Key-text spotting in documentary videos using Adaboost

M. Lalonde, L. Gagnon*
R&D Department, Computer Research Institute of Montreal (CRIM),
550 Sherbrooke West, Suite 100, Montreal, QC, CANADA, H3A 1B9

ABSTRACT

This paper presents a method for spotting key-text in videos, based on a cascade of classifiers trained with Adaboost. The video is first reduced to a set of key-frames. Each key-frame is then analyzed for its text content. Text spotting is performed by scanning the image with a variable-size window (to account for scale) within which simple features (mean/variance of grayscale values and x/y derivatives) are extracted in various sub-areas. Training builds classifiers using the most discriminant spatial combinations of features for text detection. The text-spotting module outputs a decision map of the size of the input key-frame showing regions of interest that may contain text suitable for recognition by an OCR system. Performance is measured against a dataset of 147 key-frames extracted from 22 documentary films of the National Film Board (NFB) of Canada. A detection rate of 97% is obtained with relatively few false alarms.

Keywords: Text detection, Adaboost, video indexing, multimedia systems

1. INTRODUCTION

Important progress regarding text detection in video has been achieved over the last ten years [1-2], driven, in particular, by the need for automatic tools in the area of content-based video indexing [3]. However, many existing systems (academic/commercial) concentrate on the extraction of caption text only. Locating any type of text or portions of text (e.g. text appearing on street signs in urban scenes, adds, with or without occlusions) is still an active research area even for still images, as demonstrated by the 2003 and 2005 ICDAR Text Locating Contests [4-5].

The aim of this paper is to present an approach for spotting key-text in videos. Our definition of key-text is text that is important for a human to understand the story flow (e.g. text close-up, transition text between shots, subtitles, etc.). Key-text conveys important information that is emphasized during film montage.

Our basic architecture for text spotting is similar to that of Chen and Yuille [6], namely a cascade of classifiers trained using Adaboost [7-10], where the first stage is made of a simple classifier that discards as many false alarms as possible while letting potential candidates go through the following stages (with more complex classifiers). In our implementation, we make no effort to constrain the learning algorithm to use predefined features, so our feature pool is much richer. Another difference is that lookup tables are used as weak classifiers. The output of the cascade is a decision map of the size of the input image showing regions of interest that may contain text, and on which a commercial OCR can be applied. Tests are done on documentary films provided by the National Film Board (NFB) of Canada [11].

The paper is organized as follows. Section 2 presents an overview of the weak classifier cascade and Adaboost learning. Section 3 gives a description of the methodology followed for the feature selection, training and testing steps. Section 4 presents performance results. Finally we conclude with a discussion about the advantages and limitations of our approach, as well as direction for future works.

This work is part of on-going R&D activities at CRIM regarding the application of automatic video-content indexing tools for documentary films indexing within the MPEG-7 standard [12-13]. At this time, such tools include audio segmentation, speech recognition, visual shot boundary detection, keyframe and key-faces identification, face

* langis.gagnon@crim.ca; phone 514 840 1234; fax 514 840 1244; www.crim.ca/vision

recognition and motion characterization. Almost all of these tools can run automatically or semi-automatically as plug-ins for the public video processing/editing tool VirtualDub [14]. Using the VirtualDub framework simplifies the development process, especially for the I/O functionalities and the interface customization. The plug-ins extracts the visual content and optionally output MPEG-7 tags.

2. TECHNICAL BACKGROUND

2.1 Cascade of classifiers

A cascade of classifiers is used for locating text in images. It consists of several stages of classifiers with increasing complexity. The goal is to make the classifiers at early stages efficient and simple enough to reject a significant amount of bad text candidates while letting the good ones through. Additional downstream classifiers face a more difficult job because the trivially bad candidates have already been eliminated but since the remaining candidates are less numerous, more computationally expensive classifiers can be added to the cascade. Positive candidates are those that go through all the stages of the cascade (Figure 1). As described in [7-8], training the cascade implies training each stage with specific goals in terms of minimum detection rate and maximum false alarm rate, so that overall performance is satisfactory.

