Visual instance mining of news videos using a graph-based approach

Degree's Final Project Dissertation

by David Almendros Gutiérrez

supervised by Horst Eidenberger and Xavier Giró-i-Nieto

Technische Universität Wien (TU Wien)
Universitat Politècnica de Catalunya (UPC)
2013-2014

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my supervisors, Prof. Horst Eidenberger and Prof. Xavier Giró-i-Nieto, for the support of my study and research. I would also thank for their patience, motivation and encouragement during my thesis. Their advices and comments helped me in all the time of research, development and writing of this thesis.

In addition, I would like to show my gratitude to the UPC Image Processing Department, especially to Albert Gil and Carles Ventura, who helped me solving problems of the different technologies and methods used in the thesis.

Last but not least, my special thanks must go to my family and my girlfriend who have given me the greatest support all this time. Words cannot express how grateful I am to you for all of the sacrifices that you've made on my behalf.

ABSTRACT

The aim of this thesis is to design a tool that performs visual instance search mining for news video summarization. This means to extract the relevant content of the video in order to be able to recognize the storyline of the news.

Initially, a sampling of the video is required to get the frames with a desired rate. Then, different relevant contents are detected from each frame, focusing on faces, text and several objects that the user can select. Next, we use a graph-based clustering method in order to recognize them with a high accuracy and select the most representative ones to show them in the visual summary. Furthermore, a graphical user interface in Wt was developed to create an online demo to test the application.

During the development of the application we have been testing the tool with the CCMA dataset. We prepared a web-based survey based on four results from this dataset to check the opinion of the users. We also validate our visual instance mining results comparing them with the results obtained applying an algorithm developed at Columbia University for video summarization. We have run the algorithm on a dataset of a few videos on two events: 'Boston bombings' and the 'search of the Malaysian airlines flight'. We carried out another web-based survey in which users could compare our approach with this related work. With these surveys we analyze if our tool fulfill the requirements we set up.

We can conclude that our system extract visual instances that show the most relevant content of news videos and can be used to summarize these videos effectively.

CONTENTS

1. INTRODUCTION 1

1.1. Focus of the thesis... 2

1.2. Motivation 3

1.3. Applications 3

1.4. Outline of the thesis 4

2. RELATED WORK 5

2.1. News summarization... 5

2.1.1. News Rover (2013) 5

2.1.2. Name-it 6

2.2. Visual instance mining 8

2.3. Face detection and face recognition 9

2.4. Object detection and object recognition......................................12

2.5. Caption detection 14

3. REQUIREMENTS 16

3.1. Requirements analysis 16

3.2. Content requirements 17

3.3. Structural requirements 18

3.4 . Priorities..19

4. DEVELOPED SOLUTION 21

4.1. Overview...21

4.2. Temporal sampling 22

4.3. Instances detection 24

4.3.1. Face detection..24

4.3.2. Object detection 25

4.3.3. Text detection 30

4.4. Graph-based selection of visual instances 32

4.4.1. Pre-processing of detection boxes 33

4.4.2. Features extraction 34

4.4.3. Similarity graph (Full connectivity) 34

4.4.4. Clustering by Edge filtering 35

4.4.5. Selection of the representative visual instances 36

4.5. Presentation of the results 38

5. EVALUATION.. 43

5.1. User study 1 43

5.1.1. Method 44

5.1.2. Test data 44

5.1.3. Procedure 45

5.1.4 Participants 50

5.1.5. Experimental results 51

5.2. User study 2 55

5.2.1. Method 56

5.2.2. Test data 56

5.2.3. Participants 57

5.2.4. Procedure 57

5.2.5. Experimental results 59

6. CONCLUSIONS 62

7. FUTURE WORK 63

BIBLIOGRAPHY.. 65

APPENDIX I. Working plan 68

APPENDIX II. Test data 70

APPENDIX III. Development... 75

Using the application 76

Appendix IV. Job array technique 77

LIST OF FIGURES

Fig. 1 Goal of the Thesis 1
Fig. 2 Result of our system 2
Fig. 3 News Rover's system architecture overview [3] 6
Fig. 4 Name-it System architecture overview [5] 7
Fig. 5 Examples of mined clusters [6]........ 8
Fig. 6 Three clusters discovered from the Rome dataset [7] 9
Fig. 7 Haar-like wavelets used in OpenCV 10
Fig. 8 LBP code creation example [14]........ 11
Fig. 9 Histograms are concatenated to generate a feature vector for each image [14] 11
Fig. 10 Deformable parts-based example [19] 13
Fig. 11 The flowchart of the algorithm for caption extraction [20] 14
Fig. 12 Position of text in news videos 15
Fig. 13 Partition of the frame [21] 15
Fig. 14 Proposed system architecture 22
Fig. 15 Uniform sampling and selected frames to process related to user's desired rate 23
Fig. 16 Output frames after uniform subsampling 24
Fig. 17 Face detected in a frame 25
Fig. 18 Ambulances training images extracted from ImageNet........ 26
Fig. 19 Detected keypoints from reference objects of training images........ 27
Fig. 20 Matches of the reference ambulance and one in a frame 28
Fig. 21 Test done with ambulances to set the detection threshold 29
Fig. 22 Test done with police car to set the detection threshold 29
Fig. 23 Object detected in a frame 30
Fig. 24 Edge detection of a frame [22] 30
Fig. 25 Components (letters) detected from an image [22] 31
Fig. 26 Text detected in a frame 32
Fig. 27 Overview of the different stages of the clustering block........ 33
Fig. 28 Pre-processing faces to improve the accuracy 33
Fig. 29 Similarity graph with full connectivity 35
Fig. 30 Example of sub-graphs 36
Fig. 31 The representative face of each sub-graphs........ 37
Fig. 32 Initial prototype design of the GUI we want to design 38
Fig. 33 List of some videos users can select 39
Fig. 34 Check buttons with some specific objects 39
Fig. 35 Widget to select the frame rate........ 40
Fig. 36 Widget to select the region to show........ 40
Fig. 37 The START push button of the GUI 40
Fig. 38 Example of a result in the GUI 41
Fig. 39 GUI with the skin applied 41

Fig. 40 GUI while running the application 42
Fig. 41 Result of the tool in the final interface 42
Fig. 42 First part of the evaluation of one summary 48
Fig. 43 Second part of the evaluation of one summary 49
Fig. 44 Last part of the evaluation of one summary 50
Fig. 45 Statistics of genre and age of the participants............................ 51
Fig. 46 Global rating of our system 52
Fig. 47 Results of redundancy in the summaries 52
Fig. 48 Rate of the summaries generated... 55
Fig. 49 Part of the second user study survey 58
Fig. 50 Statistics of genre and age of the participants............................ 59
Fig. 51 Results of the rates of the visual summaries related to the Boston bombing 60
Fig. 52 Results of the rates of the visual summaries related to the Malaysia airlines flight disappearance 61
Fig. 53 Gantt diagram that shows the time planning of the Thesis........... 69
Fig. 56 Visual summary of the news video 1 of the first user study 70
Fig. 57 Visual summary of the news video 2 of the first user study 71
Fig. 58 Visual summary of the news video 3 of the first user study 71
Fig. 59 Visual summary of the news video 3 of the first user study 72
Fig. 60 Visual summary 1 of the news video collection 1 of the second user study 73
Fig. 61 Visual summary 2 of the news video collection 1 of the second user study 73
Fig. 62 Visual summary 1 of the news video collection 2 of the second user study 74
Fig. 63 Visual summary 2 of the news video collection 2 of the second user study 74

LIST OF TABLES

Table 1 Priorities of the requirements 19
Table 2 Test data for the first user study 44
Table 3 Hypothesis we try to verify in the first user study with the questions of the survey 1 46
Table 4 Hypothesis we try to verify in the first user study with the questions of the survey 2 47
Table 5 Keywords of the news story 1 53
Table 6 Keywords of the news story 2 53
Table 7 Keywords of the news story 3 54
Table 8 Keywords of the news story 4 54
Table 9 Test data of the second user study 56

1. INTRODUCTION

In our society, where technological systems are evolving daily, the amount of data we receive through the media is huge. At present, video material and video services are more available than ever. In particular, with the growing popularity of digital news broadcast, the number of collections of news video databases has recently exploded.

With this thesis we propose a solution to the problem of having to watch a whole video so as to find out what is exactly its content. In particular, we aim at extracting visual instance mining [1] of videos that show the most interesting and relevant elements of them. Our approach performs an analysis of all content of the video to show its representative faces, objects and texts.

In this Chapter we talk about the focus of the thesis, explain the motivation and show some applications of the developed tool. We can see in Fig. 1 which is the goal of this Thesis, summarize a video with the most representative content of it.

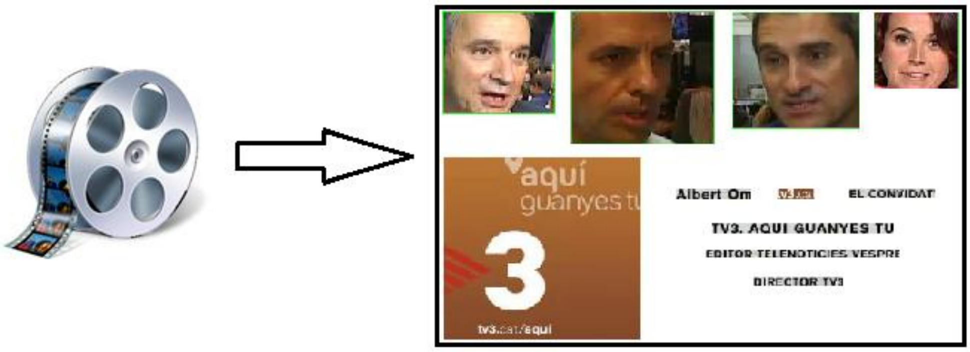


Fig. 1 Goal of the Thesis

1.1. Focus of the thesis

This Thesis is an extension and adaptation of the previous work of Manuel Martos [2]. He developed a content-based summary for films trailers. We studied the future work that could be done and some suggestions Manuel wrote in his report. Firstly we changed the domain of films trailers and we focused on news videos because this is a field where we consider that visual summarization has a greater potential.

In this Thesis we developed an application that extracts visual instances from a video input in order to help the user detect which is the most interesting content. This visual instance mining shows the most representative people, objects and texts.

We decided to extract these content because we think that it could be the best to sum up a video. With the most representative faces we can know who are the main characters of the video and to whom is related the story. Detecting the appearance of some predetermined objects could help to knowing the context of the news. For example, the detection of a fire truck in the video could determine that the story of the news is related to a fire. Finally, extracting the most important texts gives the user extra information just like names of people or the country where the news take place. Fig. 2 shows the most relevant people of a news video and some caption detection that helps the user to know who these characters are.

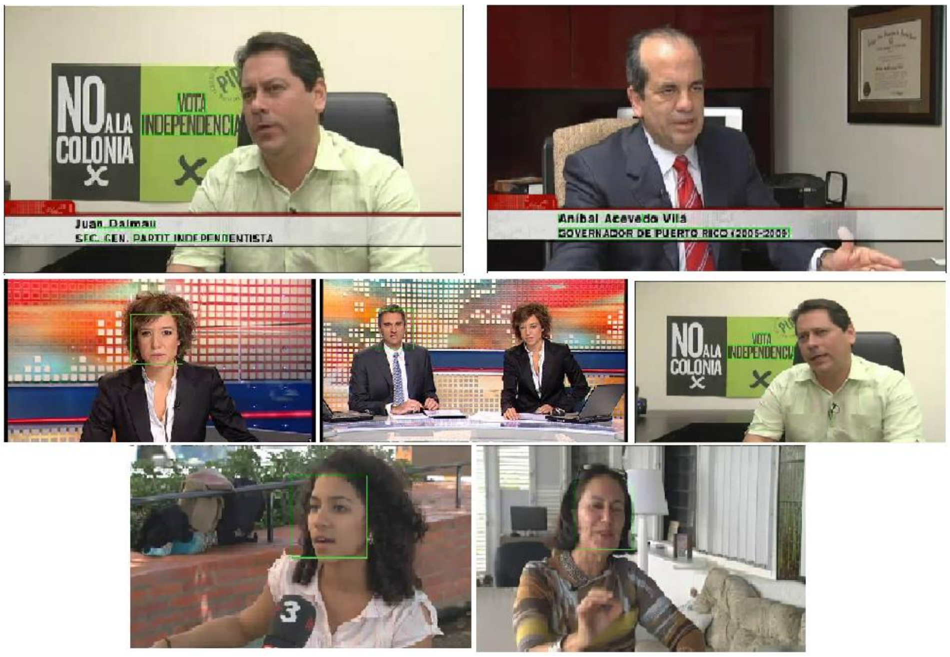

NO A LA COLONIA
VOTA INDEPENDENCIA
Juan Dalmau
SEC. GEN. PARTIT INDEPENDENTISTA
Anibal Acevedo Vila
GOVERNADOR DE PUERTO RICO (2005-2009)

Fig. 2 Result of our system

In the following subchapter we present the motivation of the thesis as well as mentioning different application fields.

1.2. Motivation

At present, people use to read newspapers, watch news bulletins or enter in news websites to be informed of what is happening in their city, their country or around the world. With the growth of the Internet it is relatively easy to find the particular news you are looking for.

In order to know if an specific news is relevant or not, there are various techniques to summarize news. One of them is what we can see at the beginning and at the end of news bulletins, when with really short clips they try to resume the most important news of the day, but sometimes they are not the most relevant for the watcher and other there are not enough information.

Another type of news video summarization are in websites. We can find textual descriptions, which not always give the whole information and forces the user to browse video content in order to determine if this news is relevant or not.

With our visual instance mining, we try to group in a visual way which are the most relevant information of the news video. Compacted in a summary, people will be able to know the most important people of the video and some of the most interesting content for example, as well as they will be able to select if an object they are interested in appears or not in the video.

1.3. Applications

Visual instance mining has some useful applications. One of them is that users can recognize with a simple glance if specific people or a particular object appears in that video. This allows the user to search for news where a particular person or a determinate object appears.

Additionally, as the instances show the most relevant content of the video, it can be used for video summarization. Our results sum up the storyline of the news. Seeing the visual instance mining that we extract, users can infer the overall topic for the news.

Moreover, it can be used as a complement to textual metadata. Nowadays, text is usually used to summarize news in digital newspapers and news websites. Sometimes textual descriptors are not enough but by using the visual instances that we extract, the user could be able to realize if the news is relevant to him or her.

Finally, the proposed approach can also be useful not only for news videos. We focused our thesis in this way but it can also be used for other types of videos such as films or personal videos. In this personal domain, our tool could be an interesting application because the amount of user generated content is increasing so much in the last years.

1.4. Outline of the thesis

The rest of the thesis is structured as follows. In Chapter 2 we describe the state of the art. We especially focus on the different blocks in which our work consists: faces, objects and text detection algorithms to extract semantic content from the video will be analyzed.

In Chapter 3 we analyze the system requirements in order to achieve our goal. In Chapter 4 we propose our developed solution. Using news videos, we are able to extract visual instances and we have developed a graphical user interface to show them to the user. In Chapter 5 we validate the tool thanks to a study conducted to several test users and a comparison with a system developed at Columbia University that creates a summary from news stories. Finally, in Chapter 6 and 7 we discuss our conclusions and what can be done as future work.

2. RELATED WORK

In this Chapter we describe the *state of the art* for visual instance mining and visual summaries techniques to achieve new levels of understanding. The first Section is an explanation of existing types of video summarization techniques in news domain as well as the state of the art of visual instance mining in section 2.2. Then, in subsequent sections we will explain technologies of the main parts of the process. In Section 2.3 we discuss about different face detection techniques. Finally, in Sections 2.4 and 2.5 we present object detection and caption detection methodologies.

2.1. News summarization

When we started talking about the focus of this thesis, we were discussing to change the domain of the previous work and we agreed on working in the news domain. Our first priorities were to assemble a ground truth from the newscast and try to find a benchmark.

In this Section we explain some existing tools in this domain. The first one we want to describe is the News Rover, developed in Digital Video & Multimedia Lab, in Columbia University.

2.1.1. News Rover[1] (2013)

News Rover is a web newsreader that presents a new method for finding messages. This tool extract information from Usenet newsgroups. The user just has to introduce some keywords and the application find all messages in newsgroup that match with these words.

News Rover [3][4] automatically get the keywords that the user has entered and searches for messages that are related to these words. Once all the information is ready, the graphical user interface of the tool shows the user all messages and pictures that the application has collected. Fig. 3,

[1] http://www.ee.columbia.edu/ln/dvmm/newsrover/publications/ (30/6/2014)

which is extracted from [3], shows the system architecture and the links between the different parts of the tool.

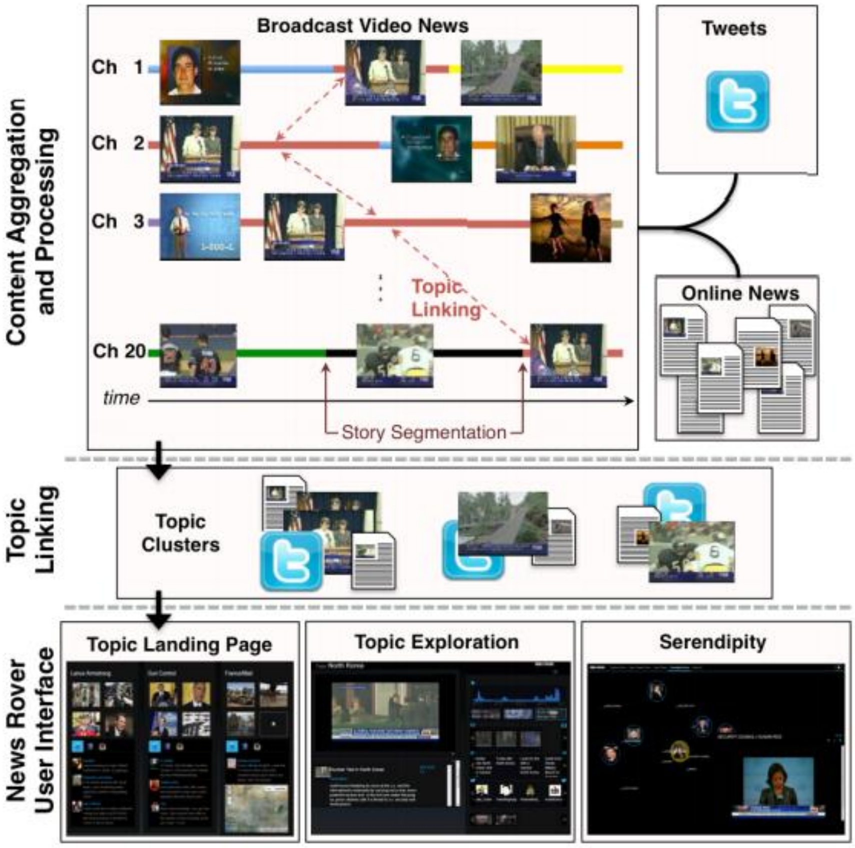


Fig. 3 News Rover's system architecture overview [3]

The main contribution of the system is that the tool is able to link and index content from a great variety of sources, including broadcast TV news, online articles, and social media feeds (like Twitter), organizing them into a structure by topics for a story context.

2.1.2. Name-it

When we talk about the news domain and looking for the state of the art, the theme that is worked more often is the relation between a face and the name of the person. The goal of the tool Name-it [5] is to link people with their name.

Name-It is a tool able to associate faces and names in news videos. The system only needs news videos, which include image sequences and written records obtained from audio tracks or caption texts. The system can then either deduce possible name candidates for a given face, or just find a face in news videos by name.

To achieve this task, the system carries out a video analysis approach: face extraction and a subsequent recognition from videos, name extraction from the written records or transcripts, and video caption detection and recognition. Each of these methods includes several images and language processing techniques: face detection, face identification or face recognition, intelligent name extraction using a dictionary and an analyzer, text region detection, character recognition, and integrates them in the system. In Fig. 4, extracted from [5], it is shown the different parts of the system and the relation among the different blocks.

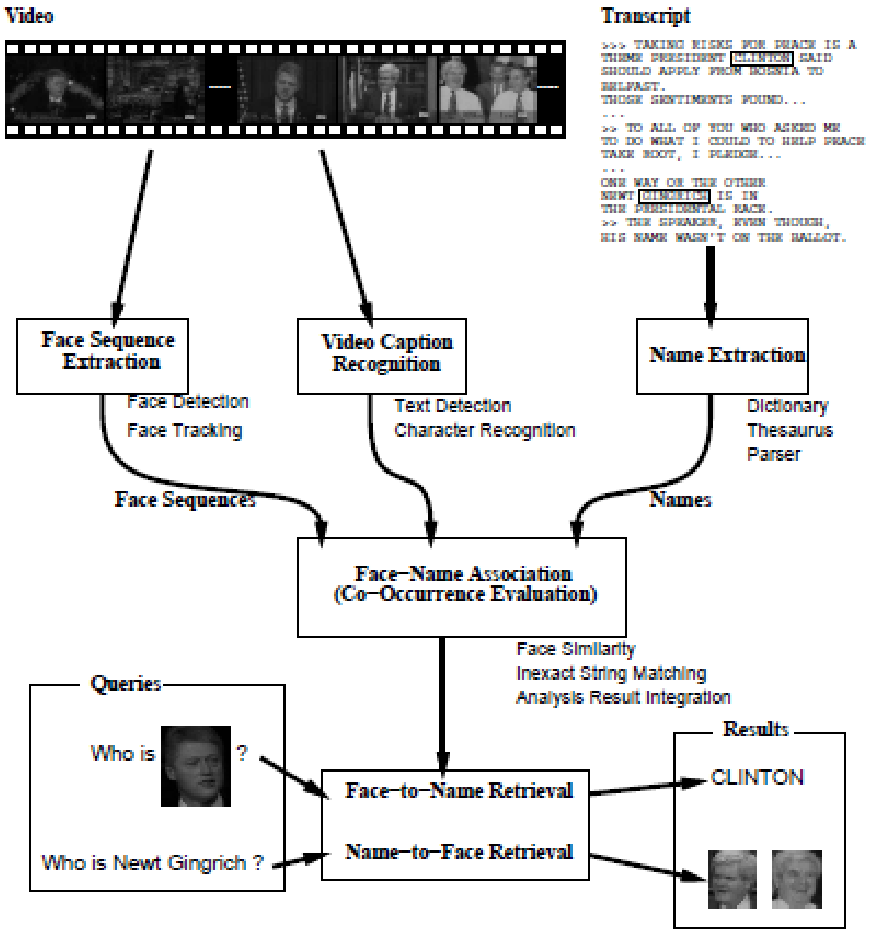


Fig. 4 Name-it System architecture overview [5]

As we have commented previously, the inputs they use are the video and the transcript. From the video, which is an image sequence, the tool extracts faces using the neural network-based face detector and then face similarity is evaluated using an eigenface-based method. From the video, the application also detects text and performs a caption extraction. From the transcription, the tool extract names thanks to a dictionary. Each name candidate should satisfy some conditions as positional score.

2.2. Visual instance mining

This section describe the state of the art of visual instance mining, which is the problem we are trying to solve with our thesis. We can define it as automatically discovering and extracting frequent visual instances out of a single video.

