InVID

Deliverable D2.3: Social media filtering and extraction, pre-processing and annotation, final version

Lyndon Nixon, Daniel Fischl, Fabian Fischer / MODUL Technology
Evlampios Apostolidis, Foteini Markatopoulou, Vasileios Mezaris / CERTH
Denis Teyssou / AFP

29/06/2018

Work Package 2: Media Selection and Analysis

InVID - In Video Veritas: Verification of Social Media Video Content for the News Industry

Innovation Action

Horizon 2020, Research and Innovation Programme

Grant Agreement Number 687786

Dissemination level	CO (A public version of the deliverable will be made available)
Contractual date of delivery	30/06/2018
Actual date of delivery	29/06/2018
Deliverable number	D2.3
Deliverable name	Social media filtering and extraction, pre-processing and annotation, final version
File	`InVID_D2.3_v1.0.tex`
Nature	Report
Status & version	Final & V1.0
Number of pages	44
WP contributing to the deliverable	2
Task responsible	MODUL
Other contributors	CERTH, AFP
Author(s)	Lyndon Nixon, Daniel Fischl, Fabian Fischer / MODUL Technology Evlampios Apostolidis, Foteini Markatopoulou, Vasileios Mezaris / CERTH Denis Teyssou / AFP
Quality Assessors	Rolf Fricke / Condat; Jochen Spangenberg / DW
EC Project Officer	Alberto Rabbachin
Keywords	Topic Detection, Story Detection, Breaking News Detection, Burst Detection, Social Media, Social Networks, Social Web, Twitter Video, Social Media Collection, Social Media Extraction, Social Media Filtering, Social Media Retrieval, Video Fragmentation, Concept Detection, Video Annotation, Thumbnail Extraction, Social Media Metrics, Social Media Reach, Social Media Authoritativeness

Abstract

This deliverable provides a final summary of Social Media Filtering and Extraction in the InVID project. The update reflects on the progress made in the InVID project, references all methods and components that have been implemented and integrated into the InVID platform workflow, and provides a final evaluation of those methods and components both against their prior versions and against current third party competitors. The methods and components cover:

- Story Detection - an algorithm to extract distinct newsworthy stories out of a Twitter stream, label those stories in terms of the most significant keywords that define that story, and rank those stories by volume of social media content being generated that refers to that story.
- Social Media Extraction - a set of components to generate story-based queries at regular intervals on social platform APIs in order to retrieve timely and relevant video content for those stories.
- Social Media Annotation - a set of components for the fragmentation of user-generated video content, the extraction of thumbnails for videos and their fragments, and the labeling of each video fragment with a set of most representative visual concepts in that fragment.

Content

List of Figures

List of Tables

1 Introduction

The topic of this deliverable is to outline the successful completion of the social media filtering and extraction pipeline in the InVID project according to internal benchmarking of the implemented InVID components as well as external comparison of the provided functionalities with similar, competing software or services. We update on the implementation of each component since the last report was made one year earlier in deliverable D2.2; additionally we provide a section comparing our work to the state of the art in the area including other tools which provide similar functionality, and we conclude each section with a final evaluation of the components, both by comparing their performance to the evaluations conducted one year ago (see deliverable D2.2) and by comparing the provided functionality with the other state of the art tools.

The structure of the deliverable is as follows:

- Story Detection - the first section will describe our chosen technological method for extracting stories from the social media stream, their disambiguation and relevance.
- Social Media Extraction - the following section will describe our applied approach to information retrieval from large scale social media sources in order to acquire references to media items relevant to the current stories.
- Social Media Annotation - the next section will present the final extensions to the metadata model used to describe extracted social media and the development and evaluation of a service for temporal fragmentation of user-generated videos and the conceptual annotation of those fragments.
- Future Outlook - based on the results reported in this document, we conclude with a look into the future and the role we foresee for these technologies in both supporting InVID as a video verification solution for journalists as well as other potential uptake opportunities.

1.1 History of the document

Table 1: History of the document.

Date	Version	Name	Comment
2018/04/19	V0.1	L. Nixon	first structure created after agreement with partners
2018/05/03	V0.2	L. Nixon	first content creation and updates, incl. state of the art and evaluation notes
2018/05/18	V0.4	L. Nixon, D. Fischl, F. Fischer	updates on planned evaluations and completed implementation work
2018/05/18	V0.5	E. Apostolidis, F. Markatopoulou, V. Mezaris	updates on API of video analysis service, and concept-based video labeling
2018/06/08	V0.6	L. Nixon, D. Teyssou	updates on the evaluations of story detection and social media retrieval
2018/06/08	V0.7	E. Apostolidis	updates on web app for reverse keyframe search
2018/06/11	V0.8	L. Nixon, E. Apostolidis	completing deliverable and finalisation for the QA
2018/06/22	V0.9	L. Nixon, E. Apostolidis	after-QA version, ready for final check
2018/06/25	V1.0	L. Nixon, E. Apostolidis	final version, ready for submission to EC

1.2 List of abbreviations

Table 2: Acronyms used.

Acronym	Explanation
API	Application Programming Interface
CCTV	Closed-Circuit TeleVision
DCNN	Deep Convolutional Neural Networks
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IPTC	International Press Telecommunications Council
JSON	JavaScript Object Notation
KB	Knowledge Base
LMGE (algorithm)	Label correlation Mining with relaxed Graph Embedding
MAP	Mean Average Precision
MDL	Multi-Domain Learning
MTL	Multi-Task Learning
MXinfAP	Mean eXtended inferred Average Precision
NEK	Named Entity keywords
NEL	Named Entity Linking
NER	Named Entity Recognition
NLP	Natural Language Processing
REST	Representational State Transfer
RGB (color model)	Red, Green, Blue
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SKB	Semantic Knowledge Base
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
UGC	User Generated Content
UGV	User Generated Video
UI	User Interface
URL	Uniform Resource Locator

2 Story Detection

In our initial proposal we presented a conceptual workflow for ”Topic Detection”. Our goal was to automatically identify newsworthy events which could guide journalists to online media being posted in association with that event (and which may require verification before it can be used in the professional news cycle). We had three primary requirements to address in the InVID context, which took our work away from the classical research activities in topic detection:

- Timeliness of detection of a new newsworthy event;
- Addressing multilinguality and alternative names in the detection approach;
- Quantifying the newsworthiness of the event as suitable for extracting eyewitness media.

Given that the results of our approach are referenced as Stories and that the InVID Dashboard already has a model for classification of content called Topics, we have chosen to use the terminology ”Story Detection” to refer to the InVID activity of news event extraction from social media, with Topics being used in the dashboard as an additional tool to classify those extracted news events.

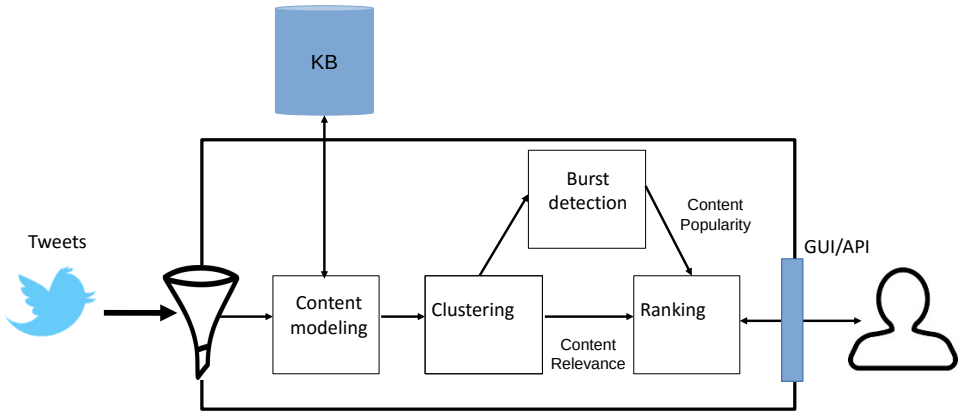

Figure 1: InVID workflow for story detection.

We have previously presented a workflow model for story detection (Fig. 1). In this section, we will focus on the optimisations made to our story detection algorithm since the last reporting of the work (deliverable D2.2):

- Content modeling - since modeling is based on a bag of keywords we have sought to improve the keyword detection and disambiguation;
- Clustering - having chosen a community detection algorithm as a means to cluster documents as bags of keywords, we have since focused on optimising the detection of the correct boundaries between clusters to uniquely identify individual news stories;
- Labeling - we seek to identify each news story to the human user by automatically generating a label for each cluster which can unambiguously provide a meaningful summary of the main items that play the significant role in that story.

Thus we complete this section by presenting a final evaluation of the story detection work based first on a comparison to the state of the art (which is presented in the next subsection) and then a benchmarking against the presented algorithm from one year ago (see deliverable D2.2). Hence we will show how the InVID work on story detection differs from other tools available to journalists as well as how it has improved against its own baseline through the various optimisations described in this section.

2.1 Relation to State of the Art

The previous deliverable (D2.2) reported on the state of the art in scientific research on the subject of "topic detection". Some work applies the task of topic detection to datasets about news, thus being a similar form of news story detection as pursued by InVID. The final subcategory of work in this area is where the identification of a news story should be achieved as quickly as possible following the actual event which causes the news story, thus referred to as breaking news story detection. Besides our own publications on tweet clustering for breaking news detection which help establish the work as part of the state of the art (Vakulenko, Nixon, & Lupu, 2017) (Nixon et al., 2017) other recent publications have shown that the topic of event detection (particularly from Twitter) remains very relevant and a subject of ongoing research. A problem with this research area is the lack of consistency in how work is evaluated, with different datasets being used in publications for evaluation (or often created in an ad hoc manner for each publication). This makes it very difficult to draw direct comparisons, especially as the purpose of event detection may differ (e.g. in InVID we consider also how the detected stories may be used to also precisely collect related media content from social networks. Information retrieval is usually not a considered purpose of other event detection publications.)

(Yılmaz & Hero, 2018) used Twitter hashtags with text and geolocation features, and thus needed to both determine if a hashtag was associated with an event and which group of hashtags were associated with the same event. They demonstrate computational efficiency in the event detection and tweet clustering but do not address the newsworthiness of the events they detect nor evaluate the accuracy of the identification of (news) events.

(Hammad & El-Beltagy, 2017) focused on burst detection (which we have already addressed in D2.2) for Arabic language Twitter. They used tf/idf and entropy measures over sequential time frames. Evaluation was restricted by the lack of gold standards for the Arabic language. However they looked only for detection of up to three significant events in 12-23 hour periods, which is much less detailed than InVID (with topics, up to 100 distinct stories in a 24 hour period).

(Srijith, Hepple, Bontcheva, & Preotiuc-Pietro, 2017) extends previous work of the PHEME project on story detection from Twitter to now 'sub-story' detection. For this, they make use of annotated datasets restricted in each case to one 'main' story e.g. Ferguson riots. Sub-stories are definitely an area of future interest in story detection for InVID, however it can be noted that the Story Flow visualisation already provides users with a means to identify and track larger stories over time (and while the story is connected across time frames, one can still see changes in the associated keywords as the story develops) whereas the reported 'automated' results reflect the challenge of sub-story disambiguation (highly variable evaluation scores from 0.11 to 0.7).

(Alsaedi, Burnap, & Rana, 2017) focus on 'disruptive event' detection in Twitter through text-based Naive Bayes classification and cosine similarity-based clustering. A temporal tf/idf approach determines the top terms for each cluster. Precision@K results of event detection return values of 0.7 to 0.85. They then restrict this further to the 'disruptive events'. There is no event labelling, particularly for information retrieval, as in InVID - rather event summarization is done using centroid tweets in the clusters, closer to our initial work in D2.1.

(Qin, Zhang, Zhang, & Zheng, 2018) present a frame based approach to event detection as opposed to clusters. Frames can capture triples with relations between two arguments, thus modelling an event in a more structured manner than a cluster of keywords or tweets. A more structured modelling of stories is something we hope to add in InVID in the future, as the shift to the use of a Semantic Knowledge Base for typing and disambiguating the keywords will allow such determination of agency and relations between participants. Reported precision is 0.65 to 0.71; this is lower than what we found for InVID and other state of the art systems but is attempting to detect events with a more structured representation of those events which understandably adds new complexity to the task.

(Mele & Crestani, 2017) perform event detection by using topic mining, named entity recognition and burst analysis. They evaluated this approach using news articles rather than tweets and reported an average precision of the event clusters of 0.93. Interestingly they reported similar errors as we have experienced in D2.2, e.g. cluster merges where there are shared keywords among two distinct stories. We have been able to show significant removal of these false merges in our updated algorithm, as per our 'distinctiveness' measure.

(Tonon, Cudré-Mauroux, Blarer, Lenders, & Motik, 2017) propose 'semantic' tweet analysis for event detection. It models the tweet text in a structured manner using NLP then links to entities in Knowledge Graphs such as DBPedia. As such, it improves keyword search for events in Twitter. The work is thus distinct from ours as it is looking at improved 'pull' of events (by search) whereas we seek to automatically 'push' events (by detection). However the extension of event detection with additional semantics for the

terms being discussed is also for us very relevant; the keywords update with the Semantic Knowledge Base will provide us too with this additional semantic layer for event understanding.

(Katragadda, Benton, & Raghavan, 2017) may follow the most similar approach to ours for event detection. Their paper focuses however on whether additional sources further improve results rather than whether Twitter alone works well. Reported f-score for event detection with Twitter alone is 0.85, and reaches 0.9 with the addition of one more source. As stated, this work entitled 'Event Detection at Onset (EDO)' appears closest to what has been done in InVID, where we are reporting evaluation figures over 0.9 (see the Section 2.5) and providing shortened, accurate labels (EDO identifies events by their full set of keywords) which also serve for precise information retrieval.

So we can consider our own implementation to represent the scientific state of the art in this area. To compare, we will consider other tools or services that claim to offer the same functionality, i.e. automatic detection of news stories in a timely manner from a given data set. Based on the services mentioned in the previous deliverable and comparable tools (referred to as "Discovery Platforms") identified as part of the InVID exploitation and dissemination plan, we can refer to the following:

- Banjo - claims to offer near real time detection of news events based on big data analysis over social network postings
- Dataminr for News - offers event detection by clustering tweets and geo-locating the story
- Echosec - does not automatically detect news events
- Event Registry - groups news articles into recent events
- Facebook Signal - uses Facebook data to highlight recently occurring events to journalists
- Nunki - have a beta platform Signal for news story detection from the Twitter stream
- Spike by Newswhip - does not automatically detect news events
- Truthnest - is focused on verification tools for tweets based on text and user analyses

While all of these platforms provide something similar to what we provide through InVID, it is already clear not all have story detection capability. Some do allow searches for social media about a news story, so some may be tested in the social media extraction chapter. We will note in the evaluation section which tools we could compare directly with the InVID dashboard. The SNOW 2014 Data Challenge had confirmed newsworthy topic detection to be still a challenging task: the top F-score of the competing solutions had been only 0.4 (Precision: 0.56, Recall: 0.36). Unfortunately to date no comparable cross-evaluation of news story detection tools has been performed and due to the particular idiosyncrasy of the SNOW data set we can not directly compare our implementation to the SNOW 2014 participating tools and their results. However, in last year's evaluation we could report a correctness score of 0.895 and a distinctiveness score of 0.598. Besides comparing these past results to the results from our implementation this year, we will also evaluate the implementation against the other story detection tools introduced in this section as far as this is feasible.

2.2 Keywords (Semantic Knowledge Base)

One aspect of the story detection which was identified as critical to the accuracy of the results is the quality of the keywords.

To form a basis for improved keyword disambiguation, we have set up a Semantic Knowledge Base (SKB) which models distinct lexical senses as resources in a graph. This graph is extended by data for machine translation of natural language and for supporting keyword annotation. These three parts are connected in a Triple Store and then made available for distinct use cases to the InVID Platform through a synchronization with its ElasticSearch indices and via a REST API. A Web-based user interface (SKB-Viewer) will support eased means to explore the data in the Triple Store (see Fig. 3).

The central piece of the SKB is a lexical model based on the Lemon model (see Fig. 2). The Lemon model has at its core these types:

- Lexical sense: A certain meaning
- Lexical entry: A word that evokes the lexical sense. A lexical entry usually has a part of speech (i.e. one entry can be a verb evoking the meaning, another can be a noun evoking the same meaning) and a language.

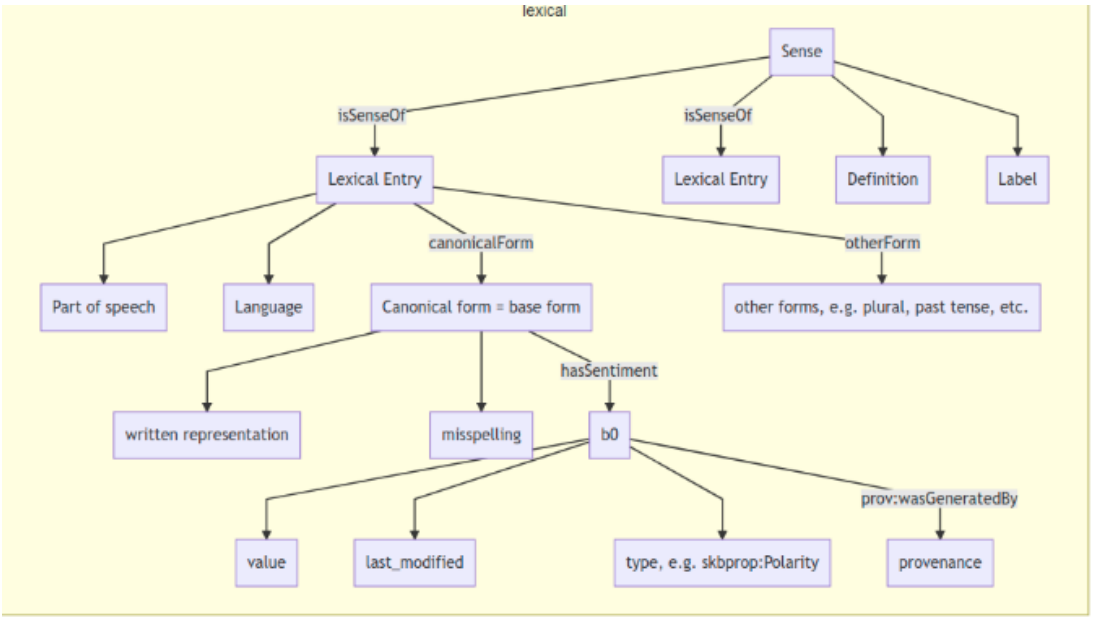

Figure 2: The Lemon lexical model.

