Handwritten Editor

Hidehisa Arai
TUM Department of Informatics, Chair for Applied Software Engineering
Technical University of Munich
Munich, Germany
koukyo1213@g.ecc.u-tokyo.ac.jp

***Abstract*—Handwriting is still in strong demand as a way of communication and saving knowledge even in the era of new technology. Driven by the rise of electronic pens and tablets, the need for handwriting recognition on electronic devices is increasing recently.**
This report focuses on handwriting recognition on iOS device and discusses how to use the recognition results using auto-completion as an example. Since handwritten document recognition technology is mature, this report focused more on reducing the amount of computation for on-device inference.

I. Introduction

Handwriting recognition (HWR) is a field which study and develop algorithms to interpret handwritten inputs into a format that can be easily handled by computers from sources such as papers, photographs, electronic tablets, and other devices. HWR can be roughly devided into two appproaches: online approach and offline approach [1].
While online approach uses information on the trajectory of the pen tip obtained from a special pen for classification, offline method uses optically scanned images as input and performs recognition using computer vision techniques. This report focuses only on offline approach, and tackle on the recognition problem using Convolutional Neural Network (CNN), which have shown a remarkable development in recent years.

HWR is a field that has been studied for a long time, and many applications have already been made. However, in the case of tablet devices that have spread rapidly in recent years, not so many applications have been created even after APIs to incorporate with pattern recognition algorithms on device are published by developpers of those devices. Although some applications have achieved very good result on normal handwritten text recognition, it is not the case when elements in other domains such as handwritten illustlations are mixed in addition to sentences.
I therefore endeavor to recognize handwritten documents which contain not only text, but also handwritten illustlations or mathematical formulas. Since this type of documents are very common in our daily life, the success of the project can potentially bring the fusion of digital technology and the long-standing human skills of handwriting.

In the following section, typical approaches of HWR and related works are introduced. Section three provides technical details of the approach to the problem addressed in this report. Section four describes the result of the approach and discuss on that. Section five concludes this report with future prospects.

II. Related Work

The problem settings of this project can be positioned as one variant of Scene Text Detection/Recognition, which is a field to study algorithms to extract and recognize text information written in natural images. Due to the recent development of Neural Networks technology, much research has been done in this field to this day [2].
Except for few methods [3] [4], most approaches of Scene Text Detection/Recognition separate detection and recognition and perform stepwise inference.

A. Detection

Scene Text Detection can be subsumed under general object detection, therefore those methods usually follow the same procedure of object detection, which is dichotomized as one-stage methods and two-stage ones [5].

After the emergence of FasterR-CNN [6], most of the modern text detection algorithms are based on FasterR-CNN, YOLO [7], SSD [8]. In addition to the general object detection model, text detection models are devised to detect tilted bounding boxes and character regions of arbitrary shapes, or to simplify the pipeline.

Since this report does not elaborate on text detection with the reasons described below, no more detailed explanation are provided.

B. Recognition

Some text recognition algorithms devide the task into character segmentation and character recognition [9] [10]. Character segmentation is considered as the most challenging part of scene text recognition, and may affect overall accuracy. It is especially difficult to segment connected characters such as cursive. Therefore some techniques which do not rely on character segmentation have been developped so far. This report introduces a method called Connectionist Temporal Classification (CTC) [11].

CTC was first introduced to handle sequence labeling of arbitrary length, requiring no pre-segmented training data. A CTC network outputs probabilities for each label at each time step. Time step length can be any length longer than label length. The output at each time step is the probability of the classess to be recognized plus the extra class representing "blank". Let this output probabilities be $\mathbf{y} = (y_1, y_2, \cdots, y_w)$ and denote by $y^t_{\pi_t}$ the activation of label π_t at time step t. Given this probability distribution, the conditional probability of the sequence is calculated as follows.

$$p(\pi|\mathbf{y}) = \prod_{t=1}^{w} y^t_{\pi_t} \tag{1}$$

Then a many-to-one mapping $\mathcal{B}$ is defined to transform the sequence π to a shorter sequence. The final predicted label is obtained by this mapping. This mapping removes all blanks and repeated continuous labels from the sequence. For example, $\mathcal{B}$ maps the predicted sequence "aa-p-pl—-ee" to "apple", where "-" represents the "blank". Since this mapping is many-to-one mapping, different sequences may be mapped to the same sequence. Therefore the probability of the final output sequence is the sum of all possible conditional probabilities of all π corresponding to that final sequence.

$$p(l|\mathbf{y}) = \sum_{\pi} p(\pi|\mathbf{y}) \tag{2}$$

where π represents all π which produces $l = \mathcal{B}(\pi)$.

The output of the classifier should be the most probable labeling for the input sequence.

$$h(\mathbf{y}) = \arg\max p(l|\mathbf{y}) \tag{3}$$

In general, there are a large number of mapping paths for a give sequence, thus calculation of $\arg\max$ requires heavy computation. In practice, following two approximate methods are known to give us a good result.

