

## Chapter

# 3

## Crowdsourcing & Multimedia: Enhancing Multimedia Activities with the Power of Crowds

Ricardo M. C. Segundo<sup>1</sup>, Marcello N. de Amorim<sup>1</sup> e Celso A. S. Santos<sup>1</sup>

<sup>1</sup>Universidade Federal do Espírito Santo (UFES)

### *Abstract*

*This short course aims to present the concept of crowdsourcing and empower participants to implement this model of production in various activities linked with Multimedia and Web Systems, such as annotation, generation, summarization, synchronization, recommendation, retrieval, presentation and evaluation of the content quality. The idea behind crowdsourcing is to take advantage of the processing power of a multitude of employees to accomplish tasks that are "difficult for a computer", but which are apparently "simple to human intelligence". Describing the contents of an image or a video as inappropriate is an example of such difficult tasks, because the description of the problem by means of algorithms and automated techniques applied to parameters of this content is very complex and inaccurate. Another complex task is the subjective assessment of the quality of video encoders, the results depend on the user's perception and not only on parameters such as signal-to-noise ratio, resolution or frame rate. The crowdsourcing model tends to provide reliable results for these kinds of problems related to Multimedia and Web Systems. The additional content support for this course brings the fundamental concepts of crowdsourcing, a discussion of suitable scenarios for their use within the multimedia and examples of practical use of the concept in real-world examples.*

### **3.1. Introduction**

The use of crowdsourcing facilitates large or complex tasks, which would be: (i) non-viable to be resolved automatically by a computer, because of their characteristics; and (ii) non-viable to be resolved by one or few people due to the effort required to perform them. A typical example in the area of multimedia is the annotation of large multimedia databases (containing images, videos, audios) to identify which emotions each of these content tends to provoke. The task in question can be performed by any human collaborator, while any automatic system would have enormous difficulties in executing it.

Additionally, the huge amount of images makes it very costly to be held by a single person, or even a small group. In order to achieve the expected result the processing power of the crowd can be applied. This short course aims to empower participants to shape and build applications based on crowdsourcing, considering important issues such as the identification of activities that require human intelligence, the definition of the microtasks, as well as some points concerning implementation and integration of applications, in addition to essential issues, such as the characteristics of the crowd, employee retention and verification of the reliability of contributions.

Videos, for example, have a high-cost processing that encourages optimization of techniques and the use of alternative ones. Similar challenges for other media format (images) had success with crowdsourcing approaches. One such approach is the ESP Game, by [Von Ahn *et al.* 2008], that helps to determine the contents of images, providing meaningful labels based on contributions gathered from the crowd. Also, the investigation of crowdsourcing by the Multimedia community (<http://www.crowdmm.org/>) encourages a deeper investigation of the use of crowdsourcing with multimedia. Crowdsourcing is an approach to gather different kinds of collaborations such as services, information, knowledge and different kinds of data content from large communities of volunteers.

This material main purpose is to introduce readers to what is crowdsourcing, what kind of multimedia research can be done using it and how rapidly develop an application. To this purpose, the rest of this text presents: in Section 3.2 some concepts about crowdsourcing systems; Section 3.3 presents some challenges to multimedia research; Section 3.3 lists some researches involving crowdsourcing in diverse areas of multimedia, from text to videos; Section 3.4 guides readers through the development of a crowdsourcing image annotation application; and Section 3.5 concludes the text.

## **3.2. The Crowd**

Crowdsourcing is a term coined by [Howe 2006] to describe systems that are characterized by the contribution of people to forge from small tasks a bigger solution. But before we talk about crowdsourcing, lets discuss two important concepts: Crowd Wisdom and Human Computation.

### **3.2.1. Crowd Wisdom**

Collective intelligence is all intelligence that arises from the different knowledge shared on the same subject, assuming that no one has full knowledge, however we all know and have knowledge of something. [Levy 1994] says that collective intelligence is a distributed intelligence that is everywhere, incessantly valued, coordinated in real time, resulting in effective mobilization of competences, which seeks recognition and enrichment of people.

An example used by [Levy 1994] demonstrates a Crowd Wisdom situation: a student wants to know the weight of his backpack. Before measuring it in a scale, he asks his friends how much they think it weights. In the end, the mean of the opinions is almost the same as the real one measured in the balance. This story tries to show that with each person experiences together and probed, he can solve problems, as how to find out the weight of the backpack.

Nowadays the Internet has been used as a way to make more agile this kind of collective intelligence and the concept has gotten new forms. There are three ways to generate collective intelligence: conscious, unconscious and full intelligence.

On unconscious the user is working without even knowing it, simply by the act of navigating a web page or using an application that registers information. Examples of unconscious collective intelligence are: filling forms, click on pictures and links. "Each click with the mouse or the keyboard is a decision to be registered and used by a particular system that organizes and allows others to take advantage of the trail left by those who came before. In this way, a user can know what is the best-selling book in a bookstore, or in the case of a blog, which published articles was more read or reviewed, creating a relevant criterion, which speeds up the visitor's decision"[Cavalcanti & Nepomuceno 2007].

In conscious intelligence, users involved must strive to achieve a result and those involved can develop something for a cause. Examples of this are the intelligence development of free software and user support provided through forums where people are willing to solve given problems.

The full intelligence is one that involves both the unconscious and conscious intelligence in a single environment.

### **3.2.2. Human Computation**

Human computation [Law & Ahn 2011] is a term introduced by Luis von Ahn in his thesis and it refers to identify which part of a problem can be automated and which part require human intelligence to solve the problem. In this approach humans take the processing of a HIT (Human Intelligent Task), that usually presents better performance and efficiency, in comparison with a computer, because these tasks are associated with activities hard for computers but easy for humans. [Law & Ahn 2011] defines Human Computation as:

"... human computation is simply computation that is carried out by humans. Likewise, human computation systems can be defined as intelligent systems that organize humans to carry out the process of computation whether it be performing the basic operations (or units of computation), taking charge of the control process itself (e.g., decide what operations to execute next or when to halt the program), or even synthesizing the program itself (e.g., by creating new operations and specifying how they are ordered). The meaning of "basic" varies, depending on the particular context and application. For example, the basic unit of computation in the calculation of a mathematical expression can be simple operations (such as additions, subtractions, multiplications and divisions) or composite operations that consist of several simple operations. On the other hand, for a crowd-driven image labeling system, a user who generated a tag that describes the given image can also be considered to have performed a "basic" unit of computation." [Law & Ahn 2011]

Human Computation is extremely suitable to Crowdsourcing once it provides the base to identify, in the problem to be solved, the HITs that can be modelled as micro-tasks. Micro-tasks are simple and small, and demands an ordinary human to solve it.

Not all Crowdsourcing projects are based in micro-tasks, although it is the most common and usually associated to success cases. Fragmenting the problem into micro-tasks allows spreading the processing among a crowd of contributors, maximising the use of the processing power of this crowd.

Other characteristic that make micro-tasks suitable to Crowdsourcing is the independence between different micro-tasks. The independence of the job execution is very important in a scenario where you have potentially an enormous amount of users. Actually in these terms, Crowdsourcing is a particular case of Collaborative Approach where it is not desirable that users interact to others in order to avoid biasing their judgement, and a job can't depend on other results. Unlike the conventional Collaborative Approaches based in the 3C Model (Collaboration, Communication and Coordination), Crowdsourcing asks the collaborators for independent results that are merged in a future phase.