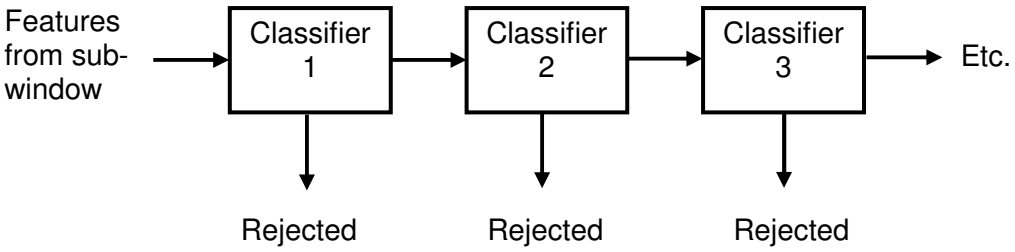


Figure 1: Diagram of a cascade of classifiers

2.2 Adaboost

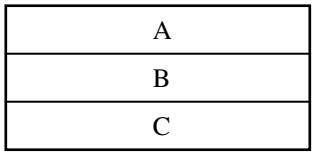
Each classifier of the cascade is trained using Adaboost, which builds a strong classifier based on the contribution of many weak classifiers (a weak classifier has very low complexity and is simply required to perform better than chance). Since weak classifiers are limited to one feature, Adaboost effectively performs feature selection as it manages to identify the best performing weak classifier. A short description of the algorithm goes as follows: with N-dimensional feature vectors x_i^j (j=1, ... ,N), and their corresponding class labels y_i (which equals 1 for vectors from good examples or 0 otherwise), weak classifiers are trained to classify them along each dimension. Among the N trained weak classifiers, the best one is stored along with a weight which is inversely proportional to its classification error. If the error is too high, an additional weak classifier is sought but examples that were previously classified without error are now given less weight in the error computation. Classification is the weighted sum of the responses of all its weak classifiers. The boosting algorithm to create a strong classifier based on weak ones is summarized in Table 1 of reference [8].

In this work (and following [16]), a pair of histograms play the role of a weak classifier. During training, all values corresponding to feature under consideration are extracted from the examples and stored into the appropriate histogram (the 'good' histogram for positive examples and the 'bad' histogram otherwise). After histogram normalization, classification of a candidate simply amounts to test whether the most populated bin, where the tested feature value would fall, belongs to the 'good' or 'bad' histogram.

3. METHODOLOGY

3.1 Features

Let us assume an analysis window within which text may be present or absent. The window is swept through the image and at each location a classifier must take a decision as to whether text is present or not, based on features inside the window. The pool of features explored by the Adaboost algorithm is similar to that found in [17], i.e. the window is split into (2x5) blocks stacked vertically and statistics (mean υ and standard deviation σ) are computed inside each block, either from the grayscale image or its corresponding X- or Y-derivative image. Features are then the actual value of these statistics or the signed difference of the statistics from various block configurations. For example, Figure 3 shows the case of a 3-block window with the features F characterizing this configuration.



$$F = [\mu_A, \sigma_A, \mu_B, \sigma_B, \mu_C, \sigma_C,$$
$$\mu_A - (\mu_B + \mu_C), \sigma_B - (\sigma_A + \sigma_C),$$
$$\mu_C - (\mu_A + \mu_B), \sigma_C - (\sigma_A + \sigma_B)]$$

Figure 2: Features extracted from a three-block vertical configuration

If vertical configurations are considered, with 2, 3, 4 and 5 blocks per window, each feature vector used for training has 240 elements, and the Adaboost training procedure will retain the most relevant ones.

