of powerful GPUs was a driver for the evolution of this field and now it is common to train networks with many hidden layers in hours or days.

The small number of labelled training examples is a common problem in video datasets, making it difficult to train a deep network from scratch without over-fitting its parameters on the training set [67]. For this reason, it is common to use transfer learning that uses the knowledge captured in a source domain in order to learn a target domain without caring about the improvement in the source domain. When a small-sized dataset is available for training a DCNN, a transfer learning technique is followed, where a conventional DCNN, e.g. [26], is firstly trained on a large-scale dataset and then the classification layer is removed, the DCNN is extended

by one or more fully-connected layers that are shared across all of the tasks, and a new classification layer is placed on top of the last extension layer (having size equal to the number of concepts that will be learned in the target domain). Then, the extended network is fine-tuned in the target domain [49]. Experiments presented in [49] show that extending by one or more fully-connected layers works better than simply re-learning some of the pre-trained fully connected layers.

1.2.1.2 Multi-task Learning and Structured Outputs

As described in Section 1.2.1.1, video concept annotation is a challenging multi-label classification problem that in recent years is typically addressed using DCNN models that choose a specific DCNN architecture [64, 26] and put a multi-label cost function on top of it [79, 78, 7]. As is the case in other multi-label problems, there exist relations between different concepts, and several methods attempt to model and leverage these relations so as to improve the performance or reduce the complexity of classification models that treat each concept independently. These methods can be roughly divided in two main categories. In the first category, methods that fall under the framework of multi-task learning (MTL), attempt to learn representations or classification models that, at some level, are shared between the different concepts (tasks) [2, 46, 45, 18, 11, 3, 90, 71, 41, 33, 87, 40, 82]. In the second category, methods that fall under the framework of structured-output prediction attempt to learn models that make multi-dimensional predictions that respect the structure of the output space using either label constraints or post-processing techniques [65, 80, 12, 15, 43, 50, 83, 50, 77, 76, 85, 36, 4, 37, 8, 73, 13, 70, 61, 14, 89, 40]. Label constraints refer to regularizations that are imposed on the learning system in order to exploit label relations (e.g. correlations) [50, 83, 88, 61, 14, 89, 40]. Post-processing techniques refer to re-calculating the concept prediction results using either meta-learning classifiers or other re-weighting schemes [65, 80, 12, 15, 43].

1.2.2 Methodology

As discussed in Section 1.2.1.1, the dominant approach for performing concept-based video annotation is training DCNN architectures where the concepts share features up to the very last layer, and then branch off to T different classification branches (using typically one layer), where T is the number of concepts [49]. However, in this way, the implicit feature-level relations between concepts, e.g. the way in which concepts such as a *car* and *motorcycle* share lower-level features modelling things like their wheels, are not directly considered. Also, in such architectures, the relations or inter-dependencies of the concepts at a semantic level, i.e. the fact that two specific concepts may often appear together or, inversely, the presence of the one may exclude the other, are also not directly taken into consideration.

In this section we present a DCNN architecture that addresses the problem of video/image concept annotation by exploiting concept relations at two different levels. More specifically it captures both implicit and explicit concept relations, i.e. both visual-level and semantic-level concept relations, as follows. First, implicit concept relations are modelled in a DCNN architecture that learns T concept-specific feature vectors that are themselves linear combinations of $k < T$ latent concept feature vectors. In this way, in the shared representations (i.e. the latent concepts feature vectors), higher-level concepts may share visual features - for example, concepts such as *car*, *motorcycle*, and *airplane* may share features encoding the *wheels* in their depiction [28]. This bears similarities to multi-task learning (MTL) schemes, like GO-MTL [33] and the two-sided network proposed in [40] that factorise the 2D weight matrix to encode concept specific features. However, in contrast to GO-MTL [33], in our case the factorization is achieved in two standard convolutional network layers, and in contrast to [40], our network does not only verify whether a certain concept that is given as input to the one side of the network is present in the video/image that is given as input to the other side. Instead, it provides scores for all concepts in the output, similar to classical multi-label DCNNs. Second, explicit concept relations are introduced by a new cost term, implemented using a set of standard CNN layers that penalise differences between the matrix encoding the correlations among the ground truth labels of the concepts, and the correlations between the concept label predictions of our network. In this way, we introduce constraints on the structure of the output space by utilizing the label correlation matrix - this explicitly captures, for example, the fact that *daytime* and *nighttime* are negatively correlated concepts. Both of the above types of relations are implemented using standard convolutional layers and are incorporated in a single DCNN architecture that can then be trained end-to-end with standard back-propagation. This method was originally presented in [42] and the source code is available on GitHub[2].

1.2.2.1 Problem Formulation and Method Overview

We consider a set of concepts $C = \{c_1, c_2, ..., c_T\}$ and a multi-label training set $\mathscr{P} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : \boldsymbol{x}_i \in \mathscr{X}, \boldsymbol{y}_i \in \{0,1\}^{T\times 1}, i = 1...N\}$, where $\boldsymbol{x}_i$ is a 3-channel keyframe/image, $\boldsymbol{y}_i$ is its ground-truth annotation (i.e. contains the T labels of the i-th keyframe/image), and N is the number of training examples. A video/image concept annotation system learns T supervised learning tasks, one for each target concept c_j, i.e. it learns a real-valued function $f : \mathscr{X} \rightarrow \mathscr{Y}$, where $\mathscr{Y} = [0,1]^{T\times N}$.

Figure 1.3 presents a DCNN architecture that exploits both implicit visual-level and explicit semantic-level concept relations for video/image concept annotation by building on ideas from MTL and structured output prediction, respectively. Specifically, Fig. 1.3 (i) shows a typical $(\Pi + 1)$-layer DCNN architecture, e.g. ResNet, that shares all the layers but the last one [64, 26]; Fig. 1.3 (ii) shows how the typ-

[2] https://github.com/markatopoulou/fvmtl-ccelc

Table 1.1: Definition of main symbols

Symbol	Definition
$\boldsymbol{x}$	A keyframe/image
$\boldsymbol{y}$	A vector containing the ground-truth concept annotations for a keyframe/image
N	The number of training keyframes/images
c	A concept
T	The number of concepts, i.e. number of tasks
$\hat{\boldsymbol{y}}$	A vector containing the concept prediction scores for a keyframe/image
$\boldsymbol{L_x}$	Latent concept feature vectors of a keyframe/image
$\boldsymbol{S}$	Concept-specific weight matrix, each column corresponds to a task containing the coefficients of the linear combination with $\boldsymbol{L_x}$
$\boldsymbol{L_x S}$	Concept-specific feature vectors incorporating information from k latent concept representations
$\boldsymbol{U}$	Concept-specific parameter matrix for the final classification
k	The number of latent tasks
d_1	The size of the output of the previous network layer
$\boldsymbol{\Phi}$	The concept correlation matrix calculated from the ground-truth annotated training set
$\boldsymbol{m}$	A cost vector utilised for data balancing

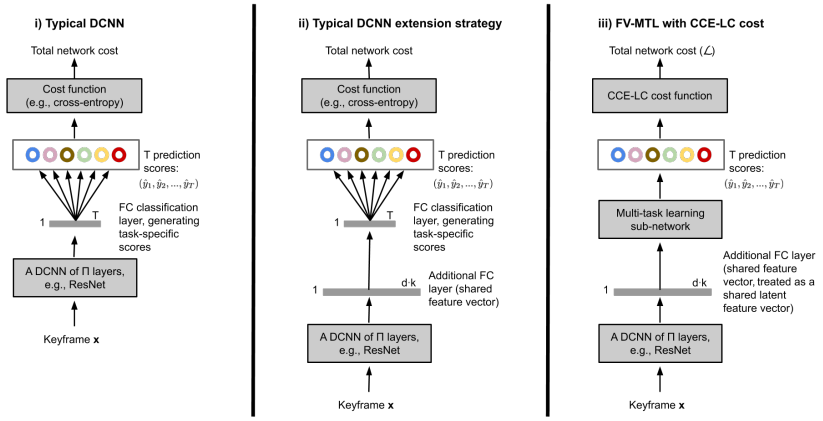

Fig. 1.3: Sub-figure (i) shows the typical DCNN architecture (e.g. ResNet [26]). Sub-figure (ii) shows the typical DCNN extension strategy proposed in [49]. Sub-figure (iii) presents the FV-MTL with CCE-LC cost function approach of [42].

ical DCNN architecture of Fig. 1.3 (i) can be extended by one FC extension layer, to improve the results in transfer learning problems [49]; and finally, Fig 1.3 (iii) shows the adopted DCNN architecture. In the next subsections we briefly introduce the parts of this architecture. For more details the interested reader can refer to our original paper [42]. Specifically, we first introduce the FV-MTL approach for learn-

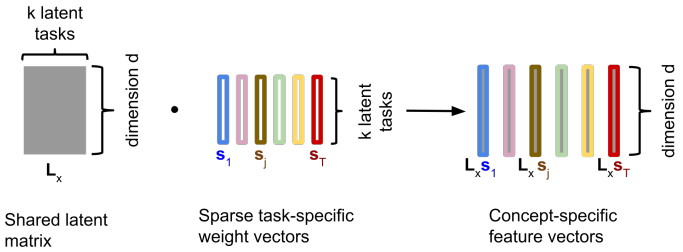

Fig. 1.4: Shared latent feature vectors using multi-task learning (FV-MTL).