### **3.2.3. Crowdsourcing**

Crowdsourcing is a term that was first described in 2006 by Jeff Howe and Mark Robinson, at the time, both were editors of a magazine called Wired Magazine, that defined the term in the article: The Rise Of Crowdsourcing [Howe 2006].

Crowdsourcing uses the knowledge of a group of people who will be collaborators of information regardless of their creed, or geographic location. Crowdsourcing can be said as a type of outsourcing, where collective knowledge and volunteers are used to create content, develop new technologies, solve everyday problems and disposal services.

A more elaborated definition is found in [Estellés-Arolas & González-Ladrón-de Guevara 2012]:

”Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowd-sourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken.”

The commitment to the task, complexity and modularity, and the crowd participation by bringing their work, money, knowledge and/or experience, always implies mutual benefit to workers and crowdsourcers. The user will receive the satisfaction of a particular type of benefit, whether economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and use the advantage over what the user has brought to the venture, which will depend on the type of activity developed.

Crowdsourcing makes use of collective intelligence, the experiences and knowledge that are acquired through the crowd connected via the internet for the resolution of

problems, which may be responsible for the creation of new content and even the development of new technologies.

The Crowdsourcing can be considered a method of solving problems that in general has four fundamental challenges:

- How to recruit and retains users?
- What contribution can these users provide?
- How to join and use the contributions made by the crowd to solve the target problem?
- How to evaluate the user contributions and the users themselves?

Here we won't go into details on how to solve these questions, this may vary widely depending on how you will use the crowd within your problem. [Pedersen *et al.* 2013] defines a conceptual model that describes most of these dependences. The model is composed of: Problem, People (Problem Owner, Individual, and Crowd), Governance, Process, Technology and Outcome. Crowdsourcing systems can be classified according to its dimensions. These dimensions [Doan *et al.* 2011] are:

1. Nature of collaboration: can be implicit or explicit and specifies if the users know that their actions are generating contributions that are being used by the system, even if the main purpose of the work is unknown to the user.
2. Type of target problem: refers to the main problem approached in the paper, not the crowdsourcing aspect.
3. How to recruit and retain users: the crowd of users may be gathered by an open call recruitment, through a crowdsourcing platform, or by a direct selection made by the projects conductors. Also, collaborations may be extract from other systems.
4. What can users do: describes the classes of actions that can be performed by the users, such as annotation, evaluation and content generation.
5. How to combine their inputs: describes the methods used in order to merge individual contributions in a final output.
6. How to evaluate them: details which techniques are applied in order to verify reliability and quality of the contributions, such as gold patterns and control questions. Additionally, define if malicious users should be blocked or bad contributions lead to punishment.
7. Degree of manual effort: describes if the individual tasks are trivial or require a significant cognitive or manual effort.
8. Role of human users: establishes if users may collaborate providing self-generated content, executing individual tasks in a divide-and-conquer approach, acting as a system component over the target artifact, or as a perspective provider producing different perspectives of the problem that often can be combined in a better solution.

9. Standalone versus piggyback architectures: Some crowdsourcing applications count on standalone specific systems headed to collect collaborations. When collaborations are extracted from existing systems that weren't built specifically to collect the contributions the application is using a piggybacking approach.

### 3.3. Multimedia Powered By the Crowd

Crowdsourcing can be used in multiple activities linked with Multimedia and Web Systems, such as annotation, generation, summarization, synchronization, recommendation, retrieval, presentation and evaluation of the contents quality. In this section we describe examples of this use in some multimedia activities involving text, audio, image and video objects.

#### 3.3.1. CrowdText

Analysing text to detect emotions, digitalizing, summarizing, and have characteristics extracted are examples of text usage in multimedia. We present next some examples that may show readers how extensive the possibilities could be.

In medical context we can use the crowd to extract medical terms in patient-authored texts [MacLean & Heer 2013]. This work is based on the increasingly number of health-oriented applications, where users describe their conditions and receive feedback. They extract from forms medical terms that can be used to train classifiers. Other example in medical scope is the use of gamification techniques and crowdsourcing for training Natural Language Processing Tools on medical context [Dumitrache *et al.* 2013].

Analysing texts to extract characteristics is other activity that the crowd can do. In [Borromeo & Toyama 2015], the crowd is used to analyse sentiments in the text. It compared four different techniques: two crowdsourced ones (paid and volunteered), a manual and a automatic. Text simplification is other aspect that can be analysed by the crowd. [Lasecki *et al.* 2015] use the crowd to find the best simplified text, which is reduced complexity text that retains the same meaning of the original.

Maybe, one of the most common use of corwdsourcing regarding text is its digitalization. [Von Ahn *et al.* 2008] presented the reCAPTCHA system, where he implicitly ask humans to perform the digitalization of texts. This is implicitly, because the humans think they are simply entering a CAPTCHA to have access to a link, but in fact, they are transcribing a piece of text that a computer could not digitalize.

Translation is another common activity. [Schlippe *et al.* 2013] used the crowd to normalize texts to be used in their statistical machine translation method. [Corney *et al.* 2010] uses humans to directly translate non-Roman texts to be catalogued in libraries. As a final example, we have Duolingo ([von Ahn 2013]), an application that helps people learn other languages, while also helping to translate texts from one language to other.

#### 3.3.2. CrowdImage

In this subsection we present some examples of using the power of the crowd for processing images, such as quality evaluations and annotations.

We start with some examples of quality assessment. [Ghadiyaram & Bovik 2016]

created both a image dataset ("LIVE In the Wild Image Quality Challenge Database") that represents the real world scenario (not the ones generated by algorithms), and a crowd-sourcing platform to perform a quality assessment on their dataset, with more than 1162 images, resulting in more than 350,000 scores on this dataset. In their experiment, users had to say what kind of distortions the images presented. [Ribeiro *et al.* 2011] also presented a quality evaluation on images. They use a crowdMOS subjective measure, where users score from 1 (Bad) to 5 (Excellent) depending on the quality of the images. [Saad *et al.* 2016] focus are not in presenting a form for assessing images using the crowd, their focus is presenting challenges and pitfall in using the crowd for that. Examples of these points are tying the crowd responses to the stimulus response, using full-screen mode with real images, the length of the study and others.

Medical problems can also be solved with aid of the crowd. [Foncubierta Rodríguez & Müller 2012] used the crowd to create a ground truth to evaluate computer-based tools. They in first instance use the crowd to annotate medical images within multiple medical categories. The initial results are used as input for an automatic tool that annotates the rest of the dataset. The crowd is then used in a second instance to validate the results. Not directly related to medical tools, but also in the context of helping people, we find the UCap application ([Hoonlor *et al.* 2015]), that aims in helping blind people in their daily activities. The task to the crowd is to identify consumer products and inform the nutrition values to blind people when shopping.

Clustering images is other common computational problem that requires high performance, being a good candidate to be solved by the crowd. [Guo *et al.* 2015] used a hybrid solution to the clustering problem. Its proposal is to use the crowd to validate the output from a similarity algorithm, in this way the time and cost of the entire process is optimized.

