SEVENTH FRAMEWORK PROGRAMME

ForgetIT

www.forgetit-project.eu

ForgetIT

Concise Preservation by Combining Managed Forgetting and Contextualized Remembering

Grant Agreement No. 600826

Deliverable D4.4

Work-package	WP4: Information Consolidation and Concentration
Deliverable	D4.4: Information analysis, consolidation and concentration techniques, and evaluation - Final release.
Deliverable Leader	Vasileios Mezaris (CERTH)
Quality Assessor	Walter Allasia (EURIX)
Dissemination level	PU
Delivery date in Annex I	31-01-2016 (M36)
Actual delivery date	31-01-2016
Revisions	0
Status	Final
Keywords	multidocument summarization, semantic enrichment, feature extraction, concept detection, event detection, image/video quality, image/video aesthetic quality, face detection/clustering, image clustering, image/video summarization, image/video near duplicate detection, data deduplication, condensation, consolidation

Revision History

Version	Major changes	Authors
0.1	Toc Draft	CERTH
0.2	Initial input	All partners
0.3	First update after first intra-WP review	All partners
0.4	First draft for internal review	CERTH
1	Final version after QA review	CERTH

List of Authors

Partner Acronym	Authors
CERTH	V. Solachidis, E. Apostolidis, F. Markatopoulou, D. Galanopoulos, C. Tzelepis, S. Arestis-Chartampilas, A. Pournaras, D. Tastzoglou, V. Mezaris
IBM	D. Chen, D. Harnik, E. Khaitzin
DFKI	B. Eldesouky, H. Maus
USFD	M. Greenwood
TT	A.S. Tan

Table of Contents

Glossary

AH	“ad-hoc”. 67
AP	average precision. 38
ASR	Automatic Speech Recognition. 65
AUC	Area Under ROC Curve. 48
BG	“background”. 67
BoW	Bag of Words. 66
BR	bit rate. 47
C-VQA	Compressed Domain VQA. 51
CCT	Correlated Color Temperature. 44
CIF	Common Intermediate Format. 51
CL	can-not link. 27
CLM	Concept Language Model. 66
CPU	Central Processing Unit. 32
DC	Data Compression. 54
DCNNs	Deep Convolutional Neural Networks. 12
DD	Data Deduplication. 54
DoW	Description of Work. 85
DR	Data Reduction. 54
DVQPM	DCT-based video quality prediction model. 51
ELM	Event Language Model. 65
ESA	Explicit Semantic Analysis. 66
F1	F-measure. 62
FR	Full-Reference. 43
GATE	General Architecture for Text Engineering. 18

GC Geometric Coding. 55

GPU Graphics Processing Unit. 12

GPU-AGSDA Generalised Subclass Discriminant Analysis. 32

IAS Image Analysis Service. 70

IQA Image Quality Assessment. 42

IT-IST Instituto de Telecomunicacoes, Instituto Superior Tecnico. 52

JI Jaccard Index. 62

JM Joint Model. 51

KSVM-iGSU Kernel Support Vector Machine with Isotropic Gaussian Sample Uncertainty. 45

LapPyr Laplacian Pyramid. 51

LBP Local Binary Patterns. 12

LM-DL Language Models with Dirichlet smoothing. 68

LM-JM Language Models with Jelinek-Mercer smoothing. 68

LOD Linked Open Data. 24

LR Logistic Regression. 30

LSVM-GSU Linear Support Vector Machine with Gaussian Sample Uncertainty. 34

LSVMs Linear Support Vector Machines. 30

M Mean Motion Vector Magnitude. 47

MAP Mean Average Precision. 35

MCC Multi-criteria clustering. 58

MED Multimedia Event Detection. 29

ML must-link. 27

MOVIE Motion-based Video Integrity Evaluation index. 51

MS Multimodal Search. 68

MS-SSIM Multiscale Structural Similarity index. 50

MSI Modified Spatial Information. 47

MV	Motion Vector. 43
MXinfAP	Mean Extended Inferred Average Precision. 35
NDD	Near duplicate detection. 53
NLP	Natural Language Processing. 24
NR	No-Reference. 43
OCR	Optical Character Recognition. 65
ORB	Oriented FAST and Rotated BRIEF. 32
PCA	Principal Component Analysis. 32
PCC	Pearson Correlation Coefficient. 50
PHA	Potential-based Hierarchical Agglomerative. 58
PIMO	Personal Information Manager. 21
PoF	Preserve-or-Forget. 70
PRF	Pseudo-Relevance Feedback. 68
PS	“pre-specified”. 67
PV	Preservation Value. 25
QP	Quantization Parameter. 43
RBF	Radial Basis Function. 38
RD-KSVM	Relevance Degree kernel SVM. 34
REST	representational state transfer. 19
ROUGE	Recall-Oriented Understudy for Gisting Evaluation. 19
RR	Reduced-Reference. 43
RTP	Real-Time Transfer Protocol. 47
SCD	Strong Components Discovery. 59
SIFT	Scale Invariant Feature Transform. 25
SIN	Semantic Indexing. 35
SQG	Semantic Query Generator. 68

SROCC Spearman Rank Order Correlation Coefficient. 50

SURF Speeded Up Robust Features. 37

SW-SSIM Saliency Weighed Structural Similarity index. 50

TF.IDF Term Frequency - Inverse Document Frequency. 18

VAQ Video Aesthetic Quality. 43

VAS Video Analysis Service. 70

VLAD Vector of Locally Aggregated Descriptors. 32

VM Virtual Machine. 55

VNDD Video Near duplicate detection. 54

VQM Video Quality Metric. 50

VSIN Video Semantic Indexing. 68

VSNR Visual Signal to Noise Ratio. 50

WASP Web Application Service Provider. 19

WBQM Wavelet Based Quality Metric. 51

WEP Word Embedding Mapping. 68

XML Extensible Markup Language. 71

Z Zero Motion Vector Ratio. 47

Executive summary

This is the final release of ForgetIT techniques for information analysis, consolidation and concentration. Some of the methods presented in this document are extensions of earlier versions developed in the previous years of the project and documented in [Papadopoulou et al., 2014] and [Solachidis et al., 2015], while others are completely new. For the method requirements, the user studies as described in the personal and organizational preservation deliverables [Maus et al., 2013], [Maus and Schwarz, 2014], [Maus et al., 2014] and [Damhuis et al., 2014] as well as in the integration ones ([Gallo et al., 2014a], [Gallo et al., 2015b], [Gallo et al., 2014b] and [Gallo et al., 2015a]) have been taken into account. Furthermore, the results of the proposed techniques' evaluation are also documented.

The problems addressed and the methods that are presented in this deliverable can be classified into text, multimedia (image/video), and joint text and multimedia ones, as follows:

Text processing methods:

- *Multi-document summarization*
- *Text analysis for semantic text composition*

Multimedia processing methods:

- *Image/video face detection and clustering*
- *Image/video annotation*
- *Image/video quality assessment*
- *Image/video near duplicate detection and condensation*

Combined method:

- *Associating text and image/video items*

The detailed presentation of the above methods is followed by a section titled *"Overall WP4 analysis components"*, which illustrates all the software components and services that have been created during the project in WorkPackage 4.

Finally, the Conclusions section summarizes the work performed and outlines how and to what extent the methods described in this deliverable fulfil the success indicators of the five expected outcomes of WP4, as these are described in the DoW of the project.

1 Introduction

This deliverable presents the final release of the ForgetIT text and visual information analysis techniques for condensation and summarization. It is based on the state-of-the-art and requirements analysis that were reported in deliverable [Papadopoulou et al., 2013], and the first and second sets of ForgetIT analysis techniques that were developed, evaluated and implemented as ForgetIT components in [Papadopoulou et al., 2014] and [Solachidis et al., 2015]. It contains improvements and extensions of the methods presented in previous deliverables as well as new methods performing information analysis, consolidation and condensation.

In Section 2, the positioning of the WP4 components described in this document in the overall framework of ForgetIT is illustrated.

Section 3 deals with text analysis. Specifically, it presents an extension of the single document summarization that was presented in [Solachidis et al., 2015], which can now be applied to a set of documents (e.g., a set of journal or diary entries, email conversations), thus performing multi-document summarization.

Section 4 presents the latest development in Seed, the SEmantic EDitor software component developed in the project. In the last version of Seed [Solachidis et al., 2015], frequent text editing resulted in frequent re-analysis of the text, and hence frequent communication between frond-end and back-end that affected Seed's speed and responsiveness. Also, the previous version supported only single-author editing for each entity. These issues were investigated in Year 3 of the project and the approaches utilized to tackle them are presented in Section 4.

Moving on to multimedia analysis, in Section 5 we extend the face detection and clustering methods presented in previous WP4 deliverables. Instead of local binary pattern (LBP)-based features, we employed representations created by using deep convolutional neural networks (DCNNs) in order to increase the method's performance. Furthermore, face detection and clustering are extended in videos, by pre-processing the videos so as to automatically extract video shots and scenes, and apply our face detection and clustering method to the shot keyframes.

Image and video annotation methods are presented in Section 6. The concept detection procedure is improved by replacing the hand-crafted local features with DCNN-based ones. A classifier combination using a cascade approach is also presented, that combines hand-crafted with DCNN-based features. Furthermore, the annotation process is accelerated by developing and employing a new dimensionality reduction technique and by exploiting Graphics Processing Unit (GPU) processing capabilities. Finally, image concept detection is extended to video and is also extended to annotation with more complex labels (often termed event labels).

In Section 7, quality assessment methods for images and videos are presented. In [Papadopoulou et al., 2014] image quality assessment for images was introduced. During the last year of the project we developed a quality assessment method for videos. Additionally, we created methods that assess the aesthetic quality of images and videos, namely methods that quantify the appeal of the image or video.

Section 8 addresses the problem of duplicate and near duplicate detection. The near duplicate detection method that was initially created for images and presented in [Solachidis et al., 2015] is improved and extended for videos as described in this section. Along with this, a more generic deduplication method is presented, that is able to detect duplicate copies of (any kind of) repeated data, saving storage space by reducing the amount of exact duplicates that are stored. Finally, in Section 8, an update of the image collection clustering method originally reported in [Solachidis et al., 2015] is presented.

In Section 9, a method that correlates text with multimedia data is introduced. This method gets as input a media collection and a short textual description, and returns the media collection items sorted according to their relatedness to the text description. To do so, the method of Section 9 relies heavily on the methods presented in previous sections of this document for, e.g., image annotation.

Section 10 presents a technical description of the final release of the overall analysis software and services developed in WP4. Information about the functionality, the usage and the output of each service is reported in detail.

Finally, in Section 11, conclusions are presented along with a discussion on the Success Indicators of WP4 and of how they have been fulfilled.

1.1 Target Audience

Although this document is quite technical, it could be read by multiple audiences having different backgrounds, since it is structured in such a way that each subsection targets different audiences. The first subsection of Sections 3 to 9, entitled “Problem statement”, describes shortly the method and targets a broad audience. It defines the problem to be solved and discusses how solving this problem supports the project’s scenarios.

The next subsection, entitled “ForgetIT approach”, contains the technical description of the presented method and targets more specialized technical audience. In this subsection, the methods’ algorithmic details are presented.

Then, the “Experimental evaluation and comparison” subsection follows, which describes the method evaluation, presenting the employed dataset, the evaluation procedure, the measures that were adopted and the evaluation results.

Finally, Section 10 describes briefly the implemented software of each method and includes usage instructions and examples. This section is most useful for the ones that want to test and experiment with the methods described in this document, and also use the WP4 components in the overall ForgetIT system and applications, or other applications.

2 The big picture

The WP4 methods and components described in this deliverable are the constituent parts of the two master components of the ForgetIT architecture that WP4 contributes to, the *Extractor* and the *Condensator*. These are parts of the Middleware layer of the ForgetIT architecture (Fig. 1).

The *Extractor* takes as input the original media items (e.g., a text, a collection of texts, or a collection of images or videos) and extracts information that is potentially useful not only for the subsequent execution of the *Condensator*, but also for other components or functionalities of the overall ForgetIT system (e.g., search).

The *Condensator* gets as input the *Extractor's* output and, when necessary, also the original media items, in order to generate a summary of the target data (or a subset of these media items). The *Condensator* performs further text, image and video analysis tasks whose results are specific to the condensation process. The final output of the *Condensator* is the condensed (i.e. summarized) media items or collections, or pointers to them.

The *Extractor's* and *Condensator's* output is fed to other ForgetIT components such as the *Contextualizer* and the *Collector/Archiver*.

2.1 Extractor

The *Extractor* subcomponents are illustrated in Fig. 2. In this figure the reader is also able to view the subcomponents of the previous releases which were presented in [Papadopoulou et al., 2014] and [Solachidis et al., 2015] and see which are extensions and improvements of previously introduced subcomponents and which are new. Gray arrows indicate a method update in successive years of the project; method names in black font indicate new or updated methods, and, method names in gray indicate components that were created during a previous year and are still in use.

2.2 Condensator

The subcomponents of the *Condensator* are presented in Fig. 3. They are, similarly to the *Extractor's* subcomponents, extensions and improvements of the ones introduced in [Papadopoulou et al., 2014] and [Solachidis et al., 2015].

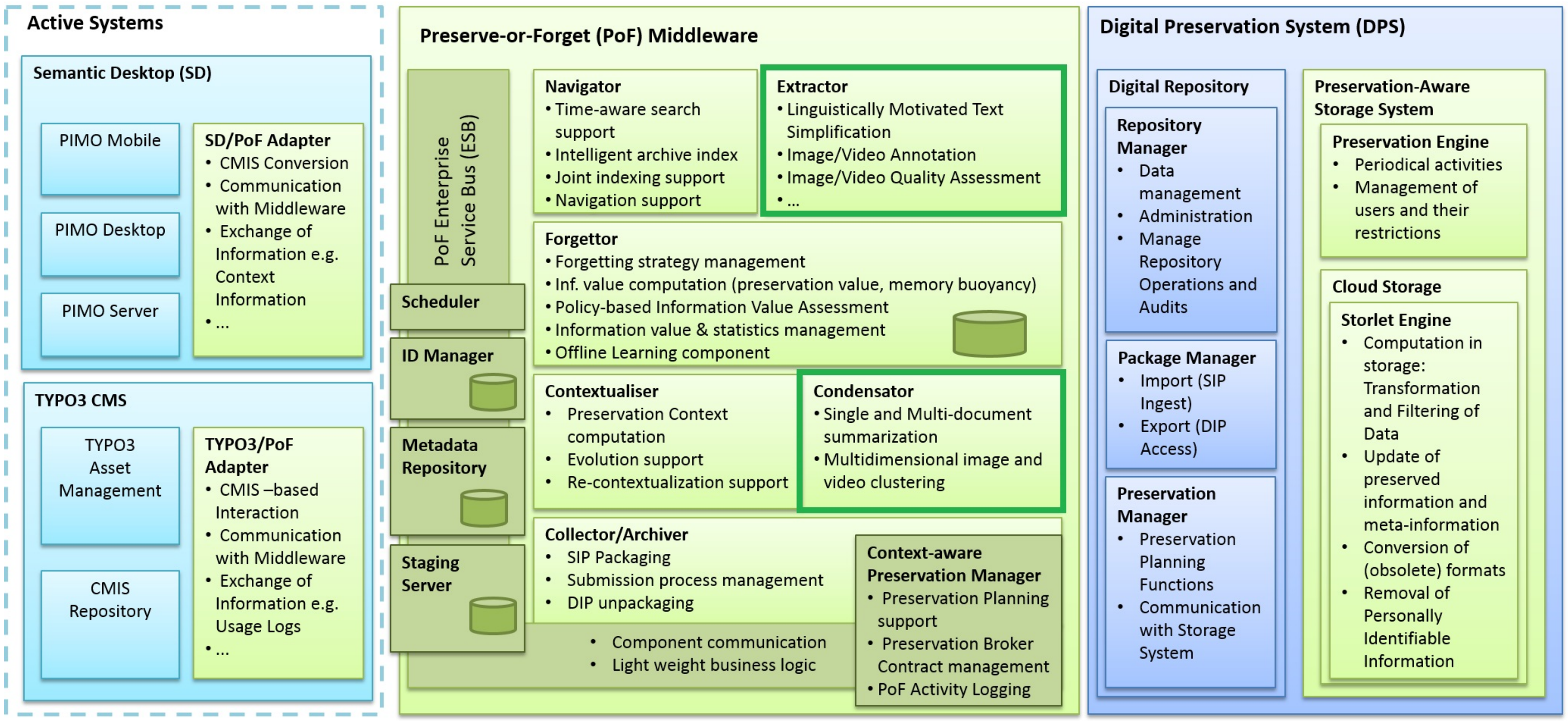

Active Systems
Semantic Desktop (SD)
PIMO Mobile
PIMO Desktop
PIMO Server
SD/PoF Adapter
• CMIS Conversion
• Communication with Middleware
• Exchange of Information e.g. Context Information
• …
TYPO3 CMS
TYPO3 Asset Management
CMIS Repository
TYPO3/PoF Adapter
• CMIS –based Interaction
• Communication with Middleware
• Exchange of Information e.g. Usage Logs
• …
Preserve-or-Forget (PoF) Middleware
PoF Enterprise Service Bus (ESB)
Scheduler
ID Manager
Metadata Repository
Staging Server
Navigator
• Time-aware search support
• Intelligent archive index
• Joint indexing support
• Navigation support
Extractor
• Linguistically Motivated Text Simplification
• Image/Video Annotation
• Image/Video Quality Assessment
• …
Forgettor
• Forgetting strategy management
• Inf. value computation (preservation value, memory buoyancy)
• Policy-based Information Value Assessment
• Information value & statistics management
• Offline Learning component
Contextualiser
• Preservation Context computation
• Evolution support
• Re-contextualization support
Condensator
• Single and Multi-document summarization
• Multidimensional image and video clustering
Collector/Archiver
• SIP Packaging
• Submission process management
• DIP unpackaging
• Component communication
• Light weight business logic
Context-aware Preservation Manager
• Preservation Planning support
• Preservation Broker Contract management
• PoF Activity Logging
Digital Preservation System (DPS)
Digital Repository
Repository Manager
• Data management
• Administration
• Manage Repository Operations and Audits
Package Manager
• Import (SIP Ingest)
• Export (DIP Access)
Preservation Manager
• Preservation Planning Functions
• Communication with Storage System
Preservation-Aware Storage System
Preservation Engine
• Periodical activities
• Management of users and their restrictions
Cloud Storage
Storlet Engine
• Computation in storage: Transformation and Filtering of Data
• Update of preserved information and meta-information
• Conversion of (obsolete) formats
• Removal of Personally Identifiable Information

Figure 1: The *Extractor* and *Condensator* components and their positioning in the Middleware layer of the ForgetIT architecture [Gallo et.al., 2016]

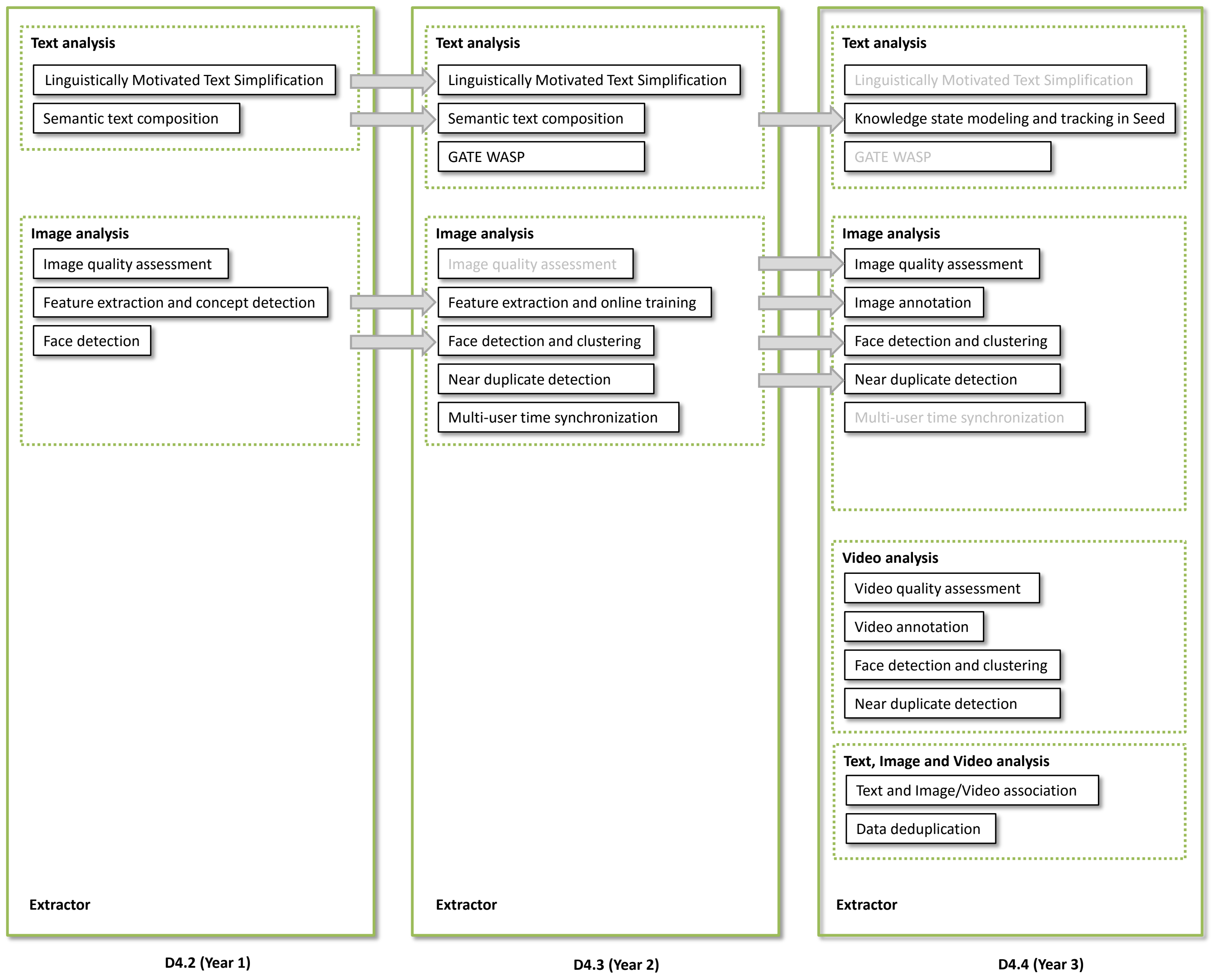

Text analysis
Linguistically Motivated Text Simplification
Semantic text composition
Image analysis
Image quality assessment
Feature extraction and concept detection
Face detection
Extractor
D4.2 (Year 1)
Text analysis
Linguistically Motivated Text Simplification
Semantic text composition
GATE WASP
Image analysis
Image quality assessment
Feature extraction and online training
Face detection and clustering
Near duplicate detection
Multi-user time synchronization
Extractor
D4.3 (Year 2)
Text analysis
Linguistically Motivated Text Simplification
Knowledge state modeling and tracking in Seed
GATE WASP
Image analysis
Image quality assessment
Image annotation
Face detection and clustering
Near duplicate detection
Multi-user time synchronization
Video analysis
Video quality assessment
Video annotation
Face detection and clustering
Near duplicate detection
Text, Image and Video analysis
Text and Image/Video association
Data deduplication
Extractor
D4.4 (Year 3)

Figure 2: The *Extractor's* subcomponents that were presented in D4.2, D4.3 and currently presented in this (D4.4) deliverable. Gray arrows indicate a method update in successive years of the project; method names in black font indicate new or updated methods, and, method names in gray indicate components that were created during a previous year and are still in use.

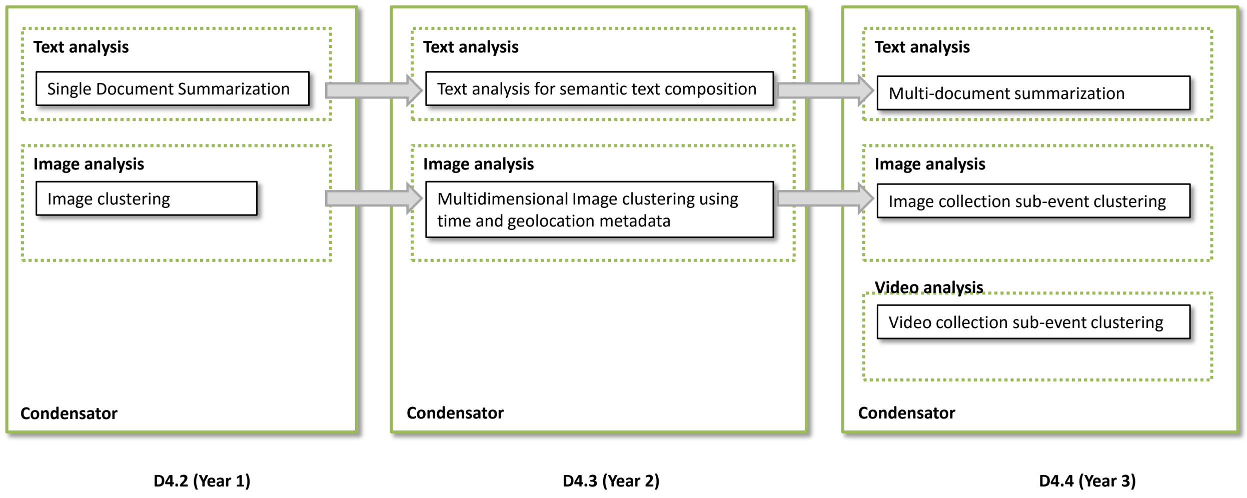

Text analysis
Single Document Summarization
Image analysis
Image clustering
Condensator
Text analysis
Text analysis for semantic text composition
Image analysis
Multidimensional Image clustering using time and geolocation metadata
Condensator
Text analysis
Multi-document summarization
Image analysis
Image collection sub-event clustering
Video analysis
Video collection sub-event clustering
Condensator
D4.2 (Year 1)
D4.3 (Year 2)
D4.4 (Year 3)

Figure 3: The *Condensator's* subcomponents that were presented in D4.2, D4.3 and currently presented in this (D4.4) deliverable. Gray arrows indicate a method update in successive years of the project; method names in black font indicate new or updated methods, and, method names in gray indicate components that were created during a previous year and are still in use.

3 Multi-document summarization

3.1 Problem statement

Previously, in [Solachidis et al., 2015], we reported a number of approaches for reducing the size of textual content. These techniques, however, only considered a single document in isolation. While summarising a single long document can be very useful, documents can, and often are, grouped together. For example, individual journal entries are grouped to form a diary, and an e-mail conversation is not a set of independent documents. In these cases processing a set of documents together provides scope for more effective summarization.

We envisage multi-document summarization being used both as an aid to navigation (a summary of a document collection could be returned by a storlet before the collection is retrieved from the archive), or it could be used to generate entirely new collections of information, as with the diary generation idea being developed within WP9.

3.2 ForgetIT approach

The approach to multi-document summarization we adopted within the ForgetIT project, similarly to the single document approach detailed in [Solachidis et al., 2015], uses the freely available[1] SUMMA toolkit [Saggion, 2008], which adds support for text summarization to GATE [Cunningham et al., 2011] (General Architecture for Text Engineering). SUMMA provides numerous GATE processing resources which can be used to analyse text in order to produce a summary. These include both linguistically motivated resources as well as those that rely on document or corpus statistics to select relevant sentences as part of a summary.