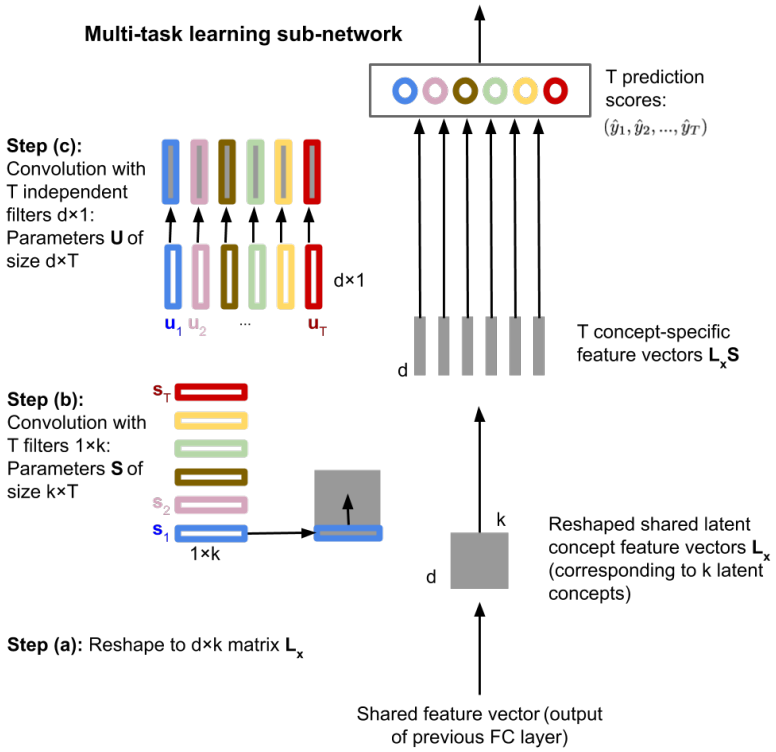

Fig. 1.5: MTL part of the proposed FV-MTL with CCE-LC cost function.

ing implicit visual-level concept relations; this is done using the multi-task learning sub-network shown in Fig. 1.3 and Fig. 1.5. Second, we introduce the CCE-LC cost function that learns explicit semantic-level concept relations. CCE-LC predicts

structured outputs by exploiting concept correlations that we can acquire from the ground-truth annotations of a training dataset.

1.2.2.2 Shared Latent Feature Vectors Using Multi-task Learning (FV-MTL)

In the FV-MTL approach, similarly to GO-MTL [33], we assume that the parameter vectors of the tasks that present visual-level concept relations lie in a low-dimensional subspace, thus sharing information; and, at the same time, dissimilar tasks may also partially overlap by having one or more bases in common. To allow this sharing of information, we learn T concept-specific feature vectors that are linear combinations of a small number of latent concept feature vectors that are themselves learned as well (Fig. 1.4). Specifically, we use a shared latent feature vector $\boldsymbol{L_x} \in \mathbb{R}^{d \times k}$ for all task models, where the columns of $\boldsymbol{L_x}$ correspond to d-dimensional feature representations of k latent tasks; and we produce T different concept-specific feature vectors $\boldsymbol{L_x s}_j$, for $j = 1...T$, where each of them incorporates information from relevant latent tasks, with $\boldsymbol{s}_j \in \mathbb{R}^{k \times 1}$ being a task-specific weight vector that contains the coefficients of the linear combination. Each linear combination is assumed to be sparse, i.e, $\boldsymbol{s}_j$'s are sparse vectors; in this way we assume that there exist a small number of latent basis tasks, and each concept-specific feature vector is a linear combination of them. The overlap between the weight vectors $\boldsymbol{s}_j$ and $\boldsymbol{s}_{j'}$ controls the amount of information-sharing between the corresponding tasks.

The above are implemented in a DCNN architecture by using the network layers depicted in Fig. 1.3 and Fig. 1.5. Specifically, an input training-set keyframe is processed by any chosen DCNN architecture (e.g. ResNet) and a fully-connected layer, to produce a shared representation of the keyframe across all of the tasks. Subsequently, the output of the fully-connected layer is reshaped to the matrix $\boldsymbol{L_x}$ (Fig. 1.5: step (a)); thus, the reshaped layer outputs k feature vectors that correspond to k latent concepts. Those representations are shared between the T concepts. The subsequent layer calculates T concept-specific feature vectors, where T is the number of the concepts we want to detect. Each of those feature vectors is a combination of k latent concept feature vectors, with coefficients that are specific to the concept in question. This is implemented as a 1D convolutional layer on the k feature masks (Fig. 1.5: step (b)). Once T feature vectors are extracted, then an additional layer (Fig. 1.5: step (c)) transforms each of the T feature vectors into T concept annotation scores, one for each of the concepts that we want to detect. This process leads to a *soft* feature sharing, because the latent concept feature vectors adjust how much information and across which tasks is shared. By contrast, both the typical DCNN and the DCNN extension architecture of [49] (Fig. 1.3 (i) and (ii)) output a single feature vector that is shared across all of the target concepts and it is subsequently *hard* translated into concept annotation scores independently for each concept. Finally, a sigmoid cross entropy cost term is used at the top of the network in order to optimize the sigmoid cross entropy between the predictions and the ground truth labels; we refer to this classification cost term as λ_1.

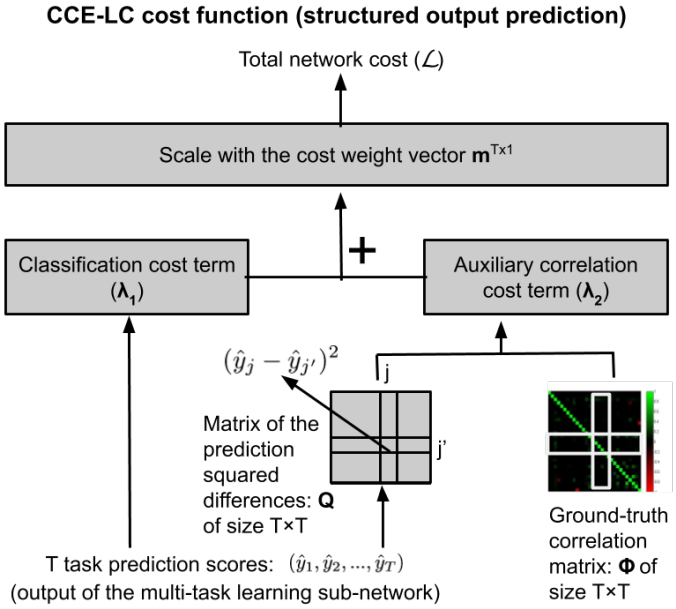

Fig. 1.6: Structured output prediction part of the proposed FV-MTL with CCE-LC cost function.

1.2.2.3 Label Constraints for Structured Output Prediction

The cross-entropy cost is not adequate for capturing semantic concept relations. For this reason in [42] we proposed an additional cost term that constitutes an effective way to integrate structural information. By structural information we refer to the inherently available concept correlations in a given ground-truth annotated collection of training videos/images. It should be noted that information from other external sources, such as WordNet [19] or other ontologies, could also be used but we have not tried it in our experiments. In order to consider this information we firstly calculate the correlation matrix $\boldsymbol{\Phi} \in [-1,1]^{T \times T}$ from the ground truth annotated data of the training set. Each position of this matrix corresponds to the ϕ-correlation coefficient between two concepts c_j, $c_{j'}$ calculated as discussed in [42]. The auxiliary concept correlation cost term uses the above correlation matrix $\boldsymbol{\Phi}$, however, the way that this term is formed is omitted here because this is out of the scope of this book chapter. It should only be noted that this term works as a label-based constraint and its role is to add a penalty to concepts that are positively correlated but were assigned with different concept annotation scores. Similarly, it adds a penalty to concepts that are negative-correlated but were not assigned with opposite annotation scores. Contrarily, it does not add a penalty to non-correlated concepts.