### 3.3.3. CrowdAudio

Audio is the first media that presents a characteristic that not images nor text have: audio is a continuous media object. With the continuous characteristic, we need more time from the users to receive their contribution, and probably some more effort. In the sequence, we present some examples of using the crowd with audio.

First we have an example of using the crowd to generate media. In [Rizvi *et al.* 2015], the owner ask for content within an explicit criteria, such as noise pollution, where the crowd shall record noise pollution from a city. From the contributions, the owner can measure noise level in a city. This activity is done by the crowd, that receives the request on their devices, such as smartphones and tablets, and uses this device to collect and send the requested content.

As in image and text contributions, annotation is broadly used for audio context. The AMG1608 dataset ([Chen *et al.* 2015b]) used crowd to annotate 1608 30-second music clips using 665 subjects. Their goal is to provide a base to test and train automated recognition approaches. Similar to the AMG1608 dataset, the YourSpeech dataset ([Fretitas *et al.* 2010]) aims to provide training for Automatic Speech Recognition, but they use the crowd to both collect the audio and transcript it.

Transcription methods using crowdsourcing are broadly approached by researchers. [Lee & Glass 2011] use the crowd to transcribe short audio clips of about five to six seconds. These audios had originally a duration of an hour or more, but an audio of this size would require too much effort of one crowd member, therefore they automatically partitioned the audio and made each crowd task with 5 audio clips. [Saito 2015] also uses small chunks of audio to be transcribed by the crowd. But their goal was not just transcribe the audio files, their goal was create caption data to hearing-impaired people.

### 3.3.4. CrowdVideo

Video combines characteristics of the objects described previously: frames are composed images, the video has the continuous characteristic also present in audio, and subtitles, for example, are composed of text units. That is why we present a deeper analysis of crowdsourced video works, describing not only key papers, but also describing some characteristics of the crowd.

#### 3.3.4.1. Crowd Characteristics in Video Domain

We analysed multiple papers that use Crowdsourcing with videos, for multiple purposes, from annotation to video evaluation. Now we describe some works in relation to their crowdsourcing dimensions.

**Nature of Collaboration** The nature of collaboration reveals if the crowd actively contributed with the problem or if their contributions were implicitly collected. In explicit systems users can evaluate, share, network, build artifacts, and execute tasks [Doan *et al.* 2011].

A systems can let users build artifacts: [Wilk *et al.* 2015] created a composed live video presentation of user generated video. The workers selected the most suitable view of a recorded scene to switch, in other words, they chose the best video with the best audio.

Implicitly standalone systems provides a service such that when using it users implicitly collaborate. Games and video players are used to annotate and evaluate videos. Some presented papers that uses games as an approach to extract information from videos. In [Pinto & Viana 2013] the goal is to annotate the videos in form of games, where the crowd annotate a video (or a frame from it) and gains some punctuation from it.

Using video players, the systems can identify regions of interest in a video by analyzing where users zoomed a video [Carlier *et al.* 2011]; video highlights using the user navigation times inside a video; reconstruct 3D models using videos published by users; improve QoE using prefetching for video streaming [Pang *et al.* 2011]; and [Zegarra Rodriguez *et al.* 2015] used the crowd attention to a video to evaluate the quality of transmission.

Some systems presented both explicit and implicit collaborations. In [Liao & Hsu 2013] users that trace a route using GPS to their destination can be summoned to take photos of specific places that require a little detour from the original route. The

contributions are explicit, but rely on other applications to select the members of the crowd.

**Recruitment of users** Recruitment of user (worker) is an essential part of crowdsourcing: no users means no collaborations. Most studies pays their users using a Crowdsourcing recruitment platform (Amazon Mechanical Turk, Microworkers, CrowdFlow). In these systems the payment done to the crowd varies depending on the task, and sometimes depending on the user profile. In [Baveye *et al.* 2015] workers were paid US\$0.05 for answering five comparisons from a pair of video excerpts, selecting the one that conveys "the most positive emotion" or "the calmest emotion". [Vondrick *et al.* 2013] paid \$0.05 per object annotated, where a complete video clip annotation could cost between \$1 and \$2. In [Park *et al.* 2014] a total of 19 workers participated in experiments, who worked for an effective hourly wage between \$4 and \$6 as compensation. In general, however, the median value of a task used by most was between: 0.2 and 0.3 dollars.

Regarding recruitment of users we found papers that use from 9 to more than half a million contributions. The median of members is 288.7, disregarding the paper by [Huberman *et al.* 2009], that had 579471 members. This high value occurred because they used a piggybacking approach on youtube platform, allowing them to achieve such a number in their analyses.

**Retaining Plans** Crowdsourcing systems should consider how to further encourage and retain users after they are recruited. Gamification is a common technique used and brings to users an enjoyable experience that leads them to return and collaborate again. Among multiple best practices detailed in [Hoßfeld *et al.* 2014], they support the use of gaming factors in crowdsourcing systems. They claim that 80% of the users returns to a game, compared to only 23% for a regular task and unreliable ratings in their task annotation game are reduced to 2.3% instead of 13.5%, compared to a non-gaming task. Enjoyable experience with gamification has an intersection with another retaining strategy: competitions, such as top rated users.

**Tasks** The tasks executed by the crowd may have direct influence in collaborations resulted from the crowd. If a crowdsourcer delegates a task that is cognitively hard, he may receive less collaborations from users, while easy tasks stimulates users to contribute more. In [Keimel *et al.* 2012] from 99 users, only 7 accomplished all proposed tasks. In principle they expect users to access all videos from videoset. But probably the members of the crowd thought that the effort was not worth it, doing only part of the experiment.

As highlighted before, gaming is a common feature found in our primary studies. In [Di Salvo *et al.* 2013] users play a game where they mark fishes in the videos to gain score. For the TAG4VD game [Pinto & Viana 2013], players watch a video, and annotate a frame, resulting in points.

Another common task to the crowd is to make them watch videos and evaluate their quality: in [Rainer & Timmerer 2014] users watched and evaluated videos with a 0 - 100 quality scale using a single stimulus approach (one video at a time); in opposition, we

found that in [Figuerola Salas *et al.* 2013] users would watch two simultaneous videos (Double Stimulus Impairment Scale) and evaluated them in a 0-10 scale; instead of scales, users may watch videos and answer question on their perception about purpose, content and emotions [Kim *et al.* 2013]; a set of videos can be used instead to achieve a goal: in [Hossfeld *et al.* 2014] after watching the same video in 3 different resolutions each user filled a QoE test form.

Users may also receive tasks asking them to submit videos. These videos may contain extra information besides the video itself (like gps position, tags and others) to help on further processing [Chen *et al.* 2015a; Ferracani *et al.* 2015] or may be focused in collecting content from a specific event [Venkatagiri *et al.* 2015; Wilk *et al.* 2015] where it will provide a multiple view environment to spectators. On the other hand, the spectators may receive tasks to choose the best cameras and audio inputs [Wilk *et al.* 2015] to compose a video stream, or implicitly [Leftheriotis *et al.* 2012] users watching videos have their action monitored to collect navigation data.