Currently the single document summarization application we are using within ForgetIT performs topic-centred extractive summarization [Saggion et al., 2003, Barzilay and Mckeown, 2005] using a number of SUMMA components. The application uses the well-known cosine similarity measure in conjunction with a term frequency-inverse document frequency (TF.IDF) weighting function [Gerald et al., 1997] to extract the main information-bearing sentences so as to form a summary.

The application consists of two main parts. The first part is simply the single document summarization application used to score all sentences in the document collection. While the details of this application can be found in [Solachidis et al., 2015] the main steps of the application are repeated here for completeness:

- calculates the TF, IDF, and TF.IDF values for each token annotation in the document
- builds a normalized TF.IDF token vector to represent the whole document
- builds a normalized TF.IDF token vectors for each sentence
- assigns a score to each sentence based on its position within the document
- uses the cosine similarity measure to score each sentence based on how similar it is to the whole document
- uses the cosine similarity measure to score each sentence based on how similar it is to the first sentence (in general the first sentence in a document contains pertinent information and is highly likely to be included in a single document based summary)

Once each document has been processed independently, the application enters a second phase where the collection of documents is treated as a whole. The approach to multi-document summarization adopted here is based around the calculation of the centroid vector of the set of documents being summarized. This approach has previously been shown to work well across a range of document types [Radev et al., 2004, Saggion and Gaizauskas, 2004]. Once calculated, the centroid vector allows the algorithm to determine which words are statistically significant across the document set, and each text unit (in our case, sentences) which could be added to the summary is scored based on its similarity, calculated using cosine similarity, to the centroid.

3.3 Experimental evaluation and comparison

As outlined earlier in this section we envisaged multi-document summarization playing an important role in both generating new summary documents and as an aid to navigation

[1] `http://www.taln.upf.edu/pages/summa.upf/` (Accessed date: 31-01-2016)

within ForgetIT. In order to evaluate multi-document summarization, we tested it on a widely-used public dataset the DUC-2004 Task 2 dataset[2]. This dataset consists of 50 sets of documents, each of which is focused on a different topic, event or timespan. Each set contains on average 10 documents from the newswire services of the Associated Press and the New York Times. Short model summaries (665 characters or less) of each set were constructed by up to four different assessors working independently, for use in evaluation.

The systems entered into DUC-2004 were evaluated using the ROUGE metric [Lin, 2004], which is also used here for evaluation to aid in comparisons between our work and other approaches. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation and is actually a set of evaluation measures. For our purposes we are focusing purely on ROUGE-N which is a recall focused n-gram metric that evaluates an automatically produced summary against a set of model reference summaries. The size of the n-grams used for evaluation can vary, but often simple uni-grams (giving a ROUGE-1 metric) are used. DUC-2004 reported ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4, although the varying n-gram size did not overly affect the comparative performance of the systems evaluated.

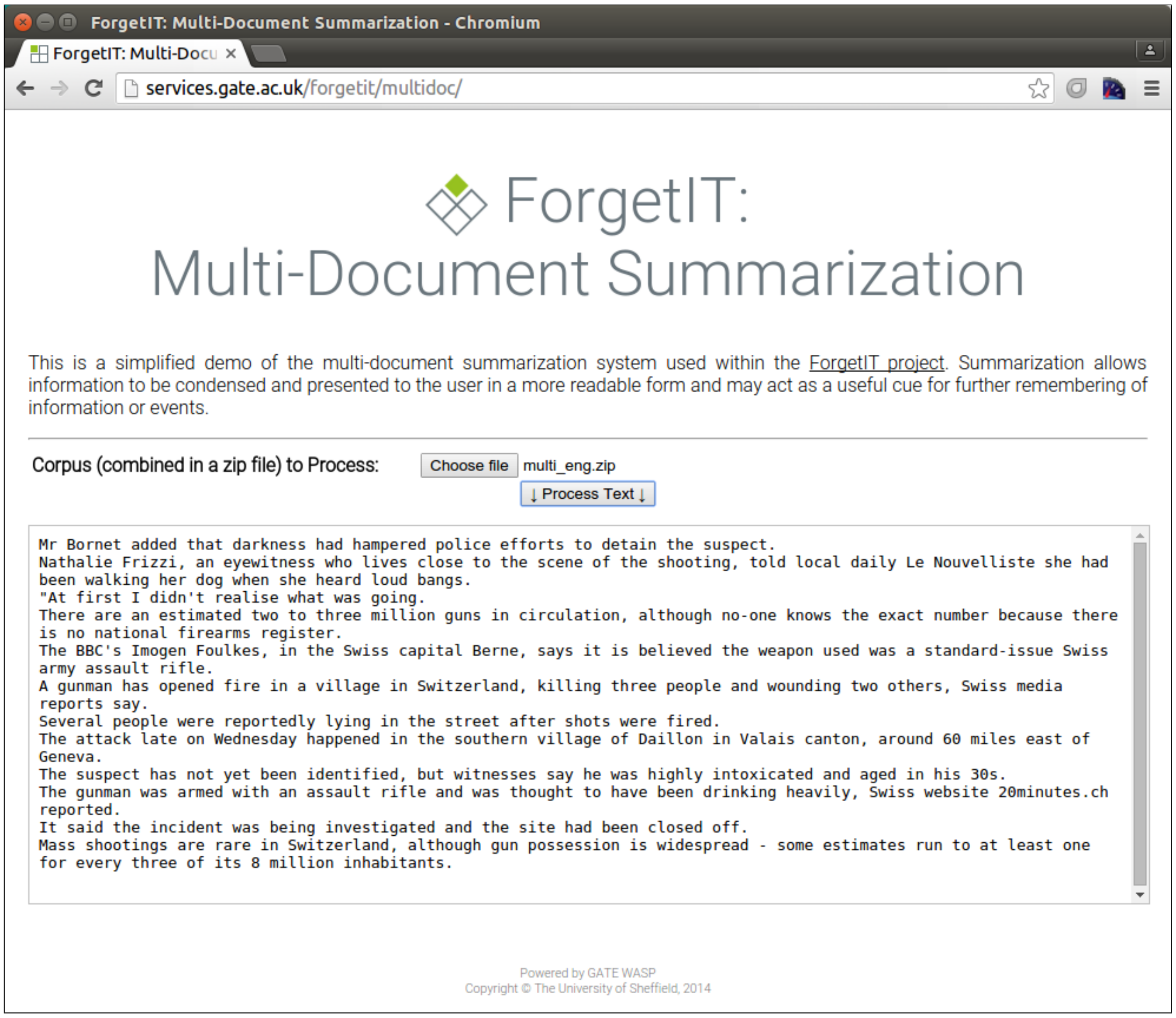
Evaluating our approach using ROUGE-1 across the 50 document sets gives a result of 0.2948. This compares with the best and worst recorded at DUC-2004 of 0.3822 and 0.2419 respectively. This suggests that while our current implementation is not an improvement on the current state-of-the-art, it's performance should be acceptable. Due to the modular nature of our approach upon integration of the component within a use case tool, further tuning and adaption will be possible. While such tuning might not result in higher ROGUE scores on a benchmark dataset, it should improve performance over the intended task.

3.4 Software implementation

As mentioned previously the approach we adopted for multi-document summarization is implemented as a GATE application. This makes it easy to embed it within other applications either directly, via the ForgetIT middleware, within an archive as a storlet, or in a more generic fashion as a RESTful (representational state transfer) web service. Each of these is straightforward to provide as we have previously developed, and reported, generic components for hosting GATE applications in all four ways.

Currently the approach has been made available as a web service, with an interactive demo (Fig. 4), using GATE WASP [Solachidis et al., 2015] (Web Application Service Provider) to allow for easy experimentation and integration within the use case tools. The service can be found at `http://services.gate.ac.uk/forgetit/multidoc/` and its technical details are listed in Table 1.

[2]`http://duc.nist.gov/duc2004/tasks.html` (Accessed date: 31-01-2016)


ForgetIT: Multi-Document Summarization - Chromium
ForgetIT: Multi-Docu
services.gate.ac.uk/forgetit/multidoc/
ForgetIT:
Multi-Document Summarization

This is a simplified demo of the multi-document summarization system used within the ForgetIT project. Summarization allows information to be condensed and presented to the user in a more readable form and may act as a useful cue for further remembering of information or events.

Corpus (combined in a zip file) to Process: Choose file multi_eng.zip

↓ Process Text ↓

```
Mr Bornet added that darkness had hampered police efforts to detain the suspect.
Nathalie Frizzi, an eyewitness who lives close to the scene of the shooting, told local daily Le Nouvelliste she had
been walking her dog when she heard loud bangs.
"At first I didn't realise what was going.
There are an estimated two to three million guns in circulation, although no-one knows the exact number because there
is no national firearms register.
The BBC's Imogen Foulkes, in the Swiss capital Berne, says it is believed the weapon used was a standard-issue Swiss
army assault rifle.
A gunman has opened fire in a village in Switzerland, killing three people and wounding two others, Swiss media
reports say.
Several people were reportedly lying in the street after shots were fired.
The attack late on Wednesday happened in the southern village of Daillon in Valais canton, around 60 miles east of
Geneva.
The suspect has not yet been identified, but witnesses say he was highly intoxicated and aged in his 30s.
The gunman was armed with an assault rifle and was thought to have been drinking heavily, Swiss website 20minutes.ch
reported.
It said the incident was being investigated and the site had been closed off.
Mass shootings are rare in Switzerland, although gun possession is widespread - some estimates run to at least one
for every three of its 8 million inhabitants.
```

Powered by GATE WASP
Copyright © The University of Sheffield, 2014

Figure 4: Demo interface of the ForgetIT Multi-document Summarization service, for easy experimentation

3.5 Conclusions

Summarization can be compared with the way in which human memories preserve pertinent details but allow extraneous information to fade over time and as such these approaches are a perfect fit with the aims of the ForgetIT project. Within the project we have developed both single and multi-document summarization approaches (see [Solachidis et al., 2015] for details on the single document approach) and which have been made available in numerous ways (storlet, as a middleware component, and a RESTful web service) to aid their usage both within the ForgetIT use case and potential outside the project.

Table 1: Software technical details of ForgetIT Multi Document Summarization

Functional description	Multi-document summarization
Input	text
Output	text
Limitations	configuration options not currently exposed
Language/technologies	Java, GATE, and SUMMA
Hardware Requirements	Any hardware with Java support and sufficient RAM
OS Requirements	Any OS with Java support

4 Knowledge state modeling and tracking in Seed

4.1 Problem statement

In this deliverable we focus on challenges that affect the performance of Semantic Editor (Seed) component, which was initially introduced in [Papadopoulou et al., 2014] and further extended in [Solachidis et al., 2015], and in some cases hinder certain semantic text composition user scenarios.

- Scenarios for semantic text composition involve frequent interaction with the knowledge present in the text in the form of annotations. This interaction is a result of frequent editing of text. In Seed, changes in the text require re-analyzing it to discover potential modified entities, new entities or mentions thereof. This in turn causes frequent communication between Seed's front-end and back-end. For long texts containing a lot of entities, this affects the speed of Seed and its responsiveness.
- In ForgetIT, the main use case of semantic text composition involves utilizing Seed as the main text editor when interacting with Personal Information Manager (PIMO). Therefore, texts edited in Seed are themselves modeled as entities. So far, interaction with texts in PIMO best serves single-author scenarios. For multiple authors working on the same text, there is potential for improving knowledge consumption and transfer beyond the reuse of semantic annotations embedded in text.

In order to tackle both of the previously mentioned points, we have been experimenting with ways of modeling and tracking changes to knowledge in the text.

4.2 ForgetIT approach

Both of the problems presented earlier can be addressed by realizing a model for tracking the state of knowledge in text being authored in Seed.

Fig. 5 from [Eldesouky et al., 2015] shows a logical representation of processes taking place during a typical text authoring session in Seed. In the front-end, the content authoring process takes place, while in the back-end, the content analysis process is performed.

The content analysis process in the back-end, is stateless. This means that it does not keep track of the state of the extracted knowledge across multiple iterations of analysis. For example if a user changes the text in Seed, the analysis process is initiated and results at its end in suggested annotations for the text. Upon a later change of the text, the process is carried out again including all of the analysis steps on all of the text.

By incorporating a state representation of the extracted annotations, we can reuse annotations generated in the previous query instead of repeating redundant steps. We can then selectively analyze only the changed or newly introduced parts of the text.

Additionally, we can benefit from the existence of a knowledge state representation on the back-end in the case of multiple authors. By saving the state of the knowledge content along with the textual content after authoring by one user, we can reuse that state for other authors, thereby improving knowledge transfer between multiple authors. The saved knowledge state representation can be utilized to provide multiple views upon the same content as seen by different authors.

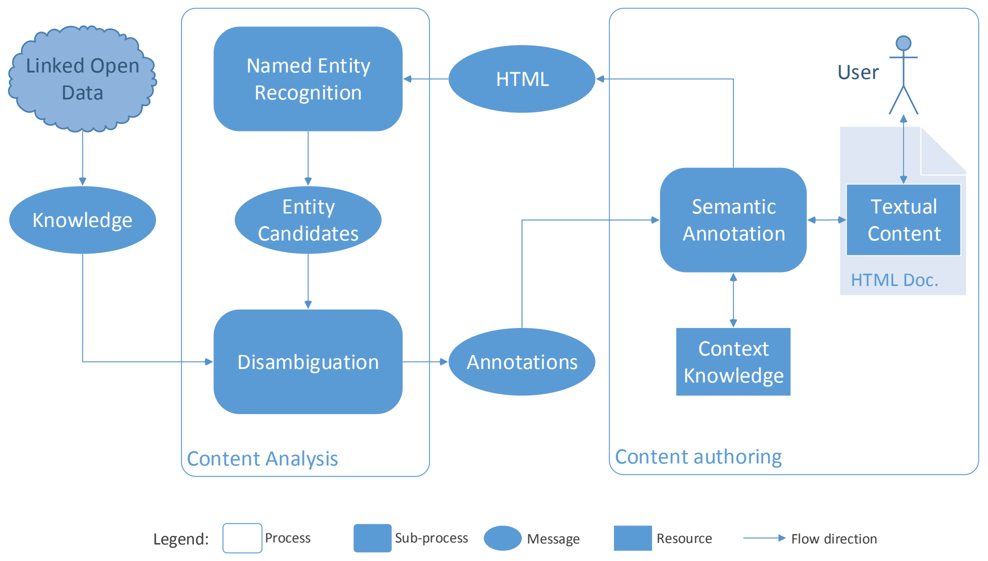


Figure 5: Simplified overview of analysis in Seed as a process diagram

We realized this by:

1. Implementing a mechanism for uniquely identifying distinct authoring sessions.

2. Implementing a model for tracking the state of entities in the text and relations among them.

4.2.1 Advances in comparison to the previous version

The previous version of Seed had a stateless back-end, which didn't support storing or tracking a shared model for knowledge extracted from texts being authored in Seed's front-end instances. On the other hand, the realization of the shared state model on the back-end of the current version adds the following benefits:

1. Generated annotations in one query to the back-end can be reused later, instead of repeated extraction and annotation.
2. Implementing collaborative authoring/editing at the same time is feasible now.
3. Interactions with annotations in the same text by authors on different instances of the front-end can be compared and aggregated.

4.3 Experimental evaluation and comparison

In order to assess the benefits of implementing a shared state model on the back-end, we measured the percentage of requests to the server, which are spared due to annotation reuse. We tracked outgoing requests for disambiguating entity candidates in texts with an average length of 460 words. Each text was loaded in the front-end and had at least 4 interactions (e.g., insertion, deletion or update). Over the course of a 24-hour experiment, the average hit rate in the request cache on the back-end was 70.38%, whereas the standard deviation thereof was 18.71%. This preliminarily shows that the performance of the back-end benefits from sharing a state model across requests and multiple users. With the ongoing utilization of the shared state model in the front-end, its benefit can be further assessed.

4.4 Software implementation

There are several technical alternatives on which we could base our state representation in Seed's back-end. Following are a few alternatives:

- **HTML Hidden fields** Utilizing unique hidden fields in the HTML markup to distinguish between different users.
- **URL Rewriting** Appending a session identifier parameter with every request and response to keep track of an authoring session. This is very tedious and error/prone for our case.
- **Cookies** Small pieces of information that are sent by web servers in response headers and get stored in the browser. This was not suitable due to the nature of Seed, being a component embedded in surrounding web interfaces, which have their own cookies.

- **Session Management APIs** Built on top of one or more of the above methods to realize a custom server-based session tracking mechanism.

We chose the fourth alternative to avoid limitations of the first three. So, we built on the Java Servlet session management API using extra parameters to extend the concept of a per-user-session. Using a combination of user identification tokens in PIMO and session identification tokens from the front-end as well as the back-end, we identify individual text authoring sessions, which could be working on the same text or on different ones, by the same user or different ones.

Afterwards, we identified relevant information to be stored in the knowledge state representation in the back-end as follows:

- The first obvious piece of information that would be stored in the session is entity candidates resulting from the NLP-based (Natural Language Processing) analysis. Those can be reused when needed to spare the repeated analysis of unchanged parts of the text. It also spares repeated searches for entity candidates in the unchanged text in Linked Open Data (LOD) sources.

- In addition to storing entity candidates, we also stored the result of their disambiguation. We take into consideration the user's modifications, rejections or additions, too.

A summary of the technical details of the software is shown in Table 2.

Table 2: Software technical details of Seed's back-end

Functional description	Added features to existing back-end of Seed
Input	HTML-based rich text
Output	In-memory / serialized representation of knowledge about entities in the text
Limitations	N/A
Language/technologies	HTML, JavaScript, Java
Hardware Requirements	N/A
OS Requirements	Any OS with Java support

4.5 Conclusions

We have implemented a mechanism for knowledge state tracking and update in Seed. This not only helps overcome limitations of the existing implementation, but also allows for extending its functionality by implementing new features such as state sharing and reuse.

5 Image/video face detection and clustering

5.1 Problem statement

Person existence in an image or a video is quite important for estimating the Preservation Value (PV) of it. Many parameters can affect the PV such as number of faces, location of faces in the image/video, relative size of faces. This information can be extracted with the use of a face detector. Furthermore, the frequency of the appearance of a specific person in the collection (for example the dominant face) is an important piece of information. Such information can be extracted by clustering the detected faces using a face clustering method.

In this deliverable we present the updated editions of the face detection and face clustering methods introduced in [Solachidis et al., 2015]. In the previous editions, several low-level face representations such as SIFT [Over et al., 2014], LBP [Ahonen et al., 2006], Gabor [Liu and Wechsler, 2002] features or combinations of them [Wolf et al., 2010] were employed in order to represent faces. However, while these representations are based on heuristics, deep learning networks can learn and create data-adaptive features from face samples, thus generating more effective representations. Then, these representations are used as similarity features for face clustering. Finally, the methods for image face detection and clustering is upgraded to videos by including a pre-processing step where the videos are segmented into shot, and then video keyframes are extracted from each of them is treated as an image.

5.2 ForgetIT approach

5.2.1 Image face detection

While the previous version of face detection [Solachidis et al., 2015] employs facial features (eyes, nose or mouth) the current one omits the calculation of these features. We have noticed that by employing facial landmarks, face detection performance is slightly improved but at the expense of significantly higher computational cost; as a result, we decided to omit them. Furthermore, the number of face detectors used is limited from four to three by removing the “haarcascade_frontalface_default” Haar Cascade detector. This detector detects slightly more faces than the other detectors, but it also detects much more non-facial regions, leading to low precision. As a result, the proposed method accepts a detected face if the number of skin-color pixels is above a threshold or if the facial region is detected by all (three) face detectors.

Table 3: DCNN network

Layer name	Type	Filter size/stride	Output size	Type/size/stride
Conv1	Convolution	5x5/1	20	
Pool1	Pooling	Max		Max/2/2
Conv2	Convolution	5x5/1	20	
Pool2	Pooling	Max		Max/2/2
Conv3	Convolution	5x5/1	50	
Pool3	Pooling	Max		Max/2/2
Conv4	Convolution	5x5/1	50	
Pool4	Pooling	Max		Max/2/2
Ip1	innerproduct		500	
ReLU1	ReLU			
Ip2	innerproduct		10	

5.2.2 Image face clustering

The latest face clustering version is based on the one presented in the previous WP4 deliverable [Solachidis et al., 2015]. The method outline is the same as presented in Fig.14 of [Solachidis et al., 2015]. The main difference is that the images are not represented by features extracted using local binary patterns, but instead a DCNN is employed.

A “Siamese” architecture [Bromley et al., 1993] is employed according to which a network is replicated twice and a pair of faces is fed as input. Then, the network is trained based on the distance of the feature representations of the face pair. If the pair face belongs to the same person, then the distance should be below a threshold; otherwise it should be above. This threshold value is arbitrarily set (in our case is set equal to 1).

The parameters of the network used are illustrated in Table 3. It consists of convolutional layers followed by max pooling ones. As a result, each face is represented by a 10-dimensional vector and for each face pair the Euclidean distance of their vector representations is calculated. The *loss* that is employed is the Contrastive Loss [Hadsell et al., 2006]. Let F_1, F_2 be a pair of facial images given as input to the algorithm and $\vec{V_1}, \vec{V_2}$ the output of the network for each face respectively. Let also Y denote the pair label, namely let $Y = 0/1$ if F_1, F_2 are facial instances of the same person/ different people. Contrastive Loss is given by

$$L_C = \frac{1}{2}\left(Y \cdot D + (1-Y)\max(m-D, 0)\right)$$

where D is the square of the Euclidean distance $(||\cdot||_2)$ of $\vec{V_1}, \vec{V_2}$ $(D = ||\vec{V_1} - \vec{V_2}||_2^2)$ and $m > 0$ is a distance margin which, in other words, defines a sphere of radius m around $\vec{V}$ which should contain only similar pairs.

The network has been initially trained in a face dataset (more details about this dataset are presented in Section 5.3). Similarly to the approach presented in [Solachidis et al., 2015], side information extracted from the hair and costume regions adjacent to the face are employed in order to increase the clustering accuracy. Thus, after the initial training, the network weights are fine-tuned by re-training the network using the must-link (ML) and can-not link (CL) constrains. Due to the small number of the ML and CL pairs, the fine-tuning procedure is fast and doesn't increase significantly the overall execution time.

5.2.3 Video face detection and clustering

Before performing face detection and clustering on video, we segment it temporally. Initially, a shot segmentation algorithm is applied that detects the video shot changes [Apostolidis and Mezaris, 2014]. Then, using the extracted shots, a scene detection algorithm [Sidiropoulos et al., 2011] estimates the video scenes (which are groups of consecutive shots depicting the same event or dealing with the same theme). Furthermore, for each shot, the temporally-middle frame is selected and extracted. Besides face detection and clustering, these representative keyframes will be used in several subsequent analysis tasks, (e.g., video annotation), as explained in the sequel. Several image analysis methods will be applied on this frame, that acts as shot representative, and their output will characterize the entire shot. This keyframe-based approach was selected due to the large size of video data and, at the same time, the need for fast algorithms that are required for the ForgetIT applications (e.g., PIMO).

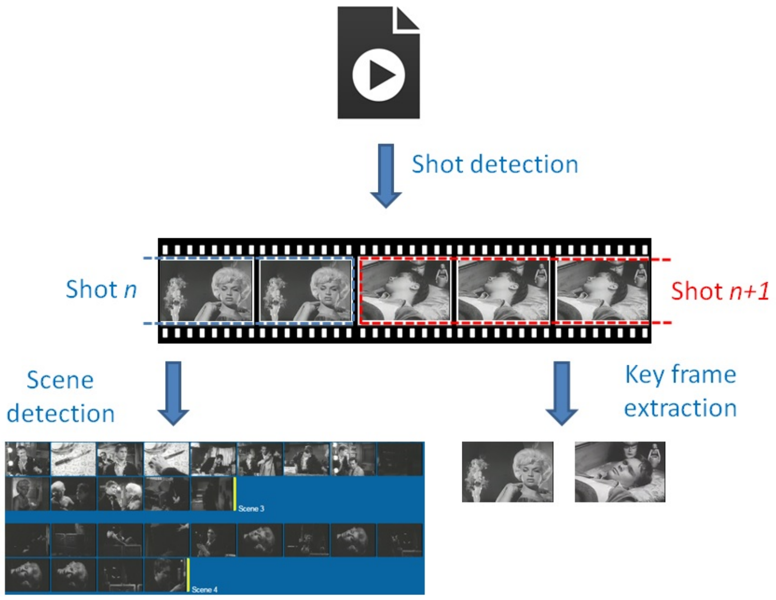


Figure 6: Video temporal segmentation

Following video temporal segmentation and keyframe extraction, each keyframe is treated as image and fed into the image face detection algorithm. For video face clustering, the bunch of the extracted face regions are clustered using the time information coming from examining the video structure and the relative ordering of the keyframes in the video, instead of the "time taken" information used for the images. Additionally, for the faces that belong to the same scene, costume similarity is also employed, based on the assumption that within a scene the costumes of the people/actors do not change.

5.2.4 Advances in comparison to the previous version

The previous version of the method presented in [Solachidis et al., 2015] used local binary patterns for feature extraction, instead of a DCNN that is employed in the current version. Furthermore, the previous method for still images was extended in order to also support face detection and clustering in video keyframes.

5.3 Experimental evaluation and comparison

The dataset used for training is the FaceScrub dataset [Ng and Winkler, 2014]. It comprises a total of 107,818 face images of 530 male and female celebrities, with about 200 images per person. However, since the actual images are not provided but only the images URLs are given, the downloaded dataset contain less images (88,191 images of 530 people).

The Caffe library [Jia et al., 2014] was employed for setting up and running the neural network. The method was tested on the same dataset used in [Solachidis et al., 2015], that is, a collection gathered from partners of the ForgetIT project that consists of 484 images from the vacation of a group of five people.

The metrics used for evaluating the clustering method are the ones used in previous deliverable (see p. 47-48 in [Solachidis et al., 2015]), namely $F_{measure}$ (calculated from precision and recall) and compression ratio. The updated method attained higher $F_{measure}$ than the previous one (0.894 vs 0.855), having similar compression ratio.

5.4 Software implementation

The face detection and clustering methods for still images and video collections are already parts of the overall image/video analysis web service described in detail in Section 10. As explained in previous sub-sections, in the video case, keyframe extraction is initially performed using a video shot segmentation method and then, face detection and clustering methods are applied on the extracted video keyframes. Both face detection and clustering methods are implemented using C++. Furthermore, in network training the benefits in terms of execution time are exploited if CUDA capable[3] NVIDIA graphic card is available while parallel processing software manages the fast face detection on images and keyframes. In Table 4 the software technical details of the methods are listed.

[3]`http://www.nvidia.com/object/cuda_home_new.html` (Accessed date: 31-01-2016)

Table 4: Software technical details for the ForgetIT face detection and face clustering methods (for still images and videos)

Functional description	Face detection and clustering
Input	An image or video collection
Output	Coordinates of bounding boxes where a face is located and clusters of similar faces
Limitations	N/A
Language/technologies	C++
Hardware Requirements	CUDA capable NVIDIA graphic card with compute capability 3.0 or greater
OS Requirements	Windows

5.5 Conclusions

In this section an update of the face detection and clustering methods introduced in [Papadopoulou et al., 2014] and [Solachidis et al., 2015] is presented. Face detection and clustering can be used in order to find the dominant or main persons of an image or video collection. The results of face detection and clustering are used in image collection summarization (e.g., to keep the images that contain a specific person or dominant persons that appear in large clusters), mainly in the personal preservation scenario.

