九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Detection of Inserted Text in Images

Hirano, Hiromi
The University of Electro-Communications, Department of Informatics | Rakuten Co.,Ltd.

Okabe, Makoto
The University of Electro-Communications, Department of Computer Science | JST CREST

Onai, Rikio
The University of Electro-Communications, Department of Informatic

https://hdl.handle.net/2324/1430827

出版情報：MI lecture note series. 50, pp.64-67, 2013-10-21. 九州大学マス・フォア・インダストリ研究所
バージョン：
権利関係：

Detection of inserted text in images

[(1)(4)]**Hiromi HIRANO** [(2)(3)]**Makoto OKABE** [(1)]**Rikio ONAI**

[(1)]The University of Electro-Communications, Department of Informatics
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
[(2)]The University of Electro-Communications, Department of Computer Science
[(3)]JST PRESTO
[(4)]Rakuten Inc., 4-13-9 Higashishinagawa, Shinagawa-ku, Tokyo, Japan

It is possible to embed text into images using an image-editing software, and this can be used to make misleading or unfounded claims in advertisements, which do not comply with advertising standards. To monitor the large volume of advertising that now exists on the Internet, it is desirable to automatically detect and ‘read’ the text inserted into images. Here we describe a technique to determine regions of images corresponding to inserted text using the FAST algorithm, by finding corners in the image that lie along a straight line, which we term a supercorner. We then create a graph by connecting supercorners and apply cost functions describing the geometrical relations between the corners. Using a graph cut algorithm, we can separate the text from the background. Using this method in a sample set of 130 images with inserted text, we were able to detect 81% of the inserted text with a false positive rate of only 4%.

1. Introduction

A large number of merchant sites advertise their services on the Internet. Some merchant sites present images that do not conform to trading standards regulations, such as advertising goods or services along with unfounded claims. Figure 1 presents some examples, including a bogus weight-loss product (see the image on the right) and claims that do not meet the standards of the Pharmaceutical Affairs Act (see the image on the left). Internet site operators are responsible for detecting and removing such images. Text regions in an image can be classified into two categories: one kind appears on the goods directly, and the other kind is inserted into the image using some kind of image-editing software (see Figure 2). Here, we propose a method to detect text that has been inserted using image-editing software.

Figure 1. Illegal Image Examples

2. Method

Our strategy consists of two stages: first we extract the corners of the image using the FAST[6] algorithm; then we determine whether each corner is part of the inserted text region. Corners in inserted text involving Japanese or Kanji characters are likely to fall along a straight line: we refer to this distribution of corners as a supercorner. Additionally, inserted text typically has a large color contrast with the background image, and tends to have a simple color profile. Text that is included as part of the image, such as the labeling on packaging, generally does not fall on a straight line and may have a more complex color profile, making it harder to detect.

(Left) The locations of the corners in an image located using FAST. The small pink circle indicates a FAST corner. (Right) The extracted supercorners.

FIGURE 2. Extracted FAST and Supercorner

We observed that the corners in inserted text exhibited the following trends:

(1) the supercorner is located along a line of text;
(2) supercorners are typically perpendicular to each other;
(3) the corners in a supercorner are of the same color.

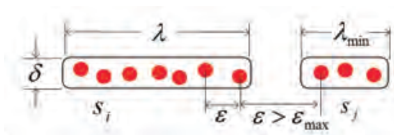
We need to assign a label of 'inserted text' or 'background' to each supercorner. To achieve this, a graph cut technique[1] can be used. We construct a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of supercorners, $\mathcal{E}$ is a set of weighted edges between any two supercorners.

2.1. **The Supercorner.**

Corners located using FAST may appear anywhere in the image, but corners that lie in a straight line are more likely to form a text region. We introduce a line segment with a length that corresponds to the set of corners. We start to detect a supercorner by selecting a FAST corner randomly. We then iteratively add the other corners to the supercorner if the following four conditions are satisfied:

(1) all FAST corners lie on a line segment within a width δ;
(2) the length (λ) of the line is longer than the threshold (λ_{min});
(3) the gap (ϵ) between two FAST corners is shorter than the given threshold(ϵ_{max});
(4) all FAST corners in a set have the same color cluster.

This is illustrated in Figure 3.



A bounding box including corners.

FIGURE 3. Extracted FAST and Supercorner

2.2. **Minimizing Energy Function using Graph Cuts**[4].

To apply the graph cut algorithm to the problem, we define an energy function on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The energy function $E(L)$ is composed of a data term $\sum \hat{D}(l_p)$ and a smoothness term $\sum \hat{V}(s_i, s_j)$. The graph is based on Potts interaction energy model as follows:

$$E(L) = \mu \sum_{p \in \mathcal{V}} \hat{D}(l_p) + \sum_{s_i, s_j \in \mathcal{V}} \hat{V}(s_i, s_j) \tag{1}$$

This energy function can be optimally solved using max-flow[7]. Where L is a label set that indicates whether each supercorner is contained in the text region, so that l_p denotes label of p-th supercorner($\in \mathcal{V}$), and μ is parameter of the data term. And $label = \{"txt", "bkg"\}$.

