Automatic Extraction of Web Comics

Ben E. Cline
braininfo *at* benjysbrain.com

Abstract

Adult humans have no difficulty differentiating web comics from advertising and web page decorative images; however, state-of-the-art software does have difficulty with this task, especially given the dynamic nature of many comic strip web pages. But the ability to identify and extract comics from the web is useful for a number of applications such as the creation of personalized daily comics pages from a number of disparate web pages. Features that help differentiate comics from advertising and decorative images are described. To demonstrate the usefulness of these features, a comics discriminating filter based on an artificial neural network (ANN) was constructed. The filter provides a highly accurate means for differentiating between comics and other images found on comics web sites. It was tested in a web crawler for comics.

Introduction

Adult humans have no difficulty recognizing which images are comics and which are not on a newspaper comics page or editorial page. The same is true with a web page containing one or more comics; however, state-of-the-art software does have trouble differentiating comics from decorations such as banners and from advertisements. In some cases, characters from a comic strip are used in banners and in advertisements, making the identification task more difficult.

Web pages dedicated to comic strips and editorial cartoons are typically dynamic, showing a new cartoon daily, weekly, or at some indefinite interval. Advertising images can change with each visit. Site designers frequently change the layout of a comics site to make them fresh for regular visitors. Having software that can extract comics from dynamically changing web pages while discarding advertisements and decorations without human intervention is a desirable feature in several applications such as a personal comics page constructor and a crawler designed for extracting comics for indexing and online search.

Existing applications for this task are somewhat brittle, requiring human modification of extraction parameters as comic sites change (Tanvekski 1996, Medico 2001, Dunham and Bowditch 2003, Parker 2007). Some of these applications use regular expressions to identify which HTML tag contains the source address of a comic image while others reference the n^{th} image of a particular size on a page. Another approach is to use the size and aspect ratio of comics to decide which images are comics [Cline 2008]; however, this approach tends to allow a number of false positives, i.e., advertising images that are marked as being a comic.

Comics Features

Newspaper comic strips are typically displayed in one of three formats: The first is the typical strip, such as the daily "Pearls Before Swine" comic by Stephan Pastis, which is a long horizontal rectangle sometimes divided into three frames. The second format is slightly taller than wide and is used for some editorial cartoons and comics such as "Pluggers" by Gary Brookins. The third format is a larger format used in Sunday comics pages. Because many of the comic strips that appear in newspapers also appear on the web, these formats are replicated for many comics pages; however, some web-only strips such as "xkcd" by Randall Munroe vary their format with each strip. Even given the variability in formats, the size and aspect ratio of images on a comics web page are important clues in identifying comics. Image sizes and aspect ratios have been used successfully in web image retrieval systems to filter images (Nakapan, Halin, Bignon, and Wagner 2004, Hu and Bagga 2003).

Images presented with the HTML IMG tag have a true size and a rendered size. The rendered size can differ from the true size if the rendered size of the image is specified using the height and width attributes on the IMG tag. For example, a web page designer might use a full-size comic image as a link thumbnail by specifying a small rendered size for the image using IMG tag attributes. Software to

identify primary comic images can take the true and rendered sizes into account when inspecting images.

Low-level image processing techniques have been used in a number of web image crawlers to determine which images are photographs and which are line drawings. For example Frankel, Swain, and Athitsos (1996) used color features and Hu and Bagga (2003) used both color features and the DCT (Digital Cosine Transform) of image subregions. Since most comics are line drawings, image features that can identify drawings are useful in identifying comics.

Image position on a web page can also be a clue as to whether the image is a primary comic strip or a decoration or advertisement. Although layouts vary, many comics pages devoted to a single daily comic strip have banner and navigation images at the top of the page. The primary comic image is typically below the banner and on the left part of a medium-sized browser window. Images below the primary comic and those to the right of it are typically advertisements.

Besides image position, it is useful to know the parent tag containing the IMG tag. For images created by JavaScript which are not immediately displayed, there is no parent tag. Other images have the BODY tag as their parent, while others are positioned inside other structures. This structure surrounding an image is a clue to how the image is used.

The ALT attribute of the IMG tag allows the web designer to provide alternative text that a browser can display if there is a problem displaying the image and which can be rendered in audio for the visually impaired. The ALT tag could provide clues as to which image on a page is a comic; however, the ALT attribute is almost universally absent from comic sites.

ComicsExplorer Software

A web comics crawler called the ComicsExplorer, which uses the features described in the previous section, was constructed for this study to process a list of web pages and identify which images are comics and which are not. Because of the dynamic nature of comics pages and variability in layout, a flexible technique was needed to process images. For this reason, an artificial neural network (ANN) is at the heart of this software.