We implement the auxiliary concept correlation cost term, noted as λ_2, using a set of standard CNN layers, as presented in Fig. 1.6. One matrix layer encodes the correlations between the ground-truth labels of the concepts (denoted as $\boldsymbol{\Phi}$), and the other matrix layer contains the correlations between the concept label predictions of our network in the form of squared differences (denoted as $\boldsymbol{Q} \in \mathbb{R}^{T \times T}$, i.e. the matrix $\mathbf{Q}$ contains the differences of activations from the previous layer). Matrix $\mathbf{Q}$ gets multiplied, by element-wise multiplication, with the correlation matrix $\boldsymbol{\Phi}$, i.e. $\mathbf{Q} \circ \boldsymbol{\Phi}$, and all the rows in the resulting $T \times T$ matrix are added, leading to a single row vector.

1.2.2.4 FV-MTL with Cost Sigmoid Cross-entropy with Label Constraint (FV-MTL with CCE-LC)

The two cost terms discussed in Sections 1.2.2.2 and 1.2.2.3, and also denoted in Fig. 1.6 as λ_1 and λ_2 respectively, can be added in a single cost function that forms our total FV-MTL with CCE-LC network's cost. In our overall network architecture, an additional layer is used to implement the complete FV-MTL with CCE-LC cost function. In this way, the complete DCNN architecture learns by considering both the actual ground-truth annotations and also the concept correlations that can be inferred from it. In contrast, a typical DCNN architecture simply incorporates knowledge learned from each individual ground truth annotated sample. For more details on this cost function the interested reader can refer to our original paper [42].

1.2.3 Results

1.2.3.1 Datasets and Experimental Setup

Our experiments were performed on the TRECVID-SIN 2013 dataset [48]. For assessing concept annotation performance, the indexing problem as defined in [48] was evaluated, i.e. given a concept, the goal was to retrieve the 2000 video shots that are mostly related to it. The TRECVID-SIN 2013 [48] dataset consists of approximately 600 and 200 hours of Internet archive videos for training and testing, respectively. The training set is partially annotated with 346 semantic concepts. The test set is evaluated on 38 concepts for which ground truth annotations exist, i.e. a subset of the 346 concepts.

Since the available ground truth annotations for this dataset are not adequate in number in order to train a deep network from scratch without overfitting its parameters, similarly to other studies [49], we used transfer learning, i.e. we used as a starting point the ResNet-50 network [26], which was originally trained on 1000 ImageNet categories [58], and fine-tuned its parameters towards our dataset. In order to evaluate the methods' performance we used the mean extended inferred average

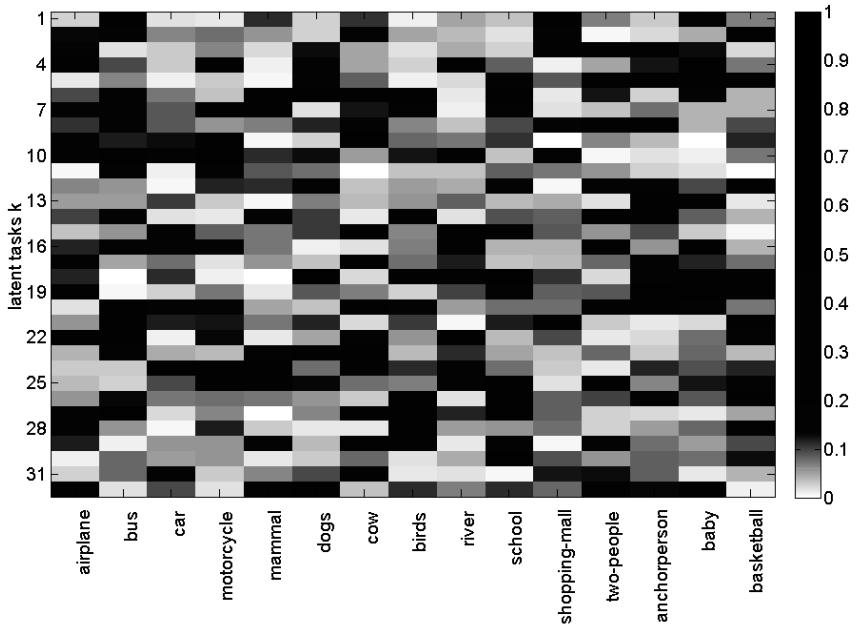

Fig. 1.7: Recovered sparsity patterns (the matrix $\boldsymbol{S}$) with FV-MTL with CCE-LC, for k equal to 32 and d equal to 64, for 15 selected concepts of the TRECVID-SIN 2013 dataset. Darker color indicates higher absolute value of the coefficient. The horizontal axis depicts the 15 observed concepts and the vertical axis the 32 latent tasks.

precision (MXinfAP) [84], which is an approximation of MAP. MXinfAP is suitable for the partial ground truth that accompanies this dataset.

1.2.3.2 Visual-level and Semantic-level Concept Relations of the Presented DCNN Srchitecture

According to our preliminary experimental results presented in our journal paper [42], FV-MTL with CCE-LC for k equal to 32 and d equal to 64 was the pair that reached the best overall MXinfAP. In this subsection we will try to visualise what this model has learned with respect to visual-level and semantic-level concept relations. As explained in 1.2.2.2, the overlap in the sparsity patterns of any two tasks, (i.e. the overlap between task-specific weight vectors $\boldsymbol{s}_j$ and $\boldsymbol{s}_{j'}$) controls the amount of sharing between them. Based on this in Fig. 1.7, we recovered sparsity patterns (the matrix $\boldsymbol{S}$) using FV-MTL with CCE-LC for 15 selected concepts of the TRECVID SIN dataset (darker color indicates higher absolute value of the coefficient). The horizontal axis depicts the 15 observed concepts and the vertical axis

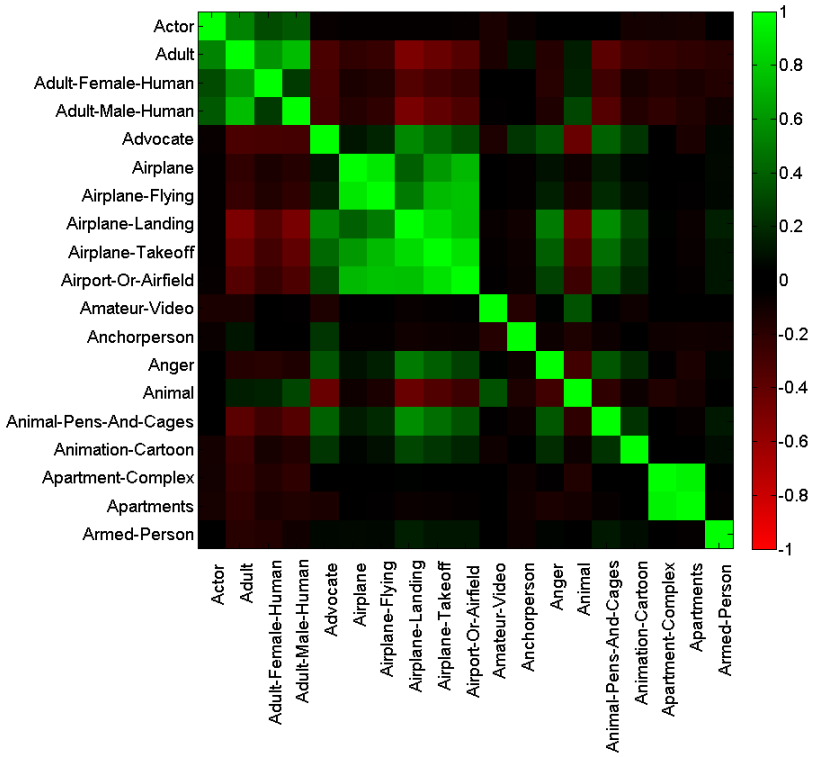

Fig. 1.8: Colormap of the phi-correlation coefficient calculated on the final prediction scores of the proposed FV-MTL with CCE-LC, for k equal to 32 and d equal to 64, when applied on the TRECVID SIN 2013 test dataset for 20 selected concepts.

the latent tasks (k=32) in this case. It is difficult to recover the grouping and overlap structure for the observed concepts based on this figure, but some interesting observations can be made. For example, concepts with the same sparsity pattern can be considered as belonging to the same group, while concepts with orthogonal sparsity patterns can be considered as belonging to different groups. The 9th and 10th latent tasks are always active for the transport-related concepts (e.g. airplane, car, bus, motorcycle) but they are inactive, at least one of the two, for any of the other concepts. Transport-related concepts can be considered as belonging to the same group. In addition, those latent tasks that are active for the concept "river" are always inactive for the concept "shopping-mall" (except for the 11th latent task), which indicates that these are two disjoint groups.