6 Image/video annotation

6.1 Problem statement

Image/video annotation refers to the task of assigning semantic labels to images, videos, or video-fragments (e.g., shots) based on a predefined list of labels [Snoek and Worring, 2009]. Images are commonly annotated with concept labels (aka keywords or tags) that refer to objects (e.g., “car” and “chair”), activities (e.g., “running” and “dancing”), scenes (e.g., “hills” and “beach”), etc. Video annotation can be performed at the fragment level where, video is initially segmented into meaningful fragments, called shots; one or more keyframes are extracted from each shot and these keyframes are subsequently annotated with concept labels. This process is known as concept-based video annotation. The entire video (or fragments of it) can additionally be annotated with complex event labels, which is known as video event detection. Indicative examples of complex events are: “Attempting a bike trick”, “Cleaning an appliance”, or “Beekeeping”. The latter have been retrieved from the Multimedia Event Detection (MED) task of the TRECVID benchmarking activity [Over et al., 2014]. Automatically understanding the content of images/videos in order to annotate them is a challenging and intensively investigated problem. In the previous deliverables ([Papadopoulou et al., 2014], [Solachidis et al., 2015]) we followed a typical two-step label detection procedure for image annotation: 1. Extraction of hand-crafted features; referring to interest point detection, local feature extraction and feature encoding. 2. Classification; referring to the process of building classifiers that learn the associations between the extracted features and semantic labels. A similar process can be followed for video annotation. In this case, video sequences are firstly segmented into one or more characteristic keyframes, the keyframes can subsequently been treated as images.

A recent trend in video/image annotation is to learn features directly from the raw keyframe pixels using deep convolutional neural networks (DCNNs). 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 and also as generators of image/video-fragment features (descriptors). Several DCNN software libraries are available in the literature, e.g., Caffe [Jia et al., 2014], MatConvNet [Vedaldi and Lenc, 2015], and different DCNN architectures have been proposed, e.g., CaffeNet [Krizhevsky et al., 2012], GoogLeNet [Szegedy et al., 2014], VGG ConvNet [Simonyan and Zisserman, 2014]. For learning the associations between the image/video representations and semantic labels, algorithms reported in [Solachidis et al., 2015], such as Linear Support Vector Machines (LSVMs), Logistic Regression (LR) and kernel SVMs (preferring the histogram intersection kernel), still represent the state-of-the-art.

The current deliverable describes the extensions regarding the work presented in deliverable [Solachidis et al., 2015]; there extensions can be summed up by the following points:

- The introduction of DCNN-based features for image/video representation.
- A video annotation system for concept-based detection at video-fragment level.
- A dimensionality reduction method that improves the video annotation accuracy [Sidiropoulos et al., 2014, Arestis-Chartampilas et al., 2015].
- A cascade algorithm [Markatopoulou et al., 2015b] for combining the DCNN-based features with the hand-crafted features of [Solachidis et al., 2015].
- A video annotation system for complex event detection on the entire video.

6.2 ForgetIT approach

In order to improve the performance of the image annotation approach of [Solachidis et al., 2015] we developed methods in the following directions: In the image annotation pipeline we improve the visual information analysis pipeline by replacing hand-crafted features with DCNN-based features. In addition, we introduce two different strategies for dealing separately with the image retrieval and the image annotation problem. The experimental results show a significant increase to the accuracy of the image annotation system compared to [Solachidis et al., 2015]. Furthermore, we extend the visual analysis with video annotation approaches. Specifically, 1) We introduce a new video annotation pipeline in order to analyze videos fragments. 2) We improve the video-fragment annotation accuracy by developing a dimensionality reduction technique. 3) We optimally combine many classifiers trained on different types of features using a cascade architecture. 4) We annotate video sequences not only with simple concept labels but also with complex event labels.

6.2.1 DCNN features for concept-based annotation of images

For the visual representation of images we replace the hand-crafted features, used in visual concept detection method of [Solachidis et al., 2015], with DCNN-based features. Specifically, we use the pre-trained 22-layer GoogLeNet network [Szegedy et al., 2014]. Two different directions are followed with respect to the image retrieval and the image annotation problem that consequently result to two different sets of concept detection scores. With respect to the image annotation problem, GoogLeNet is used as a standalone classifier, where the direct output of the network that corresponds to the final class label prediction for 1000 ImageNet ISLVRC categories constitutes the returned set of scores. With respect to the image retrieval problem we use the TRECVID Semantic Indexing (SIN) 2013 development dataset [Over et al., 2013] that consists of 800 hours of video. We apply the GoogLeNet on the TRECVID keyframes and we use as a feature each of the last fully connected layers that correspond to the three classifiers of GoogLeNet. Then, each of the three 1000-element feature vectors is used separately to represent each TRECVID keyframe. We train concept detectors for 183 concepts, which constitute a subset of the 346 concepts, used in our previous work in ForgetIT [Solachidis et al., 2015]. This subset of concepts was carefully selected according to the ForgetIT needs. These feature vectors serve as input for training SVMs, separately for each target concept, resulting in 3×183 concept detectors. Finally, the method of online training, which was introduced in [Solachidis et al., 2015], for automatic collection of positive training data from the Web [Papadopoulou and Mezaris, 2015] was also updated with the DCNN-based features in order to populate the concept list with more user-defined concepts.

6.2.2 Extension of concept-based annotation to video fragments

Figure 7 presents the video annotation pipeline at the video-fragment level. Particularly, a video is given as input to the video decomposition component (Section 5.2.3), it is segmented into shots, and one keyframe is extracted from each shot. Then, each keyframe is treated as an image and a procedure similar to that of the concept-based image annotation pipeline (Section 6.2.1) is followed. The difference is that a modification of the GoogLeNet [Szegedy et al., 2014] is used for video-fragment representation (as explained below). Similarly to image annotation, two sets of scores are extracted, one optimized for the retrieval and one for the annotation problem.

Specifically for the video annotation problem we fine-tune GoogLeNet [Szegedy et al., 2014] on the development set of the TRECVID SIN 2013 positive samples for 323 concepts. Specifically, the output classification layer for the 1000 ImageNet ISLVRC categories of the pre-trained GoogLeNet network is discarded and replaced with a 1024-dimensional fully connected layer, along with RELU and dropout layers, followed by a 323-dimensional fully connected layer, which constitutes the classification layer for the 323 concepts. Due to the fact that the GoogLeNet has three output classification layers, the above procedure is repeated for all of them. This approach extends each classifier of the GoogLeNet [Szegedy et al., 2014] by one layer. We use the output of the last classification layer of the GoogLeNet fine-tuned network (*Loss3-323*) as a feature vector to train SVMs separately for each concept. Furthermore, we use the output of the extended hidden layer corresponding to the first (*Loss1-fci*), the second (*Loss2-fci*) and the third classifier (*Loss3-fci*) of the GoogLeNet fine-tuned network, which results to a 1024-element vector for each one of the layers. Each of these features is also used as input to SVMs. In total, 4 SVMs per concept are trained and then fused (in terms of arithmetic mean) for each target concept. With respect to the indexing problem, in addition to the DCNN-based features we also use the three local descriptors that were used in [Solachidis et al., 2015] for image annotation (ORB, RGB-ORB, OpponentORB). All the local descriptors are compacted using Principal Component Analysis (PCA) and are subsequently aggregated using the Vector of Locally Aggregated Descriptors (VLAD) encoding. The VLAD vectors are reduced to 4000 dimensions. Thus, in total, 7 SVMs per concept are trained and then fused (in terms of arithmetic mean) for each target concept.

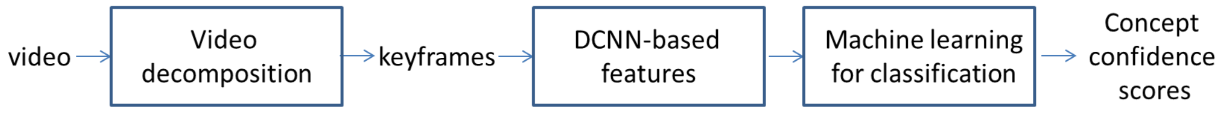

video
Video decomposition
keyframes
DCNN-based features
Machine learning for classification
Concept confidence scores

Figure 7: Concept-based annotation of video fragments

6.2.3 GPU-AGSDA method for improved concept-based annotation

GPU-AGSDA is a Generalised Subclass Discriminant Analysis method [Arestis-Chartampilas et al., 2015], accelerated by utilizing the framework proposed in [Gkalelis and Mezaris, 2015] and by the use of GPU hardware. It is designed with the purpose of dimensionality reduction for classification. Its strength is that it can greatly reduce the feature vector dimensionality (possibly to just 2 or 3 elements per vector) and when combined with linear SVMs, it can still achieve kernel SVM classification accuracy in orders-of-magnitude less time than a state-of-the-art Central Processing Unit (CPU)-only Kernel SVM method would require. In this deliverable we exploit the benefits of GPU-AGSDA by reducing the dimensionality of the features extracted by the local descriptors and the DCNN-based feature vectors of our video annotation pipeline presented in subsection 6.2.2. Experimental results show improvement for the reduced feature vectors over the full feature vectors (Table 6).

6.2.4 Cascade architecture for combining multiple features

Image/video annotation requires for each concept to train and combine many base classifiers (SVMs) that have been trained on different types of features (e.g., DCNN-based, ORB, RGB-ORB etc.). This process of combining many base classifiers instead of using a single one (trained on a single type of features), showed to improve the video/image annotation accuracy. The simplest way to combine the output of the employed classifiers for the same concept is late fusion e.g., in terms of arithmetic mean as in subsection 6.2.2. However, we introduce here a more elaborate approach that arranges the base classifiers on a cascade architecture. We have developed a cascade of classifiers presented in [Markatopoulou et al., 2016], where the trained classifiers are arranged in stages using a search-based algorithm. An image/keyframe is classified sequentially by each stage and the next stage is triggered only if the previous one returns a positive prediction (i.e. that the concept appears in the image/keyframe). The rationale behind this is to rapidly reject images/keyframes that clearly do not match the classification criteria, and focus the complete set of classifiers on those images/keyframes that are more difficult and more likely to depict the sought concept. Consequently, this procedure is able to reduce the number of classifier evaluations and subsequently the number of classifiers to be fused.

6.2.5 Video annotation with complex event labels

In the domain of video annotation with complex event labels, there are numerous challenges associated with building effective video event detectors. One of them is that often there is only a limited number of positive video examples available for training. In this case, it is of high interest to further exploit video samples that do not exactly meet the requirements for being characterized as true positive examples of an event, but nevertheless are closely related to an event class and can be seen as "related" examples of it (i.e. videos that are closely related with the event, but do not meet the exact requirements for being a positive event instance). This is simulated in the TRECVID MED task [Over et al., 2014] by the "near-miss" video examples provided for each target event class.

In the proposed approach, for video representation, 2 keyframes per second are extracted at regular time intervals from each video. Then, each keyframe is represented using the last hidden layer of a pre-trained DCNN. More specifically, a 16-layer pre-trained deep ConvNet network provided in [Simonyan and Zisserman, 2014] is used. This network was trained on the ImageNet data [Deng et al., 2009], providing scores for 1000 ImageNet concepts; thus, each keyframe has a 1000-element vector representation. Then, the typical procedure followed in state-of-the-art event detection systems, which includes the computation of a video-level representation for each video by taking the average of the corresponding keyframe-level representations, is followed [Yu et al., 2014, Guangnan et al., 2014, Bolles et al., 2014].

Except for [Tzelepis et al., 2013], none of the state-of-the-art works take full advantage of these related videos for learning from few positive samples; instead, the "related" samples are either excluded from the training procedure [Guangnan et al., 2014, Bolles et al., 2014], or they are mistreated as true positive or true negative instances [Douze et al., 2014]. In contrast, in our earlier work in ForgetIT [Tzelepis et al., 2013] we exploited related samples by handling them as weighted positive or negative ones by applying an automatic weighting technique during the training stage. To this end, a relevance degree in $(0, 1]$ is automatically assigned to all the related samples, indicating the degree of relevance of these observations with the class they are related to. It was shown that this weighting resulted in learning more accurate event detectors.

Another challenge is that video representation techniques usually introduce uncertainty in the input that is fed to the classifiers, and this also needs to be taken into consideration during classifier training. In ForgetIT [Tzelepis et al., 2015a, Tzelepis et al., 2016], we dealt with the problem of learning video event detectors when a limited number of positive and related event videos are provided. For this, we exploited the uncertainty of the training videos by extending the Linear Support Vector Machine with Gaussian Sample Uncertainty (LSVM-GSU), presented in [Tzelepis et al., 2015b], in order to arrive at non-linear decision functions. Specifically, we extended this version of LSVM-GSU that assumes isotropic uncertainty (hereafter denoted LSVM-iGSU) into a new kernel-based algorithm, which we call KSVM-iGSU. We also further extended KSVM-iGSU, drawing inspiration from the Relevance Degree kernel SVM (RD-KSVM) proposed in [Tzelepis et al., 2013], such that related samples can be effectively exploited as positive or negative examples with automatic weighting. We refer to this algorithm as RD-KSVM-iGSU. We show in the experimental results section (Section 6.3) that the RD-KSVM-iGSU algorithm results in more accurate event detectors than the state-of-the-art techniques used in related works, such as the standard kernel SVM and RD-KSVM.

6.2.6 Advances in comparison to the previous version

The current image/video annotation system extends the previous system described in deliverable [Solachidis et al., 2015] on the following points:

- Introduces DCNNs as concept detectors and shows that they are more accurate than the corresponding detectors of [Solachidis et al., 2015] for solving both annotation and retrieval problems based on the way they are used; i.e. as standalone classifiers or as generators of features, respectively.

- Introduces a video annotation system for concept-based detection on video-fragment level.

- Improves the concept detection accuracy by using the GPU-AGSDA method for dimensionality reduction.

- Combines a subset of hand-crafted features presented in [Solachidis et al., 2015] (ORB, RGB-ORB, OpponentORB) with DCNN-based features using a cascade of classifiers.

- Supports the annotation of video not only with simple concept labels (on video-fragment level) but also with complex event labels (on the entire video level).

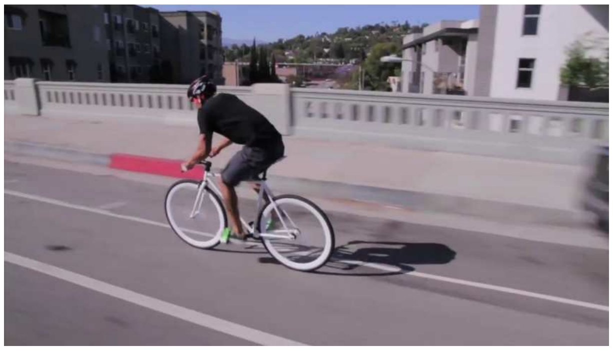
6.3 Experimental evaluation and comparison

Our experiments with respect to concept-based image and video-fragment annotation were performed on the TRECVID 2013 (SIN) dataset [Over et al., 2013] for a set of 38 concepts as stated in [Solachidis et al., 2015]. Tables 5 presents the results related to the improvements introduced in Section 6.2.1, in terms of Mean Extended Inferred Average Precision (MXinfAP) [Yilmaz et al., 2008], which is an approximation of the mean average precision (MAP) suitable for the partial ground truth that accompanies the TRECVID dataset. In Table 5 we compare the following methods: Method A refers to the previous version of image annotation that uses hand-crafted features only [Solachidis et al., 2015], Methods B and C refer to the current versions for concept-based image and video annotation, respectively that use the DCNN features presented in Sections 6.2.1 and 6.2.2.

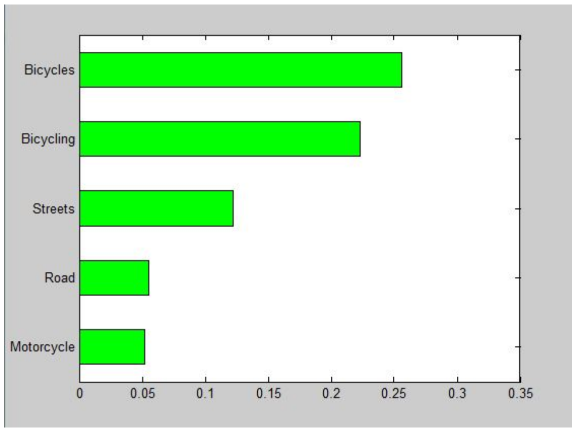
Table 5: Mean Extended Inferred Average Precision (MXinfAP) for 38 single concepts on the TRECVID 2013 SIN dataset. The use of DCNNs improves the method's accuracy from 17.45 to 25.25 for images and 30.45 for videos.

	Method A	Method B	Method C
	Images, Vlad - encoding	Images - DCNNs	Videos - DCNNs
MXinfAP	17.45	25.58	30.45

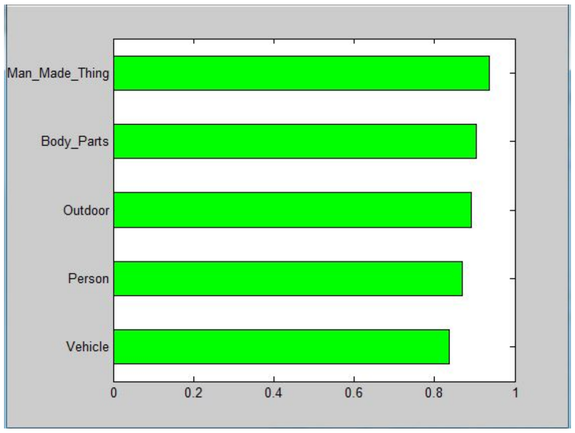
Fig. 8 and 9 show two examples of our video annotation approach for the two different video analysis problems (i.e., video-fragment annotation, video-fragment retrieval). In section 6.1 we have seen that DCNNs can be used both as standalone classifiers and as feature generators. For each of the video analysis problem we collect two different set of scores based on the way that DCNNs have been used (i.e., as standalone classifiers, or as feature generators). Fig. 8 refers to the video-fragment annotation problem, where using the DCNN as standalone detectors returns a more meaningful label ranking (Fig. 8(b)) compared to the label ranking returned by the DCNN when it is used as feature generator (Fig. 8(c)). In the former case, the DCNN correctly annotates the figure with the label “bicycle”. In the latter case, labels “man made thing”, “body parts”, etc. are returned as most suitable annotations, which are not specific enough and representative of the keyframes content. We conclude, that regarding the video/image annotation problem it is more important to train many concepts together, as a DCNN does, than learning concepts independently, e.g. by using DCNN-based features as input to individual SVM classifiers. Fig. 9 refers to the video-fragment retrieval problem for the concept “airplane”. For this problem, using the DCNN as feature generator (Fig. 9(b)) outperforms using it as a standalone classifier (Fig. 9(a)). In the latter case we see some wrongly retrieved keyframes that do not depict an airplane.



(a)

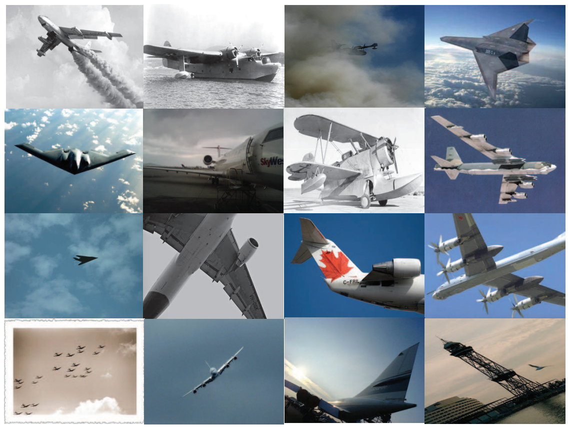


(b)

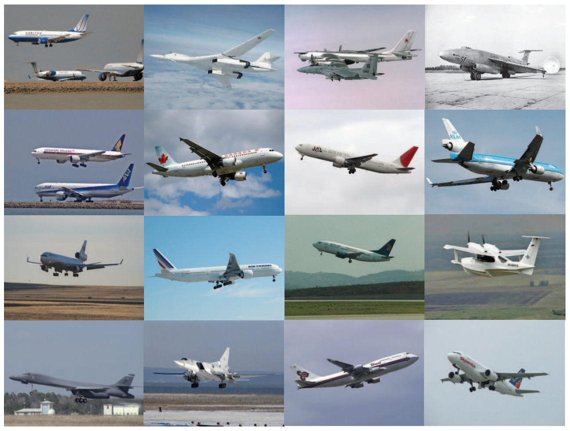


(c)

Figure 8: Example results for the video annotation problem: (a) image (b) DCNN is used as standalone classifier, (c) DCNN is used as generator of features.



(a)



(b)

Figure 9: Example results for the video retrieval problem:(a) DCNN is used as standalone classifier, (b) DCNN is used as generator of features.

Concerning the introduction of the cascade architecture (Section 6.2.4), we extend the set of extracted features presented in Section 6.2.2 with more DCNN-based but also hand-crafted features. A big pool of features is needed in this case in order to benefit from the use of the cascade architecture. As a result, we train a cascade of 11 different visual descriptors that have been arranged in four cascade stages. Specifically, the cascade consists of one binary local descriptor and its two color variants (ORBx3), two types of non-binary local descriptors with their color variants (SURFx3, SIFTx3), the output of the last hidden layer of ConvNet (fc7) network and the output of the last classification layer of the GoogLeNet fine-tuned network. We train one Linear SVM (LSVM) for each of the 11 visual descriptors. Therefore, 11 LSVMs per concept are trained and arranged on cascade stages. The scores returned from them are fused using the cascade. Table 6 presents the accuracy of the proposed cascade in terms of MXinfAP (Approach A). We also compare the proposed cascade with the late fusion alternative (Approach B), where the scores returned from the LSVMs are fused using the arithmetic mean. Finally, to analyse the influence of GPU-AGSDA on the vectors of the extracted features, we compare in Table 6 a) the proposed cascade (Approach C) and b) the late fusion alternative (Approach D) when all features are firstly reduced using the GPU-AGSDA method.

Table 6: Mean Extended Inferred Average Precision (MXinfAP) for 38 single concepts on the TRECVID 2013 SIN dataset

	Approach A	Approach B	Approach C	Approach D
	Cascade	Late Fusion	Cascade+AGSDA	Late Fusion+AGSDA
MXinfAP	30.37	30.17	31.79	31.89

We observe that cascade and late fusion present similar accuracy. However, these differences, although small, are accompanied with considerable improvements in computational complexity when the proposed cascade is employed. Specifically, the employed cascades lead to approximately 37% relative decrease in the amount of classifier evaluations compared to the late fusion alternative. Regarding the use of the AGSDA dimensionality reduction technique, according to Table 6, it improves the concept detection results, compared to the use of the initial feature vectors. Details on the very significant computational efficiency gains of using AGSDA can be found in [Arestis-Chartampilas et al., 2015].

On the subject of video annotation with event labels the large-scale video dataset of the TRECVID Multimedia Event Detection (MED) 2014 benchmarking activity [Over et al., 2014] is used for evaluation in our experiments. Similarly to [Tzelepis et al., 2015b, Tzelepis et al., 2015a], we used only the training portion of the TRECVID MED 2014 task dataset, which provides ground-truth information for 30 complex event classes, since for the corresponding evaluation set of the original TRECVID task there is no ground-truth data available. Hereafter, we refer to the aforementioned ground-truth-annotated dataset as MED14 and we divide it into a training subset, consisting of 50 positive and 25 related (near-miss) samples per event class, together with 2496 background samples (i.e. videos that are negative examples for all the event classes), and an evaluation subset consisting of approximately 50 positive and 25 related samples per event class, along with another 2496 background samples.

For assessing the detection performance of each trained event detector, the average precision (AP) [Robertson, 2008] measure was utilized, while for measuring the detection performance of a classifier across all the event classes we used the mean average precision (MAP), as is typically the case in the video event detection literature, e.g., [Gkalelis et al., 2014, Over et al., 2014, Tzelepis et al., 2013, Tzelepis et al., 2015a].

In contrast to the existing event detection literature, in the case of RD-SVM-iGSU (or also KSVM-iGSU and the original LSVM-iGSU), the keyframe-level video representations can be seen as observations of the input Gaussian distributions that describe the training videos. That is, let $\mathcal{X}$ be a set of l annotated random vectors representing the aforementioned video-level model vectors. We assume that each random vector is distributed normally; i.e. for the random vector $\mathbf{X}_i$ representing the i-th video, we have $\mathbf{X}_i \sim \mathcal{N}(\mathbf{x}_i, \Sigma_i)$. Also, for each random vector $\mathbf{X}_i$, a number N_i, of observations, $\{\mathbf{x}_i^t \in \mathbb{R}^n \colon t = 1, \ldots, N_i\}$ is available; these are the keyframe-level model vectors that have been computed. Then, the mean vector and the covariance matrix of $\mathbf{X}_i$ are computed respectively as follows

$$\mathbf{x}_i = \frac{\sum_{t=1}^{N_i} \mathbf{x}_i^t}{N_i}, \quad \Sigma_i = \frac{\sum_{t=1}^{N_i} (\mathbf{x}_i^t - \mathbf{x}_i)(\mathbf{x}_i^t - \mathbf{x}_i)^\top}{N_i - 1} \tag{6.1}$$

Due to the assumption for isotropic covariance matrices, we approximated the above covariance matrices as multiples of the identity matrix, i.e. $\widehat{\Sigma}_i = \sigma_i^2 I_n$ by minimizing the squared Frobenious norm of the difference $\Sigma_i - \widehat{\Sigma}_i$ with respect to σ_i^2. It can be shown (by using simple matrix algebra [Golub and Van Loan, 2012]) that for this it suffices to set σ_i^2 equal to the mean value of the elements of the main diagonal of Σ_i.

The kernel extensions of LSVM-iGSU [Tzelepis et al., 2015b] (KSVM-iGSU, RD-KSVM-iGSU) were tested on the MED14 dataset, and compared to standard kernel SVM (KSVM), LSVM-iGSU [Tzelepis et al., 2015b] and RD-KSVM [Tzelepis et al., 2013]. We note here that for the problem of video event detection (and especially when only a few positive training samples are available), kernel SVM is the state-of-the-art approach [Yu et al., 2014, Bolles et al., 2014]. On the other hand, when a few related samples are available, RD-KSVM leads to state-of-the-art detection performance [Tzelepis et al., 2013]. We experimented on the problem of learning from 10 positive examples per each event class, together with 5 related samples, that were drawn from the set of 25 related samples provided for each event class following the method presented in [Tzelepis et al., 2013]; i.e. the 5 nearest to the median of all 25 related samples were kept for training both RD-KSVM and RD-SVM-iGSU. Also, we randomly chose 70 negative samples for each event class, while we repeated each experiment 10 times. That is, for each different experimental scenario, the obtained performance of each classifier (KSVM, RD-KSVM, LSVM-iGSU, KSVM-iGSU, and RD-SVM-iGSU) was averaged over 10 iterations, for each of which 10 positive samples have been randomly selected from the pool of 50 positive samples that are available in our training dataset for each target event class.

