# Enhancing and Re-Purposing TV Content for Trans-Vector Engagement

Deliverable 1.1 (M10)

# Data Ingestion, Analysis and Annotation

Version 1.0

This document was produced in the context of the ReTV project supported by the European Commission under the H2020-ICT-2016-2017 Information & Communication Technologies Call Grant Agreement No 780656

## DOCUMENT INFORMATION

| | |
|---|---|
| **Delivery Type** | Report |
| **Deliverable Number** | 1.1 |
| **Deliverable Title** | Data Ingestion, Analysis and Annotation |
| **Due Date** | M10 |
| **Submission Date** | October 31, 2018 |
| **Work Package** | WP1 |
| **Partners** | CERTH, MODUL Technology |
| **Author(s)** | Vasileios Mezaris (CERTH), Kostas Apostolidis (CERTH), Lyndon Nixon (MODUL Technology) |
| **Reviewer(s)** | Willy Lamm, Martin Gordon (RBB) |
| **Keywords** | Data Ingestion, TV Program Annotation, TV Program Analysis, Social Media Retrieval, Web Retrieval, Video Analysis, Concept Detection, Brand Detection |
| **Dissemination Level** | PU |
| **Project Coordinator** | Vrije Universiteit Amsterdam<br>De Boelelaan 1081 , 1081 HV,<br>Amsterdam, The Netherlands |
| **Contact Details** | Coordinator: Prof Lora Aroyo (lora.aroyo@vu.nl)<br>R&D Manager: Dr Lyndon Nixon (lyndon.nixon@modultech.eu)<br>Innovation Manager: Bea Knecht (bea@zattoo.com) |

## Revisions

| Version | Date | Author | Changes |
|---|---|---|---|
| 0.1 | 27/8/18 | V. Mezaris, K. Apostolidis | Created template and ToC |
| 0.2 | 11/9/18 | V. Mezaris, K. Apostolidis | First drafts Chapters 5 and 6 |
| 0.3 | 18/9/18 | L. Nixon | First drafts Chapters 3 and 4 |
| 0.4 | 27/9/18 | L. Nixon | Update Chapter 3 |
| 0.5 | 1/10/18 | L. Nixon | Update Chapter 4 |
| 0.6 | 8/10/18 | V. Mezaris, K. Apostolidis | Wrote Introduction and Conclusions sections |
| 0.7 | 9/10/18 | V. Mezaris, K. Apostolidis | Updated Chapters 5 and 6 |
| 0.8 | 10/10/18 | L. Nixon | Correcting text (minor improvements) |
| 0.9 | 11/10/18 | V. Mezaris, K. Apostolidis | Further corrections and send to QA |
| 0.10 | 15/10/18 | M. Gordon (QA) | Review/minor edit |
| 0.11 | 19/10/18 | W. Lamm (QA) | QA review (ReTV internal) |
| 0.12 | 23/10/18 | K. Apostolidis, L. Nixon | Updates from QA review |
| 0.13 | 27/10/18 | A. Scharl | Final check by R&D Lead |
| 0.14 | 29/10/18 | L. Nixon | Final check by Project Lead |

## Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

This deliverable reflects only the authors' views and the European Union is not liable for any use that might be made of information contained therein.

## TABLE OF CONTENTS

# EXECUTIVE SUMMARY

This deliverable outlines the identified media vectors to be supported by the ReTV Trans Vector Platform (TVP) as well as the implementation plan for acquiring data from each vector, based on existing APIs and requirements. It describes the initial deployment of (i) concept-based video abstraction technologies and (ii) a brand detection solution, both including a benchmark evaluation. The deliverable also reports on the chosen TVP annotation model, outlines a first deployment of the metadata extraction from online sources according to that annotation model, and the setting up of a *Knowledge Graph* to capture additional knowledge and the relations between entities referenced by TVP annotations.

## ABBREVIATIONS LIST

| Abbreviation | Description |
|---|---|
| API | Application Programming Interface: a set of functions and procedures that allow the creation of applications which access the features or data of an application or other service. |
| DCNN | Deep Convolutional Neural Network: a type of artificial neural network. |
| DCT | Discrete Cosine Transform: a transformation that expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. |
| EPG | Electronic Program Guides: menu-based systems that provide users of television with continuously updated menus displaying broadcast programming or scheduling information for current and upcoming programming. |
| HTTP POST/GET | Types of method in the Hypertext Transfer Protocol (HTTP). The HTTP POST method is used to send data to a server to create/update a resource. The HTTP GET method is used to request data from a specified resource. |
| IPTV | Internet Protocol Television: is the delivery of television content over Internet Protocol (IP) networks. |
| JSON | JavaScript Object Notation: a data-interchange format. |
| LSTM | Long Short Term Memory networks: a type of recurrent neural network. |
| MTL | Multi-task learning: a field of machine learning in which multiple learning tasks are solved at the same time, exploiting commonalities and differences across tasks. |
| OTT | Over The Top: content providers that distribute streaming media as a standalone product directly to viewers over the Internet, bypassing telecommunications that traditionally act as a  distributor of such content. |
| RDF | Resource Description Framework: a method for conceptual description or modeling of information that is implemented in web resources. |
| REST | Representational State Transfer: an architectural style that defines a set of constraints to be used for creating web services. |
| RNN | Recurrent Neural Network: a type of an artificial neural network. |
| TVoD | Transactional Video on Demand: a distribution method by which customers pay for each individual piece of video on demand content. |
| URL | Uniform Resource Locator: a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. |

# 1. Introduction

This deliverable reports on the work done during the first ten months of the ReTV project in all tasks of WP1, namely tasks T1.1 (Content Collection Across Vectors), T1.2 (Concept-Based Video Abstractions), T1.3 (Brand Detection in Video) and T1.4 (Annotation and Knowledge Graph Alignment).

Firstly, the annotation model was set-up. Specifically, we ascertain what knowledge is required about TV programming to be sufficient for subsequent analysis/enrichment, and we need to establish how this knowledge will be represented and where this knowledge can be acquired. For the semantic representation of annotation targets (the concepts an annotation refers to), a Knowledge Graph has been created. Technologies for performing temporal segmentation and visual analysis at video fragment level were deployed. These methods will label input videos with concept-based representations. Additionally, a baseline method for brand recognition at video fragment level and at video level was developed. These components were evaluated on various standard benchmark datasets as well as on ReTV videos to demonstrate their applicability on such content. The software components are provided via their REST APIs, which are also documented in this deliverable.

The structure of the document is as follows: The Content Collection Across Vectors is described in Section 2. The Annotation and Knowledge Graph Alignment is described in Section 3. In Section 4, analysis components for Video Fragmentation and Concept-based Annotation and related problems are addressed, we review related work in the literature and analyze the adopted methods. In Section 5, we describe related work and the adopted method regarding Brand Detection and provide documentation of implemented APIs for all video analysis components. Finally, we conclude this deliverable in Section 6 with a brief summary and overview of future outlook.

## 2. Content Collection Across Vectors

The content collection task in ReTV relates to the collection of TV programming metadata and online content relating to TV programming in order to support subsequent data analysis, powering services for content re-purposing, recommendation and scheduling. Collection is set up across "vectors", meaning the distinct channels on which content is published: broadcast TV (also rebroadcast on IPTV or OTT TV), web media streaming (live stream, Catch-up or TVoD, archive), website content (articles, media) or social networks.

### 2.1 Problem Statement

In ReTV, data ingestion refers to creating a knowledge base of TV programming and their descriptions, in order to be enriched with content and audience metrics by time and vector (WP2) and used in content re-purposing and recommendation (WP3). The main problem to be addressed is to define what knowledge about the TV programming is required to be suffice for subsequent enrichment and analysis, how this knowledge is to be represented (annotation model) and where this knowledge can be acquired. In the state of the art survey (2.2) we briefly reflect on past work which has attempted something similar, and the challenges encountered and solutions offered. In Section 2.3 we identify what knowledge we require. In the remainder of this chapter we describe the sources of the knowledge. Chapter 3 will address how the acquired knowledge is represented and from where we acquire the knowledge that is missing from the data we receive from our acquisition sources.

### 2.2 State-of-the-Art Survey

The storage of references to content of TV programmes (or any digital media asset) and/or its metadata in Media Asset Management Systems (MAMS)/Content Management Systems (CMS) is a standard part of broadcaster infrastructure. The used infrastructure can vary from organization to organization, ranging from smaller organizations storing media files on a server including only their technical metadata, to large media companies using expensive and complex MAMS (though not necessarily with any standardized media metadata model, as all media management is seen as a purely internal matter). The concept of ingestion of data from multiple vectors and its combination via analytics and services using the TVP is an innovation of the ReTV project; there is nothing comparable to the TVP on the market right now. For example, for organizations ingesting EPG data for TV broadcasts, there may be a mapping from the received format to their internal format (e.g. for display in their own EPG application or website) but not an augmentation of the data with content annotation, as is done in ReTV.

In the ‘semantic multimedia’ research field, augmentation of Electronic Programme Guide (EPG) data for re-use in other services (e.g. recommendation) has been a topic for many years - the NoTube project (Aroyo, 2009) (Aroyo, 2011), which involved a number of people now part of the ReTV consortium, was perhaps the first to "prepare TV for the future Internet" which included enrichment of EPG data with entities from Linked Data sources. This has inspired other work in the research community e.g. (Macedo, 2014) but, in subsequent years, a comprehensive initiative to actually bring the approach into the broadcasters/media organisations own workflows has been missing. The TVP is a significant opportunity to allow organisations to make use of such EPG enrichments, mainly because (i) the semantic annotation is in the "back end" where the organisation does not need to directly handle it and (ii) there will be "front end" applications which make direct use of those annotations, which are the main drivers for the technology uptake.

Such a "knowledge base" for TV - to be used by the TVP - is a USP of ReTV. There is a growing interest inside some of the larger broadcasters to augment their internal TV metadata for re-use - the BBC Programmes Linked Data being the most public example to date (Kobilarov, 2009) - but the authors have also spoken to representatives of French and German broadcasters who have mentioned internal projects, e.g. to use speech-to-text APIs to transcribe automatically TV programmes and use those transcriptions in labelling the program with a description of its content. Of course, each of these projects then creates a new data silo which is inside a media organisation and only applies to the media assets of that organisation - whereas the TVP collects media knowledge across broadcasters (and other providers).

Platforms for monitoring the web and social media for mentions of specific terms have been in existence for some time now; in many industries the goals of "brand intelligence" have led to the need for software to provide feedback on when, where and how a brand is mentioned online. In digital marketing, this information can be used to analyse the success of online marketing activities, as well as to gain insights generally into the "brand reputation" among different online communities. Often referred to as "web and social media intelligence", the monitoring of online content has also become relevant and important to media organisations, who want to know how their content is reacted to by the audience. Typical cases are analysing tweets using the hashtag promoted by a TV broadcast. In the TVP, there will be a connection between the "knowledge base" of TV and the metrics generated around TV programming from web and social media monitoring (refer to deliverable D2.1, Section 3), whereas current solutions for monitoring would operate only within their ‘vector’ (Twitter analytics, Facebook analytics etc.) rather than tying online "intelligence" about media content to other vectors such as the TV broadcast itself (audience figures) or media streaming (viewing rate).

As can be seen from the brief state of the art overview, a key added value foreseen for the TVP is the cross-vector aspect, which begins here with the collection of data from multiple vectors, stored in the TVP using a common annotation model (next chapter) which allows subsequent cross-vector analyses and services. The state of the art in TV annotation is considered in the following chapter.

## 2.3 ReTV Approach

The principle type of content item will be a TVProgram or a TVAd. However, we also want to collect related content which is associated to a TVProgram, such as web pages (synopsis, review) or social media posts. When a TVProgram or TVAd is re-purposed, we also have a new TVProgram or TVAd item, e.g. for a video trailer. Rather than to try to collect data for every single TVProgram or TVAd being created, due to the scale involved and the need to test initially the data collection approach at a smaller scale, we need a seed list of programs and ads to begin with. This can be defined:

- Statically, via a program/ad list sent by a TV content source, ideally using their internal IDs for the programs/ads so that the TV source’s content database can be queried;
- Dynamically, via access to EPG data for a set of TV channels, extracting program IDs/titles and aligning them to other data source references to those programs.

Based on a finite number of TV program (titles) and TV ad (brands/products), we can collect related content by setting up web crawlers or social media monitoring configured to find documents which have an association with the program or ad.

This data collection can be categorized as either on-demand or continual, where:

- On-demand means the collection process is only executed when explicitly requested through a push mechanism from the TV content source, e.g. a new static list of TV programs is sent;

- Continual means the collection process is executed on a periodic basis without the TV content source taking any action (here it is particularly the responsibility of the source that its systems are available), e.g. EPG data could be downloaded on a daily basis.

The data ingestion pipeline is pre-configured for each data collection task across all data sources (one or more per vector), with a definition of when data is collected as well as how data is collected. The form of access may vary by vector, data source and purpose:

- By API
  - Public APIs such as social network APIs[1]
    - Monitoring new content on social network channels
    - Monitoring mentions of associations to TVPrograms or TVAds in social media
  - Private APIs such as TV content sources CMS or MAMS
    - Querying for metadata for TV programs or ads
      - Getting links to binary video for analysis tasks
    - Querying for EPG data
- By web crawling
  - Websites
    - Lists of URLs to download copies of webpage content, e.g. program synopses[2]
    - Lists of term queries to perform web search and find webpages of relevance, e.g. find reviews of TVPrograms
- By direct downloading
  - Some sources may only make data available as downloadable archives

The next section clarifies the implementation of the data ingestion pipeline in the ReTV project.

## 2.4 Implementation Details and Use

We need to set up the data ingestion pipeline with the various data source configurations (the separate components are called data mirrors). There is a general three-step procedure for performing content collection in ReTV:

- We create a seed list of TV programs and TV ads which we wish to consider in the data collection:
  - The partner Genistat can share EPG data with us which informs us which TV program is broadcast on which TV channel at what time;
  - The EPG data does not indicate when TV ads are broadcast, however we can classify TV channels by when they run TV advertising (if at all) and a first naive assumption is that ads are shown around the program boundaries;

[1] External access to social network data is always an issue in social media monitoring, with Instagram recently being more restrictive about API access and Facebook not supporting text-based search over all public content. There are also free API limits in place, although in many cases they are generous enough for acquiring a good sample of content.

[2] Regarding our policy for Web and Social Media crawling, please refer to https://www.modultech.eu/privacy-policy/

  - Each EPG entry is represented by a new document in the content store, i.e. each document created in the 'TV content' data ingestion represents the broadcast of one TV program on one TV channel at one time.
- We complete documents according to our annotation model to represent each piece of TV content where the TV program/ad is in our seed list:
  - The EPG data already contains additional metadata we can include in the document, e.g. title and description;
  - Additional textual metadata may be acquired to enrich the description of the content further, e.g. a transcript of the program;
  - Text analysis can be applied to identify e.g. keywords and entities of interest in the program;
  - Video analysis can be used in video fragmentation, visual concept labeling and brand detection (see Chapters 4 and 5).
- For the TV program/ad in the seed list, we use other data mirrors to collect content related to the TV program brand (i.e. TV series title) or the advertised product/service brand.
  - Crawling websites that belong to the broadcaster or otherwise associated with the TV program;
  - Querying social networks for content of channels belonging to the broadcaster or otherwise associated with the TV program;
  - Crawling other websites (news or media) for content associated with the TV program;
  - Querying social networks for content of user postings mentioning the TV program.

The data mirrors repeat the data collection procedure at regular intervals to detect new content. We clarify here the content being collected for each of the ReTV content partners:

### 2.4.1 RBB Data Collection Procedure

We seed the data collection for RBB initially with 42 selected TV programmes. For each programme, we collect broadcast times and initial metadata from the EPG. We also find associated content with video for those TV programmes on RBB's YouTube (single channel), Facebook (16 Pages) and 11 Twitter channels.

