Hypothesis Preservation Approach to Scene Text Recognition with Weighted Finite-State Transducer

Takafumi Yamazoe, Minoru Etoh, Takeshi Yoshimura, and Kousuke Tsujino
Service & Solution Development Department and Research Laboratories, NTT DOCOMO
3-6, Hikarino-oka, 239-8536 Japan
yamazoet at nttdocomo.com, {etoh, yoshimura.takeshi, tsujino} at nttdocomo.co.jp

Abstract*—This paper shows that the use of Weighted Finite-State Transducer (WFST) significantly eliminates large-scale ambiguity in scene text recognition, especially for Japanese Kanji characters. The proposed method consists of two WFSTs called WFST-OCR and WFST-Lexicon. WFST-OCR handles the multiple hypotheses caused by erroneous text location, character segmentation and character recognition processes. The following WFST-Lexicon and its convolution of WFST-OCR resolve the hypotheses. The WFSTs integrate the conventional OCR and post-processing processes into one process. The benefit from the proposed method is that all the ambiguities are held as WFST data, and solved in one integrated step; the system outputs texts that are statistically consistent with regard to segmentation possibilities and the given language model. An experimental system demonstrates practical performance in spite of the hypothesis complexity inherent in the ICDAR test set and Kanji character texts.

Keywords-component; scene text, natural scene, character recognition, text extraction, WFST, Kanji character.

I. INTRODUCTION

This research is inspired by demands for general reading systems that can locate and read text in scene images. The last decade is the era of mobile portable devices that can capture images everywhere anytime and are always connected to media servers, i.e., clouds. Natural scene optical character recognition (OCR) has become of practical importance ever the introduction of the commercial product "Evernote[1]" which opened the vision of images uploaded to servers being automatically made searchable.

Five years prior to that practical service, International Conference on Document Analysis and Recognition (ICDAR) conducted text location competitions in 2003[2] and 2005[3], with given ground truth data. The ICDAR papers emphasized connected component-based methods and region-based methods as basic analysis approaches. Such extraction and localization methods are evolving toward the integration of image preprocessing and machine learning sub-systems. Typical approaches[4][5][6] combine several sophisticated methods such as Histogram of oriented Gradient, multi-sale local binary pattern histograms, MRF-based connected component analysis, Adaboost, and SVM classifier.

The steps of pre-processing and text extraction based on character image features, one contest thread in ICDAR, are beyond the scope of this paper. Its main contribution is the practical large-scale integration of language models which cuts over-generated and erroneous hypotheses (say, 1 million) that are mainly raised by the many extracted regions, text lines, character segmentation, and OCR results. The implementation proves that the use of Weighted Finite-State Transducer (WFST) [7] can significantly eliminate large-scale ambiguity in scene text recognition. It yields also a unified framework to resolve different levels of hypotheses.

Figure 1. Natural scene examples.

There have been similar contributions [8][9][10][11] that target the use of language statistics. They mainly deal with "error correction" based on pre-defined parametric error models such as a confusion matrix among characters. The system developed here follows the concept introduced by Breuel for his system OCRopus[11]. OCRopus also adopts WFST with the aiming of resolving both errors in segmentation and OCR with a language model. In view of OCRopus, this paper describes one of the best practical implementations with extension to handle Kanji characters and a large-scale multiplicity description, while OCRopus remains as a framework.

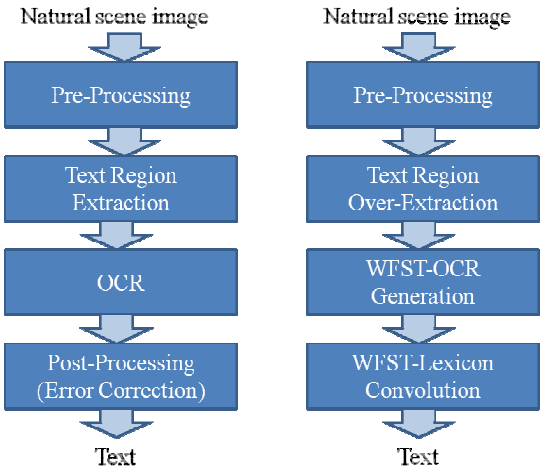
Consider the examples shown in Fig. 1. The left picture, from the ICDAR 2003 test set, contains English texts. The right picture is from the authors' collection of captured images and contains Japanese texts. As discussed in [5] [6], the extraction of Japanese texts is difficult due to the number of Kanji characters[12] (approximately 2000) and the absence of an explicit boundary separator <space>. Such a high level of ambiguity in scene Kanji character recognition motivates the use of a language model based on WFST.