– Lexical form: An entry can have several forms. A distinction is made between canonical forms (the lemma of a word) and other forms, e.g. tenses, plural, male/female forms, etc.

– Written representation: Each form has one or more written representations. This can e.g. account for regional differences in spelling (UK/US) or maybe frequent misspellings.

An additional connection can be that a sense references e.g. a DBpedia resource. Eventually, we hope to integrate the lexical senses in the SKB with the entities returned by the RECOGYNZE NER service. To seed the initial SKB, we took a dump of the English language OmegaWiki from the lemonUby project and translations into German, French, Spanish, Portuguese, Czech, Russian and Chinese were included, where available. Since OmegaWiki only contains base forms, we collected (for English and German) frequencies of terms and the part of speech tags they got assigned from documents ingested into the webLyzard platform. Together with a lemmatizer we tried to attach non-lemmas to the correct base form. Additionally we attached adverbs to the senses where the corresponding adjective got attached and similarly adjectives to where the verb got attached (if the lemmatizer reduced them to an adjective or verb, e.g. killed is the adjective to kill). This resulted in:

– 38758 distinct German terms

– 55058 distinct English terms

These are attached to:

– 66827 English forms

– 76998 German forms

Additionally for the other languages, each form has, by now, only one term written representation/term attached:

– 69929 Spanish forms

– 56605 French forms

– 28399 Portuguese forms

– 15119 Russian forms

– 12273 Czech forms

Figure 3: SKBViewer screenshot of a Lemon lexical sense.

– 4579 Chinese/Mandarin forms

Which eventually are attached to:

– 31330 senses

– 57921 lexical entries, of which 34236 have PoS specified

– 342102 lexical forms

Spelling differences will be attached as additional written representations to word forms. As an initial start, 300 UK spellings were added to the US variant and 235 US spellings added to the UK variant.

We added frequent misspellings obtained from a list of Wikipedia for English and German. 7618 misspellings (for EN and DE only currently) were added.

To use the SKB in improving keyword results, we experimented with the use of two services driven by the SKB in the first step of story detection: the creation of the keyword graph.

– A Synonym service (REST endpoint) provides for a given term and language a list of sense definitions and base forms of the lexical entries, such that two or more keywords that carry the same lexical sense can be merged in the graph. e.g. 'murder' as input would output, among others, 'assassinate, kill, homicide, polish off, slay, bump off'.

– A Variants index (ElasticSearch sync) provides to the InVID platform for one lexical form a set of variants of that form (plural/singular, tenses of verbs, declinations of nouns/adjectives, common misspellings) and a set of lexical senses that can be represented by that form. The idea is that in the story detection we can query for the variants of any given keyword and get candidate senses. Based on the other keywords in the story cluster, it may be possible to choose the correct sense, providing for a better keyword disambiguation. Cluster merging may also be improved since currently distinct keywords (lexical form) can be disambiguated and mapped to the same lexical sense.

Keywords which are not aligned to named entities via the RECOGNYZE NER service - labelled by us as 'Non-Entity Keywords' or NEKs - will eventually be connected through the SKB to reasonably equivalent, existing LexicalEntry subjects via a seeAlso relation.

The SKBViewer provides a way to inspect the data in the SKB. Currently an alpha version allows to provide a lexical term and it collects all the senses that are connected to that term. Additionally it shows all the other lexical entries, forms and written representations and all their properties for these senses. We plan to extend this to become a SKB Editor where the entries may not only be viewed but (for logged in users) also edited in order to correct errors or complete details in the dataset.

2.3 Story Clustering

To begin with, we perform hourly calculations over the Twitter news stream of the top keywords in the data and for each keyword, the top associated keywords. This results in a List of 'SimpleStories' (= a single descriptive keyword + a list of co-occurring keywords). From this, the keyword graph is constructed as the basis for the clustering, where each cluster should ideally group together keywords for a distinct and unique news story. Prior to the graph construction, we have added an initial redundancy check for the keywords. This will eventually be replaced by the use of the Semantic Knowledge Base which should ensure no redundancy in the keywords in the story clusters. For each keyword we now check for component matches with already (previously gathered) existing keywords in the input set for the keyword graph, e.g. 'trump' is a component term of both 'Donald Trump' and 'President Trump'. Thus we store 'trump' (always the shortest common component) as the preferred label for also the n-grams 'Donald Trump' and 'President Trump'. So the assumption is, when building the nodes and links of the graph, to name the nodes 'donald trump' and 'president trump' simply 'trump', so that a link can be established to both previous mentions of trump and also stories mentioning trump, donald trump and president trump will be connected.

We also created a debugging environment to test varying the threshold set for the merging of clusters in the graph. This merge threshold was set at 0.5 in the initial implementation, meaning that there needs to be a 50% overlap in the unigrams which form the set of keywords in the cluster (i.e. the set of keywords is seen as a bag of words) for both clusters to be merged. Testing showed that some duplicate stories could be merged by reducing the merge threshold but of course at the same time this increased

the chances of merging two stories which were actually distinct (e.g. keywords like 'trump' and 'usa' could co-occur for multiple, different stories). We found that the larger the document set from which the keyword graph was constructed, the more effective a lower merge threshold was in correctly merging two clusters about the same news story while not merging two clusters about different news stories. As a result, while we retain the default of 0.5 in the InVID dashboard, in our evaluations we applied the lower merge threshold of 0.25 for the TOP stories (the stories being generated from ALL documents, which were in the thousands) whereas the original threshold of 0.5 was retained for the stories filtered to individual TOPICS (which were in the hundreds).

2.4 Story Labelling

Once the stories are detected, they are created in the sense that the set of keywords within the cluster are ranked by weight and used in (i) creating a label for the story from the three highest ranked keywords and (ii) collecting documents for the story based on an ElasticSearch relevance-weighted search using the keyword set over all documents. Keyword weighting was corrected so that the labels for stories should be more relevant. Each keyword cluster depicting a story was then used to search for an initial set of documents in this story by finding documents containing some of the top keywords and to improve label relevance, an aggregation was performed on this initial document set to retrieve the top keywords again on this set, which were then used as an updated description for the story. An effect of the corrected labelling procedure was the appearance of stories with the same label as separate stories in the dashboard. So we implemented a simple post-clustering step where two stories are merged when the labels of the two stories overlapped. The corrected labels also meant that we also adapted the story documents query again to reflect these new description keywords. This lead to a second set of documents, more relevant to the new descriptive keywords.

2.5 Story Detection Evaluation

We aim to evaluate our current story detection algorithm against its implementation from one year earlier, to indicate the possible improvements in quality. We also perform an additional evaluation for this year (and final version of the algorithm) against state of the art story detection tools. This latter evaluation is restricted by what access we could gain to those tools, some of whom are commercial products and did not agree to provide us trial log-ins to compare their results with ours. This was the case with Banjo, Dataminr and Facebook Signal. On the other hand Echosec, Nunki and Spike require an initial keyword-based search whereas Truthnest requires a specific tweet URL and provides text and user based verification services. So only EventRegistry could be used to compare detected news events with InVID; Nunki gave us access to their beta event detection service Signal however it only provides reverse chronological tweets matching a 'newsworthy' template similar to our 'Twitter News' data feed and no clustering into distinct news stories.

To benchmark the current and final implementation of story detection, we chose to look at its output on the same data evaluated one year ago in the previous project deliverable. This follows the same methodology, reported in InVID deliverable D2.2, Section 2.6. We look again at the stories detected for the period June 19-23, 2017 (top stories from the Twitter Accounts feed). The results are shown in Table 3. Previously we had noted that a major issue of concern was the distinctiveness, which penalises both merged and split stories in the results. It can be seen that we have improved the values in both correctness and, significantly, distinctiveness. We had now three non-stories compared to five from the previous run, and had noted already that these all were the result of the tweets of a single news organisation, which we subsequently removed from our data feed. The results still include this news organisation so the improved correctness value indicates that we more successfully rank more newsworthy stories more highly and can in cases remove unnewsworthy clusters from the results. However, the greatest improvement can be seen in the value of distinctiveness jumping from 0.597 to 0.96 **over the exact same data**. This is a clear demonstration that our efforts to improve the correct splitting and merging of clusters into a set of distinct news stories has been significantly effective. Finally, the values for homogeneity and completeness had already been good in the previous year, yet we could still show an increase in both - 97% correctness and an almost perfect 99% in homogeneity, i.e. that the documents provided for each story are almost always relevant to that story.

We also want to look at more current news detection. Here, since we do not wish to take any single news source as a 'ground truth' for the task, we can compare current news detected by our algorithm with the news stories provided via the interfaces of other story detection platforms. Here we can consider precision/recall in the sense of whether we detect newsworthy stories that they do

Table 3: Our story detection benchmarked against last years results.

LAST year	Cluster measures		Document relevance measures	
	Correctness	Distinctiveness	Homogeneity	Completeness
June 19, 2017	1	0.5	0.87	0.84
June 20, 2017	0.9	0.44	0.94	1
June 21, 2017	0.7	0.71	1	1
June 22, 2017	1	0.625	0.92	0.95
June 23, 2017	0.875	0.71	0.91	0.87
Avg over 5 day	**0.895**	**0.597**	**0.93**	**0.93**
THIS year				
June 19, 2017	1	1	1	0.97
June 20, 2017	0.9	0.89	1	1
June 21, 2017	0.9	1	1	1
June 22, 2017	0.9	1	1	1
June 23, 2017	1	0.9	0.94	0.86
Avg over 5 day	**0.94**	**0.96**	**0.99**	**0.97**

not or they detect newsworthy stories that we do not. As explained above, there is only one story detection platform available for us to compare to: EventRegistry. Over a period of three days (28-30 May 2018) we took the top 10 stories from the InVID dashboard and the top 10 'recent events' from EventRegistry. We considered for each story list whether all stories are newsworthy (correctness) and distinct (distinctiveness); we also looked at the top 10 documents from every story (just as InVID sorts documents by 'relevance', EventRegistry offers a sorting of documents by 'story centrality'). We also looked at overlap between the top 10 stories each day, with the caveat that both systems of course detect many more stories so the absence of a story detected by one system in the list of the other does not mean the story was not detected. Thus the overlap can only be a measure of the similarity of the story ranking of both systems, rather than an evaluation of story detection per se.

In Table 4 we show the evaluation of the InVID stories for the three days. The almost perfect values demonstrate that the improvements demonstrated by the benchmark of last years evaluation are also consistent with the current output of the InVID dashboard. Comparing it to EventRegistry, it can be said that they also perform almost perfectly on providing newsworthy events and separating them distinctly; sharing a 100% correctness score they scored 97% in distinctiveness due to one story duplication on the third day (where the story label was once in English and once in Russian, suggesting they may be issues in the cross-lingual translation). In terms of story coverage, the top stories between the platforms did vary with only between 2 and 5 stories being shared in the top 10 on both on the same day. EventRegistry ranked sports stories with US sports (basketball, baseball) higher, appearing three times whereas InVID had cricket twice; InVID had a single football (soccer) story in the top 10 while EventRegistry had five. EventRegistry also included a story on the Star Wars film Solo twice. InVID might also detect such stories but they tend to not reach the top 10 when there is other significant news and should be findable using the Topics (Sports, Arts and Entertainment). It was our feeling that InVID highlighted more stories of news significance in its top 10, for example on the first day this included the Storm Alberto in Florida and former President George Bush Sr.'s hospitalization, both of which were not shown by EventRegistry. Likewise, on May 29 InVID detected the memorial day celebrations at the Arlington Cemetery and on May 30 the Supreme Court rejecting a challenge to abortion law in Arkansas. We have already acknowledged that every news platform may have its own focus in the news it provides and thus it is not possible to say one list of news stories is 'better' than another. Hence we can only say that InVID seems to perform just as well as any other story detection tool - while we were unable to test other competitors, scores of 93-100% already indicate little more that can be perfected.

Table 4: Story detection comparison for May 2018.

InVID Story Detection	Correctness	Distinctiveness	Homogeneity	Completeness
May 28, 2018	1	1	0.94	0.9
May 29, 2018	1	1	0.95	0.95
May 30, 2018	1	0.8	0.98	0.98
Three day average	**1**	**0.93**	**0.96**	**0.94**

3 Social Media Extraction

We have set up a social media extraction pipeline which is configurable and extendable to support additional sources. The initial pipeline supported YouTube, DailyMotion and Vimeo APIs. Each component, as implemented for the platform, is called a "social media mirror" (previously, other mirrors already existed in the platform but were focused on retrieval of textual documents). In this section, we consider how our work on social media extraction compares to the state of the art, i.e. scientific publications on news video information retrieval as well as current services and tools available to journalists to uncover current online media (especially on social networks) relevant to (breaking) news stories. We outline the further work done on social media extraction since the last deliverable and provide an evaluation of this work based on current video collection in the InVID dashboard, compared to content being discovered for the same news stories in other discovery tools.

3.1 Relation to State of the Art

Looking for recent scientific publications on news video retrieval from social networks, our own paper (Nixon et al., 2017) is returned. Indeed, the scientific literature seems to tend to consider video retrieval as a research subject when it, for example, researches query by visual input (and thus a visual similarity matching problem). The precision of keyword-based queries to social network APIs has not been a recent topic. Our own experiments (with single, double and triple keyword queries) were reported in deliverable D2.1.

Some of the tools considered previously for comparison with our story detection approach also exhibited the functionality of collecting online media about the stories. Some were, as it turned out, lacking any automatic detection of news stories but acted as content discovery platforms for news where the user initiated the search. They could then, potentially, be subjects of a comparative evaluation with our social media extraction approach, whether based on detected stories or text searches. The state of the art tools that we could compare in terms of news story-based online social media content discovery (and, for InVID, with a focus on **VIDEO** retrieval) are: Echosec, Newswhip and Nunki. Neither Banjo, Dataminr nor Facebook Signal gave us access to their platforms. EventRegistry, considered in the story detection, only collects news articles and not social media of any type. TruthNest only validates provided tweets. We compared the videos provided for news stories by the InVID dashboard with those for the same story from the three competitors in the Evaluation subsection below.

3.2 Query construction

The previous deliverable D2.2, Section 3.3 compared video retrieval based on the keyword-association pairs (our initial implementation) and on the story labels (the proposed implementation). As reported in this deliverable (Section 2.2) we have further improved the quality of story labelling. The evaluation of story detection (Section 2.3) confirmed the greater story breadth that can be provided by using the story labels to construct the queries (distinctiveness for top stories of 93%). We evaluate again the resulting relevance of returned video documents in this deliverable. Assured that the switch to using story labels to query for video documents will ensure appropriate story breadth and depth in video collection, the switch is planned to be activated in the dashboard in the final phase of the dashboard updating (2nd half of 2018).

3.3 Information Retrieval sources

We have extended our video sources with Facebook Video and Reddit. Both required different approaches to add them to the social media extraction. In the case of Facebook, it is not possible to query the API directly for (public) videos that would match some search term. So we can only identify relevant

(news-related) videos on the Facebook platform when there is news-related content on other platforms which link to the Facebook video. We chose to initiate collection based on our Twitter Accounts feed (professional news accounts). Our approach is to identify links to external Websites in tweets and where that link is to a Facebook video URL, we queue that URL in a newly developed Facebook Video mirror, which can query the Facebook API for the video document. However few professional news channels on Twitter post with Facebook video, preferring the native video or YouTube as platforms. We tested also with our Twitter News feed (user-generated content around 'breaking news'), however we found the vast majority of FB video posted there was not newsworthy.

As for Reddit, one can monitor via an API a selected 'subreddit' (akin to a single channel around a given topic where any Reddit user can post). Subreddit posts may include video links, often YouTube. Reddit has also launched its own native video hosting but the majority of video embeds observed in our tests are still YouTube. So we have concentrated on implementing a queue for a list of YouTube URLs extracted from subreddits, where we directly access the video metadata via the API. We started with the subreddits /r/politicalvideos and /r/videos (200 documents per channel and day).

As also observed in the distribution of sources of the video documents provided in the InVID dashboard sorted by relevance to the news stories, YouTube and then Twitter are definitely the most relevant sources for news video on social media. DailyMotion, Facebook and Vimeo may contribute some individual additions to the news video collection, with the caveat that Facebook is not publicly searchable and hence much more restricted as a video source, limited to finding Facebook videos already found and used by professional news outlets.

3.4 Relevance filtering

Monitoring the videos returned for our dynamically generated queries, we noted that there was some noise generated from certain query terms when they could be more generally interpreted. While the YouTube API itself does a good job in relevance sorting query results, in those cases certain other irrelevant videos were being returned as relevant, probably because the videos themselves are popular on the YouTube platform and hence 'gain' in relevance for Google's algorithm. Two examples for this are 'audiobook' (which for example appears together with political topics) and 'live concert' (which appears together with a concert location, which may match a news story location being searched for). We implemented a filter after the video API responses to remove videos whose title matched these and similar terms as they were constantly irrelevant to the newsworthy video collection task.