Regarding the semantic-level concept relations, Fig. 1.8 presents the color map of the phi-correlation coefficients, when calculated on the final prediction scores of the model when applied on the TRECVID SIN 2013 test dataset for 20 selected concepts. We can see that the model has captured many pairs of positive correlated

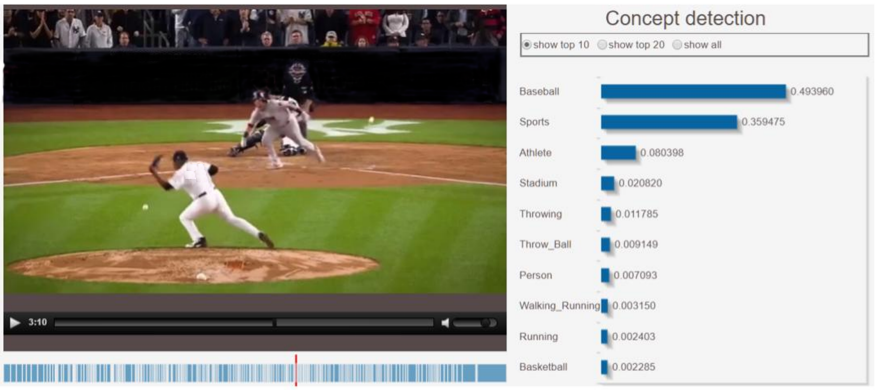

Fig. 1.9: Visual inspection of the results of our DCNN trained model, when applied on a specific video (downloaded from YouTube); here we are considering the concept-based keyframe annotation problem, i.e. whether we can annotate a given keyframe with the most relevant concepts.

concepts such as “adult”-“actor”, “adult”-“female human person” (green areas of the figure), pairs of negative correlated concepts such as “animal”-“airplane landing” (red areas of the figure), and non-correlated concepts such as “animal”-“actor”, “anger”-“actor” (black areas of the figure). According to the observations recovered from Figs. 1.7 and 1.8, we can see that our proposed method is able to capture both visual-level and semantic-level concept relations.

Finally, Fig. 1.9 and Fig. 1.10 present examples of concept-based keyframe annotation and retrieval results of our method, respectively. We can see that our method works very well for both problems retrieving correct results on top positions.

1.2.3.3 Main Findings - Comparisons with Related Methods

Figure 1.11 presents some of our main findings. The interested reader can refer to our original paper [42], where an extensive experimental evaluation has been performed. It should be noted that in [42] the FV-MTL with CCE-LC method, presented in this chapter, has been extensively evaluated and compared with many other concept-based video annotation methods. The compared methods have been categorised into three groups i) those that do not consider neither MTL nor SO, ii) those that either consider MTL or SO, and iii) those that consider both MTL and SO.

The FV-MTL with CCE-LC cost method presented in this chapter jointly exploits implicit visual-level and explicit semantic-level concept relations. This integrated DCNN architecture that emerges from combining these approaches was shown to improve concept annotation accuracy and outperformed the related state-of-the-art methods. Specifically, according to the left diagram of Fig. 1.11, it outperforms methods that do not impose any concept relations from 1.5% to 5%, methods that

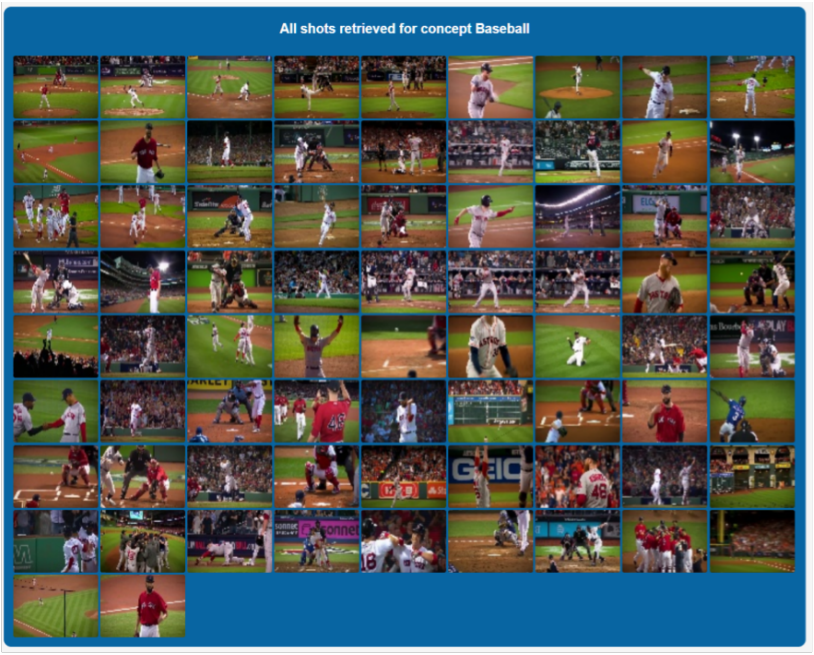

Fig. 1.10: Visual inspection of the results of our DCNN trained model, when applied on a specific video (downloaded from YouTube); here we are considering the concept-based keyframe retrieval problem, i.e. whether we can retrieve all the relevant keyframes of a video, for a given concept.

solely introduce either MTL or structured outputs by $\sim$2%, and finally methods that jointly consider MTL and structured outputs by $\sim$4%, in the TRECVID SIN dataset.

In addition, we evaluate the two intermediate versions of the integrated DCNN architecture (right part of Fig. 1.11): a) Extension strategy [49] for DCNNs with the proposed CCE-LC cost, i.e. the typical complete DCNN architecture illustrated in Fig. 1.3 (ii), but replacing the sigmoid cross-entropy cost with the proposed CCE-LC cost, and b) a subset of the FV-MTL with CCE-LC method, in which only the MTL part is used (i.e. without considering concept correlations). We observe that the two intermediate versions of our proposed method perform quite well; however, jointly considering both of them into a single architecture further improves the concept-based video retrieval accuracy.

To sum up, FV-MTL with CCE-LC always presents the best accuracy in terms of MXinfAP, which is equal to 33.77% (as presented on the right part of Fig. 1.11. All the other methods perform worse. Due to lack of space we did not present all these comparisons, so on the left part of Fig. 1.11 we show the performance of the second best method and also the performance of the worst method from each of the different groups evaluated in our original paper [42].

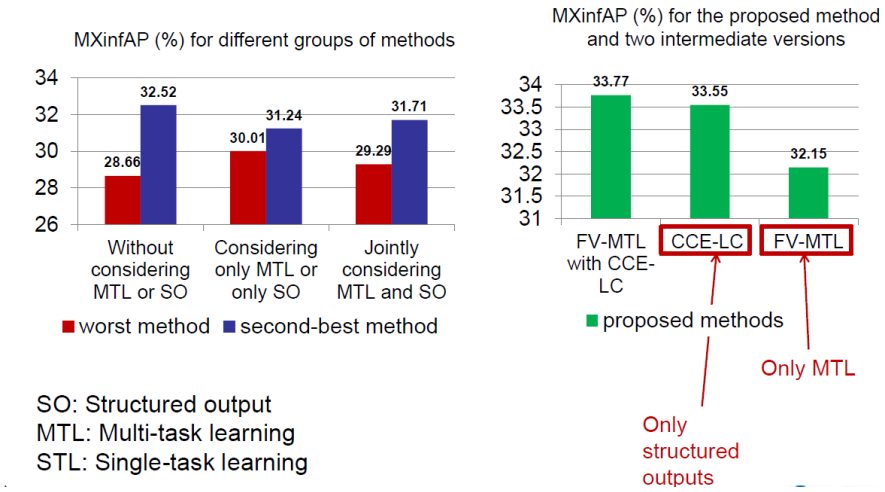

Fig. 1.11: Main findings on the TRECVID-SIN 2013 dataset. Evaluation in terms of MXinfAP.

Finally, it should be noted that a thorough analysis of the execution times of the proposed method appears in our original paper [42] that shows that our method is not considerably more computationally expensive than DCNN methods that use single-task learning cost functions. In terms of scalability, if we provide more concepts, then the network could model more and stronger task and label relations. So, we expect that the proposed method could work very well for larger number of concepts. In addition, in our preliminary experiments presented in the original paper of the method we have shown that parameters k and d are not sensitive to the number of concepts so the complexity of this part would not significantly increase when more concepts are to be learned. However, more experimentation towards this direction is needed.