3.2 Training set

The training set containing positive examples is a subset of 150 images from the training and test sets made available to the participants of the ICDAR 2003 text locating competition [4]. These sets include images of objects and urban scenes containing text, along with ground truth data stored as regions of interest in XML format. In the case of fairly large regions corresponding to long strings of text, they were split into smaller chunks. Negative examples were generated from the same images (outside the regions of interest, of course) as well as from an in-house collection of 950 images of natural panoramas. A total of 3411 good examples and 12243 bad examples were used for training.

3.3 Final system characteristics

Training the cascade is performed using the features described in Section 3.1. Each stage n has to achieve a certain detection rate of 0.95^n and a false alarm rate of 0.5^n, with overall false alarm rate below 5% for the whole cascade. All error are computed against a validation set whose size is about 1/3 of the entire training set, the rest is used for actual Adaboost training. Training time is about 3 hours, including the computation of the integral images [7-8] for the 1100 greyscale images as well as their x- and y-derivative images, feature extraction, bad candidate generation and cascade training. Training *per se* (once feature vectors are built) is 2-3 minutes on a desktop PC (3 GHz). The resulting architecture is made of 3 stages, with 4, 5 and 4 weak classifiers each. It is interesting to note that some of the features found to be discriminant by the training procedure are similar to those proposed in [6]. The 'best' feature of the first stage is a difference of variances drawn from the X-derivative image with the 4-block configuration (Figure 3, top left), which makes a lot of sense. As one can notice, the X-derivative image is highly exploited.

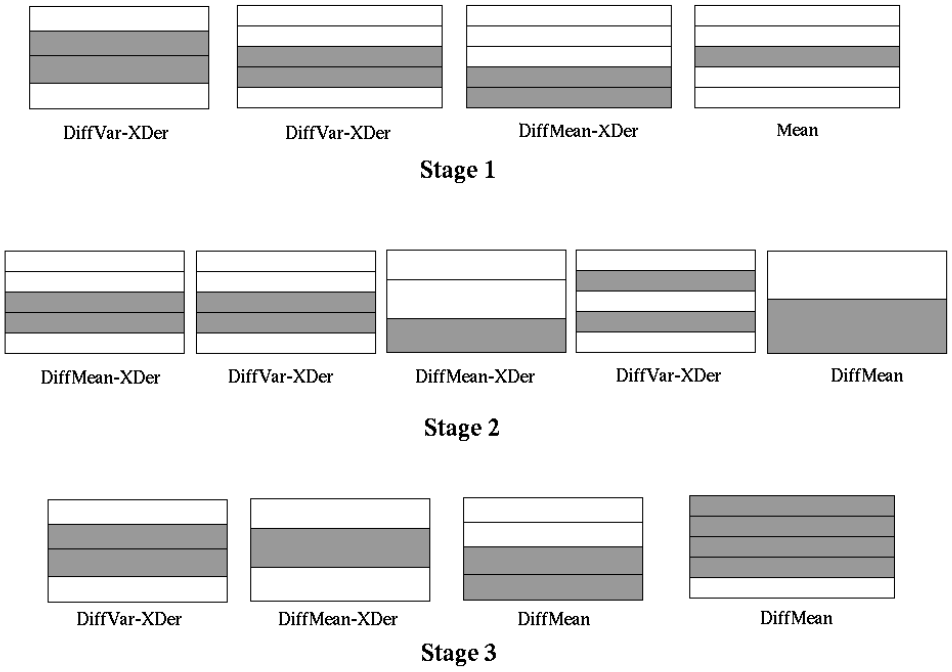


Figure 3: Best features found during cascade training with Adaboost. 'DiffMean' means that the feature is the difference between the sum of the means of the grey blocks and the sum of the means of the white blocks (idem for variance)

4. TESTS AND RESULTS

4.1 Test set

Our test set is a collection of 147 key-frames extracted by hand from excerpts of 22 documentary films provided by the National Film Board (NFB) of Canada [11]. Films are quite diversified as they take place in natural and urban settings, and feature people, animals, sceneries, etc. Some films are old (around 1950), black and white, and of poor image quality. Others are more recent and in color. As far as text extraction is concerned, besides the opening title, they may include subtitles, superimposed explanatory text, shots of newspaper front pages, various signs, etc. Text appearance may vary significantly with respect to font, size, contrast, etc. From these 147 key-frames, 443 text strings were manually labelled as ground truth data. Figure 4 shows some examples of key-frames.

