



---

## **Deliverable 7.1: Activities and outcome of the Pilots, first report**

---

Gerhard Rudinger / APA-IT

23/6/2017

Work Package 7: Pilots, Testing and User Feedback

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

Innovation Action

Horizon 2020, Research and Innovation Programme

Grant Agreement Number 687786

Dissemination level	PU
Contractual date of delivery	30/6/2017
Actual date of delivery	30/6/2017
Deliverable number	D7.1
Deliverable name	Activities and outcome of the Pilots, first report
File	InVID_D7.1_v1.0.docx
Nature	Report
Status & version	Final, V1.0
Number of pages	65
WP contributing to the deliverable	WP7
Task responsible	APA-IT
Other contributors	AFP DW
Author(s)	Gerhard Rudinger / APA-IT
Quality Assessors	Markos Zampoglou / CERTH Jochen Spangenberg / DW
EC Project Officer	Miguel Montarelo Navajo
Keywords	Testing, evaluation, user feedback

## Abstract

In WP7, the technologies integrated into the InVID platform and applications are tested and evaluated in various editorial cases and trials. The first section of this document gives an overview of the state-of-the-art testing and evaluation methodology. It summarises the classifications according to the target of testing, the objectives of testing and testing techniques. The available evaluation methods offer different ways of structuring the process of receiving feedback from users. The second section of this document describes the first three test cycles, starting with a general description of the concept of the test cycles. It gives an overview of the components or applications that were tested in the three test cycle. The testing methods that were applied in the test cycles are listed together with the testing method that was used for the components or applications in the respective test cycle. The profiles of the testers from the first three test cycles are provided to illustrate that the feedback was qualified. This is followed by the detailed results of the first three test cycles. This third section is structured according to the different components and applications. For each component or application, the major outcomes of the test cycles are reported together with the absolute number of feedback comments. The main feedback from the testers is listed in detail for each application or component with information on how the feedback has been addressed.

## Table of contents

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
1.1	History of the document .....	6
1.2	Glossary of acronyms.....	7
<b>2</b>	<b>Testing and evaluation methodologies.....</b>	<b>8</b>
2.1	Testing methodology .....	8
2.2	Evaluation methods.....	11
<b>3</b>	<b>Overview of the first three test cycles .....</b>	<b>13</b>
3.1	Objectives of WP7.....	13
3.2	General description of the test cycles.....	13
3.3	Overview of test cycles 1 to 3.....	14
<b>4</b>	<b>Results of the first three test cycles .....</b>	<b>22</b>
4.1	General remarks .....	22
4.2	Video Fragmentation & Annotation Service .....	22
4.3	Web Application for Video Fragmentation & Reverse Image Search .....	26
4.4	Tool for Social Media Retrieval and Topic Detection .....	31
4.5	Context Aggregation & Analysis Service .....	35
4.6	Near Duplicate Detection Service.....	42
4.7	Logo Detection Service .....	45
4.8	InVID Verification Plugin.....	49
4.9	InVID Rights Management Service.....	51
4.10	InVID Verification Application .....	54
4.11	InVID Multimodal Analytics Dashboard .....	57
4.12	InVID Core Platform API .....	61
<b>5</b>	<b>Conclusions and outlook for the next test cycles.....</b>	<b>64</b>
	<b>References.....</b>	<b>65</b>

## List of Figures

Figure 1: Tested components in test cycles 1 to 3 .....	15
---	----

## List of Tables

Table 1: History of the document .....	6
Table 2: Glossary of acronyms .....	7
Table 3: Time plan of the test cycles .....	13
Table 4: Tested components of test cycles 1 to 3 .....	14
Table 5: Use of the different testing methods for the different applications/components .....	16
Table 6: Testers of the user interfaces .....	18
Table 7: Testers of the technical interfaces (APIs) .....	20
Table 8: Testing times of test cycles 1 to 3 .....	21
Table 9: Number of received feedback comments from the different test cycles for the Video Fragmentation & Annotation Service .....	23
Table 10: Number of received feedback comments from the different test cycles for the Web Application for Video Fragmentation & Reverse Image Search .....	26
Table 11: Number of received feedback comments from the different test cycles for the Tool for Social Media Retrieval and Topic Detection .....	31
Table 12: Number of received feedback comments from the different test cycles for the Context Aggregation & Analysis Service .....	35
Table 13: Number of received feedback comments from the different test cycles for the Near Duplicate Detection Service .....	42
Table 14: Number of received feedback comments from the different test cycles for the Logo Detection Service .....	45
Table 15: Number of received feedback comments from the different test cycles for the InVID Verification Plugin .....	49
Table 16: Number of received feedback comments from the different test cycles for the InVID Rights Management Service .....	51
Table 17: Number of received feedback comments from the different test cycles for the InVID Verification Application .....	54
Table 18: Number of received feedback comments from the different test cycles for the InVID Multimodal Analytics Dashboard .....	57
Table 19: Number of received feedback comments from the different test cycles for the InVID Core Platform API .....	61

# 1 Introduction

In WP7, the technologies integrated into the InVID platform and applications are tested and evaluated in various editorial cases and trials. The primary aim is to collect user feedback both on the tools and components themselves, their usability and appropriateness for various tasks, and the results of the system as a whole (e.g. in terms of reliability and accuracy).

Using an iterative approach over the entire project duration (with overall nine validation cycles), the feedback and results obtained from each cycle of the trials are used to improve the subsequent versions of the applications, platform and components. Tests and evaluations are focused on video (breaking) news emerging from social networks and media websites.

This deliverable reports insights gained and user feedback on performance, usability, fitness-for-purpose and functionalities of InVID components and the complete InVID applications/systems. This includes the outlining of positive features as well as current shortcomings. D7.1 covers the test cycles 1 to 3.

## 1.1 History of the document

**Table 1: History of the document**

Date	Version	Name	Comment
09/06/2017	V0.8	Gerhard Rudinger (APA-IT)	Initial version
13-16/06/2017	V0.81	Markos Zampoglou (CERTH), Jochen Spangenberg (DW)	Performed quality review on draft version. Provided comments, changes, additions, proposals for modifications, etc.
20/06/2017	V0.9	Gerhard Rudinger (APA-IT)	Refinements and updates after QA.
23/06/2017	V1.0	Gerhard Rudinger (APA-IT)	Final version

## 1.2 Glossary of acronyms

**Table 2: Glossary of acronyms**

<b>Acronym</b>	<b>Explanation</b>
API	Application Programming Interface
JSON	JavaScript Object Notation
UGC	User Generated Content
UGV	User Generated Video
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
REST	Representational State Transfer
XML	Extensible Markup Language

## 2 Testing and evaluation methodologies

This chapter gives an overview of the state-of-the-art methodologies for software testing and user evaluation. This is not a complete overview of all available methodologies but only of the ones that are relevant to the InVID project.

### 2.1 Testing methodology

Testing methodologies can be classified in many different and often overlapping ways. In this document we follow the classifications and definitions by (Bourque, Fairley 2014), (ISTQB) and by (Karner, Bach, Pettichord 2002). Software testing is usually performed at different levels throughout the development and maintenance processes. Levels can be distinguished based on the object of testing, which is called the target, or on the purpose of testing, which is called the objective (Bourque, Fairley 2014). The classification of testing techniques is based on how tests are generated. From the tester's intuition and experience, the specifications, the real or imagined faults to be discovered, predicted usage, models, or the nature of the application (Bourque, Fairley 2014).

#### 2.1.1 Classification by the target of testing

##### 2.1.1.1 Unit testing or module testing

The testing of a minimal software item that can be tested in isolation.

##### 2.1.1.2 Integration testing

Testing performed to expose defects in the interfaces and interactions between integrated components.

##### 2.1.1.3 System testing

Testing the integration of systems and packages; testing interfaces to external organisations (e.g. Electronic Data Interchange, Internet).

#### 2.1.2 Classification by the objectives of testing

In the following the classification of the tests is done by the objectives of testing. The different types of testing can be performed on various targets (see above).

##### 2.1.2.1 Alpha testing

Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organisation. Alpha testing is often employed as a form of internal acceptance testing.



### 2.1.2.2 Beta testing

Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for commercial off-the-shelf software in order to acquire feedback from the market.

### 2.1.2.3 Acceptance testing

Formal testing with respect to user needs, requirements and business processes, conducted to determine whether or not a system satisfies the acceptance criteria and to enable the users, customers or other authorised entities to determine whether or not to accept the system.

- Operational acceptance testing

Operational testing in the acceptance test phase typically performed in a (simulated) operational environment by operations and/or systems administration staff focusing on operational aspects, e.g. recoverability, resource-behaviour, installability and technical compliance.

- User acceptance testing

Acceptance testing carried out by future users in a (simulated) operational environment focusing on user requirements and needs.

### 2.1.2.4 Smoke testing

A subset of all defined/planned test cases that cover the main functionality of a component or system, to ascertain that the most crucial functions of a program work, but not bothering with finer details.

### 2.1.2.5 Regression testing

Testing of a previously tested program following modifications to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.

### 2.1.2.6 Functional testing

Testing based on an analysis of the specification of the functionality of a component or system.

### 2.1.2.7 Non-functional testing

Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, maintainability and portability.

### 2.1.2.8 Performance testing

Testing to determine the performance of a software product.

- Load testing

A type of performance testing conducted to evaluate the behaviour of a component or system by increasing the load, e.g. the number of parallel users and/or the number of transactions, to determine what load can be handled by the component or system.

### 2.1.2.9 Usability testing

Testing to evaluate the degree to which the system can be used by specified users with effectiveness, efficiency and satisfaction in a specified context of use.

## 2.1.3 Testing techniques

### 2.1.3.1 Ad hoc testing

Testing carried out informally. No formal test preparation takes place, no recognised test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity.

### 2.1.3.2 Exploratory testing

An informal test design technique where the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests.

### 2.1.3.3 Fault-based techniques

With different degrees of formalisation, fault-based testing techniques devise test cases specifically aimed at revealing categories of likely or predefined faults.

### 2.1.3.4 Subject matter expert testing

An informal review in which the reviewers are subject matter experts

### 2.1.3.5 Walkthrough testing

A non-formal review that is led by the developer. The developer guides the participants through the application according to his or her thought process to achieve a common understanding and to gather feedback.