For annotation tasks: [Aran *et al.* 2014] used users to produce crowdsourced annotations on the vlogs: the first type included several aspects of vlogs, such as resolution, amount of motion, and framing. The second one was the personality impressions about the vloggers; in [Vondrick *et al.* 2013; Burmania *et al.* n.d.] users identified objects inside key frames and annotated what they were; [Park *et al.* 2014] also made users watch the video, but they identified previously determined annotations on them just like what happened in [Baveye *et al.* 2015] where annotations were based on users relating video frames with emotions.

**Management of users and contributions** Crowdsourcing systems often must manage malicious users (users that try to receive the benefits by any means, without a real contribution). To do so, we can use a combination of techniques that block, detect, and detain those cases [Doan *et al.* 2011].

Gold standard (also found in papers as ground truth or control questions) consists in the fact that if a worker answers too many questions incorrectly, it suggests he is an unethical worker [Le *et al.* 2010]. Among the tasks sent to the crowd, questions with known answers are presented to users. Gold standard answers are hidden from the workers and used in postprocessing to estimate the true worker error-rate. With videos, however, we can use their temporal aspect to identify these users. We can verify how long a user watched a video before answering questions about it. We can even monitor if users are focused on the video (listening if user moved to other pages).

[Anegekuh *et al.* 2014] presents in their paper intituled "A Screening Methodology for Crowdsourcing Video QoE Evaluation" most of these techniques. They present an algorithm that takes into account some of the issues that have been identified to impact the quality of crowdsourcing results ((i) Accurate answers to questions; (ii) Determine if an evaluator has unique identification; (iii) Determine if an evaluator actually performs the task (i.e. watch the video before scoring); (iv) Accurate scores to hidden reference video; (v) verify if the evaluator not syndicate member).

### 3.4. Developing a Crowdsourced Multimedia Application

Until now, only fundamental concepts of crowdsourcing and examples of its use were addressed in the text. In this section we will setting out the steps involved with the development of crowdsourcing applications. There are multiple segments where crowdsourcing could be used, but we focus in only one to guide you through the development and use it as base to develop it in your own context. The context chosen in this guide is the annotation one. We will build an application to classify a human image dataset into: old and young people. The characteristics of the application are:

**Case:** Classification of a Image Dataset according to the age group of people in its pictures.

**Human Computation:** Humans can easily estimate the age group of people but this a very hard task for computers.

**Job:** The crowd classifies a person from an image into an age Group: baby, child, teen, adult or senior. In each job, each member makes 5 contributions (classifies 5 images);

**Dataset:** The images used for this case will be retrieved from Facebook user profiles.

**Application:** The job will be presented to users on a webpage. For each job 5 images will be displayed and should be annotated one by one. These images are retrieved from Facebook at the beginning of the job. In order to annotate each image the user must press the button associated to the age group of the person in the picture. The options are Baby, Child, Teenager, Adult, Senior, and Not a Person for the cases where there is not a human in the picture.

**Infrastructure:** All services used are certificated by SSL. An easy, free, and practical solution is used: we used Dropbox to host HTML, JS and CSS files, and Google Spreadsheets to persist the contributions.

Thinking about the nine dimensions we could characterize the application as follows:

1. Nature of collaboration: explicit.
2. Type of target problem: annotate images.
3. How to recruit and retain users: voluntary service of "friends".
4. What can users do: choose from six option in which category an images fits.
5. How to combine their inputs: we going to use majority voting to determine an image category.
6. How to evaluate them: if a user misses a golden standard, we disregard his contribution.

7. Degree of manual effort: see and click.
8. Role of human users: executing individual tasks in a divide-and-conquer approach.
9. Standalone versus piggyback architectures: standalone.

Figure 3.1 shows the crowdworkers' interface. You can see the image to be annotated, the options (buttons) and our question to the crowd.

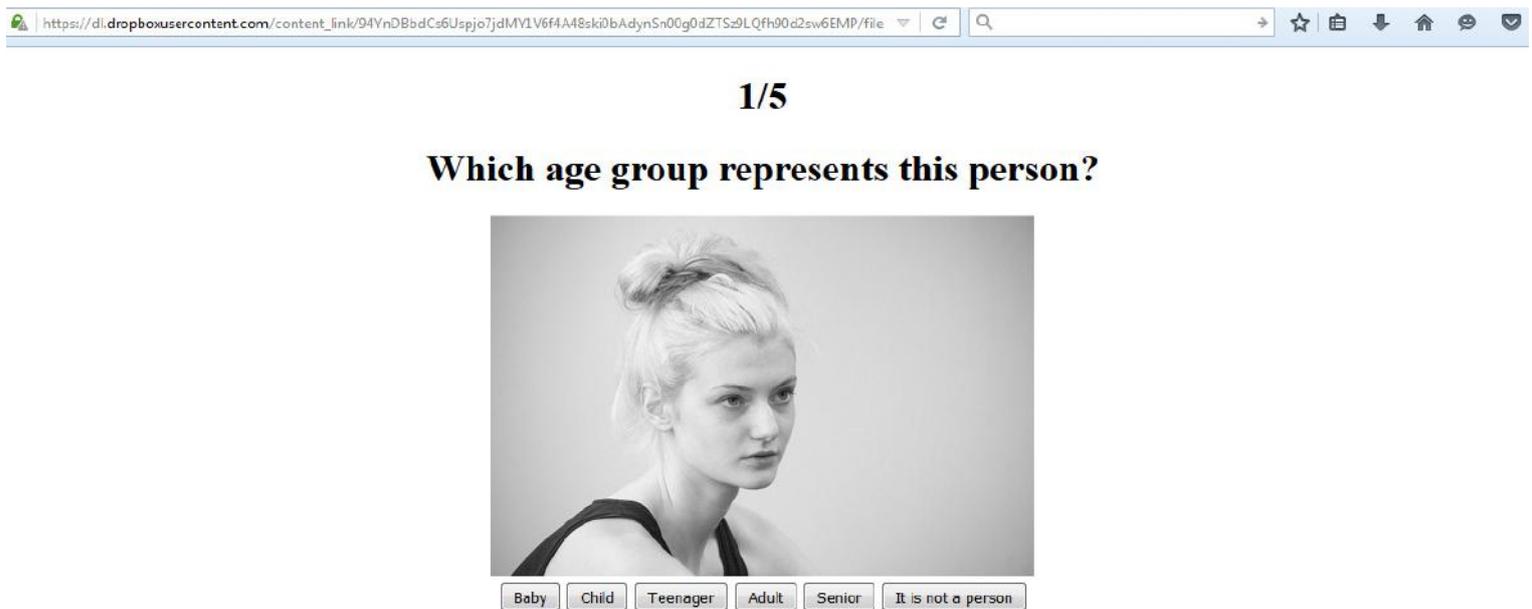


Figure 3.1. Application screen shot

### 3.4.1. Architecture

For this course, we developed an Architecture (Figure 3.2) that will allow users to implement a simple crowdsourcing APP task using free services and in a reduced time.