1.3 Logo Detection

1.3.1 Related Work

The problem of detecting overlaid logos in videos is essentially a sub-problem of object detection in images. However, the problem definition of our particular case has a number of inherent constraints, which simplify the task, making it relatively easier than general object detection. By definition, object detection [17, 16] describes the task of identifying broad object categories (e.g. “helicopter”, “human”, “airplane”) in images. These object categories have extremely high within-class variation in

comparison to detecting overlaid video/image logos, which are expected to be near-identical in every instance they appear. In this sense, the problem is more relevant to the task of logo detection [27, 47], which, despite the common name, has certain differences from the InVID use case. In the task commonly referred to as *logo detection*, the aim is to detect trademark logos on depicted objects, e.g. the brand of a shoe or a poster on a building. This includes perspective distortions and variants of the same logo, which again make the task broader than what we have to tackle within InVID. Our problem concerns logos that typically appear on the screen at the same angle, usually at approximately the same size, and often at the same position. We will use the term *TV logo detection*, as it is established in literature, although it is clear that in our case we are often not dealing with actual TV channels and, in fact, the most important part of the task is identifying the logos of unofficial video sources, such as paramilitary groups. The case of TV logo detection is a much more narrow field than logo detection, and the respective methods exploit the constraints of the task to increase detection accuracy and speed.

Fig. 1.12: Top: an example of a generic logo detection task; Bottom: an example of the much more specific TV logo detection task.

The most common assumption of such methods is that the logo remains static throughout a video shot, while the rest of the frame contents change through time. Thus, approaches such as frame accumulation and thresholding [32] and brightness variance thresholding [86, 62] have been proposed to take advantage of these characteristics of the specific task. While seemingly a reasonable approach, a major issue with such approaches is that this assumption does not hold consistently, especially when dealing with arbitrary logos. It is not uncommon, for example, in the case of middle eastern paramilitary or clandestine political organizations to use animated logos (Fig. 1.13). In that case, any method based on the static logo assumption would fail entirely.

Fig. 1.13: Three frames from a Free Syrian Army clip displaying a rotating and changing logo.

We thus decided to explore more powerful and robust algorithms for the problem. The options we considered were drawn from current literature, namely keypoint-based methods, sliding windows, region proposal methods, and object detection Convolutional Neural Networks (CNNs).

Keypoint-based approaches have been quite popular in the past, and generally provided relatively robust solutions to the task [55, 54, 57, 34]. To learn a candidate logo, the algorithm extracts keypoints from a logo template, and retains their relative coordinates and local descriptors. For detection in an unknown image, keypoints are similarly extracted from the candidate image, and then their descriptors are matched against those of the candidate logos. Generally, these methods combine the matching of keypoints with some geometrical analysis of the feature location to ensure a match between the query and the candidates, and take into account possible geometrical transformations (in the case of logos positioned on objects, which may be distorted due to perspective.

Another option is a sliding window approach [9, 20], where a global descriptor is extracted from each logo. Then candidate overlapping windows are extracted from the image, at multiple scales, and the corresponding descriptor is extracted from each window. Consecutively, the descriptor is compared to the descriptors of all candidate logos. The comparison step is much faster than in keypoint-based methods, and can achieve much higher accuracy. However, due to the need of extracting descriptors from multiple overlapping windows, such approaches are prohibitively slow for real-time operational settings.

A much faster variant to sliding windows is region proposal. In that case, we can use a region proposal algorithm to extract a small number of candidate regions from the image, which are more likely to contain objects of interest (i.e. logos). We then only evaluate these regions [25, 6] as candidate windows. While faster than sliding window methods, these approaches often still require several seconds to propose the candidate windows for a single image. Furthermore, the success of the algorithm depends strongly on how strictly at least one of the proposed regions corresponds to the logo in the image. However, preliminary experiments showed that in many cases none of the proposed regions contained the logo, and thus the algorithms would simply not work in these cases.

Currently, the best performance in object detection is achieved using Deep Neural Networks, and specifically Region proposal Convolutional Neural Networks (R-CNN) [22, 53]. These methods train a region proposal network together with a

classification network, and are very fast in terms of detection time since they only require a single forward pass to return both classification and localization information. While Faster-RCNN remains a dominant architecture for object detection, other variants such as YOLO attempt to further reduce complexity [52], while recently a novel loss function was proposed to allow simpler and faster networks to reach the accuracy of R-CNN and its variants. However, a common issue with deep learning architectures is that they typically require a lot of annotated training data which are not easily available.

1.3.2 Methodology

In the first steps of the project, the possibility of using Deep Learning to solve the problem was not considered viable, since the large amount of annotated data that would be required by the system was unavailable. Thus, based on speed considerations, the first approach we opted to use was a point matching method that compared an image or video keyframe under investigation with a collection of stored logo templates. An implementation of a keypoint-based algorithm was developed and deployed for quantitative evaluations and as a first version of the service.

However, as the project progressed, it became apparent that the main limitation of the keypoint-based method was scalability. Each new template that would be added to the database would need to be compared to the image, which would lead to the computational cost rising linearly with the number of known logos. Thus, during the second year of InVID we decided to move to a Deep Learning solution which, due to the parallelised, single-pass nature of the model, would retain its time complexity constant, independent of the number of known logos. This was combined with an innovative solution that generated ample artificial training examples using data augmentation, to address the need of the network for large numbers of annotated examples.

Both implementations are image-based. They were designed to deal with both single images and video frames. For videos, the approach relies on integration with the Video Fragmentation component of InVID, and operates at the shot level by processing keyframes of each shot and producing a separate set of estimates per shot. The approach allows for multiple logos to be present in the same image or shot. The main reason for this is that, as videos are shared or re-transmitted, organizations may add their own logos alongside the original ones, and the identities of all agencies involved may be important information for an investigator. In the case of videos, since each shot may contain different logos, the detection was done on a per-shot basis. In that case, the detection process can take place on one or more individual keyframes per shot.

1.3.2.1 Keypoint-based Method

In our keypoint-based implementation, detection is performed against a list of logo templates. Each potentially detectable logo is linked with one template image and a corresponding database entry. Each entry contains the name of the organization the logo belongs to, the name of the corresponding template image file, the link to the Wikipedia article corresponding to the organization/group, and the dimensions of the frame size from which the logo template was extracted. While following an inherently multi-scale representation, the advantage of dealing with TV logos is that, in the vast majority of cases, they tend to appear at roughly the same dimensions in the frame. In our preliminary investigations we found that resizing all candidate images to roughly the same dimensions as the image from which the template was extracted can provide a performance boost without increasing the computational complexity, while the fact that we use a scale-invariant representation means that we can still deal with certain scale variations.

For each logo in our list, we applied a pre-processing step where we extracted SURF [5] features from the corresponding template image. The features and the corresponding logo information were then stored to be used for detection. At detection time, when presented with a new candidate image, the image is rescaled to the appropriate dimensions, and SURF features are also extracted from it. Feature matching is then performed between the logo template and the candidate image using a k-nearest neighbors approach. The process is repeated for all logos in the list, and only logos returning a number of matches $\geqslant M$ are retained, where M is an experimentally determined threshold.

For all logos where a sufficient number of matching points are found, a second-level processing step takes place, where the geometrical consistency of the matches is evaluated. A RANSAC approach is then used to find the optimal perspective projection modelling the point correspondences, and to keep only the subset of matched points that conformed to the model. If the number of points surpasses a second threshold N (in our current implementation, $M = N$), the logo is considered to exist in the image.

For the keypoint-based implementation, it was found to be beneficial for accuracy to get more than one keyframe per shot and consecutively average them, to get a more salient version of the logo. Given a static logo, we can assume that in the averaged image the logo will remain intact while the rest of the image will appear blurred-out. As a result, this will produce fewer salient keypoints in the overall image. Having fewer candidate SURF points in the image means much smaller chance of false matches. However, this option can only work for static logos. Thus while this approach was used in the method's offline evaluations, it had to be abandoned during integration with InVID, and instead the middle keyframe of each shot was used for detection.

1.3.2.2 Deep Learning Method

As network architecture, we chose the Faster Region-proposal Convolutional Neural Network (Faster-RCNN) [53]. This architecture simultaneously outputs a large number of region proposals and classification estimates for each region in a single pass, making it extremely fast during detection. Furthermore, the region proposal and the classification parts are trained simultaneously, making its training faster than its counterparts. Its performance is among the best in the state-of-the-art, and open-source implementations exist for the Caffe[3] and Tensorflow[4] frameworks. Thus, it was straightforward to experiment and adapt to the project's needs.

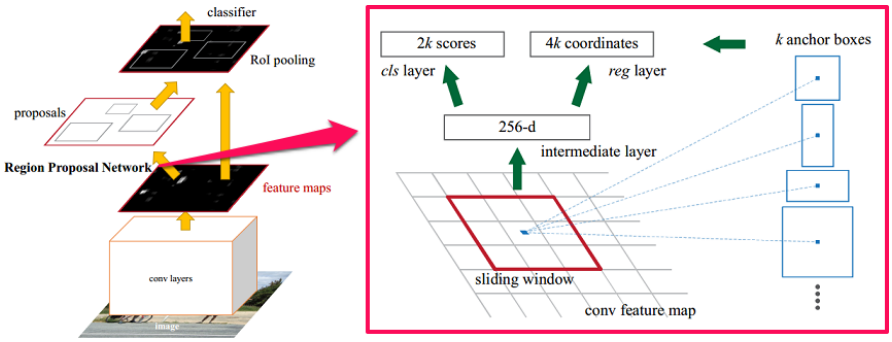

Fig. 1.14: The Faster-RCNN architecture (image taken from [53]).

