Crowdsourcing Event Detection in YouTube Videos

Thomas Steiner[1], Ruben Verborgh[2], Rik Van de Walle[2],
Michael Hausenblas[3], and Joaquim Gabarró Vallés[1]

[1] Universitat Politécnica de Catalunya – Department LSI
08034 Barcelona, Spain
{tsteiner,gabarro}@lsi.upc.edu
[2] Ghent University – IBBT, ELIS – Multimedia Lab
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium
{ruben.verborgh,rik.vandewalle}@ugent.be
[3] DERI, NUI Galway IDA Business Park, Lower Dangan Galway, Ireland
michael.hausenblas@deri.org

Abstract. Considerable efforts have been put into making video content on the Web more accessible, searchable, and navigable by research on both textual and visual analysis of the actual video content and the accompanying metadata. Nevertheless, most of the time, videos are opaque objects in websites. With Web browsers gaining more support for the HTML5 <video> element, videos are becoming first class citizens on the Web. In this paper we show how events can be detected on-the-fly through crowdsourcing (i) textual, (ii) visual, and (iii) behavioral analysis in YouTube videos, at scale. The main contribution of this paper is a generic crowdsourcing framework for automatic and scalable semantic annotations of HTML5 videos. Eventually, we discuss our preliminary results using traditional server-based approaches to video event detection as a baseline.

1 Introduction

Official statistics [26] from YouTube—owned by Google and one of the biggest online video platforms—state that more than 13 million hours of video were uploaded during 2010, and that 48 hours of video are uploaded every single minute. Given this huge and ever increasing amount of video content, it becomes evident that advanced search techniques are necessary in order to retrieve the few needles from the giant haystack. Closed captions allow for keyword-based in-video search, a feature announced in 2008 [7]. Searching for a phrase like *"that's a tremendous gift"*, a caption from Randy Pausch's famous last lecture titled *Achieving Your Childhood Dreams* [4], indeed reveals a link to that lecture on YouTube. If no closed captions are available, nor can be automatically generated [20], keyword-based search is still available over tags, video descriptions, and titles. Presented with a potentially huge list of results, preview thumbnails based on video still frames help users decide on the most promising result.

A query for—at time of writing—recent events such as the London riots[5] or the shooting in Utøya[6] reveals a broad selection of all sorts of video content, either professionally produced or, more often, shaky amateur videos taken with smartphones.

[4] http://www.youtube.com/watch?v=ji5_MqicxSo
[5] http://en.wikipedia.org/wiki/2011_London_riots
[6] http://en.wikipedia.org/wiki/2011_Norway_attacks

Despite these and other differences, their thumbnails are typically very similar, as can be seen in Figure 1. These thumbnails are automatically generated by an unpublished computer vision-based algorithm [6]. From a user's point of view, it would be very interesting to see whether a video contains different shots. For example, a back-and-forth between a news anchorman and live images can be an indicator for professionally produced content, whereas a single shot covering the entire video can be an indicator for amateur-generated eyewitness footage.

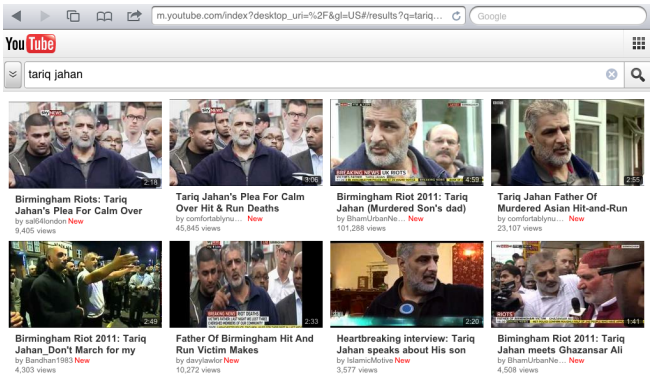

Fig. 1: YouTube search for "tariq jahan", father of a victim of the London riots.