Our client (the crowd) needs a modern browser compatible with HTML5 specifications. Through the browser, the user accesses (1) the URL for the Web Application that is hosted in DropBox. The Application is composed only of HTML + JS + CSS specifications and is detailed in section 3.4.2. The DropBox returns (2) the code to the browser that constructs the application and enables the user contribution. After the user evaluates the 5 images, the contribution is sent through HTTPRequest through a get requisition.

The Google Spreadsheet (<https://support.google.com/drive/answer/37603?hl=en>) service allows to include new records as spreadsheets lines (over two million lines) from GET or POST HTTP requests, therefore we can use this service to store contributions from the crowd. In order to proceed it a HTTP request must be composed, including as parameters the columns' IDs and the values to store. We choose the method GET for this request for demonstration reasons because the parameters are explicit in the URL.

This request follows the pattern described latter, including the spreadsheet's ID, columns' IDs and values. To submit the HTTP request using JS you can use a XMLHttpRequest, although, as we are using Dropbox to host HTML and JS files we are in a Cross Domain scenario, and is not possible to proceed with the correct configuration for HTTP Access control (CORS).

The chosen way to skip this restriction was configure the request's mode as NO-CORS, and in this case is more practical to use the `fetch()` method from Fetch API instead of the `XMLHttpRequest.send()` method. The Fetch API is a modern replacement for XMLHttpRequest and works well, but as it is a new resource some old browsers do not support it.



Figure 3.2. APP Architecture

This kind of architecture is identified as a serverless architecture. This refers to applications that significantly depend on third-party services and puts much of the application behavior and logic on the front end. Such architectures remove the need for the traditional server system sitting behind an application.

The main weakness of our crowdsourcing application is related to the users(crowdworkers). We don't have control on recruiting them, nor there is any procedure to give to the user any benefits. This shall be solved aggregating other services such as a gamified one that would rank the users.

### 3.4.2. Coding

Based on our architecture and requisites, some development prerequisites for the first part of the code are necessary:

- A DropBox Account (<http://dropbox.com/>)
- A GoogleDrive Account (<http://drive.google.com/>)
- A JavaScript/HTML Editor (any)

#### 3.4.2.1. Configuring GoogleDrive

We are going to use Google Spreadsheets as our database. All contributions from users are stored in a spreadsheet for further analysis. In GoogleDrive you will: create a form with the fields you wish to recover from users and create a spreadsheet for this form.

In our tutorial application we want to recover the following informations for each image:

- Duration - we save how long the user took to annotate the image;
- Classification - we save the classification option selected by the user;
- IMG URI - the URI that identifies the annotated image;
- User Id - there is no way to guarantee 100% accuracy for anonymous contributions in our architecture, but we can have clues of which users made more than one contribution using fingerprinting techniques;
- Session Id - an ID, to identify all contributions made in sequence (each user identifies 5 images in sequence);

The first step is to create a new Google form. Title it as you wish, it will not interfere in the application. After, add the fields as short answers, in our case we used: time \_ duration, , image\_ url ,image\_ classification, user\_id and session\_id. Now go to the responses Tab, and click on Create Spreadsheet. This ends for now our conventional use of Google.

Now we have an informal use, where we need to identify each field and the document to be accessed by our application. To get these information we will use a pre filled link. This is a tricky part, so let's do it step by step:

1. First, in your form, click on the "Get pre-filled link" option;
2. Answer the questions (what you answer does not matter) then click submit;
3. Copy the generated URL;
4. Identify the form Reference and the fields:
  - (a) The reference to the form is the value starting with the **https://** until right before the **/viewform**;
  - (b) the fields start with the **entry** keyword followed by a dot and a number. The order of the fields are the same of the ones in the form;

As an example, the following lines present the original URL, the reference and the fields.

```
Pre-Filled URL:
https://docs.google.com/forms/d/e/1FAIpQLSfoexmSLhvScIx4OEcZxXHx6VikDjN5QApXMw6ZNqgEgaYVg/viewform?
entry.274168468&entry.1599775086&entry.1175968285&entry.405326048&entry.1284275155&entry
.595916278

Reference :
https://docs.google.com/forms/d/e/1FAIpQLSfoexmSLhvScIx4OEcZxXHx6VikDjN5QApXMw6ZNqgEgaYVg

Fields :
entry.274168468
entry.1599775086
entry.1175968285
entry.405326048
entry.1284275155
entry.595916278
```

### 3.4.2.2. Configuring DropBox

Dropbox will be used to host our web application. So to achieve this, create a folder to host all files needed. In our case, we called it "mypage". Also create an **index.html** to test and put some content inside. From this file, copy its dropboxlink, and concatenate it with the parameter: **& raw=1**. The new link now can be accessed as a web page. Paste it in the browser, and see your Web Page.

You also need the script link concatenated with **& raw=1**, to use it to import the script.

Figure 3.3 shows an example:

```
original link:
https://www.dropbox.com/s/2t48p3pu3qbn9pu/index.html?dl=0

Modified link:|
https://www.dropbox.com/s/2t48p3pu3qbn9pu/index.html?dl=0&raw=1
```

**Figure 3.3. A typical figure**

### 3.4.2.3. Codification

Now let's put our hands on some code. In the folder created in DropBox, you will need to have: an **index.html** and a **script.js**, so if you don't have it yet, create it.

The first step now, is to create the structure of our interface in the HTML file. We are going to need an image tag to define the image to be annotated, buttons for the options and a text area to inform how many images the user has done. The image will be set in the script, and we will get them from a remote service.

The following code contains this structure. We comment each important part of it, so you won't get lost. Note that the link to the script can't be a relative one, using dropbox we need the external link, described in section 3.4.2.2

```
<!-- Application Structure -->
<!DOCTYPE html>
<html>
<head>
  <title>MyFirst Crowd</title>
  <meta charset="utf-8">
  <!-- Turn our app more beautiful and "responsive", using Pure CSS modules -->
  <link rel="stylesheet" href="http://yui.yahooapis.com/pure/0.6.0/pure-min.css">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <!-- Include FingerPrinting Script -->
  <script src="https://cdn.jsdelivr.net/fingerprintjs2/1.4.1/fingerprint2.min.js"></script>
  <!-- Include our script (this must be the dropbox link to the script, concatenated to &raw=1) -->
  <script src="https://www.dropbox.com/s/0sifut74vfiz44n/script.js?dl=0&raw=1"></script>
</head>
<body>
  <div id="app" style="text-align:center">
    <!-- Output for the messages -->
    <h1 id="out">1/5</h1>
    <!-- Question to the user -->
    <h1 id="question">Which age group represents this person?</h1>
    <!-- Image area, with a default one -->
    <img height="300" id="image"/>
    <!-- Hides the buttons, as there is no images at beggining -->
    <div id="btn_Bar" style="display:none">
      <button id="btn_1" value="baby" class="button-xlarge_pure-button">Baby</button>
      <button id="btn_2" value="child" class="button-xlarge_pure-button">Child</button>
      <button id="btn_3" value="teenager" class="button-xlarge_pure-button">Teenager</button>
      <button id="btn_4" value="adult" class="button-xlarge_pure-button">Adult</button>
      <button id="btn_5" value="senior" class="button-xlarge_pure-button">Senior</button>
    </div>
  </div>
</body>
</html>
```

```

        <button id="btn_6" value="not_person" class="button-xlarge-pure-button">It is not a person</
            button>
    </div>
</div>
</body>
</html>

```

Next, we need to develop the logic in the script. The script will: get the images from a dataset, get the user contribution, the user id, include a golden standard image and send it to our database. Now we can go to the code. As the code is a little more complex than the HTML, we going to show and explain what each part of the code does.

