H-010

Preliminary Experiment on Khmer OCR

Vanna KRUY Wataru KAMEYAMA
vanna@ruri.waseda.jp wataru@waseda.jp
Graduate School of Global Information and Telecommunication Studies, WASEDA University

1. INTRODUCTION

OCR technology has already matured for major languages such as Japanese and English. However, there is no reliable OCR system for Khmer Language. This is largely due to the lack of Khmer OCR research efforts and the complex nature of Khmer characters. There are two main issues. First, some characters in a word connect each other while parts of a single character are often disconnected. Second, Khmer word typing is not always in the same order as visually seen. Thus proper segmentation and character ordering are needed.

In this paper, we tackle these issues using Connected Component Analysis (CCA) and Word Semantic (WS). We present our early results, discussion, and conclusion.

2. RELATED WORKS

There are only few research efforts on Khmer OCR. Two of which are Khmer OCR using Wavelet Descriptors and Khmer Printed Optical Recognition Using Lagendre Moment both by Chey et al which reaches 92.99% accuracy [1] and 92% accuracy [2], respectively. Ing L.I. experimented Khmer OCR for Limon R1 font, size 22 using Discrete Cosine Transform and Hidden Markov Model which reaches 98.88% accuracy [3]. [1] & [2] experimented with several Khmer fonts, but segmentation and character ordering after detection was outside the scope. [3] did character ordering but was limited to fixed font and fixed size.

3. PROPOSED METHOD

We would like the system to be able to do segmentation as well as character ordering. We have used CCA for segmentation and WS for character ordering. We have chosen Scale Invariant Feature Transform (SIFT) [4] as the character feature since it is invariant to scale, translation, rotation, and local geometric distortion.

3.1 SYSTEM OVERVIEW

The system is divided into two main modules—Training Module and Recognition Module. Each module depends on other sub-modules. Fig.1 shows all the modules in the system. The upper layers depend on the lower layers. Training Module is used to annotate WS. It relies on WS annotation, Vertical Component (VC) extraction, CCA, SIFT extraction, WS database and Annotated CCs database modules. Recognition Module is used to test the system. It depends on Connected Component Recognition (CCR), Character Ordering, CCA, SIFT extraction and Annotated CCs database modules. Section 3.2 and 3.3 give detailed information about these processes.

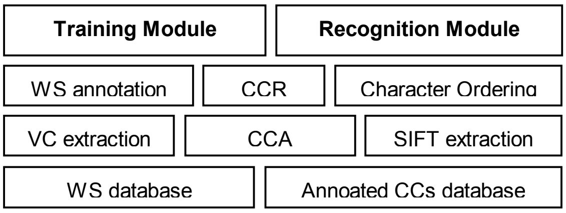

Fig.1 System overview

3.2 WORD SEMANTIC ANNOTATION

Since Khmer word writing is not always in the same order as visually seen, ordering after detection is needed. We propose WS for ordering of the detected Connected Components (CCs). We extract words from the Khmer word corpus [5] and generated WS by converting the word string into image, and extract VCs which are components separated by vertical space. Then for each VC, we extract CCs. We annotate the extracted CCs to form the WS. Fig.2 shows the flowchart of WS annotation process. Fig.3 gives an example.

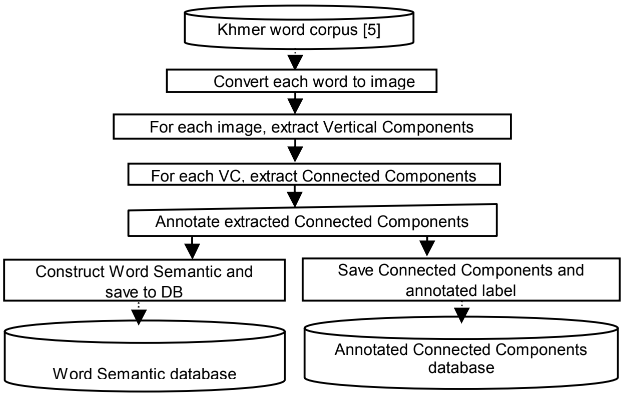
Khmer word corpus [5]
Convert each word to image
For each image, extract Vertical Components
For each VC, extract Connected Components
Annotate extracted Connected Components
Construct Word Semantic and save to DB
Save Connected Components and annotated label
Word Semantic database
Annotated Connected Components database

Fig1. Word Semantic annotation flowchart

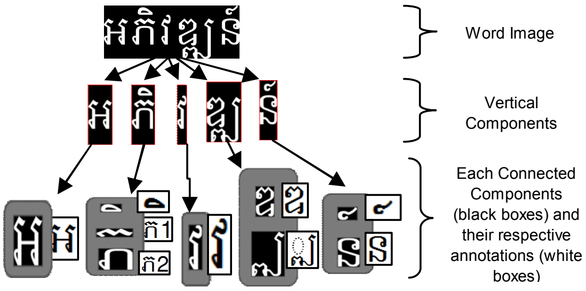
Word Image
Vertical Components
Each Connected Components (black boxes) and their respective annotations (white boxes)
ក1
ក2

Fig2. An example of Word Semantic of the word អភិវឌ្ឍន៍ (Development)

3.3 RECOGNITION PROCESS

We have used CCA to segment CCs. To recognize CCs, we use SIFT [4] as the feature. Finally we use WS to do character ordering by comparing words' score. Section 3.3.1 and 3.3.2 describes the process of comparing CCs and the process of comparing words, respectively. Fig.5 explains the recognition process. The word's VCs' size and the VC's CCs size filter out most words. Finally, the word with the highest word score will be chosen as the target word.