In addition to the information provided by the separation of a video in shots, listing occurrences of named entities and their disambiguation can help users quickly decide whether a given video is of interest. For example, if a video about Utøya contains an occurrence of the Norwegian Prime Minister Jens Stoltenberg, or a video about the London riots contains an occurrence of the Prime Minister of the United Kingdom David Cameron, they can potentially be considered more trustworthy than other videos. It is up to the user to judge the trustworthiness aspect, however, the more context is available, the easier this decision gets.

While the detection of persons and their identification would be possible through face detection and face recognition techniques, this task is computationally expensive. As we have shown in [18], however, good results are possible through the analysis of the available textual metadata with Natural Language Processing (NLP) techniques, especially given the availability of (possibly automatically generated [20]) closed captions on YouTube. Finally, for videos that are longer than the attention span of a typical YouTube user, exploiting purposeful in-video navigation data can help determine points of interest within videos. For example, many users might skip the intros typically contained in professionally produced video content, or jump to spectacular shots directly.

We define three types of events: *visual events* in the sense of shot changes, *occurrence events* in the sense of the appearance of a named entity, and *interest-based events* in the sense of purposeful in-video navigation by users. In this paper, we report on a browser extension that enables crowdsourcing of event detection in YouTube videos

through a combination of *textual*, *visual*, and *behavioral* analysis techniques. When a user starts watching a video, three event detection processes start:

Visual Event Detection Process We detect shots in the video by visually analyzing its content [19]. We do this with the help of a browser extension, *i.e.*, the whole process runs on the client-side using the modern HTML5 [12] JavaScript APIs of the `<video>` and `<canvas>` elements. As soon as the shots have been detected, we offer the user the choice to quickly jump into a specific shot by clicking on a representative still frame.

Occurrence Event Detection Process We analyze the available video metadata using NLP techniques, as outlined in [18]. The detected named entities are presented to the user in a list, and upon click via a timeline-like user interface allow for jumping into one of the shots where the named entity occurs.

Interest-based Event Detection Process As soon as the *visual events* have been detected, we attach JavaScript event listeners to each of the shots and count clicks on shots as an expression of interest in those shots.

Fig. 2: Screenshot of the YouTube browser extension, showing the three different event types: *visual events* (video shots below the video), *occurrence events* (contained named entities and their depiction at the right of the video), and *interest-based events* (points of interest in the video highlighted with a red background in the bottom left).

Figure 2 shows the seamless integration of the detected events into the YouTube homepage. Contributions of this paper are the browser extension itself as well as the underlying crowdsourcing framework for automatic and scalable semantic annotations of HTML5 videos.

2 Related Work

Many different approaches to event detection in video exist. A first category is artificial vision, which tries to extract visual characteristics and identify objects and patterns. A second option is to reuse existing metadata and try to enhance it in a semantic way. Finally, using the combined result of collaborative human efforts can lead to data that is otherwise difficult or impossible to obtain.

2.1 Computer Vision Techniques

Searching through multimedia objects is inherently more difficult than searching through text. Multimedia information retrieval is still an active research topic with many challenges left to address [8]. One possibility is the generalization of text-based search to nontextual information [16], in which the query is posed as a multimedia object itself, the so-called query-by-example strategy. Another strategy is semantic indexing, *i.e.*, to annotate a multimedia item's content using textual or ontological means [9]. In this context, various feature extraction algorithms can be used, an interesting option being face detection [23] followed by face recognition [22].

2.2 Semantic Enrichment of Existing Metadata

In addition to automatically available metadata such as recording time and location, video creators can add metadata to their creations, such as title, textual description, and a list of tags. Also, YouTube automatically provides closed captioning in some cases. Unfortunately, these elements are not constrained to any framework or ontology, making automated interpretation difficult. Therefore, several efforts have tried to semantically enrich these existing metadata. Choudhury *et al.* [2] describe a framework for the semantic enrichment, ranking, and integration of Web video tags using Semantic Web technologies. They use existing metadata and social features such as related videos and playlists a video appears in. Gao *et al.* [4] explicitly model the visual characteristics of the underlying semantic video theme. This semantic model is constructed by finding the common features of relevant visual samples, which are obtained by querying a visual database with keywords associated with the video. Recently, Bræck Leer [1] also provided an interesting method to detect events in videos using semantic subtitle analysis. We previously described [18] a Web application that allows for the automatic generation of Resource Description Framework (RDF) video descriptions based on existing metadata. Textual information is enriched by extracting named entities via multiple Natural Language Processing Web services in parallel. The detected named entities are interlinked with DBpedia concepts. These entities are explicitly anchored to a point in the video thanks to the closed captions. In combination with a shot detection framework, entities can be anchored to shots instead, which is context-wise the better option.