We also monitor social media for references to the RBB TV programming (Facebook (public content), Twitter) and record that content to create Content Metrics for each programme. For this, we set up a termlist with (unambiguous) TV programme titles and, to further collect relevant content, added 17 terms of reference to the broadcaster (incl. hashtag terms, such as 'rbb', rbbsport' or the slogan 'blossnichtlangweilen') as well as 34 terms of reference to personalities on RBB (restricted to those whose names would be relatively unambiguous and whose primary activity is on RBB, e.g. Eva-Maria Lemke who is a moderator of the RBB news show Abendschau).

### 2.4.2 Zattoo Data Collection Procedure

We initially seed the data collection for Zattoo with 37 TV programmes from 17 different broadcasters, choosing the most watched programming in Switzerland and Germany in the French and German languages. From this initial list, we collect TV programme broadcast information and initial metadata from Genistat's EPG API.

For the TV programmes matching the seed list, we extend the program metadata through text and video analysis. In this case, binary video recordings of the broadcast TV programme is available via Genistat's video database, allowing the video analysis for fragmentation and concept detection.

We also monitor social media for references to those TV programmes (Facebook (public content), Twitter) and record that content to create Content Metrics for each programme. Only 26 out of the 37 TV programme titles are considered unambiguous enough for social media monitoring. In the termlist we added 10 terms related to well-known characters or established hashtags for some of the programming; however we consider that in most cases online content refer to the TV programme itself (by title) and that content referencing TV characters is more often unrelated to discussions about the programme as a whole (e.g. we found characters being quoted on Twitter or being part of memes).

To ensure the collection of a wider range of content, we will also ingest content related to any of the 17 broadcasters. RBB in their use case already mentioned the relevance of monitoring their competitors (i.e. the private German broadcasters) as well as comparing RBB's performance to other public German channels. We set up a termlist with 22 terms for those broadcasters (a slightly higher number as some can be referenced in more than one way, e.g. 'kabeleins' or 'kabel1').

### 2.4.3 Sound and Vision Data Collection Procedure

We will begin data collection using the YouTube channel of VPROBacklight. A second channel has also been suggested: VPROinternational. Both provide episodes of VPRO-produced programming of different types - four series in particular where originally chosen: The Westerners, Light on the North, O'Hanlon's Heroes and The Mind of the Universe. All videos are available with subtitles.

Finally, we also monitor social media for references to that content, so that we can create an aggregated Content Metric for each video. Only 'O'Hanlon's Heroes' has a series title unambiguous enough to monitor social media for content. For the rest, we set up a termlist with 6 terms of reference to VPRO content, including hashtags used.

## 2.5 Results

We started EPG data collection on 17 September 2018 and social media collection on 18 September 2018. Webpage collection will start later as we currently consolidate data about websites across languages and countries, defining the split between news articles and TV-related content (many broadcaster websites contain both). The initial data collection is slightly smaller than the planned implementation described above, in order to more clearly test the functioning of the entire data ingestion pipeline.

We are ingesting the EPG data daily, initially with schedules for the next 24 hours for ten TV channels.

For web and social media collection we created a shortlist of 24 series which are currently (Sept 2018) on air in Zattoo channels (10 distinct TV channels) as well as 4 series which are broadcast on YouTube by web TV channel VPRO. Of the 24 Zattoo TV series:

- 15 on public, 9 on private channels (6 series from RBB);
- 15 different genres;
- 16 from German, 6 from Swiss, 2 from Austrian channels;
- 22 German and 2 French language;
- 19 are during prime time (6-9pm) and 17 compete with one another for weekday evening viewing.

From this list, we identified and monitor 11 Facebook pages, 6 Twitter accounts and 5 YouTube channels. We also track 50 whitelist terms (i.e. we collect social media content which references a term in the whitelist). The first 6 days of data collection give us an initial idea of the scale of documents: 2 217 Facebook posts, 3 289 tweets and 206 YouTube videos.

In the next phase, we will sample documents being created by the data ingestion pipeline to test for relevance - modifying in particular the term whitelist if necessary. We will also expand the initial data collection to cover all TV series of the initial seed list as described in section 2.4.

# 3. Annotation and Knowledge Graph Alignment

As indicated in the previous chapter, when collecting TV broadcast, web and social media data, we describe each data item in a document with a metadata description of the subject of the data (i.e. a TV program broadcast, a web page or a social media post). Our goal of annotating each data item is to allow subsequent search and browsing of the data via API or Dashboard, including structural metadata (e.g. video duration and fragmentation), topical/conceptual metadata and media lifecycle metadata (e.g. re-use of previous content). On top of annotation-enabled search and browsing the ReTV services of prediction, re-purposing and recommendation are enabled. We present our annotation model and how we use it in combination with a Knowledge Graph for the conceptual annotation.

## 3.1 Problem Statement

For each content item (a TV program broadcast, a webpage or a social media post), the ReTV annotations should meet the following requirements:

- Every content item has an unique ID
- Every content item has a content description, which provides a structured and semantic model for the concepts and topics of the content item.
    - The semantic model will use Linked Data[3] for disambiguated reference to concepts
    - The structural model will use Media Fragment URIs[4] for spatial or temporal fragmentations
    - The visual-conceptual model, based on multimedia analysis, will use an agreed controlled vocabulary for reference to visual concepts
    - All other property values where appropriate will use controlled vocabularies / taxonomies / ontologies instead of simple data types (like strings) e.g. TVProgram Genre
- Every TVProgram and TVAd will be associated within the annotation model with each and every Publication event of that TV content by time and vector
- Every TVProgram and TVAd will be associated within the annotation model with every instance of other related content (webpage, social media) that references the TVProgram or TVAd

## 3.2 State-of-the-Art Survey

As a grounding in the state of the art, we note the LinkedTV ontology (LinkedTV Deliverable 2.2)[5], which similarly addressed the semantic representation of a TV program and its contents, automatically generated as a result of semantic and structural analyses (see Fig. 1). For our conceptual model, we will align the principles of this ontology with the "realities" of the webLyzard document model - the webLyzard platform, already using this document model, is the basis for the TVP development. As such, we also consider the extensions already made by MODUL Technology (see Table 1) to the webLyzard document model in the InVID project (InVID Deliverable 2.2[6]), due to the ingestion of online video documents (from YouTube and other platforms). Resulting from this, we define a complete conceptual model for ReTV after analysis of the

[3] http://linkeddata.org
[4] https://www.w3.org/TR/media-frags/
[5] http://semantics.eurecom.fr/linkedtv/
https://www.slideshare.net/linkedtv/d22-specication-of-lightweight-metadata-models-for-multimedia-annotation
[6] https://www.invid-project.eu/wp-content/uploads/2016/01/InVID_D2.2_v1.0.pdf

ReTV use cases (and hence the data needed to support the subsequent workflow of the Trans Vector Platform), align it to the current webLyzard document model and identify the ReTV extensions needed when performing data collection from the TV, web and social media data sources.

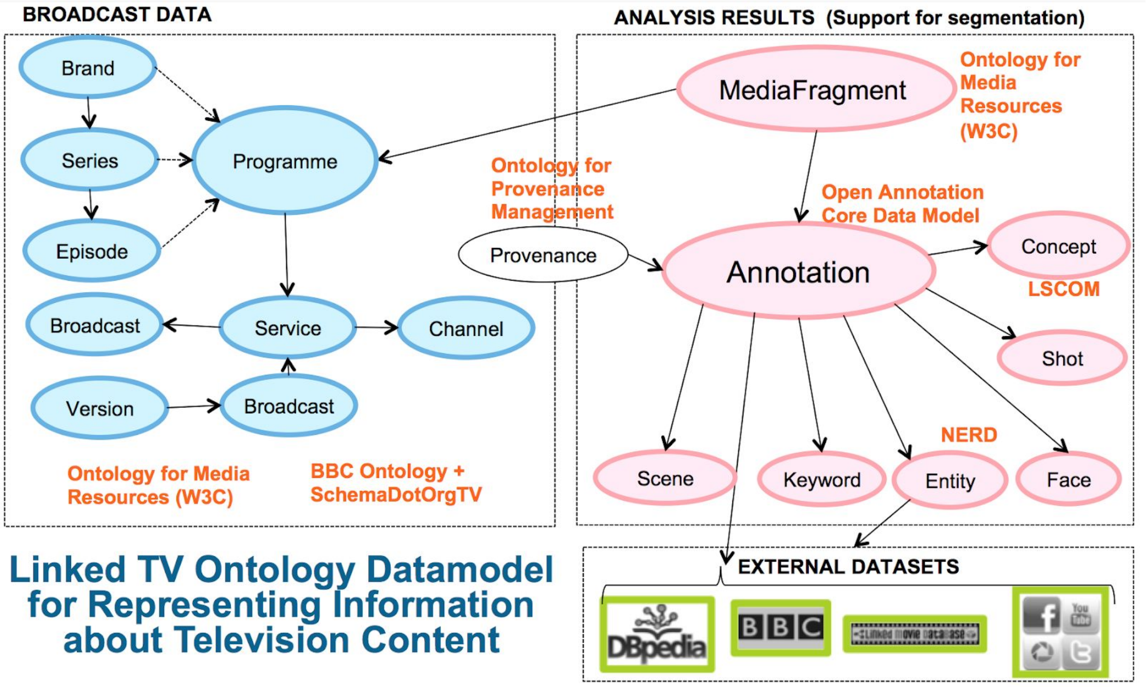
BROADCAST DATA
Brand
Series
Episode
Programme
Broadcast
Service
Channel
Version
Broadcast
Ontology for Media Resources (W3C)
BBC Ontology + SchemaDotOrgTV
ANALYSIS RESULTS (Support for segmentation)
MediaFragment
Ontology for Media Resources (W3C)
Ontology for Provenance Management
Provenance
Open Annotation Core Data Model
Annotation
Concept
LSCOM
Shot
NERD
Scene
Keyword
Entity
Face
EXTERNAL DATASETS
DBpedia
BBC
Linked Movie Database
Linked TV Ontology Datamodel for Representing Information about Television Content

Figure 1: LinkedTV Ontology Data Model

| Document attribute | Description |
|---|---|
| media_type | key attribute to filter media content controlled list |
| duration | duration of the video in seconds |
| media_license | empty or a Creative Commons URL |
| thumbnail | link to a thumbnail image for the video |
| keyword | list of keywords |
| viewcount | number of video views |
| comments | number of comments |
| likes | user rating for the video, number of likes |
| user_id | platform specific identifier of the user |
| user_name | user name |
| media_url | url link to the actual image or video file |
| media_recordingLocation | Geolocation where the media was created |
| media_recordingDate | Date when the media was created |
| title | title of the document |

| | |
|---|---|
| extracted_content (text) | text of the document |
| url | link to the document |
| valid_from (date) | created |
| last_modified | timestamp of the last changes to the document |

Table 1:webLyzard document model for video content in InVID project

### 3.3 ReTV Approach

Metadata for each content item will be retrieved in the data collection phase from the data source, however the extent of available metadata will vary across sources. Thus we will need additional metadata creation steps to complete the metadata within the annotation model.

- We will use the RECOGNYZE Webservice (of MODUL Technology) for Named Entity Recognition and Linking (NE/NEL) over textual metadata to extract (semantic) concepts of relevance to the content description:
  - We are extending this in ReTV to include, besides the entity types of Person, Organisation and Location, also to Works (such as references to other TV programs) and Events, which will be represented as entity instances in our own Semantic Knowledge Base (SKB).
- We will use video analysis algorithms to determine the structural and visual-conceptual model of the content items which are represented in audiovisual form (i.e. the TVProgram):
  - Visual concept detection in ReTV is being extended to brand logos in advertising
  - The annotation model will also represent the outputs of analysis algorithms such as visual content classifiers
- We will map the various type schemas for property values from the sources' content metadata to the controlled vocabularies/schemas agreed on internally in ReTV (supported by the metadata interoperability task, see Deliverable D3.1)

We designed a conceptual model which should cover all of the above features in an annotation. We defined the foreseen properties of a document representing a TV program broadcast (i.e. an EPG entry):

- type
  - Genre
  - Season / Episode
  - Brand (for TVPrograms with equivalent entities in our Knowledge Graph, we use the entity URI)
- id (internal)
  - see_also (URLs of web content relevant to the TVProgram)
- title
  - Description
    - Entities (persons, organisations, locations, events, works)
  - Thumbnail URL (optional)
  - Language of program
  - Media Format
    - (video) duration (in seconds)
  - Fragments (scenes, shots, subshots)

        - Keyframes
        - Concepts (visual descriptors)
        - Logos (for brands)
    - Sentences (text from transcripts, subtitles or speech2text)
        - Entities
- publications
    - PublicationEvent
        - Media URL (optional DVB or HTTP URL for the content)
        - Service (organisation at the last mile, e.g. DVB-based Broadcast TV, OTT provider, Web platform....)
            - name
            - URL
        - Channel (the TV station or other "content provider" using the service)
            - name
            - URL
        - Date-time
            - start
            - end (optional for time-limited publication like broadcast or TVoD)
- version-of (if re-purposed from another TVProgram)

The other main class of document is the RelatedContent (i.e. webpages or social media posts): both the TVRelatedContent and the UserRelatedContent. The TVRelatedContent is related content to a TVProgram produced and controlled by the content owner (e.g. social media posts in the channel of the TV programme). The UserRelatedContent is related content to a TV program created by users (e.g. social media posts on any public channel mentioning a TV programme).

The foreseen properties of these documents:

- type
- id
- title
    - Description
    - Entities (persons, organisations, locations, events, works)
    - URL
    - Media (urls of images or videos of relevance which are part of the webpage/social media post)
- Publications
    - PublicationEvent
        - Service (the content delivery network: website domain name, social media platform)
            - name
            - URL
        - Channel (the content originator: e.g. the TV station identifier, the social media channel (user id))
            - name
            - URL
        - Date-time (of publication)

- responds-to (another RelatedContent, e.g. a reply to a tweet already ingested)

We also take care how to align our model with other ontologies and metadata vocabularies used in the web and TV domains, which can help support interoperability and the ingestion of external metadata. By re-using properties and values from well-known and well-defined specifications, we facilitate the uptake of ReTV services using the same data model for input/output; in the other cases we will define how our annotation model may be mapped into other standards/specifications (cf. Deliverable D3.1).