3.5 Social Media Extraction Evaluation

While we did evaluate our information retrieval queries in the past deliverable, we have updated the story labelling approach and hence we repeat an evaluation of the quality of the resulting queries. As in the deliverable D2.2, Section 3.3, we will evaluate the 'proposed story based approach' using the measurements for precision, accuracy, recall and f-score. We will use the story labels from the top 10 stories from Twitter Accounts in the InVID dashboard for the time period May 28-30 2018 as conjunctive query inputs. We will test results relevance by querying the YouTube API, using the default result sort by relevance. Since Web based information retrieval excludes the possibility of knowing the total number of correct documents, recall in its classical form is no longer a meaningful metric and therefore 'precision at n' is commonly used where n provides the cut-off point for the set of documents to evaluate for relevance. In line with the first page of search results, a standard choice in Web Search evaluation, we choose n=20. In Table 5 we compare the results from last year on 13 June 2017 (reported in Deliverable 2.2) and the results this year for the dates 28-30 May 2018 (and their average). It can be seen that our recall value has increased considerably, meaning that when we make a query for a newsworthy story we are more likely to only get videos that are precisely relevant to that story than video of any newsworthy story. So while accuracy has remained more or less the same (the proportion of newsworthy video being collected into the InVID platform is probably still around 80% for YouTube) our precision value - that the collected video is precisely about the news story we detected - shows an over 20% improvement.

Given the updates made in the past year to collect more relevant videos for each news story and also to sort them by relevance in the story view, we also evaluate the relevance of the social media video retrieval for news stories in the InVID Dashboard in comparison to other state of the art tools for journalists. We gained trial access to a number of other commercial platforms which also provide functionality to find online social media video for news stories. The three platforms available to us (Echosec, Newswhip, Nunki) all work with keyword-based search and provide content filters to type and social network, so that results can be filtered to videos. Just as the default in the InVID dashboard, we

Table 5: Our social media retrieval tested on the new story labels.

Metric	**13 June 2017 value**	**28 May 2018 value**	**29 May 2018 value**	**30 May 2018 value**	**2018 avg value**
Precision	**0.54**	0.79	0.7	0.79	**0.76**
Accuracy	**0.82**	0.85	0.74	0.84	**0.81**
Recall	**0.64**	0.93	0.95	0.94	**0.94**
F-score	**0.59**	0.82	0.72	0.81	**0.78**

set the time range to the last 24 hours and considered for each story detected by InVID the relevance of the video results on each platform. Note the differences in the social networks from which videos were retrieved:

- Echosec: Reddit, Vimeo, YouTube
- InVID: DailyMotion, Twitter, Vimeo, YouTube
- Newswhip: Facebook, Twitter, YouTube
- Nunki: Twitter, VKontakte, YouTube

We compared the volume of search results and percentages of relevant video across the same stories on the InVID dashboard, Echosec and Newswhip. Unfortunately Nunki provided log-in credentials to us late due to a problem that they had with their office space, so they could not be evaluated with the other three - for completeness, we looked at Nunki separately on the 6th June. For relevance, we look at precision at n=10, and note that whereas InVID can sort story results by relevancy, Echosec and Nunki only support sort by recency, whereas Newswhip uses various sort options where we chose ”highest velocity” which means video being currently spread (shared) at a higher frequency. We add volume since it may also be significant HOW MANY videos each platform can return for the current top news stories. We take absolute totals for search results based on the time restriction of the last 24 hours. Table 6 shows the direct comparison of relevance and volume for all three platforms over all three days and their average. Looking at InVID compared to Echosec, which can be considered a state of the art tool for journalistic discovery of news video on social media, the results are very similar for relevance. While both tend to provide a significant number of videos for each news story in the past 24 hours, it can be seen that InVID offers more content on average, which is not only due to the additional sources (particularly Twitter) but also due to more matching video from YouTube. Comparing to Newswhip, the relevance figure for them is almost perfect but this must be seen in the context of returning far fewer video results. To take an example from the 30th of May, ABORTION LAW + ARKANSAS + SUPREME COURT was a story with 21 videos in the InVID dashboard and 25 videos in Echosec, but Newswhip returned just 6. With apparently between 5 and 20% of the video coverage of the other two platforms, it must be acknowledged that a platform with 1000 videos, of which 90% are relevant to the current news, compared to a platform with perfect relevance but just 100 videos, still means the former has nine times the amount of video material for a journalist to browse. Nunki was tested separately on 6th June, using the stories detected by InVID on that day. Here, the experience was comparable to InVID with both the same or more videos returned for news story searches and 100% precision at n=10 for those stories. Concluding, it seems that Nunki would be our strongest competitor right now for video discovery around news stories on social media - however it requires the specific keyword-based search (and it seems the keywords suggested by InVID’s story detection work very well with it).

Table 6: Comparison of tools for news video retrieval.

Metric		**28 May 2018 value**	**29 May 2018 value**	**30 May 2018 value**	**2018 avg value**
Relevance	**InVID Video**	0.91	1	0.8	**0.9**
Volume	**InVID Video**	1060	728	548	
Relevance	**Ecosec**	1	0.81	0.85	**0.89**
Relevance	**Ecosec**	609	328	309	
Volume	**Newswhip**	1	0.94	1	**0.98**
Relevance	**Newswhip**	52	44	114	

4 Social Media Annotation

4.1 Relation to State of the Art

Image or 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 (Simonyan & Zisserman, 2014; He, Zhang, Ren, & Sun, 2016) and put a multi-label cost function on the top of it (Wei et al., 2016; M. Wang, Luo, Hong, Tang, & Feng, 2016; Bishay & Patras, 2017). As is the case in other multi-label problems, there exist relations between the different concepts, and several methods attempt to utilise/model them 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) (also investigated in D2.2), attempt to learn representations or classification models that, at some level, are shared between the different concepts (tasks) (Argyriou, Evgeniou, & Pontil, 2007; Obozinski & Taskar, 2006; Mousavi et al., 2014; Evgeniou & Pontil, 2004; Daumé, 2009; Argyriou, Evgeniou, & Pontil, 2008; Zhou, Chen, & Ye, 2011; Sun, Chen, Liu, & Wu, 2015; Markatopoulou, Mezaris, & Patras, 2016b; Kumar & Daume, 2012; Z. Zhang, Luo, Loy, & Tang, 2014; Markatopoulou, Mezaris, & Patras, 2016a; Yang & Hospedales, 2015). 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 (Smith, Naphade, & Natsev, 2003; Weng & Chuang, 2012; J. Deng et al., 2014; Ding, Deng, Murphy, & Neven, 2015; Markatopoulou, Mezaris, Pittaras, & Patras, 2015; Qi et al., 2007; Yang et al., 2012; Qi et al., 2007; H. Wang, Huang, & Ding, 2011, 2009; M.-L. Zhang & Zhang, 2010; Lu, Zhang, Zhang, & Xue, 2012; Baumgartner, 2009; Luo, Zhang, Huang, Gao, & Tian, 2014; Cai, Nie, Cai, & Huang, 2013; Taskar, Guestrin, & Koller, 2003; J. Deng, Satheesh, Berg, & Li, 2011; Sucar et al., 2014; Schwing & Urtasun, 2015; Z. Deng, Vahdat, Hu, & Mori, 2015; Zheng, Jayasumana, & et al., 2015; Markatopoulou et al., 2016a). Label constraints refer to regularizations that are imposed into the learning system in order to exploit label relations (e.g. correlations) (Qi et al., 2007; Yang et al., 2012; Zhao, Li, & Zhang, 2015; Schwing & Urtasun, 2015; Z. Deng et al., 2015; Zheng et al., 2015; Markatopoulou et al., 2016a). Post-processing techniques refer to re-calculating the concept prediction results using either meta-learning classifiers or other re-weighting schemes (Smith et al., 2003; Weng & Chuang, 2012; J. Deng et al., 2014; Ding et al., 2015; Markatopoulou et al., 2015). In what follows, we first review works in those two broad categories and then highlight their relation and differences with the proposed method.

4.1.1 Multi-task Learning

Multi-task Learning (MTL) refers to jointly learning classifiers for many tasks by sharing knowledge across them so as to improve their accuracy, instead of learning individual models for each task. Video/image concept annotation can be treated as a MTL problem, where each task is about recognizing one concept. MTL methods can be divided into two broad categories: i) Shallow MTL methods that focus on shallow linear models and typically require pre-computed features as input, for example local descriptors or DCNN-based pre-computed features and ii) MTL methods that are an integral part of deep network architectures. The first category has been extensively reviewed in D2.2. Here we present methods belonging to the second category.

For a start, DCNNs themselves are MTL models that consist of many layers of feature extractors, with the bottom layers learning more generic features that are shared across all of the tasks and the top-most layers being more concept-specific (Yosinski, Clune, Bengio, & Lipson, 2014). Typical DCNN architectures follow a *hard* feature/parameter sharing, i.e. each task uses exactly the same weight matrix for the corresponding layer; and similarly a *hard* feature/parameter separation, i.e. the last layer (a.k.a. the classification layer) takes as input the output of the second-last layer and translates it into a set of concept annotation scores learning weight matrices independently for each task (Yang & Hospedales, 2017; He et al., 2016; Simonyan & Zisserman, 2014). However, more elaborate MTL methods that introduce *soft* feature/parameter sharing, i.e. adjusting how much information and across which tasks should be shared, have been presented. Such methods mainly focus on reformulating existing shallow linear MTL methods in order to be incorporated in DCNNs. For example, (Yang & Hospedales, 2015) proposes a two-sided neural network that unifies several shallow linear MTL methods that use a predictor matrix factorization approach, e.g. $\boldsymbol{w}_j = \boldsymbol{V}\boldsymbol{s}_j^\top$ (Kumar & Daume, 2012) (for the explanation of these variables we refer the reader to Section 4.2.2). MTL in deep learning architectures has also been proposed for facial landmark detection (Z. Zhang et al., 2014) and human pose estimation (Ouyang,

Chu, & Wang, 2014). In (Z. Zhang et al., 2014) the single task of facial landmark detection is optimized with the assistance of an arbitrary number of related tasks. This is a special case of the conventional MTL that typically aims to maximize the performance of all tasks. In (Ouyang et al., 2014), the task of human detection is learned jointly with the task of body locations estimation, which results in improved human pose estimation. In (Markatopoulou et al., 2016a) the two-sided neural-network of (Yang & Hospedales, 2015) is modified and extended, for transferring a network that has been originally trained on a source image dataset for concept annotation, to a target video dataset and a corresponding new set of target concepts. The latter is the DMTL_LC method presented in D2.2.

Transfer learning is another related problem 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 then a transfer learning technique is followed, where a conventional DCNN, e.g. (He et al., 2016), 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 the 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 (Pittaras, Markatopoulou, Mezaris, & Patras, 2017) (the FT3-ex strategy presented in D2.2).

4.1.2 Structured Output Prediction

Structured output prediction refers to methods that exploit semantic relations that may exist between the concept labels, and has received a lot of attention in the deep learning and the broader machine learning field. In contrast to MTL that exploits the common structure that task parameters or low-level features may have across the different tasks, structured output prediction focuses on the semantic relations that exist at the outputs, e.g. concept correlations. Video/image concept annotation is a multi-label learning problem, where given a set of concept labels, each keyframe/image is often associated with more than one label. In most concept annotation datasets, ground-truth annotation is provided without any accompanying structure information concerning the concept labels; however, in many cases the concept labels are statistically related. For example, in the TRECVID-SIN video annotation dataset (Over & et al., 2013), which is one of the datasets used in this study, there are several groups of mutually exclusive labels, such as *indoor-outdoor* or *nighttime-sun*. The dataset also includes several positive correlations, such as *car-vehicle* and *dog-animal*. The automated learning of such relationships can incorporate useful knowledge into the model, improving the accuracy of the DCNN. In order to do so, many structured output prediction methods impose some label structural constraints either explicitly, i.e. using predefined rules that are known for the training dataset, or implicitly, i.e. the model is forced to discover existing label relations and considers them as label constraints. Existing methods can again be divided in those that take as input any pre-computed features and those that are tightly integrated with deep learning architectures.

With respect to the first category, i.e. methods that take as input pre-computed features, two main sub-categories have appeared in the literature: a) Stacking-based approaches that collect the concept annotation scores produced either by a baseline set of concept detectors (e.g. SVMs) or by a DCNN when used as a standalone classifier, and introduce a second learning step in order to refine them, and b) Inner-learning approaches that follow a single-step learning process, which jointly considers extracted features and semantic relations. Stacking approaches aim to detect relations across concepts in the last layer of the stack. In (Smith et al., 2003) concept annotation scores are obtained from individual concept detectors in the first layer, in order to create a *model vector* for each shot. These vectors form a meta-level training set, which is used to train a second layer of independently trained concept detectors. In (Weng & Chuang, 2012), a graph-based method is proposed that uses the ground-truth annotation to build decision trees that describe the relations across concepts, separately for each concept, and refines the initial scores by approximating these graphs. Using external knowledge of label relations, Deng et al. (J. Deng et al., 2014) proposed a representation, the HEX graph, to express and enforce exclusion, inclusion and overlap relations between labels. This model was further extended for “soft” label relations using the Ising model by Ding et al. (Ding et al., 2015). A different approach that outperforms the above was proposed by (Markatopoulou et al., 2015). There, the authors use model vectors to train multi-label classification algorithms that explicitly exploit label relations, instead of learning a second round of independent concept detectors or graph-models. All the above-mentioned approaches implicitly capture label relations from the meta-level training set of model vectors; as a result, they rely on starting with good concept probability estimates in the model vectors, otherwise the errors are propagated to the next layers.

Inner-learning approaches, on the other hand, use the extracted features and exploit concept relations in a single learning step. For example, the authors of (Qi et al., 2007) and (Yang et al., 2012) propose methods that simultaneously learn the relation between visual features and concepts and also the correlations between concepts. In (Zhao et al., 2015) a joint learning-to-rank approach is proposed, which naturally combines the benefits of training a DCNN with a structural SVM model that is used for concept ranking. In (M. Wang et al., 2009) the temporal consistency of concept labels across neighboring video shots is exploited. While in (Hong et al., 2014) an AdaBoost classifier is trained by carefully selecting positive and negative correlated concepts that will be used per iteration. However, inner-learning approaches suffer of computational complexity. For example, (Qi et al., 2007) has complexity at least quadratic to the number of concepts, making it inapplicable to real video/image concept annotation problems, where the number of concepts is large (e.g. hundreds or thousands). The LMGE algorithm (Label correlation Mining with relaxed Graph Embedding) (Yang et al., 2012) is a faster approach with linear complexity with respect to the number of concepts; however, the complexity of the training process is about n^3, where n refers to the number of training samples. Many more methods can be found in this category for multi-label image annotation, which explore such label relations to improve the classification accuracy at the expense of increased computational complexity compared to the stacking-based ones, e.g.(H. Wang et al., 2011, 2009; M.-L. Zhang & Zhang, 2010; Lu et al., 2012; Baumgartner, 2009; Luo et al., 2014; Cai et al., 2013; Taskar et al., 2003; J. Deng et al., 2011; Sucar et al., 2014).

With respect to the second category, i.e. methods that are an integral part of DCNN architectures, structured output prediction techniques have been proposed for application mainly to the pixel-wise semantic segmentation problem. The most popular approach is to combine a DCNN with a graphical model (Schwing & Urtasun, 2015), (Z. Deng et al., 2015), (Zheng et al., 2015). For example, in (Schwing & Urtasun, 2015) a Markov random field is jointly used on top of a DCNN architecture in order to incorporate the spatial relations and label correlations of the assigned labels on the pixels of an image. Similarly, in (Zheng et al., 2015) the conditional random field model is formulated as a recurrent neural network (RNN) and plugged in as part of a DCNN. Structured output prediction for DCNNs has also been proposed for other visual recognition problems, such as group activity recognition (Z. Deng et al., 2015). All of these methods employ probabilistic inference to correct the marginal probability of each label. In contrast to the above methods that use graphical models, in (Markatopoulou et al., 2016a) an auxiliary cost function that approximates the correlations between the concept labels gets added to the total network cost. This auxiliary cost function takes the form of a constraint over the task-specific parameters of the network and is shown to improve its accuracy.

4.2 Video Fragmentation and Conceptual Annotation

In this subsection we describe the developed DCNN (Deep Convolutional Neural Network) architecture that addresses the problem of video/image concept annotation by exploiting concept relations at two different levels. At the first level, we build on ideas from multi-task learning, and propose an approach to learn concept-specific representations that are sparse, linear combinations of representations of latent concepts. By enforcing the sharing of the latent concept representations, we exploit the implicit relations between the target concepts. At a second level, we build on ideas from structured output learning, and propose the introduction, at training time, of a new cost term that explicitly models the correlations between the concepts. By doing so, we explicitly model the structure in the output space (i.e. the concept labels). Both of the above 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. The developed approach has been reported in (Markatopoulou, Mezaris, & Patras, 2018).

As discussed in D2.2 the dominant approach for solving the concept-based annotation problem is training DCNNs in whose architectures 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 (Pittaras et al., 2017). 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 modeling 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. While some methods have been proposed for exploiting in a more elaborate way one of these two different types of concept relations, and in D2.2 we presented the DMTL_LC method that implicitly exploits visual and semantic-level concept relations using a two-sided network. Here we present a single method that jointly exploits visual- and semantic-level concept relations in a unified

Table 7: Definition of the symbols

Symbols	Definitions
$\boldsymbol{x}$	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
i	Keyframe/image index, i.e. $i = 1...N$
j	Concept/task index, i.e. $j = 1...T$
$\hat{\boldsymbol{y}}$	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
$\sigma(.)$	The sigmoid function
$\boldsymbol{\Phi}$	The concept correlation matrix calculated from the ground-truth annotated training set
$\boldsymbol{m}$	A cost vector utilized for data balancing
β	Regularization parameter
$\boldsymbol{z}$	Normalization factor vector

DCNN architecture. More specifically, in contrast to the DMTL_LC method presented in D2.2, our current 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 penalize 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 will explicitly capture, for example, the fact that *daytime* and *nighttime* are negatively correlated concepts.

