Training an End-to-end Model for Offline Handwritten Japanese Text Recognition by Generated Synthetic Patterns

Nam-Tuan Ly, Cuong-Tuan Nguyen, Masaki Nakagawa
Tokyo University of Agriculture and Technology
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Proposed method</th>
<th>Experiments</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Outline**

1. Introduction
2. Proposed method
3. Experiments
4. Conclusion & Future Work
Outline

1. Introduction
2. Proposed method
3. Experiments
4. Conclusion & Future Work
Background

- Offline handwritten Japanese text recognition:
  - Big challenging problem.
  - Receiving much attention from numerous business sectors.

- The existing systems are still far from perfection:
  - Thousands of classes (4,438 classes) and various characters: Kana, Kanji, numerals and alphabet characters.
  - Diversity of writing styles.
  - Multiple-touches between characters.
  - Noises...

- Handwritten Japanese text database, TUAT Kondate:
  - 13,685 text line images.
  - Covers ~1200 categories characters
  → Data is not enough.
Related Work (1/3)

- Segmentation based methods (*).
  - Pre-segment text lines into characters.
  - Individually recognize each character by MQDF or CNN.
  - Finally recognize text lines while integrating linguistic and geometric contexts.
  - They were dominant for Japanese.

→ Problems:
  - Pre-segmentation of text lines is quite costly.
  - Early errors have domino-effect on the performance.

Segmentation-free methods: avoiding segmentation errors.

- Traditional segmentation-free methods are HMM-based (*).
  - Deep Neural Nets have proven superior to HMM.
- Based on Deep Neural Nets and CTC, many segmentation-free methods have been proposed and proven to be very powerful.
  - Graves et al. (2009) combined BLSTM and CTC to build a Connectionist System.
  - R. Messina et al. (2015) combined MDLSTM-RNN and CTC to build an end-to-end trainable model.

We propose an end-to-end model of Deep Convolutional Recurrent Network (DCRN) for offline handwritten Japanese text recognition.

(*) Su et al., 2009, Suryani et al. 2016.
Related Work(3/3)

- Deep Neural Networks typically require a large set of data for training.
  - Handwritten Japanese Text dataset, TUAT Kondate: data is not enough.
    → apply data argumentation.

- Many data argumentation methods have been proposed by modifying the original patterns:
  - Affine transformations, nonlinear combinations...
    → However, such methods just modify the original patterns, can’t gain the real text line images.

- We propose a synthetic pattern generation method.
Outline

1. Introduction
2. Proposed method
3. Experiments
4. Conclusion & Future Work
Deep Convolutional Recurrent Network (DCRN) consists of three components.

- **Convolutional Feature Extractor.**
  - Using a standard CNN network (FC and Softmax layers are removed).
  - Extract a feature sequence from a text line image.

- **Recurrent layers.**
  - Employing a Bidirectional LSTM.
  - Predict pre-frames from a feature sequence.

- **Transcription layer.**
  - Using CTC – decoder.
  - Convert the pre-frame predictions into a label sequence.
Deep Convolutional Recurrent Network (2/3)

Previous works(*): overlapped sliding windows DCRN model

Convolutional Feature Extractor.
- Pretrain CNN by isolated character patterns.
- Using the pretrained CNN and overlapped sliding windows to extract a feature sequence.

(*) Nam-Tuan Ly et al. 2017
Deep Convolutional Recurrent Network (2/3)

Previous works(*): overlapped sliding windows DCRN model

Convolutional Feature Extractor.

- **Pretrain** CNN by isolated character patterns.
- Using the pretrained CNN and **overlapped sliding windows** to extract a feature sequence.
- **Two configurations:**
  - Remove just Softmax layer from CNN.
    - \( \rightarrow \text{DCRN}_o-s \)
  - Remove both FC and Softmax layers from CNN.
    - \( \rightarrow \text{DCRN}_o-f&s \)

(*) Nam-Tuan Ly et al. 2017
This works: End-to-end Model

- Remove softmax and FC layers from CNN.
- Do not use sliding windows.
- Do not pretrain CNN.
- End-to-end training System.

→ End-to-End
**Synthetic pattern generation method.**

- Sentences in corpora and handwritten character pattern database (HCP).
- Local and global elastic distortion model.

- Following 6 steps:
  1. Get a sentence (S) from a corpus.
  2. Randomly choose a writer (A) from the HCP.
  3. For each character of the sentence (S), a handwritten image of this character is randomly chosen from the writer (A).
  4. Apply a local elastic distortion to each handwritten character pattern in the step 3.
  5. Synthesize a handwritten text line image from the sentence (S) and handwritten character images in the step 4 with random spacing.
  6. Apply a global elastic distortion to the generated synthetic text line image.
**Synthetic Data Generation (2/3)**

### Local Elastic Distortion
- Performs affine transformations on each handwritten character image.
- Employs the shearing, rotation, scaling, translation transformations.

![Local elastic distortion graphs]

### Global Elastic Distortion
- Performs affine transformations on a whole text line image.
- Employs the rotation and scaling transformations.