3.3.1 SIFT SCORE

To compare the extracted CC with CCs in the database, we calculate the SIFT score with the following formula. SIFT parameters are shown in Fig.4.

$$score = \frac{2mp}{i_1kp + i_2kp} \qquad \textbf{(1)}$$

mp: number of matched key points
$i_1$kp: number of key points in component 1
$i_2$kp: number of key points in component 2

```
UP_SCALE = true                              STEP_PER_SCALE = 6
MIN_SIZE = 20                                MAX_SIZE = 1024
FEATURE_DESCRIPTOR_SIZE = 4                  INITIAL_SIGMA = 1.6
FEATURE_DESCRIPTOR_ORIENTATION_BINS = 8
```

Fig.4 SIFT parameters

3.3.2 WORD SCORE

To compare which word is the most likely candidate, we calculate the word score with the following formula.

$$Word\ score = \frac{\sum CC'\ SIFT\ score}{Number\ of\ CCs} \qquad \textbf{(2)}$$

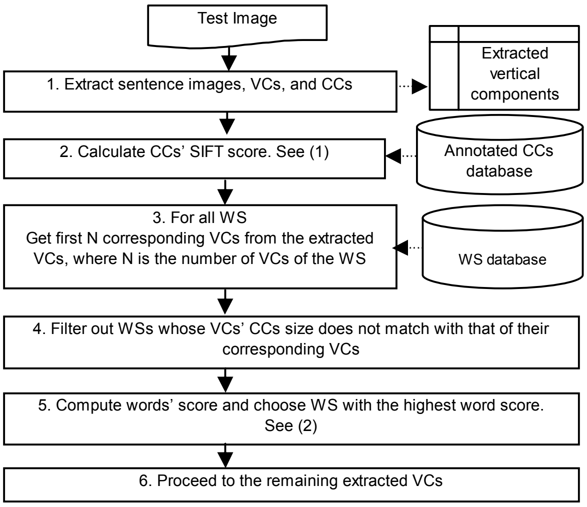
Test Image
1. Extract sentence images, VCs, and CCs
Extracted vertical components
2. Calculate CCs' SIFT score. See (1)
Annotated CCs database
3. For all WS
Get first N corresponding VCs from the extracted VCs, where N is the number of VCs of the WS
WS database
4. Filter out WSs whose VCs' CCs size does not match with that of their corresponding VCs
5. Compute words' score and choose WS with the highest word score. See (2)
6. Proceed to the remaining extracted VCs

Fig.5 Recognition flowchart

4. EXPERIMENT SETTING AND RESULT

We have tested 2 documents taken from newspaper with 1104 words using Khmer OS System font. The WS database contains 1396 annotated words and CC database contains 277 CCs.

Table.1 Document Test Result

Precision	0.77
Recall	0.73
F-Measure	0.75

To test how efficient SIFT feature is, we conducted another experiment. We tested the recognition of Annotated CCs and count number of hits.

Table.2 SIFT Test Result

# CC	Number of hits	Percentage
277	267	96%

4.1 DISCUSSION

The proposed system accuracy is not high. Most detections give false result when a short word is a sub string of another longer word as pointed out in Fig.6. The word score often gives higher score to shorter words. This will corrupt the rest of the longer word. However, if we regard all characters as words and annotate them in the WS database, this may not be the issue anymore. This will be done in the future work.

គុយ(គ១:0.25+គ២:0.3389830508474576+◌ុ:0.15384615384615385+
យ:0.24079320113314448)/4=**0.245905601456689**,
គុយទាវ(គ១:0.25+គ២:0.3389830508474576+◌ុ:0.15384615384615385
+យ:0.24079320113314448+ទា:0.2755681818181818
+វ:0.21052631578947367)/6=**0.2449528172390686**

Fig.6 A shorter word គុយ is the substring of a longer word គុយទាវ. Their word scores are in bold.

5. CONCLUSION

The system is not robust to noise. The recognition process assumes perfect segmentation (comparing size of VCs and size of CCs). In the future experiment, we consider using top N CCs detected by SIFT which eliminate the CCs size dependency.

From SIFT experiment, we get high hit rate which justifies itself as a good feature. However, we need more feature to get higher accuracy.

REFERENCES

[1] C. CHEY, P. KUMHOM, and K. CHAMNONGTHAI. *Khmer Printed Character Recognition by using Wavelet Descriptors*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems Vol.14 NO.3. (2006) 337-350. Word Scientific Publishing Company.

[2] C. CHEY, P. KUMHOM, and K. CHAMNONGTHAI. *Khmer Printed Characters Recognition Using Lagendre Moment Descriptor*. Department Electronic and Telecommunication, King Mongkut's University of Technology Thomburi.

[3 I. LENG IENG, *Khmer OCR for Limon R1 Size 22 Report*, PAN Localization Cambodia (PLC) of IDRC. (2009). URL:http://www.panl10n.net/english/ Outputs%20 Phase%202/CCs/Cambodia/MoEYS/Papers/2009/KhmerOCRLimonR122.pdf visited: July 14th, 2010.

[4] D. G. LOWE, *Distinctive Image Features from Sclae-Invariant Keypoints*, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[5] C. VAN & W. KAMEYAMA, *Building a Khmer Text Corpus*, 72nd IPSJ National Convention, 4W-1.