Work that addresses the same issue that we are considering in our thesis has been done. Josef Sivic and Andrew Zisserman in their paper [6] proposed a method that can extract the principal people, objects and scenes of video by measuring the frequency of appearance of viewpoint invariant features. As they commented, there are two important problems, which are firstly the fact that an object can suffer important changes in its appearance throughout a video (due to viewpoint and illumination change for example) and secondly because detections are imperfect, so that different configurations must be matched. They use the SIFT descriptor to detect these spatial viewpoint invariant features. We also mention and describe briefly the SIFT descriptor later in this chapter. In Fig. 5 we can see an example of the results extracted by this tool in a particular news videos. Each row shows ten samples from one cluster.

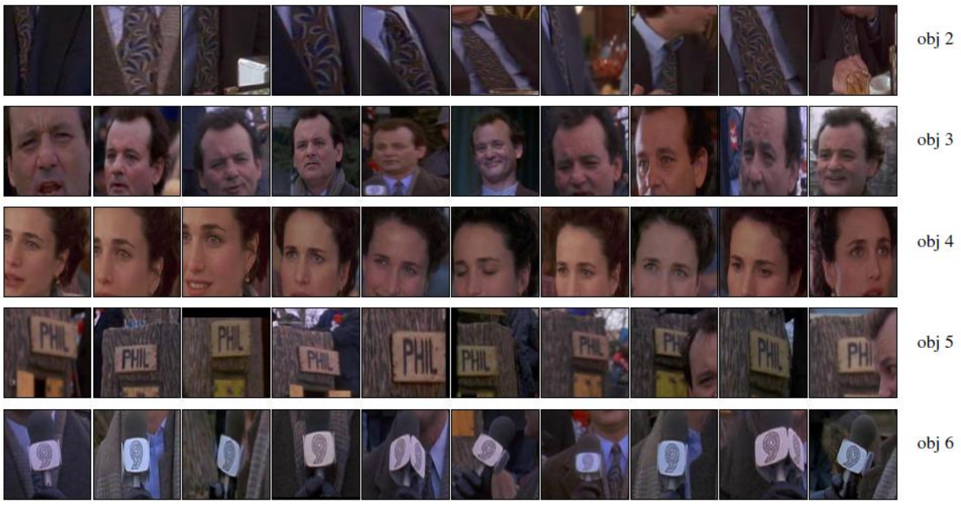


Fig. 5 Examples of mined clusters [6]

In the same line of working, there are other methods that try to obtain visual instance mining from a collection of images instead of a video in order to know what is the most representative content. Again Andrew Zisserman now with his colleague James Philbin developed a method to deal with a very large collection of images [7]. They present a procedure that groups images containing the same item, despite significant changes in scale or viewpoint. In order to cluster these images (more than one million) they use a graph-based matching algorithm. Fig. 6 shows three clusters for the 1M image Rome dataset. This figure also shows that the method can handle with extreme changes in imaging conditions.

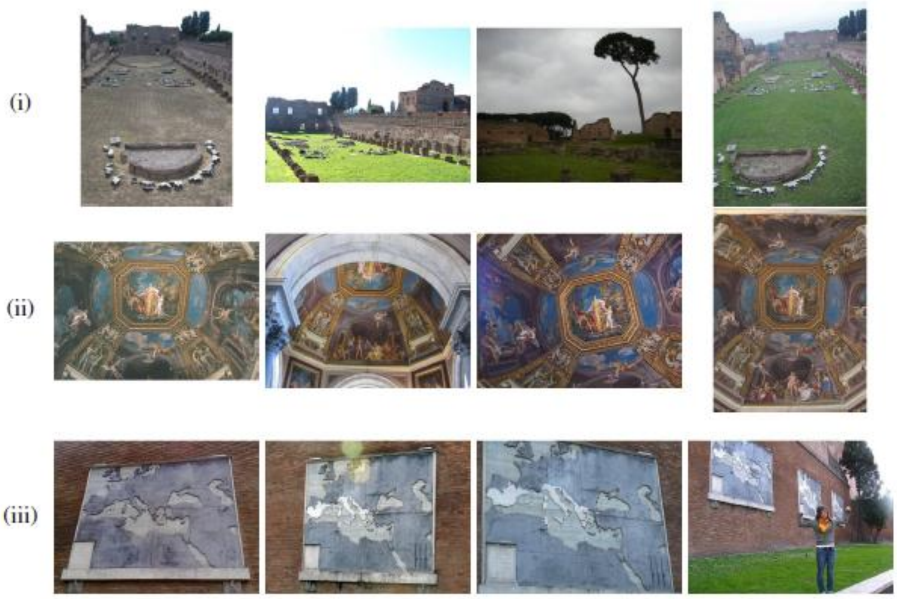


Fig. 6 Three clusters discovered from the Rome dataset [7]

In the next sections we will review different technologies used to construct the commented visual summary as well as techniques for detecting an extracting relevant and interesting object content (faces, objects and text).

2.3. Face detection and face recognition

In this section we analyze some methods and algorithms for face detection and face recognition.

We will focus on the Viola-Jones algorithm. The Viola-Jones [8] approach belongs to a block-based methods for face detection and currently is the most popular approach. This algorithm is capable of processing images quickly and achieving high detection rates. It is distinguished by three key contributions to the object detection field: *integral image,* a variant of *AdaBoost* for feature selection and *cascade of classifiers* focused to achieve an increased detection performance. As we have commented before it works fast and it also is scale invariant. However, it is not rotation invariant and requires long training time. This method was proposed by Viola and Jones and after it was improved by R. Lienhart [9] in 2002.

It uses different features. Fig. 7 shows the Haar-like wavelets used in OpenCV to extract the features of detected faces. These wavelets are simple functions at different scales and positions.

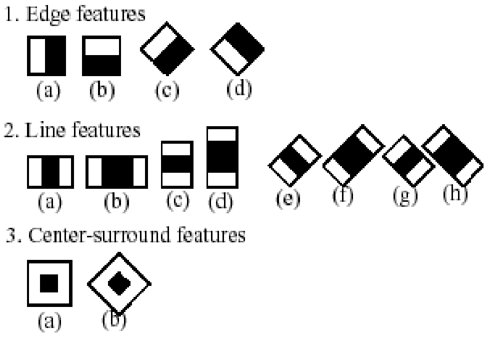


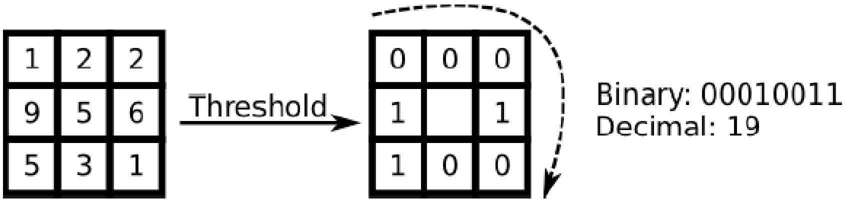
Fig. 7 Haar-like wavelets used in OpenCV[2]

Other algorithms for the face detection are on the one hand, Pixel-based methods, which are older. These methods firstly detect facial features as eyes, noses or mouths to deduce the existence of a face, and then group the pixels according to these features to fins a candidate face. On the other hand there are the Region-based methods [10] which initially create a partition of the images into homogeneous regions. After that, the method merges and analyzes these regions using visual attributes.

Next, we explain a method of features extraction in order to be able to recognize the faces detected. We focus on the Local Binary Pattern Histograms (LBPH). The basic idea of LBPH [11] is to summarize the local structure in a block by comparing each pixel with its neighborhood. As we can see in Fig. 8, each pixel is coded with a sequence of bits, depending on the relation between the pixel and its neighbors. If the intensity of the

[2] http://fileadmin.cs.lth.se/graphics/theses/projects/facerecognition/1_all_haar_wavelets.png (2/3/2014)

center pixel is greater or equal to that neighbor's, then the binary code is a 0; code with 1 otherwise.



1	2	2
9	5	6
5	3	1

0	0	0
1		1
1	0	0

Fig. 8 LBP code creation example [11]

It follows the following formula (see Eq. 1):

$$LBP\ (x_c, y_c) = \sum_{p=0}^{P-1} 2^p\ s(i_p - i_c) \quad \text{where } s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

Eq. 1

Then, as we can see in Fig. 9, face images are divided in NxN rectangular windows of equal size and one histogram is computed for each window.

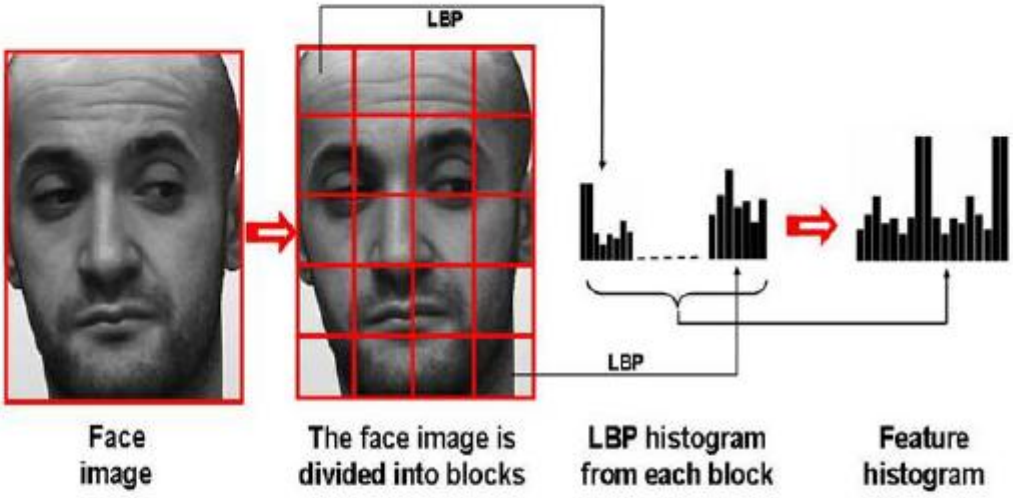

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

Fig. 9 Histograms are concatenated to generate a feature vector for each image [11]

Finally it can be applied the face recognition based on 1-Nearest Neighbor. We can compute different options for similarity:

- Histogram intersection: $D(S,M) = \sum_i \min{(S_{i,}M_i)}$
- Log-likelihood statistic: $L(S,M) =$ - $\sum_i S_{i,} \log M_i$
- Chi square statistic: $X^2(S,M) = \sum_i \frac{(S_i - M_i)^2}{S_i + M_i}$

Where S and M are the two face images to compare.

We would also like to comment some different methods to summarize features. First of all, there is the Principal Component Analysis (PCA) method. This method extracts features from structural data. The main concept that they use is that face images are very redundant and the face data lies on a lower dimensional variety. Other extracting features methods are the Bayesian, which was developed to improve the PCA results. The main different is that this method introduces a probabilistic measure instead of the Euclidean distance.

2.4. Object detection and object recognition

In this Section we mention and explain different object detection techniques. [12] makes an overview of all techniques used for object class detection.

As we know, object detection is not an easy task. We have to keep in mind that we need not only to determine whether or not any object appears in an input image, but also to locate them in this image to separate them from the background.

First of all we talk about the description of relevant visual items. According to their different levels of locality we can recognize three groups of descriptions. The first one is the pixel-level feature description, which is calculated at each pixel. The second one is the patch-level feature description, which is calculated at a small local areas in the image. On this level the SIFT descriptor and its variants stood. The SIFT descriptor [13] is probably the most known feature descriptor. This method encodes the information of the local area in a planar rotations and invariants to lightning changes. An efficient alterative to SIFT is Speeded Up Robust Features

(SURF) proposed by Bay in [14], which makes use of integral images to speed up the computation of descriptor extraction and evaluation as well as of key-point description. Finally, the third one is the region-level feature description, whose difference is that this 'local' region could be as big asthe entire image. In all cases, a popular aggregation of all the features contained in an image is based on the Bag-of-Features (BoF) method [15], which adopts a histogram-based region representation.

Next, we can talk about the different types of models depending on their structure. First, we have the window-based model, which computes the descriptor explained before inside a box that surrounds the object. But the object shape does not normally correspond to a box, so these representations will typically mix features from the object with the context that surrounds it. Another method which may adapt better to the shape and intra-variations of an object is the part-based model. This approach consists on a set of parts and the relations among them. This collection of parts depends on the shape of the object to be detected. Fig. 10 represents two objects detected by the deformable part-based method. Of this method we can differentiate three other models: the Star-structured models, which all the set of parts depend only on a central reference part, the tree-structured models, which includes connections in a same branch and the grammar-based models, which comprehend all part-based models, using variable hierarchical structures. The last method is the mixed model which can adapt its structure to variation in objects position.

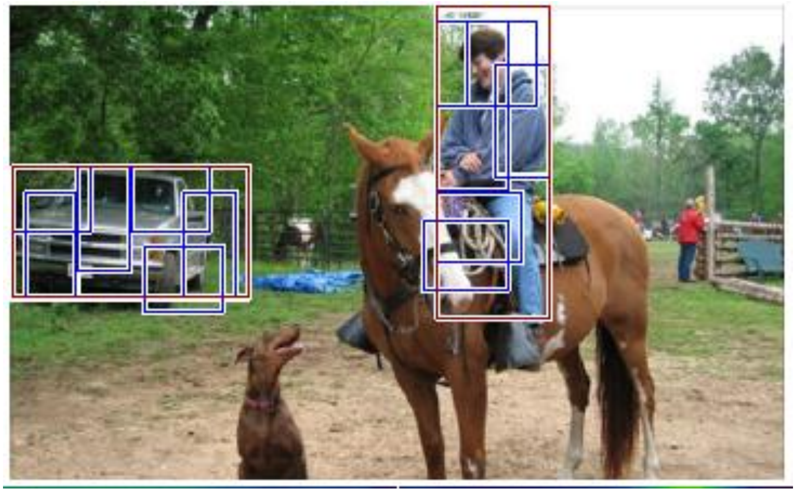


Fig. 10 Deformable parts-based example [16]

2.5. Caption detection

In this section we talk about the caption detection whose goal is to find out all text that appears in a video frame.

Boris Epshtein, Eyal Ofek and Yonatan Wexler [17] present an algorithm to detect caption and extract the text detected based on the stroke width of the caption for each image pixel.

The main parts of the detection algorithm are a Stroke Width Transform, grouping pixels into letters candidates, a filtering and finally grouping letter to construct words (see Fig. 11).

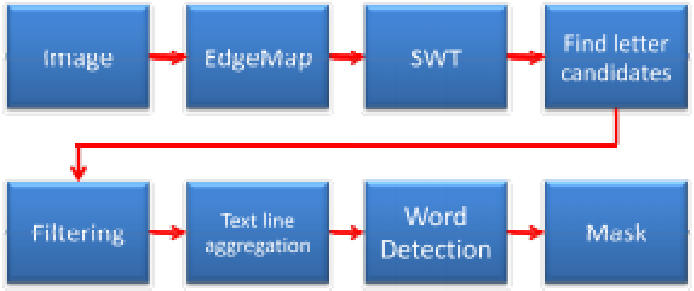


Fig. 11 The flowchart of the algorithm for caption extraction [17]

Following the previous figure, we explain the algorithm of this method. Firstly we explain what the Stroke Width Transform (SWT) is. It is an operator that computes for each pixel the width of the most likely stroke containing the pixel. In the paper define the stroke as "an adjacent part of an image that forms a constant width". The output of applying this transformation is another image with the same size as the input one in which each pixel contains the width of the stroke associated to this pixel.

Next step is to group these pixels into letter candidates. In order to group these pixels there must be a SWT ratio similarity between two neighboring pixels. Once letters candidates are found, then these letters have to verify a filtering method. First of all, components whose size is too large or too small are rejected. Remaining components are considered letters candidates and finally there is a block that cluster them into words or lines of text.

This block computes some criteria to determine words of the letters candidates. In order to determine these words, there must be similarities among the letters. Their stroke width, letter height and width must be similar and also the spaces between letters should be the same. Moreover, the color of the letters use to be the same in letter chains.

Initially, only pairs of letters are grouped. Then two pairs can be merged together if they share some characteristics. This process is repeated till it could not be more pairs merged.

Xinbo Gao and Xiaoon Tang propose in [18] a system in order to recognize text from Chinese news videos. They looked for characteristics that could make the detector more efficient. As we can see in Fig. 12, text in news videos tends to appear in the low part of the screen and usually in the center. This is for example, the position of subtitles or captions named in the news video. Keeping in mind this idea, they developed a method to extract text from news videos, focusing on recognizing text in the central section of a frame (see Fig. 13).

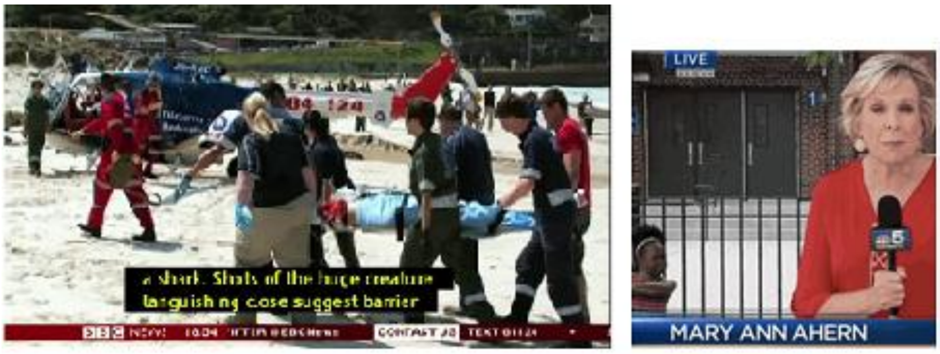

a shark. Shots of the huge creature
languishing case suggest barrier
BBC NEWS
LIVE
MARY ANN AHERN

Fig. 12 Position of text in news videos

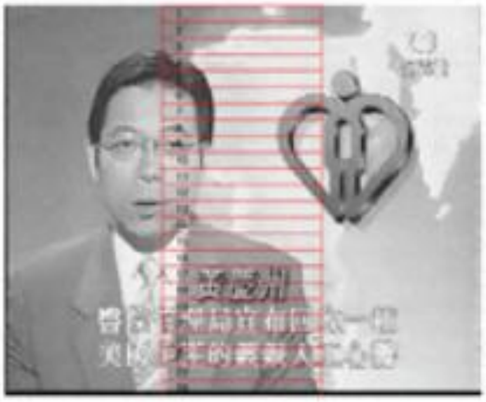


Fig. 13 Partition of the frame [18]

3. REQUIREMENTS

In this chapter we are going to explain the requirements our system should fulfill in order to be used as the different applications explained in Chapter 1. In Section 3.1 we develop a requirements analysis. We narrow down the content requirements in section 3.2 and structural requirements of our system are detailed in Section 3.3.

3.1. Requirements analysis

When we started thinking about this project and looking at what had already been done. The research questions that we asked were:

- Which is the state of the art of news video summarization?

The state of the art refers to the highest level of development at a particular time. We have commented in Chapter 2.1 the related work for video summarization in the news domain.

- How good can visual instances mining of video content be?

One requirement of our system is that the user must be able to understand the story of the news thanks to the visual instance mining with the most relevant content. This brings to the following issue.

- Which content must be selected to understand a news video?

We extract visual instance mining with the most representative content of the video in order to create a visual summary. Selecting the appropriate content is essential to obtain the best final result. We shall focus particularly on faces, objects and text. The approach should be validated by verifying whether its results fulfill the original user requirements. This also entails to the next question.

- How can we evaluate the visual summary results taking into consideration the subjective point of view of each user?

Evaluating our approach is an important task. Our system is evaluated by participants through a web-based survey. In this way we are able to evaluate the results of our final visual summary.

As we decided to develop the application in the C++ language, we thought that working in ImagePlus could be the best option in order to make the most of its libraries. ImagePlus is the software development platform of the Image Processing Group of the Technical University of Catalonia (UPC).

In the following sections we analyze which could be the requirements that we have to keep in mind when it comes to design and develop the software. The outcome of this analysis is a list of seven requirements grouped in two categories: content and structural.

3.2. Content requirements

The visual summary should provide the most representative content. We have to select which is the content to be shown in the visual instance mining to obtain the maximum information from the news videos.

Frequent people

The system has to focus on the most frequent people in the source video. Viewers naturally are interested in seeing the main characters that are part of the news; therefore, we have to keep in mind the time of appearance of people on the video. Usually, the most representative characters appears more often during the whole news than secondary ones.

Object selection

The system has to read which are the specific objects the user want to know if they appears or not in the video. It is interesting that users can decide which objects could be relevant for them. The application should be able to detect and recognize these objects and if found, show them in the resultant visual summary.

Caption extraction

The system should be able to recognize all text that appears in the news video and select the most important texts. Not all text have the same importance in the video. The application should select which of them could be relevant for the user. Related to what was commented in the first content requirement, the more time the text appears in the video, the more relevant it is in the news.

3.3. Structural requirements

These requirements provide presentation rules that can make the final visual summary more efficient. Our summary should provide non-redundant information to be effective.

Representative selection

The system must be able to looking the summary with the visual instance mining, the user could recognize the story and what is the news related to.

Region of interest

Users could decide the region to be shown in the final visual summary. The system should be able to show the region of interest detected as well as the frame in which the content appears. This will allow our system to be more efficient and satisfy the necessity of the users. Showing just the region of interest makes the user to focus on the content. However, if the whole frame is shown, more information to the user will be given. Moreover, it will help to maximize the understanding of the final summary. Nevertheless, showing the frame in which the content was selected may imply redundancy in the summary.

Diversity

We know that the more content we want to represent, the more possibilities of redundancy. The final visual summary should try to maximize showing as different content as possible by minimizing this redundancy.

This means that visually, the content included in the visual instance mining should be different of each other.

Computational cost

The user should be able to influence the computational cost of the final application. Depending on the time users are able to wait for the results of our applications, they should be able to choose to increase or decrease this time by also increasing or decreasing the efficiency of the tool respectively.

3.4 . Priorities

In this section we analyze the importance of the requirements described before on the development of our approach. The table below (Table 1) shows the priorities we assigned to all requirement. We designate the score 1 to these requirements that our system must fulfill with the highest priority. Priority with score 2 is set to these requirements that its influence in the final result is not essential.

Requirements	Priority
Content requirements	
Frequent people	1
Object selection	1
Caption extraction	1
Structural requirements	
Representative selection	1
Region of interest	2
Diversity	2
Computational cost	2

Table 1 Priorities of the requirements

We set the highest priority to select the most representative content. These requirements could be very important because they represent the main characters, the objects selection and the most relevant texts of the news video. We also establish priority with score 1 to the representative selection because, looking the visual summaries that our application generate, the story should be completely understandable by the user.

We decided to set priority with score 2 to the rest of the structural requirements. The computational cost and the region of interest requirements are decisions that the users can make according to their preferences. The diversity affects to the redundancy of the final result. It is difficult to manage but it does not concern to the understanding of the visual summary but to the effectiveness.

4. DEVELOPED SOLUTION

After considering the state of the art and the requirements that our tool should fulfill, we specify our approach in Chapter 4. In this chapter we further describe the implementation of the elements, methods and functions that we have used in the tool. In Appendix I there is the working plan with the time we spent in the different parts of our system. In Appendix III we explain how we developed the environment and the environment we used. It is also explain the usage of the tool and the inputs needed.