First, let's see the variables of our script. There is no logic here, and the comments explain well what they are for.

```

var images = []; //Array to store the images URL;
var contrs = []; //Array to store the contributions: {time_start, time_stop, image_url, image_class,
    user_id, contr_id}
var goldenImgsIds = //Array with our golden images
    ["http://goo.gl/ZXa6MJ",
    "http://goo.gl/VRAKdt"];
var id; //Stores the user fingerprinting;
var counter = 0; //Sets in which step the user is;
var contrNumber = 5; //HowMany contribution we want;
var btnBar; //Reference to the ButtonBar;
var img; //Reference to the image container;
var out; //Reference to the output field;
var time = 0; //Stores how long the took the contribution;
var stamp = new Date().getTime(); //A stamp to this contribution;

```

When the HTML is built (window.onload), our application begins. We randomly load the images from the remote server, we include a golden standard image among them to help on evaluating the contribution, get references to the HTML components, add listeners to the button(the user action), get a timestamp to identify the user and start collecting the contributions.

```

//Function called when the HTML is done;
window.onload = function(){
    //Connects to the image service, and retrieves images;
    images = loadRemotelImages();
    //Inserts a Golden Standard
    images = goldenStandardIn(images);
    //Gets the references to the html elements;
    img = document.getElementById("image");
    btnBar = document.getElementById("btn_Bar");
    out = document.getElementById("out");
    //Gets the user fingerprinting and attributes it to the id variable;
    //This is an asynchronous function;
    new Fingerprint2().get(function(result, components){
        this.id = result;
    });
    //Starts the application listeners;
    btnBar.addEventListener("click",buttonClick);
    //Starts the first contribution;
    nextContribution();
}

```

A Golden standard is useful to help on judge the user contribution. A Golden Standard image is an image that we already know, and is included to help judge the user contribution. If the user mistake the golden standard, there is a chance that his other contributions are also wrong. In our APP we include among the images, one of those that we already know (goldenImgsIds) into a random position. In this case, we have only two in one example to keep it short.

```

//Randomly inserts a golden image in one of the images position;
function goldenStandardIn(images){
    //The position to put the golden image;
    var r1 = Math.floor(Math.random() * contrNumber);
    //One of the listed golden images;
    var r2 = Math.floor(Math.random() * (goldenImgsIds.length));
    //replaces the position;
    images[r1] = goldenImgsIds[r2];
    return images;
}

```

The contributions are started when we call nextContribution. This function writes in the screen how many contributions the user has done (2/5, for example), loads the image to be annotated on the screen, and when the image is loaded it starts to count how long the user will take to activate the buttons.

```
//Loads the next contribution from the user;
function nextContribution(){
  out.innerHTML = (counter+1)+ '/' + contrNumber;
  //Sets the image to be evaluated and shows the buttons;
  img.src = images[counter];
  //Ao carregar a imagem;
  img.addEventListener("load",function(){
    btnBar.style.display = 'block';
    //Starts counting time;
    time = new Date().getTime();
  });
}
```

When the image is loaded, the user can click on the option he thinks is right. When the button is activated, it starts some logic: the timer is finished and we store the duration of the contribution, we hide the buttons and the image and create a Contribution Object to store all information about that contribution (duration, image-url, user-choice, user-id, contr-id). If the user made enough contributions (in this case 5 times), we send it to the server, otherwise we get the next contribution.

```
//Handles the Click of the buttons;
function buttonClick(event){
  //Stops counting time, and saves its duration;
  time = (new Date().getTime()) - time;
  //Hides the image and blocks the buttons, so users don't click twice;
  img.src = '';
  btnBar.style.display = 'none';
  //Creates an contribution;
  var c = {
    time: time,
    image_url: images[counter],
    image_class:event.target.value,
    user_id: id,
    contr_id: id + stamp
  }
  //Adds the contribution to the contribution array;
  contrs.push(c);
  //Increments the counter;
  counter++;
  //If all the contributions are done, sends the contribution to the database;
  if(counter >= contrNumber){
    sendContributions(contrs);
  }else{
    //Else, gets the next contribution;
    nextContribution();
  }
}
```

When all contributions are done, they are sent one by one to the spreadsheet (store-Contribution, that comes next) and we inform it to the user. In the end we hide the figure, the button and invite the user to contribute again.

```
//Send all contributions, one by one;
function sendContributions(cs){
  //Hide all interface components;
  btnBar.style.display = 'none';
  img.style.display = 'none';
  document.getElementById('question').style.display = 'none';
  //Informs that all informations are being sent;
  out.innerHTML = 'Sending_Answers.';
  for(var i = 0; i < cs.length; i++){
    storeContribution(cs[i]);
  }
  //Thanks the contribution.
  out.innerHTML = 'Answers_Sent, Thank_YOU!!!:.';
  //Creates a link to reload the page, and contribute again;
  var a = document.createElement('a');
  var linkText = document.createTextNode("my_title_text");
  a.innerHTML = "Click_here_to_Contribute_again";
  a.href = "../index.html";
  document.getElementById('app').appendChild(a);
}
```

This is the function that communicates with Google Spreadsheet.

```

//Sends the contribution to the Database
function storeContribution(c){
  //Your Spreadsheet URL;
  var url = "https://docs.google.com/forms/d/e/1FAIpQLSfoexmSLhvScIx4OEcZxXHx6VikDJjN5QApXMw6ZNqgEgaYVg";
  //The form we send our information;
  var action = 'formResponse';
  //The field's id extraceted earlier:
  var fields = ['entry.274168468','entry.1175968285','entry.405326048','entry.1284275155','entry.595916278'];
  //We construct the URL to the database;
  var str = url+'?'+action+'?' +
    fields[0]+'='+c.time + '&' +
    fields[1]+'='+c.image_url + '&' +
    fields[2]+'='+c.image_class + '&' +
    fields[3]+'='+c.user_id + '&' +
    fields[4]+'='+c.contr_id;
  //We send the form;
  fetch(str,{mode: 'no-cors'});
}

```

Finally, this is the function to retrieve images from the image dataset, five for each request.

```

//Loads the images from the picture server;
function loadRemotelImages(){
  //Local array to store the images URL;
  var imgs = [];
  //Gets five images from facebook randomly (from a specific range)
  for(var i=0; i < 5; i++){
    rnd = Math.floor(Math.random() * 1000)%100 + 100000;
    url = "http://graph.facebook.com/v2.5/" + rnd + "/picture?height=300&height=300";
    imgs[i] = url;
  }
  //Retruns the array;
  return imgs;
}

```

### 3.4.3. Data Analysis

All workers' contribution are stored in a spreadsheet. Hence, we can use multiple approaches to analyse it: use metrics, manual analysis, MOS, its up to you. But one approach that the spreadsheet allows us to do is create a Google App Script that allows us to program an script to analyse data.