### 2.1.3.6 Scenario testing

A specified sequence of actions is predefined for the execution of a test. The sequence should be realistic and reflect something a customer would actually do. It should also include the testing of several features in a way that are challenging the program.

## 2.2 Evaluation methods

Evaluation methods focus on how to structure the process of getting feedback from users. In this document we follow the suggested classifications from (Jones, Marsden 2006) and (Stone, Jarrett, Woodroffe, Minocha 2005).

### 2.2.1 Quick and dirty

The “quick and dirty” method addresses an evaluation by end users. It is done in an informal setting. The result is anecdotal and unstructured. The goal is to get rapid feedback.

### 2.2.2 Conceptual model extraction

The “conceptual model extraction” method addresses an evaluation by end users in a controlled setting. It is used for new development where the users have no precedent. The goal of this technique is to extract how users interpret a completely new interface, given their existing mental models of how interfaces should work.

### 2.2.3 Direct observation

Direct observation is used for an evaluation by end users. This method is done in a controlled setting and gives qualitative feedback. The users are given specific tasks which they shall solve with the software. The users are observed in how they solve the tasks and what problems they have in solving them.

Observations of the users should also include what the users are thinking as they complete the tasks. A solution for this is the “think-aloud” technique developed by Ericson and Simon. In this technique the users are asked to speak aloud what they are thinking throughout the period of evaluation. As users tend to forget to speak after a while, a variant of this method called “constructive interaction” was developed by Nielsen and Mack. In this variant a second person (for example the observer) prompts the users to explain their actions more fully.

The recording of the observations can be done with the following methods:

- Screen and audio recording  
Benefit: Replaying of the interactions is possible. But time-consuming for re-evaluation.
- Note taking  
Major drawback: the observers must take their eyes off the subject in order to record an event and might miss something.
- Automatic logging  
The software logs all interactions by the users. This leaves the observers with the only task of recording the think-aloud thoughts of the users.

In direct observation it is important not to bias the experiment. It furthermore has to be taken into account that most people react negatively to being observed. It is important to make it

clear in advance that the system is being tested and not the user, and to also explain what will happen and what will be expected from the user.

#### **2.2.4 Interviews**

Interviews are used for an evaluation by end users in a controlled setting. The users are asked to reflect on their experience with the software in their own words. The value of the session depends on the quality of the interviewer. The interviewer should not influence the testers and be able to detect and follow up on any interesting points made in the course of the conversation. Conducting interviews as an alternative to observations can lead to information being missed. People are usually not good at explaining how they have achieved a given task. The recommendation is to use interviews always in conjunction with some other form of observation.

#### **2.2.5 Questionnaires**

Questionnaires have the potential of reaching a very wide audience, are cheap to administer and can be analysed rapidly. As a matter of fact, questionnaires can never be as flexible as an interview and require a lot of effort in design, especially if the users are completing them with no external help. The first step is to define the exact points that are to be found out. Next, questions must be posed in a way that is unambiguous to all possible respondents. Question types include: open-ended, scalar, multi-choice.

### 3 Overview of the first three test cycles

#### 3.1 Objectives of WP7

In WP7, the technologies integrated into the InVID platform and applications are tested and evaluated in various editorial cases and trials. This is done with different user groups. The primary aim is to collect user feedback both on the tools and components themselves, their usability and appropriateness for various tasks, and the results of the system as a whole (e.g. in terms of reliability and accuracy).

Tests and evaluations focus on video (breaking) news emerging from social networks and media web sites, as well as user-generated content provided by a user community built by regional newspapers.

#### 3.2 General description of the test cycles

##### 3.2.1 Time plan of the test cycles

Using an iterative approach (nine validation cycles), the feedback and results obtained from each cycle of the trials are used to improve the subsequent versions of the applications, platform and components. The work of WP7 started in month 7 with preparations for the test cycles. The test cycles themselves started in month 10 (see Table 3).

**Table 3: Time plan of the test cycles**

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Test cycle							Pilots prep.			Test cycle 1	Test cycle 2	Test cycle 3	Test cycle 4	Test cycle 5	Test cycle 6	Test cycle 7	Test cycle 8	Test cycle 9																		

##### 3.2.2 Phases of a test cycle

Each test cycle consists of the following phases:

- Preparations for the testing
  - Clarification which applications/components are included in the test cycle.
  - Deployment of a stable version on a test system.
  - Gathering all necessary access data such as URL, username, password.
  - Documentation of the changes for the different applications/components.
  - Development of testing guidelines.
  - Setup of templates for the test results.
  - Coordination of the dates for testing.
- Testing
  - Verification of the fixed issues from the prior test cycles.

- Testing the different applications/components.
- Filling in the results in the testing templates.
- Follow-ups
  - Merging all results from the different testers.
  - Providing the test results to all partners.
  - Conference call to clear all questions regarding the test results.

### 3.3 Overview of test cycles 1 to 3

#### 3.3.1 Applications tested in test cycles 1 to 3

In general we included as many applications and components as possible in each test cycle. However, any application or component that was not yet ready for testing was skipped in the respective test cycle. This could occur in the case of new applications and components and also for components which had longer development phases and were not yet stable enough for testing. Table 4 gives an overview of the tested components in the test cycles 1 to 3.

**Table 4: Tested components of test cycles 1 to 3**

Applications	Test cycle	1	2	3
Video Fragmentation & Annotation Service		Yes	Yes	Yes
Web Application for Video Fragmentation & Reverse Image Search		Yes	Yes	Yes
Tool for Social Media Retrieval and Topic Detection		Yes	Yes	Yes
Context Aggregation & Analysis Service		Yes	Yes	Yes
Near Duplicate Detection Service		No	Yes	Yes
Logo Detection Service		Yes	Yes	Yes
InVID Verification Plugin		No	No	Yes
Rights Management Service		Yes	Yes	Yes
InVID Verification Application		Yes	No	Yes
InVID Multimodal Analytics Dashboard		Yes	Yes	Partial <sup>1</sup>
InVID Core Platform-API		Yes	Yes	Yes

In Figure 1 all tested components are marked with a red border in the InVID architecture diagram. The unmarked components in Figure 1 were either not ready for testing (i.e. the

<sup>1</sup> The InVID Multimodal Analytics Dashboard took part in test cycle 3 in the walkthrough testing between testers and developers and also partially in the testing for the Tool for Social Media Retrieval and Topic Detection.

Mobile App or the Forensic Analysis Service) or were not directly accessible for tests. The components not directly accessible were tested through the applications containing these components. In particular the components Location Detection and Knowledge Extraction were tested via the Multimodal Analytics Dashboard. Also the different backends and repositories were tested with the according applications and services.

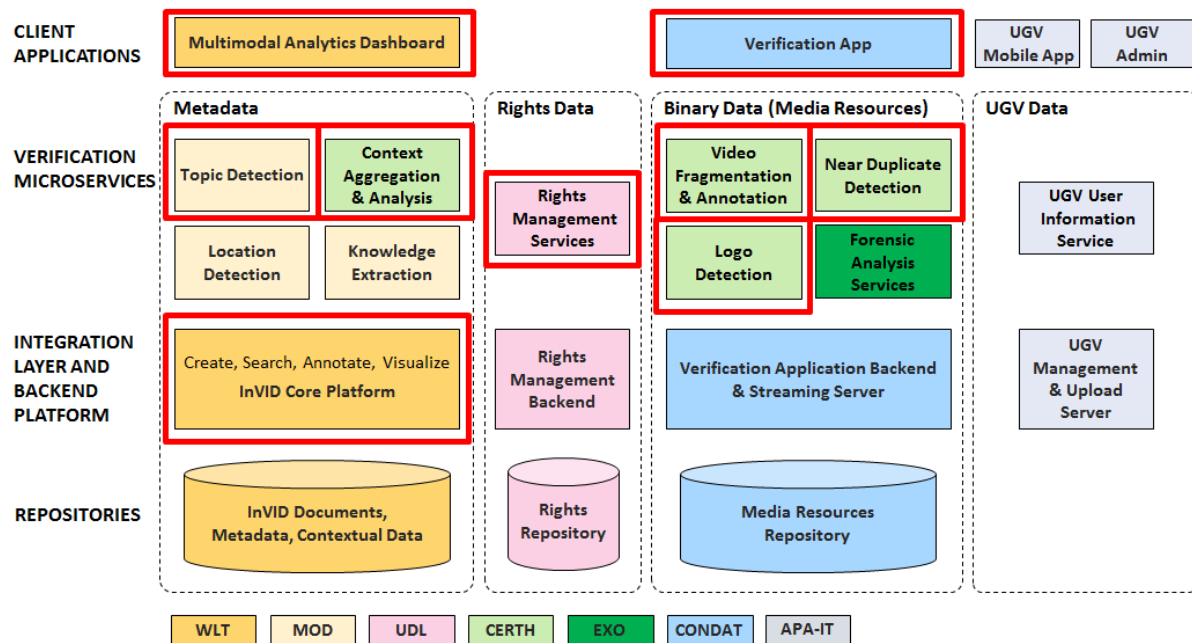


Figure 1: Tested components in test cycles 1 to 3

### 3.3.2 Testing methods used in test cycles 1 to 3

The following testing methods have been used in test cycles 1 to 3:

Targets of testing (see section 2.1.1)

- Integration testing  
Integration testing, especially interface testing, was used in all three test cycles with all services that provide APIs.
- System testing  
System testing was used for all user interfaces and services in all three test cycles.

Remark: The target unit testing is part of the development process and not addressed in WP7.

Objectives of testing (see section 2.1.2)

- Alpha testing  
All applications and components were tested by users within the consortium in all three test cycles.
- Functional testing  
In all test cycles the functionality of the components/applications was tested.
- Non-functional testing

In all test cycles the non-functional aspects of the components/applications such as usability and reliability were tested.

- Performance testing

For all components/applications the performance of the application was tested in the different test cycles. These tests were done qualitatively for the user interfaces, e.g. is the performance good enough for a journalistic workflow, and also quantitatively on the interface level.

- Regression testing

In test cycles 2 and 3 the functionality of the components/applications was retested to ensure that defects had not been introduced as a result of the changes made.

Testing techniques (see Section 2.1.3)

- Exploratory testing

Besides guidelines for testing from the other testing methods, every tester also tested exploratorily in all three test cycles.