We implemented this software in C++ in the framework of the ImagePlus library from GPI (Image Processing Group of the UPC). We chose this library and development environment given its integration with OpenCV as well as additional methods that allowed us to both use a clustering algorithm and generate a graphical user interface as it is explained in section 4.4 and 4.5 respectively.

The rest of this chapter is structured as follows: Section 4.1 provides an overview of our approach that is further explained in subsequent subchapters. The following subsections explain the different blocks of the architecture.

4.1. Overview

First of all we have to process the video input in order to get images from this video. This processing is the temporal sampling. Once we have all frames from the video we can apply different algorithms to detect faces, objects and text in each frame. The following step is to group all these instances (faces, objects and texts). The clustering block is the responsible for carrying out this process. The next block select which are the most representative and relevant content of the clusters. Finally, the results of the selected items are shown to the user. Fig. 14 shows the process that our system follows.

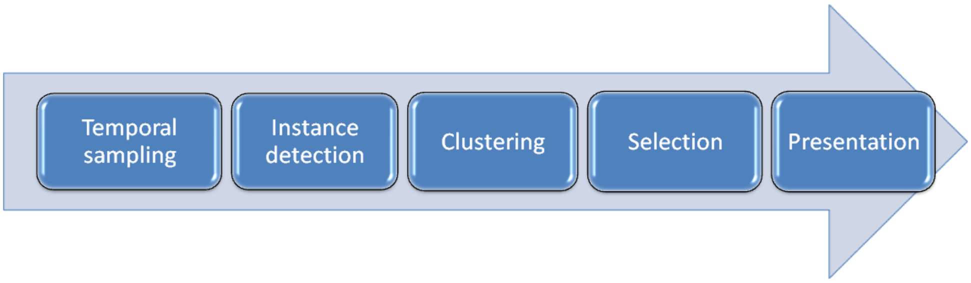


Fig. 14 Proposed system architecture

In next sections it is described all block's implementation we have used for the application developed.

4.2. Temporal sampling

To create the video summary it is required a temporal segmentation of the source video.

At the beginning of this project, we were talking about working with a graph-based shot detector cause we think that this could be the best implementation for the tool. Due to some problem running some of the techniques based on this method and integrating them in the implementation, we finally decided to use the uniform sampling and work with a graph-based method for the clustering, with is explained in Section 4. These problems are basically errors during compiling and running the programs and codes. We tried to compile the source codes with MinGW and cMake, but with no successful results, so we decide to use the uniform sampling which is simple and easy to implement using OpenCV libraries .

So first, we perform a uniform sampling of the source video. This uniform sampling of the input video is carried out with the class VideoCapture[3] from the external library.

So that the user can decide which is the computational cost of the final application, we developed a system that the users can choose how many frames per second (fps) do they want to get. With this mechanism we fulfill the requirement of the computational cost, explained in Chapter 3.

First of all, we have to know which is the rate of the original video and ask the user for a desired rate. In order to know the original video rate we call the method *get(CV_CAP_PROP_FPS)* from the class VideoCapture. Once we know this parameter we use an easy equation (See eq. 2) to know which frames must be processed.

[3] http://docs.opencv.org/java/org/opencv/highgui/VideoCapture.html (12/4/2014)

$$\#\ of\ frame\ to\ process = \left\lfloor \frac{original\ video\ rate}{user'sdesired\ rate} \right\rfloor$$

Eq. 2

This number of frames to process are the frames which will enter in the detection block. You can see Fig. 15 and Fig. 16 that show the selection of these frames.

We round the value downward, returning the largest integer that is not greater than the result of the division, because this could not be exact. Then, we will only process frames which are multiple of that resulting number. In Fig. 15 we can see that we only get the frames that are multiple of fifteen.

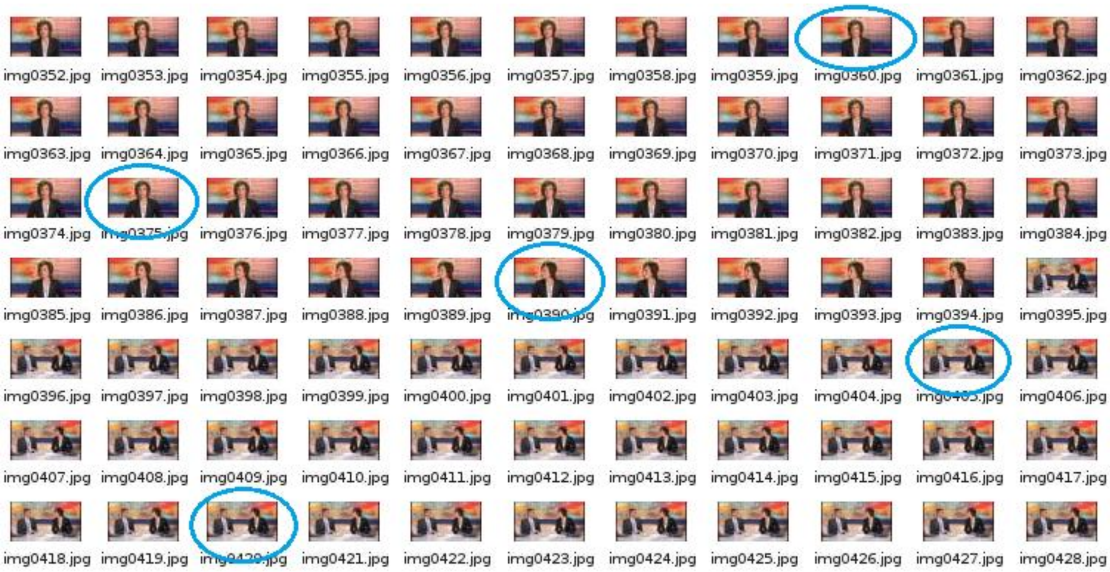


Fig. 15 Uniform sampling and selected frames to process related to user's desired rate

To sum up, we perform a uniform sampling with OpenCV tools and we process an specific number of frames according to a user's parameter, the desired rate.

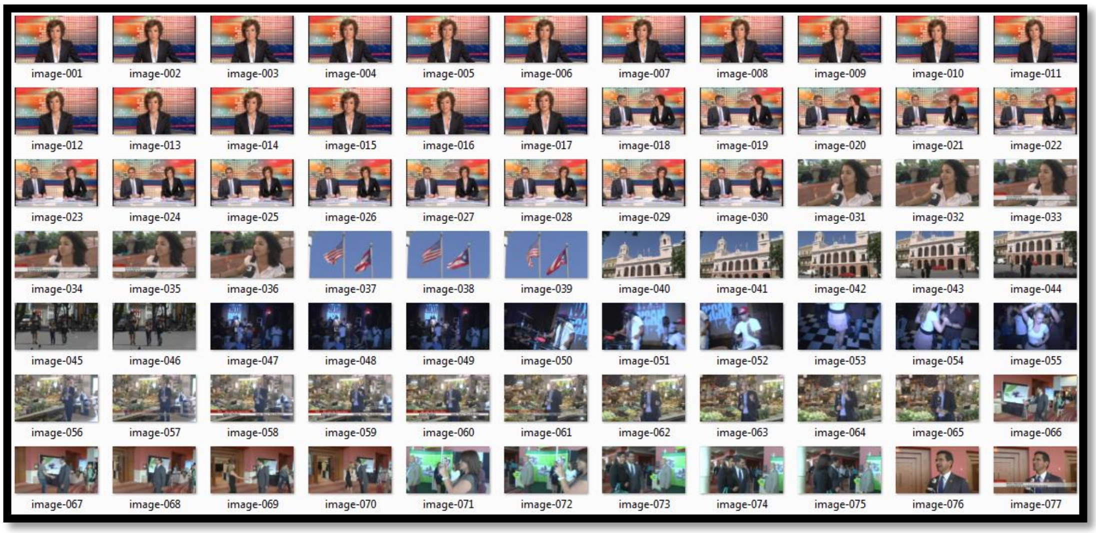


Fig. 16 Output frames after uniform subsampling

4.3. Instances detection

In this Section we explain the different methods for content instance detection we have used to develop the tool. We also comment the algorithms and why we have chosen them. We start focusing on face detection, followed by object and text detection.

4.3.1. Face detection

In order to detect faces of each frame we run a generic face detection engine provided by OpenCV based on the Viola-Jones algorithm [8], which is explained in Section 2.4. In this thesis we focus on detecting frontal faces. We do not take into account right and left profile cause we consider that the most relevant and representative people of a news video must be shown in a frontal view.

We load the Haar cascade face classifier based on the Viola and Jones algorithm, and apply the detectMultiscale method from OpenCV in each frame to detect all faces in it. In order to use this method we need the following parameters:

- The Haar classifier cascade (OpenCV 1.x API only). It can be loaded from .XML or YAML file using the method load().

- The input image Mat of the type CV_8U .

- The vector of rectangles where each rectangle contains a detected face.

- A scale factor that specifies how much the image size is reduced at each image scale.

- A minimum number of neighbors which specifies how many of them each candidate rectangle should have to retain it.

- A minimum and a maximum face size. Faces smaller than the minimum or larger than the maximum are ignored.

Using the vector of rectangles we are able to compute the coordinates of the detected faces in a frame. The detected faces are boxed with a green color. In the next Fig. 17 it is shown how the face of the anchorwoman is detected in one of the news video we are testing with.

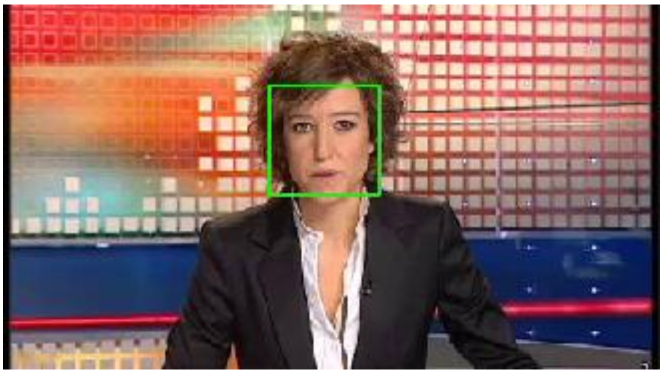


Fig. 17 Face detected in a frame

4.3.2. Object detection

The previous section has focused on a very specific class of instance: faces. The system should be able to work with any kind of object class. The solution we have applied is to adopt a generic object detector and allow the user to detect the object of interest they prefer. This project aims at providing the users with the opportunity to choose the object classes they want to detect.

We have developed a solution based on the matching of SURF features [14]. This approach is described in Chapter 2.5.

We have chosen this method because it could be easily implemented with OpenCV. As we started working with OpenCV developing the sampling and the face detection, we became familiar with its classes and methods. We thought that we could make the most benefit using these libraries.

In contrast to a cascade classifiers, used for the face detection, training data is required for this solution. The user needs to provide a collection of example images of the object to be detected as an input parameter. We create a directory with different sub-folders, one folder for each object class. In every sub-folder, there is a collection of images of the same object class extracted from ImageNet[4]. In this website we can find images of several object classes. It is also very important that the training image will mainly show the object with no other salient instances that may confuse the detector. Fig. 18 shows a example of training images of an ambulance that we have used in our experiments.

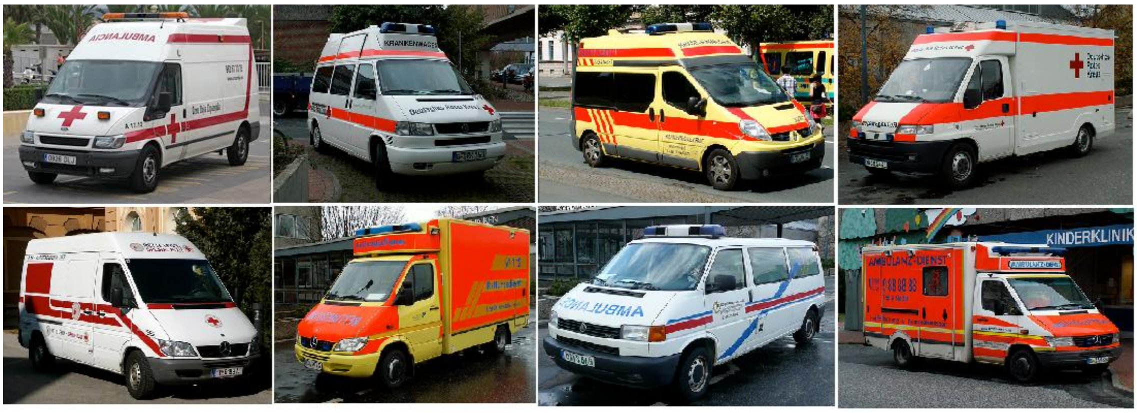


Fig. 18 Ambulances training images extracted from ImageNet

As commented in [2], the strength of this method relies on being scale and rotation invariant, robust, fast and most importantly, its ability to work with a single training image. We split this detection algorithm into two stages: descriptors extraction and matching strategy.

First of all we have to extract the keypoints and SURF descriptor from the training images. We use the class *SurfFeatureDetector*[5] and its function *detect* from OpenCV to perform the detection of these keypoints. In order to

[4] http://www.image-net.org/ (22/3/2014)

[5] http://docs.opencv.org/doc/tutorials/features2d/feature_detection/feature_detection.html (26/4/2014)

find the feature vector correspondent to the keypoints we use the class *SurfDescriptorExtractor*[6] and its function *compute*. Fig. 19 shows the keypoints of an ambulance and a police car that we used while we were testing the method.

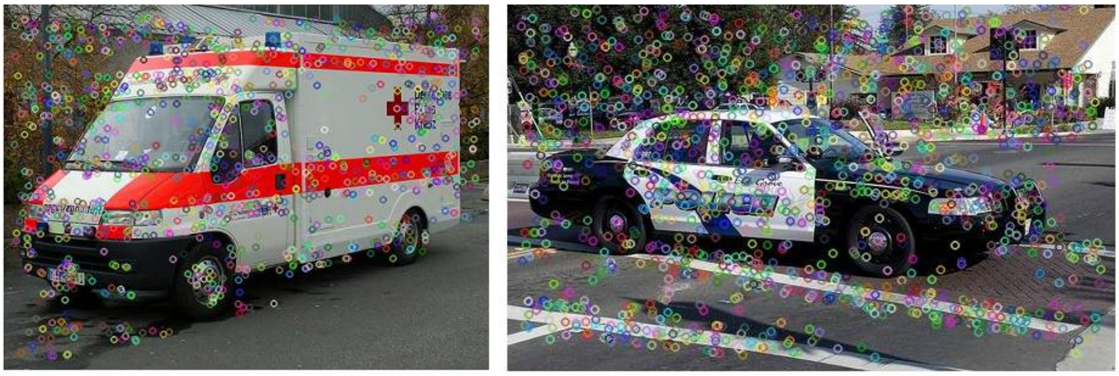


Fig. 19 Detected keypoints from reference objects of training images

Then, for each considered video frame it is also needed to extract its keypoints and calculate the SURF descriptors with the same functions described before. Next, we match the descriptor vectors of the training images with the descriptor vectors of the frame using a Brute-force descriptor matcher (*BFMatcher*)[7] and its function *match*. After that, a quick calculation of maximum and minimum distances between keypoints is carried out. We use the function *distance* of the class *DMatch*[8] from OpenCV. Once we have calculated these distances, we only keep the "good" matches, this means that we only keep in mind these matches whose distance is less than $2 * minimum\ distance$ or a small heuristic value (0.02) in the event that the minimum distance is very small, as it is proposed in the link below.

The following Fig. 20 shows all matches between the reference ambulance that we have extracted the keypoints in Fig. 19 and an ambulance in a keyframe of a news video. In general, good matches are shown as horizontal lines, while 'bad' matches are shown diagonal.

[6] http://docs.opencv.org/doc/tutorials/features2d/feature_description/feature_description.html (26/4/2014)

[7] http://docs.opencv.org/modules/features2d/doc/common_interfaces_of_descriptor_matchers.html#bfmatcher-bfmatcher (13/4/2014)

[8] http://docs.opencv.org/java/org/opencv/features2d/DMatch.html (13/4/2014)

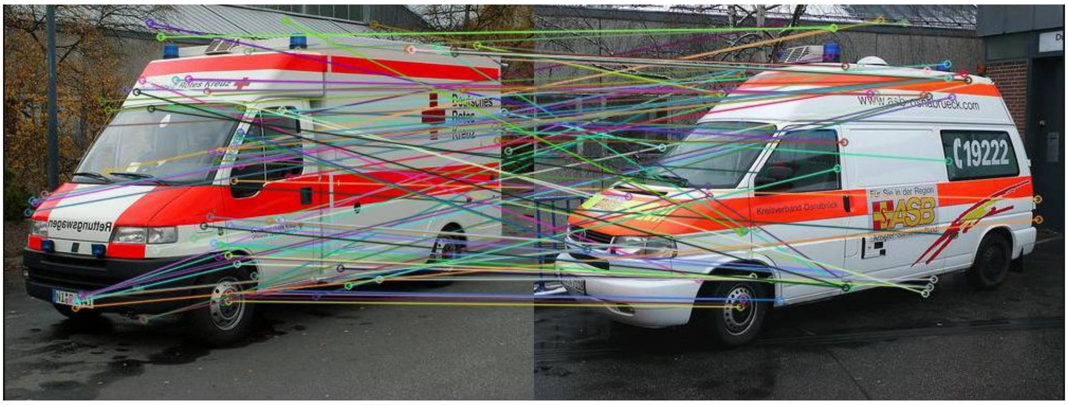


Fig. 20 Matches of the reference ambulance and one in a frame

As we have commented in the previous paragraph, there will be "good" matches and "false" matches, but we only keep the "good" ones. Now, knowing the percentage of "good" matches we can decide if the object appears in the frame. For this reason we set a detection threshold based on this rate. In Eq. 3 we can see how we compute this threshold:

$$detection\ threshold < \frac{\#\ good\ matches}{\#\ total\ matches}$$

Eq. 3

After some empirical test (see Fig. 21 and Fig. 22), we set the this detection threshold to 0.6. All frames that its detection threshold value is higher than 0.6 will be considered that an object was detected, the rest will be rejected.

In Fig. 21 and Fig. 22 we can see the results of these tests. We set the threshold to 0.6 because in tests, it is the best percentage of correct detections among all detections. With a threshold of 0.9 in the test with ambulances and a threshold of 0.8 and 0.9 in the test with police cars we do not obtain any 'good match'.

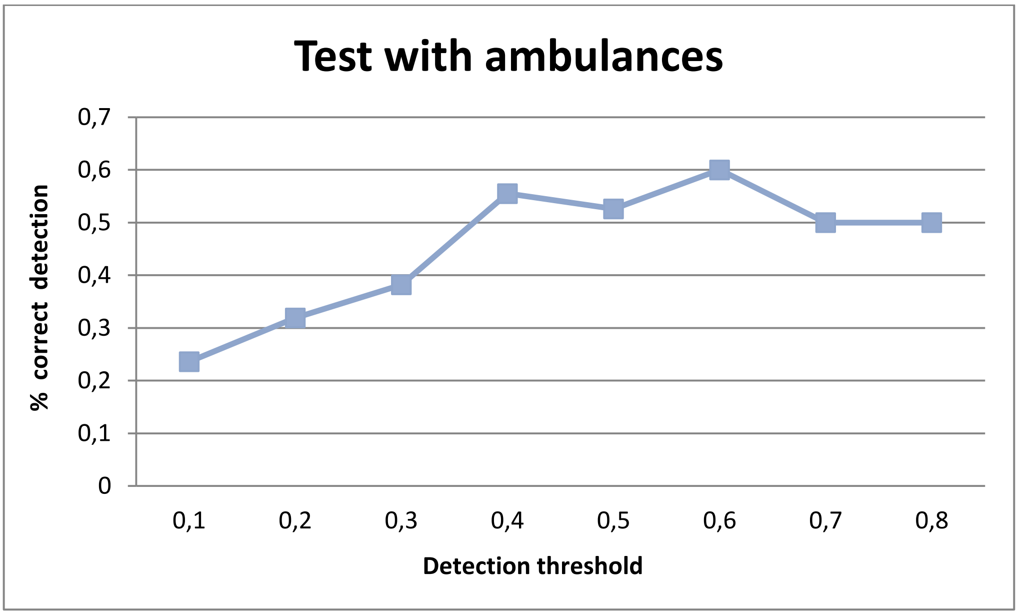


Fig. 21 Test done with ambulances to set the detection threshold

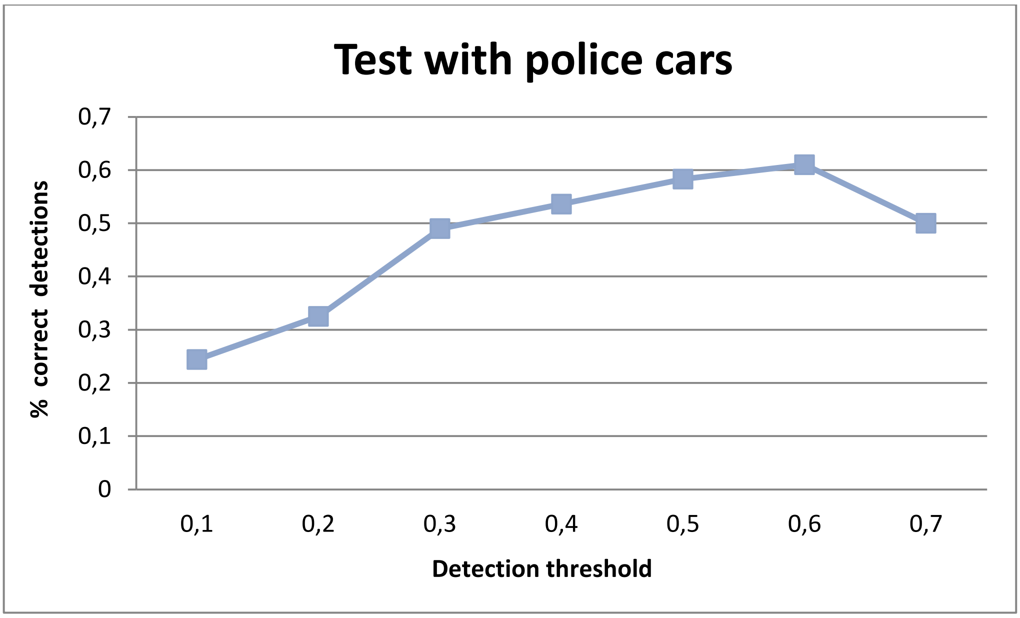


Fig. 22 Test done with police car to set the detection threshold

Finally, it is time to get the region of the object. To extract the region of interest (ROI) we get the values that belong to a good match located in the maximum and minimum X axis and the values that belong to a good match located in the maximum and minimum Y axis. Once we have this values, we are able to draw a bounding box and afterwards extract the desired region of interest. In the following Fig. 23 it is shown how the ambulance that we could see in Fig. 20 has a detection threshold higher than 0.6 and it is considered a detection.