Our analysis follows these steps:

1. We have two spreadsheets indeed, one with the contributions and a second one with the Golden Standards;
2. The first step in our algorithm is to order by the contribution id, followed by the creation of an object for each session of an user (the five contributions in a row that he does);
3. Verify if in that session, the user scored the right value to the golden standard. If he missed, all contributions are discarded;
4. For each image, we count how many opinions for each possibility it had;
5. With these values we calculate the mode for each image, finding the opinion on that image.

The results for fifty images from Facebook profiles during 5 minutes with 4 crowd members resulted in 365 contributions. The list of the answers can be found in Appendix A.

### 3.5. Final Remarks

In this tutorial, we guided you through an overview of what is crowdsourcing and how to use it in diverse multimedia scenarios: text digitalization, audio transcription, video quality assessment and audio translation. We also showed, step by step, how to develop a web application to provide a simple platform to perform some crowdsourcing work, in our case, we performed the annotation of a image dataset about the age of the people in the images. This application lacks control of the crowd and maybe some security (anyone can submit to our form once the url is public), however it provides a free architecture that can be implemented by anyone, it also provides scalability and fast development (in four hours you were able to perform it). Next we showed how to aggregate value to our applications putting it in CrowdFlower Crowdsourcing platform.

### 3.6. Acknowledgments

This work is funded by the Brazilian Research Funding Agencies CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo).

### References

- Anekekuh, Louis, Sun, Lingfen, & Ifeakor, Emmanuel. 2014. A screening methodology for crowdsourcing video QoE evaluation. *Pages 1152–1157 of: Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE.
- Aran, Oya, Biel, Joan-Isaac, & Gatica-Perez, Daniel. 2014. Broadcasting oneself: Visual discovery of vlogging styles. *Multimedia, IEEE Transactions on*, **16**(1), 201–215.
- Baveye, Yoann, Dellandrea, Emmanuel, Chamaret, Christel, & Chen, Liming. 2015. LIRIS-ACCEDE: A video database for affective content analysis. *Affective Computing, IEEE Transactions on*, **6**(1), 43–55.
- Borromeo, Ria Mae, & Toyama, Motomichi. 2015. Automatic vs. Crowdsourced Sentiment Analysis. *Pages 90–95 of: Proceedings of the 19th International Database Engineering & Applications Symposium*. ACM.
- Burmania, Alec, Parthasarathy, Srinivas, & Busso, Carlos. Increasing the Reliability of Crowdsourcing Evaluations Using Online Quality Assessment.
- Carlier, Axel, Ravindra, Guntur, Charvillat, Vincent, & Ooi, Wei Tsang. 2011. Combining content-based analysis and crowdsourcing to improve user interaction with zoomable video. *Pages 43–52 of: Proceedings of the 19th ACM international conference on Multimedia*. ACM.
- Cavalcanti, Marcos, & Nepomuceno, Carlos. 2007. *O conhecimento em rede: como implantar projetos de inteligência coletiva*. Elsevier.
- Chen, Si, Li, Muyuan, Ren, Kui, & Qiao, Chunming. 2015a. Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. *Pages 1–10 of: Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*. IEEE.

- Chen, Yu-An, Yang, Yi-Hsuan, Wang, Ju-Chiang, & Chen, Homer. 2015b. The AMG1608 dataset for music emotion recognition. *Pages 693–697 of: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.
- Corney, Jonathan, Lynn, Andrew, Torres, Carmen, Di Maio, Paola, Regli, William, Forbes, Graeme, & Tobin, Lynne. 2010. Towards crowdsourcing translation tasks in library cataloguing, a pilot study. *Pages 572–577 of: 4th IEEE International Conference on Digital Ecosystems and Technologies*. IEEE.
- Di Salvo, R, Giordano, D, & Kavasidis, I. 2013. A crowdsourcing approach to support video annotation. *Page 8 of: Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications*. ACM.
- Doan, Anhai, Ramakrishnan, Raghu, & Halevy, Alon Y. 2011. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, **54**(4), 86–96.
- Dumitrache, Anca, Aroyo, Lora, Welty, Chris, Sips, Robert-Jan, & Levas, Anthony. 2013. Dr. Detective: combining gamification techniques and crowdsourcing to create a gold standard in medical text. *Pages 16–31 of: Proceedings of the 1st International Conference on Crowdsourcing the Semantic Web-Volume 1030*. CEUR-WS. org.
- Estellés-Arolas, Enrique, & González-Ladrón-de Guevara, Fernando. 2012. Towards an integrated crowdsourcing definition. *Journal of Information science*, **38**(2), 189–200.
- Ferracani, Andrea, Pezzatini, Daniele, Bertini, Marco, Meucci, Saverio, & Del Bimbo, Alberto. 2015. A System for Video Recommendation using Visual Saliency, Crowdsourced and Automatic Annotations. *Pages 757–758 of: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*. ACM.
- Figuerola Salas, Óscar, Adzic, Velibor, Shah, Akash, & Kalva, Hari. 2013. Assessing internet video quality using crowdsourcing. *Pages 23–28 of: Proceedings of the 2nd ACM international workshop on Crowdsourcing for Multimedia*. ACM.
- Foncubierta Rodríguez, Antonio, & Müller, Henning. 2012. Ground truth generation in medical imaging: a crowdsourcing-based iterative approach. *Pages 9–14 of: Proceedings of the ACM multimedia 2012 workshop on Crowdsourcing for multimedia*. ACM.
- Freitas, Joao, Calado, António, Braga, Daniela, Silva, Pedro, & Dias, M. 2010. Crowdsourcing platform for large-scale speech data collection. *Proc. FALA*.
- Ghadiyaram, Deepti, & Bovik, Alan C. 2016. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, **25**(1), 372–387.
- Guo, Xintong, Gao, Hong, & Wang, Hongzhi. 2015. Image Clustering Based on the Human Intelligence. *Pages 366–373 of: Intelligent Systems and Knowledge Engineering (ISKE), 2015 10th International Conference on*. IEEE.