2.3 Crowdsourced Annnotation Approaches

A radically different approach is to tackle the plethora of videos with the driving force behind it: an enormous community of users. The idea of crowdsourcing [3] is that, given the current limitations of automated vision and semantic analysis, we use human intelligence to perform those tasks in which humans currently excel. The aim is to make this task as easy and as less time-consuming as possible, in order to avoid disturbing a user's experience. Soleymani and Larson describe the use of crowdsourcing for annotating the effective response to video [17]. They discuss the design of such a crowdsourcing task and list best practices to employ crowdsourcing. The trade-off between the required effort versus the accuracy and the cost of annotating has been described by Vondrick *et al.* [24]. The quality of annotations generated by a crowdsourcing process has been assessed by Nowak and Rüger [14]. They conclude that a majority vote is a good filter for noisy judgements to some extent, and that under certain conditions the final annotations can be comparable to those of experts. Welinder and Perona [25] devise a model that includes the degree of uncertainty and a measure of the annotators' ability. It should be noted, however, that the usefulness of annotations also depends on their envisioned functional value, *i.e.*, what purpose they should serve in the application.

3 Crowdsourcing Event Detection in Videos

The term *crowdsourcing* was first coined by Jeff Howe in an article in the magazine Wired [11]. It is a *portmanteau* of "crowd" and "outsourcing". Howe writes: *"The new pool of cheap labor: everyday people using their spare cycles to create content, solve problems, even do corporate R&D"*. The difference to outsourcing is that the crowd is undefined by design. For our specific use case, any YouTube user with the browser extension installed could be part of that crowd.

Event detection in videos is an ideal candidate for crowdsourcing, as each video is an independent object in itself, *i.e.*, the whole set of all existing YouTube videos can be easily split into subtasks by just analyzing one video at a time. We store analysis results centrally, as outlined in Section 4. In the following, we explain for each event type the crowdsourced parts: for *visual* and *occurrence events*, shots and named entities in the video are detected once by whatever the first YouTube user that watches the video. Subsequent viewers can directly profit from the generated annotations. For *interest-based events*, acknowledging that points of interest within a video might change over time, we capture purposeful navigation events by all users. This allows for the generation of a heat-map-like overlay on top of the video shots, which results in an intuitive representation of popular scenes. Our advancement here is that we do not need write access to YouTube, but through our browser extension generate that metadata layer on top, while still creating a seamless and crowd-enriched experience for the user.

4 Implementation Details

We first provide an overview of the background technologies used in the framework and then explain how our browser extension works.

4.1 Background Technologies

Google Chrome Extensions Google Chrome extensions are small software programs that users can install to enrich their browsing experience with the Google Chrome browser. They are written using a combination of standard Web technologies, such as HTML, JavaScript, and CSS. There are several types of extensions; for this paper we focus on extensions based on so-called *content scripts*. Content scripts are JavaScript programs that run in the context of Web pages via dynamic code injection. By using the standard Document Object Model (DOM), they can modify details of Web pages.

Google Analytics Google Analytics is Google's Web analysis solution allowing for detailed statistics about the visitors of a website. The software is implemented by adding an unobtrusive snippet of JavaScript code on a website. This code collects visitor data through a request for an invisible image, during which the page and user data is reported back in the query part of the image's URL. The snippet also sets a first party cookie on visitors' computers in order to store anonymous information such as whether the visitor is a new or returning visitor, or the website the visitor came from.

4.2 Event Detection Processes

This paper is a first step in the direction of future work outlined in a prior publication [19]. Therein, we described the visual analysis-based shot detection algorithm in isolation and noted the potential of combining the visual results with textual analysis results following a method detailed in [18].