The main challenge with Deep Neural Networks is training. They tend to require large amounts of annotated data and generally require a lot of time to train. However, the task at hand is significantly simpler than most detection tasks, since in our case the candidate object (i.e. a logo) has very little variability between instances. This characteristic allowed us to use an innovative training technique that removes the need for manually annotated data.

Normally for an object detection task, we would require a large number of annotated images (hundreds or thousands) containing each object, and each image would have to be annotated with the class and the localization coordinates of the object. Creating such a training dataset for logos would be impossible within the scope of InVID, and impractical when extending the dataset with new logos. However, since in our case all appearances of a logo would be expected to be very similar, we were able to devise a way to automatically generate training images on-the-fly using a single logo example. A training image can be generated by taking a random base image from any realistic image dataset (such as MIRFlickr[5]), and a logo from our collection, and placing the logo at a random position on the image. To ac-

[3] https://github.com/rbgirshick/py-faster-rcnn/

[4] https://github.com/tensorflow/models/tree/master/research/object_detection/

[5] http://press.liacs.nl/mirflickr/

count for variations of the logo, a set of data augmentation techniques are applied, such as scaling (sometimes non-proportional), blurring by a random-sized kernel, brightness and color modification. In this way, we can generate a practically infinite number of training images. To further speed up the training process, we place a number of logos in each training image, in a non-overlapping manner, ranging from 1 to 3 (Fig. 1.15). This process allows us to train a classifier without going through the process of collecting and manually annotating a dataset. It also allows for extensibility, since adding a new entry in the list of known logos does not require additional examples, but only the new logo template. It also solves the problem of a channel using many variants of its logo, since the addition of more examples adds little complexity to the task. Similarly, in the case of animated logos such as those depicted in Fig. 1.13, it is easy to add multiple frames from the animation into the known logo dataset, which means that we will be able to match the logo in a random keyframe from the video, whatever part of the animation it may contain. It should be noted that, roughly at the same time that we were implementing this approach, a publication presented a very similar data generation method for logo detection [69].

Following training, the detection process is simple: an image is passed through the trained model, and the model outputs a list of region estimates, plus the estimate of the logo class that was detected within them.

A further advantage of the CNN-based approach is that it is much more robust with respect to the logo background and potential transparency. When the keypoint-based approach detected points along the logo borders (generally a common case), the corresponding local descriptors were also affected by the color of the background. In some cases, it was necessary to extend the template collection with instances of the same logo over different backgrounds, to cover all such eventualities. Furthermore, in the keypoint-based algorithm both the keypoint detection and the description steps were strongly affected by semi-transparent logos, which returned different results depending on the background they appeared on. In contrast, the CNN-based approach can easily learn to recognise such logos, provided it can be trained with enough artificially generated examples containing the logo overlaid on different backgrounds.

Fig. 1.15: Three artificially generated training samples with logos, some of which are semi-transparent.

1.3.3 Results

Both approaches were implemented in Python. The SURF-based keypoint-based approach was implemented using methods from the OpenCV[6] library, and fast feature matching was done with the KDTree algorithm using OpenCV's FLANN-based[7] implementation. Faster-RCNN was implemented using existing frameworks, namely the Caffe-based `py-faster-rcnn`[8] implementation during the second year of the project, and the TensorFlow object detection framework[9] during the third year. TensorFlow has comparably fewer library dependencies, its Python integration is simpler, and the object detection framework is part of the official release, unlike `py-faster-rcnn` which is a custom adaptation. Thus, as InVID approached its final release stage, TensorFlow was a more reliable choice for future maintenance.

The template list developed for the keypoint-based approach contained 503 logo templates from 169 channels and news sources. The number of logo templates was generally higher than for the deep learning approach, since in the keypoint-based approach we need to include many more variants of the templates (e.g. with different backgrounds). For the Faster-RCNN method, all the logos associated with a particular channel were grouped together, thus the network had 169 classes for our evaluation experiments. The training of the Faster-RCNN was done with roughly 200,000 automatically generated examples.

The evaluation dataset we used consisted of 2,752 videos originating from various YouTube channels, containing videos that featured at least one of the known logos in at least one shot. The videos were then separated in 54,986 shots using the InVID Video Fragmentation and Annotation service.

Table 1.2 shows the comparison between the performance of the keypoint-based approach and two CNN models. As shown in Table 1.2, the Faster-RCNN version of the algorithm is currently comparable to the keypoint-based approach. We tested two RCNN models, one trained with early stopping (Fr-RCNN 1) and one trained for longer period of time (Fr-RCNN 2). Fr-RCNN 1 shows slightly lower True Detection (TD) rates than keypoint-based methods, and comparable False Detections (FD). On the other hand, Fr-RCNN 2 has better TD rates, but significantly higher FD rates. One explanation is that the logo template collection contains several images of relatively low quality that are blurred. For the keypoint-based method these were necessary, in order to be able to detect logos in low-quality images. However, in Faster-RCNN training, especially after the potential additional blurring of the augmentation step, the network might be trained on extremely blurred templates, which could lead to finding false matches on arbitrary regions. Another observation is that the false positives appear disproportionately higher per video than per shot. This means that the relatively few false positives in the shots (0.01) are very scattered across the shots, with few (usually one at most) in each video. Thus in practice

[6] http://opencv.org/

[7] http://www.cs.ubc.ca/research/flann/

[8] https://github.com/rbgirshick/py-faster-rcnn

[9] https://github.com/tensorflow/models/tree/master/research/object_detection/

these spurious matches are not distracting for professionals, since they can be easily discarded by visual inspection.

Table 1.2: Logo detection evaluation results

	Videos			Shots		
	Keypoints	Fr-RCNN 1	Fr-RCNN 2	Keypoints	Fr-RCNN 1	Fr-RCNN 2
True Detections	0.83	0.80	0.85	0.63	0.64	0.72
False Detections	0.06	0.06	0.13	0.004	0.01	0.01

Overall, we consider the Faster-RCNN approach to be a superior choice, for two reasons: 1) the results for Faster-RCNN have significant potential for improvement by improving the template dataset – with the help of the user partners – and by tweaking the training parameters, and 2) the Faster-RCNN approach is significantly faster, and its detection speed is much less dependent on the number of logos that are possible to detect. To confirm this hypothesis, we ran a series of evaluations with respect to detection speed. For fairness, we had to account for certain additional computational costs that the Faster-RCNN algorithm requires. Specifically, as the neural network runs on a PC equipped with a GPU, it had to be placed on a separate server, and it is possible that the communication between the logo detection server and the neural network server may incur additional delays. This means that the reported times include the service communication delays, which reflects the actual user experience. Table 1.3 gives the current differences in speed between the two services, per single image, per video shot, and per video. The reasons that the performance per shot is improved more than the performance per image, is that a) the keypoint-based method was run on both the middle image and the mean image of the shot in order to reach its optimal performance, while the Faster-RCNN algorithm only runs on the middle image of each shot and b) the impact of the communication overhead is much smaller, since the major load is accessing the image/video, which only happens once per video. In fact, the speed of the new service is so superior that it outweighs even the added time requirements of fragmenting the video (which we do not have in images), leading to the much higher per-shot improvement compared to the per-image one.

Table 1.3: Logo detection time requirements (in seconds)

	Image	**Shot**	**Video**
Keypoint-based	8.47	6.56	383.50
Faster-RCNN	4.17	1.18	69.00
Speedup	203%	556%	556%

While it is conceivable that adding many new logos may increase training time, we consider that any potential increase will be manageable. Furthermore, it is pos-

sible that the overall training time can be reduced by tweaking the training hyperparameters and improving the data augmentation procedure.

1.4 Conclusions and Future Work

In this chapter, we explored two tasks to assist investigators in identifying and organizing semantically related items in unstructured collections gathered from the Web in order to assist verification. One component was a concept-based annotation system, while the other was a logo detection system. Following an analysis of the state of the art, a choice of the most relevant approaches, and significant improvements, refinements and innovations beyond state-of-the-art methods, both systems were implemented and integrated in the final InVID platform.

Specifically, with respect to concept detection we presented a machine learning architecture that is based on deep learning, referring to it as FV-MTL with CCE-LC. Overall, the lesson we learned is that a good video annotation and retrieval architecture can be developed by carefully taking into account many different directions such as feature extraction, classifier combination, feature-level and semantic-level concept relations. Deep learning architectures are the best way of jointly considering all these, with the presented FV-MTL with CCE-LC deep architecture consistently outperforming other related state-of-the-art approaches.