For all the above experimental scenarios where a kernel classifier was used, the Radial Basis Function (RBF) kernel had been used. Training parameters (C for LSVM-iGSU;

C, γ for KSVM, KSVM-iGSU; and C, γ, and c for RD-KSVM, RD-KSVM-iGSU) were obtained via cross-validation. For C, γ, a 10-fold cross-validation procedure (grid search) was performed with C, γ being searched in the range $\{2^{-16}, 2^{-15}, \ldots, 2^{2}, 2^{3}\}$. For c, an approach similar to that presented in [Tzelepis et al., 2013] was followed. That is, related samples were initially treated as true positive and true negative ones (in two separate cross-validation processes) and C, γ were optimized as described above; then, by examining the minimum cross-validation errors of the two above processes, we automatically chose whether to treat the related samples as weighted positive or weighted negative ones, and also fixed the value of C to the corresponding optimal value. Using this C, we proceeded with a new cross-validation process (again grid search) for finding the optimal γ, c pair (where c was searched in the range $[0.01, 1.00]$ with a step of 0.05).

Table 7 shows the performance of the latest ForgetIT methods KSVM-iGSU and RD-KSVM-iGSU, compared to LSVM-iGSU [Tzelepis et al., 2015b], the standard KSVM, and the RD-KSVM [Tzelepis et al., 2013], respectively, in terms of average precision (AP), for each target event, and Mean AP (MAP), across all target events. Bold-faced values indicate the best performance for each event class. We can see that LSVM-iGSU, whose improved performance over the standard linear SVM was studied extensively in [Tzelepis et al., 2015b], cannot outperform the kernel methods that are typically used for the video event detection problem, achieving a MAP of 0.1761. Without using any related samples, KSVM-iGSU that takes into account the input uncertainty, outperformed the standard kernel SVM for 25 out of 30 target event classes, achieving a MAP of 0.2527 in comparison to KSVM's 0.2128 (achieving a relative boost of 18.75%). Moreover, when related samples were used for training, the proposed RD-KSVM-iGSU outperformed the baseline RD-KSVM for 27 out of 30 target event classes, achieving a MAP of 0.2730, in comparison to RD-KSVM's 0.2218 (i.e. a relative boost of 23.08%). This RD-KSVM-iGSU result also represents a 8% relative improvement (MAP of 0.2730 versus 0.2527) in comparison to KSVM-iGSU, which does not take advantage of related video samples during training. The above results suggest that using uncertainty for training video event detectors leads to promising results, while the additional exploitation of related samples can further improve event detection performance.

Finally, in Fig. 10 we present indicative results of RD-KSVM-iGSU [Tzelepis et al., 2016] in comparison with RD-KSVM [Tzelepis et al., 2013] for four event classes, showing the top-5 videos each classifier retrieved. Green borders around frames indicate correct detection results, while red ones indicate false detection. These indicative results illustrate the practical importance of the AP and MAP differences between these two methods that are observed in Table 7.

6.4 Software implementation

For image/video annotation we developed our software, using C++. In the case that a video is given as input to the software, the temporal video segmentation method described in previous section is called first, in order to extract keyframes from the input

Table 7: Evaluation of event detection approaches on the MED14 dataset (AP (per event class), MAP (total))

Event Class	LSVM-iGSU	KSVM	KSVM-iGSU	RD-KSVM	RD-KSVM-iGSU
E021	0.1741	0.1763	0.1923	0.1823	**0.2167**
E022	0.1847	0.1903	0.2495	0.2009	**0.2604**
E023	0.4832	0.5665	0.6361	0.5435	**0.6432**
E024	0.0536	0.0482	**0.0667**	0.0489	0.0549
E025	0.0117	0.0210	0.0257	0.0200	**0.0287**
E026	0.1002	0.1388	0.1530	0.1385	**0.1701**
E027	0.1600	0.2882	**0.4162**	0.2899	0.4002
E028	0.2030	0.2234	0.2338	0.2250	**0.2495**
E029	0.2394	0.2321	0.2948	0.2521	**0.3106**
E030	0.1612	**0.2464**	0.2220	0.2398	0.2451
E031	0.4911	0.4595	0.6122	0.4762	**0.6497**
E032	0.0706	0.1278	0.1490	0.1301	**0.1729**
E033	0.2217	0.3170	0.3731	0.3265	**0.3971**
E034	0.1658	0.2129	0.3302	0.2231	**0.6541**
E035	0.2331	0.2650	0.3580	0.2874	**0.3771**
E036	0.1753	0.1897	0.2139	0.1923	**0.2230**
E037	0.2454	0.2928	0.3325	0.3133	**0.3569**
E038	0.0745	0.1127	0.1231	0.1187	**0.1259**
E039	0.2161	0.2531	**0.3990**	0.3294	0.3986
E040	0.5809	**0.3205**	0.3157	0.3095	0.3021
E041	0.0489	0.1589	0.2166	0.1782	**0.2254**
E042	0.1021	0.1358	0.1787	0.1532	**0.1799**
E043	0.0967	0.1568	0.2037	0.1890	**0.2101**
E044	0.0732	**0.2697**	0.2087	0.2543	0.1968
E045	0.1307	0.2315	0.2517	0.2385	**0.2786**
E046	0.1952	0.2457	0.2668	0.2412	**0.2721**
E047	0.0531	0.0837	0.1796	0.1187	**0.1865**
E048	0.0672	0.0642	0.0672	0.0654	**0.0674**
E049	0.0641	0.1250	0.1245	0.1189	**0.1329**
E050	0.2076	0.2321	0.1867	**0.2489**	0.2039
MAP	0.1761	0.2128	0.2527	0.2218	**0.2730**

video. Then the software takes the following steps to annotate the images/keyframes: 1) extracts DCNN-based features from the input images/keyframes, 2) Serves the extracted features into trained concept detectors, 3) Returns two sets of scores that contain the probabilities of the presence of each concept; each set of scores corresponds to the two analysis problems: video/image annotation, where DCNN is used as standalone classifier, video/image retrieval where DCNN is used as feature generator. If a CUDA capable NVIDIA graphic card is available the process is accelerated following the GPU-processing, otherwise, parallel processing is followed using openMP[4] that manages the execution time. Both the image and video annotation methods are integrated at the overall ForgetIT system as web services and a detailed description of each service functionality and instruction of use is given in Section 10. The technical characteristics of the methods

[4]`http://openmp.org/wp/` (Accessed date: 31-01-2016)

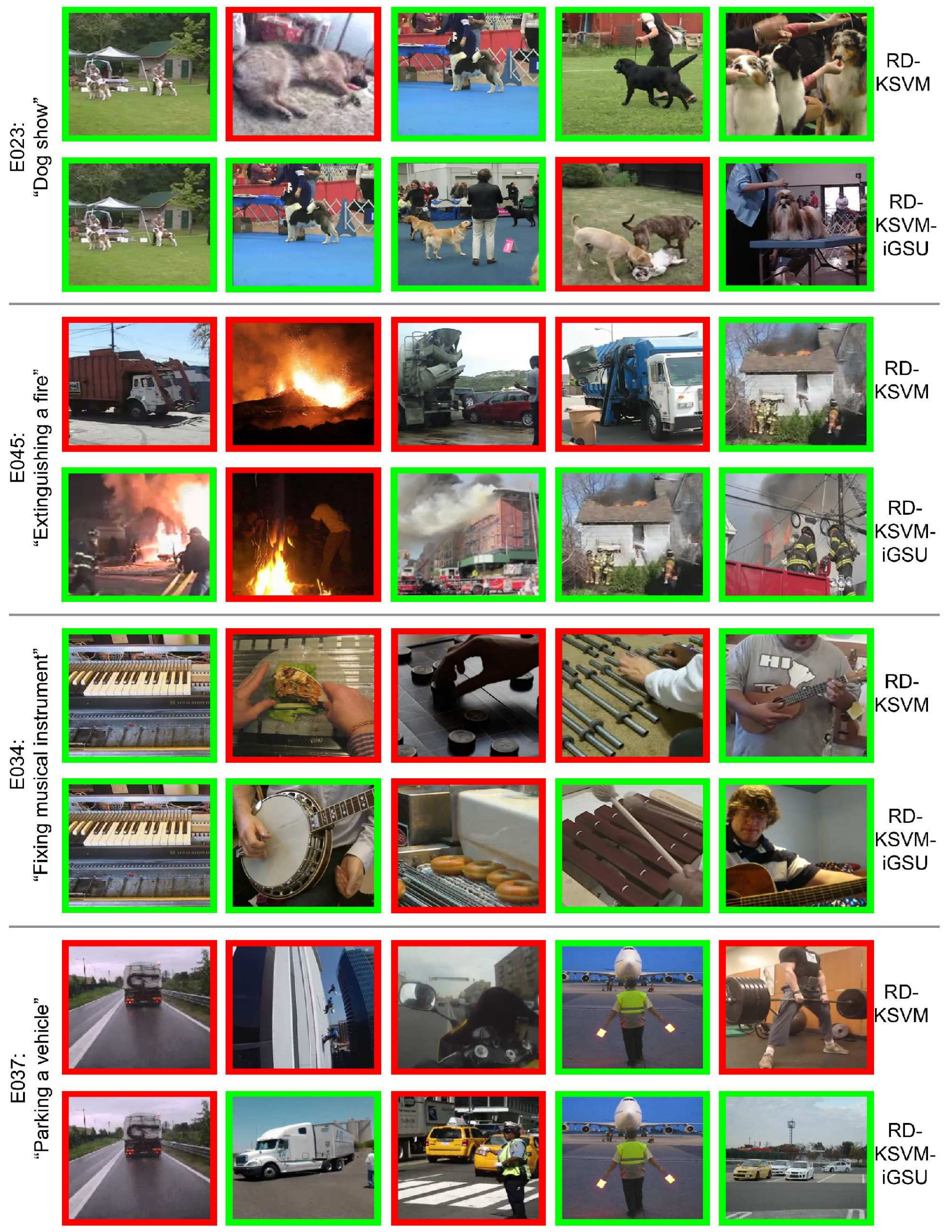


Figure 10: Indicative results (top-5 returned shots) for comparing RD-KSVM-iGSU with RD-KSVM, for four event classes

are presented in Table 8.

6.5 Conclusions

In this section the updated methods for image annotation is presented and new methods are introduced for video annotation. For concept-based video annotation on the video-fragment level, we extended the image annotation method by adding a pre-processing

Table 8: Software technical details for the ForgetIT concept-based annotation for image collections and videos

Functional description	Image/video annotation
Input	An image or an image collection or video keyframes (result of temporal video segmentation)
Output	Vectors of scores per image - keyframe indicating the presence of concepts
Limitations	N/A
Language/technologies	C++ (Microsoft visual studio 2013 compiler), parallel processing using openMP
Hardware Requirements	CUDA capable NVIDIA graphic card with compute capability 3.0 or greater
OS Requirements	Windows

step, which segments the video in keyframes, and then treat each of them as an image. Additionally, we experimented with more elaborate approaches of combining the outputs of classifiers trained on different types of features (hand-crafted, DCNN-based, etc.) and also exploit the benefits in terms of accuracy while using the GPU-AGSDA method. Finally, we presented methods that can be used in order to annotate video sequences with more complex events.

7 Image/video quality assessment

7.1 Problem statement

Knowledge of the quality of an image or a video is important since it can be exploited for the estimation of the Preservation Value (PV). For example, one simple rule that can be easily applied is not to preserve media items that were assessed to be of low quality. An image quality assessment (IQA) method was introduced in [Papadopoulou et al., 2014], where four quality measures were applied on each image of the collection. Separately for each measure a quality score was extracted and subsequently a final score, calculated by fusing the four measures, was used for quantifying the overall quality of an image. In this deliverable we study a more complex measure, the perceived photographic appeal of the image, aiming to calculate the aesthetic quality of an image. Low-level features, such as brightness, texture, color distribution etc. are used for assessing the aesthetic quality of an image [Tong et al., 2005, Datta et al., 2006, Luo and Tang, 2008] as well as combinations of low- and high-level features, such as generic image descriptors (e.g., SIFT), concepts and objects depicted in photo, object positioning and the rule of thirds [Ke et al., 2006, Lo et al., 2012, Marchesotti et al., 2011, Dhar et al., 2011]. Moreover, in this document we introduce new methods for both the quality assessment and the aesthetic quality assessment of videos. In the relevant literature the Video Quality Assessment (VQA) metrics are categorized as 1) Full-Reference (FR), 2) Reduced-Reference (RR) [Winkler and Mohandas, 2008] and 3) No-Reference (NR). NR video metrics, as the name suggests, are much more flexible for user-end applications. However, the studies on the NR metrics are rather limited. There are basically two different non-hybrid approaches used for the NR quality metrics: artifact- and network quality- based. In the artifact-based methods, quality is described as a function of artifacts such as blurriness, blockiness, jerkiness etc. Various methods are used to parameterize these artifacts in this metric type [Farias and Mitra, 2005, Eden, 2007, Lin et al., 2012b, Zhu et al., 2013a]. On the other side, in the network quality-based metric type, the quality is related with parameters of compression, transmission etc. and the state-of-the-art methods follow two directions; finding visibility of packet losses [Staelens et al., 2010, Argyropoulos et al., 2011] and bitstream-based methods which use different parameters such as bit count, Quantization Parameter (QP), Motion Vector (MV) information, frame types etc. [Keimel et al., 2011, Liu et al., 2011, Staelens et al., 2013, Zhao et al., 2014]. Finally, although Video Aesthetic Quality (VAQ) is a challenging task, it has not been extensively studied. Some of features have been introduced in order to support VAQ assessment in videos, such as semantically independent (motion space, hand-shaking, color and frame composition) and semantically dependent features (motion direction entropy, color saturation and lightness) [Yang et al., 2011], and photo- and motion-based features [Yang et al., 2011]. Furthermore, in [Niu and Liu, 2012], the authors design a variety of aesthetics-related features for video, such as visual continuity and shot length, and examine their performance in finding professional videos.

7.2 ForgetIT approach

In the following the methods developed in ForgetIT for assessing the quality and the aesthetic quality of still images and videos are presented.

7.2.1 Image aesthetic quality assessment

The evaluation of the aesthetic quality of a photo can be considered as a subjective task. However, there are some rules and guidelines which are widely used by amateur and professional photographers and contribute to the aesthetic appeal of the photo. We developed an image aesthetic quality assessment approach which is based on a set of photographic rules. Specifically, we introduce a set of four feature vectors that correspond to four basic rules of photography and describe the photo's simplicity, colorfulness, sharpness and pattern. The aforementioned features are concatenated in a final feature vector where a SVM classifier is applied in order to perform the aesthetic quality evaluation. The overall procedure of the ForgetIT image aesthetic quality assessment method is presented in Fig. 11.

Simplicity: Simplicity depends on two basic rules, the color difference between the main subject and the background and the spatial arrangement of the main subject. The assumption that the main subject is the least uniform portion of the image acts as a starting

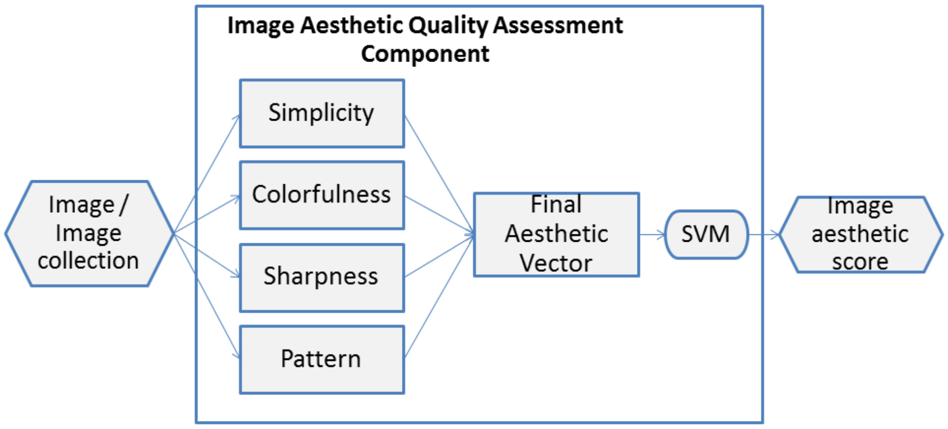


Figure 11: Image Aesthetic Quality Assessment component

point. Then, we identify our main subject by edge detection (using the image gradient magnitude and orientation), followed by the application of simple morphological operations on a binarized edge mask, so as to result in a limited number of connected regions. Subsequently, in order to capture color differences, we i) compute the HSV histograms of the main subject and the background, and we estimate the mean value, the standard deviation, the kurtosis, the skewness and the Kullback-Leibler divergence of both distributions for each color component, and ii) considering two color regions of the image, the main subject and the background, we estimate a $5-$bin histogram of the color-difference formula [McLaren, 1976], CIE 1976 (L* a* b*). In parallel, in order to capture the spatial arrangement of the main subject, we compute the percentage of its pixels that appear in each of the 9 patches of the rule of thirds grid.

Colorfulness: Capturing photographs with vibrant colors and intense contrast are among the most powerful ways to arouse viewers attention. To evaluate the colorfulness, we detect three main color image regions using k-means clustering. Afterwords, for each of these color regions we estimate a $10-$bin color histogram, where each bin corresponds to one of the ten basic colors, and we calculate the percentage of pixels that each color region occupies. For each of the nine patches of the rule of thirds grid we additionally estimate a $5-$bin histogram for each color component of the HSV color space and the Correlated Color Temperature (CCT) [McCamy, 1992]. Finally, we also extract a feature to measure contrast and darkness, following the method presented in [Papadopoulou et al., 2014].

Sharpness: To evaluate the sharpness of photography we apply the no-reference image blur detection scheme described in [Papadopoulou et al., 2014] so as to extract features that can describe the blurriness of the image, resulting in a 500-element sharpness feature vector, according to the Algorithm 1 of [Mavridaki and Mezaris, 2014]. In addition, in order to detect more complicated forms of blur, such as motion blur, we use the Haar wavelet transform. We estimate the horizontal H, vertical V and diagonal D detail coefficients for three levels of the Haar wavelet transform and we compute a $5-$bin histogram of the edge map ($Emap = \sqrt{H^2 + V^2 + D^2}$) for each of the nine patches of the rule of thirds grid.

Pattern: In order to extract the appropriate information to describe the aforementioned characteristic of the images we divide the image in multiple half-image parts. For every pair of half-image patches we detect SURF interest points and subsequently we perform point feature matching, capturing the similarities of these patches. In addition, our aim is to examine the presence of intense edges and if these edges are distributed in the same manner in both half-image patches, for each of the four pairs. In order to achieve this, we estimate the mean value, the standard deviation and the Kullbak-Leibler divergence after applying the Haar wavelet transform on both half-image patches of each pair.

For a more detailed description of the IAQ method refer to [Mavridaki and Mezaris, 2015].

7.2.2 Video aesthetic quality assessment

The evaluation of the aesthetic quality of a video is a challenging task where several methods aim to find the appropriate features. We use a set of photo- and motion-based features, and a set of features which are based on shot-detection and shot-transition type. The photo-based features are extracted based on the IAQ assessment method proposed in [Mavridaki and Mezaris, 2015] which is presented above. The motion-based features include: 1) a measure of similarity between frames (cross-correlation between consecutive frames), 2) a measure of the diversity of motion directions (motion direction entropy), 3) a measure of the stability of camera during the capturing process (hand-shaking) and 4) a measure which can distinguish the difference between three categories of shots: focused shots, panorama shots and static shots (shooting type). These motion-based features are presented in [Yeh et al., 2013]. Additionally, we employ two video-based features, where for each shot we estimate its duration (shot duration) and for the whole video we estimate the percentage of abrupt and gradual transitions (transition type). The overall procedure of the VAQ assessment method is presented in Fig. 12.

The VAQ assessment procedure of feature extraction is performed at two levels, shot- and video-level, as described in the following. Initially, we divide each video into its shots applying the shot detection method presented in [Apostolidis and Mezaris, 2014]. For each shot one representative keyframe and its time-position on the video sequence are extracted. The photo- and the motion-based features are extracted for each individual shot of the video, while the video-based features are extracted only once for a video. After the execution of this process we have a feature vector for each shot, which contains the photo-, motion- and video-based features. On this set of features we apply a learning algorithm which uses Kernel Support Vector Machine with Isotropic Gaussian Sample Uncertainty (KSVM-iGSU) [C. Tzelepis and Patras, 2016] (see also Section 6.2.5), in order to classify the video as video of high- or low-aesthetic quality.

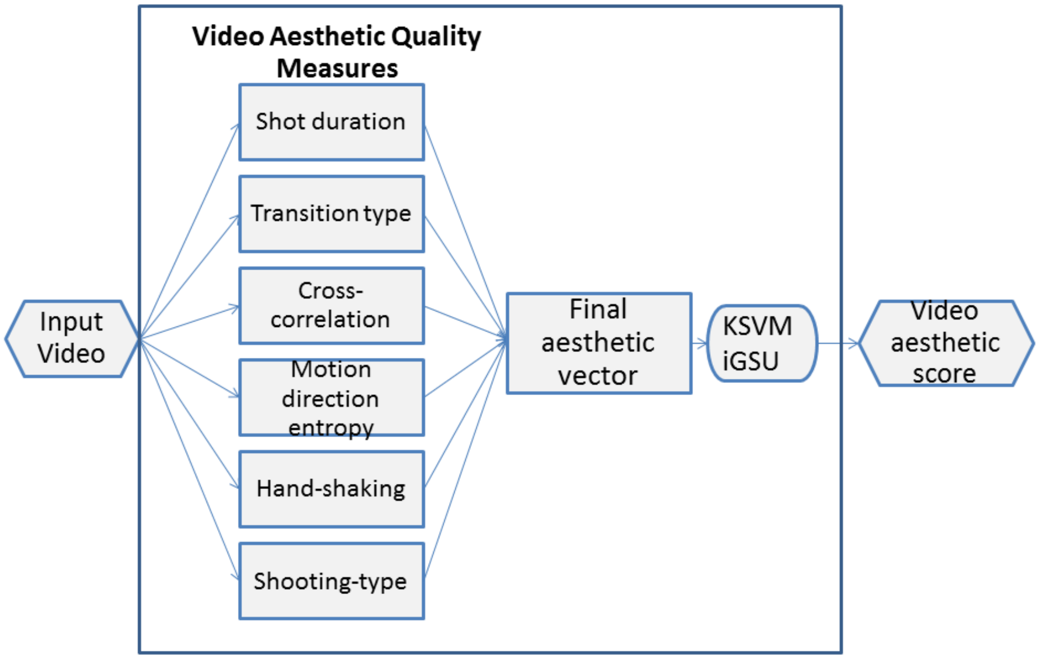


Figure 12: Block diagram of the Video Aesthetic Quality (VAQ) Assessment method

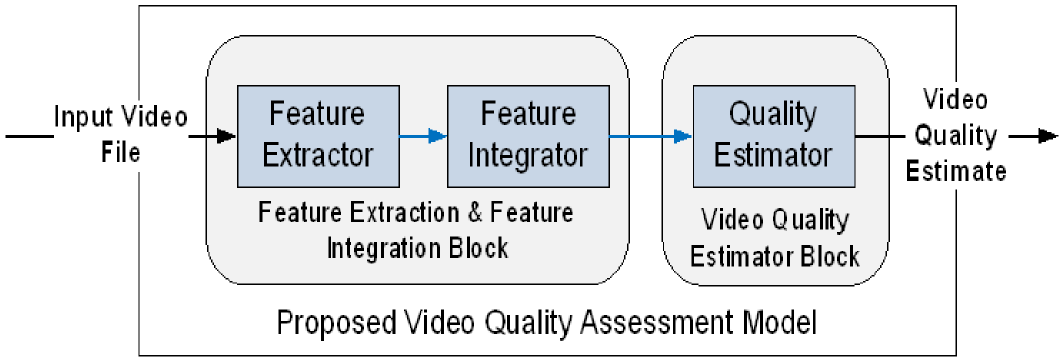

Input Video File
Feature Extractor
Feature Integrator
Feature Extraction & Feature Integration Block
Quality Estimator
Video Quality Estimator Block
Video Quality Estimate
Proposed Video Quality Assessment Model

Figure 13: Block diagram of the Video Quality Assessment (VQA) method

7.2.3 Video non-aesthetic quality assessment

In this project we use a novel spatiotemporal NR VQA metric that utilizes the video specific parameters, such as spatial complexity, motion, bit rate, and packet loss ratio for network conditions, which are extremely important for the judgment of the quality perceived by the user. The block diagram of the proposed VQA model is depicted in Fig. 13. As seen from the figure, the proposed VQA metric consists of three main blocks, namely the feature extractor block, the feature integrator block and the quality estimator block.

In our formulation we utilize DMOS (Differential Mean Opinion Score) which indicates the quality loss of the tested video, compared to its reference video.

Feature Extractor Block: The first block in the VQA model is the feature extractor block.

The video whose perceived quality is desired to be estimated is the input of this block. This block is responsible for extracting five different features of the video. Three of these features are utilized in order to have information about the spatiotemporal characteristics of the video being analyzed. The first of the three features, Modified Spatial Information MSI, is utilized in order to understand the spatial complexity of the video. The second and third features, namely Zero Motion Vector Ratio Z and Mean Motion Vector Magnitude M, focus on the temporal complexity of the video. After extracting features related to the spatiotemporal characteristics of the video, average bit rate BR is extracted as the fourth feature in order to utilize the compression amount of the video. The last feature in our VQA model is the packet loss ratio β, the percentage of the number of lost Real-Time Transfer Protocol (RTP) packets to the total number of RTP packets. This feature is considered due to the fact that perceived video quality degrades in case of packet losses, though many different error concealment algorithms are employed in order to compensate lost packets and it will be used in quality estimator block.

Feature Integrator Block: The second block in the VQA model is the feature integrator block. Four features, i.e. MSI, Z, M, BR are inputs of this block which is responsible for determining parameters that will be used in the video quality estimator block. These parameters are spatial distortion S (7.2) and temporal distortion T (7.3).

$$S = \frac{MSI}{BR} \tag{7.2}$$

$$T = \frac{(1-Z)M}{BR} \tag{7.3}$$

After evaluating the S and T parameters for all the videos in Live VQA dataset, it has been observed that the rate of increase in $DMOS$ depends on S. In addition, the variance of the T values is larger than that of the S values in the order of $1:100$ in all videos. In order to reflect both the individual and joint effects of S and T and to bring the $T4$ and the S values into the same scale, the final value of T is calculated as:

$$T = \begin{cases} T & \text{, if } T < 100S \\ 100S & \text{, if } T \geq 100S \end{cases} \tag{7.4}$$

Video Quality Estimator Block: The last block in the VQA metric is the video quality estimator block. The S and T parameters along with the β parameter are inputs of this block. This block is responsible for estimating perceived video quality based on these inputs as:

$$DMOS_{INITIAL}(S,T) = a + bS + cT + dS^2 + eST + fT^2 \tag{7.5}$$

The expression in (7.5) is a quadratic equation both in S and T. Suppose there exist a number of different videos having identical S and different T values. In this case, it is expected that the smallest $DMOS$ estimate (highest perceived video quality) would be produced for the video with the smallest T value, since these videos have the same spatial distortion and the video with the smallest temporal distortion should be perceived as the highest quality. At this point, S in (7.5) can be replaced by S_o since each video is assumed to have an identical S value:

$$DMOS_{INITIAL}(S_o, T) = fT^2 + \gamma T + \lambda \tag{7.6}$$

where γ and λ are given in (7.7) and (7.8), respectively:

$$\gamma = c + eS_o \tag{7.7}$$

$$\lambda = a + bS_o + dS_o^2 \tag{7.8}$$

Reorganizing (7.6) results in (7.9):

$$DMOS_{INITIAL}(S_o, T) = \left(\sqrt{f}T + \frac{\gamma}{2\sqrt{f}}\right)^2 + \left(\lambda - \frac{\gamma^2}{4f}\right) \tag{7.9}$$

Clearly, expression in (7.9) is minimum when T is $T_{min}(S_o)$:

$$T_{min}(S_o) = \frac{-\gamma}{2f} = -\frac{c + eS_o}{2f} \tag{7.10}$$

In order to overcome the sufferance due to the quadratic structure in (7.5), we insert $T_{min}(S_o)$ in (7.5) instead of T and modify the $DMOS$ estimate of video$_k$ as

$$DMOS_{COMP}(S, T) = \begin{cases} DMOS_{INITIAL}(S, T) & \text{, if } T \geq T^* \\ DMOS_{INITIAL}(S, T^*) \cdot h_{CRF}(T, T^*) & \text{, if } T < T^* \end{cases} \tag{7.11}$$

where $T^* = T_{min}(S)$ and $h_{CRF}(T, T_{min}) = (T/T_{min})^{0.05}$.