We serialise below the conceptual model using RDF triples. RDF, as part of the Semantic Web stack, is designed for knowledge representation and thus can conceptualize some domain of discourse in a machine-processable manner (RDF Schema). It also has vocabulary interoperability built into its semantic model, either through re-use of properties and values from other schema or vocabularies or through statements of semantic relationships between them (e.g. owl:sameAs for equivalence). Some namespaces - the term for one distinct set of classes and individuals defined as a single RDF Schema and vocabulary - are re-used in the below RDF serialisation:

| | |
|---|---|
| ReTV | (we assume this to be the default namespace for ReTV specific vocabulary) |
| genre | http://www.ebu.ch/metadata/ontologies/skos/ebu_ContentGenreCS.rdf# |
| ma | http://www.w3.org/ns/ma-ont# |
| po | http://purl.org/ontology/po/ |
| dc | http://purl.org/dc/elements/1.1/ |
| tl | http://purl.org/NET/c4dm/timeline.owl# |

**RDF serialisation of one TVProgram and one RelatedContent document:**

```
:_0     is     :TVProgram
        ma:genre   genre:_3.4
        po:brand   http://dbpedia.org/resource/Breaking_Bad
        po:season  "3"
        po:episode "10"
        rdfs:see_also http://dbpedia.org/resource/Fly_(Breaking_Bad)
        ma:language "de"
        ma:title  "Fly"
        ma:description "Walt, suffering from insomnia, stares up at his smoke detector's flashing light while
trying to get back to sleep. Later, he arrives with Jesse at the superlab, where they begin making another
batch of meth. At the end of the day, Walt calculates that their yield, while above what they are required to
produce, falls short of what he expects. Jesse suggests it may be from other losses from spillage, but Walt
insists there is another reason…"
        ma:keyword      http://dbpedia.org/resource/Walter_White_(Breaking_Bad)
        ma:keyword      http://dbpedia.org/resource/Insomnia
        ma:keyword      http://dbpedia.org/resource/Jesse_Pinkman
        ma:keyword      http://dbpedia.org/resource/Methamphetamine
        ma:duration   "2820"
        :has_fragment    :_0shot1
        :has_sentence   :_0text1
        :has_publication  :_0pub1

:_0shot1   is   :TVFragment
          :temporalStart  "2152"
          :temporalEnd "2208"
          :has_keyframe "ftp://server/0shot1.jpg"
          :concept   http://trecvid.certh.gr/C008
          :logo  http://logos.certh.gr/BMW

:_0text1    is   :TVSentence
          :temporalStart  "2184"
          :temporalEnd "2200"
          :text "Oh God, I don't know why I got myself into this situation...."
          :entity     http://dbpedia.org/resource/God

:_0pub1     is   :PublicationEvent
           :service      http://zattoo.com
           :channel      http://rbb-online.de
           tl:start        "2009-06-15T13:45:30"
           tl:end          "2009-06-15T14:30:00"

:_0summary  is  :TVProgram
             :version_of   :_0
             :duration      "120"
             :has_publication  :_0summarypub1

:_0summarypub1   is   :PublicationEvent
                  ma:locator    "http://twitter.com/rbb/video/235253"
                  :service       http://twitter.com/rbb/
                  :channel       http://rbb-online.de
                  tl:start         "2009-06-09T19:30:00"

:_1          is    :TVRelatedContent
          ma:description "Next week don't miss the exciting new episode of Breaking Bad, exclusively on
RBB!"
          ma:keyword       http://dbpedia.org/resource/Breaking_Bad
          ma:locator          "http://www.twitter.com/rbb/163623623752326735"
          :service    http://www.twitter.com/rbb/
          :channel   http://rbb-online.de/
          :links-to    "http://rbb-online.de/programmes/BreakingBad"
          :has_media      :_0summary
          :associated_with   http://dbpedia.org/resource/Breaking_Bad

:_2          is   :UserRelatedContent
          ma:description "YAY!!! The next Breaking Bad episode is coming to RBB on Monday! I can't
wait to see if Walt survives!"
          ma:keyword       http://dbpedia.org/resource/Breaking_Bad
```

```
ma:locator        "http://www.facebook.com/jonsnow/posts/623752326735"
:service    http://www.facebook.com/jonsnow
:channel   user/8372nccnncndfdvls[7]
:responds-to    :_1
:associated_with   http://dbpedia.org/resource/Breaking_Bad
```

### 3.4 Implementation Details and Use

The documents will be stored on the webLyzard platform (the basis for the Trans Vector Platform). They should capture the semantics of the ReTV conceptual annotation model, extending webLyzard's own (structural) document format, which consists of five parts:

1. Metadata - document metadata
2. Sentences - everything NLP
3. Features - project specific metadata
4. Relations - doc2doc
5. Annotations - surface forms and entities

Below we provide the mapping defined for the ReTV conceptual model into the webLyzard document model, using also the namespaces given above for vocabulary interoperability.

| **Abstract ReTV annotation model** | **Mapping to webLyzard document model** |
|---|---|
| based on the conceptual RDF model above | red marks new elements in the documents |

| **TVProgram** | **Document (wl:page)** |
|---|---|
| Genre | ma:genre (new) |
| Brand | po:brand (new) |
| Season | po:season (new) |
| Episode | po:episode (new) |
| See_also | dc:references (new) |
| Language | dc:language |
| Title | dc:title |
| Description | wl:description |
| Keyword | Entity [key] |
| Duration | wl:duration |
| Fragment | wl:feature/wl:key="video analysis" |
| - temporalStart | wl:feature [begintime] |
| - temporalEnd | wl:feature [endtime] |
| - Keyframe | wl:feature [keyframes] |
| - Concept | wl:feature [concepts] |

[7] A user hash (mapping a set of user names to a fixed value) would uniquely disambiguate the user of the service across vectors which could be very valuable for user targeting at the individual level. However, we will not pursue this in ReTV outside of Zattoo's own user IDs which are also not shared outside of the organisation and hence user IDs of any form will not be stored in TVP documents, thus respecting EU data privacy laws.

| | |
|---|---|
| - Logo | wl:feature [logos] (new) |
| Sentence | |
| - temporalStart | wl:sentence/temporalStart |
| - temporalEnd | wl:sentence/temporalEnd |
| - Text | wl:sentence |
| - Entity | wl:sentence/Entity [key] |
| Publication | |
| - Locator | ma:locator |
| - Service (preferred: URI) | wl:user_id (URL) |
| - Channel | po:broadcaster (new, URI) |
| - start | tl:start (new) |
| - end | tl:end (new) |
| is_version_of | dc:isVersionOf |
| **RelatedContent** | **Document (wl:page)** |
| Title | dc:title |
| Description | wl:description |
| Keyword | wl:annotation/wl:key |
| Locator | dc:identifier |
| Links_to | dc:references |
| Associated_with | po:brand (new) |
| Has_media | -- |
| Responds_to | dc:source |
| Publication | |
| - Locator | ma:locator |
| - Service (preferred: URI) | wl:user_id (string) |
| - Channel | po:broadcaster[8] (new) |
| - start | dc:issued |

As could be seen in the conceptual model, where relevant we make use of Linked Data URIs. We set up a local knowledge graph (the Semantic Knowledge Base, a.k.a. SKB) to store entities we will regularly reference in documents, e.g. Works (the TVProgram brands) and Events (see Deliverable D2.1). In terms of Works entities, we queried WikiData for entities of type TV station or TV series, starting with our own seed lists, storing a local copy of each matching entity found.

An example for a TV program (Brand):

```
<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:type>
<https://www.wikidata.org/wiki/Q15416>

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:type> <po:Brand>

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:type>
<http://dbpedia.org/ontology/TelevisionShow>

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:label> "The Big Bang
Theory"@en
```

[8] The channel could also be an individual web and social media user but as noted in the previous footnote we will not store any form of user identifier with the documents.

```
<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:label> "The Big Bang
Theory"@de

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:label> "The Big Bang
Theory"@nl

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:label> "The Big Bang
Theory"@es

<http://dbpedia.org/resource/The_Big_Bang_Theory> <rdfs:label> "The Big Bang
Theory"@fr
```

An example for a TV station (Broadcaster):

```
<http://dbpedia.org/resource/ProSieben> <rdfs:type>
<https://www.wikidata.org/wiki/Q15265344>

<http://dbpedia.org/resource/ProSieben> <rdfs:type> <po:Broadcaster>

<http://dbpedia.org/resource/ProSieben> <rdfs:label> "prosieben"@en

<http://dbpedia.org/resource/ProSieben> <rdfs:label> "prosieben"@de

<http://dbpedia.org/resource/ProSieben> <rdfs:label> "prosieben"@nl

<http://dbpedia.org/resource/ProSieben> <rdfs:label> "prosieben"@es

<http://dbpedia.org/resource/ProSieben> <rdfs:label> "prosieben"@fr
```

A local knowledge graph can be more efficient in joining queries when additional metadata from the SKB is to be used by the platform, especially as we keep the size of the graph limited to the entities we are using (e.g. create a new TVProgram entity only when we start to collect data for it). Also we can keep the graph more easily up-to-date and perform data cleaning (since the metadata directly extracted from public Knowledge Graphs like DBPedia or WikiData may be incomplete or incorrect). Every entity in the SKB is linked to its equivalent entities in the public Knowledge Graphs as much as possible when we know we derive such links correctly.

Documents in the webLyzard platform combine the TVProgram as a single instance of a broadcast with the PublicationEvent itself, as both sets of metadata can be mapped from each EPG entry. We will use our NER/NEL service RECOGYNZE to link documents to the TVProgram entity in the Semantic Knowledge Base, where an entity represents the "brand" of the TV program (we use "brand" here following the BBC Programmes Ontology modelling of a TV program as made up of a Brand, Series and Episode).

## 3.5 Results

We have defined a mapping from the EPG data provided by Genistat to our annotation model:

| API response (one TV program) | webLyzard document mapping (using JSON for readability and comparability) |
|---|---|
| {<br>    "pid": 135738682,<br>    "dc_name": "sf-1.AxelSpringer",<br>    "epg_source": "AxelSpringer", | {<br>    "wl:id": "135738682",<br>    "po:brand":<br>"http://de.dbpedia.org/resource/Kulturplatz", |

```
    "description": "Die Kulturstiftung Pro
Helvetia soll das Schweizer Kunstschaffen im
Ausland bekannt machen. Sie verhilft
hiesigen Kulturschaffenden nicht nur zu
Ausstellungen und Veranstaltungen in aller
Welt, sie gewährt ihnen auch Atelier- und
Recherchestipendien auf vier Kontinenten.
\"Kulturplatz\" fragt, wie nachhaltig diese
Fördermassnahmen sind.",
    "id": 158,
    "channel_id": 1,
    "title": "Kulturplatz",
    "subtitle": "Was bringt
Kulturaustausch?",
    "start_broadcast_stream_time_s":
1520219400,
    "end_broadcast_stream_time_s":
1520220900,
    "start_stream_time_s": null,
    "end_stream_time_s": null,
    "source": "zattoo_program",
    "scene_sequence_id": null,
    "srt_url": null,
    "parent_id": null,
    "position": 0,
    "inserted_time":
"2018-03-08T16:10:20.335472",
    "last_updated_time": null
  },
```

```
    "dc:references":
"https://www.imdb.com/title/tt1025770/",
    "dc:references":
"https://www.srf.ch/sendungen/kulturplatz",
    "dc:language" : "de",
    "dc:title" : "Kulturplatz",
    "wl:description" : "Die Kulturstiftung Pro
Helvetia soll das Schweizer Kunstschaffen im
Ausland bekannt machen. Sie verhilft hiesigen
Kulturschaffenden nicht nur zu Ausstellungen
und Veranstaltungen in aller Welt, sie gewährt
ihnen auch Atelier- und Recherchestipendien
auf vier Kontinenten. \"Kulturplatz\" fragt, wie
nachhaltig diese Fördermassnahmen sind.",
     "Entity" :
"http://de.dbpedia.org/resource/Pro_Helvetia",
     "Entity" :
"http://de.dbpedia.org/resource/Atelier",
     "Entity" :
"http://de.dbpedia.org/resource/Kulturplatz",
     "wl:user_id" : "http://www.zattoo.com/#",
     "po:broadcaster" :
"https://www.srf.ch/tv/srf-1#",
     "tl:start" : "2018-03-05T03:10:00+00:00",
     "tl:end" : "2018-03-05T03:35:00+00:00"
},
```

An example document created by the social media mirror is given below (spacing added for readability):

```
<wl:page xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:wl="http://www.weblyzard.com/wl/2013#" parent_url="https://www.facebook.com/cafepuls/"
dc:format="text/plain" dc:title="Re: CafÃ© Puls" xml:lang="de" wl:id="4657387175078898792"
wl:jonas_type="facebook"
wl:nilsimsa="382AD90406089988203A8882A2B020C244200A9DDF342262D13502410C63E506"
wl:post_type="post">

<wl:sentence wl:dependency=""-1:ROOT 0:p 0:appos 0:attr""
wl:id=""70688a366a0897b41582de9877b2a827"" wl:is_title=""true"" wl:pos=""NN $. NN NN""
wl:token=""0    2 2     3 4     8 9     13""><![CDATA[Re: CafÃ© Puls]]></wl:sentence>

 <wl:sentence wl:dependency=""-1:ROOT 0:NMOD 7:det 7:advmod 7:nsubj 6:advmod 7:neg 0:NMOD""
wl:id=""4769329ded8ebd6b0fad2b5146f838a8"" wl:pos=""ADV VAFIN ART ADJD NE ADV PTKNEG ADJD""
wl:token=""0    11 12    16 17    20 21    25 26    33 34    39 40    45 46    60""><![CDATA[Hoffentlich wird
der pfau Ronaldo heuer nicht weltfussballer]]></wl:sentence>
```

```
</wl:page>
```

This XML output is indexed by webLyzard in their Elasticsearch index for scalability. The above example is for a Facebook post of the channel of Cafe Puls (a program of Austrian TV channel PULS4) - note that the channel title and text content of the post are both modelled as sentences and parsed by a Natural Language Processing (NLP) parser, which is a precondition for the Named Entity Recognition (NER) done subsequently in the ReTV data ingestion pipeline.

We expect our annotation model and vocabularies to stay stable throughout the project, with extensions possible on request from other partners as the services develop which are making use of the collected data on the platform.

# 4. Concept-Based Video Abstractions

We have implemented technologies to label input videos with low level concept-based representations, i.e. to analyze video content in simpler, more abstract representations that consist of concepts depicted in parts of the video. Specifically, we implemented methods that perform temporal segmentation and visual analysis in video fragment level. The adopted temporal segmentation algorithms automatically partition the video in smaller meaningful fragments, called scenes, shots and subshots. Regarding the visual analysis of the representative keyframes, a state of the art deep learning architecture has been adopted. In the following we describe the initial methods for video fragmentation (Section 4.1) and concept-based annotation (Section 4.2) in ReTV.

## 4.1 Video Fragmentation

### 4.1.1 Problem Statement

The scope of the task is to identify the underlying temporal structure of the video considering various levels of granularity. As a first step, the task aims to detect the elementary building blocks of an edited video, which are called shots and are defined as sequences of frames captured uninterruptedly with the use of a single camera. Then, building on the extracted knowledge about the shot-level structure of the video the task will try to identify the story-telling parts of the video. These parts are called scenes and formed by grouping consecutive shots into bigger sets, thus defining a less fine-grained segmentation of the video compared to shots. Finally, shots with dynamic and gradually changing visual content will be decomposed into smaller and visually coherent parts, called subshots, thus producing another more fine-grained segmentation of the video.

The results of video fragmentation analysis, i.e. the defined video scenes, shots and subshots, will serve as input to the brand-detection algorithms of T1.3, to allow a fragment-level spatio-temporal labeling of the video regarding the shown brands. Moreover, the extracted information about the temporal structure of the video (in the aforementioned three levels of temporal granularity) will be used by T1.4, to extend the annotation model. Finally, the defined video fragments will be utilized by T3.3, to create new video representations, i.e. static and dynamic video summaries, that are fully adapted to the users' profiles.