- Fault-based techniques

Fault-based techniques were mainly used in the interface tests in all three test cycles. These tests on interface level enable a stable interaction between the different components of the system and thus also a stable basis for all user interfaces.

- Scenario testing

In test cycle 2 the focus was placed on scenario testing of the user interfaces. These tests were based on a typical journalistic workflow for video verification and provided information on how the applications help with these tasks. Scenario testing was also used partially in test cycle 3.

- Walkthrough testing

Test cycle 3 included walkthrough testing in a dedicated session with developers and testers of all user interfaces.

These testing methods and techniques were chosen because they are suitable for providing good feedback on applications and components that are at an early stage. The different testing methods were often used in combination, e.g. functional, non-functional and performance testing was done in one test sequence.

Table 5 gives an overview of which testing method was used for the components in the different test cycles.

**Table 5: Use of the different testing methods for the different applications/components (where TC refers to Testing Cycle)**

	Targets of testing	Objectives of testing	Testing techniques



<b>Application</b>	<b>Integration testing</b>	<b>System testing</b>	<b>Alpha testing</b>	<b>Regression testing</b>	<b>Functional testing</b>	<b>Non-functional testing</b>	<b>Performance testing</b>	<b>Exploratory testing</b>	<b>Fault-based techniques</b>	<b>Scenario testing</b>	<b>Walkthrough testing</b>
Video Fragmentation & Reverse Image Search Service	TC1 TC2 TC3		TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3		
Web Application for Video Fragmentation & Reverse Image Search		TC1 TC2 TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3		TC2	TC3
Tool for Social Media Retrieval and Topic Detection		TC1 TC2 TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3			TC1 TC2 TC3		TC2 TC3	TC3
Context Aggregation & Analysis Service	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC2	TC3
Near Duplicate Detection Service	TC2 TC3	TC3	TC2 TC3	TC3	TC2 TC3	TC2 TC3	TC2 TC3	TC2 TC3	TC2 TC3		
Logo Detection Service	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC2	TC3
InVID Verification Plugin		TC3	TC3		TC3	TC3		TC3			TC3
Rights Management Service	TC1 TC2 TC3	TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC3	TC3
InVID Verification Application		TC1 TC3	TC1 TC3	TC3	TC1 TC3	TC1 TC3		TC1 TC3			TC3
InVID Multimodal Analytics Dashboard		TC1 TC2 TC3	TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC2	TC1 TC2 TC3		TC2	TC3
InVID Core Platform-API	TC1 TC2 TC3		TC1 TC2 TC3	TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3	TC1 TC2 TC3		

The tests were performed manually, and in the case of the interface tests also automatically. The automatic API tests ensure the functionality of the tested APIs and reduce time and effort for retesting the APIs after a change.

### 3.3.3 Testers of the user interfaces

As the components and applications were at an early stage during test cycles 1 to 3, the testing of the interfaces was mainly done by members of the consortium. Nonetheless these members brought in journalistic knowledge. They verified the applications and components from a journalistic point of view and checked if the applications were useful in a journalistic workflow.

**Table 6: Testers of the user interfaces**

Name	Jochen Spangenberg
Company	Deutsche Welle
Job role	Innovation Manager
Domain know-how	Media strategy, digital media, user-generated content for newsgathering, UGC verification, social media, project management, journalism
Technical experience	Normal (end user focus)

Name	Tim Koch
Company	DW
Job role	Innovation Manager
Domain know-how	Project manager, long experience in IT and media-centred research and development
Technical experience	Medium

Name	Ruben Bouwmeester
Company	DW
Job role	Innovation Manager
Domain know-how	UGC verification, social media, software development
Technical experience	Medium to high

Name	Julia Bayer
Company	DW

Job role	1) journalist and editor in DW's social media team (social newsgathering, reporting, fact-checking / verification) 2) Innovation Manager with a focus on research and development of tools and technology for social newsgathering and part-automated / algorithm-supported verification
Domain know-how	Expert in social newsgathering and verification
Technical experience	High

Name	Denis Teyssou
Company	Agence France-Presse
Job role	Medialab R&D editorial manager
Domain know-how	Journalistic
Technical experience	High (for a journalist)

Name	Dominique Ferrandini
Company	AFP
Job role	Journalist
Domain know-how	Journalistic
Technical experience	Medium (for a journalist)

Name	Jean-Michel Leung
Company	AFP
Job role	IT engineer
Domain know-how	Technical
Technical experience	IT engineer

### Additional testers

The InVID Verification Plugin was installed on some 14 journalists' computers at AFP, namely at the Social Networks team, the Technical Editor-in-chief's department and the

Video Editor-in-chief's department, to be tested in a real production environment. Feedback was gathered from these testers. Additionally, journalists of the social media team at Deutsche Welle were involved in providing feedback to the DW core project team on individual components and features.

### 3.3.4 Testers of the technical interfaces (APIs)

**Table 7: Testers of the technical interfaces (APIs)**

Name	Bertrand Goupil
Company	AFP
Job role	IT engineer

Name	Michael Gratzl
Company	APA-IT
Job role	IT engineer

Name	Andrei Turcanu
Company	APA-IT
Job role	IT engineer

Name	Thomas Volpini
Company	APA-IT
Job role	IT engineer

Name	Gerhard Rudinger
Company	APA-IT
Job role	IT engineer

### 3.3.5 Testing dates

Table 8 shows the dates of the testing for test cycles 1 to 3. The testing times were between three and six weeks. This data does not include the preparations and the follow-up phases of a test cycle.

**Table 8: Testing times of test cycles 1 to 3**

Test cycle	Testing times
Test cycle 1	2016-11-04 to 2016-12-15
Test cycle 2	2017-02-21 to 2017-03-16
Test cycle 3	2017-05-12 to 2017-06-02

## **4 Results of the first three test cycles**

### **4.1 General remarks**

The results of the first three test cycles are structured according to the different components and applications. The section for each component/application starts with a short description of the service. This is followed by the major outcome from the test cycles for the component/application and how this feedback was taken into account in the further development of the component/application.

The number of feedback comments received is provided for each component and test cycle. This number of feedback comments also includes duplicate feedback such as the same feedback from different persons or the same feedback from another test cycle. Then the main feedback is listed in detail with information on how the feedback was addressed. The same or very similar feedback from different persons or the same feedback from another test cycle were merged to one item. Minor feedback comments are not listed to keep this document manageable.

During the first three test cycles the focus of the development was on the major feedback regarding functionality, usability and performance. Some feedback from these test cycles has therefore been queued for later development. In addition test cycle 3 ended close to the writing of this document. For this reason there is also feedback listed for issues that are currently still being resolved.

### **4.2 Video Fragmentation & Annotation Service**

#### **4.2.1 Description of the service**

The Video Fragmentation and Annotation Service is a web service (API) that performs a temporal decomposition of a video into three different levels of granularity; scenes (i.e. semantically and temporally coherent segments that correspond to the story-telling parts of the video), shots (i.e. sequences of frames captured uninterruptedly by a single camera) and sub-shots (i.e. sub-parts of a shot with visually discrete content; useful when analysing single-shot videos). Consecutively, it identifies the semantics of the video at the most fine-grained level (either shots or sub-shots, depending on the type of analysed video) by detecting a number of high-level visual concepts after analysing one representative key frame per video fragment.

#### **4.2.2 Major outcome of the test cycles**

As the Video Fragmentation and Annotation Service is a web service all tests were done on interface level. One major outcome of the testing was that the documentation was too complex and some explanations were missing. This was addressed by an improvement of the documentation to make it clearer and easier to understand. Also the missing

explanations were added. Another major point concerned the structure of the API calls. Without knowledge of the type of the video file (single-shot video such as the biggest part of the user-generated videos shared online, or multi-shot such as the edited professional videos of media organizations) a set of calls had to be done to get a qualified result. This test result led to a new structure of the API calls. Now it is no longer necessary to know the type of the video file in advance. A new API call can handle both single-shot and multi-shot videos. This makes the handling of the service much easier. Regarding performance, the speedup of the video analysis was valued very highly.

### 4.2.3 Overview of the feedback

**Table 9: Number of received feedback comments from the different test cycles for the Video Fragmentation & Annotation Service**

Test cycle	Feedback comments
Test cycle 1	21
Test cycle 2	14
Test cycle 3	25

### 4.2.4 Tested API functions

- shot-scene
- shot-scene-concept
- subshot
- subshot-concept
- segmentation
- segmentation-annotation
- results
  - \_shots
  - \_scenes
  - \_subshots
  - \_concepts
  - \_json
- keyframes
  - shots
  - subshots

### 4.2.5 Feedback on documentation

- **Documentation is complex**

**Feedback:** Documentation is complex and not easy to read. Operation descriptions are not very clear. Several reads are required to figure out what the API can really do.

**Resolution:** Improvement of the documentation for test cycle 2 and furthermore for test cycle 3

- **Explanations are missing**

**Feedback:** An explanation of the difference between shot, scene and sub-shot is missing. Also the information how segmentation and segmentation-annotation work with GET Status is missing.

**Resolution:** Improvement of the documentation

**Feedback:** With the simplification of the operations the documentation is now good. Operations are well-described and examples complete the operations descriptions.

- **Swagger documentation suggested**

**Feedback:** A swagger documentation that helps with working on this API is missing.

**Resolution:** open

#### 4.2.6 Feedback on reliability

**Feedback:** In general the API is stable. However, a small set of videos stuck in the processing queue, or downloading of the submitted video for analysis failed.

**Resolution:** under testing and bug fixing

#### 4.2.7 Feedback on performance

**Feedback:** The ratio of the processing time to the video length is quite good (about 1:4 to 1:5). But for longer videos this still seems long for practical purposes and has to be handled in the applications.

**Resolution:** open

#### 4.2.8 Main feedback

- **Structure of the API-calls**

**Feedback:** Operation results are not clear. Some operations give the same results. For example processing a video with shot-scene and shot-scene-concept gives the same results.

In the same way applying multiple operations and using one-way to get the result is uncommon. It would be better to have one operation for all, or provide different ways to get results with regard to operations that have been performed on the video.

**Resolution:** Changes in the API

**Feedback on resolution:** Having two operations for fragmentation is more understandable than previously. The general feeling is that this API has now become very usable and is very promising.