4.2.1 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, and $\boldsymbol{y}_i$ is its ground-truth annotation. A video/image concept annotation system learns T supervised learning tasks, one for each target concept c_j. More specifically, it learns a real-valued function $f : \mathscr{X} \rightarrow \mathscr{Y}$, where $\mathscr{Y} = [0,1]^{T\times N}$ could then be binarized (e.g. thresholded) in order to provide a hard classification result, if needed.

We propose a DCNN architecture that exploits both visual-level and semantic-level concept relations for video/image concept annotation by building on ideas from MTL and structured output prediction, respectively. In Fig. 4 (i) we illustrate a typical $(\Pi + 1)$-layer DCNN architecture, e.g. ResNet, that shares all the layers but the last one (steps (a),(b)) (Simonyan & Zisserman, 2014; He et al., 2016); in Fig. 4 (ii) we illustrate how the typical DCNN architecture of Fig. 4 (i) is extended by one FC extension layer, which was shown to outperform the typical DCNN architecture when used in transfer learning problems (Pittaras et al., 2017) (i.e. the FT3-ex strategy of D2.2) (steps (c)-(e)); and finally, in Fig. 4 (iii) we present the proposed DCNN architecture (steps (f)-(k)). In the next subsections we first introduce the new FV-MTL approach for learning implicit visual-level concept relations; this is done using the network layers as shown in Fig. 4 in steps (f) to (i). Second, we introduce the new CCE-LC cost function that learns explicit semantic-level concept relations, which is done in step (k). CCE-LC predicts structured outputs from concept correlations that we can acquire from the ground-truth annotations of the training dataset.

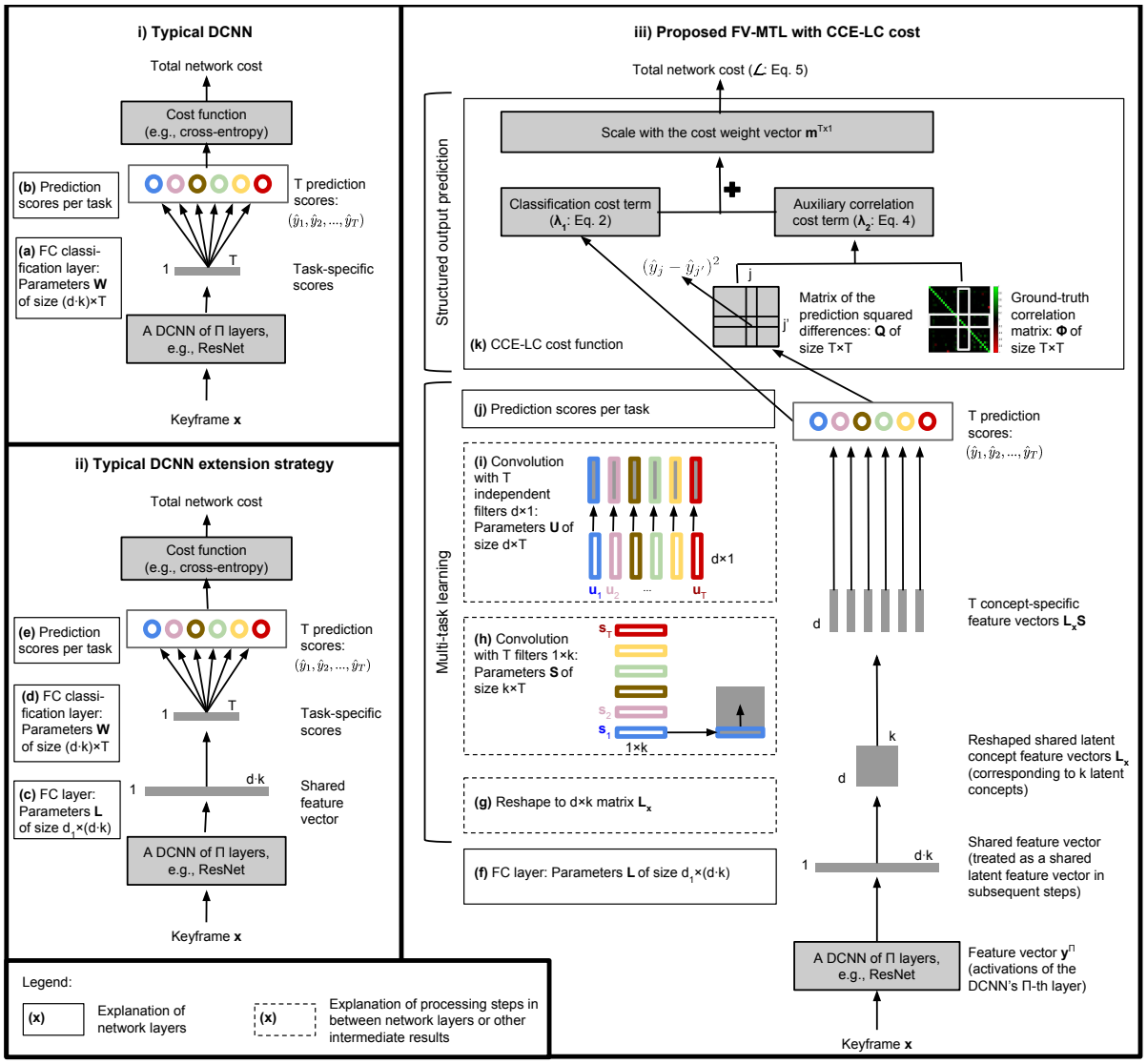

Figure 4: The developed DCNN architecture for video/image concept-based annotation.

4.2.2 Shared Latent Feature Vectors using Multi-task Learning (FV-MTL)

In our approach, similarly to GO-MTL (Kumar & Daume, 2012), we assume that the parameter vectors of the tasks that present visual-level concept relations (i.e. defined in GO-MTL as belonging to the same group) lie in a low-dimensional subspace, thus sharing information; and, at the same time, dissimilar tasks (i.e. belonging to different groups) may also partially overlap by having one or more bases in common. Allowing the sharing also between dissimilar tasks is more natural than creating disjoint groups of task models. In order to do so, 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. Specifically, our approach uses 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 produces 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 $\boldsymbol{s}_j \in \mathbb{R}^{k\times1}$ being a task-specific weight vector that contains the coefficients of the linear combination. Each linear combination is assumed to be sparse in $\boldsymbol{L_x}$, 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 in the sparsity patterns of any two tasks, (i.e. how much overlap is observed between two different task-specific weight vectors $\boldsymbol{s}_j$ and $\boldsymbol{s}_{j'}$, where $j \neq j'$) controls the amount of sharing between them.

The above can be implemented in a DCNN architecture by using the network layers depicted in Fig. 4 in steps (f) to (i). Specifically, an input training-set keyframe is processed by a typical DCNN architecture

(e.g. ResNet) and a fully-connected layer, to produce a shared representation of the keyframe across all of the tasks (Fig. 4: step (f); this is the same as step (c) in the typical DCNN extension architecture). Subsequently, the output of the fully-connected layer is reshaped to the matrix $\boldsymbol{L_x}$ (Fig. 4: step (g)). Consequently, the reshaped layer outputs k feature vectors that correspond to k latent concepts. Those representations are shared between the T concepts. The subsequent layers calculate T concept-specific feature vectors, where T is the number of the concepts we are interested in detecting. 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 - in the case that the 1D convolutional layer implements a linear transform, i.e. we do not use a non-linear activation function, then these two layers implement a feature extraction scheme with a bilinear factorization of the weight matrix (Fig. 4 step (h)). Once T feature vectors are extracted, then an additional layer (Fig. 4: step (i)) transforms each of the T feature vectors into T concept annotation scores, one for each of the concepts that we are set to recognize (Fig. 4: step (j)). The above process leads to a *soft* feature sharing, because the latent concept feature vectors adjust how much information and across which tasks should be shared. By contrast, both the typical DCNN and the DCNN extension architecture of (Pittaras et al., 2017), also presented in D2.2 as FT3-ex, output a single feature vector (Fig. 4: step (a) and (d), respectively) that is shared across all of the target concepts and it is subsequently *hard* translated into concept annotation scores independently for each concept (Fig. 4: step (b) and (e), respectively), as was also discussed in Section 4.1.

Formally, the predicted score for the j-th task (concept) and the i-th datapoint (keyframe/image) is given by:

$$\hat{y}_{i,j} = \text{diag}(\boldsymbol{u}_j^\top (\boldsymbol{L}_{\boldsymbol{x}_i} \boldsymbol{s}_j)), \tag{1}$$

where $\boldsymbol{L}_{\boldsymbol{x}_i}$ is the output of the last fully-connected layer of the right part of Fig. 4 (see step (f)), after reshaping the calculated vector of dimension $1 \times (d \cdot k)$, in order to have a matrix of d rows and k columns (Fig. 4: step (g)). Specifically, $\boldsymbol{L}_{\boldsymbol{x}_i} = \text{reshape}(\alpha(\boldsymbol{L}' \boldsymbol{y}_i^{(\Pi)} + \boldsymbol{b}))$, where $\boldsymbol{L} \in \mathbb{R}^{d_1 \times d \cdot k}$ the parameters of the last fully-connected layer, $\boldsymbol{y}_i^{(\Pi)} \in \mathbb{R}^{d_1 \times 1}$ the output of the previous layer, and α the layer's activation functions, e.g. the ReLU. $G = \{g^{(\pi)}\}_{\pi=1}^{\Pi}$ is the set of network parameters for the first Π layers. $\boldsymbol{s}_j, \boldsymbol{u}_j$ are the j-th columns of the parameter matrices $\boldsymbol{S} \in \mathbb{R}^{k \times T}$ and $\boldsymbol{U} \in \mathbb{R}^{d \times T}$, respectively. Each $\boldsymbol{s}_j$ contains a task-specific weight vector of the coefficients of the linear combination with the shared latent feature vector $\boldsymbol{L}_{\boldsymbol{x}_i}$. This linear combination indicates for each concept which latent tasks describe it. Each $\boldsymbol{u}_j$ contains a concept-specific weight vector that transforms the concept-specific feature vectors $\boldsymbol{L}_{\boldsymbol{x}_i} \boldsymbol{S}$ to concept scores.

Similarly to other DCNN works, we optimize the sigmoid cross entropy between the predicted and the ground truth labels that is formed as:

$$\lambda_{1_{i,j}} = y_{i,j} log \sigma(\hat{y}_{i,j}) + (1 - y_{i,j}) log(1 - \sigma(\hat{y}_{i,j})), \tag{2}$$

where $\sigma(\cdot)$ refers to the sigmoid function $\sigma(x) = 1/(1 + exp(-x))$. That is, we optimize Eq. 2 with respect to the parameters of the network. This is the cost of the *classification cost term* branch in Fig. 4 and differs from the GO-MTL cost function (Kumar & Daume, 2012) in the following ways:

First, while GO-MTL aims to approximate the parameter vector of the j-th observed task $\boldsymbol{w_j}$ by a linear combination of a subset of latent tasks $\boldsymbol{w_j} = \boldsymbol{V s_j}$, where $\boldsymbol{V} \in \mathbb{R}^{d \times T}$ is a shared knowledge basis, our goal is given a keyframe/image i, to learn a new set of concept-specific feature vectors $\boldsymbol{L}_{\boldsymbol{x}_i} \boldsymbol{s}_j$, one per task, that leverage shared properties with all the other tasks. Our assumptions are similar, and we also use a predictor matrix factorization approach $\boldsymbol{L}_{\boldsymbol{x}_i} \boldsymbol{S}$, however, in a different way: In the proposed approach, given an input keyframe/image our method transforms it into T different concept-specific feature vectors that leverage information from a set of latent concept feature vectors using a bilinear factorization of the weight matrix, as described above. Subsequently, parameter matrix $\boldsymbol{U}$ is used in order to transform these concept-specific representations to concept scores, i.e. $\boldsymbol{U}^\top (\boldsymbol{L}_{\boldsymbol{x}_i} \boldsymbol{S})$. Differently, GO-MTL factorizes the 2D weight matrix that encodes concept-specific features and directly transforms the image/keyframe into concept scores.

Second, GO-MTL (Kumar & Daume, 2012) uses iterative optimization and shallow linear models to learn the parameters. For example, in each iteration of the GO-MTL (Kumar & Daume, 2012) method all parameters except for one are kept fixed and the function is optimized towards the non-fixed parameter. In our case a complete DCNN architecture is used, which makes it easy to calculate error differentials per layer w.r.t. its inputs, in order to back-propagate them to previous layers.

Third, the GO-MTL cost function can be optimized with respect either to regression loss (e.g. squared loss) or binary/multi-class classification loss (e.g. logistic loss), thus ignoring the multi-label nature of

the problem. In contrast, our method works for any multi-label classification cost (e.g. the sigmoid cross entropy loss, presented above). It should be noted here that we use the sigmoid function on each activation separately and as a result the different outputs do not compete with each other (i.e. their sum does not equal to 1).

To make clear the difference of the proposed architecture from the typical and extension DCNN architectures (Fig. 4 (i) and (ii), respectively) we set $\alpha(\boldsymbol{L}\boldsymbol{y}_i^{(\Pi)}+\boldsymbol{b})=\Delta$ and rewrite Eq. 1 as: $\hat{y}_{i,j}=$ diag$(\boldsymbol{u}_j^\top(\text{reshape}(\boldsymbol{\Delta})\boldsymbol{s}_j))$.

Similarly, the predicted score for the j-th task and i-th datapoint with respect to the typical and extension DCNN architecture is given by: $\hat{y}_{i,j}^T=\boldsymbol{w}_j^{\top T}(\alpha(\boldsymbol{y}_i^{(\Pi)}+\boldsymbol{b}))$, and $\hat{y}_{i,j}^E=\boldsymbol{w}_j^{\top E}\boldsymbol{\Delta}$, respectively. The task-specific weight vector $\boldsymbol{s}_j$ used in our method contains the coefficients of the latent task feature vectors that will be combined with respect to concept j. This is exactly the way that our method achieves a *soft* feature sharing separately for each concept, i.e. by letting similar concepts to be described by the same latent task feature vectors according to $\boldsymbol{s}_j$. In contrast, the other two architectures of Fig. 4 do not use this linear combination of latent concept feature vectors but let the second-last layer, a single feature vector, to be shared across all of the concepts, thus, a *hard* translation into concept scores is performed independently for each concept.

4.2.3 Label Constraints for Structured Output Prediction

Cross-entropy cost is not adequate for capturing semantic concept relations. In this section we present 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. Details about the used training datasets are given in Section 4.4.1. 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:

$$\phi_{j',j}=\frac{AD-BC}{(A+B)(C+D)(A+C)(B+D)}, \tag{3}$$

where $\phi_{j',j}$ refers to j'-element of the j-th row of the correlation matrix $\boldsymbol{\Phi}$ that contains the correlation between concepts $c_{j'}$ and c_j. $A=p(c_j\wedge c_{j'}|\boldsymbol{y}_i,i=1...N)$, $B=p(c_j\wedge\neg c_{j'}|\boldsymbol{y}_i,i=1...N)$, $C=p(\neg c_j\wedge c_{j'}|\boldsymbol{y}_i,i=1...N)$, $D=p(\neg c_j\wedge\neg c_{j'}|\boldsymbol{y}_i,i=1...N)$, where $p(a|b)$ refers to the probability of a given b. The logical operator $\wedge$ expresses conjunction, e.g. $c_j\wedge c_{j'}$, means that both c_j and c_j appear on the image/keyframe, according to its ground-truth annotations; and $\neg$ expresses negation, e.g. $c_j\wedge\neg c_{j'}$, means that $c_{j'}$ does not appear on the image/keyframe.

The proposed auxiliary concept correlation cost term that uses the correlation matrix $\boldsymbol{\Phi}$ is formed as follows:

$$\lambda_{2_{i,j}}=\frac{1}{T\text{-}1}\sum_{\substack{j'=1,\\ j'\neq j}}^{T}\begin{cases}\phi_{j',j}\left\|\sigma(\hat{y}_{i,j})-\sigma(\hat{y}_{i,j'})\right\|^2, \text{if } \phi_{j',j}\geq 0\\ (-\phi_{j',j})\left\|\sigma(\hat{y}_{i,j})+\sigma(\hat{y}_{i,j'})\right\|^2, \text{otherwise}\end{cases} \tag{4}$$

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 can implement the $\lambda_{2_{i,j}}$ correlation term (Eq. 4) using a set of standard CNN layers, as presented on the top of the right part of Fig. 4. 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). Specifically, the matrix $\mathbf{Q}$ gets multiplied, by element-wise multiplication, with the correlation matrix $\boldsymbol{\Phi}$, i.e. $\mathbf{Q}\circ\boldsymbol{\Phi}$. All the rows in the resulting $T\times T$ matrix are added, which leads to a single row vector.

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

The two cost terms presented in Sections 4.2.2 and 4.2.3, i.e. Eq. 2 and Eq. 4, respectively, can be added in a single cost function that forms our total FV-MTL with CCE-LC network's cost as follows:

$$\mathscr{L} = \sum_{i=1}^{N} \frac{1}{T} \sum_{j=1}^{T} \frac{m_{i,j}}{z_j} \left(\lambda_{1_{i,j}} + \beta \lambda_{2_{i,j}} \right) \tag{5}$$

where parameter β controls the importance of concept correlation term.