Figure 1 shows the crawler architecture and its two-stage image discriminating filter for removing non-comics images. The first stage of this filter discards images that are considered too small to be a readable comic. It removes small spacing images, some decorations, and some thumbnail images. The size check is based on the true size of images, not the rendered size.

The ANN that forms the second stage of the filter has ten input nodes, ten hidden nodes in a single hidden layer,

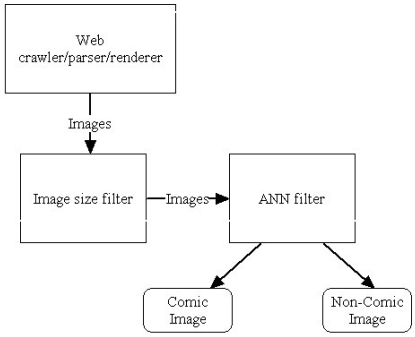

Figure 1 - ComicsExplorer Architecture

and an output node. The inputs to the ANN, based on the features discussed in the last section, are summarized in Table 1. The true and rendered dimensions of an image make up four of the inputs. The next two inputs are area and aspect ratio which are computed from the true image size. The page offset for the image is given in (x, y) pixel coordinates. The remaining two inputs, the average grayscale delta and the parent tag, are described below.

h, w	True height and width
rh, rw	Rendered height and width
area	h * w
aspect	Image aspect ratio
(x, y)	Position from upper left in pixels
avg-delta	Average grayscale difference
parent-tag	Encoded value based on IMG parent tag

Table 1 - ANN Input Parameters

Differentiating photographs and line drawings is useful in identifying comics on the web. A simple technique was developed for dealing with comics using grayscale histograms for both color and grayscale photographs and comics. Photographs tend to have smooth grayscale histograms, while comics tend to have a limited number of grayscale values that occur frequently since they are typically inked with a limited number of colors. Figure 2a shows a typical photograph histogram, while Figure 2b shows the histogram for a typical comic. A simple parameter for capturing the difference, called avg-delta in Table 1, is the average of the differences between each two adjacent grayscale values in an image histogram. Comics tend to have a higher value for this parameter than do photographs.

The "parent tag" input to the ANN is computed from the HTML tag that contains the IMG tag holding an image. Table 2 gives the values that are input to the ANN based on the parent tag. The parameter gives a simple indication of the containing structure of the image. Images that are generated by JavaScript and not rendered until some user action is performed have no parent tag and are given the value of zero. An IMG tag that is contained in the BODY tag is assumed to be part of only a limited structure, while

an IMG tag in a DIV tag is assumed to be part of some medium-scale structure. IMG tags contained in other HTML structures such as tables are assumed to be part of a greater structure.

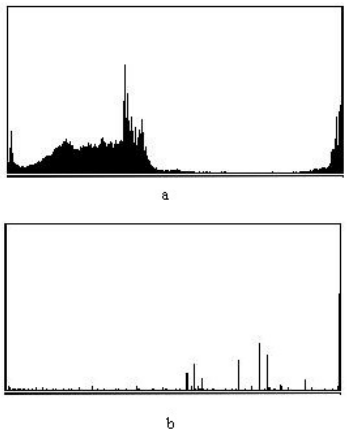

Figure 2 - Grayscale histograms of a photo (a) and a comic (b)

IMG Parent Tag	ANN Input Value
No parent tag	0.0
Body Tag	0.1
Div Tag	0.5
Other	1.0

Table 2 - Parent tag encoding

The ANN was trained using backpropagation with 104 images extracted from 33 web pages using the ten parameters listed in Table 1 along with a zero or one value indicating the desired output: not comic or comic. Some of the sites were dedicated to a daily comic, some displayed one or more editorial cartoons, and others were news or commerce sites. Some of the news and commerce sites contained no comics but did contain images of similar size.

In production, a web page is processed by passing the images from the page through the two-stage filter. Those images that pass both the size test and the ANN test are considered comic images.

System Performance

The goal for a robust comics extractor is that it properly locate comics on a set of comics web sites over time as the sites change due to daily comics changes, layout changes, and advertisement changes. The two-stage system described above processed two test sets daily over a three-month period. The first test set contained seventy-one sites including the sites in the ANN training set. (Training sites were included as they are dynamic with many of their images changing daily, i.e., the sites quickly diverged from their state when training images were extracted.) The second test contained the thirty-eight sites not contained in the ANN training set.