### 4.1.2 State-of-the-Art Survey

The fragmentation of a video into shots basically relies on the detection of transitions between pairs of consecutive shots of the video (these transitions can be both abrupt and gradual), and given the accuracy of several proposed approaches for video shot segmentation, this task is now considered as a solved one. Indicative approaches that exhibit state-of-the-art performance rely on the combination of local and global descriptors (Apostolidis, 2014), the statistical analysis of color features over sequences of video frames (Baraldi, 2015a), and the use of fully convolutional neural networks (Gygli, 2017). For the identification of the different scenes of a video, which are semantically coherent and temporally consistent video fragments that correspond to the different stories presented in it (similar to the chapters of a DVD), a variety of methods have been proposed. Some of them rely only on the visual modality of the video and determine the story-telling parts by grouping shots into bigger sets that correspond to the scenes of the video, based on similarity matrices that consider the visual similarity and the temporal proximity of shots (Baraldi, 2016a). Other techniques propose the use of hierarchical clustering (Baraldi, 2015a) or the exploitation of domain-specific rules (Liu 2013), while another method addresses scene segmentation as a general optimization problem and solves it using dynamic programming (Rotman, 2016). Multimodal solutions that combine different modalities of the video (i.e. visual, audio and text) to identify the video scenes have also been proposed. Indicatively, the algorithm of (Rotman, 2016) was extended to also incorporate features extracted from the audio modality (Rotman, 2017). Sidiropoulos et. al (2011), extract high- and low-level audiovisual features to measure shot similarity and using multiple scene transition graphs for each type of feature, they threshold a cumulative confidence value for a scene boundary at a specific location. Baraldi et. al proposed two multimodal deep network approaches (see (Baraldi, 2015b) and (Baraldi, 2016b)) that assess the similarity of the video shots using visual and textual features (extracted from the video transcripts), and cluster adjacent shots together to form the video scenes. In (Son, 2016) visual, audio and text features are extracted from the closed captions of the video to reflect the video semantics in the shot-level, and adaptive multiview spectral clustering ties data into clusters (the video scenes) by preserving complementary information in each view. Finally, the task of video subshot segmentation has been studied actively over the last years driven by the need for an even more fine-grained fragmentation of videos, which can be performed through the segmentation of shots with dynamic and gradually changing visual content into visually coherent parts. The current state-of-the-art addresses this task by evaluating the visual coherence over sequences of frames, e.g. the DCT-based algorithm presented in (Teyssou, 2017), or through the detection of different camera-related activities that take place during the recording of the video, such as camera pan/tilt, camera zoom in/out and so on. Camera motion can be estimated with the help of optical flow modeling methods (Apostolidis, 2018), or techniques that compute the homography between a pair of images with the help of local descriptors (Mei, 2013), and motion information can be also combined with saliency maps for subshot segmentation (Abdollahian, 2010).

### 4.1.3 ReTV Approach

Video segmentation into shots will rely on the existing method from (Apostolidis, 2014). This method represents the visual content of each video frame by extracting an HSV histogram and a set of ORB descriptors (Rublee, 2011), thus being able to detect the dissimilarity between a pair of frames, both in color distribution and at a more fine-grained structure level. Then, both abrupt and gradual transitions are detected by quantifying the change in the visual content of successive or neighboring frames of the video (through an image matching process), and comparing it against experimentally specified thresholds that indicate the existence of abrupt and gradual shot transitions. Erroneously detected abrupt transitions are removed by applying a flash detector, while false alarms are filtered out after re-evaluating the defined gradual transitions with the help of a dissolve and a wipe detector that rely on the algorithms introduced in (Su, 2005) and (Seo, 2009). Finally, a simple fusion approach (i.e. taking the union of the detected abrupt and gradual transitions) forms the output of the algorithm. The processing pipeline for video shot segmentation is illustrated in Fig. 2.

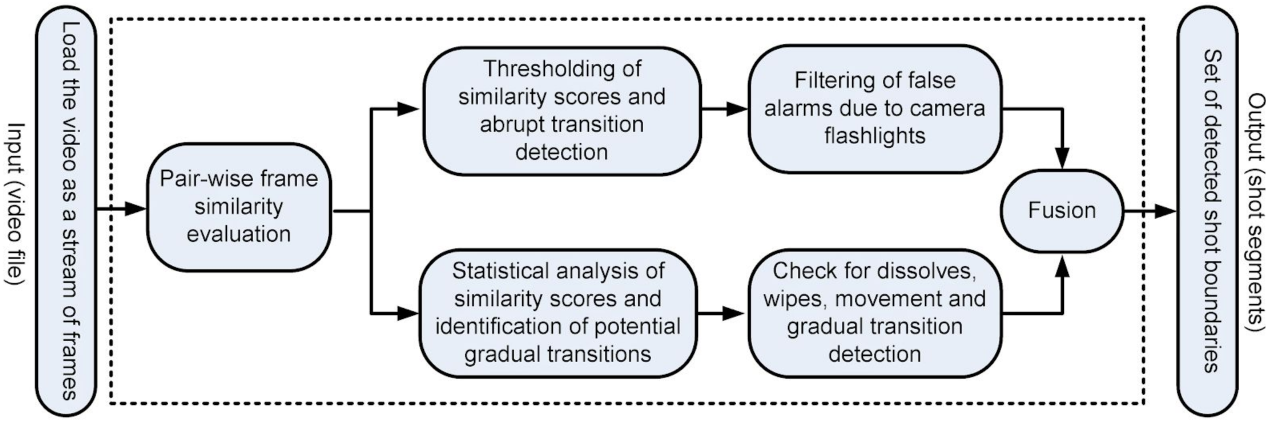
Input (video file)
Load the video as a stream of frames
Pair-wise frame similarity evaluation
Thresholding of similarity scores and abrupt transition detection
Filtering of false alarms due to camera flashlights
Statistical analysis of similarity scores and identification of potential gradual transitions
Check for dissolves, wipes, movement and gradual transition detection
Fusion
Set of detected shot boundaries
Output (shot segments)

Figure 2: The processing pipeline of the used approach for video shot segmentation.

Video segmentation into scenes will be based on the visual-based component of the algorithm from (Sidiropoulos, 2011). The algorithm jointly considers both the content similarity (i.e., visual similarity assessed by comparing HSV histograms extracted from the keyframes of each shot) and temporal consistency among the video shots, in order to group them into scenes with the help of two extensions of the Scene Transition Graph (STG) technique (Yeung, 1998). The first one reduces the computational cost of STG-based shot grouping by considering shot linking transitivity and the fact that scenes are by definition convex sets of shots, while the second one builds on the former to construct a probabilistic framework that alleviates the need for manual STG parameter selection. Based on these extensions, the algorithm can identify the scene-level structure of videos belonging to different genres, and provide results that match well the human expectations. The processing pipeline for video scene segmentation is depicted in Fig. 3.

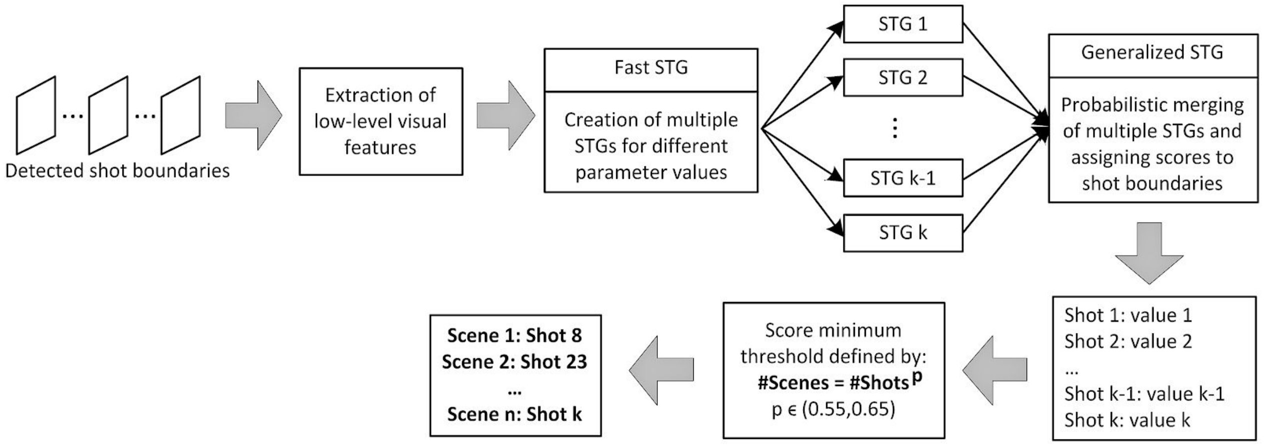

Figure 3: The processing pipeline of the used approach for video scene segmentation.

For video segmentation into subshots we assessed the performance of two existing approaches. The former one, described in (Teyssou, 2017), segments a shot (or a single-shot video) into subshots by detecting visually coherent parts of it (i.e. sequences of frames having only a small and contiguous variation in their visual content). For this, the visual resemblance of neighboring video frames is assessed by representing the visual content of each frame with the help of the Discrete Cosine Transform (DCT) (through the process presented in Fig. 4) and computing the cosine similarity of the generated descriptor vectors. After analyzing the entire set of (selected) frames the algorithm produces a series of similarity scores, which is smoothed to reduce the effect of sudden, short-term changes in the visual content of the video. The turning points of the smoothed series signify a change in the similarity tendency and therefore a subshot boundary. Through this process the algorithm indicates both subshots with minor or no activity, and subshots with gradually, but also consistently, changing visual content.

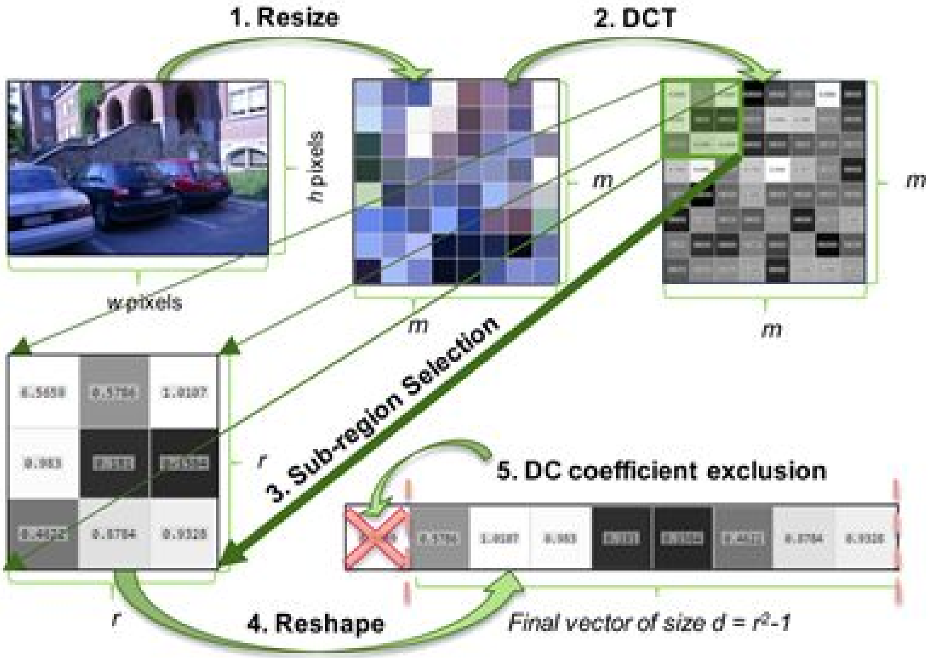

Figure 4: The applied process for extracting the DCT-based representation of each video frame.

The latter one, presented in (Apostolidis, 2018), segments a shot (or a single-shot video) into parts that relate to individual elementary low-level video recording actions, such as camera panning and tilting; camera movement in the three-dimensional space; camera zoom in/out and minor or no camera movement. Subshot detection and categorization rely on the extraction and spatio-temporal analysis of motion information, which is estimated by computing the optical flow between pairs of neighboring video frames. For this, each pair of analysed frames is resized and spatially segmented into four quartiles, and the most prominent corners in each quartile are detected and used for PLK-based (Bouguet, 2001) region-level optical flow estimation. A mean displacement vector is computed for each quartile and the 4 spatially distributed vectors represent the region-level motion activity between the pair of frames (see Fig. 5). Finally, the comparison of the computed region-level motion activity against pre-defined motion models and the quantification of the conducted change, enable the identification of video parts that are directly associated to individual video recording actions, and the fragmentation of the video into action-related su-shots.

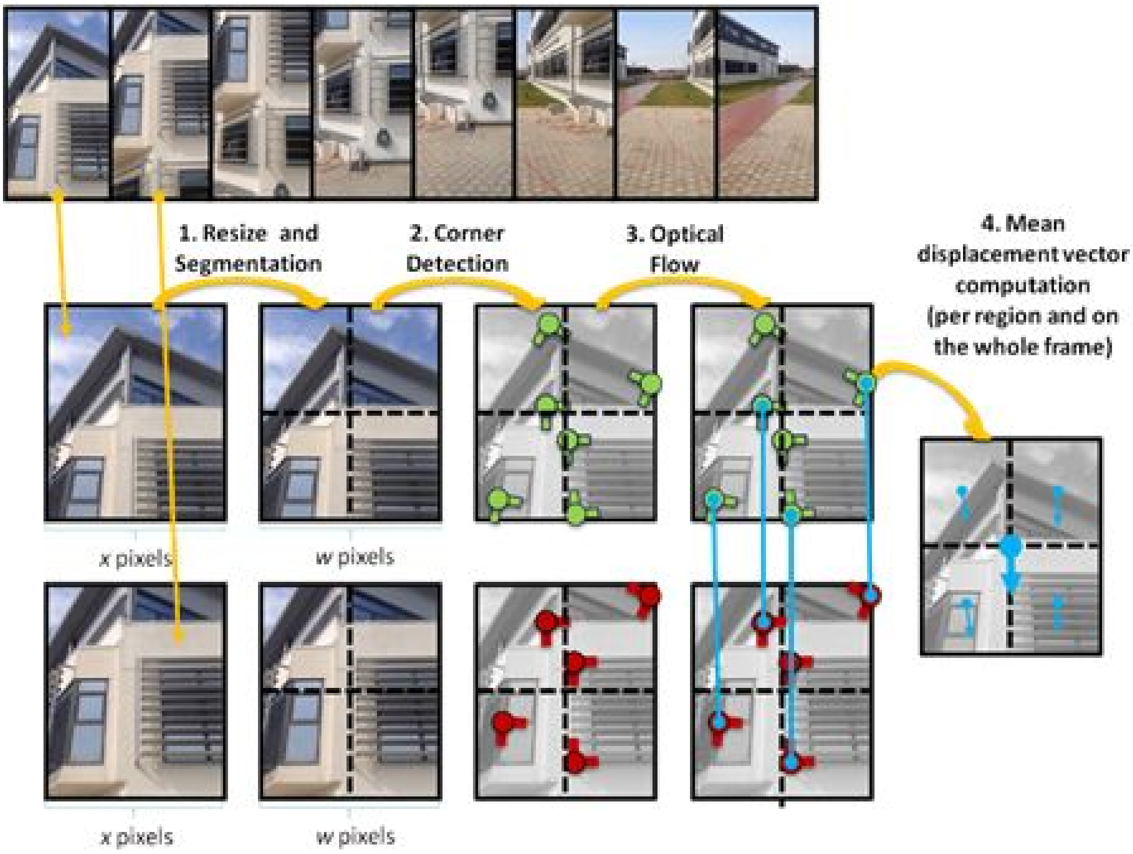
1. Resize and Segmentation
2. Corner Detection
3. Optical Flow
4. Mean displacement vector computation (per region and on the whole frame)
x pixels
w pixels
x pixels
w pixels

Figure 5: The applied process for computing the region-level motion between a pair of frames.

### 4.1.4 Implementation Details and Use

All the aforementioned algorithms have been implemented in C++ programming language, using the OpenCV library for Computer Vision[9]. Aiming at the maximum time-efficiency (i.e. the minimum processing time) for each algorithm, we exploited the processing capabilities of the modern multi-core architectures and developed these algorithms in a way that fully utilizes such parallel processing capabilities.

The functionality of the described algorithms for video fragmentation into scenes, shots and subshots (the latter is performed using the motion-based approach) described in this section is part of CERTH's Video Analysis service for ReTV. The Video Analysis service is discussed in detail in Section 6.

### 4.1.5 Results

The shot segmentation algorithm takes as input a video file and returns the detected shots (see Fig. 6). The detected shots are defined by their start and end time, and represented by a set of characteristic keyframes (by default 3 keyframes per shot). The algorithm is capable of identifying both abrupt and gradual transitions between consecutive shots of the video (see examples of these transitions in Fig. 7).