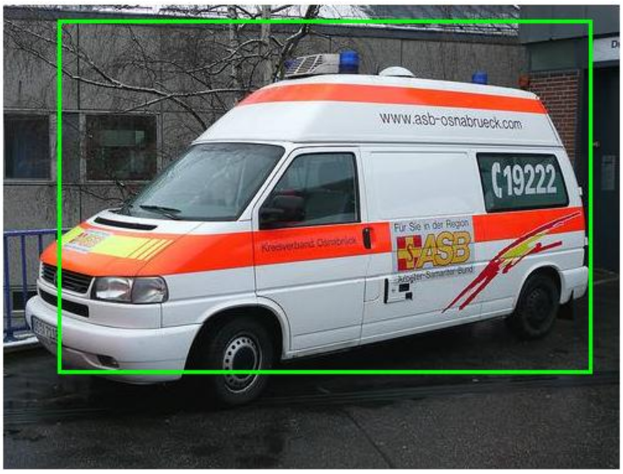


Fig. 23 Object detected in a frame

4.3.3. Text detection

This section focuses on how to extract text from a video source. As we were working with ImagePlus, an implementation of the algorithm presented in [17] and developed by Anna Gimferrer in [19] was available and it was chosen to perform the text detection. As it had been developed in the same platform that was used in this project, it was easy to implement and integrate this method in our tool.

The text detector is based on estimating the stroke width as it is explained in Chapter 2.6. In the next paragraphs we explain how this algorithm works.

Initially, an edge detection is required in order to find the width of the line components. In Fig. 24 we can see, on the right side, the edge detection of a frame from a news video.

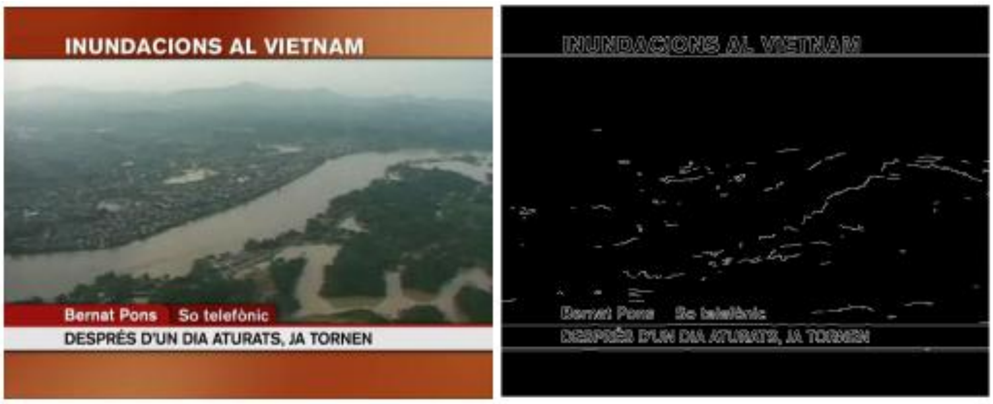


Fig. 24 Edge detection of a frame [19]

Then, we run a function twice, which is described in the following paragraphs. Once to find dark letters on a light background and then to find light letters on a dark background.

This function consists in, from the output image of the edge detector, drawing lines from one edge pixel to another. The directions of these lines are the gradient directions because they point to the direction of maximum intensity growth. When a line reaches an edge point, it is looked the gradient from this point but in the opposite direction and, and if it is equal to the initial point, allowing a certain tolerance, it creates a beam between these two points. This condition appears from the fact that the width of a line is defined as the distance between two points of the edge, which are parallel, and therefore their gradients must have the same direction.

This process is done for all pixels of the edge. If a line does not find any boundary pixel with the opposite gradient to its or reaches the limits of the image, it is discarded. At this point, the stroke width can be assigned to each pixel.

Once the stroke width of all pixels is computed, they are merged to create the candidates to be components. In order to group the pixels into components, it is processed the pixels of the whole image. For those pixels with a stroke width value assigned, it is studied if they can be merged with some neighbor pixel. Two neighbor pixels are merged if the relation between their stroke width is smaller than 3. This condition was proposed in [19]. This process is computed for all pixels and we obtain several groups of connected pixels which are called components instead of candidates. Fig. 25 shows the components created from dark text on a light background (middle image) and the components created from a light text on a dark background (right image) from a original frame (left image).

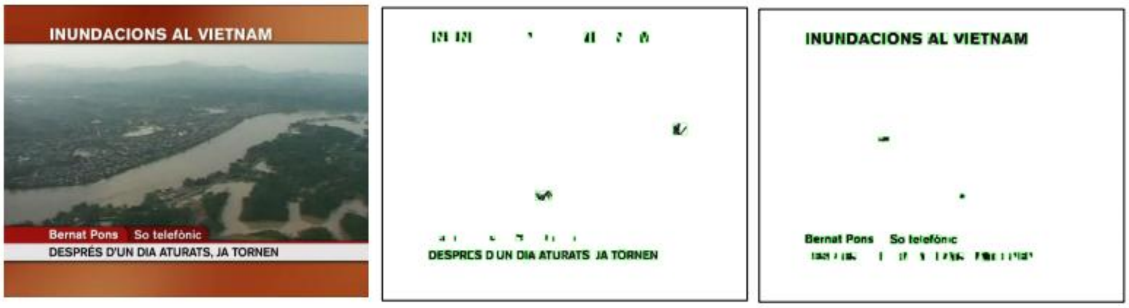


Fig. 25 Components (letters) detected from an image [19]

Once we have all the possible components, they have to fulfill a series of geometric conditions proposed in [19] in order to be considered chains of letters.

Finally, we have two groups: on the one hand, dark chains of letters on a light background and on the other hand light chains of letters on a dark background. We compare chains of both and when two chains with a common area of the image are found, one of them is discarded. If one of them is included in the other, this one is removed, if not, the shortest chain is removed. Fig. 26 shows an example of bounding box of different chains of letters detected by our system in a news video while testing this method.

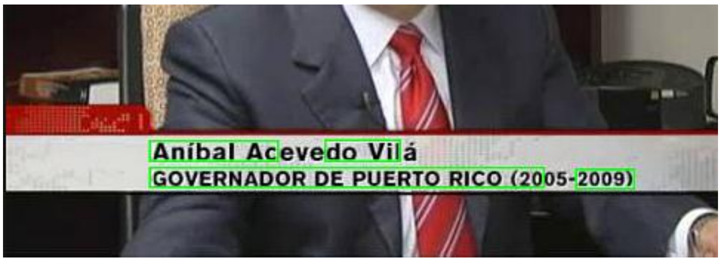


Fig. 26 Text detected in a frame

4.4. Graph-based selection of visual instances

Once we have all the instances, it is time to cluster them and select which of them are the most representatives of the video. For this task, we decided to use a graph-based method to cluster the instances and then we use the mutual reinforcement algorithm [20] to choose the representative instance for each cluster, which is just the final step of our algorithm.

Probably this is the most important block of the whole tool cause the final selection of which visual instances are the most representative, important and relevant depends entirely on this step. The most relevant contribution of this Thesis is this section. This block must process all detections to decide:

- Which detections belong to the same instance?
- Which detections appear more often in the video?

We are going to explain the process all detected faces follow to finally get the most representative. The development is exactly the same for the other type of instances: objects and texts. Fig. 27 shows an overview of the different blocks to develop in order to select the most representative instances having all detections.

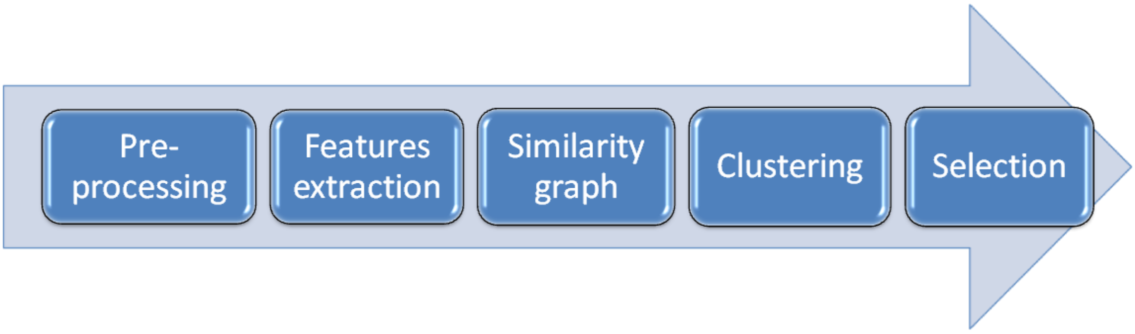


Fig. 27 Overview of the different stages of the clustering block

4.4.1. Pre-processing of detection boxes

As Manuel Martos suggested in his Thesis [2], the instances selected should undergo to a pre-processing in order to improve the accuracy. He commented "One of the most important problem is the sensitivity to lightning conditions. This problem may prevent the recognition of a same person whether if s/he is in a dark or bright room. In addition, the instance should be in a very consistent position within the detected bounding box, not including pixels coming from the background or hair" in faces.

Grayscale images are used for recognition, so the first step of our pre-processing is to convert RGB images to grayscale. Secondly, the image is cropped in order to remove background pixels that only add noise to the recognition process. For OpenCV frontal face detection, 20% of the edge pixels are removed. Resizing the image to a preset size is the next step and, finally, histogram equalization automatically standardizes the brightness and contrast of all facial detections. Fig. 28 shows all stages of the pre-processing part.

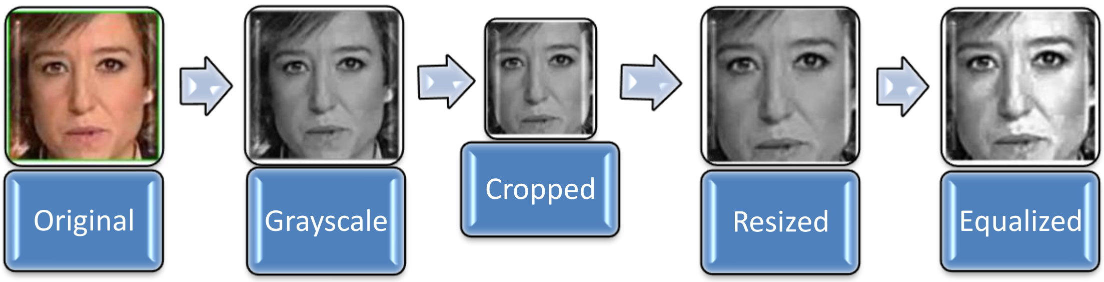


Fig. 28 Pre-processing faces to improve the accuracy

4.4.2. Features extraction

Once we have all instances from each detection pre-processed. First, we create the LBPH[9] (Local Binary Pattern Histogram), explained in Chapter 2.3, of each instance, also using the implementation in OpenCV.

Then, we compared each calculated histogram with one of the similarity metrics provided in OpenCV. Our purpose is to obtain a similarity value between instances, so we tried the Histogram Intersection, which gives already provides a similarity value between 0 (dissimilar) and 1 (similar), and the Chi-square distance which requires an adaptation to be mapped between 0 and 1. After testing these two metrics we agreed to use the Chi-square distance, which gives more accurate results[10].

Using this Chi-square distance, the larger the value, the better the match. Because of that, we added a simple exponential transformation (see Eq. 4) to obtain a similarity value where the higher the value, the more accurate the match. This transformation is needed to build the similarity graph in the next stage.

$$\text{Similarity value} = e^{(-\alpha * \text{distance})}$$

Eq. 4

In order to simplify, we are taking $\alpha = 1$.

4.4.3. Similarity graph (Full connectivity)

Once we have the descriptor of the instance (LBPH) and a similarity value capable of comparing this descriptor, we build the similarity graph [21][22] between each detected instance. This graph is built by computing the similarity values between each detected instance, connecting all instances with all of them. Fig. 29 shows the result of the full connectivity once we have applied the similarity graph.

[9] http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms (19/2/2014)

[10] http://compvis.readthedocs.org/en/latest/histograms.html (19/2/2014)

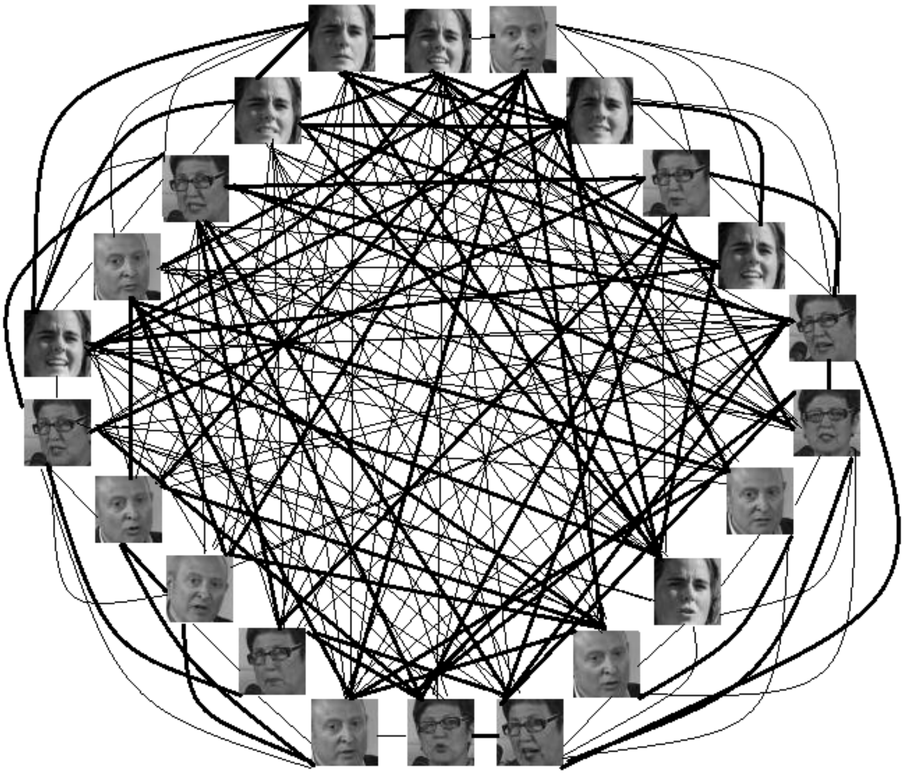


Fig. 29 Similarity graph with full connectivity

4.4.4. Clustering by Edge filtering

When we have all connections we will only keep those links between those instances whose similarity exceeds a certain threshold. This threshold is different for faces, objects and texts. After knowing the similarity values of the elements (images) in the graph and carrying out some empirical tests, we set these thresholds to:

- Threshold for faces: $1e^{-24}$
- Threshold for objects: $1e^{-10}$
- Threshold for texts: $1e^{-6}$

As a result, instances appear naturally clustered in sub-graphs, which correspond to different instances. This stage is implemented by coding the similarity graph with the The Boost Graph Library[11], and using on it the sub-graph class[12].

[11] http://www.boost.org/doc/libs/1_55_0/libs/graph/doc/index.html (19/2/2014)

[12] http://www.boost.org/doc/libs/1_38_0/libs/graph/doc/subgraph.html (19/2/2014)

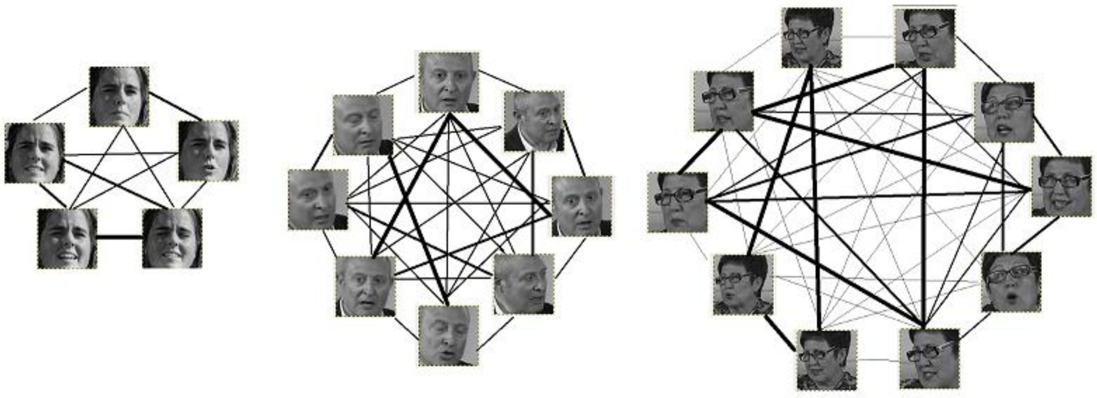


Fig. 30 Example of sub-graphs

In Fig. 30 we can appreciate three sub-graphs, where the nodes are the instances detected in the input video and the lines which link each instance with others inside the sub-graph are thinner if these two instances are less similar or thicker if are more similar.

4.4.5. Selection of the representative visual instances

The last stage of the selection on the representative detection.

In order to select the most relevant content we set a minimum threshold related to the minimum time we believe that the content must appear in the video to be considered important of this video.

We keep those subgraphs whose amount of nodes exceed the predefined threshold. The size of the subgraph is related to a tunable parameter that the user enter as an argument and this one is related to the number of seconds that the content appears in the video. For example, if a person appears during 15 seconds in the video and the user's desired rate is 1 frame per second, that means that there will be 15 frames where the instance of this person must be detected. Once we have clustered the ROI of these detected instances, the size of the sub-graph (number of nodes) that belongs to this person should be 15.

We did some empirical testing to determine how much time must the content appear in the video to be considered as representative. We were working with videos of 1-3 minutes. After these tests, we have decided that we can consider content relevant if it fulfills the following rules:

- A person is considered representative if he or she appears more than 5 seconds in the video.

- An object is considered important if it appears more than 3 seconds in the video.

- A text is considered relevant if it appears more than 3 seconds in the video.

We choose that parameters because with them we have obtained the best results of the application in terms of redundancy and accuracy.

Once the sub-graph has fulfilled the criteria explained before, it is time to use the mutual reinforcement algorithm that was used in a recent paper at CBMI 2013 [20] on keyframe selection. This algorithm had been previously proposed by [23].

The algorithm assigns a relevance score to each node in each sub-graph, which estimates how important each instance is inside the graph. The maximum value is considered to represent the most representative instance in the sub-graph and chosen to represent each instance in the video.

Finally, it is time to decide which instance of the sub-graphs is the most representative, which is the image with the highest coefficient inside the sub-graph. In Fig. 31 it is shown the selection of the representative detection of the sub-graph.

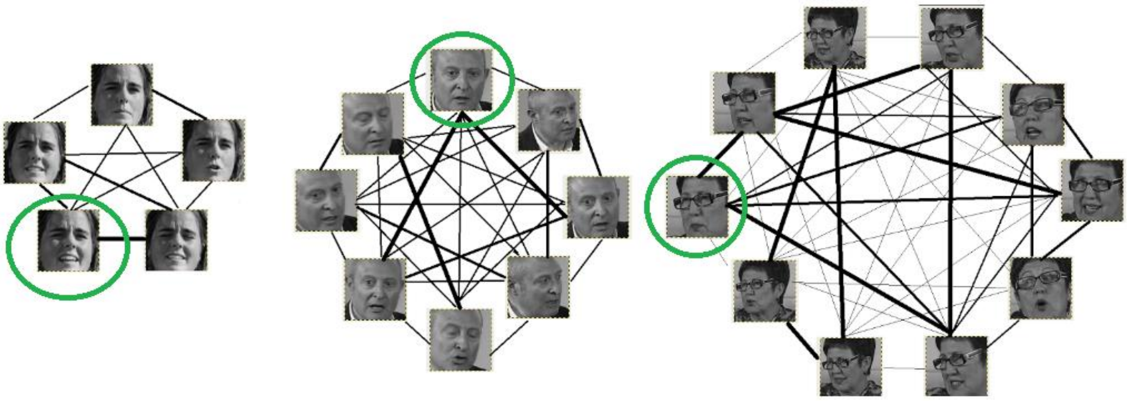


Fig. 31 The representative face of each sub-graphs

4.5. Presentation of the results

In order to show the results of our tool, we have created a graphical user interface (GUI). In this subchapter we explain the GUI that we have designed.

With the development of this interface we want to fulfil some aspects of the thesis. First of all, we thought that it could be the best way to show the extracted visual instances of our tool. Moreover, we could get an online demo.

After evaluating the possibilities that we have to create the graphical user interface, we decided to develop the GUI using C++ with Wt. Marcel Tella developed in his thesis [24] an algorithm demonstrator of an ImagePlus tool using Wt. As it is also developed in the ImagePlus library we are working in, it could be easier to call the functions of the tool by the web interface.

4.5.1. Graphical user interface with Wt

Starting with the Wt GUI[13] development, we initially developed a structure as it is shown in Fig. 32. Where the widgets are a file to upload (in our approach it is a video file), some objects to check if the user wants to detect them (some specific objects) and a text editor to introduce the desired frame rate (a number between a fixed range).

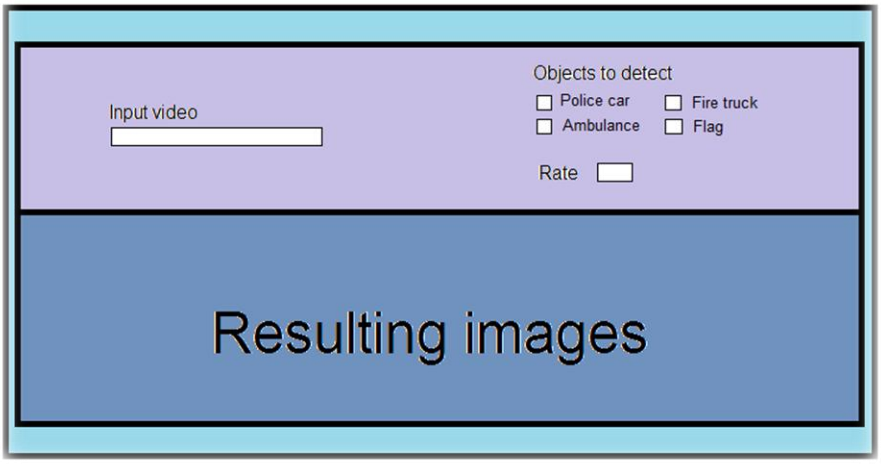
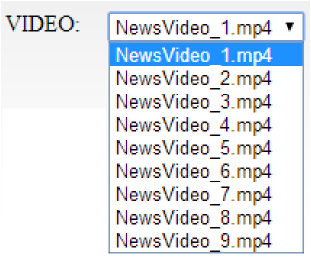


Fig. 32 Initial prototype design of the GUI we want to design

[13] http://www.webtoolkit.eu/wt (1/7/2014)

Testing how the WUploadFile widget works, we noticed that it takes a lot of time to upload a video file selected by the user and it also needs many resources from the Image Processing Group cores. Finally, we decided to select videos of the CCMA dataset that our group has and create a list of that videos. As they are already in the dataset of the UPC GPI, it is not needed to upload the video files, the user just has to select a video from the list.