In the above cost function we introduce the vector $\boldsymbol{m}_i \in \mathbb{R}^{T \times 1}$ that was originally proposed by (Bishay & Patras, 2017) to address the problem of class imbalance. Class imbalance is a common problem in concept annotation, where for most datasets the distribution between negative to positive examples per concept is highly imbalanced, with the former outnumbering the latter in most cases. This results in bias of the classifier towards the class (positive or negative) that contains the largest number of samples. Consequently, we introduce the cost vector $\boldsymbol{m}_i$ in our cost function in order to balance the number of positive to negative examples per concept. Let us denote by p_j the number of the positive examples and n_j the number of negative examples for the concept c_j. Then, the ratio r_j of the negative to positive examples is computed as:

$$r_j = \begin{cases} \dfrac{n_j}{p_j}, & \text{if } n_j \; and \; p_j \neq 0 \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

We create a weight vector $\boldsymbol{m}_i = [m_{i,1}, ..., m_{i,T}]$, for each training example i.e. for $i = 1...N$. Where $m_{i,j} = 1$ if $y_{i,j} = 0$, $m_{i,j} = 0$ if $y_{i,j}$ is unlabeled and $m_{i,j} = r_j$ if $y_{i,j} = 1$, where r_j is given by Eq. 6, and is different for each concept j. This weight vector is multiplied element-wise with the cost function. By doing so we adjust the misclassification cost of positive examples so as to prevent the biasing of the network towards the negative class when only a few positive examples are available. Furthermore, the normalization factor $\boldsymbol{z} \in \mathbb{R}^{T \times 1}$ that is introduced in Eq. 5 is calculated as: $\boldsymbol{z} = \sum_{i=1}^{N} \boldsymbol{m}_i$, where each position of this vector, i.e. z_j, denotes the sum of the weights for concept c_j.

In our overall network architecture, an additional layer is used in order to implement the complete FV-MTL with CCE-LC cost function, adding the two cost terms (λ_1, λ_2) and scaling their sum by the $\boldsymbol{m}$ (Eq., 5). 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 (Fig. 4: step (k)). In contrast, a typical DCNN architecture simply incorporates knowledge learned from each individual ground-truth-annotated sample.

4.2.5 Web application for reverse keyframe search

The first version of the developed web application for video fragmentation and reverse keyframe search was presented in Section 4.1.6 of the deliverable D2.2. This technology enables a user to segment a single-shot video (which is the most common case for UGVs) into visually and temporally coherent parts, called sub-shots, using the video sub-shot segmentation algorithm described in Section 4.1.3.2 of the same deliverable. Following, the collection of extracted keyframes can be used for fragment-level reverse video search on the Web, by utilizing the image search functionality of the Google search engine. The performance of this web-based tool is offered to the users of the InVID Verification Plugin through the integrated component for video keyframe extraction and reverse search. As stated in D2.2, the web application has a complementary role with the near duplicate detection utility of the InVID Verification Application, which has been presented in Section 4 of the deliverable D3.2. The former allows the fragment-level reverse search of videos on the Web using the extracted keyframes, while the latter enables the video-level reverse search of videos within a constantly extendable collection of selected newsworthy video material.

In D2.2 we listed a number of tools for finding near duplicates of a given image or video. In the following we provide an update of this collection, by removing a few tools that are no longer active and adding several new solutions that were released over the last year. The latter indicates the popularity and attractiveness of image/video-based search and highlights the usefulness of the visual-content-based searching procedure for performing several media asset management tasks, including the assessment of the originality and authenticity of a given video. From the technologies discussed in D2.2, the Spotter technology [1] that, according to its developers, offered functionalities for machine-learning-driven video

[1] `https://spotter.tech/`

reverse search seems to be non-active anymore. The Youtube DataViewer of Amnesty International [2] is still up and running, enabling users to find near duplicates of a given YouTube video. The same goes for the Custom Reverse Image Search of the IntelTechniques [3] which allows image search to additional platforms, including Vimeo, Facebook, Vine, Instagram, LiveLeak and Backpage, and exploits the image search functionality of several search engines, containing Google, Tineye, Yandex, Bing, and Baidu. However, as stated in D2.2, both of these solutions perform reverse video search based on a limited set of randomly selected keyframes/thumbnails that has been associated to the video, thus excluding parts of the video that could enhance the reverse search or be of particular interest to the user. Moreover, the search is supported only for videos available online, thus making impossible the reverse video search for a video stored in the user's machine.

Another (pre-existing) solution that can partially support the retrieval of near duplicates of a video is the TinEye search engine [4], which enables the online search and retrieval of a given image. The advantage of this tool is that it offers a (paid) API to anyone who wishes to perform image search requests in a more automated way instead of providing every time the URL of the image file or uploading a local copy of the file on the TinEye web application. The limitation of this technology when trying to find near duplicates of a given video is that it requires the extraction of video frames that should be used as query images, a process which implies an overhead to the overall procedure. A variation of this platform, with significantly more restricted functionalities though, is the Karma Decay [5] web application which allows to perform reverse image search on Reddit.com. Last but not least, three recently developed platforms that assist the detection and retrieval of images and videos are the Berify, the RevIMG and the Videntifier. Berify [6] is a paid service that, according to its developers, offers functionalities for image-driven search of online available images and videos; updates of the searching results are checked and forwarded to its users on a predefined basis. RevIMG [7] is another non-free solution that offers more unique functionalities, enabling the user to specify and use a portion of an image to search. However, the reverse search is performed only within closed collections of images. Videntifier [8] is a visual search engine which can be used for the retrieval of a given image or video stream (even after being modified), but similar to RevIMG, the identification of a near duplicate relies on the matching of the given media item against a closed reference collection of video content.

In contrast to the aforementioned technologies that rely on a pre-selected and limited set of video thumbnails (Youtube DataViewer, Custom Reverse Image Search), the manual extraction of video frames for performing reverse image search (TinEye, Karma Decay, Berify), or the creation of collections of (pre-analyzed) video content (RevIMG, Videntifier), our web application extracts a dynamic number of keyframes in a way which ensures that all the visually discrete parts of the video are adequately represented through the extracted set of keyframes. Furthermore, it supports the direct analysis of both online available videos from several platforms and local copies of a video from the user's machine without requiring its prior upload to any video sharing platform. In this way, it assists users to quickly discover the temporal structure of a video, to extract detailed information about the video content and to use this data in their reverse video search queries.

Regarding the technical aspects and the performance of this technology, a significant number of improvements have been made on the initial version of the tool presented in D2.2. The applied changes were based on feedback collected from users that are both internal and external to the InVID consortium. Internal users provided their recommendations after evaluating the tool during the test and validation cycles of the project. Following, external users assessed the efficiency of this technology after being integrated and publicly released in as a component of the InVID Verification Plugin, and made their suggestions for improvement via the instance feedback channel that is integrated in the plugin. These improvements include:

- the simplification and beautification of the user interface of the web application (see Fig. 5), by:
  - removing the provided information about the use of the service from the start page of the tool (the existence of such extended information in the start page was considered as discouraging for the user) and placing these details within a new webpage that is accessible by clicking on the newly added "About this tool" button (left part of Fig. 5);

[2] https://citizenevidence.amnestyusa.org/
[3] https://inteltechniques.com/osint/reverse.video.html
[4] https://tineye.com/
[5] http://karmadecay.com/
[6] https://berify.com/
[7] http://www.revimg.com/
[8] http://www.videntifier.com

- changing the aesthetics of the user interface in order to give the same “look and feel” with the other components of the InVID Verification Plugin (right part of Fig. 5);

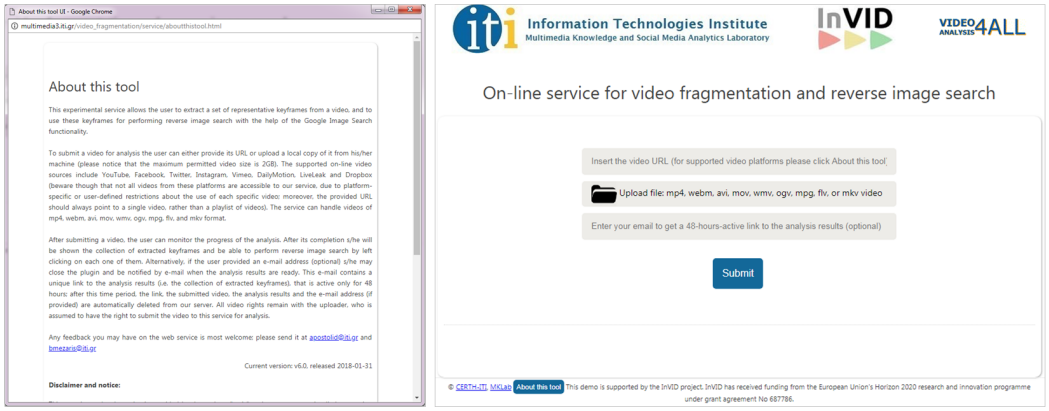

Figure 5: The new user interface of the web app for video fragmentation and reverse image search.

– the simplification of the use and functionality of the web application (see Fig. 6), by:

- removing the integrated video player which has been considered as not too relevant to the main scope of the tool, i.e. the keyframe-driven reverse search of the analysed video on the Web (left part of Fig. 6);
- directly presenting the analysis results to the user without asking him/her to follow a link to a different page showing these results (left part of Fig. 6);
- making the reverse keyframe search a “one-(right)-click” process, instead of asking the user to left click on a keyframe and then select “Search Google for this image” (right part of Fig. 6; the highlighted result leads to the original video about the event);

Figure 6: Direct provision of extracted keyframes and application of reverse image search.

– the optional provision of additional keyframes to the user in order to be utilized for a more extended search of any prior occurrence of the video on the Web, by clicking on the newly added “Show more keyframes” button that appears right after the initial collection of extracted keyframes (left part of Fig. 7); these keyframes correspond to the same video fragments with the initially provided ones, so they could contain duplicates (right part of Fig. 7);

– the extension of tool compatibility to additional video platforms by utilizing the video download mechanism of the REST service for video fragmentation and annotation (the supported platforms include YouTube, Facebook, Twitter, Instagram, LiveLeak, Vimeo, DailyMotion and Dropbox).

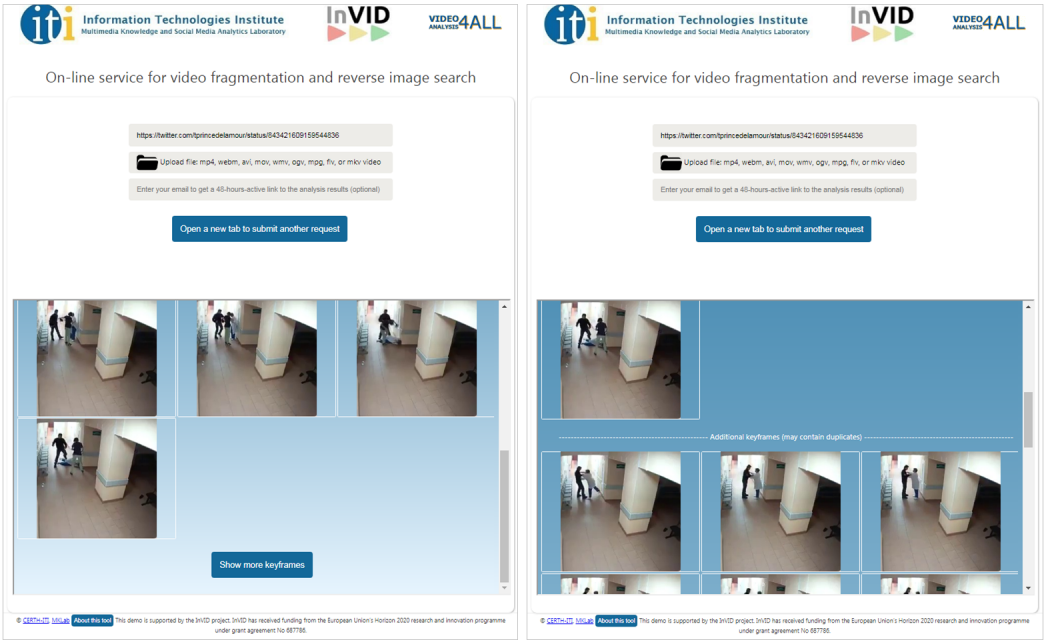

Figure 7: Additional keyframes can be optionally provided to the user for more extended search.

The feedback concerning the performance of this tool, received mainly from the users of the corresponding component of the InVID Verification Plugin, remains very positive and encouraging. According to the analytics about the use of this web application since the public release of the plugin, more than 900 users have submitted (in total) over 3000 requests, for analysing more than 240 hours of video content. The functionality of this component enabled the users to debunk a number of fake news that are based on the re-use of a previously published video. Indicative examples of such fakes and the corresponding original video sources that were identified with the help of the web application, can be found in Table 8.

Table 8: Fakes debunked using the web app for video fragmentation and reverse image search.

Fake news	Claim	Date		Original source	Fact	Date
https://www.facebook.com/ 100009631064968/videos/ 730611413936554/	Muslims attack cars in Birmingham UK	May 2018		https://www.youtube.com/ watch?v=rAoQTQE_YTY	Basel hooligans attacked Zurich in cars	May 2018
https://www.youtube.com/ watch?v=DM3fUN4qvp4	Fight of drivers in Brazil	May 2018				
https://twitter.com/ tprincedelamour/status/ 843421609159544836	Migrant attacks nurse in public hospital in France	Mar. 2017		https://www.youtube.com/ watch?v=CuyfdZKc3TQ	Drunk patient beats up doctors in Novgorod hospital in Russia	Feb. 2017
https://twitter.com/kwilli1046 /status/872106123570163712	Attack in Notre Dame, Paris	Jun. 2017		https://www.youtube.com/ watch?v=W2IA9UwmHCA	Filming World War Z (movie)	Sep. 2012
https://twitter.com/mikethecraigy /status/877248566384873472	Attack in Brussels Central Station	Jun. 2017				
https://twitter.com/FuegoNugz /status/905246797123203072	Hurricane Irma in Barbados, US	Sep. 2017		https://www.youtube.com/ watch? v=0IHDVel-NPw	Hurricane Dolores in Uruguay	May 2016
https://www.youtube.com/ watch?v=fmUEI0L2alY	Hurricane Otto in Panama	Nov. 2016				

4.3 API Layer and Integration with InVID

As described in Sections 4.1.5 and 4.2.5 of D2.2, the supported functionality by the video fragmentation and annotation component is being made available to the InVID platform and applications through the API layer of the REST service that hosts the technologies for video segmentation (into scenes, shots and sub-shots), thumbnail extraction and concept-based video (and video fragment) labeling. Building

on the system described in D2.2 and guided by the conducted evaluations of this technology during the iterative test and validation cycles of InVID, over the last 12 months of the project's life we improved several aspects of the web service (both in terms of hardware and software) aiming to address the processing and usage requirements of the InVID technologies that integrate this analysis component.

The video fragmentation and annotation service is being used by:

- the InVID Multimodal Analytics Dashboard, which requires the fragmentation and conceptual annotation of the collected video items from the monitored social media platforms;
- the InVID Verification Application, which requests the fragmentation and conceptual annotation of a video (or a small video collection) under evaluation;
- the InVID Verification Plugin, which utilizes the sub-shot fragmentation functionality of the web service via the integrated tool for video fragmentation and keyframe-based reverse video search;
- the individual InVID partners who evaluate the performance of this API during the test and validation cycles of the project.

As a starting point, after establishing the communication between the web service and the aforementioned integrated technologies of InVID (a process that involved the alignment of the service's responses with the formatting demands of other tools and services of the InVID system), we tried to identify particular requirements and patterns of use, that could guide our efforts for further improvement of this technology. The analytics regarding the use of the service, that is (reasonably) dominated by the InVID Multimodal Analytics Dashboard which collects and submits large volumes of video content for analysis on a daily basis, indicated the necessity to expand the processing capacity of the service in order to allow the analysis of larger amounts of video content. To address this challenging problem we worked in collaboration with webLyzard, who is responsible for the development of the InVID Multimodal Analytics Dashboard, on two different directions:

- webLyzard tried to fine-tune the video collection component of the dashboard, aiming to filter-out videos that according to their (con-)textual metadata seem to be worthwhile to collect and annotate but in reality they correspond to documentaries, news shows and videos of first person video games (a.k.a. "let's play" videos), thus having extremely low possibility of being newsworthy UGVs;
- CERTH worked towards the upgrade of the used infrastructure (i.e. processing units/cores) by the video fragmentation and annotation component, and the establishment of a usage plan that enables a more time-efficient and balanced use of this service by the individual components and roles of the InVID ecosystem.

To meet the first goal, i.e. the upgrade of the utilized processing resources, we installed a clone of the service in another machine (with similar specifications with the initially used one) and we built a load balancing mechanism who is responsible for distributing the incoming traffic (i.e. analysis requests, requests about the status of the analysis, the status of the service and the generated analysis results) in the two machines that now host the service. The load balancer was built on top of the initially utilized REST service, in a way that makes the applied changes for extending the processing capacity of this analysis component completely invisible to other technologies that integrate this service. As before, the base URL of the service is: `http://multimedia2.iti.gr:8080` and the access to the service is permitted only to authorized users with a valid user key. However, the performed upgrade of the used infrastructure, combined with the improvement of the time performance of the image/video annotation component, approximately doubled the processing capacity of the web service.

To address the second goal, i.e. to avoid very long waiting times due to uninterrupted use of the service by the InVID Multimodal Analytics Dashboard and to enable a time-efficient use of the service by all the different components and roles of InVID, we integrated a mechanism that relies on specific rules about the priority of an incoming analysis request. These rules are assigned to the different user keys generated for the needs of the project, and can be summarized as follows:

- 1st level priority is given to analysis requests coming from the InVID Verification Plugin, and the InVID Verification Application when the processing of a single video is requested;
- 2nd level priority is given to analysis requests coming from the InVID partners who are responsible to evaluate the API during the test and validation cycles of the project;

- 3rd level priority is given to analysis requests coming from the InVID Verification Application when the batch processing of a small video collection is requested;
- 4th level priority is given to analysis requests coming from the InVID Multimodal Analytics Dashboard when the batch processing of a large video collection is requested.