Figure 4: Examples of key-frames from documentary videos

4.2 Results

The text detector processes the 147 key-frames along three scales, i.e. three analysis windows of 40x20, 80x40 and 120x60 pixels in size (image size is 640x480 pixels). Processing time was roughly 1 second per frame on a desktop PC running Linux (C++ code slightly optimized besides using integral images). In order to evaluate whether the response of the detector was right or not, the following criteria were used:

- An analysis window is said to have found text when its centroid falls inside the bounding box of the text string (available as ground truth data); the detection error is defined as 1 minus the 'coverage' (in %) of the bounding box by analysis windows that hit the text. A detection error of 10% means that 90% of the text area contained in the image is covered.
- False alarms happen when the centroid of an analysis window falls outside all ground truth bounding boxes. The number of false alarms that is recorded corresponds to the number of clusters of neighbouring analysis windows that erroneously found text (the logic behind this is that neighbouring windows might be aggregated and the number of aggregations is an indication of the number of regions inside which more complex analysis would have to be performed)

In terms of detection error, results are quite good: 96.9% of the text areas in key-frames were found, even though the challenge was high in some cases (see Figure 5 and Figure 6: in the latter case, we don't even expect an OCR system to successfully recognize the text!). Out of the 147 key-frames, 9 caused significant errors (detection rate below 90%) as the text they contained was oriented, huge or handwritten (4 out of 9), suffered bad contrast (1), was close to the image border (1) or was fuzzy because of poor image quality (3) (Figure 7).

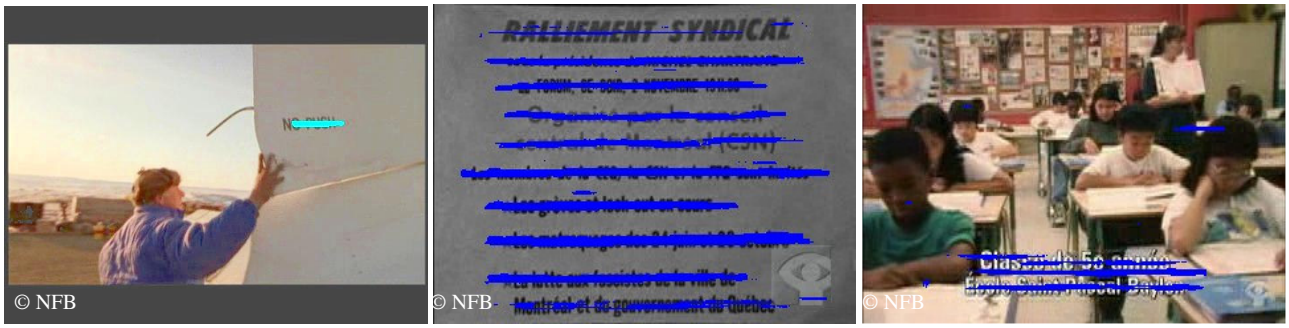


Figure 5 : Some detection example. The large and irregular stripes are made of dots locating the centroid of the analysis windows where the text detector 'fires'.

Figure 6: Challenging cases that were successfully met with low false alarms: 'For rent' (A LOUER in French) sign (top left), bus line (top right), 'ISO9001' banner filmed from moving vehicle (bottom left), license plate (bottom right).