[9] https://opencv.org/

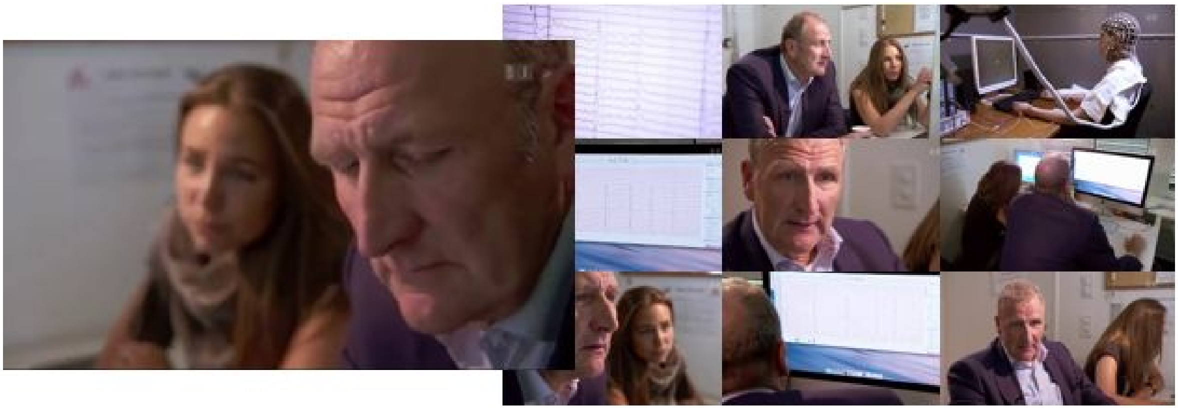

Figure 6: The input video (one indicative keyframe of this ReTV video is shown on the left), and the output (on the right) of the shot segmentation algorithm.

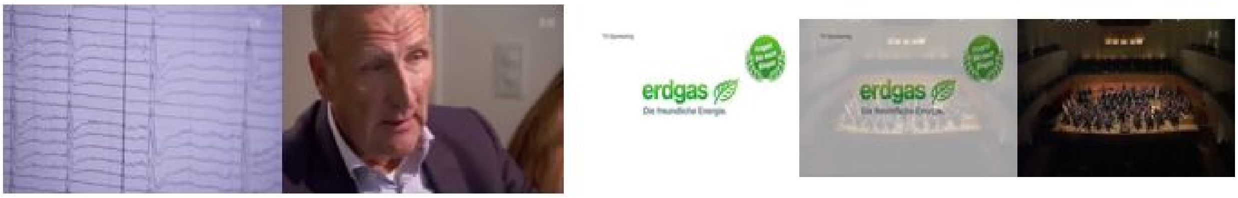

Figure 7: Examples of abrupt (on the left) and gradual (on the right) transitions, found in ReTV content, that are detectable by the shot segmentation algorithm.

The scene segmentation algorithm takes as input a video file and the relevant information about the shots of the video (that are either manually defined or automatically detected by the aforementioned shot segmentation method), and returns the detected scenes (see Fig. 8). The detected scenes are defined by their start and end time, and represented by a set of characteristic keyframes (by default up to 5 keyframes per scene).

The subshot segmentation algorithm can take as input either an un-edited single-shot video, or an edited (multi-shot) video and the relevant information about its shot segments. In the former case the algorithm returns the detected subshots of the video, while in the latter case the algorithm returns the detected subshots of each shot of the video. In any case, each subshot is defined by its start and end time, and represented by a set of characteristic keyframes (by default 3 keyframes per subshot). Fig. 9 shows examples of the subshots detected by the DCT-based approach for subshot segmentation, and Fig. 10 illustrates examples of the subshots detected by the motion-based approach for subshot segmentation, with both figures showing the results of the respective approaches applied on the same shot of a video snippet of ReTV content. We notice that the motion-based approach achieves better accuracy by producing a more fine-grained segmentation (6 subshots instead of 4 subshots detected by the DCT-based approach). The computational complexity of the two approaches is similar, with the DCT-based approach being slightly better (the DCT-based approach average process time is ~6% of the original video duration, while the motion-based approach is ~8% of the original video duration). Taking into consideration these points we choose to adopt the motion-based approach in ReTV.

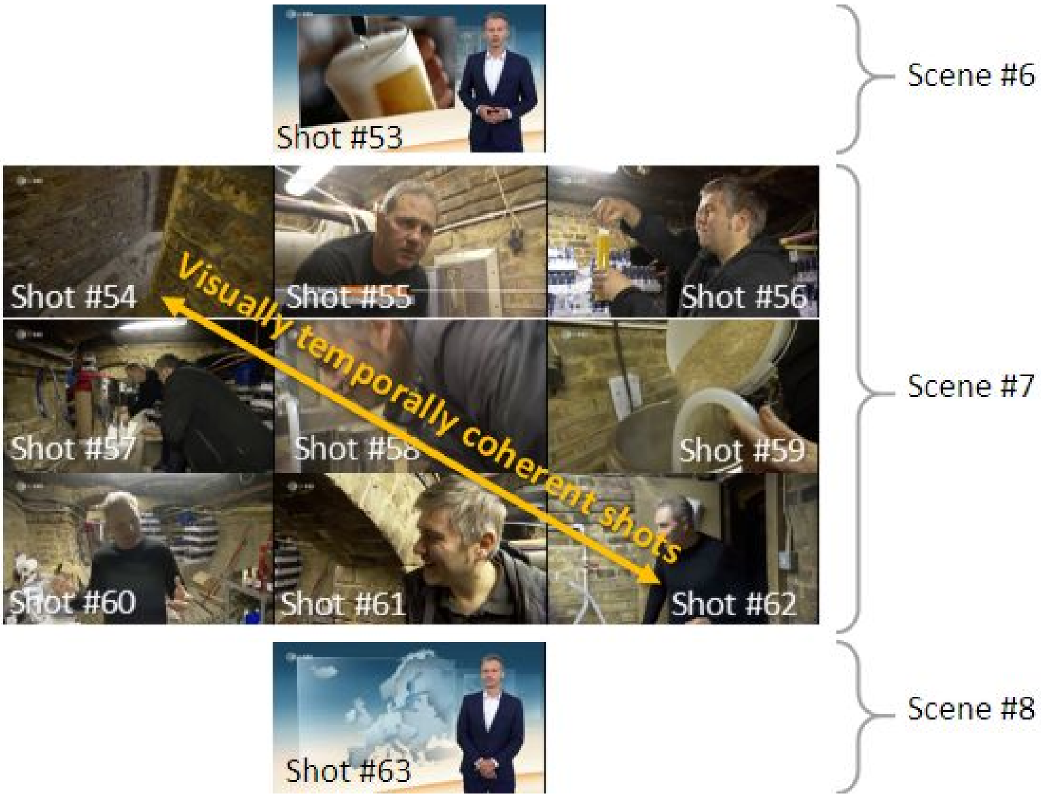

Figure 8: The scene segmentation algorithm grouped a set of visually and temporally coherent shots of the video into one scene ("scene #7" in this sample ReTV video).

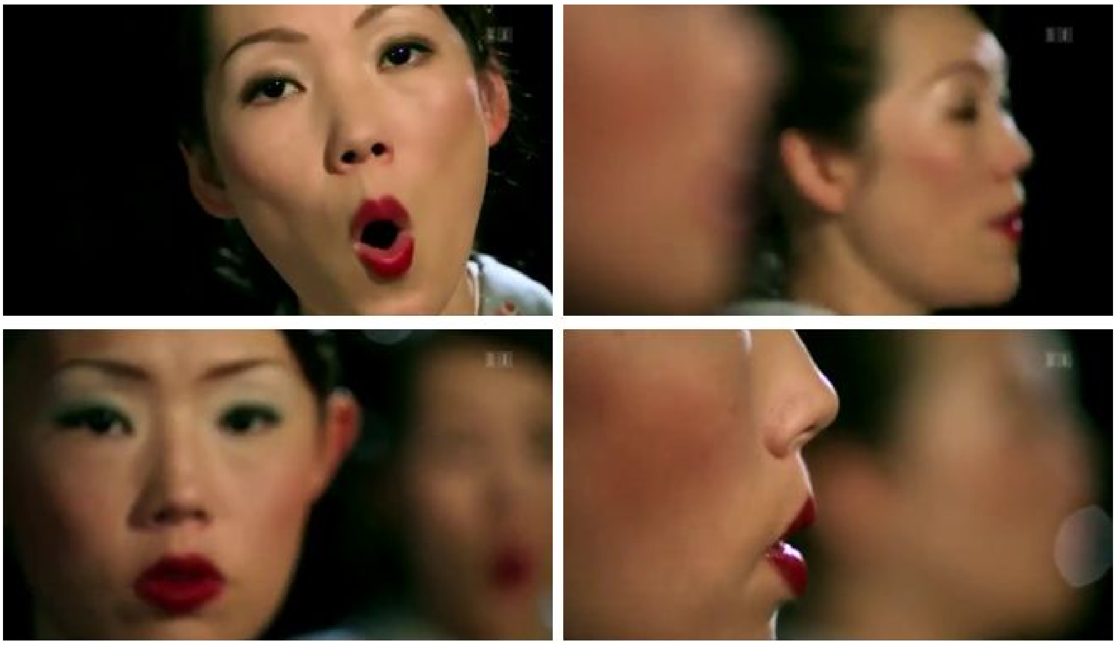

Figure 9: Examples of subshots (represented via their keyframes) detected by the DCT-based subshot segmentation algorithm.

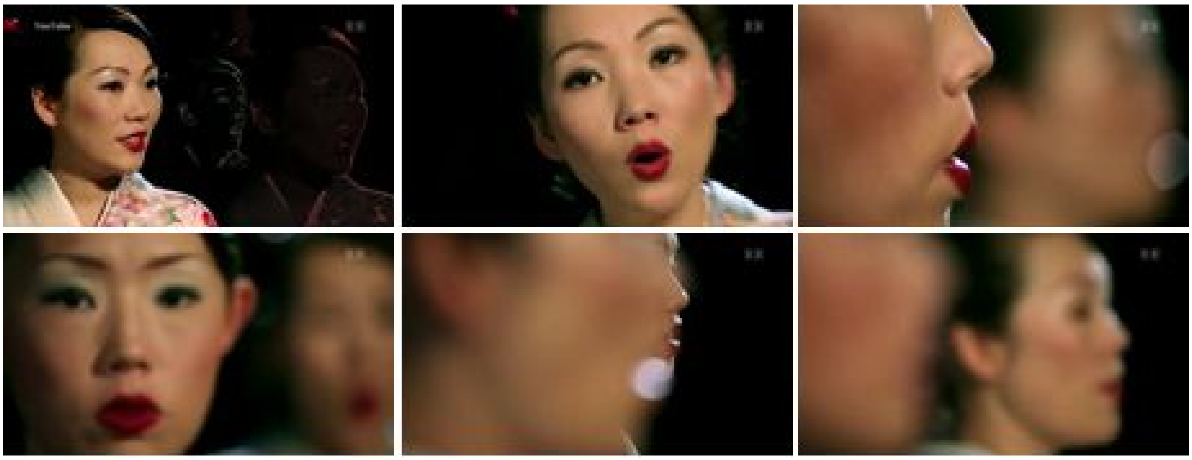

Figure 10: Examples of subshot boundaries detected by the motion-based subshot segmentation algorithm.

## 4.2 Concept-based Annotation

### 4.2.1 Problem Statement

The scope of this task is to annotate video fragments (e.g., keyframes), extracted by the automatic video fragmentation process, with semantic labels referred as concepts. A pre-defined concept pool is used that consists of hundreds of concepts in order to automatically annotate video keyframes with them. Concepts may refer to objects (e.g., "car" and "chair"), activities (e.g., "running" and "dancing"), scenes (e.g., "hills" and "beach"), etc.

Concept-based video fragment analysis are used by T1.4 to extend the annotation model and T3.3 for video summarization, in order for example to cluster conceptually similar fragments,and for video enhancements, e.g., to map text to video fragments.

### 4.2.2 State of the Art Survey

A typical concept-based video annotation system mainly follows the process below: A video is initially segmented into meaningful fragments, called shots; each shot is represented by e.g. one or more characteristic keyframes/images; and, several hand-crafted visual, DCNN-based, textual or audio features are extracted from the keyframes (or any other chosen representation) of each shot. Given a ground-truth annotated video training set, supervised machine learning algorithms are then used to train classifiers (concept classifiers) independently for each concept, using the extracted features and ground-truth annotations. The trained classifiers can subsequently be applied to an unlabeled video shot, following feature extraction, and return a set of confidence scores for the appearance of the different concepts in the shot. A recent trend in video annotation is to learn features directly from the raw keyframe pixels using deep convolutional neural networks (DCNN). DCNNs consist from many layers of feature extractors which makes them having a more sophisticated structure than hand-crafted representations. DCNNs can be used both as standalone classifiers, i.e., unlabeled keyframes are passed through a pre-trained DCNN that performs the final class label prediction directly, using typically a softmax or a hinge loss layer (He, 2016, Simonyan, 2014, Krizhevsky, 2012), and also as generators of video keyframe features, i.e., the output of a hidden layer of the pre-trained DCNN is used as a global keyframe representation (Simonyan, 2014), this latter type of features is referred as DCNN-based.

The small number of labeled training examples is a common problem in video datasets, making it difficult to train a deep network from scratch without over-fitting its parameters on the training set (Pittaras, 2017). For this reason, it is common to use transfer learning that uses the knowledge captured in a source domain in order to learn a target domain without caring about the improvement in the source domain. When a small-sized dataset is available for training a DCNN then a transfer learning technique is followed, where a conventional DCNN, e.g (He, 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, 2017). Recent advances on video annotation also focus on improved architectures (He, 2016), multi-task learning (MTL) (Markatopoulou, 2016), and structured outputs (Schwing, 2015).

### 4.2.3 ReTV Approach

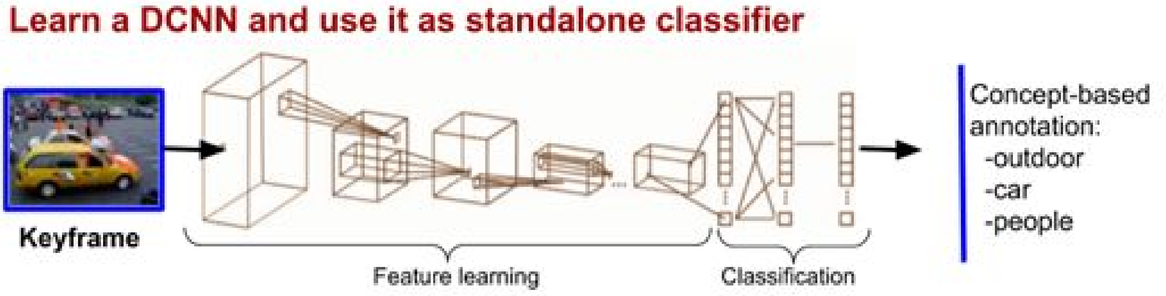
Learn a DCNN and use it as standalone classifier
Keyframe
Feature learning
Classification
Concept-based annotation:
-outdoor
-car
-people

Figure 11: Overview of the proposed method for video fragment-level concept annotation.

For concept-based video annotation we use as a starting point the method of (Markatopoulou, 2018), which is a further extension of the method proposed in (Markatopoulou, 2016). In the latter work, a two-sided deep network architecture that directly learns features from the raw keyframe pixels was presented. Specifically, it used a pre-trained deep network on the large-scale ImageNet dataset[10] and fine-tuned its parameters for the target concepts that appear in the target concept pool. After fine-tuning the network, unlabelled keyframes are passed through the trained DCNN that performs the final class label prediction directly, as presented in Fig. 11. In accordance with the findings of (Markatopoulou, 2018), before fine-tuning the network three modifications have been introduced: Firstly, the two-sided network was converted into a single-side one, that treats the concept detection problem as a multi-label learning one. Secondly, for adapting the network to each of the two pools of concepts initially selected for use in ReTV (TRECVID, Places; more on these is discussed below) we removed the initial classification layer, extended the network by one fully-connected layer, and placed on its top a new classification layer with its number of neurons being equal to the number of concepts in the target concept pool. Thirdly, instead of using the softmax loss function, the employed network adopts an extension of the sigmoid cross entropy loss that models the structure in the output space (i.e., the semantic relations that exist between concept labels, for example, the fact that *daytime* and *nighttime* are negatively correlated concepts).