Up to here, we have only considered distortions occurring in compression. In order to take distortions occurring in transmission into account, we utilize the β parameter. The $DMOS$ expression, considering also the effects of transmission distortions, becomes:

$$DMOS_{ALL}(S, T, \beta) = DMOS_{COMP}(S, T) \cdot h_{TR}(\beta) \tag{7.12}$$

where

$$h_{TR}(\beta) = \begin{cases} m \cdot \exp(n \cdot \beta) & \text{, if } \beta > 0 \\ 1 & \text{, if } \beta = 0 \end{cases} \tag{7.13}$$

and m and n are predefined constants defined based on network characteristics.

7.3 Experimental evaluation and comparison

7.3.1 Image aesthetic quality assessment

For the training and the evaluation of the IAQ performance we use several publicly available datasets which consist of photos acquired from on-line photography communities.

Specifically, we use the CUHKPQ dataset [Tang et al., 2013] which is divided into seven thematic categories, and each photo has been classified by members of the photography community as high-aesthetic or low-aesthetic quality photo. We randomly partition ten times the photos of each category, assigning half of them as training and the rest as testing data. The performance of our aesthetic assessment approach is evaluated by calculating the Area Under ROC Curve (AUC). As can be seen in Table 9, the IAQ assessment method achieves noticeable performance, both for each of the different subject categories and overall.

Table 9: Experimental results (AUC) on the CUHKPQ dataset

	Features	**Animal**	**Architecture**	**Human**	**Landscape**	**Night**	**Plant**	**Static**	**Overall**
Lo et al. [Lo et al., 2012]	**Concatenated**	0.9160	0.8414	0.9209	0.9065	0.8633	0.9330	0.8972	0.8974
Tang et al. [Tang et al., 2013]	**Concatenated**	0.8712	0.9004	0.9631	0.9273	0.8309	0.9147	0.8890	0.9044
Luo et al. [Luo and Tang, 2008]	**Concatenated**	0.8712	0.9004	0.9631	0.9273	0.8309	0.9147	0.8890	0.7792
Ke et al. [Ke et al., 2006]	**Concatenated**	0.7751	0.8526	0.7908	0.8170	0.7321	0.8093	0.7829	0.7944
Proposed method	Simplicity	0.8547	0.8155	0.8924	0.8787	0.7839	0.8961	0.8323	0.8206
	Colorfulness	0.8215	0.8661	0.9112	0.9182	0.7947	0.9084	0.8605	0.8407
	Sharpness	0.9451	0.8721	0.9615	0.8674	0.9042	0.9248	0.9066	0.9085
	Pattern	0.8966	0.8312	0.8614	0.8562	0.8954	0.8950	0.8184	0.8434
	Concatenated	**0.9542**	**0.9208**	**0.9731**	**0.9410**	**0.9426**	**0.9603**	**0.9407**	**0.9460**

7.3.2 Video aesthetic quality assessment

As far as the VAQ assessment method is concerned, the training and the evaluation of the method's performance is conducted using a video dataset collected and annotated by CERTH. The main goal is to create a new video dataset containing videos which are closer in a real life scenario where several users had captured various events of their lives, such as parties, school concerts, training processes etc. Therefore, we downloaded 700 videos of a variety of categories, such as outdoor activities, DIY videos, make up tutorials, cooking videos, lectures, home-made videos etc. Each of these videos has duration from 1 to 6 minutes so as to approximate a real life scenario but also to be able to be processed in a reasonable time. Subsequently, we conducted an experiment where twelve users rated the aesthetic value of these videos after having watched some examples of high and low aesthetic quality videos. The final aesthetic score is the median score of the user's individual scores. The CERTH's video aesthetic quality dataset consists of 350 videos of high aesthetic quality and 350 videos of low aesthetic quality. The performance of our VAQ assessment approach is evaluated calculating the accuracy. The initial dataset is divided into two parts, the first subset which contains 200 positive and 200 negative videos and it is used as evaluation set, and the second subset which contains 150 positive and 150 negative videos and it is used as training set. The ForgetIT VAQ assessment method achieves 68% accuracy on training set, which is quite good given the complexity of the task.

7.3.3 Video non-aesthetic quality assessment

Three VQA datasets, namely LIVE VQA [Seshadrinathan et al., 2010], EPFL-PoliMI [De Simone et al., 2010], and IT-IST [Brandão and Queluz, 2010] are selected to compare our VQA algorithm to the existing available VQA metrics. These VQA datasets are publicly available and widely accepted. The videos in these datasets have different properties such as content, spatial resolution, bitrate, frame rate, packet loss etc. Therefore, these datasets provide a thorough test environment since they cover a variety of videos.

Among these datasets, LIVE VQA dataset is used for training and EPFL-PoliMI and IT-IST are used for evaluation of our algorithm. The LIVE VQA dataset consists of 10 video contents distorted by 4 different processes. These processes are MPEG-4 AVC/H.264 compression (4 videos), MPEG-2 compression (4 videos), ethernet packet losses (4 videos) and wireless bit losses (3 videos). It covers a large number of videos of the whole video space due to the variety in both video characteristics and distortion types. The testing method on LIVE VQA dataset is cross-validation. We randomly selected 10 videos from each distortion type for the training process. Since there are 4 different distortion processes, we utilized 40 distorted video bitstreams for training among 150 distorted bitstreams. Remaining bitstreams are used in the evaluation step. Fig. 14 shows the scatter plots of the subjective DMOS (y-axis) against the DMOS estimates of the video quality metric (x-axis) on LIVE VQA dataset.

Results of our VQA algorithm are compared to the results of well-known FR and NR VQA algorithms which utilized the same VQA datasets. While comparing, the Pearson Correlation Coefficient (PCC) and Spearman Rank Order Correlation Coefficient (SROCC) methods are used to calculate the correlation between the subjective $DMOS$ scores and the estimated quality results.

We compare the performance of DMOS$_{ALL}$, to the FR metrics such as PSNR, VSNR (Visual SNR) [Chandler and Hemami, 2007], SW-SSIM (Saliency Weighed Structural Similarity index) [Wang and Li, 2007], MS-SSIM (Multiscale SSIM) [Wang et al., 2003], VQM (Video Quality Metric) [Pinson and Wolf, 2004], MOVIE (Motion-based Video Integrity Evaluation index) [Seshadrinathan and Bovik, 2010] and NR metrics such as WBQM (Wavelet Based Quality Metric) [Dimitrievski and Ivanovski, 2012], C-VQA (Compressed Domain VQA) [Lin et al., 2012a], LapPyr (Laplacian Pyramid based VQA) [Zhu et al., 2013b], DVQPM (DCT-based video quality prediction model) [Zhu et al., 2013a], Zero-shot prediction [Mittal et al., 2014], and Video-BLIINDS [Saad et al., 2014]. The performance results on LIVE VQA dataset are presented in Tables 10 and 11. It is worth noting that WBQM, C-VQA, LapPyr, and DVQPM are designed to be used only on MPEG-4 AVC/H.264 compressed bitstreams.

For the MPEG-4 AVC/H.264 compressed bitstream case, the results show that the DMOS$_{ALL}$ outperforms all the FR algorithms and WBQM, C-VQA and Zero-shot prediction NR algorithms. LapPyr and DVQPM seem to be more accurate; however, these algorithms utilize leave-one-out strategy while evaluating the video quality. The results of the algorithms utilizing the leave-one-out strategy are expected to provide better scores. Similarly,

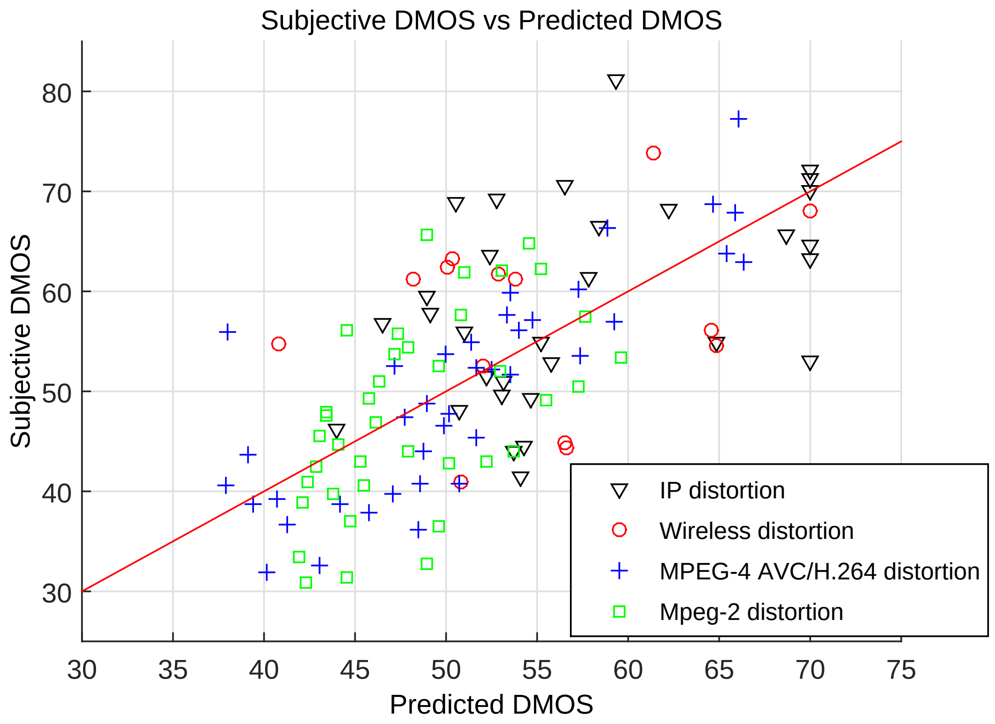


Figure 14: Subjective vs objective DMOS (DMOS$_{ALL}$)

Video-BLIINDS gives higher correlation results; however, the results of Video-BLIINDS algorithm are generated using all possible combinations of 80% train and 20% test splits. Moreover, Video-BLIINDS do not consider bit rate and PLR features in coordination with the spatial and temporal information characteristics of a video sequence in its quality evaluation. It only uses spatial and temporal features of a video sequence in the DCT domain. However, STN-VQM performs the quality assessment in a hybrid way by combining the spatial information, temporal information, bit-rate, and PLR, which are all significant parameters for the HVS judgment of a video sequence. For all data in the LIVE VQA dataset, DMOS$_{ALL}$ gives competitive results even though the MOVIE and Video-BLIINDS outperform all algorithms in FR and NR cases. The arguments for the difference in test procedure and features considered for the Video-BLIINDS are also valid in this case.

We also use the EPFL-PoliMI VQ dataset for comparisons. There are 156 video bit streams in this dataset. 78 of the bit streams are in Common Intermediate Format (CIF) resolution and the other 78 bit streams are in 4CIF resolution. Both the CIF and 4CIF videos are encoded using MPEG-4 AVC/H.264 standard Joint Model (JM) 14.2. In this dataset, the video bit streams are also corrupted by dropping RTP packets and introducing packet loss values of 0.1%, $0.4\%, 1\%, 3\%, 5\%$, and 10%. It is important to mention that all of these video bitstreams are used only to evaluate the proposed metric which is trained in LIVE VQA dataset, i.e. none of the EPFL-PoliMI video bitstreams are utilized in the training of the proposed metric. Results are shown in Table 12.

We also evaluated DMOS$_{ALL}$ on subjective data collected by the Image Group of Instituto de Telecomunicacoes, Instituto Superior Tecnico (IT-IST) [Multimedia Signal Processing

Table 10: Comparison with FR metrics (lighter coloured boxes denote better results)

Method	*Type*	*H.264*	*All data*	*H.264*	*All data*
		PCC		**SROCC**	
PSNR	**FR**	0.4385	0.4035	0.4296	0.3684
VSNR	**FR**	0.6216	0.6896	0.6460	0.6755
SW-SSIM	**FR**	0.7206	0.5962	0.7086	0.5849
MS-SSIM	**FR**	0.6919	0.7441	0.7051	0.7361
VQM	**FR**	0.6459	0.7236	0.6520	0.7026
MOVIE	**FR**	0.7902	0.8116	0.7664	0.789
DMOS$_{all}$	**NR**	0.8122	0.6730	0.8026	0.6697

Table 11: Comparison with NR metrics (lighter coloured boxes denote better results)

Method	*Type*	*H.264*	*All data*	*H.264*	*All data*
		PCC		**SROCC**	
WBQM	**NR**	0.524	-	0.563	-
C-VQA	**NR**	0.793	-	0.772	-
LapPyr	**NR**	0.911	-	0.940	-
DVQPM	**NR**	0.967	-	0.963	-
Zero-shot Pred.	**NR**	0.778	0.62	0.777	0.604
Video-BLIINDS	**NR**	0.893	0.881	0.839	0.759
DMOS$_{all}$	**NR**	0.812	0.673	0.803	0.669

Group - IST, 2015]. There are video streams with various video contents at CIF spatial resolution with corresponding subjective scores in IT-IST VQA dataset. These video streams are encoded with MPEG-4 AVC/H.264 at various bit rates ranging from 32 to 2048 kbit/s. There is no packet loss in the utilized IT-IST VQA dataset. The proposed metric gives a score of PCC=0.87 and SROCC=0.9 on this dataset, again when all the training is done using the LIVE VQA dataset.

7.4 Software implementation

The methods presented in this section are included in the image and video analysis services of the Extractor component as described in detail in Section 10. All methods were implemented using C++, with the exception of the Video Quality Assessment (VQA) one, which is based on Matlab. In all cases parallel processing is used in order to accelerate the execution of our methods. The software technical details are presented in Table 13.

Table 12: Results on EPFL-PoliMI VQ dataset (lighter coloured boxes denote better results)

Method	**PCC**	**SROCC**
PSNR	0.793	0.800
VSNR	0.894	0.895
MS-SSIM	0.915	0.922
VQM	0.843	0.838
SSIM	0.678	0.677
VIF	0.749	0.740
MOVIE	0.930	0.920
$DMOS_{ALL}$	0.848	0.906

Table 13: Software technical details for the ForgetIT quality and aesthetic quality assessment (image and video)

Functional description	Quality and Aesthetic quality assessment
Input	An image or an image collection or a video (coded in MPEG-4 AVC/H.264)
Output	Vector of scores indicating if the media item is of low or high quality
Limitations	For VQA: MCR version (MATLAB Compiler Runtime) 8.1, rtpdump, MPEG-4 AVC/H.264 decoder
Language/technologies	C++ (Microsoft visual studio 2013 compiler), Matlab
Hardware Requirements	Multicore implementation; needs processor that has more than one logical CPU core
OS Requirements	Windows

7.5 Conclusions

The Image Quality Assessment method presented in [Solachidis et al., 2015] used four quality measures in order to calculate the quality of an image. In this deliverable we introduce four new measures, which are based on photographic rules, and quantify the aesthetic quality of the image. This method, called Image Aesthetic Quality (IQA) assessment, can contribute to achieving project's aims in terms of multimedia preservation and photo collection summarization. In addition, both methods are expanded to videos aiming to make a decision about the video's aesthetic and non-aesthetic quality.

8 Image/video near duplicate detection and condensation

8.1 Problem statement

The initial Near Duplicate Detection (NDD) method of ForgetIT was introduced in [Solachidis et al., 2015], aiming to support the photo collection summarization algorithm. It identifies groups of very similar media in a large corpus and considers the presence of near-duplicates as an indication of the importance of the scene/object they represent. The procedure followed in [Solachidis et al., 2015] is: 1) local feature descriptors (e.g., SIFT) are extracted, 2) a visual vocabulary is constructed using k-means and the local feature descriptors are encoded into a single vector for each image using VLAD encoding, 3) an index is constructed on VLAD vectors using a Randomized KD forest to find k-nearest neighbors, 4) and finally the geometric consistency of keypoints of matched descriptors is checked, only on nearest neighbors of each image.

In this section we present:

- an updated image NDD method that utilizes features based on Deep Convolutional Neural Networks (DCNN) (The DCNN-based features extracted with the help of DCNN, as described in Section 6.2.1).
- a video near duplicate detection (VNDD) method, which can detect near duplicate videos under various transformations/distortions, and videos which are fractions of others in a large corpus.
- a deduplication method that can be applied in any type of data and can detect and eliminate duplicate copies of repeating data.
- an updated version of the image clustering method presented in [Solachidis et al., 2015], which utilizes DCNN-based features instead of local ones and improves the execution time.

The general framework of the VNDD method can be summarized in four stages, similarly to [Liu et al., 2013]: 1) extracting features from videos, 2) constructing a compact and robust signature from the features for each video, 3) constructing an index on the signatures for fast similarity search and, 4) comparing the signatures to retrieve near-duplicate videos. In [Zhao et al., 2009], the features extracted are categorized in global features (color histograms, ordinal signtures), and local features (keypoint-based approaches). More recently, authors in [Liu et al., 2013] used a finer categorization of features, namely video-level global signatures, frame-level global signatures, frame-level local signatures, spatio-temporal signature and visual-temporal network signatures. Generally, global signatures are fast to compute and compact but they lack robustness to geometric and photogenic transformations. On the other hand, local signatures are computationally costly, especially the frame-level local signatures where descriptor keypoints are matched and their geometric information is used to verify similar scenes. Such signatures are usually extracted only on representative frames of shots or the computations take place last in a coarse-to-fine framework. Various indexing algorithms are proposed in the literature such as trees, 1-dimensional transformations, locality sensitive hashing, inverted files, etc. [Zhao et al., 2009, Liu et al., 2013].

Concerning the generic Data Reduction (DR) algorithms, we focus on Data Compression (DC) algorithms, which detect sequences which are repeated often, and represent those sequences using less bits, and on Data Deduplication (DD) algorithms, which detect much larger sequences and eliminate duplicate copies of repeated data. Deduplication methods save storage space by reducing the amount of replicas of each data unit to the bare minimum required in order to maintain its integrity over time. Deduplication is typically performed either at a full-file granularity or at a chunk-level. In the personal preservation context, there are many cases where different users want to preserve identical copies of image or movie files that they have downloaded from the same source. In the organizational preservation context, there are cases where hundreds of users want to preserve the exact same email attachment. There are also many interesting use cases where great storage savings are possible only through chunk-based deduplication methods (which are not covered by the full-file deduplication methods). In the organizational preservation context, those include preservation of Virtual Machine (VM) images. Finally, deduplication is usually limited to a single system since, at very large scales, is hard to achieve and typically requires very large RAM resources in order to perform in a timely manner. Unfortunately, it is not always clear that data is stored redundantly, and whether repetition also exists across different volumes, pools, systems or data centers. Therefore, it is highly advisable to estimate beforehand whether there exists data redundancy at a large scale, before actually placing this data in a system that supports deduplication (and invests resources in doing so). Our study is aimed to tackle this problem and provide innovative solutions to this task.

8.2 ForgetIT approach

8.2.1 Near duplicate detection for still images

The first two steps of the previous version of our method, as reported in the previous subsection, are replaced by the DCNN-based features extracted using the Caffe framework [Jia et al., 2014] and the $22-$layer GoogLeNet DCNN pre-trained model [Szegedy et al., 2014]. The GoogLeNet model, besides the main classifier, uses two auxiliary classifiers (with *softmax loss* as the classifier, predicting the same 1000 classes as the main classifier). We choose to extract the *loss1/classifier* and *loss3/classifier* layers output resulting in two $1000-$ dimensional vectors. The $2000-$ dimensional concatenated feature vector is used to represent each image. We perform PCA to reduce its dimensionality to 300. We then construct an index on the reduced-dimensionality DCNN-based features using a Randomized KD forest, to perform an approximate kNN search of each image. The nearest neighbors of each image are refined by extracting ORB descriptors [Rublee et al., 2011] and checking the geometric consistency of keypoints of matched descriptors using Geometric Coding (GC) [Zhou et al., 2013]. To directly find groups of near-duplicate images in a collection, instead of treating each image as a query to the collection, we consider the similarity matrix of all images in the collection as a weight matrix of a graph and find the strongly connected components of the graph. Each strongly connected component is a sub-graph in which every node is connected to every other node in the sub-graph. Therefore a strongly connected component is a group of near-duplicate images.

8.2.2 Near duplicate detection for videos

Our VNDD method has three main stages: 1) feature extraction, 2) coarse similarity assessment and 3) fine similarity calculation and temporal alignment. In the first stage, we select 2 frames per second and we follow the same procedure described in subsection 8.2.1 for image NDD in order to extract a DCNN-based feature vector for each frame. The row vectors of each analyzed frame are horizontally concatenated to create the matrix signature of each video. The height of the signature matrix is fixed at $2000-$elements (the length of the feature vector), while the width is variable and proportional to the video's length. In the second stage, for each video we compute the mean value of each prediction class of GoogLeNet by averaging the last 1000 rows of the signature matrix. The produced signature vector can roughly indicate the presence of certain classes throughout the video. Constructing an index on the signature vectors using Randomized KD forests, and performing a kNN search we can quickly discover near-duplicate candidate videos and exclude from the final stage non-near-duplicates. In the third stage, we employ a $2-$dimensional normalized cross-correlation method on signature matrices of near-duplicate candidate videos to compute their similarity and accurately detect their similar parts.

8.2.3 Duplicate data identification in large scale environments

Estimating the duplication across volumes may appear to be a simple task but in fact is far from trivial when large amounts of data are at hand. Clearly the naive approach of going over the entire data and simulating the exact deduplication process is not a valid approach due to the aforementioned process that is extremely expensive in memory and computation resources. One approach is to investigate only a small part of the data and extrapolate from here about the entire data set. However, this turns out to be a very inaccurate approach when small samples are used. The problem stems from the fact that duplications can happen at a global scale and finding out, for example, whether all chunks have appeared once or twice in a data set is extremely hard to do.

An alternative approach is to scan the entire data sets with the challenge being to give good estimations with low memory usage. Such low-memory solutions were presented in [Harnik et al., 2012, Xie et al., 2013]. These works connect between this problem and the problem of counting distinct elements in a population, a problem that has been extensively studied over the years (for example, [Flajolet and Martin, 1985, Alon et al., 1999]).

The first work is a two-pass solution and the second follows the method of Gibbons for distinct elements counting [Gibbons, 2009].

Our study is aimed at providing accurate lightweight methods for identifying duplication potential in large data sets with a focus on usability and suitability to being applied in a large scale and highly distributed storage system. The solution that we implemented follows a different approach borrowed from the world of streaming algorithms (presented in [Bar-Yossef et al., 2002]). We provide a framework for estimating the duplication levels both within and across a large number of volumes and repositories. Another goal is to understand the concrete accuracy guarantees of this method, as opposed to theoretical and asymptotic guarantees provided in the literature. Our work includes both analytical statistical guarantees as well as empirical evaluations on real life data sets.

An additional challenge is how to combine local compression (zip style) into the equation. This is important due to the fact that deduplication and compression are often applied together and their joint benefit is not necessarily clear from the benefits of each one separately. We integrate compression benefits estimation into the same framework, providing a complete data reduction estimation mechanism.

In general, our technique is based on scanning volumes and generating *sketches* of these volumes. These are short data snippets that serve as short summaries and indicators of the richness of the dataset (number of different chunks that appeared in the data set). The estimation method is broken into two main phases:

- **Scan phase:** Scan a volume and generate a sketch for this volume. From this sketch we produce an estimation of the data reduction potential (deduplication and compression) for this specific volume (this only considers deduplication within the specific volume).

- **Merge phase:** Take a number of sketches and merge them to produce a joint sketch. From this sketch we produce an estimation of the joint data reduction potential across all of the volumes of the corresponding sketches. This takes into account cross volume deduplication. The merge methods can be run in many different combinations to provide a duplication estimation across pairs, triples or various clusters of volumes.

Note that each scan run is independent and can be run in parallel with other scans. Moreover, some volumes can be scanned after others, and after merges of the initial runs have been studied. The actual scan follows a very simple algorithm (following [Bar-Yossef et al., 2002]). Each data chunk is read from the storage and the data is then hashed using a cryptographic hash function (e.g., we used the SHA1 hash function which has a 20 byte output). The hash output is then treated as a number between 0 and 1. The sketch will contain the k smallest hash values that were observed in the volume. The algorithm to create the sketch is simple and basically maintains a list of the current k smallest hashes throughout the scan. It follows the steps below:

1. Read a new chunk from disk.

2. Compute the hash value of this chunk.
3. Compare the hash value to the current k^{th} smallest hash.
   - If it is larger then ignore the new hash.
   - If it is smaller then insert the new hash in to the k smallest list (and throw out the largest of the previous list).
4. While there are still chunks to scan, go to step 1.

Once a sketch is achieved, the estimation is produced as follows: Let v_i denote the value of the i^{th} smallest hash (recall that these are values between 0 and 1). The estimate on the number of distinct chunks D in the entire dataset is:

$$\widehat{D} = \frac{k}{v_k} + 1$$

The Merge function simply takes n sketches and creates a joint list of the k smallest hashes that appeared in all of the lists combined.

In order to combine compression ratios into the method, we add a compression ratio estimate for each of the chunks that appear in the list of k minimal hashes. This means that the compression task needs to be performed only for chunks that make it into the k minimal list at any time during the scan. This number is much smaller than the entire scan and this is crucial for the scan performance, since compression is typically a heavy operation.

8.2.4 Image/video collection sub-event clustering

Following the approach of image clustering using time and geolocation metadata presented in [Solachidis et al., 2015], which used several data dimensions, in this deliverable we experiment with three more approaches. The approach that achieves the best performance is selected for image collection sub-event clustering and is adapted for also handling video collections.

Multi-criteria clustering (MCC): This approach is divided into three stages: 1) In the first stage, we sort the images based on their capture time. We compute the temporal distance of each image to the next image. The timeline of the temporally sorted images is split where consecutive images have temporal distance above the mean of all temporal distances. 2) During the second stage, geolocation information is used to further split clusters of images, considering the case of concurrent sub-events at different locations. We compute all images' pairwise capture location distances using the haversine distance function. Using K-means, we cluster the pairwise capture location distances into two clusters. The cluster with the lowest mean of distances ($m1$) presumably signifies images captured in the same sub-event while the cluster with the highest mean of distances presumably corresponds to images captured in different sub-events. We check the pairwise capture location distances for the images within each cluster of the first stage, and images with distance more than $m1$, are moved to a new cluster. 3) In the third stage, clusters are merged using time and geolocation information. If the temporal distance of two clusters (i.e. the temporal distance between the last image of one cluster and the first image of the next cluster) is smaller than the mean of the intra-cluster temporal distances, these two clusters are merged. We also merge temporally neighboring clusters whose geolocation difference is less than $m1$. For the clusters that do not have geolocation information, the merging is continued by considering the concept scores similarity, essentially checking if pairwise distances of images of two clusters are below an empirically set threshold.