- Hoonlor, Apirak, Ayudhya, Srisupa Palakvangsa Na, Harnmetta, Sukritta, Kitpanon, Sutichai, & Khlaprasit, Krisanat. 2015. UCap: A crowdsourcing application for the visually impaired and blind persons on Android smartphone. *Pages 1–6 of: 2015 International Computer Science and Engineering Conference (ICSEC)*. IEEE.
- Hossfeld, Tobias, Seufert, Michael, Sieber, Christian, & Zinner, Thomas. 2014. Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming. *Pages 111–116 of: Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*. IEEE.
- Hoßfeld, Tobias, Keimel, Christian, Hirth, Matthias, Gardlo, Bruno, Habigt, Julian, Diepold, Klaus, & Tran-Gia, Phuoc. 2014. Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. *Multimedia, IEEE Transactions on*, **16**(2), 541–558.
- Howe, Jeff. 2006. The rise of crowdsourcing. *Wired magazine*, **14**(6), 1–4.
- Huberman, Bernardo A, Romero, Daniel M, & Wu, Fang. 2009. Crowdsourcing, attention and productivity. *Journal of Information Science*.
- Keimel, Christian, Habigt, Julian, & Diepold, Klaus. 2012. Challenges in crowd-based video quality assessment. *Pages 13–18 of: Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE.
- Kim, Samuel, Georgiou, Panayiotis G, & Narayanan, Shrikanth. 2013. Annotation and classification of Political advertisements. *Pages 1092–1096 of: INTERSPEECH*.
- Lasecki, Walter S, Rello, Luz, & Bigham, Jeffrey P. 2015. Measuring text simplification with the crowd. *Page 4 of: Proceedings of the 12th Web for All Conference*. ACM.
- Law, Edith, & Ahn, Luis von. 2011. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **5**(3), 1–121.
- Le, John, Edmonds, Andy, Hester, Vaughn, & Biewald, Lukas. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. *Pages 21–26 of: SIGIR 2010 workshop on crowdsourcing for search evaluation*.
- Lee, Chia-ying, & Glass, James R. 2011. A Transcription Task for Crowdsourcing with Automatic Quality Control. *Pages 3041–3044 of: Interspeech*, vol. 11. Citeseer.
- Leftheriotis, Ioannis, Gkonela, Chrysoula, & Chorianopoulos, Konstantinos. 2012. Efficient video indexing on the web: A system that crowdsources user interactions with a video player. *Pages 123–131 of: User Centric Media*. Springer.
- Levy, Pierre. 1994. *Les technologies de l'Intelligence—L'avenir de la pensée à l'ère informatique*.
- Liao, Chen-Chih, & Hsu, Cheng-Hsin. 2013. A detour planning algorithm in crowdsourcing systems for multimedia content gathering. *Pages 55–60 of: Proceedings of the 5th Workshop on Mobile Video*. ACM.

- MacLean, Diana Lynn, & Heer, Jeffrey. 2013. Identifying medical terms in patient-authored text: a crowdsourcing-based approach. *Journal of the American Medical Informatics Association*, **20**(6), 1120–1127.
- Pang, Derek, Halawa, Sherif, Cheung, Ngai-Man, & Girod, Bernd. 2011. Mobile interactive region-of-interest video streaming with crowd-driven prefetching. *Pages 7–12 of: Proceedings of the 2011 international ACM workshop on Interactive multimedia on mobile and portable devices*. ACM.
- Park, Sunghyun, Shoemark, Philippa, & Morency, Louis-Philippe. 2014. Toward crowdsourcing micro-level behavior annotations: the challenges of interface, training, and generalization. *Pages 37–46 of: Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM.
- Pedersen, Jay, Kocsis, David, Tripathi, Abhishek, Tarrell, Alvin, Weerakoon, Aruna, Tahmasbi, Nargess, Xiong, Jie, Deng, Wei, Oh, Onook, & de Vreede, Gert-Jan. 2013. Conceptual foundations of crowdsourcing: A review of IS research. *Pages 579–588 of: System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE.
- Pinto, José Pedro, & Viana, Paula. 2013. TAG4VD: a game for collaborative video annotation. *Pages 25–28 of: Proceedings of the 2013 ACM international workshop on Immersive media experiences*. ACM.
- Rainer, Benjamin, & Timmerer, Christian. 2014. A quality of experience model for adaptive media playout. *Pages 177–182 of: Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*. IEEE.
- Ribeiro, Flávio, Florencio, Dinei, & Nascimento, Vítor. 2011. Crowdsourcing subjective image quality evaluation. *Pages 3097–3100 of: 2011 18th IEEE International Conference on Image Processing*. IEEE.
- Rizvi, AS M, Ahmed, Shamir, Bashir, Minhajul, & Uddin, Md Yusuf Sarwar. 2015. MediaServ: Resource optimization in subscription based media crowdsourcing. *Pages 1–5 of: Networking Systems and Security (NSysS), 2015 International Conference on*. IEEE.
- Saad, Michele A, McKnight, Patrick, Quartuccio, Jacob, Nicholas, David, Jaladi, Ramesh, & Corriveau, Philip. 2016. Consumer-photo quality assessment: Challenges and pitfalls in crowdsourcing. *Pages 1–6 of: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE.
- Saito, Takashi. 2015. A framework of human-based speech transcription with a speech chunking front-end. *Pages 125–128 of: 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE.
- Schlippe, Tim, Zhu, Chenfei, Lemcke, Daniel, & Schultz, Tanja. 2013. Statistical machine translation based text normalization with crowdsourcing. *Pages 8406–8410 of: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

- Venkatagiri, Seshadri Padmanabha, Chan, Mun Choon, Ooi, Wei Tsang, & Chiam, Jia Han. 2015. On Demand Retrieval of Crowdsourced Mobile Video. *Sensors Journal, IEEE*, **15**(5), 2632–2642.
- von Ahn, Luis. 2013. Duolingo: learn a language for free while helping to translate the web. *Pages 1–2 of: Proceedings of the 2013 international conference on Intelligent user interfaces*. ACM.
- Von Ahn, Luis, Maurer, Benjamin, McMillen, Colin, Abraham, David, & Blum, Manuel. 2008. recaptcha: Human-based character recognition via web security measures. *Science*, **321**(5895), 1465–1468.
- Vondrick, Carl, Patterson, Donald, & Ramanan, Deva. 2013. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, **101**(1), 184–204.
- Wilk, Stefan, Kopf, Stephan, & Effelsberg, Wolfgang. 2015. Video composition by the crowd: a system to compose user-generated videos in near real-time. *Pages 13–24 of: Proceedings of the 6th ACM Multimedia Systems Conference*. ACM.
- Zegarra Rodriguez, Demostenes, Lopes Rosa, Renata, & Bressan, Graca. 2015. No-reference video quality metric for streaming service using DASH standard. *Pages 106–107 of: Consumer Electronics (ICCE), 2015 IEEE International Conference on*. IEEE.







Dr. Celso Alberto Saibel Santos is Dr. in Fondamentele et Informatique Parallelisme by the Université Paul Sabatier Toulouse III (1999), MSc. in Engineering of Electronic Systems for the POLI-USP (1994) and Electrical Engineer UFES (1991). Has great experience in guidance at all levels of training and coordination of research and innovation projects, working mainly in the areas of multimedia, hypermedia and the Web. He published numerous scientific papers related to these areas in several vehicles. He has worked at University of Salvador (Unifacs), from 2001 to 2009, and at the Dep. of Computer Science of the Federal University of Bahia (UFBA), from 2009 to 2012. Currently, he is adjoint professor at Department of Informatics of the UFES and fellow of Productivity and Technological Development of Innovative Extension from CNPq-Brazil, since 2012.



Marcello Novaes de Amorim has received his MSc. in Informatics (2007) and BSc. in Computer Science (2005), both from Federal University of Espírito Santo (UFES). He is currently Ph.D. candidate in Computer Science also at UFES and has special interest in assistive technologies, crowdsourcing, mobile devices, education and application of new technologies.



Ricardo Mendes Costa Segundo has received his MSc. in Informatics (2011) and BSc. in Computer Science (2009) from Federal University of Paraíba (UFPB). He is currently a PhD candidate in Computer Science at UFES and his research interests are related to Multimedia Systems, Video Synchronization and Mobile Applications. He is also Lecturer at the University of Vila Velha (UVV) since.