Our system consists of two WFSTs: WFST-OCR and WFST-Lexicon. Fig. 2 summarizes the conceptual difference

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.80

IEEE computer society

from a conventional method, where the major four steps are functionally similar but operationally different.



(a)Conventional method, (b) Proposed method

Figure 2. Basic idea of hypothesis preservation approach.

The main differences lie in

1. The second step, i.e., proposed text region extraction step, generates *more hypotheses for better recall by sacrificing precision.*
2. WFST-OCR represents multiple hypotheses from text location, character segmentation, and character recognition. The multiplicities are solved in the final step.
3. *The WFST-Lexicon and a convolution of WFST-OCR resolve the hypotheses.* Here the conventional OCR and post-processing parts are integrated into one process.

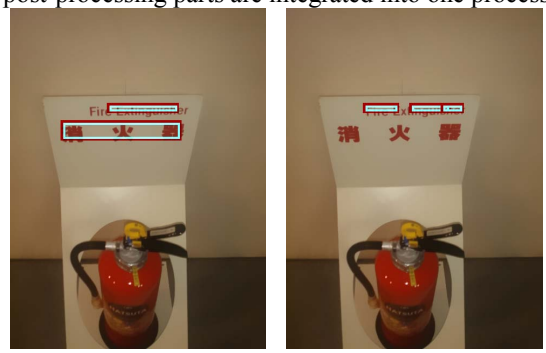


Figure 3. Output examples.

The benefit is that all the ambiguities are held as WFST data, and solved in one integrated step, so that the system outputs texts that are statistically consistent with regard to segmentation possibilities and the given language model. In Fig. 3, the proposed WFST-Lexicon convolution outputs different OCR results if given different lexicons. If the system is given only a Japanese lexicon, the system successfully identifies the Japanese text as shown in the left. Given only an English lexicon, the system reads "fire extinguisher" region while it misses the corresponding Japanese region.

The contribution of this paper is to use WFST intensively for reasoning from millions of hypotheses. This key concept will appear with the system description in the following sections.

II. PRE-PROCESSES BEFFORE WFST

Although the text region extraction and character recognition are beyond the scope of this paper, it is worthwhile to understand how to generate many hypotheses.

A. Candidate Region Extraction

The implemented method follows Zhu's work [13] and borrows their non-linear Niblack method and their basic connected component (CC)-based methods. The non-linear Niblack method decomposes an input image into binary images which are CC-regions. The proposed system analyses the CC-regions using the measures of shape regularity, i.e. contour roughness, in-holes, compactness, and occupancy ratio. Without tuning the parameters, unlike Zhu's work with Adaboost, the system outputs candidate regions for text line extraction. Note that the Niblack decomposition method has a scale parameter which adjusts locality sensitivity. By setting different scales, the system generates numerous multiple hypotheses.

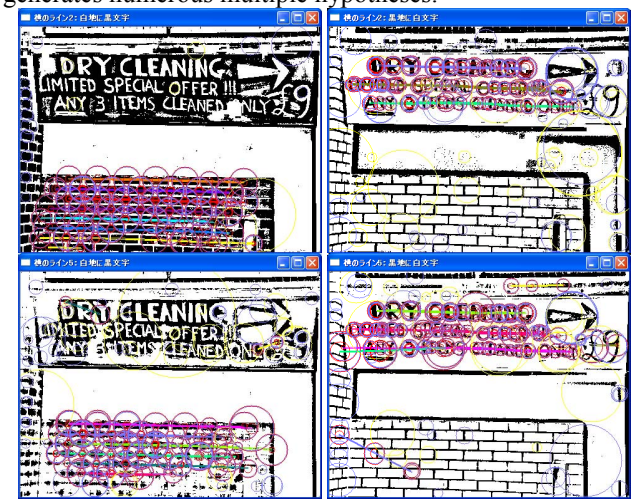


Figure 4. Character candidates at different scales.

B. Text Line Extraction from Candidate Regions

The system adopts the combination of the following two methods. Retornaz[14] proposed a region extraction method using morphological operation over extracted CC regions. Messelodi[15] proposed a method that detects the spatial position and the skew of text lines. Their combination allows the system to obtain text boundary rectangles which are calibrated through affine transformations from scenes.