Visual Event Detection Process Our approach is based on HTML5 [12] JavaScript APIs of the `<video>` and `<canvas>` elements and falls in the family of histogram-based shot detection algorithms. The complete process has been detailed in [19]. We analyze the video frames' pixels tile-wise and calculate the local histograms in steps of one second. We then calculate the frame distances and finally split the video in shots wherever the frame distance is greater than the average deviation of all frame distances.

Occurrences Event Detection Process In [18], we document an interactive Web application that allows for the automatic annotation of YouTube videos in RDF based on title, description, tags, and closed captions. In the current implementation, we use `Factor`, `Product`, and `Agent` from the Event Ontology [15] to relate events to factors (everything used as a factor in an event), products (everything produced by an event), and agents (everything that can serve as an event agent). Listing 1 shows a sample video fragment annotated with the Event Ontology.

Interest-based Event Detection Process For each scene in a video, we generate a set of `<img>` elements. These sets get injected into the YouTube homepage's DOM tree, as can be seen in Figure 2. Each of the `<img>` elements has a registered JavaScript event handler that upon click triggers two actions: first, the video seeks to the corresponding time, and second, the shot is tracked as a point of interest in the video. We therefore use Google Analytics event tracking [5], logging the video ID and the video timestamp.

```
<http://gdata.youtube.com/[...]/9oWNcw8dits> event:Event :event.

:event a event:Event;
  event:time [
    tl:start "PT0.00918S"^^xsd:duration;
    tl:end "PT0.01459S"^^xsd:duration;
    tl:duration "PT0.00541S"^^xsd:duration;
    tl:timeline :timeline;
  ];
  event:factor <http://dbpedia.org/resource/David_Cameron>;
  event:factor <http://sw.opencyc.org/2008/06/10/concept/en/↩
      PrimeMinister_HeadOfGovernment>;
  event:factor <http://dbpedia.org/resource/Plastic_bullet>;
  event:factor <http://dbpedia.org/resource/Water_cannon>;
  event:product [
    a bibo:Quote;
    rdf:value """Prime Minister David Cameron authorized police
              to use plastic bullets and water cannons,"""@en;
  ] .
```

Listing 1: Exemplary extracted named entities from a YouTube video on the London riots.

4.3 Bringing It All Together

From a Linked Data [10] point of view, the main challenge with our browser extension was to decide on an as-consistent-as-possible way to model the three different event types of *visual events*, *occurrence events*, and *interest-based events*. We decided for a combination of two vocabularies: the Event Ontology [15] mentioned before, and the W3C Ontology for Media Resources [13], which aims to foster the interoperability among various kinds of metadata formats currently used to describe media resources on the Web. The ontology also allows for the definition of media fragments. For this purpose we follow the Media Fragments URIs [21] W3C Working Draft that specifies the syntax for media fragments URIs along several dimensions. The temporal dimension denotes a specific time range in the original media denoted by the `t` parameter. In our case, a media fragment is the part of a video spun by the boundaries of the shot that contains the frame that the user clicked. Listing 2 shows an exemplary semantic annotation of a 27s long video shot containing a *visual event* (the shot itself), an *occurrence event* (the DBpedia URI representing David Cameron), and an *interest-based event* (a point of interest spanning the whole shot).

5 Discussion of our Approach

Regarded in isolation, neither of our video event analysis steps is new, as detailed in Section 2. Our contributions are situated (i) in the scalability through crowdsourcing, (ii) in the on-the-fly HTML5 client-side nature of our approach, and (iii) in the combination of the three different event type annotations. Hence, we discuss our preliminary results

```
<http://gdata.youtube.com/[...]/9oWNcw8dits> event:Event :event1.

:event1 a event:Event;
  event:time [
    tl:start "PT0.025269S"^^xsd:duration;
    tl:end "PT0.05305S"^^xsd:duration;
    tl:timeline :timeline;
  ];
  event:factor <http://dbpedia.org/resource/David_Cameron>;
  event:product [
    a bibo:Quote;
    rdf:value """on camera. DAVID CAMERON, British prime
              minister: We needed a fight back, and a fight
              back is under way. [...] there are things that
              are badly wrong in our society. [...]"""@en;
  ];
  event:product ↩
      <http://gdata.youtube.com/[...]/9oWNcw8dits#t=25,53>.

<http://gdata.youtube.com/[...]/9oWNcw8dits#T=25,53> a ↩
    ma:MediaFragment.
```