**Resolution:** not required



- **Content type for response messages**

**Feedback:** Response messages do not use the correct content-type. All json content uses content-type: text/html

**Resolution:** open

- **Different calls for multi-shot and single-shot videos are not useful**

**Feedback:** The version of the API for the first test cycle had two different calls, one for professional multi-shot videos and one for single-shot UGV. In most cases it cannot be determined in advance if the video is a multi-shot or a single shot video.

So in the first versions, all applications using the API had to call twice for UGV. The first call was an analysis for shots. If the shot analysis returned only view shots then there had to be a second run for the subshot analysis.

**Resolution:** A new call was introduced which combines the analysis of multi-shot and single-shot videos and performs multi-granular fragmentation (into scenes, shots and sub-shots).

- **Responses in JSON format**

**Feedback:** Requests for processing a video should return statuses in a more convenient format to facilitate integration, therefore returning the status formatted in JSON or XML is more appropriate.

**Resolution:** A service response in JSON format was implemented for a set of different requests.

**Feedback on the resolution:** While some responses have been converted to JSON, others (e.g. the ones reporting the status of the analysis) remain in text.

**Resolution:** open

- **Error responses**

**Feedback:** Error response is not implemented. Only an HTTP 400 is returned when something goes wrong.

**Resolution:** Meaningful error messages have been implemented.

- **No errors for wrong parameters**

**Feedback:** A specific error indication is missing in some error situations (for example if a non-existing identifier is sent).

**Resolution:** fixed

## 4.3 Web Application for Video Fragmentation & Reverse Image Search

### 4.3.1 Description of the service

The Web Application for Video Fragmentation & Reverse Image Search (user interface) allows the user to extract a set of representative keyframes from a video, and to use these key frames to perform reverse image searches with the help of the Google Image Search engine.

### 4.3.2 Major outcomes of the test cycles

A major outcome of the tests concerned the workflow in the user interface. In the first version the video had to be downloaded from a video platform (e.g. YouTube) and then uploaded again to the Web Application for Video Fragmentation & Reverse Image Search. After starting the analysis the link to the result was sent to the testers by e-mail. This forced the testers to switch from browser to e-mail and again to browser. The testers stated that this was too complex for practical use. This feedback was fully accepted and led to a major improvement in the workflow of the user interface. Now it is also possible to input a link to a video platform. In addition, the result of the processing is available directly within the user interface and the user is no longer forced to switch to email.

Another major feedback was about the performance of the fragmentation. The processing of the video files took too long for the testers to use this tool in a journalistic workflow. This was addressed via several improvements of the algorithm. The improved processing speed was appreciated by the testers. Still there are some performance issues at the moment if a lot of videos are processed at the same time.

The third major feedback concerned the number, quality and coverage of the extracted thumbnails. Based on this feedback the algorithm for thumbnail extraction has been improved.

### 4.3.3 Overview of the feedback

**Table 10: Number of received feedback comments from the different test cycles for the Web Application for Video Fragmentation & Reverse Image Search**

Test cycle	Feedback comments
Test cycle 1	18
Test cycle 2	13
Test cycle 3	21

#### 4.3.4 Feedback on documentation

- **Explanation and disclaimer too long**

**Feedback:** The explanation on the webpage and also the disclaimer are too long and unstructured. This keeps the user from using the tool. Make it more legible.

**Resolution:** Feedback will be taken into consideration for the planned update of the UI.

- **Improve explanation for reverse image search**

**Feedback:** Change the explanation for the reverse image search which currently is “By left clicking ...” “By right clicking ...”. Start with the benefit for the user, for example “To search for similar images ...”

**Resolution:** Feedback will be taken into consideration for the planned update of the UI.

#### 4.3.5 Feedback on performance

- **Performance of processing**

**Feedback:** Calculating the thumbnails takes a very long time.

**Resolution:** Changes in the algorithm enable faster sub-shot segmentation analysis (25 times faster compared to the previous one)

**Feedback on resolution:** Response time is better now.

- **Performance of processing when other videos are processed in parallel**

**Feedback:** Processing time is pretty long when other videos are processed in parallel. It seems like the system is not capable of handling many users at the same time.

**Resolution:** planned installation of the web application in a dedicated server

#### 4.3.6 Feedback on usability

- **General workflow**

**Feedback:** The process seems very complicated. First having to download a video (i.e. in a breaking news situation), then uploading it, and then waiting for it to be processed. Reading the e-mail with the result and clicking on the link in the mail. This needs to be streamlined. Also the testers were uncomfortable having to provide an email address with very limited terms & conditions.

**Resolution:** The process has been improved substantially. Now a URL to a video can be provided and the link to the result is displayed in the webpage without the need of using an email address.

- **Show estimated processing time**

**Feedback:** Providing information on how much time processing requires and why it lasts so long would be positive.

**Resolution:** open

- **Progress bars for processing a video**

**Feedback:** The display of progress bars for the video processing was appreciated.

**Resolution:** not required
- **Progress bars appear with some delay**

**Feedback:** The progress bars which indicate the processing of the video appear only with some delay.

**Resolution:** fixed
- **Redirecting to result page**

**Feedback:** After the analysis is finished the user has to click on a link to get to the result. Suggestion: Redirect to the result page automatically.

**Resolution:** Feedback will be taken into consideration for the planned update of the UI.
- **Button for reverse image search**

**Feedback:** Integrate the reverse image search directly (with a button next to each thumbnail) instead of by giving instructions on what to do ("right click this" etc.)

**Resolution:** Feedback will be taken into consideration for the planned update of the UI.
- **Avoid technical messages**

**Feedback:** Avoid too many technical messages displayed to the end user.

**Resolution:** Feedback will be taken into consideration for the planned update of the UI.
- **Open a key frame in a separate window**

**Feedback:** Clicking on a thumbnail opens the picture in a larger size in a pop-up window. This behaviour has irritated various testers.

**Resolution:** open

#### 4.3.7 Main feedback

- **Support more video formats**

**Feedback:** Support more video formats. For example, webm format video is not supported

**Resolution:** The service now supports the following video formats: mp4, webm, avi, mov, wmv, ogv, mpg, flv, and mkv.

**Feedback on resolution:** Tests with video in mp4, webm, flv, mov formats did well. A couple of tests with mkv and m4v videos went wrong with the message "Invalid file application/octet-stream" and "Invalid file video/mp4" respectively.

**Resolution:** open

- **Support should not be limited to local video files**

**Feedback:** The service only works with downloaded files. But UGV is mostly found online and has a URL to point the application to.

**Resolution:** It is now possible to also input a URL to a video file.

- **Works with all common web video sources**

**Feedback:** Video fragmentation has been tested successfully with YouTube, Facebook; Daily Motion and Twitter videos.

**Resolution:** not required

- **Reverse image search is helpful**

**Feedback:** The reverse image search is great. The reverse image search functionality certainly does help.

**Resolution:** not required

- **Image magnifier is helpful**

**Feedback:** The functionality to see a larger image of the key frame is very useful.

**Resolution:** not required

- **Video quality**

**Feedback:** The ratio (width to height) of videos is not displayed correctly.

**Resolution:** The original aspect ratio is now maintained.

**Feedback on resolution:** The fragmentation service is not always preserving the video size. It is better than it used to be at the beginning, but with video taken with a smartphone for instance the initial size of the video is not kept in the player, thumbnail and the key frames to download. With a mov 872x480 video the key frames were calculated in 720x480 pixels. This could be a problem to see details and maybe also in the reverse search.

**Resolution:** open

- **Feedback on found keyframes**

**Feedback:** In several examples interesting keyframes have been missed by the service.

**Resolution:** Feedback will be taken into consideration for the further development of the video fragmentation and keyframe extraction algorithm.

- **Show wide angle shots**

**Feedback:** Keeping wide shots key frames is important from a user point of view. Video manipulation often relies on closing up the shot, eliminating details that would help to debunk the fake. Wide shots also help to identify the location through buildings, monuments, shops, banners...

**Resolution:** Feedback will be taken into consideration for the further development of the video fragmentation and keyframe extraction algorithm.

- **One and the same sequence is marked as two segments**

**Feedback:** The system seems to detect all new sequences, but it quite often marks one and the same sequence as two segments.

**Resolution:** The results have been improved significantly. Now there are fewer shots that show the same segment.

- **Increased number of extracted key frames**

**Feedback:** The number of extracted key frames has been increased which is a very good feature.

**Resolution:** not required

- **Avoid blurred key frames**

**Feedback:** Avoid blurred key frames as they are not useful. Keep only sharp key frames.

**Resolution:** Planned but not yet implemented.

- **Repeating key frames in some videos**

**Feedback:** One and the same thumbnail is shown many times for a particular video.

**Resolution:** open

## 4.4 Tool for Social Media Retrieval and Topic Detection

### 4.4.1 Description of the service

The Tool for Social Media Retrieval and Topic Detection collects tweets from various news accounts on Twitter, collects video documents from various social networks, and collects metadata for each social media document. Through the InVID Multimodal Analytics Dashboard, the social media can be explored and the metadata can be used both as a means to filter/query the documents as well as to provide additional context for the user when a document is selected.

The story detection clusters the collected documents and identifies them as belonging to distinct news stories. Also, filtering by news topic is provided as functionality. Label stories are accurately based on keywords. The most relevant documents are provided for each story.

### 4.4.2 Major outcomes of the test cycles

Regarding the keywords a major outcome of the testing was the suggestion to use IPTC media topic keywords (<http://cv.iptc.org/newscodes/mediatopic/>) instead of the existing Wikipedia categories. This was addressed by aligning the topics list with the IPTC standard.

A major outcome of the test cycles for the story detection was that the stories are useful and meaningful. But there were some test results with very similar story titles. This has been addressed by improving the algorithm for the story disambiguation.

For the associations the tests provided singular cases where the associations were not very meaningful to help with the improvement of the algorithm.

### 4.4.3 Overview of the feedback

**Table 11: Number of received feedback comments from the different test cycles for the Tool for Social Media Retrieval and Topic Detection**

Test cycle	Feedback comments
Test cycle 1	10
Test cycle 2	11
Test cycle 3	9

### 4.4.4 Main feedback

- **Usefulness and meaningfulness of the stories**

**Feedback:** Stories look useful and meaningful.

**Resolution:** not required

- **Quality of keywords and associations**

**Feedback:** Keywords are related to the topic.

**Resolution:** not required

- **Topics categories Wikipedia vs. IPTC**

**Feedback:** Topics are mapped to Wikipedia categories and are quite close to the news taxonomy issued by IPTC.

<http://cv.iptc.org/newscodes/subjectcode/> was a first attempt by IPTC to implement a news taxonomy on three levels (subject, subject matter, subject detail) and this was later replaced by the media topic on five levels: <http://cv.iptc.org/newscodes/mediatopic/>.