Although the system was designed to read Japanese texts from natural scenes, we continue to use the "DRY CLEANING" example (in Fig. 1) for the sake of the English readers. Fig. 4 shows extracted text regions derived at different non-linear Niblack scales, where blue-yellow-green colored lines represent text lines. On the right, at coarser scales, text lines are extracted reasonably well, while on the left, at finer scales, brick regions are wrongly extracted as text region hypotheses. If the system adopts the full-implementation of Zhu□s work,

we may have a better result. That is not strongly demanded. The essential requirement is high recall coverage by over-extraction of text regions. Thus, all the results in Fig. 4 will be used in the following step.

C. Character Segmentation from Text Region

Character segmentation is realized as a projection analysis of CC-regions for each text boundary rectangles. The method used here is not new as it was already surveyed in [16]. The vertical projection of a text region consists of a simple running count of the black pixels in each column. It can serve for the detection of white space between successive letters. Fig. 5 depicts the DRY CLEANING example. Note that Japanese texts may be written vertically as well as horizontally. In the former case, the system performs a horizontal projection.

Figure 5. Vertical projection analysis for character segmentation.

The systems adopt a more sophisticated analysis based on peak-to-valley functions, so that different thresholds can derive multiple hypotheses, i.e., the over-extraction of possible character segments.

The over-segmentation strategy is inspired by Saidane's work[17]. Saidane's paper describes an optimized text region recognition scheme with a weighted directed acyclic graph, where nodes represent character segmentation positions and edges represent costs. Costs are derived from character recognition scores, position consistency of neighboring characters, and separation distance. The optimal path (i.e., OCR result) is obtained as the lowest cost traverse over the graph in terms of character segmentation and character-wise recognition confidence. The proposed method takes a similar approach in that sense. The major advancement is the integration of language statistics with two WFSTs as explained in the next section.

III. TWO WFSTS

Pattern recognition in general pursues reverse problem solving to identify the pattern generation process from what follows from the origin (or the ground truth if any). To find the most reasonable source model, we often take a hypothesis-preservation approach, where extensive different assumptions are considered and verified as to whether they consistently predict the observation. Hypotheses may have likelihood values (confidence scores) and constraints from other hypotheses. The issue is how to find *the most computationally-efficient method to maintain the numerous hypotheses.* The approach taken here is to use a graph representation which embodies hypotheses with those likelihoods and constraints. We will see how WFST effectively works in the following subsections.

A. WFST in general

A *weighted finite-state transducer* (WFST) is a finite automaton for which each transition has an input label, an output label, and a weight. The system adopts OpenFST[7], and all notations of WFST follow those of the document.

A. ***weighted finite-state transducer***

$$\boldsymbol{T = (\mathcal{A}, \mathcal{B}, Q, I, F, E, \lambda, \rho)}$$

over a semiring $\mathbb{K}$ is specified by a finite input alphabet $\mathcal{A}$, a finite output alphabet $\mathcal{B}$, a finite set of states Q, a set of initial states $I \subseteq Q$, a set of final states $F \subseteq Q$, a finite set of transitions $E \subseteq Q \times (\mathcal{A} \cup \{\varepsilon\}) \times (\mathcal{B} \cup \{\varepsilon\}) \times \mathbb{K} \times Q$, an initial state weight assignment $\lambda: I \to \mathbb{K}$, a final state weight assignment $\rho: F \to \mathbb{K}$, and empty string label $\{\varepsilon\}$. (1)

Given transition $e \in E$, *let* $p[e]$ denote its origin or previous state, $n[e]$ its next state, $i[e]$ its input label, $o[e]$ its output label, and $w[e]$ its weight. Thus e is attributed as

$$e = (p[e], i[e], o[e], w[e], n[e]) \in E.$$

over semiring $\mathbb{K}$. (2).