Potential-based Hierarchical Agglomerative (PHA) clustering: We extract a concept vector for each image of the collection, following the approach described in Section 6.2.1, and we append to it the capture time of each image if this information is present in the metadata. To group the images of the collection and divide it into sub-events, we do the following:

- We construct a similarity matrix of the feature vectors.

- We weight the similarity matrix with the inverse of the images physical locations distance, so that similarity is increased for images with close capture locations.

- We perform the clustering method proposed in [Lu and Wan, 2013] on the weighted similarity matrix.

Strong Components Discovery (SCD): This approach follows the procedure of the PHA method with the difference that at the final step we consider the weighted similarity matrix of all images as an adjacency matrix of a graph and find the strongly connected components of the graph. The groups of images that represent the sub-events of the collection are the extracted strong components.

In order to adapt the aforementioned approaches to video collections we add a preprocessing step of video temporal segmentation as described in subsection 5.2.3 and treat each extracted representative frame of a shot as an image. The evaluation results of the presented methods are shown in Section 8.3 and support the conclusion that the MCC is the best approach for image collection sub-event clustering, since it gives the best results in terms of F-mesaure and Jaccard index. However, we choose the PHA method to perform the sub-event clustering on video collections since the MCC method requires capture time information.

8.2.5 Advances in comparison to the previous version

Comparing the image NDD method of [Solachidis et al., 2015] with the current one, the advantages can be summarized as follows:

- Faster execution times: Despite the computational complexity of DCNNs, newest advances provide faster feature extraction times than local features descriptors (e.g., SIFT). The Caffe framework can optionally run on GPU yielding even faster performance.
- Better overall accuracy: The semantic nature of the employed DCNN-based features provides good recall under many transformations, while the application of the geometric verification step only on very similar images excludes erroneously retrieved near-duplicates without introducing excessive computational complexity.
- Better scalability: Our older image NDD version was based on encoding local features. Local features and their keypoints' geometric data (coordinates, scale, rotation) were required to be kept in memory for the last stage of geometric verification. On the contrary, in our new approach we produce a feature vector directly for each image, thus the memory footprint of the newest version is much smaller and the method scales better for a larger number of images.

Regarding image collection sub-event clustering, the use of DCNN-based features instead of the local ones assists in terms of accuracy, while the method's improvements aim at the execution time. Additionally, for NDD a new method that is also applicable to videos is presented, while for image sub-event clustering the introduced method is extended in order to deal with videos as well.

8.3 Experimental evaluation and comparison

8.3.1 Near duplicate detection for still images and videos

For the evaluation of image NDD we used the California-ND [Jinda-Apiraksa et al., 2013], UKBench [Nistér and Stewénius, 2006], INRIA Copydays [Jégou et al., 2008] and INRIA Holidays datasets. In successive experiments, we used each image in these datasets as a query image, and for each query, we compared the retrieved near-duplicate images to the available ground-truth, evaluating precision, recall and mean average precision (MAP). The evaluation results are shown in Table 14, where we can see that high precision and recall are consistently achieved in different datasets and are higher than those of the image NDD version of [Solachidis et al., 2015].

The previous image NDD method described in [Solachidis et al., 2015] extracts SIFT descriptors in a parallel fashion (using the OpenMP library), examining up to 8 images simultaneously. Table 15 shows the feature extraction times using the SIFT feature descriptor (serial and parallel extraction) and the DCNN-based features extraction times

Table 14: NDD performance evaluation

	NDD			NDD (using DCNN)		
	Precision (%)	Recall (%)	mAP (%)	Precision (%)	Recall (%)	mAP (%)
California ND	85.98	81.55	68.41	91.09	87.95	72.85
UK Bench	90.34	72.19	65.22	89.83	89.49	70.30
INRIA Copydays	97.51	80.76	79.87	97.88	85.58	81.90
INRIA Holidays	97.46	63.46	62.09	94.32	74.43	64.04

using the Caffe framework running on CPU, GPU and finally utilizing the nVidia cuDNN-GPU library. All experiments are executed on a Windows 7 OS 64-bit machine, Intel Core (TM) i7-4770K CPU@ 3.50GHz processor, 16 GB RAM, nVidia Geforce GTX 650 with 1GB RAM. DCNN on GPU and DCNN on GPU using cuDNN feature extraction times are calculated using batch size equal to 24. In Table 16 the feature extraction and total execution times for all the aforementioned datasets is reported. It is clear that DCNN-based method is much faster. Total execution times are also much lower, considering the fact that no vocabulary construction or encoding stages are applied in the new image NDD approach.

Table 15: Mean feature extraction times per image

	SIFT (serial)	SIFT (parallel)	DCNN (CPU)	DCNN (GPU)	DCNN (cuDNN)
mean extraction time per image (ms)	412	56	126	32	24

Table 16: Execution times

	NDD		NDD (using DCNN)	
	Feature Extraction Time (s)	Total Execution Time (s)	Feature Extraction Time (s)	Total Execution Time (s)
California ND	33.37	72.33	19.03	27.86
UK Bench	459.36	1333.43	247.26	487.21
INRIA Copydays	54.16	106.27	32.69	51.29
INRIA Holidays	102.89	206.78	64.15	86.89

With respect to VNDD, we used the CC_WEB_VIDEO dataset [Wu et al., 2009], which consists of 13.129 videos related to 24 distinct text queries. Each text query was issued to YouTube, Google Video, and Yahoo! Video. The videos gathered include exact and near-duplicates videos under various photometric variations (color / lighting change), editing changes (logo insertion, borders addition around frames, superposition of overlay text) and content modifications (unrelated frames addition with different content). Table 17 reports the evaluated precision and recall versus the method of [Wu et al., 2009], on the CC_WEB_VIDEO dataset.

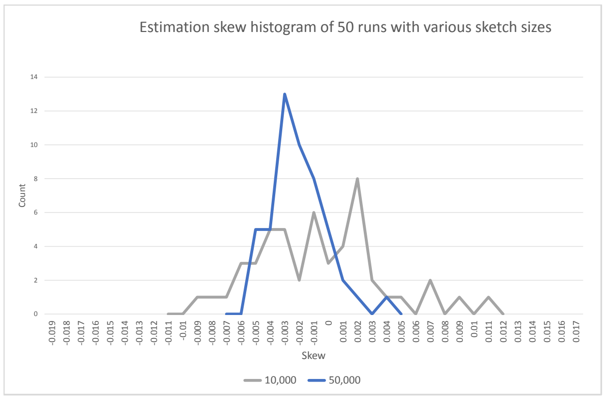
Table 17: DCNN feature extraction time for an image

	Precision (%)	Recall (%)
VNDD	91.1	77.5
[Wu et al., 2009]	87.4	41.2

Note that the VNDD method for videos that are partial-near-duplicates also detects the exact timestamp where the near-duplicate section is.

8.3.2 Duplicate data identification in large scale environments

In order to estimate the method's accuracy, we ran estimations with varying sketch sizes on a local user machine (approximately 0.5 TB in size). In order to capture the statistical variance of the estimation, we used a different hash function (by using a different seed for the SHA1) in each run. The results are seen in Fig. 15. The implementation was done in C++ and tested on various real life data sets. The scan speed depends on the storage speed and the CPU speed for calculating hash functions. Observed speeds vary from 30 to 120 MB per second.



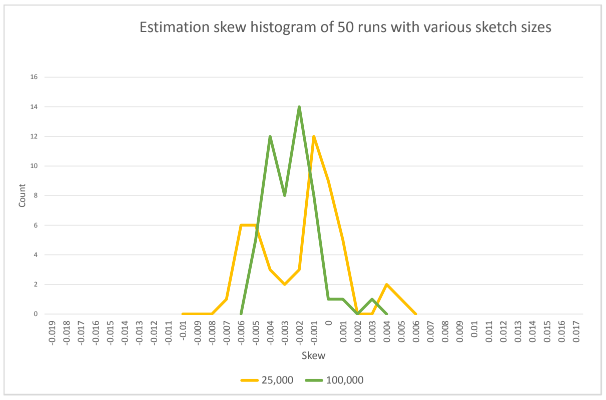


Figure 15: The distribution of the estimation skew compared over various sketch sizes: $k = 25,000$ and $k = 50,000$ in the first graph. $k = 10,000$ and $k = 100,000$ in the second graph. Each graph depicts the number of tests (out of 50) that had a skew in specific ranges. For example, the value at 0 is the number of estimations that had an additive skew of less than 0.0005 from the actual deduplication ratio.

Table 18: Image collection sub-event clustering evaluation results

	Vancouver dataset		London dataset	
	F1 (%)	JI (%)	F1 (%)	JI (%)
MCC	24.6	31.1	34.3	51.6
PHA	24.3	30.7	31.2	45.5
SCD	20.9	26.5	29.6	42.1

8.3.3 Image/video collection sub-event clustering

The aforementioned (Section 8.2.4) approaches of image/video collection sub-event clustering are evaluated on the datasets of MediaEval SEM task [Conci et al., 2014]. The *Vancouver* dataset consists of 1351 photos and captures various sub-events of the Vancouver 2010 Winter Olympic Games, while the *London* dataset consists of 1358 photos and captures various sub-events of the London 2012 Olympic Games. We define two clustering evaluation measures:

1) **F-measure** (F1) can be used to balance the contribution of false negatives by weighting recall and is defined as:
$F1 = \frac{2TP}{2TP+FP+FN}$ where a true positive (TP) is assigned when two images associated to the same sub-event also fall within the same cluster, while for a true negative (TN) two images of different sub-events are assigned to two different clusters). False positives (FP), and false negatives (FN) are calculated accordingly.

2) **Jaccard Index** (JI) is used to quantify the similarity between two datasets. It is defined as the number of unique elements common to both sets divided by the total number of unique elements in both sets. The Jaccard Index is defined as: $JI = \frac{TP}{FP+TP+FN}$

For the Vancouver dataset the MCC method performed marginally better than PHA, while for the London dataset the MCC method performed the best by a large margin. Note that MCC method requires availability of the capture time for all images.

8.4 Software implementation

The algorithms developed for finding near-duplicate items in a collection are included in the overall image and video analysis services of WP4. In the case of videos, the implemented service have some limitations which are presented in details in Section 10. The methods described in this section are C++ implementations. Regarding the near duplicate detection methods both for images and videos, their execution is accelerated if a CUDA-capable NVIDIA graphic card is available. The software technical details of these methods are listed in Table 19.

Concerning the image/video collection sub-event clustering, which is a C++ implementation, a web service is built separately for image and videos. These services constitute the

Condensator component, which is responsible for processing the extracted features of the Extractor component and constructing a summary of the input items. For a detailed description of the web services, please refer to Section 10, while for the method's technical characteristics see Table 20.

Table 19: Software technical details for the ForgetIT duplicate and near duplicate detection methods

Functional description	Duplicate - Near duplicate detection
Input	Still images or videos or text documents
Output	Group of duplicate/near-duplicate items
Limitations	N/A
Language/technologies	C++ (Microsoft visual studio 2013 compiler)
Hardware Requirements	CUDA capable NVIDIA graphic card with compute capability 3.0 or greater
OS Requirements	Windows

Table 20: Software technical details for the ForgetIT sub-event clustering methods (image and video)

Functional description	Sub-event clustering
Input	XML file/files containing features of still images or video keyframes
Output	Group of items which represent a sub-event
Limitations	N/A
Language/technologies	C++ (Microsoft visual studio 2013 compiler)
Hardware Requirements	N/A
OS Requirements	Windows

8.5 Conclusions

We proposed updated methods of the near duplicate detection and image collection sub-event clustering approaches presented in previous deliverables. The approaches that archived the best performance were selected and adapted in order to be used also on video collections. Moreover, we implemented a key building block for supporting duplication identification in a large and distributed system as a step towards reducing the amounts of data being stored in such environments. This method is flexible and can be used for many different of use cases. As a continuation of this effort, we have studied new approaches to estimating duplication when not all of the data is scanned.

9 Associating text and image/video items

9.1 Problem statement

There is an escalation of the multimedia data that users upload and store on-line in recent years. It's crucial for users to retrieve their multimedia files quickly and effectively. For example, a user often wants to find the most relevant photos of his trip by just providing a short textual description of what is looking for, e.g., "We visited the historical attractions of the town, old castles and ancients arches" or "At lunch we went to a nice pub where we drunk beer and tasted traditional dishes". The software must be able to semantically correlate the text with the available images and return only the images that the user wants. We developed a new framework that is able to correlate semantically the input text with the visual concepts that appear in each input image, building on the various multimedia analysis methods presented in the previous sections of this document.

In the literature this is a well known but non-trivial problem, and it is treated as a learning problem. It is called as "zero-shot" or "zero positive example" problem and is formulated as follows: giving as input a textual description and a set of unlabeled images or videos, the algorithm must be able to retrieve the images or videos that are most relevant to the given text. This problem has recently drawn significant attention due to its challenges and its potential applicability. Extensive research efforts have been devoted to multi-label, zero-example (or few-example) classification in images [Mensink et al., 2014]. Similarly, a method for zero-example classification of fMRI data was proposed in [Palatucci et al., 2009]. In [Elhoseiny et al., 2013], a method for predicting unseen image classes from a textual description using knowledge transfer from textual to visual features was proposed.

In the video domain, learning from zero positive examples has been investigated primarily in the context of video event detection. In [Habibian et al., 2014] this problem is addressed by transforming both the event's textual description and the visual content of un-classified videos in a high dimensional concept-based representation, using a large pool of concept detectors; then relevant videos are retrieved by computing the similarities between these representations. In [Wu et al., 2014], multiple low-level representations using both the visual and the audio content of the videos are extracted, along with higher-level semantic features coming from Automatic Speech Recognition (ASR) transcripts, Optical Character Recognition (OCR), and off-the-shelf video concept detectors. This way, both audio-visual and textual features are expressed in a common high-dimensional concept space, where the computation of similarity is possible. E-Lamp [Jiang et al., 2015] is made of four subsystems, one being the off-line indexing component and the rest three composing the on-line event search module. In the off-line module, each video is represented with 4043 visual concepts along with ASR and OCR high-level features. Then, in the on-line search module, the user-specified event description is translated into a set of relevant concepts, called *system query*, which is used to retrieve the videos that are most relevant to the event. Finally, a pseudo-relevance feedback approach is exploited in order to improve the results.

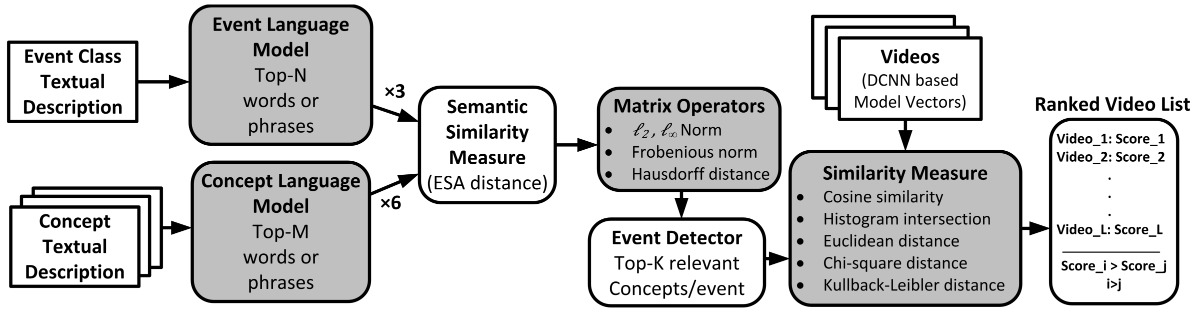

Event Class Textual Description
Event Language Model Top-N words or phrases
×3
Concept Textual Description
Concept Language Model Top-M words or phrases
×6
Semantic Similarity Measure (ESA distance)
Matrix Operators
ℓ_2, ℓ_∞ Norm
Frobenious norm
Hausdorff distance
Event Detector Top-K relevant Concepts/event
Videos (DCNN based Model Vectors)
Similarity Measure
Cosine similarity
Histogram intersection
Euclidean distance
Chi-square distance
Kullback-Leibler distance
Ranked Video List
Video_1: Score_1
Video_2: Score_2
Video_L: Score_L
Score_i > Score_j i>j

Figure 16: The framework for detecting video events using zero positive examples

9.2 ForgetIT approach

As already mentioned in Section 6, high-level (or complex) video event detection is the problem of finding, within a set of videos, which of them depict a given event. Typically, an event is defined as an interaction among humans or between humans and physical objects [Jiang et al., 2012]. We developed a framework for video event detection without using any visual knowledge about the events. This framework can retrieve still images as well. We assume that the only knowledge available, with respect to each event class, is a textual description of it, which consists of a title, a free-form text, and a list of possible visual and audio cues, as in [Younessian et al., 2012], [Jiang et al., 2014].

For linking this textual information with the visual content of video collection, we use a) a pool of 1000 concepts along with their titles and, in some cases, a limited number of subtitles (e.g., concept *bicycle-built-for-two* has the subtitles *tandem bicycle* and *tandem*), and b) a pre-trained detector for these concepts. The latter is a 16-layer pre-trained deep ConvNet [Simonyan and Zisserman, 2014] trained on the ImageNet data [Deng et al., 2009]. In Fig. 16 the structure of our framework is illustrated. Shaded blocks indicate processing stages for which different approaches for the particular stage, are applied.

Given the textual description of an event, our framework first identifies N words or phrases that are mostly related to the event; this word-set is called Event Language Model (ELM). Three different types of ELMs are constructed; The first type of ELM is based on the automatic extraction of word terms solely from the title of an event; in the second type, the visual cues of the event kit are used along with title of the event; and, the third type is the enrichment of the second type with audio cues.

In parallel, for each of the 1000 concepts of our concept pool, our framework similarly identifies M words or phrases: the Concept Language Model (CLM) of the corresponding concept. Six different types of CLMs, depending on the textual information used for each concept (the title of the concept or the Bag of Words (BoW) representation of the top-20 articles in Google or in Wikipedia), as well as the weighting technique (Tf-Idf or none) adopted for transforming this textual information in a BoW representation.

Subsequently, for each word in ELM and each word in each one of CLMs we calculate the Explicit Semantic Analysis (ESA) distance [Gabrilovich and Markovitch, 2007] between them. For each CLM, the resulting $N \times M$ distance matrix expresses the relation between the given event and the corresponding concept. In order to compute a single score expressing this relation, we apply to this matrix different operators, such as various matrix norms (l_2, Frobenius or l_∞) or distance measures (Hausdorff distance). Consequently, a score is computed for each pair of ELM and CLM. The 1000 considered concepts are ordered according to these scores (in descending order) and the K-top concepts along with their scores constitute our event detector. Multiple event detectors are produced as a result of different combinations of the above steps.

Finally, the DCNN-based detectors for the N_c concepts in our concept pool are applied to the videos of the dataset, thus these videos are represented as vectors of concept detector output scores (hereafter called model vectors). The output scores that correspond to the K concepts comprising the event detector are selected and are compared with the corresponding values of the event detector. For this comparison, different choices of a distance function are possible. The following distance functions are used: Euclidean, Histogram Intersection, Chi-square, Kullback-Leibler, and cosine distance. For a chosen distance function, repeating the above process for all videos in a dataset we get a ranked list, in descending order, of the videos that are most closely related to the sought event.

9.3 Experimental evaluation and comparison

9.3.1 Datasets and experimental setup

Our framework is tested on the large-scale video dataset of the TRECVID Multimedia Event Detection (MED) 2014 task (hereafter referred to as MED14). The ground-truth annotated portion of it consists of three different video subsets: the “pre-specified” (PS) video subset (2000 videos, 80 hours, 20 event classes), the “ad-hoc” (AH) video subset (1000 videos, 40 hours, 10 event classes), and the “background” (BG) video subset (5000 videos, 200 hours). Each video in the above dataset belongs to either one of the 30 target event classes, or (in the case of the BG subset) to the “rest of the world” class . The above video dataset (PS+AH+BG) is partitioned such that a training and an evaluation set are created, as follows:

- **Training Set**
  - 50 positive samples per event class
  - 25 related (near-miss) samples per event class
  - 2496 background samples (negative for all event classes)
- **Evaluation Set**
  - ~ 50 positive samples per event class
  - ~ 25 related (near-miss) samples per event class
  - 2496 background samples (negative for all event classes)

As, training samples are not allowed, only the Evaluation Set is used and the related samples are treated as negative samples.

For video representation, approximately 2 keyframes per second were extracted. Each keyframe was represented using the output of a 16-layer pre-trained deep ConvNet network provided in [Simonyan and Zisserman, 2014]. This network had been trained on the ImageNet data [Deng et al., 2009] and provides scores for 1000 concepts. Thus, each keyframe has a 1000-element vector representation. Then, a video-level model vector for each video is computed by taking the average of the corresponding keyframe-level representations.

9.3.2 Video event detection using the event's textual description

The framework allows for different design choices in its various stages. There are 5 stages that can be parameterized: the ELM and CLM information sources selection, the CLM textual vector weighting strategy, the matrix operator, and the distance function selected. Based on the possible choices, 360 different combinations are possible and were tested.

Table 21 presents the 10 best-performing combinations in terms of mean average precision (MAP) across all 30 events. Note that, when the "Title" is used for both ELM and CLM construction (processing stages C1 and C2a), meaning that both ELM and CLMs are represented by a single word or phrase ($N = M = 1$), then all matrix operations result in the same score, making no difference in the final result. This is the reason why a dash sometimes appears in the C3 column of Table 21. Moreover, when "Title" is selected for the construction of the CLMs (C2a processing stage), there is no need for Bag-of-Words encoding, thus no use of weighting technique (Tf-Idf), since no enrichment by searching in Google or in Wikipedia was carried out. This is why there is no choice in the weighting stage (C2b) when "Title" is selected. As can be seen, the enrichment of the CLM through Google search, without Tf-Idf weighting and the usage of as much as possible information for constructing the EML resulted in the best result overall.

Table 21: Top-10 parameter combinations in terms of MAP

C1 (ELM)	**C2a** (CLM)	**C2b** (Weighting)	**C3** (Matrix Operation)	**C4** (Distance)	**MAP**
AudioVisual	Google	No Tf-Idf	Hausdorff	Cosine	**0.1111**
AudioVisual	Google	No Tf-Idf	Hausdorff	Histog_Inter	0.1109
AudioVisual	Google	No Tf-Idf	Hausdorff	Kullback	0.1054
Title	Title	-	-	Histog_Inter	0.1045
Visual	Google	No Tf-Idf	Hausdorff	Cosine	0.1005
Visual	Google	No Tf-Idf	Hausdorff	Histog_Inter	0.0991
Title	Title	-	-	Cosine	0.0988
AudioVisual	Google	Tf-Idf	Hausdorff	Histog_Inter	0.0978
Visual	Google	No Tf-Idf	Hausdorff	Kullback	0.0956
AudioVisual	Google	Tf-Idf	Hausdorff	Cosine	0.0933

We compare our method with a state-of-the-art system, the E-Lamp framework, which is described in detail in [Jiang et al., 2015]. The E-Lamp system consists of four major subsystems, namely Video Semantic Indexing (VSIN), Semantic Query Generator (SQG), Multimodal Search (MS) and Pseudo-Relevance Feedback (PRF). The VSIN subsystem represents the input videos as a set of low- and high-level features from several modalities. The high-level features, i.e. the result of semantic concept detection, are used as input to the SQG subsystem, in which the textual description of an event class is translated into a set of relevant concepts termed *system query*. The system query is then used in the MS subsystem as input to several well-known text retrieval models in order to find the most relevant videos. These results can be then refined by the PRF subsystem.

As SQG leads to the creation of an event detector using semantic concepts, a correspondence exists (and comparison is possible) with our approach to build an event detector. Similarly, the MS subsystem corresponds to (and can be compared with) our event detection module. We compared with four SQG approaches that are presented in the E-Lamp system. These are: i) Exact word matching, ii) WordNet mapping using Wu & Palmer measure (Wu) iii) WordNet mapping using the structural depth distance in WordNet hierarchy (Path), and iv) Word Embedding Mapping (WEP). Concerning the MS stage, we compared with the following retrieval methods: the Vector Space Model [Younessian et al., 2012], the Okapi BM25, and two unigram language models with Jelinek-Mercer smoothing (LM-JM) and Dirichlet smoothing (LM-DL) respectively [Zhai and Lafferty, 2004].

Table 22 shows the performance, in terms of mean average precision (MAP), of the above combinations in comparison to the best-performing event detector and distances in our method. From this table it is clear that our method for building an event detector outperforms the rest of the compared methods, irrespective of the similarity measure that they are combined with. Out of the event detection creation methods of [Jiang et al., 2015], the *exact word* seems to perform considerably better than the others (but much worse than the proposed method). This is because the concept labels from our concept pool that are most related to an event are often well-represented in the event's textual description, e.g., for the event *Beekeeping*, the word bee is observed 31 times, and this word is directly associated with the concepts *bee* and *bee eater*. The WordNet and WEP mappings on the other hand, are not always successful in finding the semantic similarity between two words. Regarding the compared similarity measures, the VSM and LM-DL generally perform better than BM25 and LM-JM, but the proposed cosine and Histogram Intersection distances are consistently among the top-performing measures. It should be noted that the number of visual concepts used in [Jiang et al., 2015] is significantly greater than the 1000 concepts used throughout our experiments, and other modalities (e.g., audio) are also exploited in [Jiang et al., 2015]; this explains the often higher MAP values that are reported in the latter work.

9.4 Software implementation

The described method is implemented as a web service and integrated at the middleware layer of the overall ForgetIT system. A detailed description of its functionality and

Table 22: Comparison between proposed and compared methods

Event detector creation	Similarity measures [Jiang et al., 2015]				Similarity measures (proposed)	
	VSM	**BM25**	**LM-JM**	**LM-DL**	**Cosine**	**Histogram Intersection**
WordNet - Wu	0.0205	0.0201	0.0250	0.0319	0.0222	0.0318
WordNet - Path	0.0333	0.0221	0.0310	0.0379	0.0359	0.0434
Exact word	0.0833	0.0287	0.0541	0.0568	0.0828	0.0801
WEM	0.0429	0.0232	0.0269	0.0331	0.0427	0.0418
Proposed	0.0912	0.0980	0.0392	0.0993	**0.1111**	0.1109

instruction of use are listed in Section 10. The method's implementation is based on JAVA. Moreover, it connects to other ForgetIT web services (either using the provided online semantic similarity service, or the local copy of it hosted in CERTH). In Table 23, the software technical details of the proposed method are presented.

Table 23: Software technical details for the ForgetIT Associating text and image/video items method

Functional description	Associating text and image/video items
Input	The textual description of a visual event and a collection of images
Output	The images sorted from the most related to the textual description
Limitations	Connection to semantic similarity service
Language/technologies	JAVA
Hardware Requirements	N/A
OS Requirements	Windows

9.5 Conclusions

We developed a framework for zero example event detection. The framework is able to correlate semantically a free-form text with a set of images or videos, and retrieve the most relevant, to the text, images and videos. For achieving this, our method exploits the knowledge by Google search or Wikipedia articles, along with state-of-the-art techniques such as concept-based annotation using DCNNs, as described in previous sections.