The proposed topics in the GUI are a bit mixed and it appears that the law and crime category is too large, gathering items that are in fact related to politics or diplomacy.

It is therefore recommended to merge politics, elections and international relations and to add words detecting diplomacy (diplomatic, diplomats), embassy, consulate ...

**Resolution:** The topics list was corrected (in alignment with the IPTC NewsCodes) and re-definitions were made.

- **Lack of sources in languages other than English**

**Feedback:** Content in languages other than English is very limited.

**Resolution:** open

- **Add news television channels**

**Feedback:** The scope of the available data should be expanded to include news television channels.

**Resolution:** open

- **Keyword video is missing**

**Feedback:** In the list of keywords found, the keyword "video" is missing

**Resolution:** The keyword "video" is now shown.

- **Source of video is not displayed**

**Feedback:** The provenance of a video (Facebook live, YouTube, Daily Motion, Twitter, etc.) would be a useful concept.

**Resolution:** Image thumbnails for YouTube, Vimeo and Dailymotion videos have been added in story view.

- **Limitation of 10 stories and 10 documents within the stories**

**Feedback:** The tool shows up to 10 main clusters of documents, obviously ranked by the size of the clusters, while the graphical representation on top shows a timeline. The GUI only allows to see the first 10 documents (on which criteria remains unclear). So in the stories view there is no way to really explore all stories. Stories are also very limited (a



couple of text lines, most of the time without knowing which is the broadcast channel if no logo is present in the video). The top ten documents generally make sense but it is impossible to know whether there is a more pertinent document in the cluster.

**Resolution:** open

#### 4.4.5 Feedback for improving the performance of the algorithm

- **Different keywords for “Trump”**

**Feedback:** A search for “Trump” in a timeframe of a couple of weeks gives back results like “donald trump” (in lower case), “President Donald”, “president”, “Donald”, “trump”. All these results should semantically be merged under the same and unique named entity “president Donald Trump”.

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **Associations for “Manchester attack”**

**Feedback:** A search for “Manchester attack” gives a few uncommon associations such as “least”, “Australian”, “court”, “key” or “level”.

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **Associations for “Trump”**

**Feedback:** A search for “Manchester’s terror attack” was carried out. Extractions of associations showed an overweight of “Donald Trump” and some other associations (such as “concert in Manchester”).

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **Uncommon story cluster titles**

**Feedback:** The titles of the article clusters were sometimes uncommon. Search term: “Wilders”. Article cluster titles:

ERDOGAN + DUTCH + GEERT

CHALLENGE + MARK + PARTY

HAPPY + BREXIT

PRESIDENT DONALD + US PRESIDENT + DONALD TRUMP

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **Difference in stories unclear**

**Feedback:** Sometimes the story titles are very similar.

Example: A search for “Manchester” on 26/5/2017 brings up the following stories:

CONCERT + ATTACK + BOMBING

ATTACK + TERROR + SUICIDE BOMBER

ATTACK + BOMBING + TERROR

ATTACK + TERROR + VIDEO

ATTACK + CONCERT + BOMBING

ATTACK + TERROR + SUICIDE BOMBER (same cluster naming as above)

THE ATTACK + TERROR + CONCERT

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **English keywords for French content language**

**Feedback:** In French, on the keyword “Fillon” we get back some videos but the associated keywords are mostly in English, while “Le Pen” is detected as “pen”. A keyword examination also suggests a tokenisation of the French word “mis en examen” (charged with suspicions of ...).

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

## 4.5 Context Aggregation & Analysis Service

### 4.5.1 Description of the service

The Context Aggregation & Analysis Service (API and UI) is aimed at facilitating the verification of the content derived exclusively from the YouTube and Facebook platforms.

### 4.5.2 Major outcomes of the test cycles

This service provides a user interface and an API. Both were tested in all three test cycles and the major outcomes of both are listed below.

The user interface received a lot of positive feedback from various testers. This includes positive feedback about the functionality and also that the results were appropriate and accurate enough for journalistic needs. This tool helps the testers in solving video verification problems.

The ongoing development of the tool was appreciated by the testers. For example, the inclusion of Facebook videos besides YouTube videos is a key asset for the testers. The implementation of the weather tool also received a lot of positive feedback. The display of tweets for the video was seen as very valuable but the label “reliable” has to be explained in this context.

Regarding the API, the feedback valued the change of the processing call from synchronous to asynchronous because this makes integration simpler for applications.

Further feedback on the API was that error messages were missing for different situations. This has led to an improvement of the error messages.

### 4.5.3 Overview of the feedback

**Table 12: Number of received feedback comments from the different test cycles for the Context Aggregation & Analysis Service**

Test cycle	Feedback comments from testing the user interface	Feedback comments from testing the API
Test cycle 1	14	14
Test cycle 2	62	17
Test cycle 3	39	15

#### 4.5.4 Feedback on documentation

- **Explanation of the different functionalities**

**Feedback:** Functionalities and results could do with better explanations.

**Resolution:** Improved explanations have been provided.

- **Improve the general explanation of the application**

**Feedback:** The general explanation of the application should be easier to understand for new users. “Context Aggregation and Analysis” and “The demo focuses more on the video context rather than its content.” is not understandable for journalists.

**Resolution:** open

#### 4.5.5 Feedback on performance

**Feedback:** Quick enough! All good. Good response time. Pretty quick.

**Resolution:** not required

#### 4.5.6 Feedback on usability

- **General feedback on usability**

**Feedback:** The usability of this application got positive feedback from various testers. The application was easy to use for them.

**Resolution:** not required

- **Reverse image search appreciated**

**Feedback:** The possibility to perform a reverse image search directly was appreciated by the tester.

**Resolution:** not required

- **Expandable buttons**

**Feedback:** Not all features are visible at a glance. The user has to scroll far down to discover all features. Suggestion: expandable buttons to avoid displaying all the data in one long page.

**Resolution:** open

#### 4.5.7 Main feedback

- **Display of video data - duration**

**Feedback:** The video duration format is unclear (PT2M10S).

**Resolution:** The duration is now displayed in a common format (“HH:MM:SS”).

- **Display of video data - channel name**

**Feedback:** Section Channel: The channel name is missing.

**Resolution:** open

- **Display of video upload time**

**Feedback:** Video upload time is shown in Pacific Time. As the target users for the application are in the first phase in Europe it is suggested to switch to a more common time zone for these users.

**Resolution:** The upload time is now shown in GMT.

- **Display of the attribute video licensed content**

**Feedback:** Regarding the value for “video licensed content” - how does the tool determine the tag “false”? Video licensed content should be about Creative Commons with or without attribution, commercial rights reserved.

**Resolution:** A description of the field has been provided in the internal document describing the changes in the new version. However this is not yet available to the users in the interface.

- **Attribute verified for Facebook users**

**Feedback:** For videos of Facebook users an attribute “verified” is displayed. What does this mean?

**Resolution:** open

- **Found locations**

**Feedback** The system seems to tokenise all possible location names and separate them as a unique word and then reorder them in a weird order (States, United i.e.). It would be very useful to skip the comma and to reorder the location as found in the text.

**Resolution:** open

- **Location detection does not work for other languages than English**

**Feedback:** It seems that in languages other than English, locations are mostly not detected.

**Resolution:** open

- **How are comments identified that are useful for verification?**

**Feedback:** It is unclear how the tool identifies comments that are useful for verification.

**Resolution:** An explanation has been added on how the tool identifies the useful comments, highlighting of the used keywords in the comments.

- **Display the author of a comment**

**Feedback:** It would be interesting to know who has made a comment.

**Resolution:** Author and publication date of the comment are now displayed. A link to the comment author's YouTube channel is now also provided.

- **Handling a large number of comments**

**Feedback:** Displaying the comments is good as very often these include great information, but how can they be searched or filtered? It is necessary to include background info on these sources, otherwise it is not possible to evaluate if a comment is helpful or not.

**Resolution:** Partially open, used information now provided for each comment.

- **Addressing multilingualism**

**Feedback:** An extension of the word list for verification with other languages is necessary.

**Resolution:** open

- **Weather context**

**Feedback:** The addition of the weather context was appreciated by various testers.

**Resolution:** not required

- **Weather context: Adding day/night indication**

**Feedback:** It was suggested to add a day/night indication.

**Resolution:** implemented

- **Weather context: Showing time zones**

**Feedback:** An indication about the time zone of the time input for the weather context is missing. Is it GMT like the video upload time, or the local time of the chosen location?

**Resolution:** open

- **Unclear why a twitter source is reliable**

**Feedback:** The interface shows "reliable" and "unreliable" tweets. It is necessary to explain better why a source is reliable, what parameters are being checked?

**Resolution:** open

- **Clicking the distribution of reliable and unreliable tweets**

**Feedback:** The graphics with the distribution of reliable and unreliable tweets would be more helpful when it were interactive. When the green area for reliable tweets is clicked the system should only show the reliable tweets.

**Resolution:** implemented

- **Connections to port 8090**

**Feedback:** For connections, the tool uses port 8090. This will be a problem for many journalists. Journalists often only have permissions for ports 80 and 443.

**Resolution:** The service is now available at port 80.

- **Video not found**

**Feedback:** Sometimes a video cannot be processed and the message “the video has not been found” is displayed. Usually, clicking the Verify button a second time resolves the issue.

**Resolution:** open

- **Reverse image search does not work for Facebook videos**

**Feedback:** Clicking on “Reverse image search” for a thumbnail of a Facebook video results in an error in Google: “picture not public available”.

**Resolution:** open

- **Display not right-aligned for Arabic videos**

**Feedback:** On Arabic videos, display is not right-aligned. This might seem to be a detail, but as most videos coming from Syria, Iraq etc. are UGV and potential external testers could also include Al Jazeera / Al Arabiya etc. it would be great to have it.