[10] http://www.image-net.org/

We adapted our deep learning network utilizing the aforementioned technique on two datasets: a) the TRECVID-SIN video annotation dataset (Over, 2013) used for the automatic assignment of semantic tags representing high-level features or concepts to video segments and, b) the Places-365 dataset (Zhou, 2018) used for scene classification. Through this process two concept pools of 323 and 365 concepts respectively are initially adopted for use in ReTV.

### 4.2.4 Implementation Details and Use

The trained deep learning architecture for video annotation is based on the Caffe[11] deep learning framework , and the complete code has been developed in C++. The concept-based annotation method described in this section is part of CERTH's Video Analysis REST service. The Video Analysis service is discussed in detail in Section 6.

### 4.2.5 Results

Since the ReTV content is not annotated with ground truth, we cannot provide a numerical evaluation of the adopted concept-based annotation method. Instead, we perform our evaluation based on visual inspection of indicative results. Figures 12, 13, 14 and 15 illustrate examples using the adopted video segmentation method (shot/scene segmentation and subshot segmentation) discussed in Section 4.1 and the method for concept detection, discussed here, on ReTV content. For the better presentation of the results we present the segmentation of a video snippet to some scenes, a selection of the shots and subshots of this scene, and indicatevely provide the concept annotation of a randomly selected subshot keyframe. We observe that the adopted concept-based annotation method manages to rightly annotate the main elements of the image.

[11] http://caffe.berkeleyvision.org/

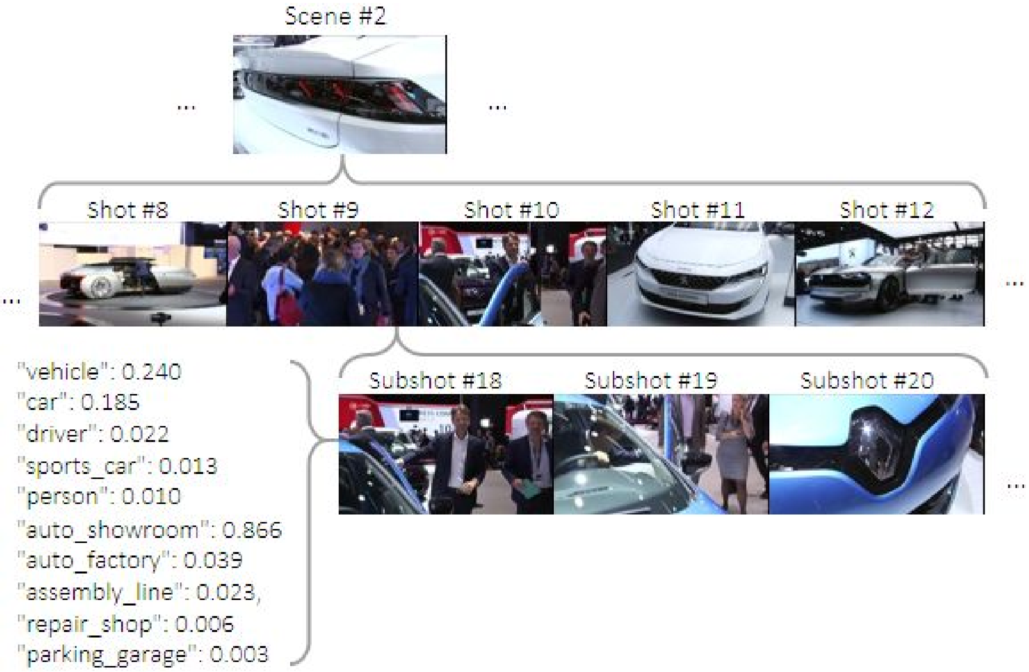

Figure 12: Example of video analysis components on ReTV content (from the German TV channel ZDF).

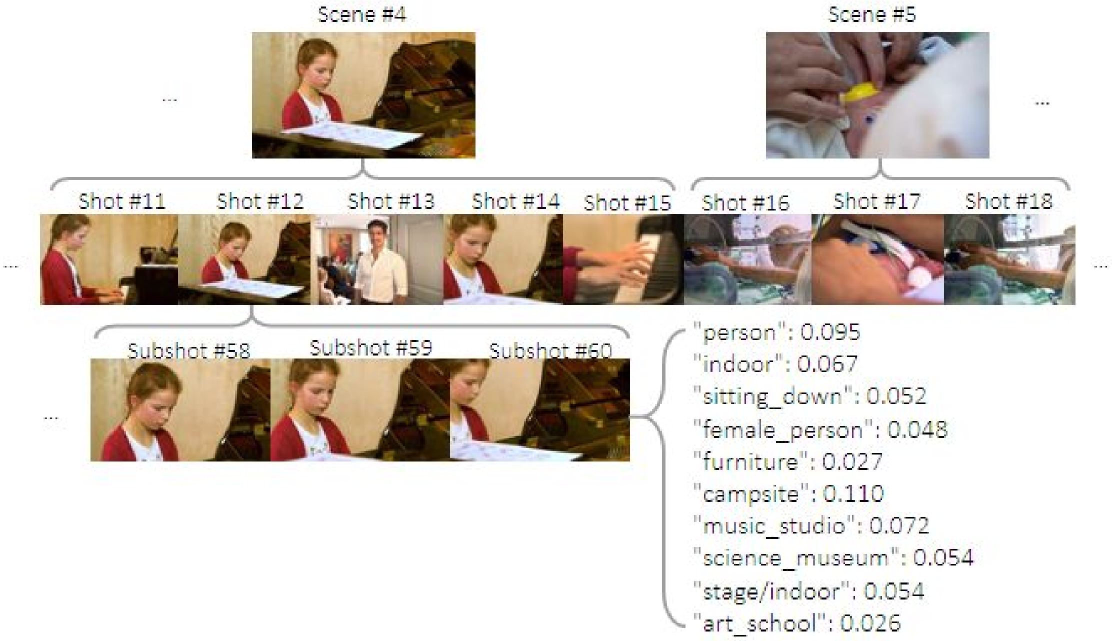

Figure 13: Example of video analysis components on ReTV content (from Zattoo live TV).

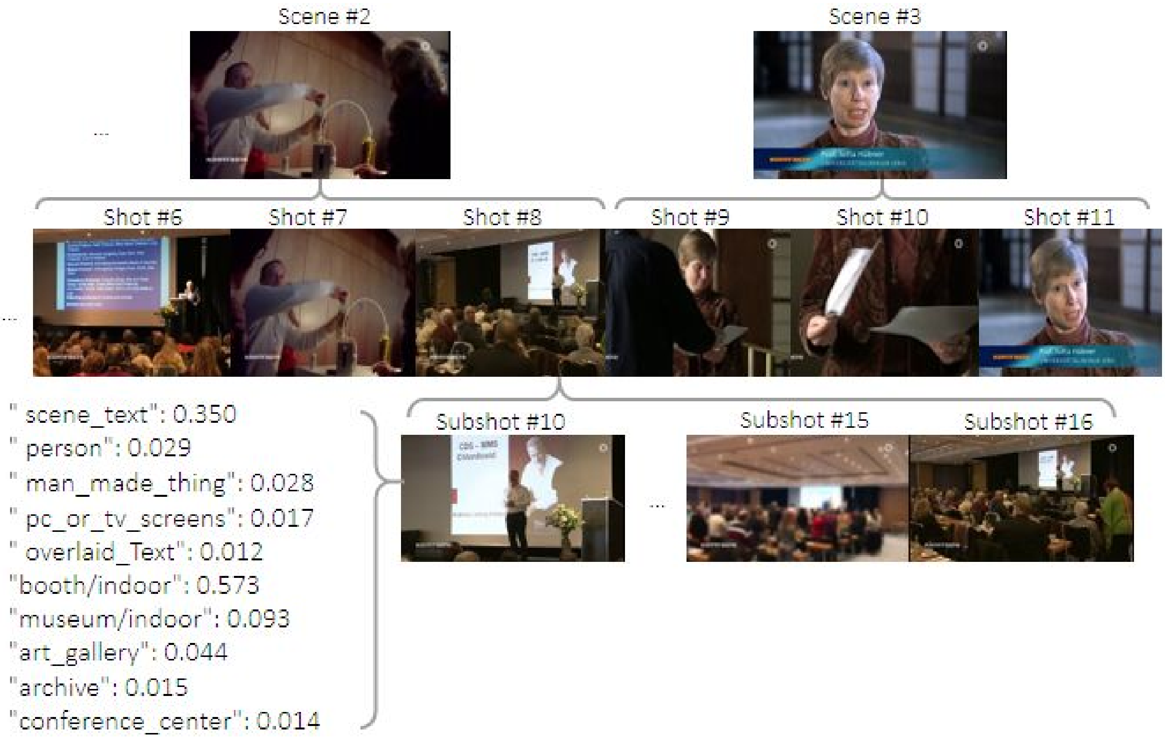

Figure 14: Example of video analysis components on ReTV content (from the rbb TV broadcaster).

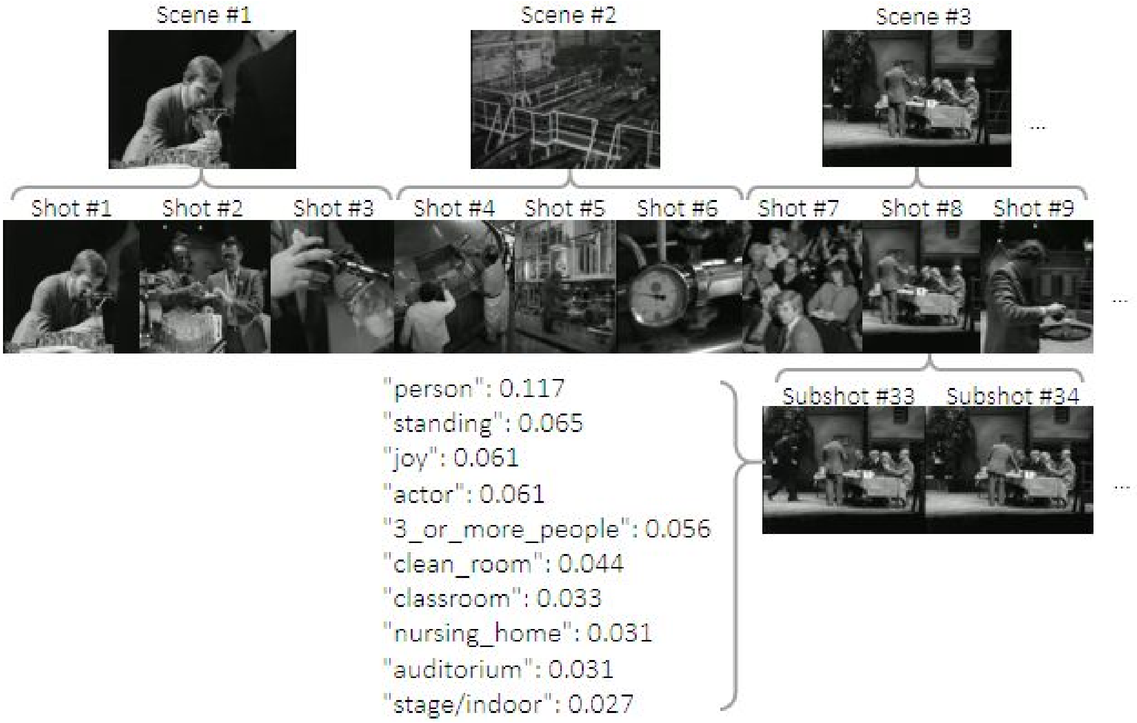

Figure 15: Example of video analysis components on ReTV content (from the Netherlands TV archive NISV).

# 5. Brand Detection

## 5.1 Problem Statement

This task concerns the localization and annotation of brand logos in video fragments, extracted by the automatic video fragmentation process. The purpose of brand detection is two-fold: a) recognize brands in keyframes of the TV program, and b) recognize brands in complete video segments labeled as advertisements by fusing the fragment-level information in order to discover the brand(s) that are the subject of the advertisement.

A subset of the analysis of task T1.2 is used as a starting point of our brand detection algorithm. Specifically, the keyframes of the detected subshots after the temporal segmentation of the video in T1.2 serve as input to the brand-detection visual analysis algorithms.

## 5.2 State of the Art Survey

Logo recognition in images and videos is the key problem in brand detection since it is the first step to detect the brand depicted in a image or video.

The first literature works regarding logo detection (such as (Kalantidis, 2011), (Joly, 2009), (Romberg, 2011)) typically utilized local features (SIFT, SURF features of MSER regions) extracted on a spatial pyramid. Despite several efforts to speed up the look up at query time e.g., construct an inverted index of such configurations, or using hashing techniques (a set of approximate nearest neighbor search methods which is an efficient alternative to exhaustive comparison) for quick retrieval, the complexity of these methods is prohibitive when dealing with a large corpus of video items or numerous classes of logos.

The problem of logo detection is closely related to that of object detection. Due to the success of deep convolutional neural networks (DCNN) on image classification and transfer learning, more recent attempts in object detection are based on the use of DCNNs. A great deal of models have been suggested in the last years. The frameworks of generic object detection methods that utilize DCNNs can mainly be categorized into two types: a) region proposal based and b) regression based. The first type follows a more traditional object detection pipeline, generating region proposals at first and then classifying each proposal into different object categories. The second type regards object detection as a regression or classification problem, adopting a unified framework to achieve final results (categories and locations) directly. The region proposal based methods mainly include R-CNN (Girshick, 2014), SPP-net (He, 2015), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren, 2015), R-FCN (Dai, 2016), FPN (Lin, 2017) and Mask R-CNN (He, 2017), some of which are correlated with each other (e.g. SPP-net modifies RCNN with a SPP layer). The regression/classification based methods mainly includes MultiBox (Erhan, 2014), AttentionNet (Yoo, Park, 2015), G-CNN (Najibi, 2016), YOLO (Redmon, 2016a), SSD (Liu, 2016), YOLOv2 (Redmon, 2016b), DSSD (Fu, 2017) and DSOD (Shen, 2017). To our knowledge, the first method that utilizes DCNN explicitly for logo detection is the method of (Hoi, 2015), with more methods following up such as (Bianco, 2017) (Su, 2017a).

Most recent logo detection methods focus mainly on efficiently expanding the logo set (Tüzkö, 2017) or finding training images from noisy web data (Su, 2017b).

### 5.3 ReTV Approach

In ReTV project we are aiming for a brand detection system that will be able to detect a wide variety of brands. Therefore for the first version of the brand detection system we reviewed the literature for datasets of logos in an effort to construct a rich training dataset. Table 2 reports our findings.

| Dataset | No. of Logos | No. of training instances | No. of test instances | Notes |
|---|---|---|---|---|
| Publicly available datasets | | | | |
| Logos in the wild<br>(Tüzkö, 2017) | 110 | 7429 | - | - |
| WebLogo-2M<br>(Su, 2017b) | 194 | - | 6555 | - |
| TopLogo10<br>(Su, 2017a) | 10 | 487 | 376 | - |
| Logos-32plus<br>(Bianco, Simone, 2017) | 32 | 12302 | - | Expanded version of FlickrLogos-32 |
| FilckrLogos-47<br>(flickrlogos-47, 2017) | 47 | 1778 | 4032 | Re-annotated version of FlickLogos-32 |
| FilckrLogos-32<br>(Romberg, 2011) | 32 | 33 | 1601 | - |
| FlickrLogos-27<br>(Kalantidis, 2011) | 27 | 4536 | 135 | - |
| BelgaLogos<br>(Joly, 2009) | 37 | 2650 | 2650 | - |
| Non-public datasets | | | | |
| Logos-18<br>(Hoi, 2015) | 18 | 16043 | - | - |
| Logos-160<br>(Hoi, 2015) | 160 | 73414 | - | - |

Table 2: Datasets of the literature for logo detection.