10 Overall WP4 analysis components

The analysis methods introduced in this deliverable (updated methods or new) are integrated in the overall ForgetIT system as web services. Following the ForgetIT architecture described in WP8, the components of the Preserve or Forget (PoF) Middleware layer that contain the WP4 analysis methods are the **Extractor** and **Condenstator**. The services developed within WP4 are listed below:

- Extractor component:
  - Image analysis service (IAS): it was introduced for first time in [Solachidis et al., 2015]. In this version, some methods are updated versions of the ones presented in [Solachidis et al., 2015] and others are new methods described in the previous sections of this deliverable.
  - Online Training service: this service implements the method of training a concept detector using web resources for collecting training data, as presented in [Solachidis et al., 2015]. This service is called once for each new concept that needs to be added to the concept pool.
  - Video analysis service (VAS): it is introduced for the first time in this document. It applies all video analysis methods described in this document on a video (one video per call can be processed).
  - Video near duplicate detection: it is introduced for the first time in this document. It takes as input more than one videos, in order to detect near-duplicates among them.
  - Text and image/video association: this service combines textual and image/video analysis results in order to associate a textual description with the images and videos that relate to it.
- Condensator component:
  - Image collection sub-event clustering: it takes the output of the image analysis methods from the Extractor component and clusters the images of the collection.
  - Video collection sub-event clustering: it takes the output of the video analysis methods from the Extractor component and clusters the shots of the videos of the collection.
  - Multi-Document Summarization: takes a collection of related text documents and produces a single summary document.

In the following, a detailed description of all services functionalities and usage instructions are presented.

10.1 Extractor component

10.1.1 Image analysis service

The image analysis service is a web service hosted by CERTH. We use Apache Tomcat[5] and JAVA programming language for building the service. Both GET and POST HTTP

[5] `http://tomcat.apache.org/` (Accessed date: 31-01-2016)

requests are available.

Input description, functionalities and output results. The image analysis service takes as input an image or a collection of images. Specifically, the user should provide a URL address either of a zip file that contains the images of the collection or of an image. If he/she wants to send multiple images which are not archived, then he/she can send the URL addresses of all images separated with special character '$\sim$' for GET method or '\n' for POST method.

The methods implemented in the image analysis service and their results, which are stored in Extensible Markup Language (XML) files, are:

- Image annotation (refer to subsection 6.2.1): it results in two vectors of scores per image, indicating the presence of the concepts, $183-$ and 1000-dimensional respectively. The higher the score the more likely the concept is contained in the image.
- Image quality assessment: the output is a vector of scores for each image corresponding to the examined quality measures, namely blur, contrast, darkness, noise and one total score indicating the total quality score of the image. All the scores have a range from 0 to 1, where 0 indicates high image quality while 1 indicates low image quality.
- Near duplicate detection: it results in N groups of near duplicate images.
- Face detection: it results in F faces retrieved in all images of the collection. Each face region is described with a bounding box which indicates the location of the face in the image.
- Image aesthetic quality: this method results in a vector of scores for each image which corresponds to the examined aesthetic quality measures, namely simplicity, colorfulness, sharpness and pattern and one overall score specifying the total aesthetic quality of the image. As in image quality assessment case, the scores range from 0 to 1 but 1 indicates high image aesthetic quality and 0 indicates low image aesthetic quality.
- Face clustering: it results in FC groups of images with similar faces.

Technical Details - Instructions of Use. The base URL of the web service is:

- GET method: `http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/EXTRACTOR/methodGET?`
- POST method: `http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/EXTRACTOR/methodPOST?`

  Set Request Header:
  - Name: Content-Type

  – Value: application/x-www-form-urlencoded[6]

Required arguments:

- **imagePaths**, the URL address of the image/images (see paragraph 10.1.1)
- **method**
  - `concept`, for image annotation
  - `quality`, for quality assessment
  - `duplicate`, for near duplicate
  - `aesthetic`, for aesthetic quality
  - `faceDetection`, for face detection
  - `faceClustering`, for face clustering
  - `all`, all the above methods will be applied on the images of the collection
  - subset of the above methods, comma-separated (e.g.,`concept, quality, faceDetection`)

Optional arguments:

- **useNoise,** the user can decide whether all four quality measures will be applied on the image/images or just the three of them. Specifically, the Noise measure requires high execution time but its contribution to the final quality score is of low importance. Thus, if useNoise equals to 0 the Noise measure will be omitted. The execution time for 100 images with all four measures is 21,4 seconds while for three measures (without Noise) is 2,24 seconds (default value = 1).
- **storeFLAG,** if the user wants to delete his/her image collection after the processing, the value of storeFLAG should be set to *true*. Otherwise, the image collection will be stored into the server for a short period and then will be deleted (default value = *false*).
- **userID,**
  - if userID is missing or it is set to 0 (zero), then the image collection is provided for first time and the selected image analysis method is applied on it
  - if userID is an integer higher than zero it indicates a previous run. The first time an image collection is provided to the service, it is assigned with an unique integer number (userID). The image collection is stored into CERTH server labeled with this number and if the user wants to re-run an image analysis method on this collection he/she just need to provide this number. Of course the user can choose to delete his/her image collection from our server using the storeFLAG argument.
  - default value = 0

[6]The Request Header specifies how the form-data should be encoded when submitting it to the server.

Error messages:

- If the provided method does not exist or there is spelling error: *Wrong method provided to the Extractor - Valid methods: 'all','concept', 'quality', 'duplicate', 'aesthetic', 'faceDetection', 'faceClustering'*

- If the images are not downloaded (either connection problems or corrupted URL addresses): *The images are not downloaded - please provide the URL addresses of the images*

- If the usedID does not exist: *The userID provided by the user is not correct - No image collection assigned with that userID or the image collection had been deleted*

- if the required arguments are missing: *MISSING ARGUMENT - Please provide required arguments – imagePaths , method*

Examples

- For sending two images for image annotation and quality assessment analysis methods using GET method: `http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/EXTRACTOR/methodGET?imagePaths=http://multimedia.iti.gr:8080/CERTH_BIN/TEST_IMAGES/2.jpg~http://multimedia.iti.gr:8080/CERTH_BIN/TEST_IMAGES/1.jpeg&method=concept,quality`

- For sending an achieved file with several images for aesthetic quality assessment and face clustering methods using GET method: `http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/EXTRACTOR/methodGET?imagePaths=http://multimedia.iti.gr:8080/CERTH_BIN/TEST_IMAGES/extractor_test.zip&method=aesthetic,faceClustering`

10.1.2 Online training service

Online training component is an automatic method of training concept detectors by collecting web images and using them as positive training samples. Then, the trained detectors are automatically included to the image annotation method of the Extractor component and the concept list is updated.

Input/Output description. The online training service takes as input the concept term and an integer number which serves as a unique ID. The outputs are three concept detectors (3×loss, refer to Section 6.2.1).

Technical Details - Instruction of Use. The base URL of the web service is:

`http://multimedia.iti.gr:8080/ForgetITonlineTraining/OnlTrain/GET`

Required arguments:

- **concept**, the name of the concept to be trained
- **userID**, a integer number which serves as the unique id of the concept

Error messages:

- If a concept already exists at the concept list of the image annotation method: *Concept already exists at ForgetIT concept list*.

If a user commits a request to train a concept but for some reason (e.g., bad classification results) wants to remove it from the concept list he/she can commit a request:

`http://multimedia.iti.gr:8080/ForgetITonlineTraining/OnlTrain/remove?concept=&conceptID`

Examples

- For training concept <grass> with id 1000:

  `http://multimedia.iti.gr:8080/ForgetITonlineTraining/OnlTrain/GET?concept=grass&conceptID=1000`

- For removing concept <grass> with id 1000:

  `http://multimedia.iti.gr:8080/ForgetITonlineTraining/OnlTrain/remove?concept=grass&conceptID=1000`

The processing time is about ∼60 to 120 minutes.

10.1.3 Video analysis service

The video analysis service is a web service hosted by CERTH. Cheerypy module[7] is used for setting up the web server and the Python programming language for implementing the service.

Input description, functionalities and output results. One video per call can be processed thus as input the user provides the URL address of a video file.

The methods included in the video analysis service and their results which are stored in XML files are:

[7] `http://www.cherrypy.org/` (Accessed date: 31-01-2016)

- Shot-scene Segmentation: it temporally segments the video in video shots. Each shot is assigned with an id (increasing integer number) and described by its start and end time (minutes.seconds)
- Video annotation: two $323-$dimensional vectors of scores are extracted for each keyframe of the video. The higher the scores the more likely the concepts are shown in the video
- Video aesthetic quality assessment: one aesthetic quality score is calculated for each keyframe and one overall score for the entire video. The higher the values, the better the aesthetic quality of the video/video keyframe.
- Face detection: it results in FV faces extracted from all video keyframes. For each face the shot id that the face is shown and a bounding box indicating the location of the keyframe where the face is depicted are stored.
- Face clustering: it results in FCV groups of keyframes containing similar faces.
- Video (non-aesthetic) quality assessment: one quality score is calculated for the entire video. The higher the value, the better the quality of the video.

Technical Details-Instructions of Use. The base URL of the web service is: `http://multimedia.iti.gr:8001`

For sending a video file for a supported method, commit an HTTP POST request on `http://multimedia.iti.gr:8001/method`, where:

- **method**
  - `shot-scene`, for shot-scene segmentation
  - `concept`, for video annotation
  - `aesthetic`, for aesthetic quality assessment
  - `fdetection`, for face detection
  - `fclustering`, for face clustering
  - `quality`, for (non-aesthetic) quality assessment

The body of the POST request should contain the following data, in json format:

- “video_url” is the URL address (either FTP or HTTP) of the video to be processed
- “login” and ’password’ are the access credentials in case that the HTTP repository that hosts the video requires authentication. In any other case (i.e. non password-protected HTTP repository, or FTP repository with restricted or unrestricted access) these should be set to ’none’. Please note that in case of password-protected FTP repository, these authentication details must be included in the URL of the video (example: ftp://username:password@host:port/path)

- "user_key" is a unique 32-digits access key that is given to the user for getting access to the web service. If this key is wrong the service will respond "Not valid user_key." and will not proceed to the processing of the video

The communication between the web service and the user is synchronous only during the transmission of the call. Then, the service's response to the user is "The REST call has been received. Processing will start as soon as the provided content is downloaded." and the communication turns to asynchronous (i.e. is terminated until the next POST or GET call of the user), enabling the submission of a new HTTP POST request from the same user.

To get the status of the processing, commit an HTTP GET request on `http://multimedia.iti.gr:8001/status/<video_name>`. The response from the service can be one of the following five:

- "VIDEO_DOWNLOAD_STARTED", which means that the download process of the video has started.
- "VIDEO_DOWNLOAD_FAILED", which means that the service couldn't download the video, which can be caused, for example, by a mistyped video URL.
- "METHOD_STARTED", which means that the analysis has started and is in progress (e.g., "SHOT_SCENE_CONCEPT_DETECTION_ANALYSIS_STARTED").
- "METHOD_COMPLETED", which means that the analysis has been completed and the user is able to retrieve the XML files with the results (e.g., "SHOT_SCENE_-CONCEPT_DETECTION_ANALYSIS_COMPLETED").
- "METHOD_FAILED", which means that the analysis has failed (most possibly due to unsupported video codec), so there are no results created (e.g., "SHOT_SCENE_-CONCEPT_DETECTION_ANALYSIS_FAILED").

To get the analysis results, commit an HTTP GET request on `http://multimedia.iti.gr:8001/result/<video_name>_resmethod`, where:

- resmethod, returns the XML file with
  - <video_name>_shots, the shot segmentation
  - <video_name>_scenes, the scene segmentation
  - <video_name>_concepts, the video annotation
  - <video_name>_aesthetics, the aesthetic quality assessment
  - <video_name>_fdetections, the face detection
  - <video_name>_fclusters, the face clustering
  - <video_name>_naesthetics, the (non-aesthetic) quality assessment

Examples Example call using curl[8]:

curl -X POST –data "{'video_url':\ "http://multimedia.iti.gr:8080/CERTH_BIN/TEST_IMAGES/sample.mp4\",'login':\"none\",'password':\"none\",'user_key':\"PsDgZsz2haf8YU4fNEkZiBKDi8ysEkqk'{" http://multimedia.iti.gr:8001/shot-scene-concept

- Response: The REST call has been received. Processing will start as soon as the video file is downloaded.
- Status:

  curl X GET http://multimedia.iti.gr:8001/status/sample
- Results:

  curl -X GET http://multimedia.iti.gr:8001/result/sample_shots

  curl -X GET http://multimedia.iti.gr:8001/result/sample_scenes

  curl -X GET http://multimedia.iti.gr:8001/result/sample_concepts

10.1.4 Video Near duplicate detection service

Although video near duplicate detection service is part of the overall video analysis service constituting the Extractor component, it is implemented independently due to the need of more than one input videos. It is also hosted by CERTH and the cheerypy module is used for setting up the web server.

Input description, functionalities and output results. The user provides the URL addresses of several videos (>2). The videos are then downloaded, the VNDD method is applied on them and finally the detected near-duplicates among them are returned in XML file format.

Technical Details-Instruction of Use. The base URL of the web service is: http://multimedia.iti.gr:8002

For sending a set of video files for VNDD analysis, commit an HTTP POST request on http://multimedia.iti.gr:8002/vndd

The body of the POST request contains the same data as the video analysis service except for the "video_url", where more than one URL addresses of the videos are provided separated by character '+'.

[8]http://curl.haxx.se/ (Accessed date: 31-01-2016)

To get the status of the processing, commit an HTTP GET request on `http://multimedia.iti.gr:8002/status/<callID>`. The responses are those presented in video analysis service, where method is "VIDEO_NEAR_DUPLICATE_DETECTION".

To get the analysis results, commit an HTTP GET request on `http://multimedia.iti.gr:8002/result/<callID>_vndd`, where "callID" callID is an alphanumeric string which is returned at the user after the call.

Examples. Example call using curl:

curl -X POST –data "{'video_url':\ "`http://multimedia.iti.gr:8080/CERTH_BIN/Video_test/part1.mp4+http://multimedia.iti.gr:8080/CERTH_BIN/Video_test/part2.mp4+http://multimedia.iti.gr:8080/CERTH_BIN/Video_test/part3.mp4\`",
'login':\"none\",'password':\"none\",'user_key':\"PsDgZsz2haf8YU4fNEkZiBKDi8ysEkqk'{"
`http://multimedia.iti.gr:8002/vndd`

- Response: The REST call has been received. Processing will start as soon as the video file is downloaded. Your call ID is: 46SHBT
- Status:

  curl -X GET `http://multimedia.iti.gr:8002/status/46SHBT`
- Results:

  curl -X GET `http://multimedia.iti.gr:8002/result/46SHBT_vndd`
- XML example and explanation

  <Video filename="videoX.mp4">
  <match offset="0" offsetHMS="00:00:00">videoY.mp4< /match>
  < /Video>

  where,
  - videoX.mp4 is detected in videoY.mp4 starting at 00:00:00 (HH:MM:SS)
  - Offset is given in seconds.

10.1.5 Text and image association service

The text and image association method combines images with textual description. Both the images and the text should correspond to a specific event. A JAVA web Service is located in a CERTH server, implementing this approach.

Input description, functionality and output results. The user should provide a textual description of the event that he/she attended together with the images that were taken at this event. A URL address of a text file containing the description and a URL address of an archived file containing the images are the required inputs of the service. The method is applied on the input items and at the end an XML file is returned containing the results.

Technical Details-Instruction of Use. The base URL of the web service is:
`http://multimedia.iti.gr:8080/Zero_example_event_detection/EXTRACTOR/GET?`

Required arguments:

- **diary**, the URL address of a text file containing the textual description of the event.
- **imgPath**, the URL address of the archived file containing the images of the event

Error messages:

- If an internal method crashes, thus you need to resend your call: *ERROR CD:An internal method could not be completed - please try again*
- If the text file containing the description cannot be downloaded (wrong URL address or connection problem): *The text file containing the textual description of the event is not downloaded*
- If the text file containing the description is empty: *The txt file containing the description of the event is empty*
- If the images are not downloaded (wrong URL address or connection problem or empty zipped file): *The image file containing the images of the event is not downloaded*

Examples. `http://multimedia.iti.gr:8080/Zero_example_event_detection/EXTRACTOR/GET?diary=http://multimedia.iti.gr:8080/CERTH_BIN/TEST_IMAGES/zeed/text.txt&imgPath=http://multimedia.iti.gr:8080//CERTH_BIN/TEST_IMAGES/zeed/Edinburgh_photos_res.zip`

The XML output contains the filenames of the images sorted from the most related to the text textual description to the less related and a confidence score for each image indicating the relatedness.

10.2 Condensator component

In the previous deliverable [Solachidis et al., 2015] we presented the methods of linguistic simplification, single document summarization and image clustering using time and geolocation metadata, which constitute the Condensator component. Regarding the textual summarization methods a document was processed and shorten/summarized. In this document we update the single document summarization method to multi-document, namely a set of documents is summarized instead of a single one. With respect to images and videos, the method of image clustering takes as input the output the image analysis methods of the Extractor component, uses the extracted features/scores and then groups the images of the collection. For videos, the service takes the output of the video analysis service of the Extractor component and groups the shots of the given video/videos. Thus, we updated the previous web service for still images and subsequently we implemented a new service as part of the Condensator component for videos. The service is a python implementation similar to that of the video analysis service.

10.2.1 Linguistic simplification

Input description, functionalities and output results. The user provides a text document to the service which is simplified using lingustic information to remove redudant words and phrases. The shorter document is returned as a plain text document.

Technical Details-Instruction of Use. The base URL of the web service is: `http://services.gate.ac.uk/forgetit/simplification/` which also provides an interactive web demo of the service.

For API level use, the text to simplify must be sent as a HTTP request to `http://services.gate.ac.uk/forgetit/simplification/service` using one of the following parameter forms:

Parameter	Supported Request	Description
text	GET or POST	Plain text to process
url	GET or POST	The URL of a document to process
file	POST	A file to process

The result of the request will be a plain text document containing the simplified text.

10.2.2 Single document summarization

Input description, functionalities and output results. The user provides a document to the service which is then processed to produce a shorter summary version which is returned to the user as plain text.

Technical Details-Instruction of Use. The base URL of the web service is: `http://services.gate.ac.uk/forgetit/summarization/` which also provides an interactive web demo of the service.

For API level use, the text to summarize must be sent as a HTTP request to `http://services.gate.ac.uk/forgetit/summarization/service` using one of the following parameter forms:

Parameter	Supported Request	Description
text	GET or POST	Plain text to process
url	GET or POST	The URL of a document to process
file	POST	A file to process

The size of the resulting summarization can also be configured using the compression parameter. The size of the summary can be set in one of two ways:

- If the value is between 0 and 1 then this is interpreted as a percentage (i.e. 0.1 is 10%) and the summary is produced by selecting that percentage of sentences from the original document.
- If the value is greater than 1 then it (after rounding to the nearest whole number) is interpreted as the number of sentences from the original document to use as the summary.

The result of the request will be a plain text document containing the summary.

10.2.3 Multi-document summarization

Input description, functionalities and output results. The user provides a set of documents as a single zip file to the service. The zip file is downloaded, unpacked and the documents processed to produce a single summary document. The summary is returned to the user as a plain text file.

Technical Details-Instruction of Use. The base URL of the web service is: `http://services.gate.ac.uk/forgetit/multidoc/` which also provides an interactive web demo of the service.

For API level use a zip file must be sent as a HTTP POST request to `http://services.gate.ac.uk/forgetit/multidoc/service`. The zip file should be supplied as the value of `corpus` param and the request must be encoded as `multipart/form-data`. The result of the post will be a plain text document that is a summary produced by processing all documents within the zip file.

10.2.4 Image collection sub-event clustering

Input description, functionalities and output results. The user provides the URL address of an XML file containing the analysis results of the image collection (output of Extractor component). The XML is downlaoded and the clustering method is applied using the information stored in the XML file. The method's results are also stored in an XML file.

Technical Details-Instruction of Use. The base URL of the web service is:
`http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/CONDENSATOR/?`
(only GET method is available)

Required arguments:

- **extractorOutput**: the URL address of the XML containing the image collection analysis results (output of Extractor component; concept detection required)

Error messages:

- If the XML file cannot be downloaded: *The XML file could not be downloaded. Please check the URL address*.
- If an internal error occurred and the execution crashed: *Error (-1): No output XML file*.

Examples. Sending an XML file of a collection containing 100 images of an event:
`http://multimedia.iti.gr:8080/ForgetITImageAnalysis_v3.1/CONDENSATOR/?extractorOutput=http://multimedia.iti.gr:8080/CERTH_BIN/XML_test/Extractor_output.xml`

XML example and explanation:

```
<Cluster id="1">
<image_filename>laurea Andrea 032.jpg laurea Andrea 034.jpg laurea Andrea 035.jpg laurea Andrea 036.jpg laurea Andrea 037.jpg laurea Andrea 038.jpg</image_filename>
<image_filename_exemplar>laurea Andrea 038.jpg</image_filename_exemplar>
</Cluster>
```

The images of the collection are grouped into clusters.

- `<Cluster id="C">`, contains the unique id (C) of each cluster (sequentially increasing integer).
- `<image_filename>`, contains the filenames of the images of each cluster separated with the special character tab.
- `<image_filename_exemplar>`, contains the filename of one image of each cluster C which is selected as the representative one.

10.2.5 Video collection sub-event clustering

Input description, functionalities and output results. The user provides the URL addressees of one or several XML files containing the analysis results of the videos (the XML files are the outputs of the video analysis service of Extractor component). The XML files are then downloaded, the clustering method is applied using the features provided in them and finally the detected clusters are returned in XML file format.

Technical Details-Instruction of Use. The base URL of the web service is:
`http://multimedia.iti.gr:8088`

For sending a set of video files for condensation, commit an HTTP POST request on `http://multimedia.iti.gr:8088/condensator`

The body of the POST request contains the same data as the video analysis service except for the "video_url" argument which is called "video_xmls" and more than one URL addresses of the XML files containing the analysis results of the videos are provided separated by character '+'.

To get the status of the processing, commit an HTTP GET request on
`http://multimedia.iti.gr:8088/status/<callID>`.
The responses are those presented in video analysis service, where method is "VIDEO_-CONDENSATOR".

To get the Condensator's results, commit an HTTP GET request on
`http://multimedia.iti.gr:8088/result/<callID>_cond`,
where "callID" is an alphanumeric string which is returned at the user after the call.

Examples. Example call using curl:

```
curl -X POST –data ``{`video_xmls':\ ``http://multimedia.iti.gr:8080/CE
RTH_BIN/Video_test/part2_all.xml+http://multimedia.iti.gr:8080/CERTH_B
IN/Video_test/part3_all.xml\'',`login':\``none\'',`password':\``none\''
,`user_key':\``PsDgZsz2haf8YU4fNEkZiBKDi8ysEkqk'{'' http://multimed
ia.iti.gr:8088/condensator
```

- Response: The REST call has been received. Processing will start as soon as the XML files are downloaded. Your call ID is: 46SHBE
- Status:

  curl -X GET `http://multimedia.iti.gr:8088/status/46SHBE`
- Results:

  curl -X GET `http://multimedia.iti.gr:8088/result/46SHBE_cond`

- XML example and explanation

```
<Cluster id="1">
<shot>vid_part3_all:Sh38  vid_part3_all:Sh39 </shot>
<shot_exemplar> vid_part3_all:Sh38 </shot_exemplar>
</Cluster>
```

  where,

  – `<shot>` tag, all shots belonging to a cluster are listed separated with special character 'tab'. (vid_part3_all:Sh38 indicate: the filename of the XML file of the video : shot id)
  – `<shot_exemplar>` tag, contains the representative shot of the this cluster

11 Conclusions

11.1 Summary and conclusions

In this deliverable the final release of the ForgetIT text and visual information analysis techniques for condensation and summarization are presented, based on the requirements and state-of-the-art approaches described in the previous work of [Papadopoulou et al., 2013] and the corresponding first and second releases presented in [Papadopoulou et al., 2014] and [Solachidis et al., 2015] respectively. Several software components performing textual and visual analysis are designed, implemented and evaluated.

11.2 Assessment of performance indicators

In this subsection, a short description is provided for explaining how and to what extent the methods described in this deliverable fulfill the success indicators of the five expected outcomes of WP4, as these are described in the Description of Work (DoW) of the project. Each of the following objectives correspond to each one of the five WP4 tasks.

11.2.1 Textual similarity and redundancy

The success indicators of this objective are the *ability to achieve deep understanding* and the *number of features considered*. With respect to the *number of features considered* the approaches to text similarity and redundancy detailed in this and previous WP4 deliverables use a wide variety of features. This includes both linguistically motivated features, such as part-of-speech information and word lemmas, as well as more semantic information such as that derived from associated Linked Open Data resources. While the approaches outlined perform adequately we hesitate to claim they *achieve a deep* *understanding*. Approaches to similarity detection and redundancy removal on their own are never going to lead to a deep understanding of the texts being processed. They are typically used to aid in the production of summaries or highlight information. This allows users to quickly get an understanding of the document content, but it is more about giving high-level view of the content. A deep understanding can only come from a user reading a document. Minimal summarization (such as the linguistic simplification reported in an earlier deliverable) may however reduce the time taken to read a document and for a user to gain a deep understanding. In summary we feel that the approaches we have developed and deployed help users to gain a deep understanding by helping them to focus on the documents they need to read, rather than the components themselves having a deep understanding and that this is reflected in the way the approaches are used within the ForgetIT use-case tools.

11.2.2 Visual quality, similarity, redundancy

The success indicators of this objective are the *ability to detect undesirable artifacts*, the *image/video similarity assessment* and the *number of information dimensions during clustering*. With respect to the detection of undesirable artifacts, this objective was addressed by employing aesthetic and non-aesthetic quality assessment methods. In [Papadopoulou et al., 2014] a non-aesthetic visual quality assessment method for images was presented, while in this deliverable a method for non-aesthetic visual quality assessment method for videos as well as an aesthetic quality assessment method both for images and videos is introduced. Concerning similarity assessment, we updated the near duplicate image detection method that was initially introduced in [Solachidis et al., 2015] by employing features extracted using DCNN-based methods. Furthermore, in this document we also extended this method to videos. Finally, the deduplication method that has been introduced in this deliverable can also be applied in order to reduct visual related data. The number of information dimensions used in clustering has been gradually increased, rising in this deliverable to two broad dimensions (time, location) and 1000 semantic dimensions, represented by the output of DCNN trained for detecting 1000 different concepts.

11.2.3 Semantic multimedia analysis

The success indicators of this objective are *the ability to detect concepts and complex events*. The ForgetIT concept detection method was further extended in this deliverable, in comparison to the previous versions, by adopting a more accurate image representation by employing DCNN based features. Concept detection has also been extended to video and optimized by using many classifiers applying a cascade approach. It has also been optimized in terms of speed by utilizing a dimensionality reduction technique. Face detection and clustering which is a specific case of concept detection, was also improved by employing facial features extracted by DCNN. Finally, event detection that is able to detect complex events has been implemented and presented in this document. Overall, the ability of the ForgetIT methods to detect concepts and complex-events has evolved significantly in the last year and is on par with the State-of-the-art in this research area, as documented by the experimental evaluation results reported in this document and in the numerous ForgetIT publications.