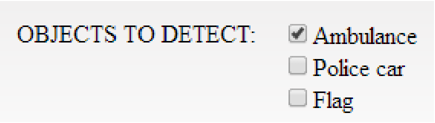


VIDEO: NewsVideo_1.mp4 ▾

NewsVideo_1.mp4
NewsVideo_2.mp4
NewsVideo_3.mp4
NewsVideo_4.mp4
NewsVideo_5.mp4
NewsVideo_6.mp4
NewsVideo_7.mp4
NewsVideo_8.mp4
NewsVideo_9.mp4

Fig. 33 List of some videos users can select

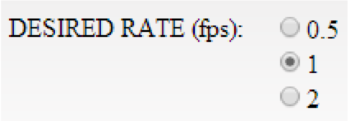
A WCheckBox widget allows users decide if s/he wants to detect a specific object or a group of them in the selected video by just checking the boxes of a particular list of objects.



OBJECTS TO DETECT:
☑ Ambulance
□ Police car
□ Flag

Fig. 34 Check buttons with some specific objects

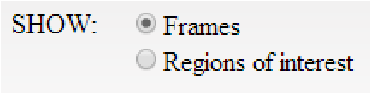
To introduce the desired rate to process the video we thought initially about a WLineEditor widget. In order to restrict the user text, we introduced a validator which ensures that what the user introduce is a number in a certain range. But with this kind of validation, the numbers must be integers. This means that a rate <1 and >0, for example 0.5 (1 frame each 2 seconds) could not be possible. To solve this problem, we decided to insert a WRadioButton widget. We set three rates that the user can decide to use: 0.5, 1 and 2. Using a rate higher than 2 frames per second, the computational cost increases and for creating this demo we thought that it is enough to test the tool on these frame rates to check how it works. We defined that rates exclusively so that only one of them could be selected.



DESIRED RATE (fps): ○ 0.5
● 1
○ 2

Fig. 35 Widget to select the frame rate

Finally, as our tool can extract the region of interest as well as the frame where the ROI appears, we thought that it could be useful to set another WRadioButton widget applying the same concept of exclusivity as we have defined before. With this new widget users can decide if the application has to show the frame with a bounding box in the ROI or just the region of interest.



SHOW: ● Frames
○ Regions of interest

Fig. 36 Widget to select the region to show

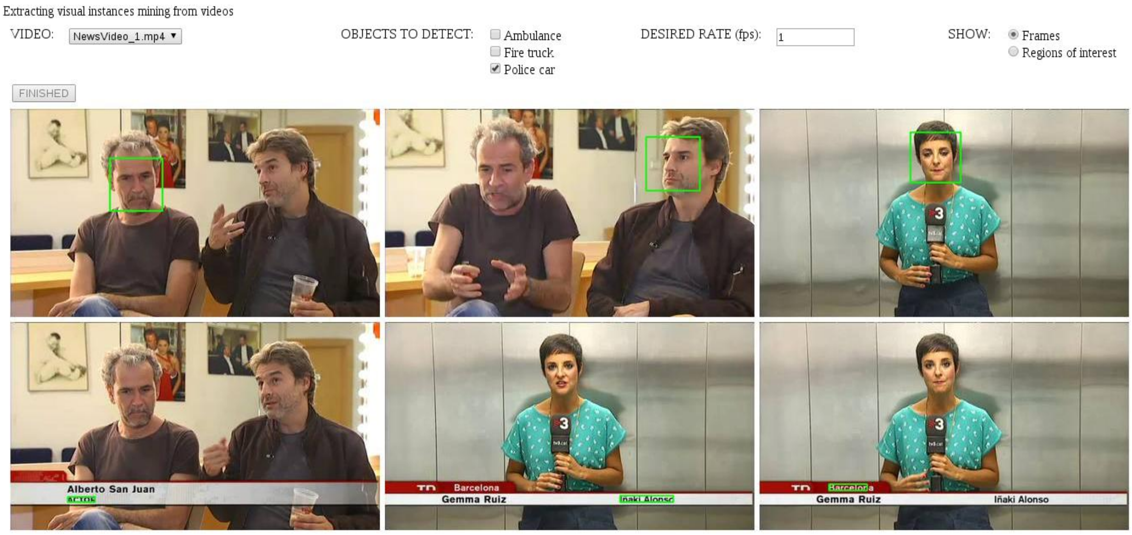
Apart from this widgets we also set a WPushButton. We use this widget to start our application when the user push it. The text of this element will be changed from START to FINISHED when the results are shown.

Fig. 37 The START push button of the GUI

Mixing all these elements in different layouts, we finally got the graphical user interface that is showing Fig. 38. To run this GUI we have to execute the following command line:

```
dalmendros/workspace/imageplus/applications/web/representative_c
ontent_gui/bin/release/representative_content_gui --docroot . --
http-address 0.0.0.0 --http-port 6060
```

If they introduced http://0.0.0.0.6060 in a local browser within the GPI intranet, we could be able to run the tool through the graphical interface.



Extracting visual instances mining from videos

VIDEO: NewsVideo_1.mp4

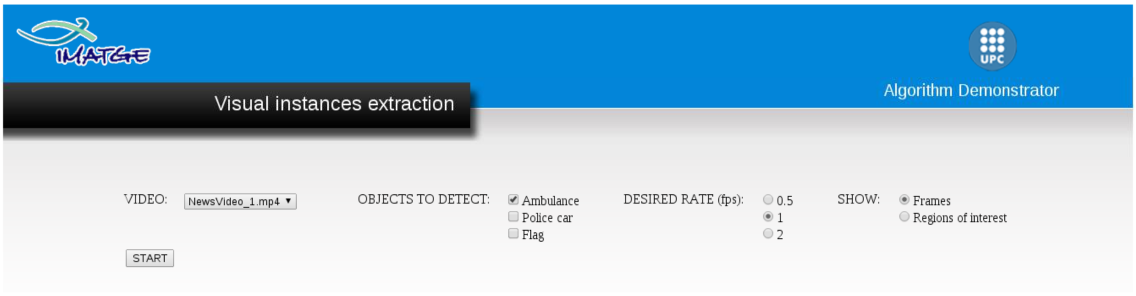
OBJECTS TO DETECT: ☐ Ambulance ☐ Fire truck ☑ Police car

DESIRED RATE (fps): 1

SHOW: ● Frames ○ Regions of interest

Fig. 38 Example of a result in the GUI

In order to improve the attractiveness of the interface, we tried to recover the skin that Marcel created in his Thesis [24]. First of all we set the folder with all elements that compose the skin in our directory. Then, we modified the style.css file that he created because the containers for the content that he created, had some restrictions that modified the layouts of our results. Finally, changing the docroot to the directory where we have all the elements that define the skin, we obtain the GUI that Fig. 39 shows.



Visual instances extraction

Algorithm Demonstrator

VIDEO: NewsVideo_1.mp4

OBJECTS TO DETECT: ☑ Ambulance ☐ Police car ☐ Flag

DESIRED RATE (fps): ○ 0.5 ● 1 ○ 2

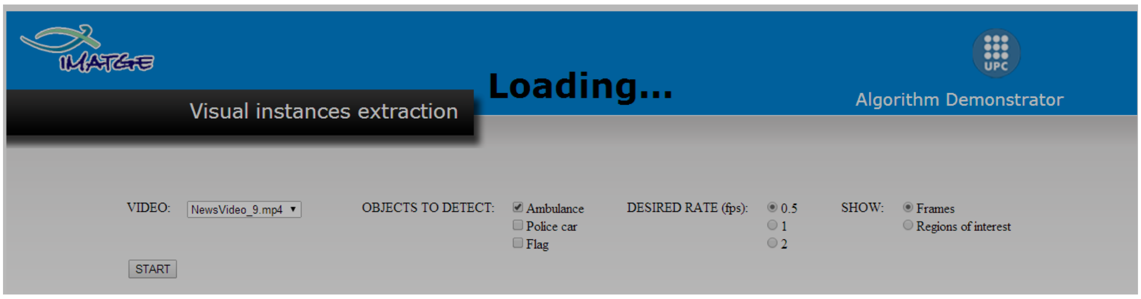
SHOW: ● Frames ○ Regions of interest

Fig. 39 GUI with the skin applied

The last step of this section was to create the online version that could be able to run by everybody who wants to test out tool. We achieved it with the following command line:

```
dalmendros/workspace/imageplus srun -w c2 --mem 4000
/imatge/dalmendros/workspace/imageplus/applications/web/represen
tative_content_gui/bin/release/representative_content_gui --
docroot /imatge/dalmendros/work/wt1 --http-address 0.0.0.0 --
http-port 8081 > /imatge/dalmendros/wt-8081-log 2>&1 &
```

Where we were getting the setups and elements of the skin from /imatge/dalmendros/work/wt1. At the moment of writing this thesis report the URL was http://imatge.upc.edu:8081/ to run the demo. In Fig. 40 we can see the loading icon that the GUI shows while the tool is running and Fig. 41 presents an example of a result that we got from a news video and the parameters that could be seen in the figure.



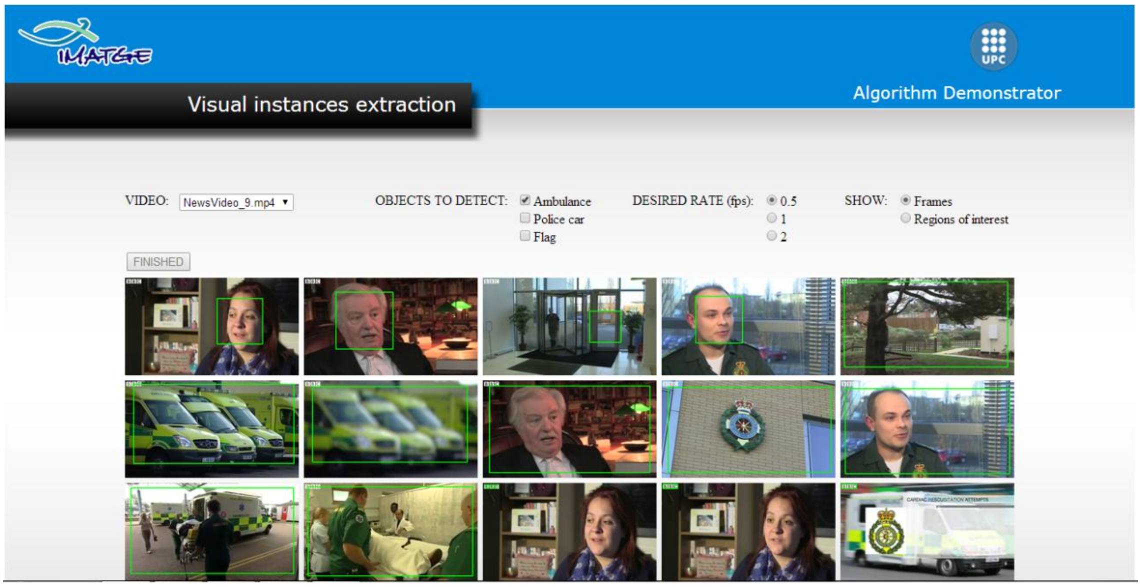
VIDEO: NewsVideo_9.mp4 ▾

OBJECTS TO DETECT: ☑ Ambulance ☐ Police car ☐ Flag

DESIRED RATE (fps): ● 0.5 ○ 1 ○ 2

SHOW: ● Frames ○ Regions of interest

Fig. 40 GUI while running the application



VIDEO: NewsVideo_9.mp4 ▾

OBJECTS TO DETECT: ☑ Ambulance ☐ Police car ☐ Flag

DESIRED RATE (fps): ● 0.5 ○ 1 ○ 2

SHOW: ● Frames ○ Regions of interest

Fig. 41 Result of the tool in the final interface

5. EVALUATION

In this chapter we evaluate our visual instance search mining approach by means of two user studies. With these evaluation we try to verify if the summaries we generate with the application fulfill the requirements we proposed in Chapter 3.

For the first user study we used the CCMA dataset, which is the dataset we used to develop the tool. With this user study we want to know if we have fulfilled the requirements of the Chapter 3. On the other hand, in the second user study we used the dataset which was used in a paper to be published in the prestigious ACM Multimedia conference [25]. With this user study we want to compare our work with a tool based on the state of the art.

In Section 5.1 we comment the method we adopted, the participants, we describe the test material and set-up of the first user study. Finally, the results of the evaluation are discussed. The second user study is explained in Section 5.2.

5.1. User study 1

To evaluate the performance, effectiveness and quality of our proposed visual instances developed, the algorithm needs to be tested. We defined three hypothesis that we wanted to verify in this test.

- **H1.** The visual summary shows the most representative content of the news video. It is shown the most relevant people, the most interesting objects desire for the user and the most important text that appears in the input video.

- **H2.** There is no redundancy in the content of the summary. The information provided by the visual summary resulting is not repetitive.

- **H3.** Users are able to recognize the story of the news clip thanks to the visual summary created with my tool.

In the following section is explained the method used to confirm the hypothesis mentioned before for evaluating the application and the rating method adopted in order to rank the specific hypothesis we have presented.

5.1.1. Method

To evaluate our tool, we decided to apply the same method proposed by Manuel Martos in his thesis. We chose an integer score ranging from 1 (Unacceptable) to 5 (Excellent) which was used by The TRECVID Summarization Evaluation Campaign to rate all summary versions and hypothesis questions [26].

We also evaluated a subjective part, asking questions to the participants to answer in their opinion. This subjective view was also validated as it is explained in the procedure (section 5.1.3).

5.1.2. Test data

For the first survey 4 news videos were selected from CCMA website[14] to evaluate our tool. The following table reports the videos used in the first experiment:

Video id	Topic	Duration (min' sec'')
4330430[15]	International	3' 14''
4229811[16]	Communications	2' 39''
4309291[17]	International	2' 21''
4170250[18]	Science	1' 46''

Table 2 Test data for the first user study

[14] http://www.tv3.cat/videos/ (20/5/2014)
[15] http://www.tv3.cat/videos/4330430 (20/5/2014)
[16] http://www.tve.cat/videos/4229811 (20/5/2014)
[17] http://www.tve.cat/videos/4309291 (20/5/2014)
[18] http://www.tve.cat/videos/4170250 (20/5/2014)

All these videos were tested with the same input arguments: the frame rate was 1 fps and the objects to detect were police cars, fire trucks and ambulances. The four summaries that the tool generated of these news videos are showed in the Appendix II.

5.1.3. Procedure

For this survey two web-based surveys were created with the same structure in Google Drive[19] and the link was given to the participant trough the social networks in order to perform the experiment. Four visual instance mining of four news videos were shown to the participants and they were asked to answer many questions to evaluate the visual summaries created.

We decided to create two surveys because we thought that to verify hypothesis **H3**, participants had to answer the same question before and after watching the video, and probably the second answer could be conditioned by the first one. So in one survey we asked the question before and in the other one we asked it after watching the video. With this method we were able to verify the question before watching the video answered for some participants with the questions answered after watching it for other participants.

The surveys were distributed according to the last digit of the user's mobile phone. Survey 1 was done by participants whose last mobile number is an odd digit and survey 2 was done by participants whose last number is an even digit. We chose this system because we were trying to find a balanced set of users for each of the two surveys.

The tests should satisfy the following constraints:

- First of all the participant have to see the summary.
- (Exclusive of survey 2) Participants have to try to guess the story the news video is talking about.
- Then the news video clip is given to the participants.
- (Exclusive of survey 1) After watching the news video, the participants have to say what is the story related to.
- Next, the rating of the summary should be done.

[19] https://drive.google.com (6/7/2014)

In the first survey the participants had to answer the next questions for the first two summaries generated for the test data in order to evaluate the hypothesis mentioned in the Section 5.1.

After looking the summary but before watching the news video:

- **Q1.1.** Are there redundant information in the summary?

After watching the news video:

- **Q1.2.** The news video is related to...
- **Q1.3.** Rate the effectiveness of the summary.
- **Q1.4.** Arguing the valuation given and explain what was expected for the participant of the application.

With questions **Q1.2** we tried to verify hypothesis **H3**. The question **Q1.1** aimed at testing hypothesis **H2**. Finally questions **Q1.3** and **Q1.4** aimed at testing hypothesis **H1**. Table 2 summarizes all this information.

Questions of the survey	Hypothesis to verify
Q1.1	H2
Q1.2	H3
Q1.3 and Q1.4	H1

Table 3 Hypothesis we try to verify in the first user study with the questions of the survey 1

And the participants of survey 2 had to answer the following questions for the last two summaries generated for the test data.

After looking the summary but before watching the news video:

- **Q2.1.** The news video is related to...
- **Q2.2.** Are there redundant information in the summary?

After watching the news video:

- **Q2.3.** Rate the effectiveness of the summary.

- **Q2.4.** Arguing the valuation given and explain what was expected for the participant of the application.

With questions **Q2.1** we tried to verify hypothesis **H3**. The question **Q2.2** aimed at testing hypothesis **H2**. Finally questions **Q2.3** and **Q2.4** aimed at testing hypothesis **H1**. Table 4 summaries all this information.

Questions of the survey	Hypothesis to verify
Q2.1	H3
Q2.2	H2
Q2.3 and Q2.4	H1

Table 4 Hypothesis we try to verify in the first user study with the questions of the survey 2

The second survey had the same stencil as the first one but the questions for the first two summaries and the other two summaries were the opposite.

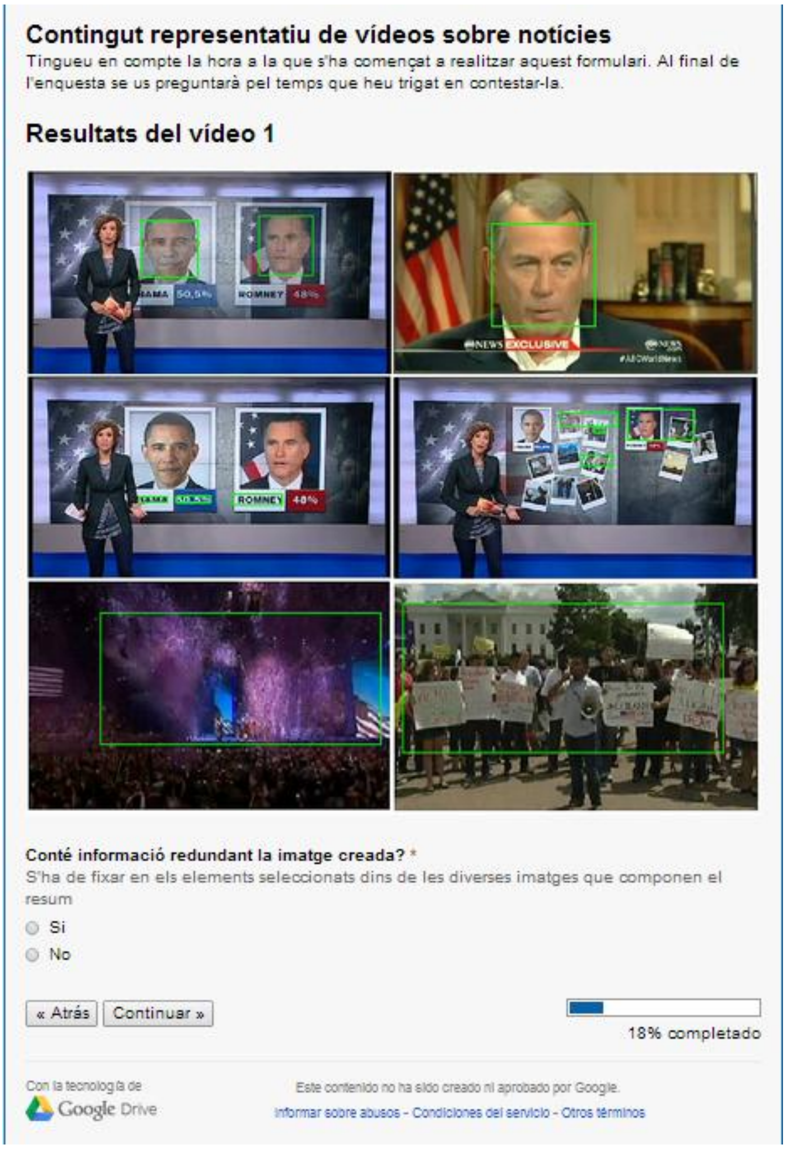
In this way, it was possibe to validate the answer for the question of the survey 1 **Q1.2.** *The news video is related to... (After watching the video)* with the answer of the question **Q2.1.** *The news video is related to... (Before watching the video)* of the survey 2.

At the end of the surveys, the participant was asked for how long it took to do the survey.

The following figures show an example of one summary created in one survey[20] [21] given to the participants to evaluate this summary from a news video:

[20] https://docs.google.com/forms/d/1ExOqmddP2S8zr1XWehR6jMrhHFewVWBH7YDjA6KSYy8/viewform?usp=send_form (23/5/2014)

[21] https://docs.google.com/forms/d/1NkMhZIZilV03epcx22hSFRTcLporcl9lzWGd1lA_LsE/viewform?usp=send_form (23/5/2014)



Contingut representatiu de vídeos sobre notícies

Tingueu en compte la hora a la que s'ha començat a realitzar aquest formulari. Al final de l'enquesta se us preguntarà pel temps que heu trigat en contestar-la.

Resultats del vídeo 1

Conté informació redundant la imatge creada? *

S'ha de fixar en els elements seleccionats dins de les diverses imatges que componen el resum

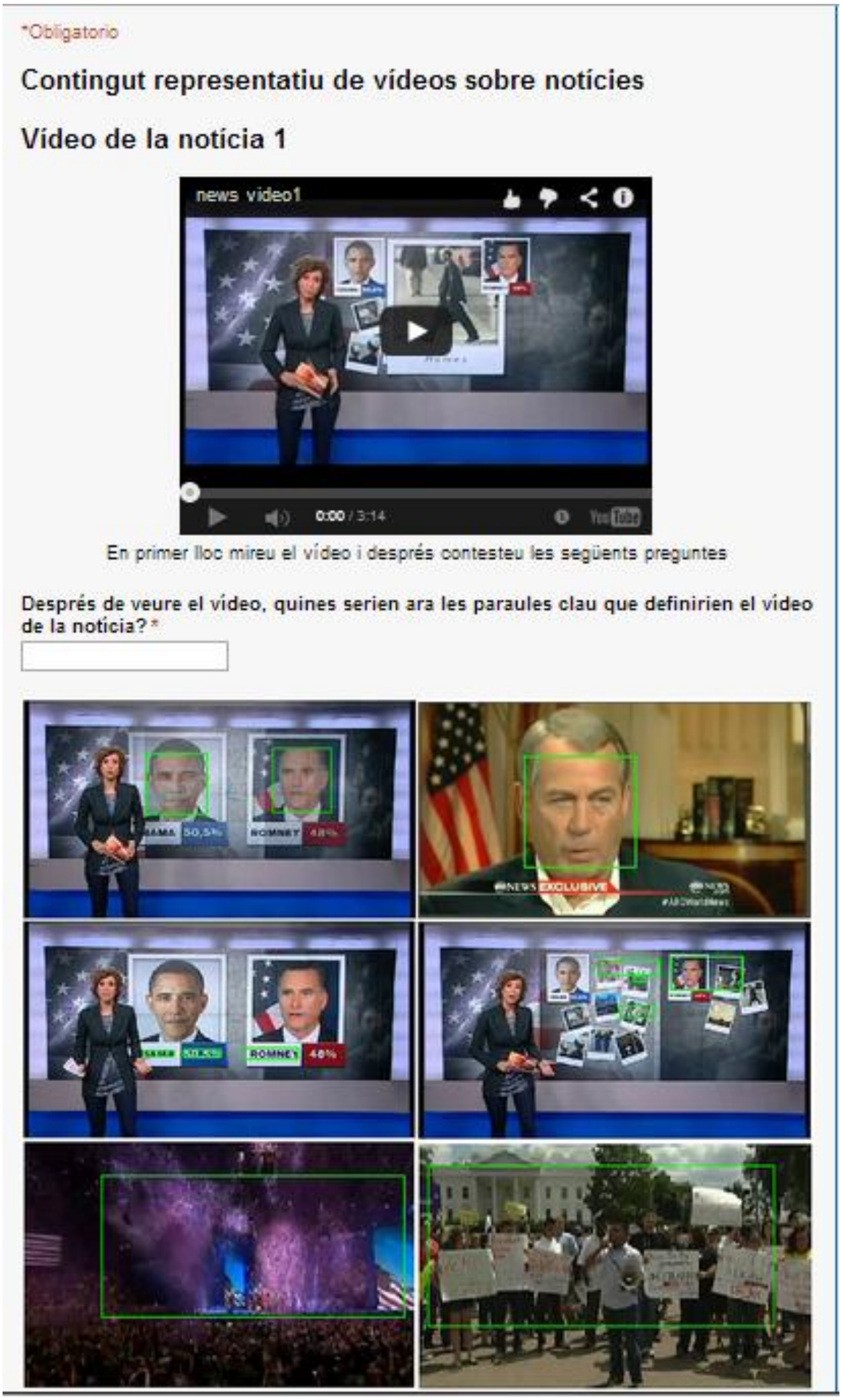
○ Si

○ No

Este contenido no ha sido creado ni aprobado por Google.