We gathered all publicly available training images from the datasets of Table 2 and we constructed the CERTH brand detection dataset. All of the datasets are publicly available except for the Logos-18 and Logos-160 datasets. We merged images regarding the same logo class. The WebLogo-2M (Su, 2017b) proposes a method to gather training images from the web and thus a training dataset is not provided. From the WebLogo-2M we used the test set images as training images, therefore we will not compare our method to literature works that utilise this dataset. Finally, the FlickrLogos-47 and FlickrLogos-32 training images greatly overlap, so we manually selected the unique images. The specifications of the constructed dataset are reported in Table 3.

| Dataset | No. of Logos | No. of training instances | No. of test images |
|---|---|---|---|
| CERTH logo detection dataset | 225 | 19921 | 9992 |

Table 3: Specifications of the CERTH logo detection dataset.

In 2017 TensorFlow[12] (an open source machine learning framework) released machine learning models capable of localizing and identifying multiple objects in images named as the TensorFlow Object Detection API (Huang, 2017). The TensorFlow Object Detection API is an open source framework[13] built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. In this API, the Tensorflow authors offer publicly their own implementations of Faster-RCNN (Ren, 2015) and many other object detection architectures. It is worth noting that this framework is the one that Google utilizes for their own computer vision needs. We used the TensorFlow Object Detection API to train a model for brand detection utilizing the CERTH logo detection dataset. We choose to use the Faster-RCNN architecture implementation because this seems to represent the best compromise between accuracy of detection and time efficiency. Regarding the feature extraction, we used the Inception V3 model (Szegedy, 2016) since this similarly offers a good balance between time efficiency and detection accuracy.

### 5.4 Implementation Details and Use

We employ the TensorFlow open source machine learning framework to use the TensorFlow Object Detection API and wrote our implementation in Python 3.6 programming language. The brand detection system described in this section relies on the outcomes of the video fragmentation of T1.2 (described in Section 4.1). The concept-based annotation of T1.2 (described in Section 4.2) also relies on video fragmentation. Aiming to minimize the dependences of CERTH services for WP1, we combined all these technologies in a single Video Analysis component which we discuss in Section 6.

### 5.5 Results

For training our model we used the merged set of the CERTH logo detection dataset (see Table 3). To compare the performance of our model to literature results of each dataset we evaluated our model using the respective dataset's test set. In Table 4 we report the evaluation results of our model. We observe that our method achieves state-of-the-art performance on three datasets (TopLogo10, FilckrLogos-32 and FlickrLogos-27) out of the seven datasets compared. On the other hand, on two datasets (BelgaLogos and FilckrLogos-47) the performance of our method is significantly lower. This is a deficiency which we will investigate in the near future.

| Dataset | Max reported detection score in the literature | Detection score of ReTV method |
|---|---|---|
| Logos in the wild | **84.2%** mAP at close set, | 79.8% mAP at closed set |

[12] https://www.tensorflow.org/

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

| | | |
|---|---|---|
| (Tüzkö, 2017) | 46.4% at open set<br>(Tüzkö, 2017) | |
| WebLogo-2M<br>(Su, 2017b) | 34.37% mAP<br>(Su,2017b) | N/A (only test set is provided and the this test set was used for training) |
| TopLogo10<br>(Su, 2017a) | 41.8% mAP<br>(Su, 2017a) | **53.3%** mAP |
| Logos-32plus<br>(Bianco, Simone, 2017) | **94.5%** F-score,<br>**95.8%** accuracy<br>(Bianco, Simone, 2017) | 94.13% F-score,<br>92% accuracy |
| FilckrLogos-47<br>(flickrlogos-47, 2017) | **48.1%** mAP<br>(J. Redmon and A. Farhadi,2016) | 27.84% mAP |
| FilckrLogos-32<br>(Romberg, 2011) | 90.3% F-score<br>(Romberg, 2013) | **94.13%** F-score |
| FlickrLogos-27<br>(Kalantidis, 2011) | 53% accuracy<br>(Kalantidis, 2011) | **81.5%** accuracy |
| BelgaLogos<br>(Joly, 2009) | **34.11%** mAP<br>(Joly, 2009) | 23.11% mAP |

Table 4: Evaluation of the initial logo detection model used in ReTV.

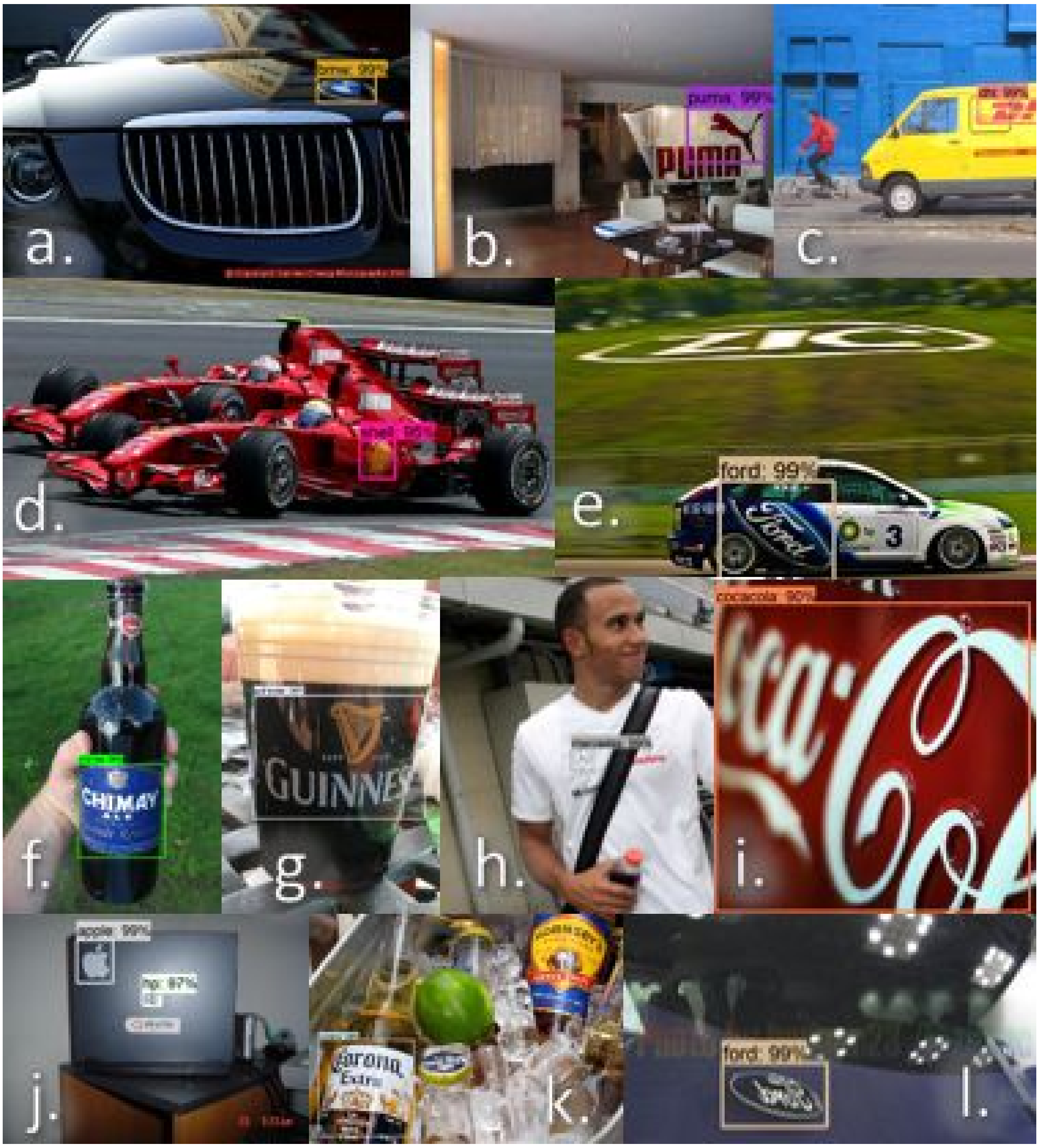

Figure 16: Brand detection examples on images of the CERTH logo detection dataset.

Figure 16 illustrates some example detections on images of the aggregated CERTH logo detection dataset. We observe that the developed method can detect brand logos with perspective distortions (see sub-figures a, g and i), logos that are partially depicted in the image (see sub-figures c and i), logos of small size with respect to the original image (see sub-figures h, j and d - note that for presentation purposes these images are included cropped in Fig. 16) and logos that are horizontally flipped (see sub-figures e and l).

# 6. Video Analysis Component

The video fragmentation techniques discussed in Section 4.1, the concept-based video abstractions discussed in Section 4.2, as well as the brand detection method discussed in Section 5, have all been incorporated into a single Video Analysis component. The Video Analysis component is a REST service that: a) retrieves a video file, either from a web page or a provided zip file, b) segments the video file to scenes, shots and subshots, c) selects a number of representative frames (keyframes) from each segmented scene, shot and subshot, d) performs concept detection on the keyframes, e) performs brand detection on the keyframes.

The component works in a asynchronous way, thus, to use the service, there are three types of calls:

1. the "Start" call, which provides the input video and initiates the processing,
2. the "Status" call, which queries the status of an initiated service instance (session),
3. the "Results" call, which returns the results of a successfully completed service instance (session).

The "start" call is an HTTP POST call, while the "status" and "results" calls are HTTP GET calls.

To issue a "status" call, we can use the following command:

```
HTTP POST http://160.40.49.127:8090/va
```

The obligatory JSON structured argument which must be used in the body of this "start" call is one of the following:

- "video_url": use a single video URL from a web page as input to the service. The video will be downloaded from the web page[14] and analysed.
- "zip_url": use a compressed (zip) file URL containing a single video file to use as input to the service. The compressed file will be downloaded, the video file will be extracted and analysed. Supported video files extensions are ".mp4", ".avi", ".flv" and ".webm".

We implemented an optional JSON structured argument, namely the "brand_detection" argument, which accepts the "0" or "1" string values. Setting this to "0", the brand detection is disabled for the current session (resulting in faster execution times). By default "brand_detection" is "1", which means that the brand detection is conducted.

The "start" call returns a JSON file. If the call is successful, the JSON file contains the following fields:

```
"message": "The call has been received"
"session": a unique id of the call (used later to get status or results)
```

If the "start" call is NOT successful, the JSON file contains the following field:

```
"message": "No video or zip url provided"
```

To issue a "status" call, we can use the following command:

```
HTTP GET http://160.40.49.127:8090/va/status/<session>
```

where <session>, is the unique ID received when calling the service.

The "status" call returns a JSON file. If the call is successful, the JSON file contains a field which provides one of the following message strings:

[14] For a complete list of supported sites, see https://rg3.github.io/youtube-dl/supportedsites.html

- "VIDEO DOWNLOAD STARTED"
- "VIDEO DOWNLOAD FAILED"
- "VIDEO DOWNLOAD COMPLETED"
- "VIDEO SEGMENTATION STARTED"
- "VIDEO SEGMENTATION FAILED"
- "VIDEO SEGMENTATION COMPLETED"
- "VIDEO ANALYSIS STARTED"
- "VIDEO ANALYSIS COMPLETED"
- "ZIP DOWNLOAD STARTED"
- "ZIP DOWNLOAD FAILED"
- "ZIP DOWNLOAD COMPLETED"
- "ZIP EXTRACTION STARTED"
- "ZIP EXTRACTION FAILED"
- "ZIP EXTRACTION COMPLETED"

Each of these messages denotes the current stage of the whole procedure. If the "VIDEO ANALYSIS COMPLETED" message has been received, you can proceed to make the results call. If the "status" call is not successful, the JSON file contains the following field:

```
"message": "The status you requested does not exist"
```

To issue a "results" call, we can use the following command:

```
HTTP GET http://160.40.49.127:8090/va/results/<session>
```

where <session>, is the unique ID received when calling the service. This call returns a JSON file. Iif the "results" call is not successful, the JSON file contains the following field:

```
"message": "The results you requested do not exist"
```

If the call is successful, the results JSON file of the Video Analysis component is returned. The output JSON contains the following fields: "filename", "url", "framerate", "duration", and the following arrays: "scenes", "shots", "subshots". The latter arrays are nested, i.e. the subshots of a shot are include in their shot and the shots of a scene are included in their scene. The JSON output from a sample Video Analysis session is reported in Listing 1. The video submitted to the service consisted of 2 scenes, the first one containing a single shot with two subshots and a second scene with a single shot and a single subshot. For presentation purposes we trimmed the links to the keyframes, and the list of the returned concepts.

```
{
  "filename": "test_for_D1.1.mp4",
  "url": "Provided via ZIP file",
  "framerate": 29.97003,
  "duration": "0:01:54.013900",
  "scenes": [
    {
      "scene_id": "sc1",
      "begintime": 0.0333666666333,
      "endtime": 114.01389988598609,
      "concepts": {
        "Desert": 0.3330736,
        "Outdoor": 0.1229313,
        "Rocky_Ground": 0.03835019,
        .
        .
        .
      },
      "keyframes": [
        {
          "time": 28.528499971471497,
          "url": "http://160.40.49.127:8090/keyframe/..."
        },
   ],
   "shots": [
      {
         "shot_id": "sh1",
         "begintime": 0.0333666666333,
         "endtime": 114.01389988598609,
         "concepts": {
            "Desert": 0.3330736,
            "Outdoor": 0.1229313,
            "Rocky_Ground": 0.03835019,
            .
            .
            .
         },
         "keyframes": [
            {
               "time": 28.528499971471497,
               "url": "http://160.40.49.127:8090/keyframe/..."
            }
         ],
         "subshots": [
            {
               "subshot_id": "sb1",
               "begintime": 0.0333666666333,
               "endtime": 7.474133325859199,
               "concepts": {
                  "Desert": 0.3330736,
                  "Outdoor": 0.1229313,
                  "Rocky_Ground": 0.03835019,
                  .
                  .
                  .
               },
               "keyframes": [
                  {
                     "time": 28.528499971471497,
                     "url": "http://160.40.49.127:8090/keyframe/..."
                  }
               ],
               "brand_detection": []
            },
            {
               "subshot_id": "sb2",
               "begintime": 7.5074999924925,
               "endtime": 13.5468666531198,
               "concepts": {
                  "Desert": 0.3330736,
                  "Outdoor": 0.1229313,
                  "Rocky_Ground": 0.03835019,
                  .
                  .
                  .
               },
               "keyframes": [
                  {
                     "time": 28.528499971471497,
                     "url": "http://160.40.49.127:8090/keyframe/..."
                  }
               ],
               "brand_detection": []
            }
         ]
      }
   ]
}
{
   "scene_id": "sc2",
"begintime": 114.0333666666333,
"endtime": 152.01389988598609,
"concepts": {
   "Desert": 0.3330736,
   "Outdoor": 0.1229313,
   "Rocky_Ground": 0.03835019,
   .
   .
},
"keyframes": [
   {
      "time": 142.528499971471497,
      "url": "http://160.40.49.127:8090/keyframe/..."
   },
],
"shots": [
   {
      "shot_id": "sh2",
      "Begintime": 114.0333666666333,
      "endtime": 152.01389988598609,
      "concepts": {
         "Desert": 0.3330736,
         "Outdoor": 0.1229313,
         "Rocky_Ground": 0.03835019,
         .
         .
         .
      },
      "keyframes": [
         {
            "time": 146.528499971471497,
            "url": "http://160.40.49.127:8090/keyframe/..."
         }
      ],
      "subshots": [
         {
            "subshot_id": "sb3",
            "begintime": 114.0333666666333,
            "endtime": 152.474133325859199,
            "concepts": {
               "Desert": 0.3330736,
               "Outdoor": 0.1229313,
               "Rocky_Ground": 0.03835019,
               .
               .
               .
            },
            "keyframes": [
               {
                  "time": 144.528499971471497,
                  "url": "http://160.40.49.127:8090/keyframe/..."
               }
            ],
            "brand_detection": [
               {
                  "class": "nestle",
                  "score": 0.8701171,
                  "x1": 604,
                  "y1": 420,
                  "x2": 767,
                  "y2": 632
               }
            ]
         }
      ]
   }
]
```

```
    }
  ]
}
```

Listing 1: The JSON output of a sample session of the Video Analysis component.