A weight in a WFST represents the cost of taking that transition, where optimization for finding plausible outputs is performed over weights. Here we use the probability semiring for$\mathbb{K}$ to combine probabilities. For implementation simplicity, however, the proposed system uses real weights normalized to [0,1] via negative-log mapping from probabilities. E represents weighted hypotheses from predefined knowledge and observation, and it, in fact, characterizes a WFST. Here is a useful WFST composition operation from two or more WFSTs. For the specific implementation addressed here,

$$T = T_{OCR} \circ T_{LEX}, \text{ where}$$

$$\mathrm{T_{OCR}} = (A_{OCR}, B_{OCR}, Q_{OCR}, I_{OCR}, F_{OCR}, E_{OCR}, \lambda_{OCR}, \rho_{OCR}), \textit{ and}$$
$$T_{LEX} = (A_{LEX}, B_{LEX}, Q_{LEX}, I_{LEX}, F_{LEX}, E_{LEX}, \lambda_{LEX}, \rho_{LEX}). \quad (3)$$

B. WFST-OCR

The system creates T_{OCR} for each text line according to the following assignments:

A_{OCR} : {Japanese and English characters}, $|A_{OCR}| \approx 2000$,
B_{OCR} : $= A_{OCR}$
Q_{OCR} : { left positions of each character segment},
I_{OCR} : leftmost position of the text line,
F_{OCR} : rightmost position of the text line,
λ_{OCR} *and,* ρ_{OCR} : set to zero, and
E_{OCR} : {e} with

$$\mathrm{w(e)} = \frac{score(i(e);p(e),n(e))}{\sum score()},$$ *where score() gives the (1,..,N)th ranking result of an OCR engine for the character segment (p(e),n(e)) unless* $i(e) = \{\varepsilon\}$. (4)

$$w(e) = 1, if\ i(e) = \{\varepsilon\}. \quad (5)$$

Eq. 4 gives transition costs as determined from OCR results, given a character segment. In the current implementation, the system uses four off-the-shelf OCR engines including NHOCR[18]. N is empirically set to 20 considering the Japanese character set size, say 2000. The number of transitions is typically around 40-60, and 80 at most when

the OCR engines return non-overlapped character lists from the first to 20th character. Empty string {ε} is used as a *penalty to skip the character segment.*

Let us go back to DRY CLEANING example, which contains approximately 1 million hypotheses as one of the most complex examples. Here we will see how the system creates T_{OCR} from the hypothesis as shown in Figs. 4 and 5. The *initial state* is label 0. The *final state* is 10. Weights are not shown in Fig. 6 for graphic simplicity. Empty {ε} transitions are added on the horizontal center line from state 0 to state 10.

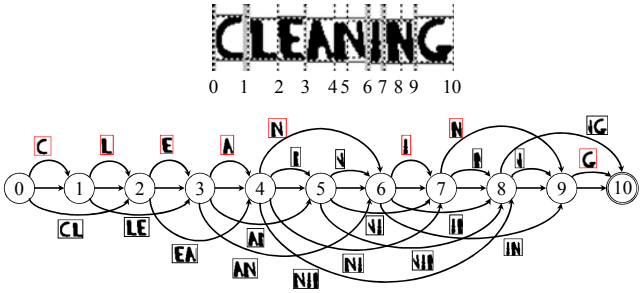


Figure 6. WFST-OCR example.

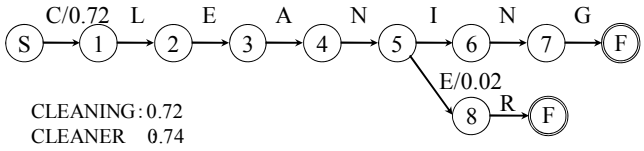


Figure 7. WFST-Lexicon example.

C. WFST-Lexicon

This WFST gives the knowledge about a language model. The system creates T_{LEX} from each lexicon according to the following assignments:

A_{LEX} : {Japanese and English characters appearing in the given lexicon. In the experiment, the size is about 1 million words},

$B_{LEX} := A_{LEX}$,

Q_{LEX} : { logical character segments in each word},

I_{LEX} : One pre-defined initial state

F_{LEX} : One pre-defined final state,

λ_{LEX} *and,* ρ_{LEX} : set to zero, and

E_{LEX} : {e} with

$$w(e) = \alpha\left(1 - \frac{\sqrt{WordLength}}{\sqrt{MaxWordLength}}\right) + \beta\left(1 - \frac{ln\,(WordFrequency)}{ln\,(MaxWordFrequency)}\right),$$

if ($i(e)/o(e)$) spans the first character of each word, else $w(e)$ is set to zero, where $\alpha + \beta = 1$. (6)

Eq. 6 represents the heuristic of *longer and more frequent words are better for text extraction*. We use this heuristic $\alpha = 1, \beta = 0$ to simplify parameter dependency in the experiments.

Fig. 7 shows an example of a WFST-Lexicon that contains "CLEANING" and "CLEANER," which are already minimized into a compact representation.