Informar sobre abusos - Condiciones del servicio - Otros términos

Fig. 42 First part of the evaluation of one summary



*Obligatorio

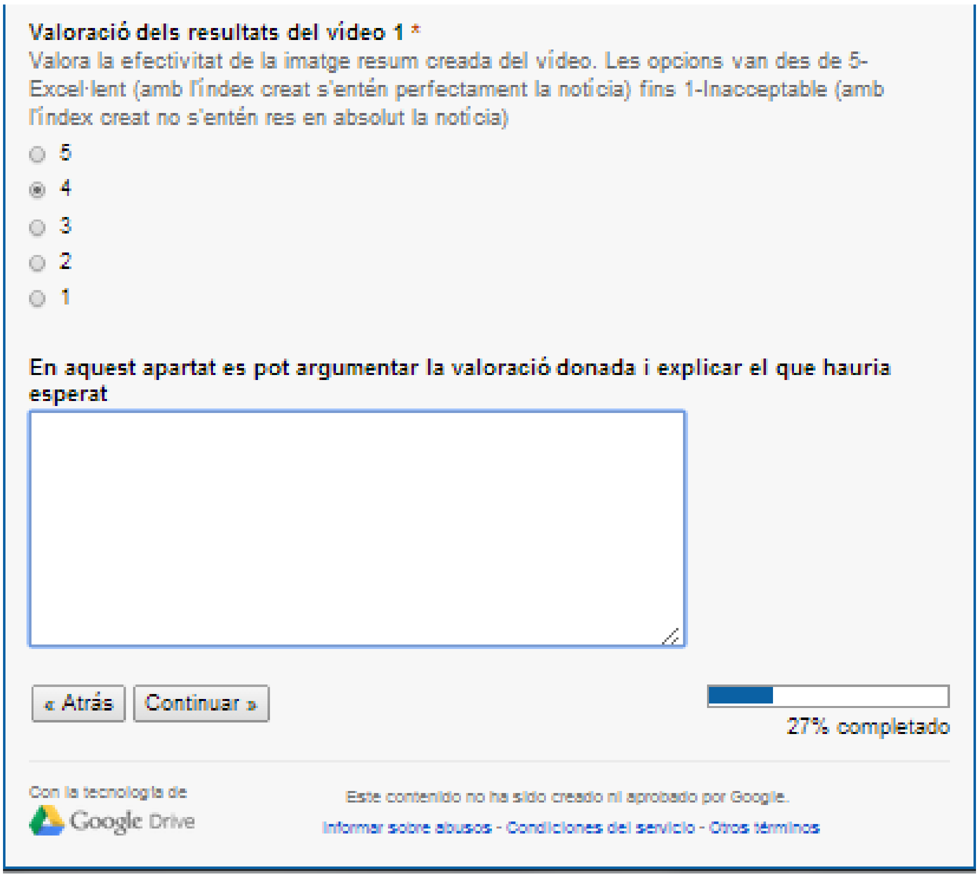
Contingut representatiu de vídeos sobre notícies

Vídeo de la notícia 1

En primer lloc mireu el vídeo i després contesteu les següents preguntes

Després de veure el vídeo, quines serien ara les paraules clau que definirien el vídeo de la notícia? *

Fig. 43 Second part of the evaluation of one summary



Valoració dels resultats del vídeo 1 *

Valora la efectivitat de la imatge resum creada del vídeo. Les opcions van des de 5-Excel·lent (amb l'índex creat s'entén perfectament la notícia) fins 1-Inacceptable (amb l'índex creat no s'entén res en absolut la notícia)

○ 5
● 4
○ 3
○ 2
○ 1

En aquest apartat es pot argumentar la valoració donada i explicar el que hauria esperat

27% completado

« Atrás
Continuar »
Con la tecnología de
Google Drive
Este contenido no ha sido creado ni aprobado por Google.
Informar sobre abusos - Condiciones del servicio - Otros términos

Fig. 44 Last part of the evaluation of one summary

5.1.4 Participants

In order to achieve a great number of people who perform the survey, we opened it to the social networks as currently it is the best way to spread any news to a wide range of people.

My advisor Xavier Giró also published the survey through his social networks and spread the survey in the Image Processing Group in the UPC.

For the first experiment, as the test data we used to create the evaluation are videos from the CCMA database, a television network from Catalonia, the survey was prepared in Catalan. This means that all people who wanted to participate in this evaluation part of the Thesis must understand the language.

A total of 40 users answered this online survey we prepared for evaluating the solution developed; 20 participants did the survey number 1 and 20 participant did survey number 2.

5.1.5. Experimental results

After obtaining all the results, the test survey latest on average 11min 08sec with a maximum of 38 minutes and a minimum of 8 minutes.

As it is shown in Fig. 45, the test was done by 23 men (57%) and 17 women (43%) and the majority of the people between 18 and 39 years old.

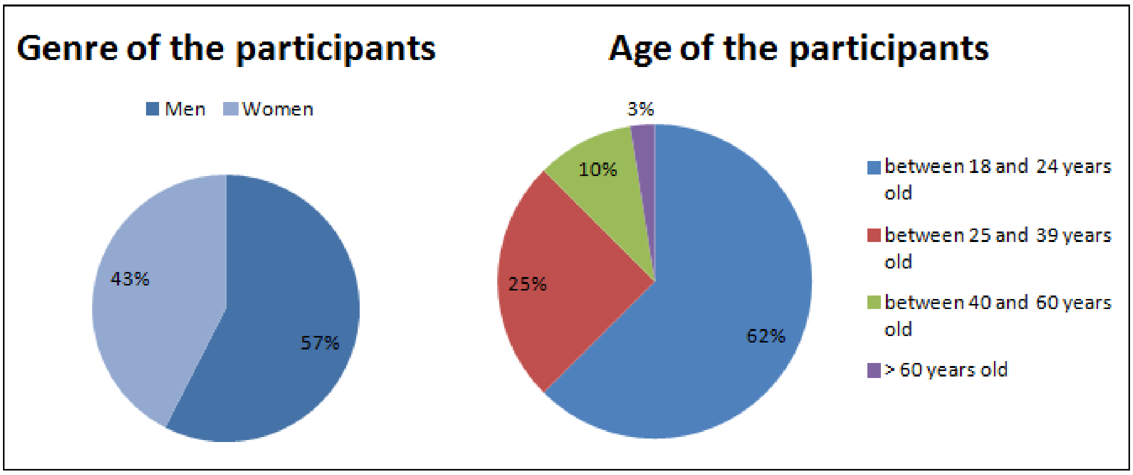


Fig. 45 Statistics of genre and age of the participants

According to the score that the participants gave to our results in question 3 of both surveys, measured the Mean Opinion Score (MOS) is measured. We get that the global evaluation of our system is 4.09 as it is shown in the following figure. With this score, is considered that the results of our system are quite good. Fig. 46 shows the global analysis of the ratings.

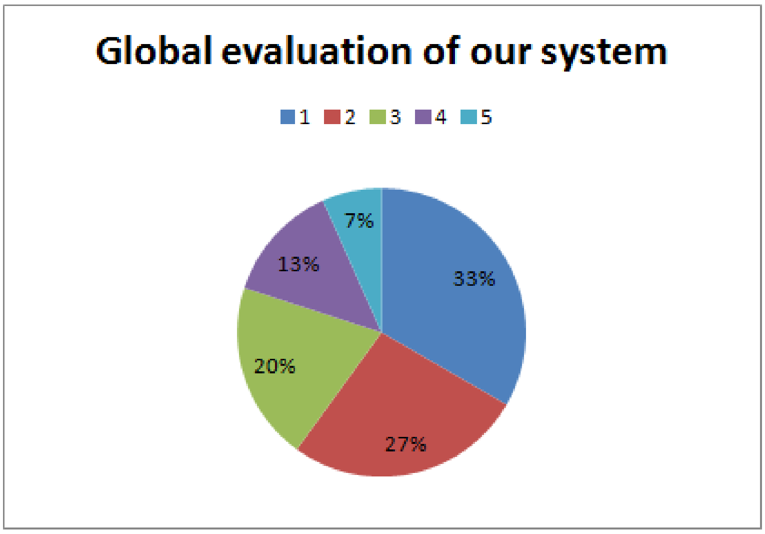


Fig. 46 Global rating of our system

We also add all 4 summaries generated and the results of the survey for each one of them.

The summaries we used for the survey are in the Appendix II. First of all we analyze the question about if it exist redundancy in the visual summary. The results are shown in the following figures.

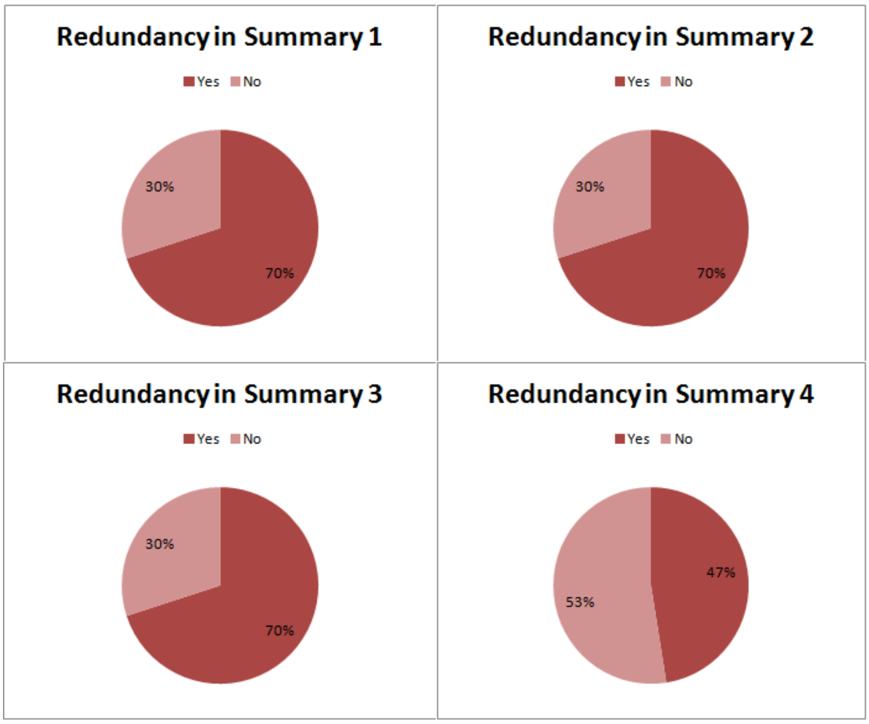


Fig. 47 Results of redundancy in the summaries

The results show that in the first three summaries 70% of the participants think that there is redundancy in the summary. Nevertheless, in the last summary more than the 50% of the participants answered that there is no redundancy in this summary. Analyzing these results, we have to say that the hypothesis H3 brought up in this Chapter which was a structural requirement (see Chapter 3) was not successfully fulfilled. However the visual summary does not seem to have repetitive content. Participants said that there were two similar images that compose the summary although one of them focused on a content and the other image focused on another content; then they assumed that it existed redundancy in the summary.

Next, we present the tables with keywords. On one hand there are the words that half of the participants consider that were the keywords which could define the story of the news without watching the video. On the other hand there are the words that the other half of participants consider that were the keywords that defined the storyline of the news after watching the video. The words that appear in these tables are the top 5 most mentioned keywords by the participants in our survey.

Ranking	Keywords before watching the video	Keywords after watching the video
1	**Politics**	USA
2	**Election**	Politics
3	**Protest**	Society
4	**Obama**	Democrats
5	**USA**	Republicans

Table 5 Keywords of the news story 1

Ranking	Keywords before watching the video	Keywords after watching the video
1	**Music**	New schedule
2	**Catalunya Radio**	Novelty
3	**Programming**	Catalunya Radio
4	**Office**	Culture
5	**Schedule**	Information

Table 6 Keywords of the news story 2

Ranking	Keywords before watching the video	Keywords after watching the video
1	**Puerto Rico**	Independence
2	**Independence**	Puerto Rico
3	**Political party**	Future
4	**Election**	Voting
5	**Opinion**	Political party

Table 7 Keywords of the news story 3

Ranking	Keywords before watching the video	Keywords after watching the video
1	**Fruit**	New technology
2	**Industrial production**	Fruits juice
3	**Pineapple**	Industrial production
4	**Fresh**	Conservation
5	**Consumption**	Process

Table 8 Keywords of the news story 4

Analyzing the results we come to conclusion that the visual summary is a good reflection of the story of the news video. In all the summaries at least two words that participants wrote before and after watching the video were the same. Furthermore, all the keywords that participants wrote before watching the video were related to the video.

The last parameter to be shown is the score ratio that the participants gave to each summary in order to evaluate the efficiency of our system. In Fig. 48 we can see these ratios.

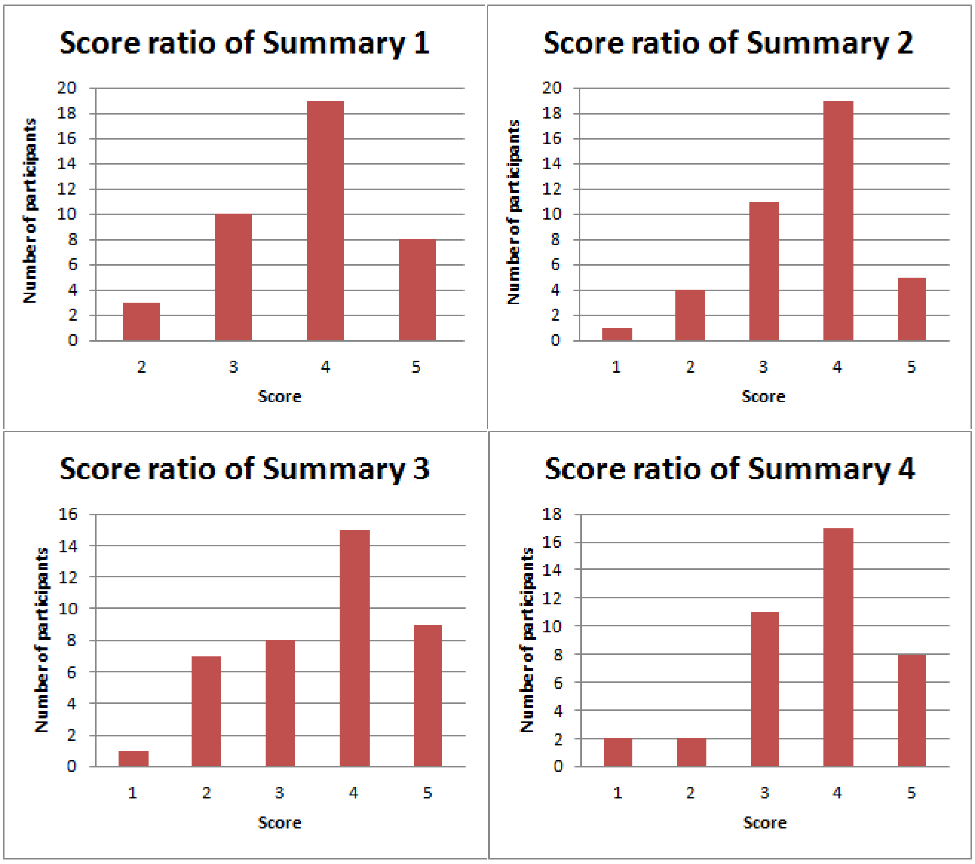


Fig. 48 Rate of the summaries generated

We can work out that in general, the score that the participants gave to our system is acceptable. The majority of the participants in our survey rated our visual summaries with three or four.

5.2. User study 2

With this user study we wanted to compare our visual instance mining approach with the state of the art.

In the following section we explain the method and procedure. We also describe the test data and finally we conclude with the results of this survey.

5.2.1. Method

As in the first survey, we decided to apply the same method Manuel proposed in his thesis [2]. We chose an integer score ranging from 1 (Unacceptable) to 5 (Excellent) which was used by The TRECVID Summarization Evaluation Campaign to rate all summary versions and hypothesis questions [26].

5.2.2. Test data

For this second survey we used a collection of videos from different American TV broadcast that the Columbia University kindly shared with us.

News	Number of videos
Boston Marathon bombings	356
Disappearance of the Malaysia airlines flight	406

Table 9 Test data of the second user study

In order to create our visual summaries, we searched in Google Images 'Boston bombing' and 'Malaysia airlines flight 370' and we downloaded the top 20 results for each search in order to use these images as a 'model' for the objects. Then we needed to process the collection of videos in order to extract visual instances. As there were more than 750 videos in total, it could take a lot of time to process them one by one. So to do the processing of these amounts of videos we asked the GPI group for advice about which possibilities we had to carry out this task. After commenting the possibilities we decided to use the Job arrays technique, developed by Josep Pujal and explained in Appendix IV. We got these results for each news video as it is also explained in Appendix IV.

Once we had the collection of visual instances from all datasets, we wanted to generate a visual summary with these instances. We thought of taking advantage of the function that we created for the clustering and selection. There was a difference between this situation and when we only processed a video. When we run a video, the clusters for faces, objects and text are threaded separately. In this situation we cluster all visual instances together. Apart from the images, the function requires other parameters as a threshold (if the distance between the images is lower than this threshold, the link of the graph between them is removed and this way, the subgraphs are created) and a minimum number of images in the subgraph to extract the resultant image with the highest score.

As we had all content in the same cluster, we had to adjust the two parameters mentioned before. If we established a high threshold, we would obtain objects and text in the visual summary. If we needed that some people appear in the final result, we needed to lower the value accordingly, without modifying too much of the other content. After several empirical tests, we finally used a threshold value of 1e^{-17}.

The other parameter happened something similar. Due to the diversity of the duration of the video collection (some were 1 minute videos and other 9 minutes), this number could vary depending on the video. We have carried out some empirical testing with the following values: 3, 4, 5 and 6 and the best results were obtained with 4.

This way we extracted the most representative and most relevant instances of the images collection and we could create the visual summary. In Appendix II we can see the two summaries that our tool generated of these news.

5.2.3. Participants

For this second survey, all datasets were in English. All videos we processed from the news of the Boston Marathon bombings and the disappearance of the Malaysia airlines flight came from American TV broadcast.

As in the first user test, my advisor Xavier Giró published the survey through his social networks and spread the survey in the Image Processing Group in the UPC. For this evaluation, as it was prepared in English, my advisor Horst Eidenberger (TU Wien) and Wei Zhang from the Columbia University (New York) also sent emails to their fellow researchers. A total of 55 users answered this second web-based survey.

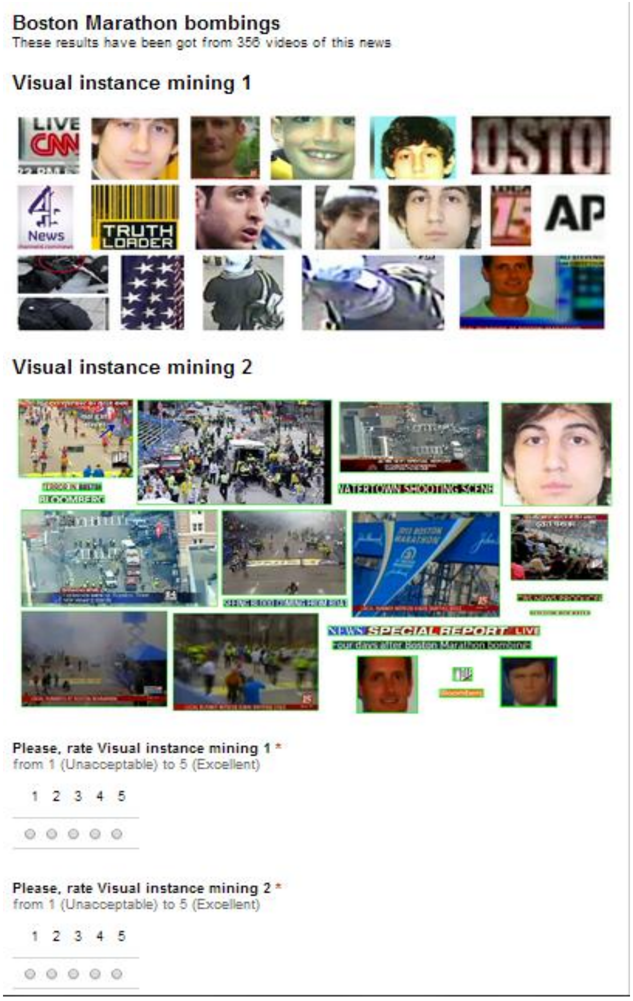
5.2.4. Procedure

We created a web-based survey in Google Drive and the link was given to the participant. A total of four visual summaries were shown to the participants and they were asked for rating these summaries to evaluate our approach.

Two of these summaries belong to the well-known news of the Boston Marathon bombings and the other two represent another news of this years, the disappearance of the Malaysia airlines flight. We compared our results with the ones obtained by Wei Zhang in [25].

This survey was quite simple. We just showed the two visual summaries that belong to the same news and we asked the participants to rate both of them according to their opinion. The procedure was the same for the two summaries of the other news. Just looking at the answer of the users, we are able to compare our visual summary with the state of the art and we have a reference if our results are better, similar or worse.

In the following Fig. 49 we can see an example of one page of the survey given to the participants to evaluate our visual summary from a news video.