In addition to the changes described above, a number of software-related improvements on several other aspects of service use were made. These improvements were decided based on the feedback collected after the performed test and validation cycles of the project, and include:

- the update of the image/video concept-based labeling component in order to exploit the detection accuracy and time efficiency of the latest developed concept detection algorithm.
- the modification of the communication protocol between the service and the user that submits an online video for analysis; an instant check on the existence of the video is performed right after receiving the analysis request in order to inform the user about any broken URL or non-existing video file.
- the enhancement of the video fetching mechanism of the service; for this purpose we: a) integrated another component for video downloading (namely the "youtube-dl" software component) on top of the previously used one (i.e. the "you-get" software component), b) updated this mechanism in order to make it less restrictive concerning the format of the Dropbox video URLs, and c) enabled the automatic update of the used third party components for video downloading; as a consequence of the above changes, the service now supports the analysis of videos from YouTube, Facebook, Twitter, DailyMotion, Dropbox, Vimeo, Instagram and LiveLeak.
- the alteration of the service's strategy for indexing the analysis results of the processed videos, in a way that permits the analysis of the same video by the different users of the service without any conflict regarding the availability of the generated results.
- the addition of a new REST call reporting the number of the queued analysis requests and the priority of a given request (i.e. its position in the queue) that allows other tools and components of the InVID system to make an estimation concerning the needed time for the completion of the processing and the generation of the analysis results.
- the update of the process for the generation of the JSON file with the analysis results in order to include: a) general information about the file (e.g. date-time of generation, expiration time, service's version etc.), b) appearance time of the selected thumbnails, c) the top-10 detected concepts at the video level that can be used as a concept-based summary of the visual content of the video, and d) the top-30 detected concepts for each video fragment instead of the exhaustive list of concepts for each fragment; the latter change allowed us to convey the most meaningful and useful information through the generated JSON file, while reducing remarkably its file-size.
- the enrichment of the service's status report, in order to provide more detailed information regarding the status of an analysis request and the progress of the performed analysis.
- the immediate removal of any downloaded video right after the completion of its analysis to prevent the servers to run out of memory (due to the volume of video content submitted for analysis on a daily basis); the generated analysis results (i.e. JSON file with structured data about the video fragments and their concept-based annotation, extracted keyframes and selected thumbnails) are stored in the server and being available for the other components of the InVID platform for 2 weeks as agreed with the InVID partners.
- the implementation of an internal process that enables the seamless operation of the service even after a service update or restart.

All the aforementioned changes were properly reflected in the updated documentation of the web service. The impact of these changes, mainly the ones related to the upgrade of the used infrastructure, the improvement and update of the image/video concept-based labeling component, and the enhancement of the video downloading mechanism, is highlighted in Table 9 below. As shown in this table, the processing capacity (expressed in terms of hours of video processed on a daily basis) has been almost doubled and the service is now capable of analysing more than 110 hours of video content per

day. Moreover, the fixing of bugs and errors in our software components led to further reduction of the, already small, number of failures due to errors in analysis. Last but not least, the improvement of the video mechanism part of the service almost eliminated the failures due to errors in the video fetching process, while the small number of analysis requests that were not processed due to unsupported video format or broken URL has been further reduced to 0.15%.

These numbers indicate that the REST service for video fragmentation and annotation is a reliable analysis component that can adequately and effectively support the processing needs of the InVID platform and integrated technologies.

Table 9: Comparison of the service's performance before and after the applied improvements.

	Mean daily processing capacity	Failure of video analysis	Failure of video downloading mechanism	Unsupported video format or broken URL
Before the service update	∼65 hours of video content	0.76%	2.28%	1.15%
After the service update	∼116 hours of video content	0.12%	0.01%	0.15%

In the following we briefly summarize the I/O (input/output) of the service to facilitate the understanding of the usage of this analysis component.

As input, the service takes the URL of the video file that needs to be analyzed; this URL can link to a video file hosted in online repositories (both FTP and HTTP), or found in video/file sharing platforms and social networks (see the bulleted list above for the currently supported online sources). As output, the service generates: a) a JSON file with the video fragmentation and annotation analysis results, b) two collections of image files that correspond to the extracted keyframes for the detected shots and sub-shots of the video, and c) a collection of image files that correspond to the selected thumbnails for the video. To submit a video for video fragmentation analysis, the user must commit an HTTP POST request on `http://multimedia2.iti.gr:8080/segmentation`, while in the case of video fragmentation and annotation analysis the HTTP POST request should be committed on `http://multimedia2.iti.gr:8080/segmentation-annotation`. The body of the HTTP POST request contains the following parameters:

- "video_url": the URL of the video to be processed
- "login", "password": optional parameters used for authentication checks in case of password-protected repositories
- "user_key": a unique 32-digits access key that allows access to the service
- "kf_num_sh": an optional argument that defines the number of extracted keyframes per video shot (default value is 3)
- "kf_num_sb": an optional argument that defines the number of extracted keyframes per video sub-shot (default value is 3)
- "thumb_num": an optional argument that defines the number of extracted thumbnails for the video (default value is 3)

The communication between the web service and the user is synchronous only during the transmission of the call. As reported before, the service checks the existence of the submitted video right after the receipt of the analysis request, and informs the user about a number of different violated conditions (e.g. wrongly formatted analysis request, non-existing video file, broken video URL) that prevent the initialization of the analysis. If the file exists, the service proceeds by assigning an identifier to the analysis request and notifies the user about this identifier, since the latter is necessary for monitoring the status of the analysis and retrieving the analysis results. The former is performed by committing an HTTP GET request on `http://multimedia2.iti.gr:8080/status/<video_id>`, where "video_id" is the automatically assigned identifier to the video file. The latter is performed through a set of specific HTTP GET requests that enable the retrieval of the extracted keyframes and thumbnails of the video (either on a one-by-one basis or as an entire collection), and the JSON file with the analysis results.

Table 10: Datasets (and their statistics) used for evaluating concept detection.

Dataset	Training Instances	Testing Instances	Training set Concepts	Test set Concepts	Concept Cardinality	Label Cardinality	Missing Labels
TRECVID-SIN	239495	112677	346	38	3206.3	2.2	294.6
PASCAL-VOC2012	5717	5823	20	20	416.6	1.5	0.0
PASCAL-VOC2007	5011	4952	20	20	379.2	1.4	0.0
NUS-WIDE	161789	107859	81	81	3066.1	1.9	0.0

4.4 Video Fragmentation and Conceptual Annotation Evaluation

This section reports the findings of the conducted evaluations regarding the performance of the developed approach for concept-based video/image annotation. The accuracy of the generated annotations was assessed with the help of datasets and metrics that are widely used in international benchmarking activities, such as the TRECVID SIN task. The time efficiency of the implemented architecture was assessed in terms of needed time for training and testing. Finally the effectiveness of the current algorithm was compared against other methods from the relevant literature, and the progress made during the last year is highlighted through the comparison with the algorithms reported in D2.2.

4.4.1 Datasets and Experimental Setup

Our experiments were performed on four large multi-label video/image classification datasets, namely the TRECVID-SIN 2013 (Over & et al., 2013), the PASCAL-VOC 2007 (Everingham, Van Gool, & et al., n.d.), the PASCAL-VOC 2012 (Everingham, Van Gool, Williams, Winn, & Zisserman, n.d.), and the NUS-WIDE (Chua et al., July 8-10, 2009), presented in Table 10. Label cardinality (i.e. the average number of concepts presented per image/video shot), concept cardinality (i.e. the average number of positive images/video shots per concept), and missing labels (i.e. the average number of non-annotated labels per image/video shot) have been calculated on the training set for each dataset. For assessing concept annotation performance, the indexing problem as defined in (Over & et al., 2013) was evaluated, i.e. given a concept, the goal was to retrieve the 2000 video shots (or images, depending on the dataset) that are mostly related with it.

The TRECVID-SIN 2013 (Over & et al., 2013) 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. The PASCAL-VOC 2007 (Everingham, Van Gool, & et al., n.d.) dataset consists of 5011 training and validation images and 4952 test images. The PASCAL-VOC 2012 (Everingham, Van Gool, Williams, et al., n.d.) dataset consists of 22531 images divided into training, validation and test sets (5717, 5823 and 10991 images, respectively). We used the training set to train the various methods of our study, and evaluated them on the validation set. We did not use the original test set because ground-truth annotations are not publicly available for it (the evaluation of a method on the test set is possible only through the evaluation server provided by the PASCAL-VOC competition, submissions to which are restricted to two per week). Both for the PASCAL-VOC 2007 and 2012 the images are annotated with 20 object classes. The NUS-WIDE (Chua et al., July 8-10, 2009) dataset consists of 269648 Flickr images that have been annotated with 81 semantic concepts. We used a subset of 161789 images for training and the rest of them for testing. Since the available ground-truth annotations for each of the four datasets are not adequate in number in order to train a deep network from scratch without over-fitting its parameters, similarly to other studies (Pittaras et al., 2017), we used transfer learning. I.e. we used as a starting point the ResNet-50 network (He et al., 2016), which was originally trained on 1000 ImageNet categories (Russakovsky, Deng, & et al., 2015), and fine-tuned its parameters towards each of these four datasets.

In order to evaluate the methods' performance in the PASCAL-VOC 2007, 2012 and NUS-WIDE datasets we used the mean average precision (MAP) measure, while, the mean extended inferred average precision (MXinfAP) (Yilmaz, Kanoulas, & Aslam, 2008), which is an approximation of MAP, was used for the TRECVID-SIN dataset. MXinfAP is suitable for the partial ground-truth that accompanies the latter dataset. Both of these measures are predominantly used in the literature for evaluating the performance of state of the art methods for concept detection, as well as within international benchmarking activities that assess and compare the performance of such methods (Simonyan & Zisserman, 2014; Wei et al., 2016; M. Wang et al., 2016).

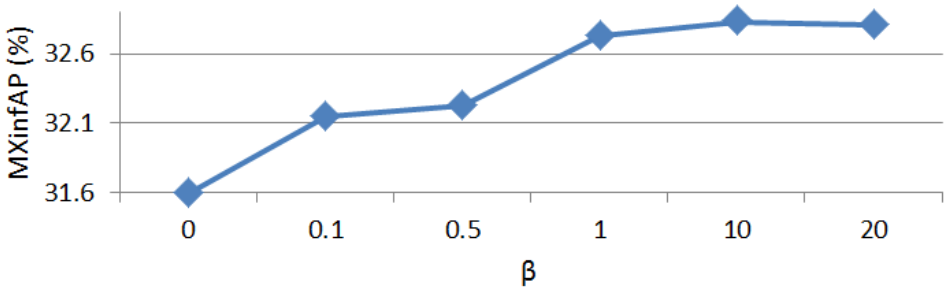

Figure 8: MXinfAP (%) for different values of β (Eq 5) for the proposed FV-MTL with CCE-LC cost.

4.4.2 Implementation Details

For the rest of this section, when DCNN training takes place we did it by using the pre-trained ResNet-50 ImageNet network (He et al., 2016) (removing the last classification layer) and fine-tuning it on the target concept annotations. The network's learning rate and momentum was set to 10^{-5} and 0.9, respectively, whereas the mini-batch size was restricted by our hardware resources and set to 32. Multi-label stratification was used in order to ensure similar distribution of positive examples per class on each batch. Stochastic gradient descent (SGD) was used as the network's optimization function. All networks were trained and implemented in Caffe (Jia et al., 2014). Regarding the proposed method, the new layers learning rate and momentum were set to 0.1 and $5 \cdot 10^{-4}$, respectively, and β was set to 10. This value for β was chosen based on preliminary experiments on the TRECVID SIN dataset (Fig. 8) that showed that this is an appropriate value, and also that the proposed approach is not sensitive to the value of β. The diagonal of the $\boldsymbol{\Phi}$ correlation matrix was set to zero. The model parameter values with respect to the compared methods were either selected experimentally or following the typical heuristics and strategies proposed in the corresponding works. We conducted our experiments on two NVIDIA TitanX GPUs.

Each trained DCNN was used in two different ways to annotate new images/keyframes with semantic concepts: a) As a standalone classifier, where each test image/keyframe was forward-propagated by the network and the network's output was used as the final class distribution that was assigned to the image/keyframe. b) As a feature generator, where the training set was once again forward-propagated by the network, and the values calculated in the last layer of the network were used as feature vectors to subsequently train one Support Vector Machine (SVM) classifier per concept. Then, each test image was firstly forward-propagated by the DCNN to extract the features and subsequently was served as input to the trained SVM classifiers.

4.4.3 Preliminary Experiments - Design Choices

In Table 11 we examine the best way of using the proposed FV-MTL with CCE-LC cost by comparing different parameters and intermediate versions of them. We performed this set of experiments on the TRECVID-SIN dataset using as a starting point the ResNet-50 network.

- As a baseline we used the extension strategy proposed in (Pittaras et al., 2017), i.e. the DCNN architecture illustrated in Fig. 4 (ii). The results are presented in Table 11: (d)). The dimensionality of the extension layer (Fig. 4: step (c)) is indicated in Table 11: (a). Sigmoid cross-entropy was used as the network's cost function.

- We compared the baseline approach with: i) The proposed CCE-LC cost when used on the top of the baseline DCNN architecture, replacing the sigmoid cross-entropy cost (Table 11: (e)), i.e. the FV-MTL method was ignored. ii) The proposed FV-MTL with CCE-LC, where for the latter parameter β was set to 0, i.e. the concept correlation term λ_2 in Eq. 5 was ignored (Table 11: (f)). iii) The complete proposed FV-MTL with CCE-LC cost for $\beta = 10$, i.e. both cost terms, λ_1 and λ_2, were considered (Table 11: (g)). Each row of Table 11 corresponds to a different dimension of our FV-MTL first FC layer (shown in Fig. 4: step (f)).

Each of the above DCNN architectures was fine-tuned on the 346 TRECVID-SIN concepts using the TRECVID development dataset (Over & et al., 2013). Using these results, we assess i) how the number of the latent tasks k and feature dimensionality d affect FV-MTL (Table 11: (a)-(c)), ii) the usefulness of exploiting semantic-level (explicit) concept relations using the CCE-LC cost instead of the typical

Table 11: Performance (MXinfAP, %) for different dimensions of the columns of the $\boldsymbol{L_x}$ matrix (Fig. 4 step (e)) that we used in the experiments.

$\boldsymbol{L_x}$ **#columns** $d \times k$	**Latent tasks** k	**Feature dimension** d	**DCNN (extension strategy (Pittaras et al., 2017)) with STL sigmoid cross-entropy**	**Proposed CCE-LC cost**	**Proposed FV-MTL with CCE-LC** ($\beta = 0$)	**Proposed FV-MTL with CCE-LC** ($\beta = 10$)
(a)	(b)	(c)	(d)	(e)	(f)	(g)
128	4	32	23.18	28.76	30.01	29.43
256	4	64	26.91	30.84	30.50	31.38
512	8	64	28.44	30.95	30.37	31.92
1024	16	64	29.76	31.21	30.25	32.1
2048	32	64	30.95	32.44	31.60	**32.83**
4096	32	128	31.06	31.94	31.65	32.02
4096	64	64	31.06	31.94	31.71	32.07

Table 12: Comparison of the complete FV-MTL with CCE-LC (for $\beta = 10$) and two intermediate versions of it, with other methods on the three datasets.

Category	**Method**	**TRECVID-SIN**		**PASCAL-VOC2007**		**PASCAL-VOC2012**		**NUS-WIDE**	
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
		direct	last layer	direct	last layer	direct	last layer	direct	last layer
i) Baseline (without fine-tuning)	ResNet-50 (He et al., 2016) as feature generator	29.21	29.78	83.90	83.76	82.98	83.04	51.30	56.20
ii) Typical DCNN fine-tuning	ResNet-50 (He et al., 2016)	27.35	28.66	76.38	83.06	81.20	82.15	51.17	56.32
iii) DCNNs (extension strategy (Pittaras et al., 2017)) with STL cost functions	Hinge-loss	29.08	30.06	78.32	79.23	86.6	87.23	52.80	57.49
	Sigmoid cross-entropy	31.06	32.2	80.74	84.97	86.94	86.80	53.94	57.20
	CCE (Bishay & Patras, 2017)	31.93	32.52	84.07	84.92	85.52	85.39	54.58	55.0
	DWE (M. Wang et al., 2016)	28.03	29.17	77.25	78.12	85.14	86.00	51.10	56.08
iv) MTL for DCNNs or shallow linear models	AMTL (Sun et al., 2015)	29.36	30.15	83.15	84.37	83.17	84.05	53.40	54.22
	CMTL (Zhou et al., 2011)	29.89	30.45	83.44	84.42	83.55	84.60	51.80	52.40
	2-sidedNN (Yang & Hospedales, 2015)	29.91	30.01	83.50	84.53	83.70	84.45	51.97	52.67
v) Structured outputs	Stacking-LP (Markatopoulou et al., 2015)	30.01	31.05	84.68	85.12	84.25	85.30	51.96	52.98
	LMGE (Yang et al., 2012)	30.17	31.24	84.32	85.02	84.52	85.64	53.07	54.62
vi) Joint MTL + Structured outputs	ELLA_LC (Markatopoulou et al., 2016b)	28.15	29.09	81.98	82.84	82.15	83.17	52.40	54.68
	DMTL_LC (Markatopoulou et al., 2016a)	28.23	31.71	82.01	84.07	82.23	84.30	52.35	54.70
vii) Proposed	CCE-LC cost	32.44	33.55	85.40	86.73	86.32	86.39	**56.40**	**60.73**
	FV-MTL with CCE-LC ($\beta = 0$)	31.60	32.15	82.21	86.96	87.10	88.51	55.45	54.69
	FV-MTL with CCE-LC ($\beta = 10$)	**32.83**	**33.77**	**85.70**	**87.00**	**87.54**	**88.69**	55.54	60.22

sigmoid cross-entropy cost, iii) the usefulness of exploiting visual-level (implicit) concept relations using the proposed FV-MTL with CCE-LC when ignoring the concept correlation term λ_2 in Eq. 5 (Table 11: (f)), and iv) the usefulness of jointly exploiting visual-level and semantic-level concept relations by adopting MTL and structured output prediction using the proposed FV-MTL with CCE-LC cost when both cost terms (λ_1, λ_2 in Eq. 5) are considered (Table 11: (g)). It should be noted that our proposed FV-MTL with CCE-LC cost is most beneficial when used on datasets with non-exclusive labels (e.g. TRECVID SIN, PASCAL-VOC, NUS-WIDE) where CCE-LC can exploit and capture concept correlations across the labels. Such concept correlations are missing in single-label classification datasets such as ImageNet.