**Resolution:** open

#### 4.5.8 Feedback from the API testing

- **Tested functions**

- verify\_video
- get\_ytverification
- get\_twverification
- get\_fbverification
- weather

with short and long videos, also in various error situations.

- **General feedback for API testing**

**Feedback:** The second version of the API shows a real service improvement. The operation that processes a video is now asynchronous and returns a status and links to get results of the analysis.

With the asynchronous feature, the service is very straightforward to use, and it makes integration simpler.

The service is straightforward to use and performance is very good.

**Resolution:** not required

- **Feedback on API documentation**

**Feedback:** Swagger documentation is good enough. All documentation for the test cycles (changes, etc.) should be set in a global API documentation.

**Resolution:** open

- **Feedback on API performance**

**Feedback:** The first processing of a video can take time, depending of the number of comments.

**Resolution:** The performance for videos that have a lot of comments has been improved.

- **Feedback on performance**

**Feedback from test cycle 2:**

Verify-video < 100ms

1500 ms <get\_ytverification <2000ms

1500 ms <get\_twverification <2000ms

1000 ms <get\_twverification <1500ms

**Feedback from test cycle 3:**

There is an improvement in response time.

Verify-video < 100ms

900 ms <get\_ytverification <1200ms

1500 ms <get\_twverification <1800ms

300 ms <get\_fbverification <400ms

**Resolution:** not required

- **Feedback on reliability**

**Feedback:** The service is very stable.

**Resolution:** not required

- **No improvement in response time if the same video is searched multiple times.**

**Feedback:** If the same video is searched multiple times the response time stays the same.

**Resolution:** If the same video ID has been sent in the past, the results are immediately returned. If more than 24 hours have passed since the previous submission, the service checks for new information (video comments, tweets). If new information exists, it is appended to the existing result.



- **An error message is displayed if the same ID is searched simultaneously**  
**Feedback:** If the same ID is searched twice at the same time, this results in an error.  
**Resolution:** fixed
- **Missing video ID does not return an error message**  
**Feedback:** Sending a request with an empty ID does not return an error.  
**Resolution:** fixed
- **Missing parameter ID results in an unexpected error message**  
**Feedback:** A missing parameter ID results in unexpected error message.  
**Resolution:** fixed
- **Bad verification returned URL for Facebook video processing**  
**Feedback:** The result for Facebook processing gives URL with get\_ytverification instead of get\_fbverification.  
**Resolution:** open
- **Processing full Facebook video URL never ends**  
**Feedback:** When sending the full Facebook URL instead of the ID, the processing result stays at this value processing.  
**Resolution:** open
- **Parameter metadata not clear**  
**Feedback:** It is not clear how the parameter metadata can be used and for what proposes.  
**Resolution:** open

## 4.6 Near Duplicate Detection Service

### 4.6.1 Description

The Near Duplicate Detection Service is a service (API) which identifies near-duplicate content. This will inform the end user whether a posted image or video was posted in the past (and is hence not an original).

### 4.6.2 Major outcomes of the test cycles

The Near Duplicate Detection Service was still in development during test cycle 1 and participated in test cycles 2 and 3. A major feedback was about many “time out” responses for the index function. This would make the API hard to integrate. As a response the service was sped up significantly and this has resolved the timeout problems. Another major feedback concerned the functionality of the API calls. This function search also indexes a video if it hasn’t been indexed already. This was unexpected for the testers as there is a separate call that has the functionality for indexing. Based on this feedback the search function now no longer indexes the videos. Finally, testers said that error messages were not uniform. This has led to an improvement of the error messages.

### 4.6.3 Overview of the feedback

**Table 13: Number of received feedback comments from the different test cycles for the Near Duplicate Detection Service**

Test cycle	Feedback comments
Test cycle 1	--
Test cycle 2	19
Test cycle 3	21

### 4.6.4 Tested functions

- Index
- Search
- Delete

also in various error situations

### 4.6.5 General feedback

**Feedback:** The API is straightforward to test and easy to use.

**Resolution:** not required

#### 4.6.6 Feedback on documentation

**Feedback:** The documentation for this service in the InVID wiki page and the swagger page is not sufficient.

Please add a list of supported video providers (YouTube, Dailymotion, Vimeo, etc.).

**Resolution:** Documentation has been expanded on the Swagger page.

#### 4.6.7 Feedback on reliability

**Feedback:** For indexation there are a lot of “time out” responses, which makes the API hard to integrate.

**Resolution:** The service has been sped up significantly. As a result, timeout features have also been fixed.

**Feedback on the resolution:** All tested videos went through without error.

#### 4.6.8 Feedback on performance

**Feedback:** Performance is good enough.

<5s for indexation

<1s for search

<500ms for deleted

**Resolution:** not required

#### 4.6.9 Main feedback

- **Search should not index videos**

**Feedback:** Non-indexed video seems to be indexed using this operation before being able to search. The search function should not index a video.

**Resolution:** A search for non-indexed videos no longer indexes the videos.

- **Error handling and error codes**

**Feedback:** There are some errors that give the status as a message (“PROCESSING FAILED”); other errors give response code 500.

**Resolution:** Error handling has replaced the returning response code 500 or 200.

- **Deleted document**

**Feedback:** After deleting a document it is still possible to search for it by using the deleted video ID. In addition, deleted videos, indexed again, cannot be deleted.

**Resolution:** Videos are now deleted correctly.

- **URL end with spaces gives different results**

**Feedback:** Spaces at the end of the URL are considered as part of the URL in the process.

**Resolution:** The service now trims the input URL of trailing spaces.

- **Display video source in search result**

**Feedback:** In the search result there is no indication about the source of the video (if it is YouTube, Vimeo, etc.). This is not very handy for integration into a UI that would show the results.

**Resolution:** open

- **Provide a list of supported videos**

**Feedback:** The list of supported video providers (YouTube, Dailymotion, Vimeo, etc.) is missing.

**Resolution:** open

- **Results from closed channels or removed videos**

**Feedback:** Some returned videos are from closed channels which can no longer be accessed or do not exist anymore.

**Resolution:** open

## 4.7 Logo Detection Service

### 4.7.1 Description

The Logo Detection Service (API and user interface) performs logo detection on a video or image. Subsequently, the service returns information concerning all detected logos, aimed at assisting investigators in identifying the origin of the image/video content.

### 4.7.2 Major outcomes of the test cycles

This service provides a user interface and an API. Both were tested in all three test cycles and the major outcomes of both are listed below. A major feedback was that the testers liked the functionality of this tool as it will help them in the verification of videos. But the testers were also aware that only previously indexed logos are recognised. The maintenance of the indexed logos will be a major task to ensure that this tool also recognises logos from new organisations. Another major feedback concerned the workflow in the user interface. The users had to switch between the source video and the source picture. This choice was easily overlooked and resulted in an error if it didn't match with the provided source. As a result from this feedback the service now automatically detects the file type and distinguishes between images and videos. The user doesn't have to choose this anymore. Last but not least, testers indicated that on API level error messages were missing for different situations. This has led to an improvement of the error messages.

### 4.7.3 Overview of the feedback

**Table 14: Number of received feedback comments from the different test cycles for the Logo Detection Service**

Test cycle	Feedback comments from testing the user interface	Feedback comments from testing the API
Test cycle 1	11	23
Test cycle 2	16	36
Test cycle 3	20	19

### 4.7.4 General feedback

**Feedback:** The testers liked the functionality of this tool as it will help them in the verification of videos.

**Resolution:** not required

#### 4.7.5 Feedback on performance

**Feedback:** The performance of new videos is slow, it takes long until progress moves away from 0%.

**Resolution:** The service has been sped up through parallelisation.

#### 4.7.6 Main feedback

- **Uploading from various images results in an error**

**Feedback:** Uploading from various images result in an error.

**Resolution:** Various bugs have been fixed that had caused errors.

- **Possibility to choose between video and images easily overlooked**

**Feedback:** The possibility to choose between video and images is easily overlooked.

**Resolution:** The service now automatically detects the file type and distinguishes between images and videos. There is no need to specify beforehand.

- **Shortened YouTube URL does not work**

**Feedback:** The tool does not work with the shortened YouTube URL.

**Resolution:** Support for shortened YouTube URLs has been added.

**Feedback on resolution:** It is now working well with youtu.be short links available in the share option under any YouTube link. It is also working with the Daily Motion shortener.

- **Wording of the logos list**

**Feedback:** The logos list should be renamed to “Known logos” or something similar.

**Resolution:** open

- **Adding a new logo**

**Feedback:** It is not clear how to add a new logo. Add an explanation: “If you want to add a new logo, contact ....”

**Resolution:** open

- **Connections to port 8080**

**Feedback:** For connection the tool uses port 8080. This will be a problem for many journalists. Journalists are often only allowed for ports 80 and 443.

**Resolution:** The service is now available at port 80.

- **Processing fails for some videos**

**Feedback:** A minor number of videos results in an error or is stuck in processing.

**Resolution:** open

#### 4.7.7 Feedback for the API testing

- **Tested function**

- fromvideourl
- fromimageurl
- fromimagefile
- analysisstatus

also in various error situations

- **General feedback for the API**

**Feedback:** This new version shows a significant service improvement that makes all operations easy to use.

The addition of the function analysisstatus gives a very useful and preformat feature to get information about an image of a video process and final results.

**Resolution:** not required

- **Feedback on Performance**

**Feedback:** Parallel processing seems to work well. It is possible to send multiple processes at the same time. Cache function seems to work well to avoid reprocessing video or image.

**Resolution:** not required

- **Feedback on documentation**

**Feedback:** Documentation of the API is not complete. The REST API services are not described in the documentation. The most information is only stored on Swagger UI.

**Resolution:** Additional documentation has been created.

- **Feedback on reliability**

**Feedback:** Technically the service is reliable. No errors or downtimes during tests were noticed. Performance is very good.

**Resolution:** not required

- **Error handling and error codes**

**Feedback:** There are some error situations where the error message is only "Internal Server Error".

**Resolution:** More informative error messages have been added.

- **Daily motion support to /fromvideourl**

**Feedback:** Sending a Dailymotion link made the service fail during processing.

**Resolution:** Dailymotion support has been added.

#### 4.7.8 Feedback for improving the performance of the algorithm

- **Detected logos**

**Feedback:** Tested successfully (both YouTube and Daily Motion) on Aljazeera, MSNBC, Sky News, Bloomberg, The Rebel Media, BFMTV, ABC, Associated Press, France 24.