Boston Marathon bombings

These results have been got from 356 videos of this news

Visual instance mining 1

Visual instance mining 2

Please, rate Visual instance mining 1 *

from 1 (Unacceptable) to 5 (Excellent)

1 2 3 4 5

○ ○ ○ ○ ○

Please, rate Visual instance mining 2 *

from 1 (Unacceptable) to 5 (Excellent)

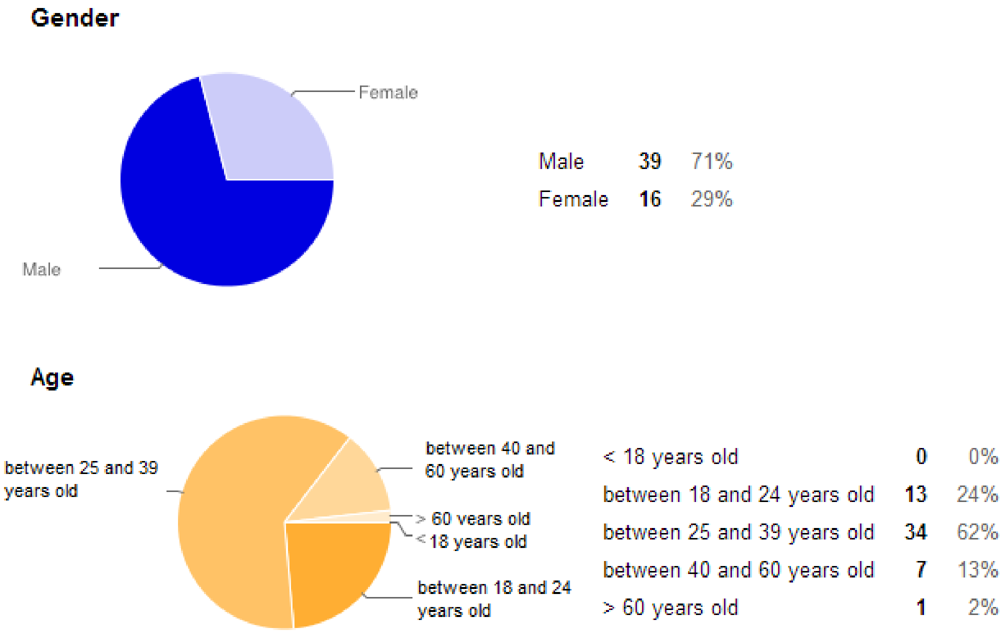
1 2 3 4 5

○ ○ ○ ○ ○

Fig. 49 Part of the second user study survey

5.2.5. Experimental results

As it has been commented before, a total of 55 participants answered this survey. In the following Fig. 50 we show the percentage of men and women that participated in the study. Clearly we can see a majority of men with 71% of the answers. In the same figure, it is also shown the age of the participants, where we can conclude that people from 18 to 39 years make up almost the totality of participants.

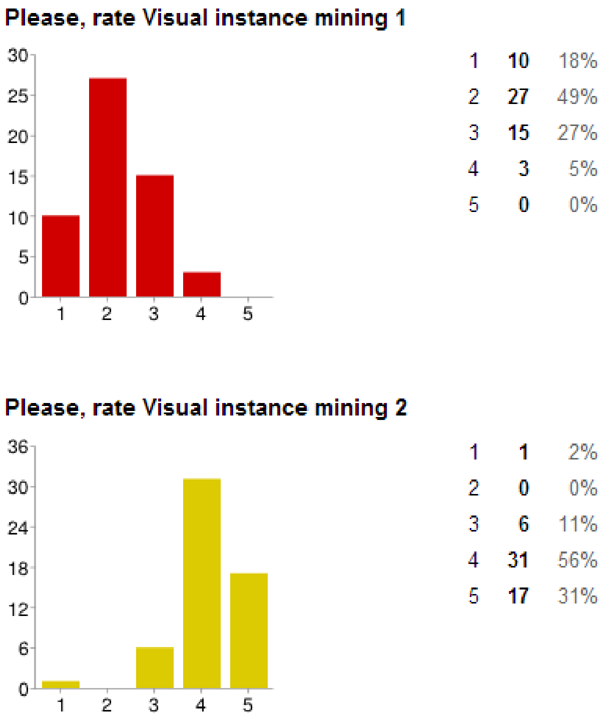


Male	39	71%
Female	16	29%

< 18 years old	0	0%
between 18 and 24 years old	13	24%
between 25 and 39 years old	34	62%
between 40 and 60 years old	7	13%
> 60 years old	1	2%

Fig. 50 Statistics of genre and age of the participants

To evaluate all visual summaries, we used the same parameter as Manuel used in his thesis, the Means Opinion Score (MOS) test, which is measured by averaging the ratings given by the users in this survey. Fig. 51 shows the global ratings of the summaries that we are comparing for the 'Boston Marathon bombings' news. The results obtained by Wei Zhang in his paper for ACM Multimedia 2014 for this news obtain a MOS = 2.2 while the result that we obtain for this news using our approach achieves a MOS = 4.15, which performs much better than the state of the art.



Please, rate Visual instance mining 1

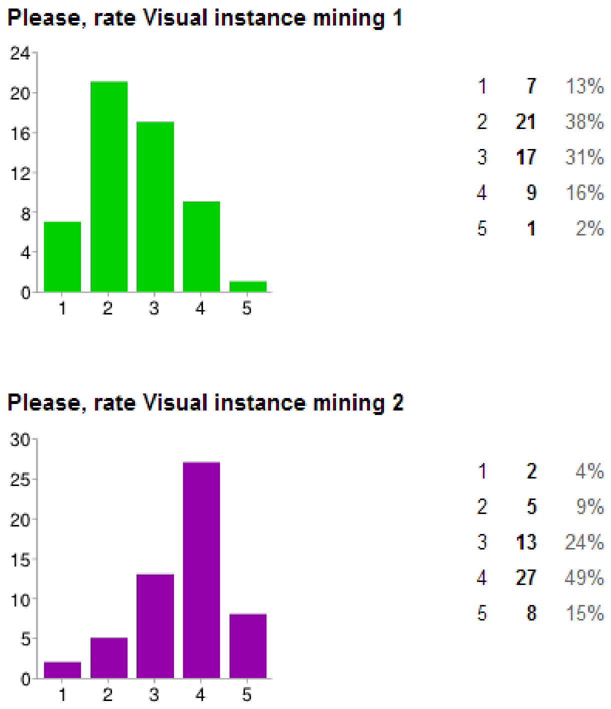
1	10	18%
2	27	49%
3	15	27%
4	3	5%
5	0	0%

Please, rate Visual instance mining 2

1	1	2%
2	0	0%
3	6	11%
4	31	56%
5	17	31%

Fig. 51 Results of the rates of the visual summaries related to the Boston bombing

For the 'Malaysia airlines flight disappearance' news we compared again the result of Wei Zhang against the result obtained applying our method. Fig. 52 shows the global ratings of the summaries that we were comparing for this news. On one hand, Wei Zhang obtained a MOS = 2.56, which is fairly better than the score he obtained in the previous task. On the other hand, the result that we obtained for this news using our approach achieved a MOS = 3.62, which performs significant better than the state of the art. However, it is a worse score than the one we got for the other news item.


Please, rate Visual instance mining 1

1	7	13%
2	21	38%
3	17	31%
4	9	16%
5	1	2%

Please, rate Visual instance mining 2

1	2	4%
2	5	9%
3	13	24%
4	27	49%
5	8	15%

Fig. 52 Results of the rates of the visual summaries related to the Malaysia airlines flight disappearance

At this point, we state that the visual summaries we create with the visual instance mining are considered a good summary of the news video and seem competitive with the state of the art.

We notice though that Wei Zhang's results did not use any external source of data as we did, so it would not be fair to compare our results with theirs because we exploited the top images retrieved on Google with the textual queries.

6. CONCLUSIONS

This Thesis has been developed as a codirected project by Technische Universität Wien (TU Wien) and Universitat Politècnica de Catalunya (UPC). In this project, it is presented a solution to video summarization with visual instances mining that shows the most important people, the most representative content and relevant text from a news video, analyzing the video and helping users to understand the news video content item in a fast and visual way.

We developed a tool that visual instances mining from an input video, showing the most important faces, objects and texts, being this results an effective visual summary of the news. The application is programmed in C++ in the ImagePlus platform and it is basically based on OpenCV libraries and functions and algorithms that the UPC Image Processing Group has developed.

We can easily split the developed tool in three main steps: the detection, the clustering and the selection. Once we have the uniform sampling of the input video, the detection block relate to detects all faces, text and the desired objects that appears in each frame. The clustering relates to recognize all detected content and cluster them in groups depending on a level of similarity. Then it is time for the selection block that apply a criteria to select which are the most important clusters and shows the most representative content of these clusters.

The results of the solution developed were evaluated with two user studies, which were created in Google Drive and shared by e-mail and published in different social networks. The results have shown that our approach is able to properly extract visual instances that shows the most relevant content of a news video and can represent an effective summary of the news that the video presents. Our application can be used for summarizing news in a visual way. In participants opinion, comparing the state of the art with our results, our visual summaries are comparable with the results obtained with the other approach. The computational effort to create the object maps is mainly related to the used object detection technique, this means that, the more number of objects the user want to detect, the more time the algorithm will take to show the results. Furthermore, the computational time is also related to the desired frame rate that the user want to process.

7. FUTURE WORK

In this Chapter we propose four different directions of research along which the presented work could be taken further:

- Improve the detection.

  As we have developed the system in blocks, it should be easy to replace the detection algorithm with other methods more effective, for example for object detection. The SURF detection developed in this Thesis is not the best tool to detect objects. Integrating another object detection to the detection block could be an option. The goal would be to improve the detection of the user's desired objects, because the accuracy of the designed algorithm is quite low. Other option could be that users do not have to train images of the objects they want to detect, the own algorithm should decide which objects are the most relevant of the video. This could be achieved with a generic object detector, such as [27][28].

- Audio transcription

  In order to improve and enrich our visual instance mining, it can be implemented in the detection block an audio transcription. With this, users could be able to visualize in the summary if a speech or some speeches are repetitive during the news. It could also detect sounds such as explosions or yells. This information could be very useful for the users to recognize what is the storyline of the news about.

- Content presentation.

  It could be created a tile-based image with the different visual instances that contains the representative content selected. This could make our visual summaries more attractive. With different sizes depending on how important is the instance. The compositing stage of a summary could be an important block to be introduced to the architecture. A good representation can make the difference between good and bad summaries. The amount of source frame pixels represented in the summary is a variable that has to be taken into account if we want to create rich visual summaries.

- Interactive prototype

  An interactive visual summary could be developed. It can solve video navigation problems. If this solution is implemented, the tool could achieve another application. With the visual summary, users could use the visual instances as a visual index that will allow a fast access to the exact shot where each instance was extracted by a simple click on the instance.

BIBLIOGRAPHY

[1]. Soukup, T., & Davidson, I. (2002). *Visual data mining: Techniques and tools for data visualization and mining*. John Wiley & Sons.

[2]. Martos M. (2013, May). Content-based Video Summarisation to Object Maps.

[3]. Li, H., Jou, B., Ellis, J. G., Morozoff, D., & Chang, S. F. (2013, October). News rover: exploring topical structures and serendipity in heterogeneous multimedia news. In *Proceedings of the 21st ACM international conference on Multimedia*(pp. 449-450). ACM.

[4]. Jou, B., Li, H., Ellis, J. G., Morozoff-Abegauz, D., & Chang, S. F. (2013, October). Structured exploration of who, what, when, and where in heterogeneous multimedia news sources. In *Proceedings of the 21st ACM international conference on Multimedia* (pp. 357-360). ACM.

[5]. Satoh, S. I., Nakamura, Y., & Kanade, T. (1999). Name-it: Naming and detecting faces in news videos. *IEEE Multimedia*, *6*(1), 22-35.

[6]. Sivic, Josef, and Andrew Zisserman. "Video data mining using configurations of viewpoint invariant regions." *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2004.

[7]. Philbin, James, and Andrew Zisserman. "Object mining using a matching graph on very large image collections." *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*. IEEE, 2008.

[8]. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, *1*(C), I–511–I–518. doi:10.1109/CVPR.2001.990517

[9]. Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. *Proceedings International Conference on Image Processing*, *1*(2002), I–900–I–903. doi:10.1109/ICIP.2002.1038171

[10]. Vilaplana Besler, V. Region-based face detection, segmentation and tracking. framework definition and application to other objects.

[11]. Giró, X., Vilaplana, V. & Marqués, F. Image Based Biometric Face Recognition. *Biometric Systems*

[12]. Zhang, X., Yang, Y. H., Han, Z., Wang, H., & Gao, C. (2013). Object class detection: A survey. *ACM Computing Surveys (CSUR)*, *46*(1), 10.

[13]. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

[14]. Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359.

[15]. Jégou, H., Douze, M., & Schmid, C. (2010). Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, *87*(3), 316-336.

[16]. Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010, June). Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on* (pp. 2241-2248). IEEE.

[17]. Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2963-2970). IEEE.

[18]. Gao, X., & Tang, X. (2000). Automatic news video caption extraction and recognition. In *Intelligent Data Engineering and Automated Learning—IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents* (pp. 425-430). Springer Berlin Heidelberg.

[19]. Gimferrer A. (2012). Text detection and recognition using stroke width stimation.

[20]. Ventura, C., Giro-i-Nieto, X., Vilaplana, V., Giribet, D., & Carasusan, E. (2013, June). Automatic keyframe selection based on mutual reinforcement algorithm. In *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on* (pp. 29-34). IEEE.

[21]. Jing, Y., & Baluja, S. (2008). Visualrank: Applying pagerank to large-scale image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*(11), 1877-1890.

[22]. Richter, F., Romberg, S., Hörster, E., & Lienhart, R. (2010, March). Multimodal ranking for image search on community databases. In *Proceedings of the international conference on Multimedia information retrieval* (pp. 63-72). ACM.

[23]. Joshi, D., Wang, J. Z., & Li, J. (2006). The Story Picturing Engine: a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, *2*(1), 68-89.

[24]. Tella M. (2011 May). Interactive Image Processing demonstrations for the web.

[25]. Zhang W. Scalable Visual Instance Mining with Threads of Features ToF : Thread of Features. ACM Multimedia 2014.

[26]. Over, P., Smeaton, A. F., & Kelly, P. (2007, September). The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proceedings of the international workshop on TRECVID video summarization* (pp. 1-15). ACM.

[27]. Cheng, M. M., Zhang, Z., Lin, W. Y., & Torr, P. (2014). BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*.

[28]. Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F., & Malik, J. (2014). Multiscale Combinatorial Grouping. CVPR.

APPENDIX I. Working plan

As this Thesis is an extension and an adaptation of Manuel's work and all this previous work was developed in Java language, we started working with Java, trying to find other methods that can improve his approach. Our first task was looking for the state of the art and search for the techniques that developers use in their projects, reading papers and try to think if they are useful for our approach and how we could develop them. Obviously we focused the interest on all papers related to the news domain. Then we started working with the shot detection, trying to find an appropriate method that could be applied in my solution developed.

We had problem compiling and linking programs we had found that could be suitable for the Thesis. cMake[22] and MinGw[23] were used to compile the source codes but some questions were set out. After having had two meetings and suggested other options, we finally agreed to change completely with the programming language and start developing in C++ in ImagePlus, the platform that people of the Image Processing Group from the UPC use to develop their tools. With this decision, people from the Group could help me with my development.

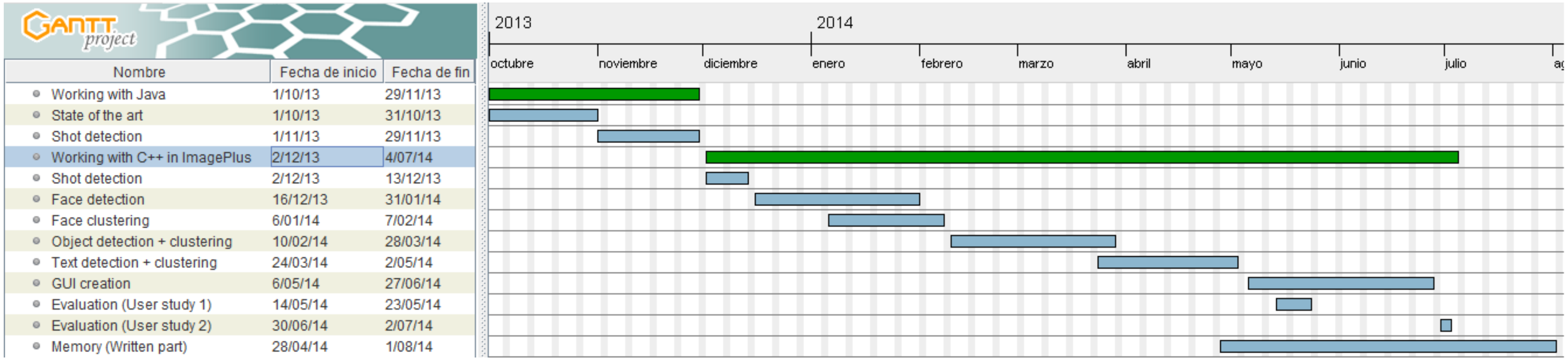
After a short period of time when I got used to the new environment, I started again working with the different blocks of the system architecture we had designed.

First of all, trying to implement a shot detector. Then we focus on the detection and clustering of the different content. Looking for which is the state of the art and trying to develop them in our system. Finally we decide a criteria to select the representative content of the clusters.

I started working on the report while I was still developing the rest of the system. When I had finished the whole system, an evaluation of the approach done creating two user surveys.

[22] http://www.cmake.org/ (6/3/2014)
[23] http://www.mingw.org/ (6/3/2014)

In Fig. 53 we can see a Gantt diagram[24] of how I carried out the work during the months. I distributed my time according to the different blocks I had to develop. I worked an average of 25 hours a week.



Nombre	Fecha de inicio	Fecha de fin
Working with Java	1/10/13	29/11/13
State of the art	1/10/13	31/10/13
Shot detection	1/11/13	29/11/13
Working with C++ in ImagePlus	2/12/13	4/07/14
Shot detection	2/12/13	13/12/13
Face detection	16/12/13	31/01/14
Face clustering	6/01/14	7/02/14
Object detection + clustering	10/02/14	28/03/14
Text detection + clustering	24/03/14	2/05/14
GUI creation	6/05/14	27/06/14
Evaluation (User study 1)	14/05/14	23/05/14
Evaluation (User study 2)	30/06/14	2/07/14
Memory (Written part)	28/04/14	1/08/14

Fig. 53 Gantt diagram that shows the time planning of the Thesis

[24] http://www.ganttproject.biz/ (11/5/2014)

APPENDIX II. Test data

First of all, we show in this Appendix the summaries generated for the survey to evaluate the system we developed.