Listing 2: Semantic annotation of a 27s long video shot (*visual event*) showing David Cameron (*occurrence event*) talk about the London riots. The shot is also a point of interest generated by a click of a YouTube user (*interest-based event*).

in contrast to a classic centralized approach. For *visual event* analysis, rather than detecting shots client-side with HTML5 JavaScript APIs, a centralized approach with low level video tools is superior in terms of accuracy and speed, as the video files do not have to be streamed before they can be processed. The crowdsourced approach is not necessarily more scalable, however, more flexible as it can be applied to any source of HTML5 video. For *textual event* detection, this is a task that necessarily runs centrally and not at the client due to the required huge text corpora. Finally, *behavioral event* detection by definition is only possible on the client. While most users are not aware that their navigation behavior can be used to detect points of interest and thus behave naturally, fraud detection is necessary to filter out spam pseudo navigation events.

In [3], Doan *et al.* introduce four questions for a crowdsourced system, the first being *how to recruit and retain users*. Our response is by seamlessly and unobtrusively enriching the user's YouTube experience. The user is not even aware that she is part of a crowdsourced system, and still profits from the crowd. Doan's next question is *what contributions can users make*. The response are annotations for the three event types defined earlier. The third question is *how to combine user contributions to solve the target problem*, with the target problem being to—in the longterm—improve video navigability, searchability, and accessibility. Our response is twofold: for *visual* and *textual events*, we consider only the first user's annotations, and for *behavioral events* we consider the annotations from all users by means of a heat map, as detailed in Section 3. The last question is *how to evaluate users and their contributions*. Our response is again

twofold. First, given that *visual* and *textual events* once detected are not questioned (as the outcome will always be the same), here the performance of individual users does not need to be evaluated. In contrast, the quality of *behavioral events* will simply improve by the combined wisdom of the crowd, always given proper fraud detection and future improvements mentioned in Section 6.

6 Future Work and Conclusion

Future work will focus on several aspects. First, given the streaming video nature, our approach inherits the speed and accuracy challenges from [19]; the solution here is to work with lower resolution versions of the video files in the background. Second, more elaborate interaction tracking for *interest-based events* is necessary. Facets like playing time after a navigational click can shine more light on the quality of the believed point of interest. If a user clicks on a supposedly interesting scene but then navigates away quickly afterwards, this is a strong indicator we need to consider. In the complete opposite, if a user never navigates within a video, this can be an indicator that the video is exciting from the first second to the last. Third, rather than just enriching the user experience for the current video, we will explore in how far the crowd-generated background knowledge gained on videos can be used for a more efficient video recommender system. This can be evaluated via A/B blind tests on clickthrough rates, where a user is randomly presented with a YouTube-generated related video recommendation, and a recommendation generated by the browser extension.

Concluding, our crowdsourced approach has shown promising results. The combination of *textual*, *visual*, and *behavioral* analysis techniques provides for high quality metadata that otherwise could only be generated through human annotators. Our framework is a scalable first step towards video event detection, with actionable steps ahead.

References

1. Bræck Leer, E.: Detecting Events in Videos Using Semantic Analytics of Subtitles. Master's thesis, University of Tromsø (Jun 2011)
2. Choudhury, S., Breslin, J., Passant, A.: Enrichment and Ranking of the YouTube Tag Space and Integration with the Linked Data Cloud. In: The Semantic Web – ISWC 2009, Lecture Notes in Computer Science, vol. 5823, chap. 47, pp. 747–762. Springer (2009)
3. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the World-Wide Web. Commun. ACM 54, 86–96 (April 2011)
4. Gao, Y., Zhang, T., Xiao, J.: Thematic video thumbnail selection. In: Proc. of the 16^{th} IEEE Int. Conf. on Image Processing. pp. 4277–4280. ICIP'09, IEEE Press, Piscataway, NJ, USA (2009)
5. Google: Google Analytics Event Tracking Guide, http://code.google.com/apis/analytics/-docs/tracking/eventTrackerGuide.html
6. Google Research Blog: Smart Thumbnails on YouTube (January 19, 2009), http://googleresearch.blogspot.com/2009/01/smart-thumbnails-on-youtube.html
7. Google Video Blog: Closed Captioning Search Options (June 05, 2008), http://googlevideo.blogspot.com/2008/06/closed-captioning-search-options.html
8. Hanjalic, A., Lienhart, R., Ma, W.Y., Smith, J.R.: The Holy Grail of Multimedia Information Retrieval: So Close or Yet So Far Away? Proc. of the IEEE 96(4), 541 –547 (Apr 2008)