False positives on Business Plus, SABC, ABC News 24, BFM Business, Channel News Asia.

False detections: CNBC Awaaz detected instead of NBC News, NBC News logo is wrongly mixed with MSNBC.

No detection for Todo Noticias (3 videos), The Rebel Media (1 video), Associated Press (1 video), CBS News (1 video), 24 Hours (2 videos) Pakistani TV (3 videos).

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.

- **Logo not detected**

**Feedback:** In some cases the logo was not detected with the API calls fromimageurl and fromimagefile. This concerns logos from ABC News, Al Jazeera, Sky News and CNN.

**Resolution:** Feedback will be taken into consideration in the further development of the algorithm.



## 4.8 InVID Verification Plugin

### 4.8.1 Description

The InVID Verification Plugin wraps up some basic tools developed to help journalists in the verification/fact-checking process. It provides a simple way to use those tools directly in the browser. It enables to get feedback from end-users on usability, needs, etc.

### 4.8.2 Major outcomes of the test cycles

The InVID Verification Plugin left the development stage in spring 2017 and therefore only took part in test cycle 3. The application was appreciated by various testers. They liked that it was easy to start and that instantly all features / possibilities were available in one place. The different tools were useful to them and the results matched their expectations.

A major feedback from the testers was that this tool only works in the browser Chrome and they also want to use it in other browsers. As a result a plugin for Firefox is currently under development and a plugin for Safari has been planned. Another feedback regarded the Twitter search of the plugin. If the date range for the Twitter search was left empty, the user did not get a result but neither any hint that he had to select a date range. This was addressed by implementing a default timespan that is used if input fields for the date range are empty.

### 4.8.3 Overview of the feedback

**Table 15: Number of received feedback comments from the different test cycles for the InVID Verification Plugin**

Test cycle	Feedback comments
Test cycle 1	--
Test cycle 2	--
Test cycle 3	25

### 4.8.4 Main feedback

- **Works in Chrome only**

**Feedback:** Works in Chrome only (thereby excluding all other browsers / non-Chrome users).

**Resolution:** Plugins for Firefox and Safari are planned already.

- **Date range mandatory for Twitter search but no feedback if it isn't set**

**Feedback:** If the input of the date range is missing in a Twitter search then the search doesn't start. But the user doesn't get any feedback why the search didn't start.

**Resolution:** The date range is now no longer mandatory. A default timespan is used if the input fields for the date range are empty.

- **Icon “@” not understandable**

**Feedback:** The icon “@” for “about” is not understandable.

**Resolution:** The icon has been changed to “? about”.

- **Connections to port 8080**

**Feedback:** For connection the plugin also uses the CERTH API with port 8090. This will be a problem for many journalists. Often journalists are only allowed for ports 80 and 443.

**Resolution:** As the used services have been switched to port 80, the plugin also uses the services on this port.

- **Image magnifier**

**Feedback:** The Image magnifier is a nice tool and is suggested to be used by all tools of the InVID project which show images.

**Resolution:** not required

- **Automatic reverse image search opens many new browser tabs**

**Feedback:** After submitting a video link in the “Thumbnails” tab, many new browser tabs open (with reverse image search). This is surprising. Add a hint for the user that this function opens a browser tab for each thumbnail or let the user choose which thumbnails to use for reverse image search.

**Resolution:** open

- **Broken image for InVID Logo**

**Feedback:** In the Twitter timeline, the InVID logo image is not loaded – a broken image symbol is displayed.

**Resolution:** fixed

- **YouTube comments: missing username and link to YouTube profile**

**Feedback:** Tab Analysis, verification comments: add username of the author and a link to his YouTube profile.

**Resolution:** fixed

## 4.9 InVID Rights Management Service

### 4.9.1 Description

The InVID Rights Management Service deals with the copyright aspects related to the reuse of UGV. It helps to discover the owner of an interesting piece of UGC, to contact the owner, set a copyright negotiation framework and establish the required rights agreements to reuse the asset.

### 4.9.2 Major outcomes of the test cycles

The API for the InVID Rights Management Service was tested in all three test cycles whereas the user interface of the InVID Rights Management Service was tested only in test cycle 3. This was for development reasons and to provide the testers a stable user interface.

The results from the API tests were very good. The API is very well documented, reliable and no bugs were found for this API in any of the three test cycles.

For the user interface the testers confirmed that it contains very useful information for a journalist. A major feedback for the user interface concerns not the user interface itself but how to explain the benefits of registering a video to a content owner. This will be necessary to motivate the content owners to use the registration part of this tool.

### 4.9.3 Overview of the feedback

**Table 16: Number of received feedback comments from the different test cycles for the InVID Rights Management Service**

Test cycle	Feedback comments from testing the user interface	Feedback comments from testing the API
Test cycle 1	--	6
Test cycle 2	--	21
Test cycle 3	19	6

### 4.9.4 General feedback

**Feedback:** Contains a lot of useful information.

**Resolution:** not required

#### 4.9.5 Feedback on usability

- **Accept button confusing**

**Feedback:** The display of the invite message is a bit confusing, partly because of the Accept button.

**Resolution:** open

- **Display text for invite message**

**Feedback:** The sentence “The InVID Invite has been created” should be removed and “send” should be replaced by “Copy and paste” as they are misleading.

**Resolution:** open

- **Summary for the license not easily recognisable**

**Feedback:** A summary for the license is available but not easily recognisable.

**Resolution:** The display of the license information has been changed. It has been implemented by using a table that distinguishes what is allowed and what not yet.

#### 4.9.6 Main feedback

- **What is the benefit for the user to register?**

**Feedback:** What is the benefit for the user to register? Possibility to negotiate the conditions, only one request per organisation, possibility to automate the process. -> explain the benefits more to the user.

**Resolution:** Work is in progress with the aim of including this information in the invitation message while keeping it not too verbose...

- **Wording “Register”**

**Feedback:** Suggestion: Wording “Register” -> “Check YouTube videos”

**Resolution:** Implemented

- **Request button disappears**

**Feedback:** The reuse request button disappears after clicking “Accept” in the invite dialog. Also there wasn’t any acknowledge message like “your request has been sent” or something like that.

**Resolution:** open

- **Arabic text is not right-aligned**

**Feedback:** Arabic text (from description, title ...) is not right-aligned

**Resolution:** open

#### 4.9.7 Feedback for the API

- **Tested functions**

- Login
- Register YouTube video
- Retrieve Content
- Reuse Request
- InVidInvite

- **General feedback**

**Feedback:** This is a very good API. All is well documented. Operations have excellent response time.

**Resolution:** not required

- **Feedback on performance and reliability**

**Feedback:** API seems very reliable. Response times are very good and it accepts any type of data without crash.

**Resolution:** not required

- **Feedback on API documentation**

**Feedback:** Documentation is very good. Everything is well described from the requests models to the responses models. Every scenario of the API is described. The documentation makes integration to third parties straightforward.

**Resolution:** not required

## 4.10 InVID Verification Application

### 4.10.1 Description

The InVID Verification Application enables journalists to find and validate newsworthy UGVs by using the different InVID platform services.

### 4.10.2 Major outcomes of the test cycles

The InVID Verification Application was tested in test cycles 1 and 3. The main development of the InVID Verification Application was the integration of the different InVID services into one user interface. This covers the communication to the different APIs in the backend also as the presentation of the results of the APIs in the user interface.

The major feedback from the test cycles was that the workflow and the usability of the application need to be improved to support better real-life use cases. This is not unexpected as until now the focus of the main development was to provide the different functionality of the InVID services and fix major bugs. The feedback from the test results is taken into consideration and as a result of this feedback a redesign of the InVID Verification Application is planned. This starts with a workshop between testers and developers in July 2017.

### 4.10.3 Overview of the feedback

**Table 17: Number of received feedback comments from the different test cycles for the InVID Verification Application**

Test cycle	Feedback comments
Test cycle 1	27
Test cycle 2	--
Test cycle 3	35

### 4.10.4 Feedback on usability

**Feedback:** The application needs to be improved to support better real life-use cases.

**Resolution:** A redesign of the InVID Verification Application is planned.

### 4.10.5 Main feedback

- **Threshold to avoid displaying many unrelated videos.**

**Feedback:** In the display of the near-duplicate search a threshold should apply to avoid displaying many unrelated videos.

**Resolution:** implemented

- **Use smaller player size**

**Feedback:** A smaller player should be displayed (keeping a larger size available if needed) to avoid pixellisation.

**Resolution:** implemented
- **Make links clickable**

**Feedback:** It should be possible to click links.

**Resolution:** implemented
- **Missing information from social media**

**Feedback:** Information provided by the APIs is missing in the GUI, such as video view count, like count, dislike count, channel view count, video comment count.

**Resolution:** open
- **Enlarge thumbnails**

**Feedback:** For thumbnails, these are currently too small. It's important to be able to see details and decide which thumbnail is most suitable for a reverse image search.

**Resolution:** open
- **Near Duplicate Search shows only "Process queued"**

**Feedback:** The Near Duplicate Search only shows "Process queued".

**Resolution:** open
- **Compare functionality very useful**

**Feedback:** Being able to compare both videos side by side is a very good feature in the near duplicate tab.

**Resolution:** not required
- **Permanent link to video in verification app**

**Feedback:** After submitting a video for verification there is no possibility to share a link to the video in the verification app.

**Resolution:** open
- **Comments: missing names and links**

**Feedback:** Missing names and links of verifying comments. The user needs to know who has made a comment saying that the video is not accurate. Is he trustable, an expert, an inhabitant of the same area, etc.?

**Resolution:** open

- **Compare button seems not to work**

**Feedback:** Clicking on the compare button in the near duplicate folder seems not to work if the detailed view of a video is closed.

**Resolution:** open

- **Provide links to the found near duplicate videos**

**Feedback:** To get access to the near duplicate videos URLs they have to be opened in YouTube. It would be better to provide links to the found near duplicate videos.

**Resolution:** open

- **No hint that Facebook videos are not supported**

**Feedback:** With regard to Facebook videos, there is no result, no error message, no warning explaining to the user that those URLs are not supported yet.