With respect to logo detection, we designed an innovative way to generate enough training data in order to fine-tune existing object detection systems to the task of logo detection, even in the absence of a large annotated training set. Given than the InVID logo detection component will have to be kept up-to-date by adding new logos submitted by users, such an approach is the only way to be able to extend the classifier in the future. Since research into data augmentation is still ongoing, and recent methods based on Generative Adversarial Networks have yielded very promising results [51, 38], it might be a promising future path with respect to improving the classification accuracy of the system.

For our next steps, we will continue to advance both components, to improve their efficiency and accuracy. With respect to semantic video annotation, we will continue to experiment with deep learning architectures, to exploit concept relations and yield better accuracy, and we will also reconsider whether the TRECVID semantic concepts are the most appropriate for the task or another set of concepts (given a correspondingly annotated dataset for training) would be more appropriate for the needs of newsworthy video annotation and retrieval. With respect to logo detection, we will keep experimenting with the automatic training data generation process in order to improve the performance of the algorithm. We will also continue expanding the known logo dataset with user-submitted logos. Finally, we will attempt to expand the synthetic training data creation process by introducing perspective-like transforms to the logos. This will allow us to move from detecting overlaid logos to detecting logos within the scene, e.g. on clothing or walls. Such an extension of the component capabilities would empower journalists to have a more

complete understanding of the provenance and history of the video, and even allow them to verify aspects of the depicted content and the associated claims.

References

1. Abadi, M., Agarwal, A., Barham, P., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). URL `https://www.tensorflow.org/`. Software available from tensorflow.org
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. Advances in Neural Information Processing Systems (NIPS 2007) (2007)
3. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning **73**(3), 243–272 (2008)
4. Baumgartner, M.: Uncovering deterministic causal structures: a boolean approach. Synthese **170**(1), 71–96 (2009)
5. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). Computer Vision and Image Understanding **110**(3), 346–359 (2008). DOI 10.1016/j.cviu.2007.09.014. URL `http://dx.doi.org/10.1016/j.cviu.2007.09.014`
6. Bianco, S., Buzzelli, M., Mazzini, D., Schettini, R.: Logo recognition using CNN features. In: Proc. of 2015 International Conference on Image Analysis and Processing, pp. 438–448. Springer (2015)
7. Bishay, M., Patras, I.: Fusing multilabel deep networks for facial action unit detection. In: Proc. of the 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG) (2017)
8. Cai, X., Nie, F., Cai, W., Huang, H.: New graph structured sparsity model for multi-label image annotations. In: Proc. of the IEEE International Conference on Computer Vision (ICCV 2013), pp. 801–808 (2013)
9. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007), pp. 1–8. IEEE (2007)
10. Csurka, G., Perronnin, F.: Fisher vectors: Beyond bag-of-visual-words image representations. In: P. Richard, J. Braz (eds.) Computer Vision, Imaging and Computer Graphics. Theory and Applications, *Communications in Computer and Information Science*, vol. 229, pp. 28–42. Springer Berlin (2011)
11. Daumé III, H.: Bayesian multitask learning with latent hierarchies. In: Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), pp. 135–142. AUAI Press, Quebec, Canada (2009)
12. Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., Adam, H.: Large-Scale Object Classification Using Label Relation Graphs, pp. 48–64. Springer, Zrich, Switzerland (2014)
13. Deng, J., Satheesh, S., Berg, A.C., Li, F.: Fast and balanced: Efficient label tree learning for large scale object recognition. In: Advances in Neural Information Processing Systems, pp. 567–575. Curran Associates, Inc. (2011)
14. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. CoRR **abs/1511.04196** (2015)
15. Ding, N., Deng, J., Murphy, K.P., Neven, H.: Probabilistic label relation graphs with ising models. In: Proc. of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), pp. 1161–1169. IEEE, Washington, DC, USA (2015)
16. Dollár, P., Appel, R., Belongie, S.J., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1532–1545 (2014)
17. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 2155–2162. IEEE Computer Society (2014)
18. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 109–117. Seattle, WA (2004)
19. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)

20. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(1), 36–51 (2008)
21. Galanopoulos, D., Markatopoulou, F., Mezaris, V., Patras, I.: Concept language models and event-based concept number selection for zero-example event detection. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR '17, pp. 397–401. ACM, New York, NY, USA (2017). DOI 10.1145/3078971.3079043. URL `http://doi.acm.org/10.1145/3078971.3079043`
22. Girshick, R.: Fast R-CNN. In: Proc. of the IEEE International Conference on Computer Vision (ICCV 2015), pp. 1440–1448 (2015)
23. Gkalelis, N., Mezaris, V.: Incremental accelerated kernel discriminant analysis. In: Proceedings of the 25th ACM International Conference on Multimedia, MM '17, pp. 1575–1583. ACM, New York, NY, USA (2017). DOI 10.1145/3123266.3123401. URL `http://doi.acm.org/10.1145/3123266.3123401`
24. Gkalelis, N., Mezaris, V., Kompatsiaris, I.: A joint content-event model for event-centric multimedia indexing. In: IEEE International Conference on Semantic Computing (ICSC), pp. 79–84 (2010)
25. Gu, C., Lim, J.J., Arbeláez, P., Malik, J.: Recognition using regions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 1030–1037. IEEE (2009)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 770–778 (2016). DOI 10.1109/CVPR.2016.90
27. Hoi, S.C.H., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., Wu, Q.: LOGO-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. CoRR **abs/1511.02462** (2015). URL `http://arxiv.org/abs/1511.02462`
28. Jalali, A., Sanghavi, S., Ruan, C., Ravikumar, P.K.: A dirty model for multi-task learning. In: Advances in Neural Information Processing Systems, pp. 964–972. Curran Associates (2010)
29. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3304–3311. IEEE (2010)
30. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proc. of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
32. Ku, D., Cheng, J., Gao, G.: Translucent-static TV logo recognition by SUSAN corner extracting and matching. In: Innovative Computing Technology (INTECH), 2013 Third International Conference on, pp. 44–48. IEEE (2013)
33. Kumar, A., Daume, H.: Learning task grouping and overlap in multi-task learning. In: the 29th ACM International Conference on Machine Learning (ICML 2012), pp. 1383–1390. Edinburgh, Scotland (2012)
34. Le, V.P., Nayef, N., Visani, M., Ogier, J.M., De Tran, C.: Document retrieval based on logo spotting using key-point matching. In: 2014 22nd International Conference on Pattern Recognition, pp. 3056–3061. IEEE (2014)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
36. Lu, Y., Zhang, W., Zhang, K., Xue, X.: Semantic context learning with large-scale weakly-labeled image set. In: Proc. of the 21st ACM International Conference on Information and Knowledge Management, pp. 1859–1863. ACM, NY, USA (2012)
37. Luo, Q., Zhang, S., Huang, T., Gao, W., Tian, Q.: Superimage: Packing semantic-relevant images for indexing and retrieval. In: Proc. of the International Conference on Multimedia Retrieval (ICMR 2014), pp. 41–48. ACM, NY, USA (2014)
38. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C.: Bagan: Data augmentation with balancing GAN. arXiv preprint arXiv:1803.09655 (2018)