Having one WFST-Lexicon and a set of WFST-OCRs (one for each text line), the system performs the convolution as described in Eq. 3 and yields the output shown in Fig. 8. The composed WFST transduces the states from 'S' to 'F', and outputs 'CLEANING' as the most plausible result from the multiple hypotheses including Japanese character candidates. The image where recognition results are plotted is shown in Fig. 9.

IV. Experiments

Fig.10 shows the result of the first experiment which evaluates the precision and recall of *word correctness* over the ICDAR test set including the DRY CLEANING example. ICDAR'03-'05 were conducted to evaluate *segmentation correctness* and its top score was (0.62, 0.67)[3]. It is hard to compare the different criteria, but we can conclude that the obtained word recognition score (0.58, 0.54) is satisfactory, since word recognition has more error-prone steps (including text line extraction and OCR) beyond text segmentation.

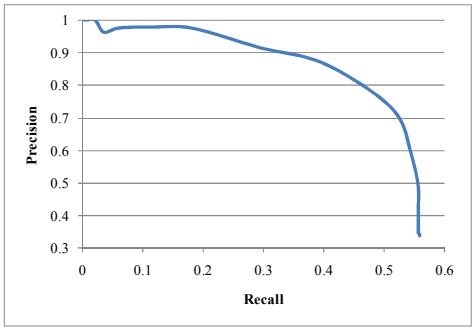


Figure 10. Precision-recall curve from ICDAR test set.

The next experiment uses Japanese Kanji characters to evaluate the complexity of the WFST approach. Fig. 11 shows examples, where the system achieved precision/recall values of (0.59,0.68). Table 1 summarizes the size of T_{LEX}, T_{OCR}, *and their composit* T . A Japanese lexicon T_{LEX} is constant for all images, but does T_{OCR} vary with input image complexity.

TABLE I. Number of WFST states and transitions (see Eq.3).

	Image (a)	Image (b)	Image (c)	Image (d)
$\#Q_{LEX}$	10,264,288	10,264,288	10,264,288	10,264,288
$\#E_{LEX}$	11,252,274	11,252,274	11,252,274	11,252,274
$\#Q_{OCR}$	1,706	988	814	2562
$\#E_{OCR}$	185,797	39,478	69,438	185,400
$\#Q$	9,105	7,877	7,417	10,548
$\#E$	13,186	11,960	11,501	14,627
#texts	2	1	1	8

The system took 86 seconds, on average, to perform all steps using a 2.8GHz Core 2 processor, 3.2GB RAM (32bit Windows XP single-thread application). There is a natural concern about the complexity of hypothesis preservation. From the above result, we can conclude that WFST

operation on realistic targets is practical in terms of search efficiency and precision in spite of the large number of states and transitions.

V. CONCLUSION

We proposed a WFST-based approach to the resolution of large-scale ambiguity with multiplicities in text extraction, text line segmentation, character segmentation, and character recognition. The key advance is a very efficient implementation of hypothesis preservation. A system was implemented that performed optimized WFST operations and tests showed that it yielded statistically correct outputs with a reasonable level of computing complexity. This is the first report, to the authors' knowledge, of large-scale integration of text-extraction in the image domain and language modeling. Further study includes integrating advanced language models into the current system via composition operations, and to further optimize it by on-the-fly composition.

ACKNOWLEDGMENT

The authors appreciate Prof. Koichi Kise and Prof. Seiichi Uchida's advice on natural scene character recognition. The authors also would like to thank Dr. Hori and Dr. Tsukada of NTT CS labs for their advice on WFST implementation.

REFERENCES

[1] Welcome to your notable world: Evernote Corporation, http://www.evernote.com/

[2] L. P. Sosa, S.M. Lucas, A. Panoretos, L. Sosa, A. Tang, S. Wong, and R. Youg, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J.Zhu, W.Ou, C. Wolf, J-M Jolion, L. Tooran, M. Worring, X. Lin, ICDAR 2003 Robust Reading Competitions, IJDAR, vol.7, pp.105-122, 2005.

[3] S.M. Lucas, ICDAR 2005 text locating competition, Proc. ICDAR '05, pp.80-84, 2005.

[4] Y. Pan, X. Hou, C. Liu, A robust system to detect and localize texts in natural scene images, Proc. DAS'08, pp.35-42, 2008.