Figure 7: Examples of detection failures

As for false alarms, visual inspection shows that they are quite low. Quantitatively however, the average number of clusters of analysis windows is about 15 per image (Figure 8). At least two reasons can explain that:

- one historical documentary contains a significant number of shots of newspaper pages, and many text strings were detected in the small print which was excluded from ground truth (only headlines were considered)
- text is sometimes found between lines of text; furthermore, the detector tends to 'stretch' the text lines, probably because of its sensitivity, i.e. it fires even when the window contains a single letter that occupies a fraction of its surface (in which case the centroid of the window is outside the ground truth bounding box, hence a false alarm). This 'feature' helped recover text strings made of characters that are far apart (Figure 8, right).

Figure 8: Detection examples that also generate many false alarms (left and center images);
Sensitivity helps recover characters that are far apart (right)

5. CONCLUSION

We have presented results about an system for detecting key-text in documentary videos using a cascade of classifiers trained with Adaboost. The learning algorithm explores a rich pool of features and lookup tables are used as weak classifiers. The output of the cascade is a decision map of the size of the input image showing regions of interest that may contain text, and on which a commercial OCR can be applied.

So far, good results (detection rate of 97% with low false alarms) have been obtained with a classifier of fairly low complexity. Before moving on to the next logical step, i.e. addition of pre-processing operations that set the stage for OCR integration, it would be interesting to address the following issues:

- Influence of weak classifier LUT vs. decision stumps. Look-up tables approximate the probability functions of the features involved in training, and as they make a decision according to the highest bin associated to 'candidates' feature values, they act as maximum-likelihood classifiers. One would imagine that this type of weak classifier would be less sensitive to potential class overlaps than decision stump classifiers that assume a threshold separating the positive and negative examples can be found. Inspection of the LUTs following training tends to show that stump classifiers would perform well in some situations but poorly in others (see Figure 5 rightmost LUT; this weak classifier is the weakest of all four classifiers of the first stage, however).

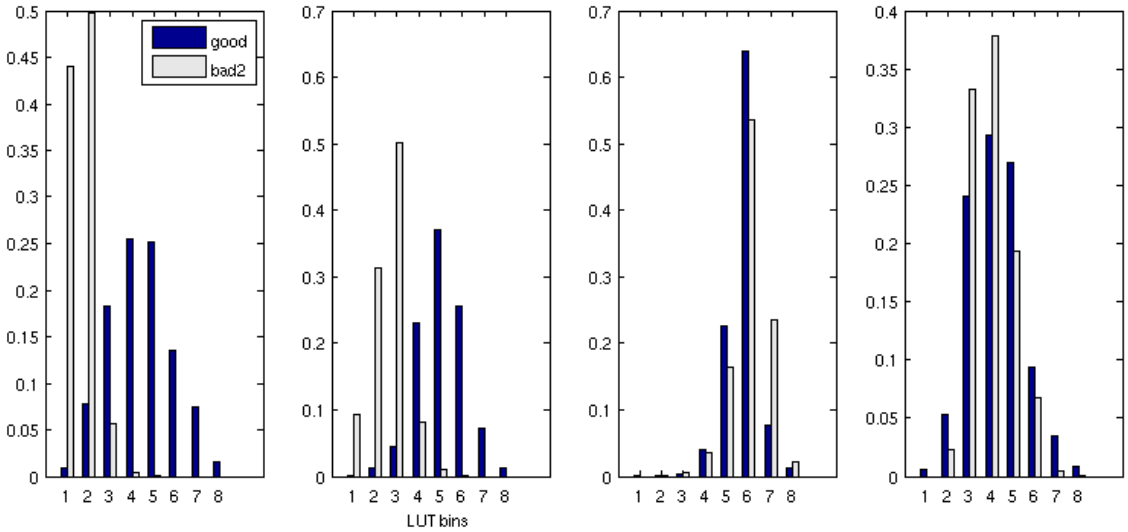


Figure 1: LUTs of the four weak classifiers of the first stage

- Image normalization. Before the actual text detection algorithm can be invoked, color images are first converted to greyscale, which are then variance-normalized, i.e. global image variance is 1. Some other schemes (or maybe no normalization at all) should be explored and compared against the current approach.
- Up to now, we have focused on roughly horizontal texts, which account for the vast majority of text strings found in a typical video document. While detecting vertical text should be fairly successful, oriented text detection should be more challenging since it would imply being capable of extracting gradients normal and parallel to text orientation and computing block statistics at reasonable speed.