39. Markatopoulou, F., Galanopoulos, D., Mezaris, V., Patras, I.: Query and keyframe representations for ad-hoc video search. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR '17, pp. 407–411. ACM, New York, NY, USA (2017). DOI 10.1145/3078971.3079041. URL http://doi.acm.org/10.1145/3078971.3079041
40. Markatopoulou, F., Mezaris, V., Patras, I.: Deep multi-task learning with label correlation constraint for video concept detection. In: Proc. of the International Conference ACM Multimedia (ACMMM 2016), pp. 501–505. ACM, Amsterdam, The Netherlands (2016)
41. Markatopoulou, F., Mezaris, V., Patras, I.: Online multi-task learning for semantic concept detection in video. In: Proc. of the IEEE International Conference on Image Processing (ICIP 2016), pp. 186–190 (2016)
42. Markatopoulou, F., Mezaris, V., Patras, I.: Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. IEEE Transactions on Circuits and Systems for Video Technology **29**(6), 1631–1644 (2019). DOI 10.1109/TCSVT.2018.2848458
43. Markatopoulou, F., Mezaris, V., Pittaras, N., Patras, I.: Local features and a two-layer stacking architecture for semantic concept detection in video. IEEE Transactions on Emerging Topics for Computing **3**, 193–204 (2015)
44. Markatopoulou, F., Moumtzidou, A., Tzelepis, C., Avgerinakis, K., Gkalelis, N., Vrochidis, S., Mezaris, V., Kompatsiaris, I.: ITI-CERTH participation to TRECVID 2013. In: TRECVID 2013 Workshop, Gaithersburg, MD, USA, vol. 1, p. 43 (2013)
45. Mousavi, H., Srinivas, U., Monga, V., Suo, Y., Dao, M., Tran, T.: Multi-task image classification via collaborative, hierarchical spike-and-slab priors. In: the IEEE International Conference on Image Processing (ICIP 2014), pp. 4236–4240. Paris, France (2014)
46. Obozinski, G., Taskar, B.: Multi-task feature selection. In: the 23rd International Conference on Machine Learning (ICML 2006). Workshop of Structural Knowledge Transfer for Machine Learning. Pittsburgh, Pennsylvania (2006)
47. Oliveira, G., Frazão, X., Pimentel, A., Ribeiro, B.: Automatic graphic logo detection via fast region-based convolutional networks. CoRR **abs/1604.06083** (2016). URL http://arxiv.org/abs/1604.06083
48. Over, P., et al.: TRECVID 2013 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In: TRECVID 2013. NIST, USA (2013)
49. Pittaras, N., Markatopoulou, F., Mezaris, V., Patras, I.: Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks. In: Proc. of the 23rd International Conference on MultiMedia Modeling (MMM 2017), pp. 102–114. Springer, Reykjavik, Iceland (2017)
50. Qi, G.J., et al.: Correlative multi-label video annotation. In: Proc. of the 15th International Conference on Multimedia, pp. 17–26. ACM, NY (2007)
51. Ratner, A.J., Ehrenberg, H., Hussain, Z., Dunnmon, J., Ré, C.: Learning to compose domain-specific transformations for data augmentation. In: Advances in Neural Information Processing Systems, pp. 3236–3246 (2017)
52. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 779–788 (2016)
53. Ren, S., He, K., Girshick, R.B., 0001, J.S.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp. 91–99 (2015). URL http://papers.nips.cc/book/advances-in-neural-information-processing-systems-28-2015
54. Revaud, J., Douze, M., Schmid, C.: Correlation-based burstiness for logo retrieval. In: Proc. of the 20th ACM international conference on Multimedia, pp. 965–968. ACM (2012)
55. Romberg, S., Lienhart, R.: Bundle min-hashing for logo recognition. In: Proc. of the 3rd ACM International Conference on Multimedia Retrieval, pp. 113–120. ACM (2013)
56. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proc. of the IEEE International Conference on Computer Vision (ICCV 2011), pp. 2564–2571 (2011)

57. Rusinol, M., Llados, J.: Logo spotting by a bag-of-words approach for document categorization. In: 2009 10th international conference on document analysis and recognition, pp. 111–115. IEEE (2009)
58. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV 2015) **115**(3), 211–252 (2015). DOI 10.1007/s11263-015-0816-y
59. Safadi, B., Derbas, N., Hamadi, A., Budnik, M., Mulhem, P., Qu, G.: LIG at TRECVid 2014 : Semantic Indexing tion of the semantic indexing. In: TRECVID 2014 Workshop. Gaithersburg, MD, USA (2014)
60. Van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1582–1596 (2010)
61. Schwing, A.G., Urtasun, R.: Fully connected deep structured networks. CoRR **abs/1503.02351** (2015)
62. Shen, L., Wu, W., Zheng, S.: TV logo recognition based on luminance variance. In: IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012), pp. 1–4. IET (2012)
63. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal video segmentation to scenes using high-level audiovisual features. IEEE Transactions Cir. and Sys. for Video Technol. **21**(8), 1163–1177 (2011). DOI 10.1109/TCSVT.2011.2138830. URL `http://dx.doi.org/10.1109/TCSVT.2011.2138830`
64. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014)
65. Smith, J., Naphade, M., Natsev, A.: Multimedia semantic indexing using model vectors. In: Proc. of the International Conference on Multimedia and Expo (ICME 2003), pp. 445–448. IEEE, NY (2003). DOI 10.1109/ICME.2003.1221649
66. Snoek, C., Sande, K., Fontijne, D., Cappallo, S., Gemert, J., Habibian, A., Mensink, T., Mettes, P., Tao, R., Koelma, D., et al.: Mediamill at trecvid 2014: Searching concepts, objects, instances and events in video (2014)
67. Snoek, C.G.M., Cappallo, S., Fontijne, D., Julian, D., Koelma, D.C., Mettes, P., van de Sande, K.E.A., Sarah, A., Stokman, H., Towal, R.B.: Qualcomm Research and University of Amsterdam at TRECVID 2015: Recognizing Concepts, Objects, and Events in Video. In: Proc. of TRECVID 2015. NIST, USA (2015)
68. Snoek, C.G.M., Worring, M.: Concept-Based Video Retrieval. Foundations and Trends in Information Retrieval **2**(4), 215–322 (2009)
69. Su, H., Zhu, X., Gong, S.: Deep learning logo detection with data expansion by synthesising context. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 530–539. IEEE (2017)
70. Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H., Larra, P.: Multi-label classification with bayesiannetwork-based chain classifiers. Pattern Recognition Letters **41**, 14 – 22 (2014)
71. Sun, G., Chen, Y., Liu, X., Wu, E.: Adaptive multi-task learning for fine-grained categorization. In: Proc. of the IEEE International Conference on Image Processing (ICIP 2015), pp. 996–1000 (2015)
72. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 1–9 (2015)
73. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Proc. of the 16th International Conference on Neural Information Processing Systems (NIPS 2003). MIT Press (2003)
74. Tzelepis, C., Galanopoulos, D., Mezaris, V., Patras, I.: Learning to detect video events from zero or very few video examples. Image Vision Comput. **53**(C), 35–44 (2016). DOI 10.1016/j.imavis.2015.09.005. URL `https://doi.org/10.1016/j.imavis.2015.09.005`

75. Tzelepis, C., Ma, Z., Mezaris, V., Ionescu, B., Kompatsiaris, I., Boato, G., Sebe, N., Yan, S.: Event-based media processing and analysis. Image Vision Comput. **53**(C), 3–19 (2016). DOI 10.1016/j.imavis.2016.05.005. URL `https://doi.org/10.1016/j.imavis.2016.05.005`
76. Wang, H., Huang, H., Ding, C.: Image annotation using multi-label correlated green's function. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 2029–2034 (2009)
77. Wang, H., Huang, H., Ding, C.: Image annotation using bi-relational graph of images and semantic labels. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp. 793–800 (2011)
78. Wang, X., Zheng, W.S., Li, X., Zhang, J.: Cross-scenario transfer person reidentification. IEEE Transactions on Circuits and Systems for Video Technology **26**(8), 1447–1460 (2016)
79. Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: Hcp: A flexible cnn framework for multi-label image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(9), 1901–1907 (2016). DOI 10.1109/TPAMI.2015.2491929
80. Weng, M.F., Chuang, Y.Y.: Cross-Domain Multicue Fusion for Concept-Based Video Indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(10), 1927–1941 (2012)
81. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. arXiv preprint arXiv:1611.05431 (2016)
82. Yang, Y., Hospedales, T.M.: A unified perspective on multi-domain and multi-task learning. In: the International Conference on Learning Representations (ICLR 2015). San Diego, California (2015)
83. Yang, Y., Wu, F., Nie, F., Shen, H.T., Zhuang, Y., Hauptmann, A.G.: Web and personal image annotation by mining label correlation with relaxed visual graph embedding. IEEE Transactions on Image Processing **21**(3), 1339–1351 (2012)
84. Yilmaz, E., Kanoulas, E., Aslam, J.A.: A simple and efficient sampling method for estimating ap and ndcg. In: the 31st ACM International Conference on Research and Development in Information Retrieval (SIGIR 2008), pp. 603–610. Singapore (2008)
85. Zhang, M.L., Zhang, K.: Multi-label learning by exploiting label dependency. In: Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010), pp. 999–1008. ACM, NY, USA (2010)
86. Zhang, X., Zhang, D., Liu, F., Zhang, Y., Liu, Y., Li, J.: Spatial HOG based TV logo detection. In: K. Lu, T. Mei, X. Wu (eds.) International Conference on Internet Multimedia Computing and Service, ICIMCS '13, Huangshan, China - August 17 - 19, 2013, pp. 76–81. ACM (2013)
87. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: the 13th Europ. Conference on Computer Vision (ECCV 2014), pp. 94–108. Springer, Zurich, Switzerland (2014)
88. Zhao, X., Li, X., Zhang, Z.: Joint structural learning to rank with deep linear feature learning. IEEE Transactions on Knowledge and Data Engineering **27**(10), 2756–2769 (2015)
89. Zheng, S., Jayasumana, S., et al.: Conditional random fields as recurrent neural networks. In: Proc. of the International Conference on Computer Vision (ICCV 2015) (2015)
90. Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. Advances in Neural Information Processing Systems (NIPS 2011) (2011)