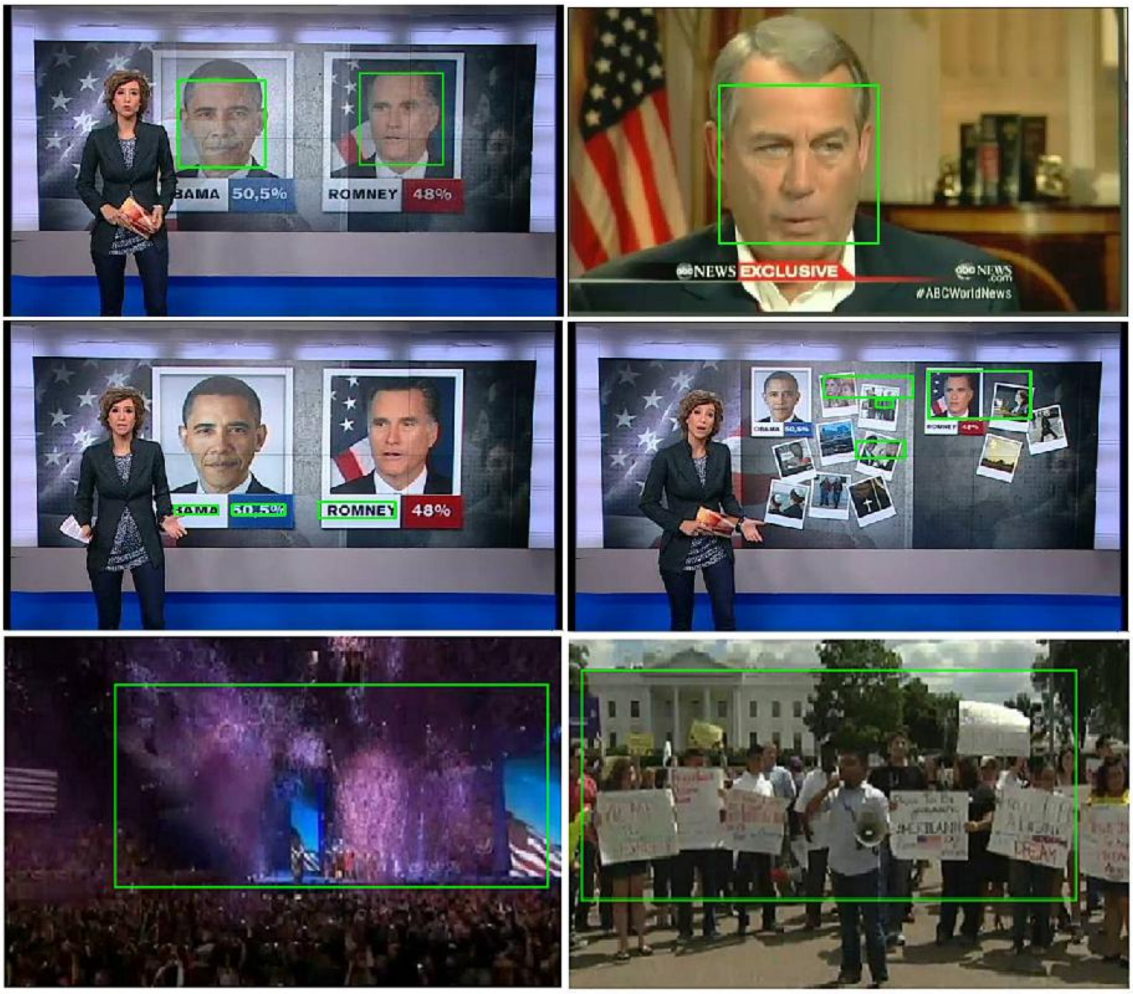


Fig. 54 Visual summary of the news video 1 of the first user study

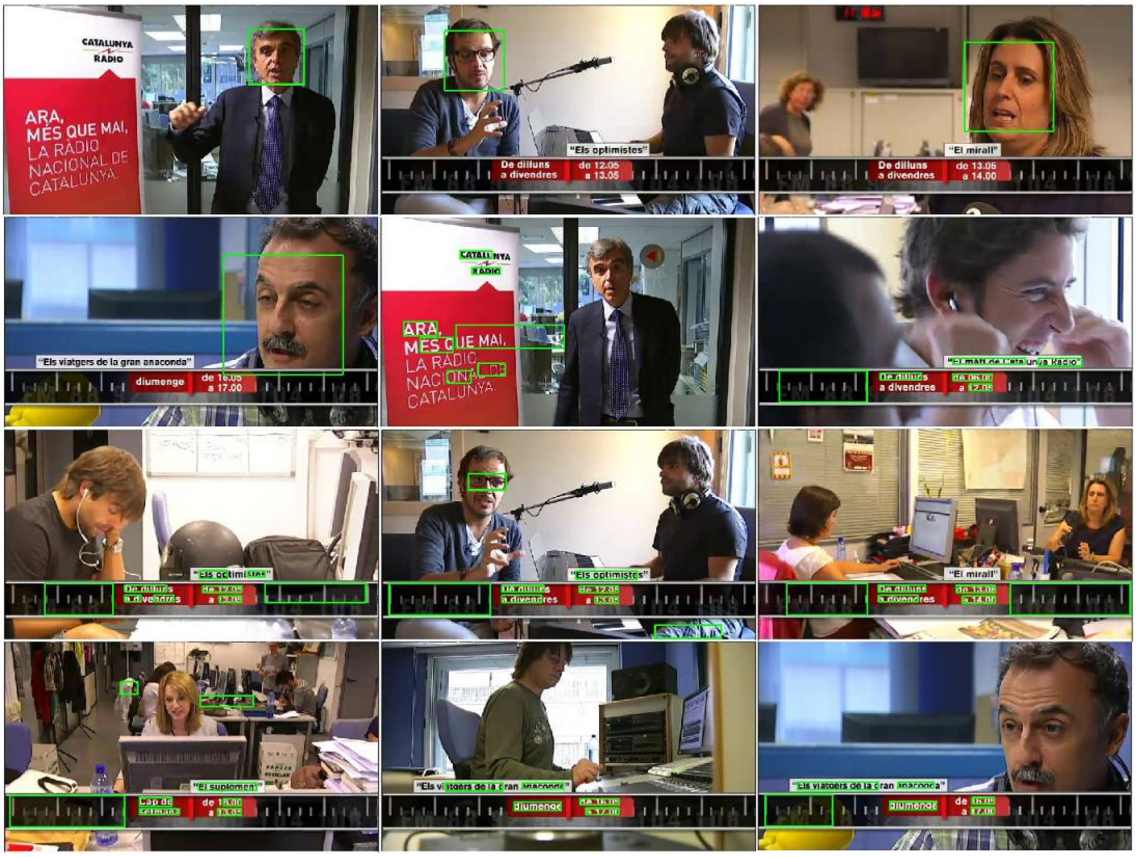


Fig. 55 Visual summary of the news video 2 of the first user study

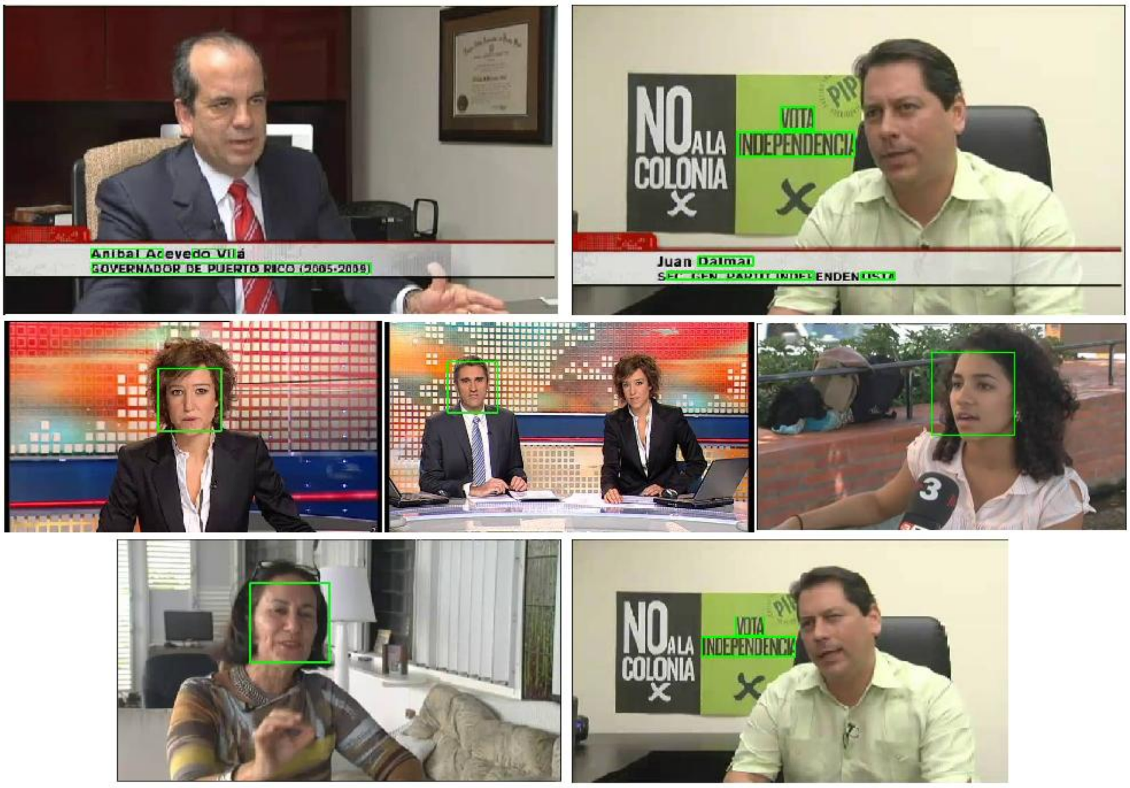


Fig. 56 Visual summary of the news video 3 of the first user study

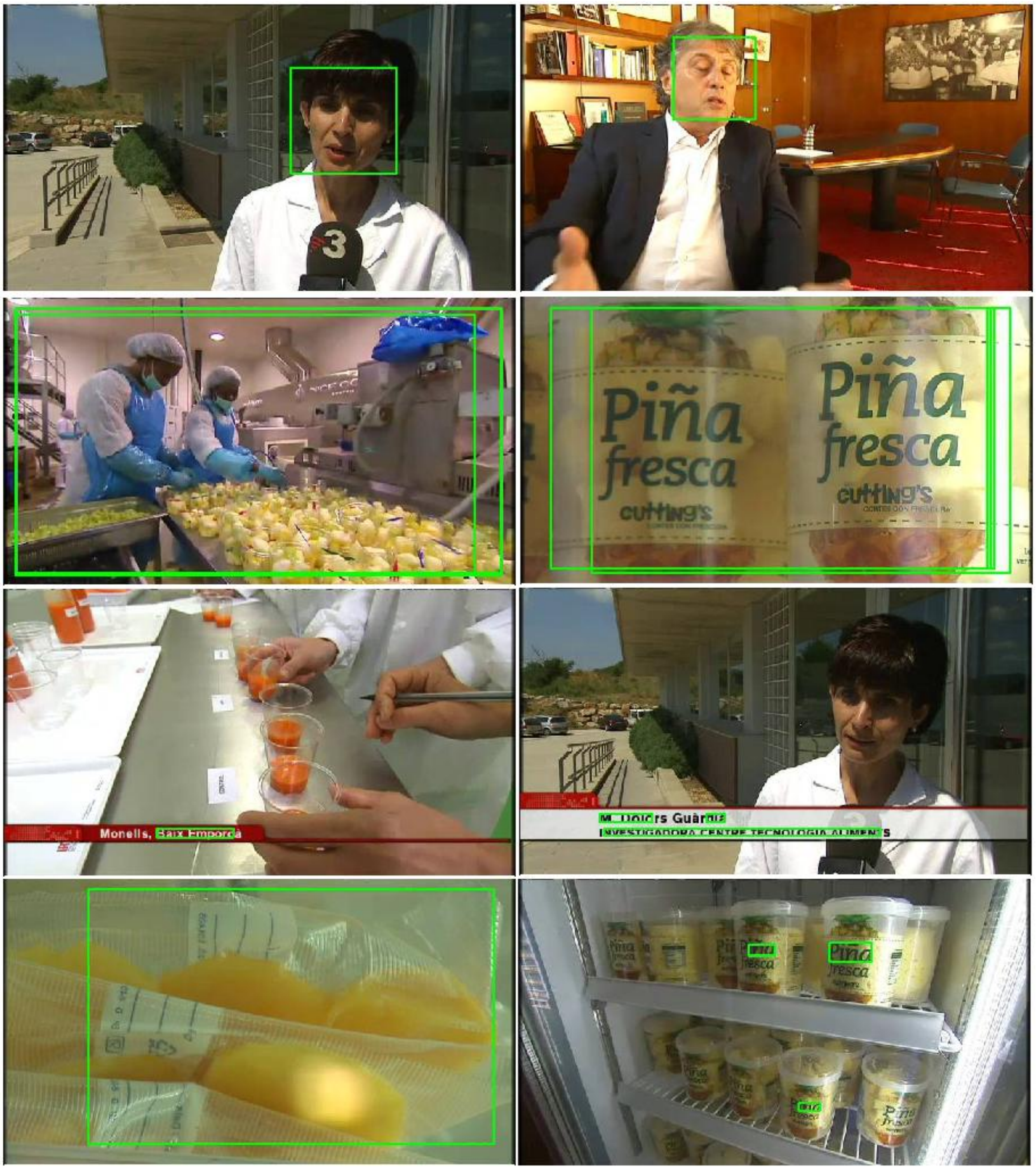


Fig. 57 Visual summary of the news video 3 of the first user study

Furthermore, we show the summaries we get from the news of the Boston Marathon bombing and the disappearance of the Malaysia airlines flight to compare with a project developed at Columbia University (New York).

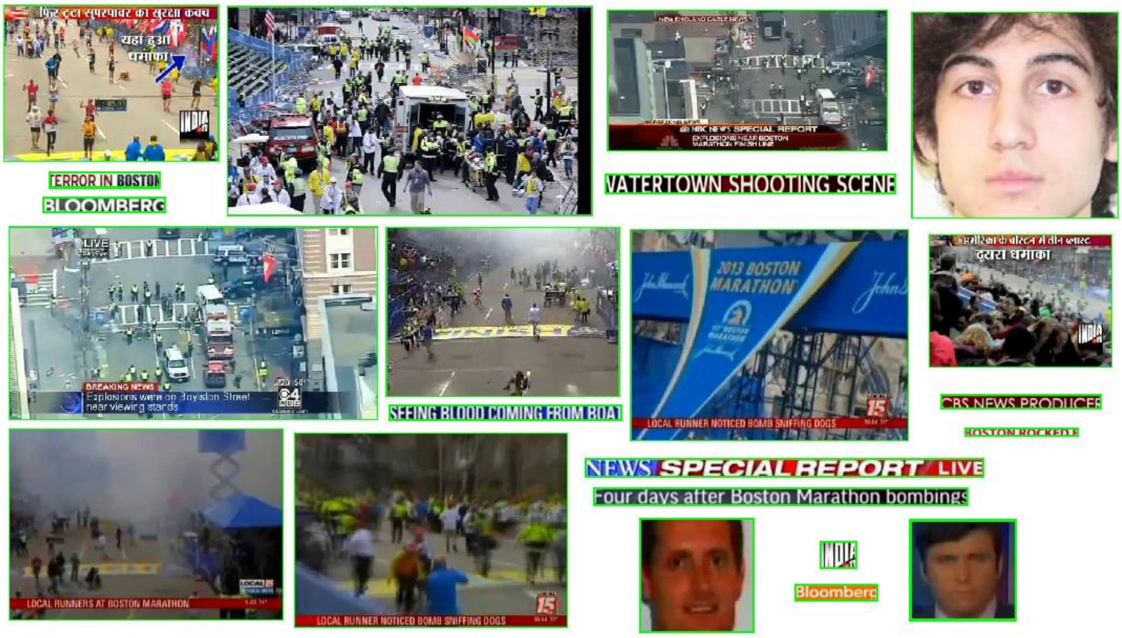

TERROR IN BOSTON
BLOOMBERG
WATERTOWN SHOOTING SCENE
NBC NEWS SPECIAL REPORT
EXPLOSIONS NEAR BOSTON MARATHON FINISH LINE
BREAKING NEWS
Explosions were on Boylston Street near viewing stands
SEEING BLOOD COMING FROM BOAT
2013 BOSTON MARATHON
LOCAL RUNNER NOTICED BOMB SNIFFING DOGS
CBS NEWS PRODUCER
BOSTON ROCKED
NEWS SPECIAL REPORT LIVE
Four days after Boston Marathon bombings
LOCAL RUNNERS AT BOSTON MARATHON
LOCAL RUNNER NOTICED BOMB SNIFFING DOGS
INDIA
Bloomberg

Fig. 58 Visual summary 1 of the news video collection 1 of the second user study

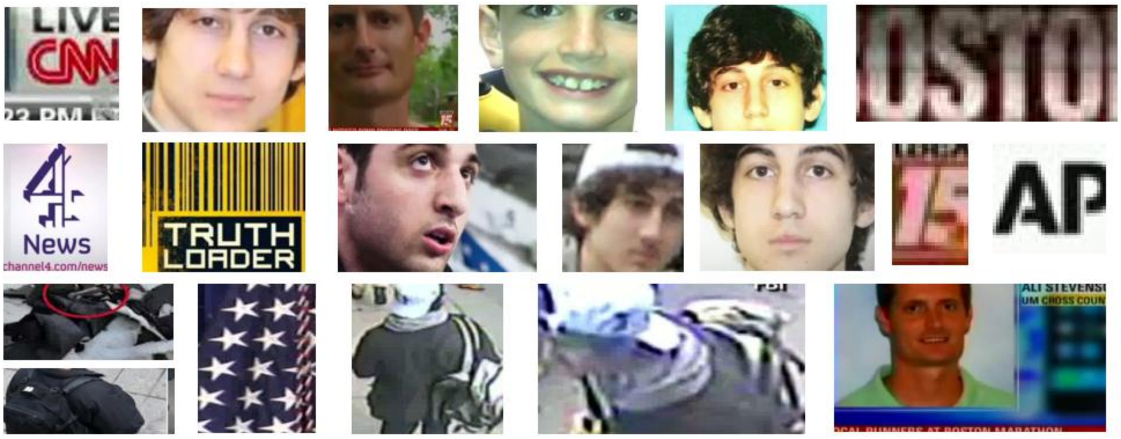


Fig. 59 Visual summary 2 of the news video collection 1 of the second user study

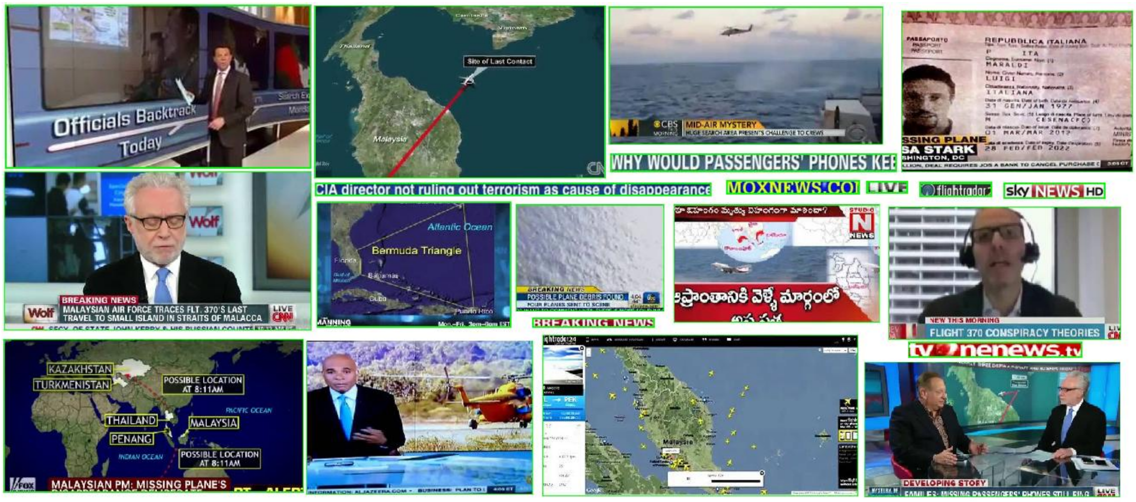

Officials Backtrack Today
Site of Last Contact
MID-AIR MYSTERY
WHY WOULD PASSENGERS' PHONES KEE
REPUBBLICA ITALIANA
SA STARK
CIA director not ruling out terrorism as cause of disappearance
MOXNEWS.COM
LIVE
flightradar
sky NEWS HD
Wolf
BREAKING NEWS
MALAYSIAN AIR FORCE TRACES FLT. 370'S LAST TRAVEL TO SMALL ISLAND IN STRAITS OF MALACCA
Atlantic Ocean
Bermuda Triangle
BREAKING NEWS
POSSIBLE PLANE DEBRIS FOUND
BREAKING NEWS
FLIGHT 370 CONSPIRACY THEORIES
tvonenews.tv
KAZAKHSTAN
TURKMENISTAN
POSSIBLE LOCATION AT 8:11AM
THAILAND
MALAYSIA
PENANG
POSSIBLE LOCATION AT 8:11AM
MALAYSIAN PM: MISSING PLANE'S
DEVELOPING STORY

Fig. 60 Visual summary 1 of the news video collection 2 of the second user study

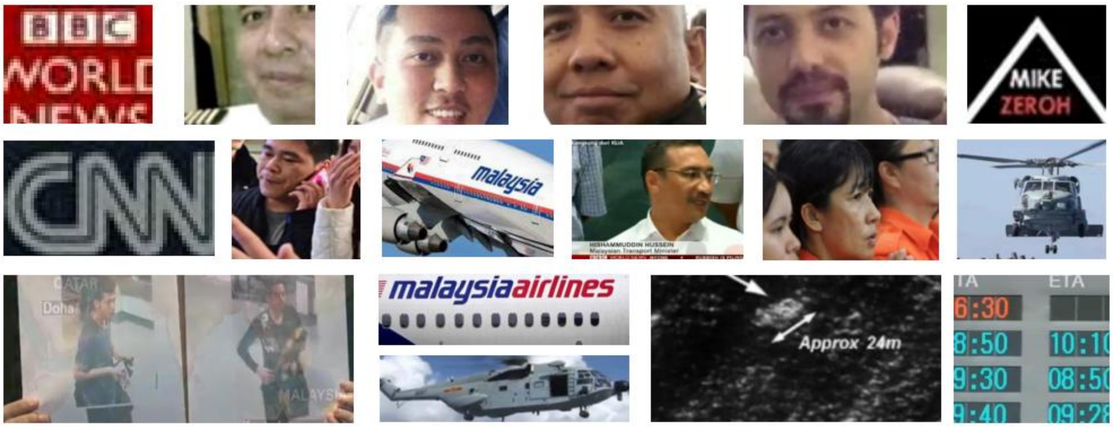


Fig. 61 Visual summary 2 of the news video collection 2 of the second user study

APPENDIX III. Development

This Appendix describes the programming tools and technologies we used for the development of the application. It is also explained how users can execute the tool.

- Image Plus:

Image Plus[25] is the new development platform in C++ language for the Video and Image Processing Group at UPC. You have to be registered in the Group to be able to work with this platform.

- Eclipse:

Eclipse[26] is an open source development platform, tools and runtimes for building, deploying and managing software. It was created by IBM in 2001. It allows developing projects in many languages as Java, C, C++, Python, etc. For the development of our tool, Eclipse has been used as an IDE.

- Subversion:

Subversion[27] is an open source Software Configuration Management tool. This version control system has been used with the Subeclipse connector as a secure code backup and allowing shared versions of the project with the advisors. Each member in the Subversion server has a branch to develop their applications and tools, and all branches are connected to a unique trunk that contains the shared version of the project.

- C++:

C++[28] is a general purpose programming language that is free-form and compiled. It is regarded as an intermediate-level language, as it comprises both high-level and low-level language features. It provides imperative, object-oriented and generic programming features.

It is one of the most popular programming languages and is implemented on a wide variety of hardware and operating system platforms. As an efficient performance driven programming language it is used in system applications, software applications, device drivers and so on.

[25] https://imatge.upc.edu/wiki/pmwiki.php?n=ImagePlus.ImagePlus (21/6/2013)
[26] http://www.eclipse.org (10/10/2013)
[27] http://subversion.apache.org/ (21/6/2013)
[28] http://www.cplusplus.com/ (20/5/2013)

- OpenCV:

OpenCV[29] is an Open source Computer Vision library of programming functions mainly, developed by Intel, and now supported by Willow Garage[30].
It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. It is written in optimized C/C++ and the library can take advantage of multi-core processing. It is free for both academic and commercial use.

Using the application

The tool visual_instances can be run with the following command line arguments:

```
      >>    visual_instances    <input_video>    <objects_folder>
<subfolders> <results> <haarcascade.xml> <rate>
```

Where each argument means:

<input_video> : This is the path of the video the user want to extract the most representative content.

<objects_folder> : Directory that contain all subfolders with the images of the objects the user want to detect in the video.

<subfolders> : Name of the subfolders which contain the images of the object that the user want to detect. Each subfolder has several images of the same object. The names must be separated only by comas. It is needed unless one object to be detected.

<results> : Folder where the results of the application will be saved.

<haarcascade.xml> : Location of the haarcascade_frontalface_default.xml file which is needed for the face detection.

<rate> : Desired user's rate (fps).

This could be an example of usage of the application:

```
      >> ./visual_instances.sh data/bbc_news.mp4 cfg/
police_car,ambulance,fire_truck data/
cfg/haarcascade_frontalface_default.xml 1
```

[29] http://opencv.org/ (28/5/2013)
[30] http://www.willowgarage.com/ (28/5/2013)

Appendix IV. Job array technique

First of all, we create the script makeindex.sh to generate a .txt file where we have the names of all mp4 videos from a directory. In this file, in each line we have the name of a video of the directory. This way, each line of this text file could be executed as a job. The code of the script we used to get all files from Boston bombing news was:

```
#!/bin/bash
for file in boston_bomb/*.mp4; do
echo $file >> index.txt
done
```

We also create a WorkerScript that call the tool with all its parameters. The code of the WorkerScript is copied below.

```
#!/bin/sh
INPUT_VIDEO=`sed -ne ${SLURM_ARRAY_TASK_ID}p <
/imatge/dalmendros/work/validation/videos/boston_bomb/index.txt`
OBJECTS_DIR=/imatge/dalmendros/work/validation/objects/
OBJECTS_LABELS=boston_bomb
CONTENT_RESULT_DIR=/imatge/dalmendros/work/validation/results/boston_b
omb/${SLURM_ARRAY_TASK_ID}/
CASCADE_FRONTAL_FILENAME=/imatge/dalmendros/work/wt_gui/haarcascade_fr
ontalface_default.xml
RATE=1

/imatge/dalmendros/workspace/imageplus/tools/representative_content/bi
n/release/representative_content/imatge/dalmendros/work/validation/vid
eos/boston_bomb/$INPUT_VIDEO $OBJECTS_DIR $OBJECTS_LABELS
$CONTENT_RESULT_DIR $CASCADE_FRONTAL_FILENAME $RATE
```

Where the variable SLURM_ ARRAY_TASK _ID depends on the execution. With the sed -ne we get the string of the line SLURM_ ARRAY_TASK _ID in the file. Once we have all the variables we can run the script.

In the case of the Boston Marathon bombings we have 356 videos in a folder. To execute our tool we use the following instruction

```
sbatch -J array --mem-per-cpu=2G --array=1-100 WorkerScript.sh
```

This way all the videos will be processed but with a maximum of 20 videos simultaneously, this means that initially videos from 1 to 20 will start processing. When one of them finishes, video 21 will start executing and so on. We will always have processing 20 videos in parallel until all the video collection finishes.