ACKNOWLEDGEMENTS

This work is supported in part by the financial support of the Department of Canadian Heritage (www.pch.gc.ca) through Canadian Culture Online, the Natural Science and Engineering Research Council (NSERC) of Canada (www.nserc.ca) and the Ministère du Développement Économique de l'Innovation et de l'Exportation (MDEIE) of Gouvernement du Québec (www.mdeie.gouv.qc.ca). The authors also thank J. Dutrisac from the National Film Board (NFB) of Canada (www.nfb.ca) for providing the documentary video test data.

REFERENCES

1. T. Sato, T. Kanade, E. K. Hughes, M. A. Smith, S. Satoh, "VideoOCR: Indexing Digital News Libraries by Recognition of Superimposed Caption", ACM Journal of Multimedia Systems, Vol. 7, No. 5, pp. 385-395, 1999
2. R. Lienhart, A. Wernicke, "Localizing and Segmenting Text in Images and Videos", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No.4, pp. 256 -268, 2002
3. *Handbook of Video Databases – Design and Applications*, B. Furht and O. Marques Eds., CRC Press, 2004
4. The ICDAR 2003 Robust Reading Competitions: http://algoval.essex.ac.uk/icdar/Competitions.html
5. The ICDAR 2005 Robust Reading Competitions: http://algoval.essex.ac.uk:8080/icdar2005/index.jsp
6. X. Chen , A. L. Yuille, "Detecting and reading text in natural scenes", Proc. CVPR 2004, Vol. II, pp. 366-373, 2004
7. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", CVPR 2001, Vol. 1, pp. 511-518, 2001
8. P. Viola, M Jones. "Robust real-time object detection". Proc. of IEEE workshop on Statistical and Computational Theories of Vision, Vancouver, Canada, July 2001.
9. R. Lienhart, J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", ICIP 2002, Vol. 1, pp. 900-903, 200
10. Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm", 13th Conf. on Machine Learning, pp. 148-156, 1996
11. The National Film Board of Canada: www.nfb.ca
12. L. Gagnon, S. Foucher, V. Gouaillier, J. Brousseau, G. Boulianne, F. Osterrath, C. Chapdelaine, C. Brun, J. Dutrisac, F. St-Onge, B. Champagne, X. Lu, "MPEG-7 Audio-Visual Indexing Test-Bed for Video Retrieval", IS&T/SPIE Electronic Imaging 2004: Internet Imaging V (SPIE #5304), pp. 319-329, 2003
13. L. Gagnon, "R&D status of ERIC-7 and MADIS – Two systems for MPEG-7 indexing/search of audio-visual content", Proc. SPIE Optic-East : Multimedia Systems and Applications VIII (SPIE #6015), pp. 341-352, 2005
14. The VirtualDub project: www.virtualdub.org
15. J. Vermaak, P. Pérez, M. Gangnet, "Rapid Summarization and Browsing of Video Sequences", BMVC 2000 (www.bmvc.ac.uk/bmvc/2002)
16. B. Wu, H. Ai, C. Huang. "LUT-Based AdaBoost for Gender Classification". Proc. Audio- and Video-Based Biometric Person Authentication 2003, p. 104-110.
17. L. Dlagnekov, "Video-based Car Surveillance: License Plate, Make, and Model Recognition", M.S. Thesis, UCSD, 2005. 93 p.