The choice of parameter k, which determines the number of latent tasks, is important because it determines the amount of sharing between the tasks. If k is very high, the tasks are not forced to share information with each other. On the other hand, if k is very low, the latent space may shrink too much. In Table 11 we compare different values for this parameter in order to see how it affects the proposed FV-MTL method (Table 11: (f), (g)). We observe that the larger the value of k the better the accuracy of the FV-MTL approach. According to the rest of the results, we observe that structured output prediction using the proposed CCE-LC cost (Table 11: (e)), and MTL using the proposed FV-MTL approach (Table 11: (f)) are two different ways to improve concept annotation accuracy, as according to Table 11 the two methods always outperform the baseline (Table 11: (d)). Jointly using MTL and structured output prediction, in a DCNN architecture (Table 11: (g)) almost always outperforms all the other methods, reaching the best result of 32.83% when parameter k equals to 32 and parameter d equals to 64, i.e. the columns of $\boldsymbol{L_x}$ equal to 2048. One exception is seen in the first row of Table 11, where we observe a small decrease in performance of $\beta = 10$ compared to $\beta = 0$. This is due to the low number of feature dimensions and latent tasks, which are not sufficient for the CCE-LC term to capture well the correlation information.

4.4.4 Main Findings - Comparisons With Related Methods

Table 12 compares the proposed complete FV-MTL with CCE-LC (for $\beta = 10$) with other related methods on the three datasets. The used metrics are MXinfAP (%) for 38 TRECVID-SIN and MAP (%) for 20

PASCAL-VOC2007, 20 PASCAL-VOC2012 and 81 NUS-WIDE concepts. In addition, we evaluate the two intermediate versions of our complete DCNN architecture that were also evaluated in Table 11. I.e. a) Extension strategy (Pittaras et al., 2017) for DCNNs with the proposed CCE-LC cost, i.e. the typical complete DCNN architecture illustrated in Fig. 4 replacing the sigmoid cross-entropy cost with the proposed CCE-LC cost, and b) FV-MTL with CCE-LC for $\beta = 0$. We set k equal to 32 and d equal to 64, which was the pair that reached the best overall MXinfAP according to Table 11; similarly, in the case that CCE-LC is used alone the dimension of the extension layer was set to 2048. We performed comparisons with the following methods:

- i) A baseline where we use the ResNet-50 pre-trained network as feature generator; one SVM classifier per concept was trained using as features either the ResNet's output or its last FC layer.
- ii) The typical DCNN architecture with sigmoid cross-entropy cost, i.e. the ResNet-50 pre-trained network fine-tuned on each of the four datasets by simply replacing the classification layer with a new layer with dimensions that equal to the number of concepts in the target domain as illustrated in Fig. 4 (i).
- iii) Extension strategy (Pittaras et al., 2017) for DCNNs, i.e. the DCNN architecture illustrated in Fig. 4 (ii), and four different STL cost functions: a) hinge-loss, b) sigmoid cross-entropy, c) cost sigmoid cross-entropy (CCE) (Bishay & Patras, 2017), an extended version of (b) that also addresses the class-imbalance problem, and d) dynamic weighted euclidean loss (DEW) (M. Wang et al., 2016), an extension of the euclidean loss suitable for multi-label classification giving a greater penalty to concept prediction scores that have been ranked higher than the negative ground-truth annotated concepts. The size of the extension layer was set to 4096, according to the findings of Table 11. This category of methods uses exactly the same architecture with the first intermediate version of our complete architecture (denoted as a) above), with the difference that each of the above three cost functions is used instead of the CCE-LC cost.
- iv) MTL, either as an integral part of DCNNs or for shallow linear models: a) AMTL (Sun et al., 2015), b) CMTL (Zhou et al., 2011) and c) the 2-sided NN that was proposed in (Yang & Hospedales, 2015) for solving the GO-MTL method objective function (Kumar & Daume, 2012).
- v) Structured output prediction: a) Stacking-LP (Markatopoulou et al., 2015), a two-layer stacking architecture combined with the label power-set algorithm (Markatopoulou et al., 2015). b) LMGE (Yang et al., 2012), an inner learning approach that uses the extracted features and exploits concept correlations in a single step.
- vi) Methods jointly using MTL and structured output prediction: a) DMTL_LC (Markatopoulou et al., 2016a), and b) ELLA_LC (Markatopoulou et al., 2016b).

We selected all the parameter values for these methods based on the training data, and in accordance with the recommendations provided in the corresponding papers.

We apply and evaluate all the above methods in two different ways (in a direct analogy to what is discussed in the last paragraph of Section 4.4.2); the specifics of these depended on whether they are complete DCNN architectures or shallow models that use pre-computed DCNN features. To the first category belong the following methods: Typical DCNN fine-tuning (group (ii)), all methods of group (iii) above, the 2-sided NN of (Yang & Hospedales, 2015), DMTL_LC (Markatopoulou et al., 2016a), the proposed FV-MTL with CCE-LC and the latter two intermediate versions. These methods are used a) as standalone classifiers, where the direct output of the complete network is evaluated (denoted as "direct" in Table 12), b) as feature generators, where SVM classifiers are trained on DCNN-based features. In the latter case, the output of the last layer of the complete trained network for each method was used as a feature vector to train one SVM per concept (denoted as "last layer" in Table 12). The remaining methods (that belong to the second category), i.e. the baseline of group (i) above, AMTL (Sun et al., 2015), CMTL (Zhou et al., 2011), ELLA_LC (Markatopoulou et al., 2016b), Stacking-LP (Markatopoulou et al., 2015) and LMGE (Yang et al., 2012), use the pre-trained ResNet-50 network as feature generator and the extracted features were used to train each of these methods. The methods specifically used in our experiments a) the ResNet-50 output layer (denoted as "direct" in Table 12), b) the ResNet-50 last FC layer (denoted as "last layer" in Table 12).

Table 12 presents the results in terms of MXinfAP for the TRECVID-SIN dataset and in terms of MAP for the PASCAL-VOC and NUS-WIDE datasets. With respect to the direct output (Table 12: (a),(c),(e),(g)) we observe that the two intermediate versions of our proposed method perform quite

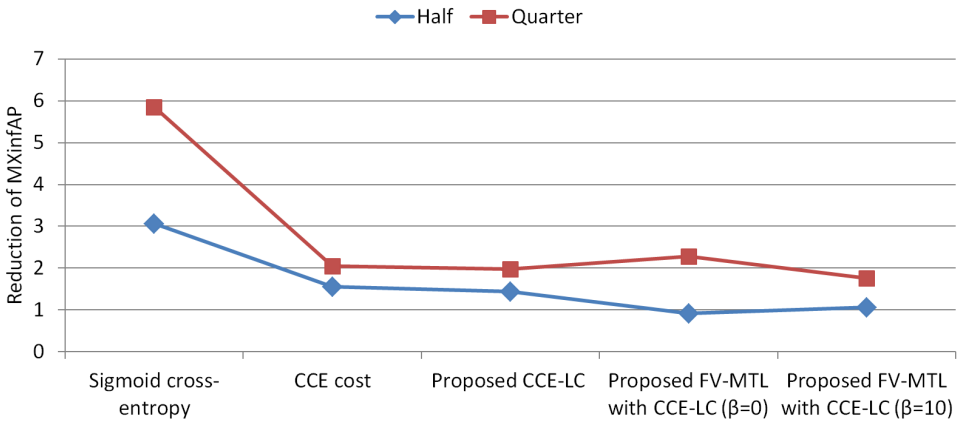

Figure 9: Reduction of MXinfAP when only a half and a quarter of the training samples respectively are used.

well, outperforming the compared methods in the majority of cases. One exception is observed between the compared extension strategy (Pittaras et al., 2017) with sigmoid cross-entropy cost and the proposed FV-MTL with CCE-LC for $\beta = 0$, where their difference is that the latter also incorporates MTL. The results present fluctuations concerning which of the two methods performs better, depending on the dataset. However, jointly combining MTL and structured output prediction, using the proposed FV-MTL with CCE-LC for $\beta = 10$, further improves the concept annotation accuracy and outperforms all the other previously-published methods across all of the evaluated datasets, reaching the best overall concept annotation accuracy of 32.83%, 85.70%, 87.54% and 55.54% for TRECVID-SIN, PASCAL-VOC2007, PASCAL-VOC2012 and NUS-WIDE, respectively. The only exception is the NUS-WIDE dataset, where our intermediate version of the typical extension strategy with CCE-LC cost presents the best accuracy, and our complete architecture reaches the second-best performance. It should be noted that we compare our method with very recent methods; even our baseline is the ResNet-50 network that was ranked first in the ImageNet 2016 competition and our method outperforms it by approximately 3 to 4 percentage points. Similarly clear differences can be observed with respect to all the other compared methods. Even compared to the most recent DCNN with CCE cost (Bishay & Patras, 2017), although the differences are smaller, we consistently outperform it by approximately 1 to 1.5 percentage points in all three datasets. Similar conclusions can be reached regarding the results presented in columns (b), (d), (f) and (h) of Table 12 that refers to the second way of applying the compared methods, as described in the beginning of this section. We also evaluated the XinfAP per task regarding the proposed FV-MTL with CCE-LC and the other two best performing methods (i.e. DCNN with sigmoid cross-entropy cost and DCNN with CCE cost (Bishay & Patras, 2017)) in the TRECVID-SIN dataset. Besides our overall best result (33.77% - Table 12), our method performs better than these other two well-performing methods for 25 out of the 38 evaluated concepts.

To investigate the statistical significance of the difference of the results of each method from the best performing method, i.e. the proposed FV-MTL, we used a paired t-test as suggested by (Blanken, de Vries, Blok, & Feng, 2005). We found that differences between the proposed FV-MTL with CCE-LC ($\beta = 10$) and all other previously-published methods that we compare with, per column of Table 12, are significant at 5% significance level.

Finally, we assess the robustness of the proposed and the other two best performing methods (i.e. Sigmoid cross-entropy, and CCE costs (Pittaras et al., 2017), (Bishay & Patras, 2017)) with respect to the TRECVID SIN dataset according to Table 12 , when they are trained on smaller datasets for the same number of concepts. Specifically, Fig. 9 presents the reduction of MXinfAP when each of the compared methods is trained a) on only half of the keyframes of TRECVID SIN training set and b) on only a quarter of the keyframes for the same dataset, compared to the complete training set. We observe that the DCNN with sigmoid cross-entropy cost is affected by the smaller training datasets, as according to Fig. 9 its concept annotation accuracy is reduced by approximately 6 and 3 percentage points when the half and quarter training sets are used instead of the complete training set, respectively. In contrast, the proposed FV-MTL with CCE-LC for $\beta = 10$ and its intermediate versions, i.e. CCE-LC cost and FV-MTL with CCE-LC for $\beta = 0$, are robust to smaller training sets, exhibiting only a small reduction of MXinfAP compared to the case of using the complete training set.

Table 13: Mean execution training/testing times in hours.

Category	Method	TRECVID-SIN	
		training	testing
i) Baseline (without fine-tuning)	ResNet-50 (He et al., 2016) as feature generator	14.25	2.45
ii) Typical DCNN fine-tuning	ResNet-50 (He et al., 2016)	17.33	2.47
iii) DCNNs (extension strategy (Pittaras et al., 2017)) with STL cost functions	Hinge-loss	17.35	2.48
	Sigmoid cross-entropy	17.45	2.47
	CCE (Bishay & Patras, 2017)	17.75	2.50
	DWE (M. Wang et al., 2016)	17.85	2.50
iv) MTL for DCNNs or shallow linear models	AMTL (Sun et al., 2015)	14.75	2.50
	CMTL (Zhou et al., 2011)	14.85	2.58
	2-sidedNN (Yang & Hospedales, 2015)	48.12	6.80
v) Structured outputs	Stacking-LP (Markatopoulou et al., 2015)	23.15	4.51
	LMGE (Yang et al., 2012)	15.17	2.68
vi) Joint MTL + Structured outputs	ELLA_LC (Markatopoulou et al., 2016b)	20.97	2.53
	DMTL_LC (Markatopoulou et al., 2016a)	49.27	6.84
vii) Proposed	CCE-LC cost	17.75	2.67
	FV-MTL with CCE-LC ($\beta = 0$)	17.53	3.17
	FV-MTL with CCE-LC ($\beta = 10$)	18.15	3.10

4.4.5 Execution Times

We continue the analysis of our results by assessing the execution times during the training and classification phase of the different methods compared in this study. Table 13 summarizes the required execution time in hours for the proposed FV-MTL with CCE-LC for $\beta = 10$ and its two intermediate versions, defined in earlier sections, and also compares it with the rest of the methods. We observe that the proposed method is not considerably more computationally expensive than DCNN methods that use STL cost functions. Training of the baseline, AMTL and CMTL methods that use pre-computed features is a bit faster than the proposed method and its intermediate versions; however, all these previous methods achieved lower accuracy than the proposed one, according to Table 12. During classification all the compared methods are executed on very similar time, except for the 2-sidedNN, Stacking-LP and DMTL_LC that are significantly slower. We conclude that our proposed FV-MTL with CCE-LC is faster than other MTL methods for DCNNs (2-sidedNN, DMTL_LC) both during training and classification, and also comparable in execution time with the second-best performing method of Table 12, i.e. DCNN with CCE cost (Bishay & Patras, 2017).

4.4.6 Data Augmentation and Comparisons

Recently, improved accuracy has been achieved by image augmentations, i.e. feeding the DCNN with more than one image crops of the same image. For example, in the PASCAL-VOC2007 dataset this was shown to improve the MAP by 6 percentage points (Wei et al., 2016). In Table 14 we compare our proposed method with these approaches, however, due to the fact that this is a very computational intensive and time consuming process we present results only on the PASCAL-VOC2007 dataset. The following three SoA PASCAL-VOC2007 methods were selected: (i) Simonyan et al. (Simonyan & Zisserman, 2014): A pre-trained ImageNet DCNN is applied on multiple image representations that are extracted and aggregated across multiple locations and scales. The resulting aggregated image descriptor (using the second-last layer as image feature representation) is used to train a linear SVM per concept. (ii) Wei et al. (Wei et al., 2016): Many object segment hypotheses are given as input to a shared DCNN that has been pre-trained in the ImageNet dataset. The shared network's output is aggregated with max pooling in order to return a single multi-label prediction. The shared network is fine-tuned on the PASCAL-VOC dataset. (iii) Wang et al. (M. Wang et al., 2016): Similar to Wei et al. (Wei et al., 2016), a pre-trained ImageNet DCNN is fine-tuned using many random crop hypotheses. Stochastic scaling and cropping over images is performed in this case in order to choose the most useful image crops by proposing

Table 14: MAP (%) for 20 PASCAL-VOC2007 concepts for methods that use image augmentations.

Method	**PASCAL-VOC2007**
Simonyan et al. (Simonyan & Zisserman, 2014)	89.3
Wei et al. (Wei et al., 2016)	90.9
Wang et al. (M. Wang et al., 2016)	92.5
FV-MTL with CCE-LC ($\beta = 10$) + augmentations (VGG16)	**93.3**
FV-MTL with CCE-LC ($\beta = 10$) + augmentations (ResNet-50)	**94.6**

Table 15: Comparison of the current method against the algorithms reported in D2.2.

Category	**Method**	**TRECVID-SIN**		**PASCAL-VOC2007**		**PASCAL-VOC2012**		**NUS-WIDE**	
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
		direct	last layer	direct	last layer	direct	last layer	direct	last layer
D2.2 method A	FT3-ex	31.06	32.2	80.74	84.97	86.94	86.80	53.94	57.20
D2.2 method B	DMTL_LC	28.23	31.71	82.01	84.07	82.23	84.30	52.35	54.70
Current	FV-MTL with CCE-LC ($\beta = 10$)	**32.83**	**33.77**	**85.70**	**87.00**	**87.54**	**88.69**	**55.54**	**60.22**

the random crop pooling approach (RCP), instead of object proposal methods such as (van de Sande, Uijlings, Gevers, & Smeulders, 2011), (Wei et al., 2016). Furthermore, the DWE loss function, also presented in Table 12, is used on the top of the network. One linear classifier is trained finally for each object class. Concerning the proposed architecture (FV-MTL with CCE-LC ($\beta = 10$)), this is fine-tuned on 20 random image object segment proposals per image extracted using the RCP method (M. Wang et al., 2016). Similarly to (Wei et al., 2016) and (M. Wang et al., 2016) a shared DCNN is used to aggregate the probability scores w.r.t. each proposal using max-pooling and one SVM is trained for each object class. In this set of experiments, we firstly used the VGG16 (Simonyan & Zisserman, 2014) ImageNet pre-trained network as the base network of the proposed architecture, in order to have a fair comparison with methods (Simonyan & Zisserman, 2014), (Wei et al., 2016) and (M. Wang et al., 2016) that also use VGG16. Then, similarly to all of our previous experiments, we repeated this experiment using ResNet-50 (He et al., 2016) as our architecture's base network. We observe that the proposed method once again outperforms all the other compared methods and also that image augmentation is a robust way of increasing the accuracy of our architecture by approximately 7 percentage points.