![Global elastic distortion graphs]
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Proposed method</th>
<th>Experiments</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Synthetic Data Generation (3/3)**

**Synthetic Handwritten Text Line Dataset (SHTL)**

- Handwritten Japanese character pattern DBs, Nakayosi and Kuchibue.
- Nikkei newspaper corpus (1.1 million sentences) and Asahi newspaper corpus (1.14 million sentences).
  - Randomly choose 30,000 sentences which contain less than 30 characters from each corpus.
  - → make sure that the end-to-end model can be trainable by SHTL.

```
この日の先行取得の要請で、計画が本格的に始まった。

そのための予算に新年度は四億五千三百万円を盛り込む方針。

毎回その結果を学校のパソコンで処理して、校内の偏差値を出す。
```

Samples of generated synthetic data.
Outline

1. Introduction
2. Proposed method
3. Experiments
4. Conclusion & Future Work
Datasets (1/2)

TUAT Kondate database

- A database of handwritten text patterns mixed with figures, tables, maps, diagrams and so on (originally online but converted to offline).
  - About 13,685 of text line patterns (from 100 Japanese writers).

Information on Kondate database

<table>
<thead>
<tr>
<th>Kondate</th>
<th>Train set</th>
<th>Valid set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of writers</td>
<td>84</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Number of samples</td>
<td>11,487</td>
<td>800</td>
<td>1,398</td>
</tr>
</tbody>
</table>

Kondate sample patterns.
### Datasets (2/2)

**Handwritten Japanese character pattern database.**
- Nakayosi & Kuchibue (originally online but converted to offline)
  - Used for generating SHTL.

<table>
<thead>
<tr>
<th></th>
<th>Nakayosi</th>
<th>Kuchibue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writers</td>
<td>163</td>
<td>120</td>
</tr>
<tr>
<td>Classes</td>
<td>4438</td>
<td>3345</td>
</tr>
<tr>
<td>Samples</td>
<td>1,695,689</td>
<td>1,435,440</td>
</tr>
</tbody>
</table>

**Synthetic Handwritten Text Line Dataset (SHTL)**
- 60,000 text line images.
  - used for training the end-to-end model.
Implementation Details

End-to-end DCRN

- Convolutional Feature Extractor: CNN network.
  - 4 cascades of 2 convolutional and pooling layers.
  - Batch normalization, Leaky ReLu.

- Recurrent layers: Deep BLSTM.
  - Three layers of 128 nodes each.
  - Dropout (dropout rate = 0.8).
  - FC and Softmax layers.

- Training by 2 datasets:
  - TUAT Kondate → End-to-End
  - TUAT Kondate + SHTL → End-to-End_SHTL
Evaluation Results

- **Label Error Rate (LER)**

\[
LER(h, S') = \frac{1}{Z} \sum_{(x,z) \in S'} ED(h(x), z)
\]

- **Sequence Error Rate (SER):**

\[
SER(h, S') = \frac{100}{|S'|} \sum_{(x,z) \in S'} \begin{cases} 
0 & \text{if } h(x) = z \\
1 & \text{otherwise}
\end{cases}
\]

- Where \(Z\) is the total number of target labels in \(S'\)
- \(\text{ED}(p, q)\) is the edit distance between two sequences \(p\) and \(q\).
## Experiment Results

Label Error Rate (LER) and Sequence Error Rate (SER) on TUAT Kondate dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Label Error Rate(%)</th>
<th>Sequence Error Rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Valid set</td>
<td>Test set</td>
</tr>
<tr>
<td>DCRN_o-f&amp;s</td>
<td>11.74</td>
<td>6.95</td>
</tr>
<tr>
<td>DCRN_o-s</td>
<td>11.01</td>
<td>6.44</td>
</tr>
<tr>
<td>End-to-End</td>
<td>5.22</td>
<td>3.65</td>
</tr>
<tr>
<td>End-to-End_SHTL</td>
<td>3.62</td>
<td>1.95</td>
</tr>
</tbody>
</table>

- The end-to-end DCRN models substantially work better than the overlapped sliding windows DCRN model.
- Recognition accuracy is improved by using the SHTL dataset for training the end-to-end model.
# Experiment Results

Label Error Rate and Sequence Error Rate when combined with the language model.

<table>
<thead>
<tr>
<th>Model</th>
<th>LER(%)</th>
<th>SER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation based [1]</td>
<td>11.2</td>
<td>48.53</td>
</tr>
<tr>
<td>DCRN_o-f&amp;s</td>
<td>6.68</td>
<td>26.97</td>
</tr>
<tr>
<td>DCRN_o-s</td>
<td>6.10</td>
<td>24.39</td>
</tr>
<tr>
<td>End-to-End</td>
<td>3.52</td>
<td>16.67</td>
</tr>
<tr>
<td>End-to-End_SHTL</td>
<td>1.87</td>
<td>13.81</td>
</tr>
</tbody>
</table>

- The DCRN models are superior to the segmentation based method.
- Recognition accuracy is further improved when the linguistic context is integrated.

Correctly recognized samples

Correctly recognized samples by End-to-End_SHTL.
Some mispredicted samples by End-to-End_SHTL.
Outline

1. Introduction
2. Proposed method
3. Experiments
4. Conclusion & Future Work
Conclusion

- The end-to-end DCRN models substantially outperform the overlapped sliding windows model and the segmentation-based method.

- The synthetic pattern generation method improves the accuracy of the end-to-end DCRN models.

- Recognition rate is further improved when combined with the language model.
Future Work

- Apply the DCRN model for offline handwritten multi-lines data.

- Apply the RNN language model and compare it with the tri-gram language model.

- Apply for the JIS level 2 characters (~7,000 categories).
Thank you for your attention.