11.2.4 Information condensation and consolidation

The success indicators of this objective are *the ability to summarize documents and multiple documents* and *the ability to combine the results of the first three objectives presented above (11.2.1-11.2.3) for selecting representative and diverse media*. With respect to single document summarization two methods were developed and presented in previous deliverables. The first method, which simplifies text by removing words or phrases, was initially introduced in [Papadopoulou et al., 2014] and was extended in [Solachidis et al., 2015] to perform a more radical reduction of the text. These techniques were extended to the multi-document case and are presented in the current document. Furthermore, image and video collection clustering methods have been developed. The updated image sub-event clustering method improves the previous version both in terms of execution time and accuracy. Finally, in this document, a method that correlates text with multimedia data is introduced; this method returns the media collection items sorted according to their relatedness to a given text.

11.2.5 Evaluation of information condensation and consolidation

The success indicators of this objective are *the number of internal evaluations on ForgetIT datasets* and *the number of external benchmarking activities in which ForgetIT technologies participated in*. Concerning the number of internal evaluations (either in ForgetIT or in other suitable and more widely used datasets), this is probably higher than originally anticipated, given the large amount of evaluations of ForgetIT methods that are reported in every section of this document, and th even more thorough evaluations that are reported in all the relevant ForgetIT published papers. Given however that there is no established methodology for *counting* these evaluations (e.g., the results reported in Table 6, Section 6.3, or those in Table 9, Section 7.3, should count as one or as multiple evaluations?), we are hesitant to provide an exact number in relation to this success indicator. With respect to external benchmarking activities, we participated with ForgetIT technologies during the last year of the project to:

- the TRECVID 2015 Semantic Indexing (SIN) Task [Markatopoulou et al., 2015a]
- the TRECVID 2015 Multimedia Event Detection (MED) Task [Markatopoulou et al., 2015a]
- the MediaEval 2015 Synchronization of multi-user event-media (SEM) Task [Apostolidis and Mezaris, 2015]

- the 2016 Video Browser Showdown (VBS) event [Moumtzidou et al., 2016]

12 References

[Ahonen et al., 2006] Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041.

[Alon et al., 1999] Alon, N., Matias, Y., and Szegedy, M. (1999). The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147.

[Apostolidis and Mezaris, 2014] Apostolidis, E. and Mezaris, V. (2014). Fast shot segmentation combining global and local visual descriptors. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6583–6587.

[Apostolidis and Mezaris, 2015] Apostolidis, K. and Mezaris, V. (2015). Certh at mediaeval 2015 synchronization of multi-user event media task.

[Arestis-Chartampilas et al., 2015] Arestis-Chartampilas, S., Gkalelis, N., and Mezaris, V. (2015). GPU accelerated generalised subclass discriminant analysis for event and concept detection in video. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM ’15, pages 1219–1222, New York, NY, USA. ACM.

[Argyropoulos et al., 2011] Argyropoulos, S., Raake, A., Garcia, M.-N., and List, P. (2011). No-reference video quality assessment for sd and hd H.264/AVC sequences based on continuous estimates of packet loss visibility. In *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, pages 31–36.

[Bar-Yossef et al., 2002] Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D., and Trevisan, L. (2002). Counting distinct elements in a data stream. In *Randomization and Approximation Techniques in Computer Science*, pages 1–10. Springer.

[Barzilay and Mckeown, 2005] Barzilay, R. and Mckeown, K. R. (2005). Sentence fusion for multi-document news summarization. *Computational Linguistics*, 31:297–328.

[Bolles et al., 2014] Bolles, R., Burns, B., Herson, J., et al. (2014). The 2014 SESAME multimedia event detection and recounting system. In *Proceedings of TRECVID Workshop*.

[Brandão and Queluz, 2010] Brandão, T. and Queluz, M. P. (2010). No-reference quality assessment of H.264/AVC encoded video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(11):1437–1447.

[Bromley et al., 1993] Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R. (1993). Signature verification using a Siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

[C. Tzelepis and Patras, 2016] C. Tzelepis, V. M. and Patras, I. (2016). Video event detection using kernel support vector machine with isotropic gaussian sample uncertainty (ksvm-igsu). In *Proceedings of 22nd International Conference on MultiMedia Modeling (MMM'16)*.

[Chandler and Hemami, 2007] Chandler, D. M. and Hemami, S. S. (2007). Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *Image Processing, IEEE Transactions on*, 16(9):2284–2298.

[Conci et al., 2014] Conci, N., Natale, F. D., and Mezaris, V. (2014). Synchronization of multi-user event media (sem) at mediaeval 2014: Task description, datasets, and evaluation. In *MediaEval 2014 Workshop, Barcelona, Spain*.

[Cunningham et al., 2011] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). Text Processing with GATE (Version 6).

[Damhuis et al., 2014] Damhuis, A., Doan, P., Dobberkau, O., Dörzbacher, M., Goslar, J., Krasteva, V., Niederée, C., Schaffstein, S., Sprenger, S., and Wolters, M. (2014). ForgetIT Deliverable D10.1: Organizational Preservation: Application Mockups and Prototypes.

[Datta et al., 2006] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Computer Vision–ECCV 2006*, pages 288–301. Springer.

[De Simone et al., 2010] De Simone, F., Tagliasacchi, M., Naccari, M., Tubaro, S., and Ebrahimi, T. (2010). A H.264/AVC video database for the evaluation of quality metrics. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2430–2433.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 248–255.

[Dhar et al., 2011] Dhar, S., Ordonez, V., and Berg, T. L. (2011). High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), Conference on*, pages 1657–1664. IEEE.

[Dimitrievski and Ivanovski, 2012] Dimitrievski, M. and Ivanovski, Z. (2012). Fusion of local degradation features for no-reference video quality assessment. In *Machine Learning for Signal Processing (MLSP), IEEE International Workshop on*, pages 1–6.

[Douze et al., 2014] Douze, M., Oneata, D., Paulin, M., Leray, C., Chesneau, N., Potapov, D., Verbeek, J., Alahari, K., Harchaoui, Z., Lamel, L., Gauvain, J.-L., Schmidt, C. A., and Schmid, C. (2014). The INRIA-LIM-VocR and AXES submissions to TRECVID 2014 multimedia event detection. In *NIST TRECVID Video Retrieval Evaluation Workshop*.

[Eden, 2007] Eden, A. (2007). No-reference estimation of the coding PSNR for H.264-coded sequences. *Consumer Electronics, IEEE Transactions on*, 53(2):667–674.

[Eldesouky et al., 2015] Eldesouky, B., Bakry, M., Maus, H., and Dengel, A. (2015). Supporting early contextualization of textual content in digital documents on the web. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1071–1075.

[Elhoseiny et al., 2013] Elhoseiny, M., Saleh, B., and Elgammal, A. (2013). Write a classifier: Zero-shot learning using purely textual descriptions. In *Computer Vision (ICCV), IEEE International Conference on*, pages 2584–2591.

[Farias and Mitra, 2005] Farias, M. C. and Mitra, S. K. (2005). No-reference video quality metric based on artifact measurements. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–141.

[Flajolet and Martin, 1985] Flajolet, P. and Martin, G. N. (1985). Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209.

[Gabrilovich and Markovitch, 2007] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.

[Gallo et al., 2015a] Gallo, F., Ceroni, A., Tran, T., Chen, D., Andersson, I., Greenwood, M. A., Maus, H., Lauer, A., Schwarz, S., Solachidis, V., Papadopoulou, O., Apostolidis, E., Pournaras, A., Mezaris, V., and Goslar, J. (2015a). ForgetIT Deliverable D8.4: The Preserve-or-Forget Framework – Second Release.

[Gallo et al., 2014a] Gallo, F., Kanhabua, N., Djafari-Naini, K., Niederee, C., Rad, P. A., Andersson, I., Lindqvist, G., Nilsson, J., Henis, E., Rabinovici-Cohen, S., Maus, H., Steinmann, F., Mezaris, V., Papadopoulou, O., Solachidis, V., Dobberkau, O., Doan, P., Greenwood, M., Allasia, W., and Pellegrino, J. (2014a). ForgetIT Deliverable D8.1: Integration Plan and Architectural Approach – Updated Version.

[Gallo et al., 2015b] Gallo, F., Niederee, C., Andersson, I., Nilsson, J., Chen, D., Maus, H., Greenwood, M., and Logie, R. (2015b). ForgetIT Deliverable D8.2: The Preserve-or-Forget Reference Model – Initial Model.

[Gallo et al., 2014b] Gallo, F., Niederee, C., Kanhabua, N., Chen, D., Maus, H., Solachidis, V., Damhuis, A., Greenwood, M. A., and Pellegrino, J. (2014b). ForgetIT Deliverable D8.3: The Preserve-or-Forget Framework – First Release.

[Gallo et.al., 2016] Gallo et.al. (2016). ForgetIT Deliverable D8.6: The Preserve-or-Forget Framework – Final Release.

[Gerald et al., 1997] Gerald, S., Amit, S., Mandar, M., and Chris, B. (1997). Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207.

[Gibbons, 2009] Gibbons, P. B. (2009). Distinct-values estimation over data streams. In *Manuscript*.

[Gkalelis et al., 2014] Gkalelis, N., Markatopoulou, F., Moumtzidou, A., Galanopoulos, D., Avgerinakis, K., Pittaras, N., Vrochidis, S., Mezaris, V., Kompatsiaris, I., and Patras, I. (2014). ITI-CERTH participation to TRECVID 2014. In *Proceedings of TRECVID Workshop*.

[Gkalelis and Mezaris, 2015] Gkalelis, N. and Mezaris, V. (2015). Accelerated nonlinear discriminant analysis. *arXiv preprint arXiv:1504.07000*.

[Golub and Van Loan, 2012] Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

[Guangnan et al., 2014] Guangnan, Y., Dong, L., Shih-Fu, C., Ruslan, S., Vlad, M., Larry, D., Abhinav, G., Ismail, H., Sadiye, G., and Ashutosh, M. (2014). BBN VISER TRECVID 2014 multimedia event detection and multimedia event recounting systems. In *Proceedings TRECVID Workshop*.

[Habibian et al., 2014] Habibian, A., Mensink, T., and Snoek, C. G. (2014). VideoStory: A New Multimedia Embedding for Few-Example Recognition and Translation of Events . In *Proceedings of the ACM International Conference on Multimedia*, pages 17–26.

[Hadsell et al., 2006] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, IEEE computer society conference on*, volume 2, pages 1735–1742.

[Harnik et al., 2012] Harnik, D., Margalit, O., Naor, D., Sotnikov, D., and Vernik, G. (2012). Estimation of deduplication ratios in large data sets. In *IEEE 28th Symposium on Mass Storage Systems and Technologies, MSST 2012*, pages 1–11.

[Jégou et al., 2008] Jégou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In Forsyth, D., Torr, P., and Zisserman, A., editors, *ECCV 2008 - 10th European Conference on Computer Vision*, volume 5302 of *Lecture Notes in Computer Science*, pages 304–317, Marseille, France. Springer.

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

[Jiang et al., 2014] Jiang, L., Mitamura, T., Yu, S.-I., and Hauptmann, A. G. (2014). Zero-example event search using multimodal pseudo relevance feedback. In *Proceedings of International Conference on Multimedia Retrieval*, page 297. ACM.

[Jiang et al., 2015] Jiang, L., Yu, S.-I., Meng, D., Mitamura, T., and Hauptmann, A. G. (2015). Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *International Conference on Multimedia Retrieval*.

[Jiang et al., 2012] Jiang, Y.-G., Bhattacharya, S., Chang, S.-F., and Shah, M. (2012). High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval*, pages 1–29.

[Jinda-Apiraksa et al., 2013] Jinda-Apiraksa, A., Vonikakis, V., and Winkler, S. (2013). California-ND: An annotated dataset for near-duplicate detection in personal photo collections. In *Quality of Multimedia Experience (QoMEX), Fifth International Workshop on*, pages 142–147. IEEE.

[Ke et al., 2006] Ke, Y., Tang, X., and Jing, F. (2006). The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition, Computer Society Conference on*, volume 1, pages 419–426. IEEE.

[Keimel et al., 2011] Keimel, C., Klimpke, M., Habigt, J., and Diepold, K. (2011). No-reference video quality metric for hdtv based on H.264/AVC bitstream features. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3325–3328.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*.

[Lin et al., 2012a] Lin, X., Ma, H., Luo, L., and Chen, Y. (2012a). No-reference video quality assessment in the compressed domain. *Consumer Electronics, IEEE Transactions on*, 58(2):505–512.

[Lin et al., 2012b] Lin, X., Tian, X., and Chen, Y. (2012b). No-reference video quality assessment based on region of interest. In *Consumer Electronics, Communications and Networks (CECNet), 2nd International Conference on*, pages 1924–1927. IEEE.

[Liu and Wechsler, 2002] Liu, C. and Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Image processing, IEEE Transactions on*, 11(4):467–476.

[Liu et al., 2013] Liu, J., Huang, Z., Cai, H., Shen, H. T., Ngo, C. W., and Wang, W. (2013). Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys (CSUR)*, 45(4):44.

[Liu et al., 2011] Liu, T., Narvekar, N., Wang, B., Ding, R., Zou, D., Cash, G., Bhagavathy, S., and Bloom, J. (2011). Real-time video quality monitoring. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–18.

[Lo et al., 2012] Lo, K.-Y., Liu, K.-H., and Chen, C.-S. (2012). Assessment of photo aesthetics with efficiency. In *Pattern Recognition (ICPR), 21st International Conference on*, pages 2186–2189. IEEE.

[Lu and Wan, 2013] Lu, Y. and Wan, Y. (2013). PHA: A fast potential-based hierarchical agglomerative clustering method. *Pattern Recognition*, 46(5):1227–1239.

[Luo and Tang, 2008] Luo, Y. and Tang, X. (2008). Photo and video quality evaluation: Focusing on the subject. In *Computer Vision–ECCV*, pages 386–399. Springer.

[Marchesotti et al., 2011] Marchesotti, L., Perronnin, F., Larlus, D., and Csurka, G. (2011). Assessing the aesthetic quality of photographs using generic image descriptors. In *Computer Vision (ICCV), IEEE International Conference on*, pages 1784–1791. IEEE.

[Markatopoulou et al., 2015a] Markatopoulou, F., Ioannidou, A., Tzelepis, C., , Mironidis, T., Galanopoulos, D., Arestis-Chartampilas, S., Pittaras, N., Avgerinakis, K., Gkalelis, N., Moumtzidou, A., Vrochidis, S., Mezaris, V., Kompatsiaris, I., and Patras, I. (2015a). Iti-certh participation to trecvid 2013. In *TRECVID 2015 Workshop, Gaithersburg, MD, USA*.

[Markatopoulou et al., 2015b] Markatopoulou, F., Mezaris, V., and Patras, I. (2015b). Cascade of classifiers based on binary, non-binary and deep convolutional network descriptors for video concept detection. In *IEEE Int. Conference on Image Processing (ICIP 2015)*, Canada.

[Markatopoulou et al., 2016] Markatopoulou, F., Mezaris, V., and Patras, I. (2016). Ordering of visual descriptors in a classifier cascade towards improved video concept detection. In *MultiMedia Modeling Conference (MMM)*, pages 874–885, Florida. Springer.

[Maus et al., 2013] Maus, H., Dobberkau, O., Wolters, M., and Niederée, C. (2013). ForgetIT Deliverable D9.1: Application Use Cases & Requirements Document.

[Maus and Schwarz, 2014] Maus, H. and Schwarz, S. (2014). ForgetIT Deliverable D9.2: Use Cases & Mock-up Development.

[Maus et al., 2014] Maus, H., Schwarz, S., Eldesouky, B., Jilek, C., Wolters, M., and Loğoğlu, B. (2014). ForgetIT Deliverable D9.3: Personal Preservation Pilot I: Concise preserving personal desktop.

[Mavridaki and Mezaris, 2014] Mavridaki, E. and Mezaris, V. (2014). No-reference blur assessment in natural images using fourier transform and spatial pyramids. In *Image Processing (ICIP), International Conference on*, pages 566–570. IEEE.

[Mavridaki and Mezaris, 2015] Mavridaki, E. and Mezaris, V. (2015). A comprehensive aesthetic quality assessment method for natural images using basic rules of photography. In *Image Processing (ICIP), IEEE International Conference on*, pages 887–891.

[McCamy, 1992] McCamy, C. S. (1992). Correlated color temperature as an explicit function of chromaticity coordinates. *Color Research & Application*, 17(2):142–144.

[McLaren, 1976] McLaren, K. (1976). XIII the development of the CIE 1976 (L* a* b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists*, 92(9):338–341.

[Mensink et al., 2014] Mensink, T., Gavves, E., and Snoek, C. G. (2014). COSTA: Co-occurrence statistics for zero-shot classification. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2441–2448.

[Mittal et al., 2014] Mittal, A., Saad, M. A., and Bovik, A. C. (2014). Zero shot prediction of video quality using intrinsic video statistics. In *IS&T/SPIE Electronic Imaging*, pages 90140R–90140R. International Society for Optics and Photonics.

[Moumtzidou et al., 2016] Moumtzidou, A., Mironidis, T., Apostolidis, E., Markatopoulou, F., Ioannidou, A., Gialampoukidis, I., Avgerinakis, K., Vrochidis, S., Mezaris, V., Kompatsiaris, I., and Patras, I. (2016). *MultiMedia Modeling: 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part II*, chapter VERGE: A Multimodal Interactive Search Engine for Video Browsing and Retrieval, pages 394–399. Springer International Publishing, Cham.

[Multimedia Signal Processing Group - IST, 2015] Multimedia Signal Processing Group - IST (2015). dataset, http://amalia.img.lx.it.pt (accessed date: 31-01-2016). *Instituto Superior Tecnico of Instituto de Telecomunicacoes*.

[Ng and Winkler, 2014] Ng, H.-W. and Winkler, S. (2014). A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347.

[Nistér and Stewénius, 2006] Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168.

[Niu and Liu, 2012] Niu, Y. and Liu, F. (2012). What makes a professional video? a computational aesthetics approach. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(7):1037–1049.

[Over et al., 2013] Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A. F., and Quenot, G. (2013). TRECVID 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA.

[Over et al., 2014] Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A. F., and Quenot, G. (2014). An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA.

[Palatucci et al., 2009] Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates, Inc.

[Papadopoulou and Mezaris, 2015] Papadopoulou, O. and Mezaris, V. (2015). Exploiting multiple web resources towards collecting positive training samples for visual concept learning. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 531–534.

[Papadopoulou et al., 2013] Papadopoulou, O., Mezaris, V., Greenwood, M. A., and Logoglu, B. (2013). ForgetIT Deliverable D4.1: Information Analysis, Consolidation and Concentration for Preservation - State of the Art and Approach.

[Papadopoulou et al., 2014] Papadopoulou, O., Mezaris, V., Solachidis, V., Ioannidou, A., Eldesouky, B. B., Maus, H., and Greenwood, M. A. (2014). ForgetIT Deliverable D4.2: Information analysis, consolidation and concentration techniques, and evaluation - First release.

[Pinson and Wolf, 2004] Pinson, M. H. and Wolf, S. (2004). A new standardized method for objectively measuring video quality. *Broadcasting, IEEE Transactions on*, 50(3):312–322.

[Radev et al., 2004] Radev, D. R., Jinh, H., and Budzikowska, M. (2004). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *ANLP/NAACL Workshop on Summarization*.

[Robertson, 2008] Robertson, S. (2008). A new interpretation of average precision. In *Proceedings of the 31st annual international ACM SIGIR Conference on Research and development in information retrieval*, pages 689–690.

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA. IEEE Computer Society.

[Saad et al., 2014] Saad, M. A., Bovik, A. C., and Charrier, C. (2014). Blind prediction of natural video quality. *Image Processing, IEEE Transactions on*, 23(3):1352–1365.

[Saggion, 2008] Saggion, H. (2008). SUMMA. A Robust and Adaptable Summarization Tool. *TAL*, 49(2):103–125.

[Saggion et al., 2003] Saggion, H., Bontcheva, K., and Cunningham, H. (2003). Robust Generic and Query-based Summarisation. In *Proceedings of the European Chapter of Computational Linguistics (EACL), Research Notes and Demos*.

[Saggion and Gaizauskas, 2004] Saggion, H. and Gaizauskas, R. (2004). Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference 2004 NIST*.

[Seshadrinathan and Bovik, 2010] Seshadrinathan, K. and Bovik, A. C. (2010). Motion tuned spatio-temporal quality assessment of natural videos. *Image Processing, IEEE Transactions on*, 19(2):335–350.

[Seshadrinathan et al., 2010] Seshadrinathan, K., Soundararajan, R., Bovik, A. C., and Cormack, L. K. (2010). Study of subjective and objective quality assessment of video. *Image Processing, IEEE transactions on*, 19(6):1427–1441.

[Sidiropoulos et al., 2014] Sidiropoulos, P., Mezaris, V., and Kompatsiaris, I. (2014). Video tomographs and a base detector selection strategy for improving large-scale video concept detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1251–1264.

[Sidiropoulos et al., 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. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(8):1163–1177.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Snoek and Worring, 2009] Snoek, C. G. M. and Worring, M. (2009). Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322.

[Solachidis et al., 2015] Solachidis, V., Papadopoulou, O., Apostolidis, K., Ioannidou, A., Mezaris, V., Greenwood, M., Eldesouky, B., and Maus, H. (2015). ForgetIT Deliverable D4.3: Information analysis, consolidation and concentration techniques, and evaluation - Second release.

[Staelens et al., 2013] Staelens, N., Deschrijver, D., Vladislavleva, E., Vermeulen, B., Dhaene, T., and Demeester, P. (2013). Constructing a no-reference H.264/AVC bitstream-based video quality metric using genetic programming-based symbolic regression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(8):1322–1333.

[Staelens et al., 2010] Staelens, N., Vercammen, N., Dhondt, Y., Vermeulen, B., Lambert, P., Van de Walle, R., and Demeester, P. (2010). Viqid: A no-reference bit stream-based visual quality impairment detector. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pages 206–211. IEEE.

[Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

[Tang et al., 2013] Tang, X., Luo, W., and Wang, X. (2013). Content-based photo quality assessment. *IEEE Transactions on Multimedia (TMM)*, 15:1930–1943.

[Tong et al., 2005] Tong, H., Li, M., Zhang, H.-J., He, J., and Zhang, C. (2005). Classification of digital photos taken by photographers or home users. In *Advances in Multimedia Information Processing-PCM 2004*, pages 198–205. Springer.

[Tzelepis et al., 2015a] Tzelepis, C., Galanopoulos, D., Mezaris, V., and Patras, I. (2015a). Learning to detect video events from zero or very few video examples. *Image and vision Computing, 2015*.

[Tzelepis et al., 2013] Tzelepis, C., Gkalelis, N., Mezaris, V., and Kompatsiaris, I. (2013). Improving event detection using related videos and relevance degree support vector machines. In *Proceedings of the 21st ACM international Conference on Multimedia*, pages 673–676.

[Tzelepis et al., 2015b] Tzelepis, C., Mezaris, V., and Patras, I. (2015b). Linear maximum margin classifier for learning from uncertain data. *arXiv preprint arXiv:1504.03892*.

[Tzelepis et al., 2016] Tzelepis, C., Mezaris, V., and Patras, I. (2016). Video event detection using kernel support vector machine with isotropic gaussian sample uncertainty (KSVM-iGSU). In *MultiMedia Modeling*. Springer.

[Vedaldi and Lenc, 2015] Vedaldi, A. and Lenc, K. (2015). Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM International Conference on Multimedia*.

[Wang and Li, 2007] Wang, Z. and Li, Q. (2007). Video quality assessment using a statistical model of human visual speed perception. *JOSA A*, 24(12):B61–B69.

[Wang et al., 2003] Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. IEEE.

[Winkler and Mohandas, 2008] Winkler, S. and Mohandas, P. (2008). The evolution of video quality measurement: from PSNR to hybrid metrics. *Broadcasting, IEEE Transactions on*, 54(3):660–668.

[Wolf et al., 2010] Wolf, L., Hassner, T., and Taigman, Y. (2010). Similarity scores based on background samples. In *Computer Vision–ACCV 2009*, pages 88–97. Springer.

[Wu et al., 2014] Wu, S., Bondugula, S., Luisier, F., Zhuang, X., and Natarajan, P. (2014). Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2665–2672.

[Wu et al., 2009] Wu, X., Ngo, C.-W., Hauptmann, A. G., and Tan, H.-K. (2009). Real-time near-duplicate elimination for web video search with content and context. *Multimedia, IEEE Transactions on*, 11(2):196–207.

[Xie et al., 2013] Xie, F., Condict, M., and Shete, S. (2013). Estimating duplication by content-based sampling. In *Proceedings of the USENIX Conference on Annual Technical Conference*, USENIX ATC'13, pages 181–186.

[Yang et al., 2011] Yang, C.-Y., Yeh, H.-H., and Chen, C.-S. (2011). Video aesthetic quality assessment by combining semantically independent and dependent features. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, pages 1165–1168.

[Yeh et al., 2013] Yeh, H.-H., Yang, C.-Y., Lee, M.-S., and Chen, C.-S. (2013). Video aesthetic quality assessment by temporal integration of photo-and motion-based features. *Multimedia, IEEE Transactions on*, 15(8):1944–1957.

[Yilmaz et al., 2008] Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008). A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st annual international ACM SIGIR Conference on Research and development in information retrieval*, pages 603–610.

[Younessian et al., 2012] Younessian, E., Mitamura, T., and Hauptmann, A. (2012). Multimodal knowledge-based analysis in multimedia event detection. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pages 51:1–51:8.

[Yu et al., 2014] Yu, S.-I., Jiang, L., Mao, Z., Chang, X., Du, X., Gan, C., Lan, Z., Xu, Z., Li, X., Cai, Y., et al. (2014). Informedia at TRECVID 2014 MED and MER. In *NIST TRECVID Video Retrieval Evaluation Workshop*.

[Zhai and Lafferty, 2004] Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

[Zhao et al., 2014] Zhao, S., Jiang, H., Cai, Q., Sherif, S., and Tarraf, A. (2014). Hybrid framework for no-reference video quality indication over lte networks. In *Wireless and Optical Communication Conference (WOCC), 2014 23rd*, pages 1–5. IEEE.

[Zhao et al., 2009] Zhao, W.-L., Tan, S., and Ngo, C.-W. (2009). Large-scale near-duplicate web video search: Challenge and opportunity. In *Multimedia and Expo, ICME, IEEE International Conference on*, pages 1624–1627.

[Zhou et al., 2013] Zhou, W., Li, H., Lu, Y., and Tian, Q. (2013). SIFT match verification by geometric coding for large-scale partial-duplicate web image search. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):4.

[Zhu et al., 2013a] Zhu, K., Asari, V., and Saupe, D. (2013a). No-reference quality assessment of H.264/AVC encoded video based on natural scene features. In *SPIE Defense, Security, and Sensing*, pages 875505–875505. International Society for Optics and Photonics.

[Zhu et al., 2013b] Zhu, K., Hirakawa, K., Asari, V., and Saupe, D. (2013b). A no-reference video quality assessment based on Laplacian pyramids. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 49–53.