4.4.7 Comparison to D2.2

In Tables 15 and 16 we compare the current method (FV-MTL with CCE-LC) with the two approaches reported in D2.2 , i.e. the FT3-ex algorithm (see Section 4.2.3.2 of D2.2) and the DMTL_LC algorithm (see Section 4.2.3.3 of D2.2), referred in the tables as D2.2 method A and D2.2 method B, respectively. The used metrics are MXinfAP (%) for 38 TRECVID-SIN and MAP (%) for 20 PASCAL-VOC2007, 20 PASCAL-VOC2012 and 81 NUS-WIDE concepts, using the ImageNet ResNet-50 as the base network. According to Table 15, the current method outperforms both of the previous methods. At the same time, the replacement of the previously developed two-sided network (DMTL_LC) with a single-side one (FV-MTL with CCE_LC) reduced the processing time and also scaled our method to more concepts (Table 16). Compared to the extension strategy (D2.2 method A) the execution time has been increased (Table 16) but important improvement on the accuracy has been achieved (Table 15). In order to further improve the overall accuracy of our video annotation system we trained the current method using also the GoogLeNet as the base network. The two trained networks (one based on ResNet-50 and the other based on GoogLeNet) were combined in terms of arithmetic mean, which reached a MXinfAP of 35.81% when evaluated on the TRECVID SIN dataset. This is the final configuration that was integrated in our service for video fragmentation and annotation. The above discussed progress in the performance of the concept-based annotation component of the service is illustrated in Fig. 10 below.

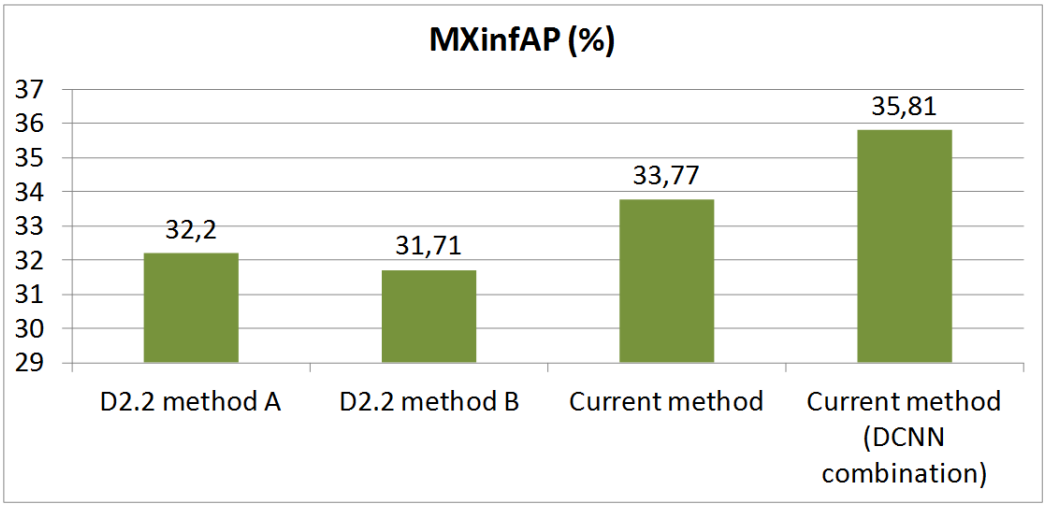

Figure 10: The progress, in terms of MXinfAP (%), regarding the performance of the concept-based annotation component.

Table 16: Mean execution training/testing times in hours.

Category	**Method**	**TRECVID-SIN**	
		training	testing
D2.2 method A	FT3-ex	17.45	2.47
D2.2 method B	DMTL_LC	49.27	6.84
Current	FV-MTL with CCE-LC ($\beta = 10$)	18.15	3.10

5 Future Outlook

This deliverable has presented the final updates and results of the InVID work on story detection, social media retrieval, and video fragmentation and concept-based labeling. We could demonstrate further improvements in the quality of our story detection, significantly in the area of story distinctiveness which was a problem we had identified previously. Also in the relevance of the documents retrieved for stories we were able to show increased precision and have the confidence to further improve the breadth and depth of collected video material for news stories on social media as a result (when we switch the video querying approach to the use of the now even more accurate story labels which we produce). In the video annotation - which involves the fragmentation of the video into visually coherent segments and the concept-based labeling of these fragments - more training data allowed further fine tuning of the developed concept detection algorithm, and a more efficient approach evaluated. The web service for video analysis (i.e. fragmentation and concept-based annotation) has been upgraded to cover the processing requirements of the different components of the InVID platform. Moreover, it is now better load balanced and maintained (updated) automatically. The web application for video fragmentation and reverse keyframe search has been significantly improved according to the feedback collected from internal (from within InVID) and external users (via the corresponding component of the InVID Verification Plugin). Evaluations also looked at a comparison with state of the art tools available to newsrooms and journalists. We found that the tools for which we were able to gain trial access typically did not offer both the story detection and the social media video retrieval feature. Most of them required the user to already decide on a news story and determine by themselves the keywords to search for, whereas InVID offers automatic story detection AND video material collection for each story from social networks. We found our story detection and our social media retrieval features to be comparable to other state of the art tools; indeed the focus on video collection meant that InVID often had more links to video for the selected stories than other platforms used by journalists.

While this is the final InVID deliverable on this work, we can assure that the work is not yet finished. It will continue to be supported in the InVID user tools- the Dashboard and the Verification App - and therefore there will be further improvements as needed. For example, we will switch to using the Semantic Knowledge Base to supporting the keyword detection and disambiguation, which will in turn provide

even clearer results for the story clustering and labelling. We will look at providing metrics in the social media metadata for the 'reach' and 'authoritativeness' (as discussed already in D2.2). Finally, we plan some further fine-tuning of our method for visual concept detection, as well as the automatic identification and removal of blurred and black keyframes that are less valuable for video content representation and annotation, and for keyframe-driven reverse video search.

MODUL Technology and CERTH will continue to support the respectively provided functionalities within as well as after the InVID project, since both the InVID Dashboard and the Verification App form part of future project exploitation efforts.

References

Alsaedi, N., Burnap, P., & Rana, O. (2017). Can we predict a riot? disruptive event detection using twitter. *ACM Transactions on Internet Technology (TOIT)*, *17*(2), 18.

Argyriou, A., Evgeniou, T., & Pontil, M. (2007). Multi-task feature learning. *Advances in Neural Information Processing Systems (NIPS 2007)*.

Argyriou, A., Evgeniou, T., & Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, *73*(3), 243-272.

Baumgartner, M. (2009). Uncovering deterministic causal structures: a boolean approach. *Synthese*, *170*(1), 71-96.

Bishay, M., & Patras, I. (2017). Fusing multilabel deep networks for facial action unit detection. In *Proc. of the 12th ieee int. conf. on automatic face and gesture recognition (fg).*

Blanken, H. M., de Vries, A. P., Blok, H. E., & Feng, L. (2005). *Multimedia retrieval*. NY: Springer.

Cai, X., Nie, F., Cai, W., & Huang, H. (2013). New graph structured sparsity model for multi-label image annotations. In *Proc. of the ieee international conference on computer vision (iccv 2013)* (p. 801-808).

Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., & Zheng, Y.-T. (July 8-10, 2009). Nus-wide: A real-world web image database from national university of singapore. In *Proc. of acm conf. on image and video retrieval (civr 2009).* Santorini, Greece.

Daumé, H., III. (2009). Bayesian multitask learning with latent hierarchies. In *the 25th conf. on uncertainty in artificial intelligence (uai 2009)* (pp. 135–142). Quebec, Canada: AUAI Press.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., ... Adam, H. (2014). Large-scale object classification using label relation graphs. In *Proc. of the 13th europ. conf. on computer vision (eccv 2014)* (pp. 48–64). Zrich, Switzerland: Springer.

Deng, J., Satheesh, S., Berg, A. C., & Li, F. (2011). 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.

Deng, Z., Vahdat, A., Hu, H., & Mori, G. (2015). Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. *CoRR*, *abs/1511.04196*.

Ding, N., Deng, J., Murphy, K. P., & Neven, H. (2015). Probabilistic label relation graphs with ising models. In *Proc. of the 2015 ieee int. conf. on computer vision (iccv 2015)* (pp. 1161–1169). Washington, DC, USA: IEEE.

Everingham, M., Van Gool, L., & et al. (n.d.). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.*

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (n.d.). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.* http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Evgeniou, T., & Pontil, M. (2004). Regularized multi–task learning. In *Proc. of the 10th acm sigkdd int. conf. on knowledge discovery and data mining (kdd 2004)* (pp. 109–117). Seattle, WA.

Hammad, M., & El-Beltagy, S. R. (2017). Towards efficient online topic detection through automated bursty feature detection from arabic twitter streams. *Procedia Computer Science*, *117*, 248–255.

He, K., Zhang, X., Ren, S., & Sun, J. (2016, June). Deep residual learning for image recognition. In *Proc. of the ieee conf. on computer vision and pattern recognition (cvpr)* (p. 770-778). doi: 10.1109/CVPR.2016.90

Hong, R., Wang, M., Gao, Y., Tao, D., Li, X., & Wu, X. (2014, May). Image annotation by multiple-instance learning with discriminative feature mapping and selection. *IEEE Transactions on Cybernetics*, *44*(5), 669-680. doi: 10.1109/TCYB.2013.2265601

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Katragadda, S., Benton, R., & Raghavan, V. (2017). Framework for real-time event detection using multiple social media sources.

Kumar, A., & Daume, H. (2012). Learning task grouping and overlap in multi-task learning. In *Proc. of the 29th acm int. conf. on machine learning (icml 2012)* (pp. 1383–1390). Edinburgh, Scotland.

Lu, Y., Zhang, W., Zhang, K., & Xue, X. (2012). Semantic context learning with large-scale weakly-labeled image set. In *Proc. of the 21st acm int. conf. on information and knowledge management* (pp. 1859–1863). NY, USA: ACM.

Luo, Q., Zhang, S., Huang, T., Gao, W., & Tian, Q. (2014). Superimage: Packing semantic-relevant images for indexing and retrieval. In *Proc. of the int. conf. on multimedia retrieval (icmr 2014)* (pp. 41–48). NY, USA: ACM.

Markatopoulou, F., Mezaris, V., & Patras, I. (2016a). Deep multi-task learning with label correlation constraint for video concept detection. In *Proc. of the int. conf. acm multimedia (acmmm 2016)* (pp. 501–505). Amsterdam, The Netherlands: ACM.

Markatopoulou, F., Mezaris, V., & Patras, I. (2016b, Sept). Online multi-task learning for semantic concept detection in video. In *Proc. of the ieee int. conf. on image processing (icip 2016)* (p. 186-190).

Markatopoulou, F., Mezaris, V., & Patras, I. (2018). Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1-1. doi: 10.1109/TCSVT.2018.2848458

Markatopoulou, F., Mezaris, V., Pittaras, N., & Patras, I. (2015). Local features and a two-layer stacking architecture for semantic concept detection in video. *IEEE Transactions on Emerging Topics for Computing*, *3*, 193-204.

Mele, I., & Crestani, F. (2017). Event detection for heterogeneous news streams. In *International conference on applications of natural language to information systems* (pp. 110–123).

Mousavi, H., Srinivas, U., Monga, V., Suo, Y., Dao, M., & Tran, T. (2014). Multi-task image classification via collaborative, hierarchical spike-and-slab priors. In *the ieee int. conf. on image processing (icip 2014)* (p. 4236-4240). Paris, France.

Nixon, L. J., Zhu, S., Fischer, F., Rafelsberger, W., Göbel, M., & Scharl, A. (2017). Video retrieval for multimedia verification of breaking news on social networks. In *Proceedings of the first international workshop on multimedia verification* (pp. 13–21). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/3132384.3132386` doi: 10.1145/3132384.3132386

Obozinski, G., & Taskar, B. (2006). Multi-task feature selection. In *Proc. of the 23rd int. conf. on machine learning (icml 2006). workshop of structural knowledge transfer for machine learning.* Pittsburgh, Pennsylvania.

Ouyang, W., Chu, X., & Wang, X. (2014). Multi-source deep learning for human pose estimation. In *Proc. of the the ieee conf. on computer vision and pattern recognition (CVPR 2014)* (p. 2337-2344). Columbus, Ohio: IEEE.

Over, P., & et al. (2013). TRECVID 2013 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Trecvid 2013.*

Pittaras, N., Markatopoulou, F., Mezaris, V., & Patras, I. (2017). Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks. In *Proc. of the 23rd Int. Conf. on MultiMedia Modeling (MMM 2017)* (pp. 102–114). Reykjavik, Iceland: Springer.

Qi et al., G.-J. (2007). Correlative multi-label video annotation. In *Proc. of the 15th int. conf. on multimedia* (pp. 17–26). NY: ACM.

Qin, Y., Zhang, Y., Zhang, M., & Zheng, D. (2018). Frame-based representation for event detection on twitter. *IEICE TRANSACTIONS on Information and Systems*, *101*(4), 1180–1188.

Russakovsky, O., Deng, J., & et al., H. S. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision (IJCV 2015)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Schwing, A. G., & Urtasun, R. (2015). Fully connected deep structured networks. *CoRR*, *abs/1503.02351*.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, *abs/1409.1556*. Retrieved from `http://arxiv.org/abs/1409.1556`

Smith, J., Naphade, M., & Natsev, A. (2003). Multimedia semantic indexing using model vectors. In *Proc. of the int. conf. on multimedia and expo (icme 2003)* (pp. 445–448). NY: IEEE. doi: 10.1109/ICME.2003.1221649

Srijith, P., Hepple, M., Bontcheva, K., & Preotiuc-Pietro, D. (2017). Sub-story detection in twitter with hierarchical dirichlet processes. *Information Processing & Management*, *53*(4), 989–1003.

Sucar, L. E., Bielza, C., Morales, E. F., Hernandez-Leal, P., Zaragoza, J. H., & Larraaga, P. (2014). Multi-label classification with bayesiannetwork-based chain classifiers. *Pattern Recognition Letters*, *41*, 14 - 22.

Sun, G., Chen, Y., Liu, X., & Wu, E. (2015, Sept). Adaptive multi-task learning for fine-grained categorization. In *Proc. of the ieee int. conf. on image processing (icip 2015)* (p. 996-1000).

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin markov networks. In *Proc. of the 16th int. conf. on neural information processing systems (nips 2003).* MIT Press.

Tonon, A., Cudré-Mauroux, P., Blarer, A., Lenders, V., & Motik, B. (2017). Armatweet: Detecting events by semantic tweet analysis. In *European semantic web conference* (pp. 138–153).

Vakulenko, S., Nixon, L. J. B., & Lupu, M. (2017). Character-based neural embeddings for tweet clustering. *CoRR*, *abs/1703.05123*. Retrieved from `http://arxiv.org/abs/1703.05123`

van de Sande, K. E. A., Uijlings, J. R. R., Gevers, T., & Smeulders, A. W. M. (2011, Nov). Segmentation as selective search for object recognition. In *Int. conf. on computer vision* (p. 1879-1886).

Wang, H., Huang, H., & Ding, C. (2009, Sept). Image annotation using multi-label correlated green's function. In *Proc. of the ieee conf. on computer vision and pattern recognition (cvpr 2009)* (p. 2029-2034).

Wang, H., Huang, H., & Ding, C. (2011, June). Image annotation using bi-relational graph of images and semantic labels. In *Proc. of the ieee conf. on computer vision and pattern recognition (cvpr 2011)* (p. 793-800).

Wang, M., Hua, X. S., Hong, R., Tang, J., Qi, G. J., & Song, Y. (2009, May). Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*(5), 733-746. doi: 10.1109/TCSVT.2009.2017400

Wang, M., Luo, C., Hong, R., Tang, J., & Feng, J. (2016, Dec). Beyond Object Proposals: Random Crop Pooling for Multi-Label Image Recognition. *IEEE Transactions on Image Processing*, *25*(12), 5678-5688.

Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., ... Yan, S. (2016, Sept). Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(9), 1901-1907. doi: 10.1109/TPAMI.2015.2491929

Weng, M.-F., & Chuang, Y.-Y. (2012). Cross-Domain Multicue Fusion for Concept-Based Video Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(10), 1927–1941.

Yang, Y., & Hospedales, T. (2015). A unified perspective on multi-domain and multi-task learning. In *Proc. of the int. conf. on learning representations (iclr 2015).* San Diego, California.

Yang, Y., & Hospedales, T. M. (2017). Deep multi-task representation learning: A tensor factorisation approach. In *Proc. of the international conference on learning representations (iclr).*

Yang, Y., Wu, F., Nie, F., Shen, H. T., Zhuang, Y., & Hauptmann, A. G. (2012, March). Web and personal image annotation by mining label correlation with relaxed visual graph embedding. *IEEE Transactions on Image Processing*, *21*(3), 1339-1351.

Yilmaz, E., Kanoulas, E., & Aslam, J. A. (2008). A simple and efficient sampling method for estimating ap and ndcg. In *Proc. of the 31st acm int. conf. on research and development in information retrieval (sigir 2008)* (pp. 603–610). Singapore.

Yılmaz, Y., & Hero, A. O. (2018). Multimodal event detection in twitter hashtag networks. *Journal of Signal Processing Systems*, *90*(2), 185–200.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NIPS 2014)*, 3320–3328.

Zhang, M.-L., & Zhang, K. (2010). Multi-label learning by exploiting label dependency. In *Proc. of the 16th acm sigkdd int. conf. on knowledge discovery and data mining (kdd 2010)* (pp. 999–1008). NY, USA: ACM.

Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2014). Facial landmark detection by deep multi-task learning. In *Proc. of the 13th europ. conf. on computer vision (eccv 2014)* (pp. 94–108). Zurich, Switzerland: Springer.

Zhao, X., Li, X., & Zhang, Z. (2015, Oct). Joint structural learning to rank with deep linear feature learning. *IEEE Transactions on Knowledge and Data Engineering*, *27*(10), 2756-2769.

Zheng, S., Jayasumana, S., & et al. (2015). Conditional random fields as recurrent neural networks. In *Proc. of the int. conf. on computer vision (iccv 2015).*

Zhou, J., Chen, J., & Ye, J. (2011). Clustered multi-task learning via alternating structure optimization. *Advances in Neural Information Processing Systems (NIPS 2011)*.