9. Hauptmann, A., Christel, M., Yan, R.: Video retrieval based on semantic concepts. Proc. of the IEEE 96(4), 602–622 (Apr 2008)
10. Hausenblas, M., Troncy, R., Raimond, Y., Bürger, T.: Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. In: WWW 2009 Workshop: Linked Data on the Web (LDOW2009). Madrid, Spain (2009)
11. Howe, J.: The Rise of Crowdsourcing. Wired 14(6) (2006), http://www.wired.com/wired/archive/14.06/crowds.html
12. HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft (August 2009), http://www.w3.org/TR/2009/WD-html5-20090825/, http://www.w3.org/TR/2009/WD-html5-20090825/
13. Lee, W., Bürger, T., Sasaki, F., Malaisé, V., Stegmaier, F., Söderberg, J.: Ontology for Media Resource 1.0. Tech. rep., W3C Media Annotation Working Group (06 2009), http://www.w3.org/TR/mediaont-10/
14. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: Proc. of the Int. Conf. on Multimedia Information Retrieval. pp. 557–566. MIR '10, ACM, New York, NY, USA (2010)
15. Raimond, Y., Abdallah, S.: The Event Ontology (October 25, 2007), http://motools.sourceforge.net/event/event.html
16. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proc. of the IEEE 96(4), 548–566 (Apr 2008)
17. Soleymani, M., Larson, M.: Crowdsourcing for Affective Annotation of Video: Development of a Viewer-reported Boredom Corpus. In: Carvalho, V., Lease, M., Yilmaz (eds.) Proc. of the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010). ACM SIGIR, ACM (Jul 2010)
18. Steiner, T., Hausenblas, M.: SemWebVid - Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In: Semantic Web Challenge at ISWC2010 (November 2010), http://www.cs.vu.nl/~pmika/swc/submissions/swc2010_submission_12.pdf
19. Steiner, T., Verborgh, R., Van de Walle, R., Brousseau, A.: Enabling On-the-Fly Video Scene Detection on YouTube and Crowdsourcing In-Video Hot Spot Identification. In: Proc. of the 2^{nd} IEEE Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (submitted). ARTEMIS 2011, IEEE (2011)
20. The Official Google Blog: Automatic Captions in YouTube (November 19, 2009), http://googleblog.blogspot.com/2009/11/automatic-captions-in-youtube.html
21. Troncy, R., Mannens, E., Pfeiffer, S., Deursen, D.V.: Media Fragments URIs. W3C Working Draft (December 8, 2010), http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/
22. Verstockt, S., Van Leuven, S., Van de Walle, R., Dermaut, E., Torelle, S., Gevaert, W.: Actor recognition for interactive querying and automatic annotation in digital video. In: IASTED Int. Conf. on Internet and Multimedia Systems and Applications, 13^{th} Proc. pp. 149–155. ACTA Press, Honolulu, HI, USA (2009)
23. Viola, P., Jones, M.: Robust real-time object detection. In: Int. Journal of Computer Vision (2001)
24. Vondrick, C., Ramanan, D., Patterson, D.: Efficiently scaling up video annotation with crowdsourced marketplaces. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010, Lecture Notes in Computer Science, vol. 6314, pp. 610–623. Springer (2010), 10.1007/978-3-642-15561-1_44
25. Welinder, P., Perona, P.: Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: Proc. of the 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, San Francisco, CA, USA (Jun 2010)
26. YouTube: Official Press Traffic Statistics (2011), http://www.youtube.com/t/press_statistics