Table 3 summarizes the performance of the system. The system was able to differentiate between comics and non-comics on comics pages, commerce pages, and news sites with an average daily success rate of 93.7% on the first test set and a success rate of 92.7% on the second test set over the three-month period. For a typical day for the first test set, the ANN processed 148 images of which 98 were comics. Of the remaining 50 images, which were not comics, the ANN identified 48 correctly and identified two advertisements as comics. During the three-month period, only about 0.6% of true comics from both test sets where rejected by the ANN.

Test Set	Accuracy
Sites including training set	93.3%
Sites without training set	92.7%

Table 3 - Percent of correctly classified images

The comics extractors that use regular expressions or simple location measures to extract comics are 100% accurate until a site is remodeled, at which time, no comics are extracted until the user reconfigures the software manually. The ComicsExplorer is more robust in dealing with layout changes and does not require intervention when a site is remodeled. It was approximately 10% more accurate than a system that used only aspect ratio and size to select comics (Cline 2008).

Future Work

Additional filtering techniques could be added to the current two-stage filter to improve the accuracy of comics identification. Several techniques under consideration for improving the ComicsExplorer software are outlined in this section.

Bigham, Kaminsky, and Ladner (2006) and Guo, Kato, Sato, and Hoshino (2006) have used Optical Character Recognition (OCR) techniques to attempt to read text that is part of web images. Although OCR processing on typical web images is not perfect, some text strings extracted from images could be used to help determine the image function on the page.

Arai and Herman (2010) have worked on extracting individual frames from comics. Identifying images with frames would be a helpful feature in selecting which images are comics and which are not. Not all comics are subdivided into frames, and some comics have different formats relative to the number of frames from day to day.

However, an image with frames is more likely to be a comic than one without frames.

Additional knowledge about comics sites could aid the filtering process. For example, if it is known that a page contains a single daily comic and the existing two-stage filter selects multiple images as comics, additional software could compare the images to attempt to determine which of the images is the primary comic. Parameters such as relative position and size could be used in the comparison. The system could also use knowledge about the site obtained from pervious visits, such as the typical comic size, to better detect the primary comic image.

The NPIC system of Wang and Kan (2006) classifies non-photographic web images into a number of categories including cartoons. Their system uses both low-level image features and the textual context of web images in their classifier system. The ComicsExplorer uses simple image features that are mostly adequate for processing comics sites; however, the more extensive low-level image features in NPIC would be useful in improving the accuracy of the ComicsExplorer's image discrimination.

The ANN in the ComicsExplorer software was trained using images from weekday sites. The ANN could be retrained using both daily and Sunday pages in order to better deal with the typical Sunday comic format.

References

Arai, K. and Herman, T. 2010. Method for Automatic E-Comic Scene Frame Extraction for Reading Comic on Mobile Devices. *Seventh International Conference on Information Technology.* 370-375.

Bigham, J., Kaminsky, R., Ladner, R. 2006. WebInSight: Making Web Images Accessible. *ASSETS 2006.* 181-188.

Cline, B. E. 2008. Automatic Extraction of Comics from Web Pages. [Online] http://www.benjysbrain.com/misc/comics/.

Dunham, D. and Bowditch, E. 2003. ComicEZ. [Online] http://a-sharp.com/comicez/.

Frankel, C., Swain, M. J., and Athitsos, V. 1996. WebSeer: An Image Search Engine for the World Wide Web, Technical Report 96-14, Computer Science Department, The University of Chicago, Chicago, IL.

Guo, Q., Kato, K., Sato, N., and Hoshino, Y. 2006. An Algorithm for Extracting Text Strings from Comic Strips. *International Conference on Computer Graphics and Interactive Techniques: ACM SIGGRAPH 2006.* Research poster.

Hu, J. and Bagga, A. 2003. Identifying Story and Preview Images in News Web Pages. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition,* 640-644. Edinburgh, IEEE Computer Society Press.

Medico, A. 2001. Dailystrips. [Online] http://dailystrips.sourceforge.net/1.0.27/readme.html.

Nakapan, W., Halin, G., Bigon, J.-C., and Wagner, M. 2004. Extraction of Building Product Image from the Web. *International Journal of Intelligent Systems,* 19:65-78.

Parker, T. 2007. Comics Grabber. [Online] http://tevp.net/projects/comicsgrab/.

Tanvekski, T. 1996. Web Comic Strip Reader. [Online] http://sourceforge.net/projects/webcomicreader/.

Wang, F. and Kan, M.-Y. 2006. NPIC: Hierarchical Synthetic Image Classification Using Image Search and Generic Features. *Image and Video Retrieval, Lecture Notes in Computer Science.* 4071/2006, 473-482.