2.3. **Definition of Graph.**

We construct a graph as shown in Figure 4, where the n-link edges($\in \mathcal{E}$) (which span between two supercorners) have been omitted for clarity. but n-link is spanned between all two supercorners. The graph is a complete graph. In Figure 4, the long rectangle represents the supercorner. The t-link edges($\in \mathcal{E}$) span between each terminal node and supercorner. A t-link corresponds to a data term and an n-link corresponds to a smoothness term.

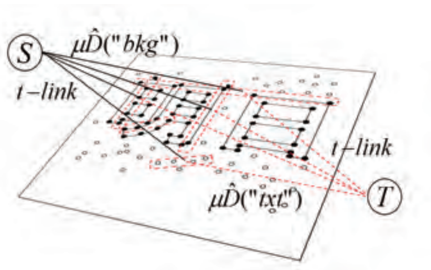


Figure 4. Supercorner and Graph Structure

2.3.1. **Cost of t-links.**

We ascribe a cost $\hat{D}(l_p)$ to an edge between a supercorner $p(\in \mathcal{V})$ and terminal $\{T, S\}$. The cost is computed using an aligned length λ(see Figure 3). The length $\bar{\lambda}_p$ of supercorner p is normalized to the dimensions of the image size (i.e., the width plus the height). The cost of edges is defined as:

$$\hat{D}(l_p) = \begin{cases} \hat{D}("txt") = -lnP(\bar{\lambda}_p | "txt") & \text{(2a)} \\ \hat{D}("bkg") = -lnP(\bar{\lambda}_p | "bkg") & \text{(2b)} \end{cases}$$

where the labels “ txt ” and “ bkg ” denote text and background regions of the image, respectively.

2.3.2. **Cost of n-links.**

If two supercorners are parallel and in close proximity, then the edge cost is low. The function $eval(s_i, s_j)$ is minimized when two supercorners are in close proximity and parallel to each other as follows:

$$A(s_i, s_j) = \exp(-\frac{eval(s_i, s_j)}{2\sigma_{ang}^2}) \tag{3}$$

If two supercorners are (nearly) orthogonal, then the edge cost is low. The function $orth(s_i, s_j)$ reflects mutually orthogonal relation between s_i and s_j, as follows:

$$B(s_i, s_j) = \exp(-\frac{orth(s_i, s_j)}{2\sigma^2_{orth}}) \tag{4}$$

If two supercorners are determined to be in same color cluster, then the edge cost is low. The function $cdist(s_i, s_j)$ describes the color distance between s_i and s_j, as follows:

$$C(s_i, s_j) = \exp(-\frac{cdist(s_i, s_j)}{2\sigma^2_{color}}) \tag{5}$$

The sum of these terms describes the smoothness of equation (1).

$$\hat{V}(s_i, s_j) = A(s_i, s_j) + B(s_i, s_j) + C(s_i, s_j) \tag{6}$$

3. **Result**

We applied the algorithm to 130 images sampled from Rakuten Mart. The method identified 81% of inserted text with a false positive rate of only 4%. Therefore, the algorithm is effective as a filter tool to detect inserted text in images. Figure 5 provides some sample images with the inserted text identified.

Figure 5. Some Results

References

[1] Yuri Boykov, Olga Veksler and Ramin Zabih: Fast Approximate Energy Minimization via Graph Cuts, *IEEE Transactions on PAMI*, 1222-1239, 2001.

[2] Chitrakala Gopalan. and Mamjula, D.:Contourlet Based Approach for Text Identification and Extraction from Heterogeneous Textual Images, *IJCSE*, 208-211, 2008.

[3] B. Gatos, I. Pratikakis, K. Kepene and S.J. Perantonis: Text Detection in Indoor/Outdoor Scene Images, *First International Workshop on Camera-based Document Analysis and Recognition*, 127-132, 2005.

[4] Vladimir Kolmogorov and Ramin Zabih:What Energy Functions Can Be Minimized via Graph Cuts?, *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.26, No.2*, 147-159, 2004.

[5] Lukas Neumann and Jiri Matas:A method for text localization and recognition in real-world images, *the 10the Asian Conference on Computer Vision*, 770-783, 2010.

[6] E.Rosten and T.Drummond: Machine learning for high-speed corner detection, *European Conference on Computer Vision*, 430-443, 2006.

[7] Kevin Wayne: Max Flow, Min Cut, *Princeton University, COS 226, Algorithms and Data Structures*, 2004.