**Resolution:** open

- **Provide the possibility to take notes for a video**

**Feedback:** Feature suggestion: Notes can be used for example to indicate whether it is graphic content or not, etc. Also this can be used to indicate when e.g. a graphic item appears to forewarn a potential other user.

**Resolution:** open

- **Indication bar not saved**

**Feedback:** The red to green bar is not functional. It is not saved with the video analysis.

**Resolution:** open

- **Sort near duplicate also by time**

**Feedback:** It would be nice to be able to sort the near duplicate search videos by similar score and also by upload date on a timeline for instance.

**Resolution:** open



## 4.11 InVID Multimodal Analytics Dashboard

### 4.11.1 Description

The InVID Multimodal Analytics Dashboard is a tool for media analytics and monitoring. In the InVID project the existing dashboard of Weblyzard is enhanced with new functionality such as displaying videos, the ability to track evolving stories and more.

### 4.11.2 Major outcomes of the test cycles

The InVID Multimodal Analytics Dashboard was fully tested in test cycle 1 and 2. In test cycle 3 the InVID Multimodal Analytics Dashboard took part in the walkthrough testing between testers and developers and also partially in the testing of the visualisation for the Tool for Social Media Retrieval and Topic Detection.

A major feedback was for the presentation of the stories from the Tool for Social Media Retrieval and Topic Detection. This concerns both the visualisation and also the textual representation of the stories. The feedback for this has been taken into account and has led to a substantial improvement in both the visualisation and the textual representation of the story view in every test cycle. Another major feedback from the tests was that the InVID Multimodal Analytics Dashboard is a powerful but complex tool. Proper training for this tool is necessary to successfully use the existing possibilities and understand all results.

### 4.11.3 Overview of the feedback

**Table 18: Number of received feedback comments from the different test cycles for the InVID Multimodal Analytics Dashboard**

Test cycle	Feedback comments
Test cycle 1	42
Test cycle 2	43
Test cycle 3	10

### 4.11.4 General feedback

**Feedback:** The testers liked the overall attractive visual appearance of the GUI.

**Resolution:** not required

**Feedback:** The results seemed fairly accurate and relevant to the testers.

**Resolution:** not required

**Feedback:** The button to export the visualisations was appreciated.

**Resolution:** not required

#### 4.11.5 Feedback on documentation

**Feedback:** Documentation covers basic features but does not seem up-to-date with video, story graph, etc.

**Resolution:** open

#### 4.11.6 Feedback on performance

**Feedback:** Response time is great, no waiting.

**Resolution:** not required

#### 4.11.7 Feedback on usability

**Feedback:** There seems to be lots of information / possibilities, but a shortage of user guidance / orientation.

**Resolution:** open

#### 4.11.8 Main feedback

- **Video source selection should be selected by default**

**Feedback:** Video source selection should be selected by default (InVID is about video verification) and the Dashboard is the tool to discover videos.

**Resolution:** implemented.

- **Story visualisation**

**Feedback:** The labels seem ok but they are difficult to read. Too much concentration. It looks like that each node of the graph is a different cluster of tweets but they have the same size so it's difficult to find out especially as it does not seem possible to explore (navigate) within the graph.

**Resolution:** The story visualisation has been redesigned.

- **Visualisation of story view complex**

**Feedback:** Visualisation of the story view is complex.

**Resolution:** open

- **Display of video**

**Feedback:** Videos only show up in the GUI when clicking on the video play button at the right of each YouTube post.

**Resolution:** Video playback has been made a permanent dashboard feature in the upper right corner. Video playback is triggered by clicking anywhere on a document.

- **Topics: difference between clicking on words and selecting a checkbox not clear**

**Feedback:** Selecting a checkbox for topics or associations does not change the search results. Clicking on the words of topics or associations changes the search results. What are the checkboxes for?

**Resolution:** open

- **Interface language settings do not work**

**Feedback:** The change of the language in the interface settings is expected to change the language of the interface (e.g. 'topics' 'current search', etc.) But this only changes the language of the analysis results.

**Resolution:** open

- **Highlight document from the displayed video**

**Feedback:** After selecting a video in an 'expanded story' it starts playing but the selected document is not marked. It would be good to highlight the document from the displayed video.

**Resolution:** open

- **Problem displaying Daily Motion videos**

**Feedback:** At least several videos from Daily motion are not readable in the GUI video player. It happens regularly on Safari and from time to time on Chrome as well. It seems that there is some policy from Daily Motion to not allow video play outside their platform for some videos (sometimes a link is displayed for watching the video on their platform)

**Resolution:** open

- **Video metadata not displayed**

**Feedback:** For video documents, metadata (name of channel for instance, number of views,) and contextual information are not displayed.

**Resolution:** open

- **Using Story graph switches to document view**

**Feedback:** When selecting Stories, then using the Story graph -> the display mode automatically brings back the Document view.

**Resolution:** open

- **No documents in bursts mode**

**Feedback:** In Story graph switching to Bursts, then choosing a circle and clicking "Explore documents" results in "No documents, please try a different search query"

**Resolution:** open

- **Time zone for date restriction unclear**

**Feedback:** Setting dates is nicely done (calendar), but the used time zone is unclear.

**Resolution:** open

- **Possibility to restrict time**

**Feedback:** At the moment the shown documents can be restricted by date. A possibility to restrict these documents also by hours would be helpful. This is especially important in the case of a breaking news situation to exclude material before an event happened.

**Resolution:** open

## 4.12 InVID Core Platform API

### 4.12.1 Description

The InVID Core Platform API provides access to the services of the core platform such as the possibility to manage documents and perform searches in the stored documents.

### 4.12.2 Major outcomes of the test cycles

The major feedback for this API was about bugs in the management of documents (insert, update and delete). The reported major bugs were fixed. Another major feedback was about missing details in the documentation such as which functions are available or which API version is the one to use. This has led to an improvement of the documentation.

### 4.12.3 Overview of the feedback

**Table 19: Number of received feedback comments from the different test cycles for the InVID Core Platform API**

Test cycle	Feedback comments
Test cycle 1	14
Test cycle 2	24
Test cycle 3	13

### 4.12.4 Tested API functions

- token
- add document
- get document
- update document
- delete document
- search

### 4.12.5 Feedback on documentation

- **Documentation: not clear what functions are available**

**Feedback:** In the API documentation there are no indications which functions are available and which functions are not.

**Resolution:** The available functions have been defined.

- **Swagger GUI cannot be used because of escaping the repository name**

**Feedback:** Swagger GUI: If a repository with a "/" (invid.weblyzard.com/api) is entered in the input field then this repository, especially the "/", will be encoded in the request:

“https://api.weblyzard.com/0.1/documents/invid.weblyzard.com%2Fapi”. This leads to an error.

**Resolution:** open

- **API version not defined**

**Feedback:** The API version for testing is not defined. There are two different API versions available (v0.1 and v0.3).

**Resolution:** The API version for testing is v0.3.

#### 4.12.6 Main feedback

- **GET documents does not work**

**Feedback:** If a document is stored then the API provides a document ID in return. If the document with this ID requested from the API it returns "NOT FOUND".

**Resolution:** fixed

- **Error when updating or deleting a document**

**Feedback:** An error occurs when a document is updated or deleted.

**Resolution:** fixed

- **Duplicate content**

**Feedback:** Sending the same request to the service with the same URI gives a different identifier from the repository. This results in duplicate documents ending up in the repository. Based on the URI, documents should be unique in the repository.

**Resolution:** fixed

- **Provide more details on error**

**Feedback:** Many error types within adding a document, like non-consistent request content, give the same error. More details on the error would be helpful.

**Resolution:** open

- **Document API: optional property responded “bad request”**

**Feedback:** The repository\_id is documented as optional in the content. If this information is provided in the URL and not set as a property in the content then the API responds with {"error": "BAD REQUEST"}.

**Resolution:** open

- **French support for annotations**

**Feedback:** French is not fully supported; there are still too many entities that have been missed.

**Resolution:** considered for future releases

- **Searching for a document does not work**

**Feedback:** Trying to search for a document produces a “Resource not found” error.

**Resolution:** fixed

## 5 Conclusions and outlook for the next test cycles

The test cycles 1 to 3 have provided the developers with a lot of valuable feedback on the different applications and components of the InVID platform. The feedback was given both on a technical level as well as on a user level from a journalistic point of view.

This feedback included:

- Found bugs
- Suggestions for better error handling
- Suggestions on functionality
  - On the function as a whole
  - On a detailed level for the output of a function
- Feedback on documentation
- Feedback on usability
- Suggestions on how to better suit the workflow of a journalist
- Feedback on performance

This feedback was taken into account in the further development of the components and led to substantial improvements of the different applications and components. The next step is to continue with internal testing and also include external users in the testing of the components and applications as they mature further.

In preparation of the next test cycles it is planned to decide which tools are technical, functional and with regard to usability ready for external users. In addition, the provided functions must bring some value to external testers so that they have a motivation to use the tool. Other incentives for participation will also be considered. For the more complex tools of the project, short screen casts and online training sessions are planned in addition to the already existing methods and their documentation.

The major goal of the remaining testing and validation periods and cycles is to continue gathering as much useful feedback as possible for the InVID developer partners. This is vital in order for the tools and services to be developed and further refined becoming as useful and beneficial as possible, meeting clear user (and market) needs. In order to achieve these goals, the base of users / testers will be further expanded, also reaching outside the consortium. The consortium feels it is on the right track, and knows what lies ahead to be tackled in order for InVID to become a success.



## References

- Bourque, Pierre, Fairley, Richard E. (Editors) 2014 SWEBOK® Guide to the Software Engineering Body of Knowledge Version 3.0 IEEE, 2014 ISBN-13: 978-0-7695-5166-1
- ISTQB International Software Testing Qualifications Board <http://www.istqb.org/> Glossary <http://glossary.istqb.org/>
- Jones, Matt, Marsden, Gary (2006) Mobile Interaction Design John Wiley & Sons Ltd., 2006 ISBN-13: 978-0-470-09089-3
- Karner, Cem, Bach, James, Pettichord, Bret (2002) Lessons learned in software testing Willey Computer Publishing, 2002 ISBN 0-471-08112-4
- Stone, Debbie, Jarrett, Caroline, Woodroffe, Mark, Minocha, Shailey (2005) User Interface Design and Evaluations Moran Kaufmann Publishers, 2005 ISBN-13: 978-0-12-088436-0