All successfully completed sessions that are older than 48 hours are automatically deleted. After that time, any "status"/"results" calls for these sessions will return the message "The status/results you requested does/do not exist". All failed sessions that are older than 48 hours are automatically moved to another folder for debugging purposes. After that time, any "status"/"results" calls for these sessions will also return the message "The status/results you requested does/do not exist".

# 7. Conclusion and Outlook

In this deliverable, we described the first version of the ReTV data ingestion, analysis and annotation components. We described which data (i.e., identified media vectors to support in TVP) is required and how to retrieve it. We set up the annotation model to be used (chosen metadata and vocabularies) and the resulting support for semantic representation of such annotations through a *Knowledge Graph*. We presented in detail the implementation and the evaluation of the first version of video analysis components.

We will monitor the EPG and social media data collection, subsequently expanding the set of TV programmes for which we collect related data, and iteratively improving data quality by sampling documents being added to the TVP, identifying any errors and implementing corrections. For example, a future challenge is to disambiguate well enough references to TV series titles when they are quite generic in natural language (e.g. RBB's TV series "Panda, Gorilla & Co." as opposed to any reference to the animals).

We will also evaluate the outputs of the annotation process, which includes both metadata extraction directly from the source (e.g. EPG title and description, broadcast time) as well as metadata generation in our data ingestion pipeline (NER/NEL over textual metadata). We will check the quality of resulting annotations and especially the entity extraction - both in building our own Knowledge Graph of entities (extended in this by Works, i.e. TV program entities) as well as alignment to public Knowledge Graphs (DBPedia, WikiData). Here we also expect opportunities to improve our NER/NEL capabilities in the context of TV data - not only TV program identification but related terms such as presenters, actors or characters. Combined with a well populated Knowledge Graph for TV (i.e. these characters played by these actors appear in this series), the graph disambiguation approach of our RECOGNYZE tool for NER/NEL can be tested to better annotate text with TV related entities.

Regarding video fragmentation, the initial set of technologies presented in Section 4.1 will be revised according to the needs of ReTV and the characteristics of the ReTV video data. The main direction will be to investigate the learning capabilities of modern neural network architectures that are able to capture the temporal evolution of objects and events in videos (such as RNN or LSTM architectures); thus, to combine all three temporal fragmentation granularity levels (scene, shot, subshot) in a joint deep-learning based fragmentation method. The expected advantage of this, beside possible gains in accuracy, is a considerably lower computational complexity, making it easier to scale the video analysis pipeline according to the ReTV needs (multiple channels and vectors).

Concerning concept-based annotation and categorization, the main immediate extension will be to investigate additional concept pools (further to the TRECVID and Places datasets that we already support) that suit the needs of ReTV. This is a requirement that has come up when assessing the results of the current implementation of concept-based annotation on ReTV content, and considering the need to use these annotations not only for further video processing tasks (e.g. summarization), but also for personalization purposes. For this, more extended concept/categories pools will be required. In parallel, at the technological level, we will continue to investigate refinements and extensions to our existing deep learning architecture, so as to improve its accuracy and efficiency. The latter may also be affected by the envisaged support of new and larger concept/categories pools; thus, generating such extended annotations is likely to dictate further algorithmic modifications besides simply re-training the existing deep architecture.

Finally, the technologies presented in Section 5 regarding brand detection will also be further extended and adapted according to the characteristics of the ReTV video data and the analysis requirements of the ReTV platform. Specifically, we will investigate the relevant literature for the most recent and accurate one-shot logo detection deep architectures (joint detection and classification, in one step), and consider replacing the existing two-step (detection, then classification) approach. The motivation behind this is that one-shot detection methods are typically faster than their two-step counterparts. Of equal importance is the extension of the set of brands that the service can detect, in response to the requirements of ReTV project and in collaboration with all partners: instead of just the logos included in publicly-available datasets, we are already in the process of collecting the brands/logos that are of interest to ReTV in relation to the content already being collected. Based on this, we will build our proprietary pool of logos and adapt the detection method to this pool. Particular focus will also be given to techniques on extending the detectable brands without having to re-train or finetune the current detection network. A component that can be used to submit training data for new brands to be detected by the service will be implemented.

Deliverable D1.2 will be submitted in August 2019 with an update on all tasks.

## References

(Abdollahian, 2010) G. Abdollahian, C. M. Taskiran, Z. Pizlo and E. J. Delp, "Camera Motion-Based Analysis of User Generated Video," in IEEE Trans. on Multimedia, vol. 12, no. 1, pp. 28-41, Jan. 2010.

(Apostolidis, 2014) E. Apostolidis, V. Mezaris, "Fast Shot Segmentation Combining Global and Local Visual Descriptors", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, May 2014.

(Apostolidis, 2018) K. Apostolidis, E. Apostolidis, V. Mezaris, "A motion-driven approach for fine-grained temporal segmentation of user-generated videos", Proc. 24th Int. Conf. on Multimedia Modeling (MMM2018), Bangkok, Thailand, Feb. 2018

(Aroyo, 2009) L. Aroyo, L. Nixon, and S. Dietze, "Television and the future internet: the NoTube project", In Future Internet Symposium (FIS) 2009

(Aroyo, 2011) L. Aroyo, L. Nixon, and L. Miller, "NoTube: The television experience enhanced by online social and semantic data". In IEEE Intl. Conf. on Consumer Electronics - Berlin, pages 269– 273, 2011.

(Baraldi, 2015a) L. Baraldi, C. Grana, R. Cucchiara, "Shot and Scene Detection via Hierarchical Clustering for Re-using Broadcast Vide". In: Azzopardi G., Petkov N. (eds) Computer Analysis of Images and Patterns 2015. Lecture Notes in Computer Science, vol 9256. Springer, Cham, 2015

(Baraldi, 2015b) L. Baraldi, C. Grana, R. Cucchiara, "A Deep Siamese Network for Scene Detection in Broadcast Videos". In Proceedings of the 23rd ACM international conference on Multimedia (MM '15). ACM, New York, NY, USA, 1199-1202.

(Baraldi, 2016a) L. Baraldi, C. Grana, R. Cucchiara, "Analysis and Re-Use of Videos in Educational Digital Libraries with Automatic Scene Detection". In: Calvanese D., De Nart D., Tasso C. (eds) Digital Libraries on the Move. IRCDL 2015. Communications in Computer and Information Science, vol 612. Springer, Cham, 2016

(Baraldi, 2016b) L. Baraldi, C. Grana, R. Cucchiara, "Recognizing and Presenting the Storytelling Video Structure with Deep Multimodal Networks", in IEEE Trans. on Multimedia , vol.PP, no.99, pp.1-1, 2016

(Bianco, 2017) Simone Bianco, Marco Buzzelli, Davide Mazzini, Raimondo Schettini, "Deep learning for logo recognition." Neurocomputing 245 (2017): 23-30.

(Bouguet, 2001) Bouguet, J.Y., "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm". Intel Corporation, 5(1-10), p.4.

(Fu, 2017) Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, Alexander C. Berg, "DSSD: Deconvolutional single shot detector," arXiv:1701.06659, 2017.

(Erhan, 2014) Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov, "Scalable object detection using deep neural networks," in CVPR, 2014.

(flickrlogos-47, 2017) http://www.multimedia-computing.de/flickrlogos/

(Gonzalez-D𝜈az , 2015) I. Gonzalez-D𝜈az, T. Mart𝜈nez-Cort𝜄s, A. Gallardo-Antol𝜈n, and F. D𝜈az-de Mar𝜈a. 2015. "Temporal Segmentation and Keyframe Selection Methods for User-generated Video Search-based Annotation". Expert Syst. Appl. 42, 1 (Jan. 2015), 488–502.

(Girshick, 2014) Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in CVPR, 2014

(Girshick, 2015) Ross Girshick, "Fast R-CNN" in ICCV, 2015

(Gygli, 2017) M. Gygli, "Ridiculously Fast Shot Boundary Detection with Fully Convolutional Neural Networks". CoRR, abs/1705.08214.

(He, 2017) Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "Mask R-CNN" in ´ ICCV, 2017.

(He, 2015) Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, 2015

(Hoi, 2015) Steven CH Hoi, Xiongwei Wu, Hantang Liu, Yue Wu, Huiqiong Wang, Hui Xue, and Qiang Wu. "Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks." arXiv preprint arXiv:1511.02462 (2015).

(Huang, 2017) Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, Kevin Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors." in CVPR 2017

(Joly, 2009) Alexis Joly and Olivier Buisson. "Logo retrieval with a contrario visual query expansion." In Proceedings of the 17th ACM international conference on Multimedia, pp. 581-584. ACM, 2009.

(Kalantidis, 2011) Yannis Kalantidis, Lluis Garcia Pueyo, Michele Trevisiol, Roelof van Zwol, and Yannis Avrithis. "Scalable triangulation-based logo recognition." In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, p. 20. ACM, 2011.

(Kobilarov, 2009) G. Kobilarov et al., "Media Meets Semantic Web - How the BBC Uses DBpedia and Linked Data to Make Connections". Proceedings of the 6th European Semantic Web Conference (ESWC2009).

(Krizhevsky, 2012) A. Krizhevsky, S. Ilya, and G. Hinton. "Imagenet classification with deep convolutional neural networks". In Advances in Neural Information Processing Systems (NIPS 2012), pages 1097{1105. Curran Associates, Inc., 2012.

(Dai, 2016) Jifeng Dai, Yi Li, Kaiming He, Jian Sun, "R-FCN: Object detection via region-based fully convolutional networks," in NIPS, 2016.

(Lin, 2017) Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, "Feature pyramid networks for object detection," in CVPR, 2017.

(Liu, 2013) C. Liu, D. Wang, J. Zhu, B. Zhang, "Learning a Contextual Multi-Thread Model for Movie/TV Scene Segmentation". Trans. Multi. 15, 4 (June 2013), 884-897.

(Liu, 2016) Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single shot multibox detector," in ECCV, 2016.

(Macedo, 2014) Macedo, P., Cardoso, J., and Pinto, A. M. "Enriching Electronic Programming Guides with Web Data". In Proceeding of the 2nd International Workshop on Linked Media (LiME2014), Crete, Greece, 2014.

(Markatopoulou, 2016), Markatopoulou, Mezaris, and Patras. (2016, October). "Deep Multi-task Learning with Label Correlation Constraint for Video Concept Detection". In Proceedings of the 2016 ACM on Multimedia Conference (pp. 501-505). ACM.

(Markatopoulou, 2018), Markatopoulou, Mezaris, and Patras. "Implicit and Explicit Concept Relations in Deep Neural Networks for Multi-Label Video/Image Annotation." IEEE Transactions on Circuits and Systems for Video Technology (2018).

(Mei, 2013) T. Mei, L.-X. Tang, J. Tang, and X.-S. Hua. 2013. "Near-lossless Semantic Video Summarization and Its Applications to Video Analysis". ACM Trans. Multimedia Comput. Commun. Appl. 9, 3, Article 16 (July 2013).

(Najibi, 2016) Mahyar Najibi, Mohammad Rastegari, Larry S. Davis, "G-CNN: an iterative grid based object detector," in CVPR, 2016.

(Over, 2013) Over, Paul. "Trecvid 2013 - An overview of the goals, tasks, data, evaluation mechanisms and metrics.", 2013.

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

(Redmon, 2016a) Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You only look once: Unified, real-time object detection," in CVPR, 2016.

(Redmon, 2016b) Joseph Redmon, Ali Farhadi "Yolo9000: better, faster, stronger," arXiv:1612.08242, 2016.

(Ren, 2015) Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in NIPS, 2015.

(Romberg, 2011) Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof Van Zwol. "Scalable logo recognition in real-world images." In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, p. 25. ACM, 2011.

(Romberg, 2013) Stefan Romberg, Rainer Lienhart, "Bundle min-hashing for logo recognition." In Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, pp. 113-120. ACM, 2013.

(Rotman, 2016) D. Rotman, D. Porat, G. Ashour, "Robust and Efficient Video Scene Detection Using Optimal Sequential Grouping", 2016 IEEE Int. Symposium on Multimedia (ISM), San Jose, CA, 2016, pp. 275-280

(Rotman, 2017) D. Rotman, D. Porat and G. Ashour, "Robust video scene detection using multimodal fusion of optimally grouped features," 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, 2017, pp. 1-6.

(Rublee, 2011) Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: “ORB: An efficient alternative to SIFT or SURF”. 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564-2571

(Schwing, 2015) A. G., & Urtasun, R. (2015). "Fully connected deep structured networks". arXiv preprint arXiv:1503.02351.

(Seo, 2009) Seo, K.-D., Park, S., Jung, S.-H.: "Wipe scene-change detector based on visual rhythm spectrum". IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 831-838 (2009)

(Shen, 2017) Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, Xiangyang Xue, "DSOD: Learning deeply supervised object detectors from scratch," in ICCV, 2017.

(Sidiropoulos, 2011) Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., and Trancoso, I. (2011). "Temporal video segmentation to scenes using high-level audiovisual features". IEEE Trans. Circuits Syst. Video Technol., 21(8):1163–1177.

(Son 2016) Son J. W., Lee S. Y., Park S. Y., Kim S. J., "Video scene segmentation based on multiview shot representation," 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2016, pp. 381-383.

(Su, 2005) Su, C.-W., Liao, H.-Y.M., Tyan, H.-R., Fan, K.-C., Chen, L.-H.: "A motion-tolerant dissolve detection algorithm". IEEE Transactions on Multimedia, vol. 7, pp.1106-1113 (2005)

(Su, 2017a) Hang Su, Xiatian Zhu, Shaogang Gong, "Deep learning logo detection with data expansion by synthesising context." In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pp. 530-539. IEEE, 2017.

(Su, 2017b) Hang Su, Shaogang Gong, Xiatian Zhu, "Weblogo-2m: Scalable logo detection by deep learning from the web.", IEEE International Conference on Computer Vision, Workshop on Web-scale Vision and Social Media, Venice, Italy, October 2017

(Szegedy, 2016) Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, "Rethinking the inception architecture for computer vision." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826. 2016.

(Teyssou, 2017) D. Teyssou, J.-M. Leung, E. Apostolidis, K. Apostolidis, S. Papadopoulos, M. Zampoglou, O. Papadopoulou, V. Mezaris. 2017. “The InVID Plug-in: Web Video Verification on the Browser”. In Proc. of the 1st Int. Workshop on Multimedia Verification (MuVer '17). ACM, New York, NY, USA, 23-30.

(Tüzkö, 2017) Andras Tüzkö, Christian Herrmann, Daniel Manger, Jürgen Beyerer "Open set logo detection and retrieval." arXiv preprint arXiv:1710.10891 (2017).

(Xu, 2015) J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh. 2015. “Gaze-enabled egocentric video summarization via constrained submodular maximization”. In CVPR. IEEE Computer Society, 2235–2244.

(Yeung, 1998) Yeung, M., Yeo, B.-L., Liu, B.: “Segmentation of video by clustering and graph analysis. Computer Vision and Image Understanding”, vol. 71, no. 1, pp. 94-109 (1998)

(Yoo, Park., 2015) (Yoo, 2015) Donggeun Yoo, Sunggyun Park, Joon-Young Lee, Anthony S. Paek, In So Kweon, "AttentionNet: Aggregating weak directions for accurate object detection," in CVPR, 2015.

(Zhou, 2018) Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Places: A 10 million image database for scene recognition." IEEE transactions on pattern analysis and machine intelligence 40.6 (2018): 1452-1464.