[5] Y. Kusachi, A. Suzuki, N. Ito, and K., Arakawa, Pattern Recognition, Kanji recognition in scene images without detection of text fields-robust against variation of viewpoint, contrast, and background texture, vol. 1, pp. 457-460, 2004.

[6] L. Xu, H. Nagayoshi, and H.Sako, Kanji character detection from complex real scene images based on character properties, Proc. 8th Int. Workshop on Document Analysis Systems, pp.278-285, 2008.

[7] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, OpenFst: a general and efficient weighted finite-state transducer library, Proc. of the CIAA'07, pp. 11-23, 2007.

[8] R. Beaufort, and C. Mancas-Thillou, A weighted finite-state framework for correcting errors in natural scene OCR, Proc. ICDAR '07, pp.889-893, 2007.

[9] G. Neubig, S. Mori, and T. Kawahara, Japanese Character Error Correction using WFSTs, Proc. NLP2009, pp.332-335, 2009 (in Japanese)

[10] R. Llobet , J. Cerdan-Navarro, J. Perez-Cortes, and J.Arlandis, OCR Post-processing Using Weighted Finite-State Transducers, Proc. ICPR'10, pp. 2021-2024, 2010.

[11] T. Breuel, The OCRopus open source OCR system, Proc. IS&T/SPIE 20th Annual Symp., 2008.

[12] Kanji, http://en.wikipedia.org/wiki/Kanji

[13] K. Zhu, F. Qi, R. Jiang, L. Xu, Using Adaboost to Detect and Segment Characters from Natural Scenes, Proc. CBDAR 2005, pp.52-59.

[14] T. Retornaz and B. Marcotegui, Scene text localization based on the ultimate opening, Proc. Int. Symposium on Mathematical Morphology, pp.177-188, 2007.

[15] S. Messelodi and C. M. Modena, Automatic identification and skew estimation of text linesin real scene images, Pattern Recognition, vol. 32, pp 791-810, 1999.

[16] R. Casey and E. Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. on PAMI, vol.18, no.7, pp.690-706, 1996.

[17] Z. Saidane, C. Garcia, and J.L. Dugelay, The image text recognition graph (iTRG), Proc. ICME 2009, pp.266-269, 2009.

[18] NHOCR, http://code.google.com/p/nhocr/. [Online]. Available: http://code.google.com/p/nhocr/

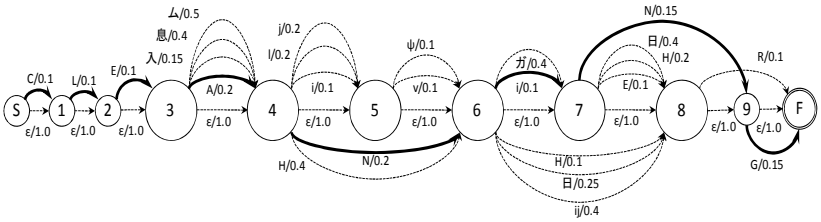
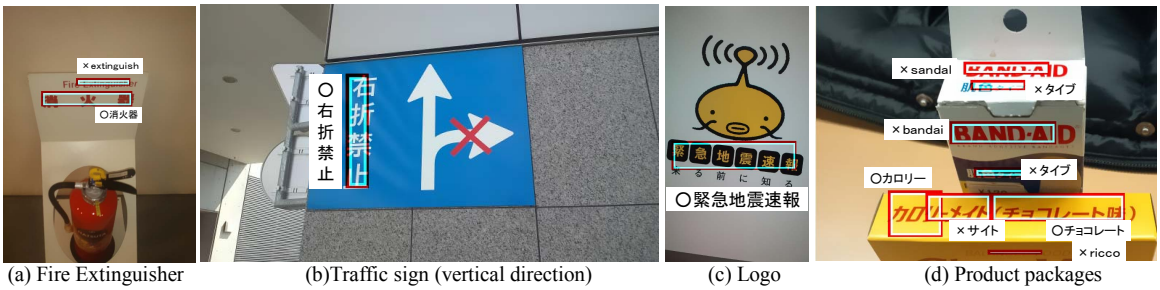


Figure 8. A composition result obtained from the WFST-OCR and WFST-LEX in Figures 6 and 7 (bold lines).

Figure 9. Recognition results of "DRY CLEANING" example



(a) Fire Extinguisher (b)Traffic sign (vertical direction) (c) Logo (d) Product packages

Figure 11. Examples for Japanese text recognition.