The first method is based on the assumption that the most probable path can be approximated by the sequence of most probable labeling

$$h(\mathbf{y}) \approx \mathcal{B}(\pi^*) \tag{4}$$

where π^* is a set of labels which get the highest probabilities at each time step. Although it works well, it is not guaranteed to get the most probable labeling.

The second method is to use forward-backward algorithm to efficiently search for the most probable sequence. With enough time, this approach can always find the most probable labeling from the input sequence, but the amount of computation increases exponentially with respect to the sequence length, it is not practical to find the exact solution.

To train the network with the dataset $\mathcal{D} = \{I_i, l_i\}$, where I_i represents the input image and l_i represents the corresponding label, maximum likelihood approach it utilized. The objective function of this can be negative log-likelihood

$$\mathcal{O} = -\sum_{(I_i, l_i) \in \mathcal{D}} \log p(l_i|\mathbf{y}_i) \tag{5}$$

where $\mathbf{y}_i = f(I_i)$ and $f(\cdot)$ represents the classifier. To minimize negative log-likelihood, Stochastic Gradient Descent (SGD) can be used.

III. Methods

This section describes in detail the approach to the classification problem in documents that contain handwritten text as well as handwritten illustrations.

As in the case of Scene Text Detection/Recognition, it is effective to separate Text Detection and Text Recognition and treat them as different problems. Furthermore, it is possible to record the written area due to the characteristics of the electronic tablet, so using a simple heuristic on the trajectory data eliminates the need to actually perform text detection. The role of this step, namely "Region of Interest Detection" step is to reduce the size of data passed to the subsequent processing and suppress the increase in the amount of calculation.

In general, electronic tablets have severe limitations on computing power, so in this report I used the heuristic to detect region of interst. Detected regions are then preprocessed and passed to the text recognition module. In the text recognition module, two patterns of recognition using CTC and recognition combining Character Segmentation and Character Recognition were verified.

A. Region of Interest Detection

The purpose of region-of-interest detection is to cut out subsequence representing words and illustrations from the dot sequence of the pen tip trajectory and cut back the data to be passed to the subsequent processing to reduce the amount of calculation and improve the recognition accuracy. Region of interest is a region including a dot sequence of the word or the illustration. To specify the region, two assumptions are made on region of interest.

1) Objects drawn in the region of interest are close in time when drawn
2) Objects drawn in the region of interest are spatially close

Based on these assumptions, region of interest is successfully detected. Figure 1 shows an example of the region detected with this heuristic.

There are some cases where such heuristics do not work. For example, when the scale of the depicted object is large, the gap between the sequence of points constituting the object may be too large and fail.It is also a major problem that temporal and spatial proximity depends on parameters and sometimes does not match intuition. However, it is certain that this method can narrow down the target area for text recognition with a very small amount of computation, so this report adopted this method.

B. Recognition

In text recognition, two models were tried: a model that directly reads the content from the result of region of interest detection using CTC, and a two-staged approach in which Character Segmentation and Character Recognition were connected in series.

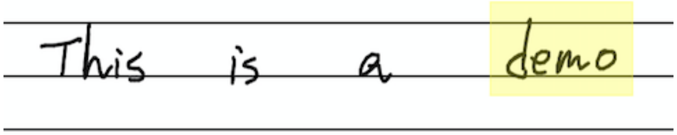

Fig. 1. The region with yellow overlay is detected region of interest

1) CTC: CTC models are usually constructed using CNN model with Recurrent Neural Network (RNN) as the output layer. However, RNN performs computations slowly compared to CNN, and in recent years its effectiveness in CTC is sometimes questioned [12]. Given that all calculations are done on iOS, and based on research that the output layer of the CTC model can also be configured using CNN [13], in this report a model which is fully constructed only by CNN was used for CTC model. To get an approximation for most probable labeling, greedy policy, the policy to take the most probable label at each time step, was used.

2) Character Segmentation: The target environment of this report is on an electronic tablet, and since it does not have a complicated background, a model designed with a Unet-like [14] architecture based on MobilenetV2 [15] was used for character segmentation with emphasis on speed and simplicity of the model.

3) Character Recognition: It is known that character recognition is sufficiently accurate even if a simple neural network model is used. Therefore, a small CNN model was designed in consideration of the amount of calculation.

C. Auto-Complete

In this report, I deal with auto-completion as an application of the inference result. This auto-completion emphasizes simplicity and presents words that start with infered result in order of frequency.

D. Dataset

Handwritten character recognition and handwritten sentence recognition are fields that have been studied for a long time, so there are many data sets, but these are often provided in different formats, and there is some difficulty in eliminating differences between formats and using them for training dataset.

I therefore took an approach to create a composite dataset by embedding a combination of existing handwritten-like fonts and randomly selected English words in the image. Figure 2 shows an example of training data generated with this method.

E. Implementation

I used Python3.6[1] and TensorFlow2.1[2] for building machine learning models and training, and used coremltools3.2[3] to convert the models into the format that can work on iOS.

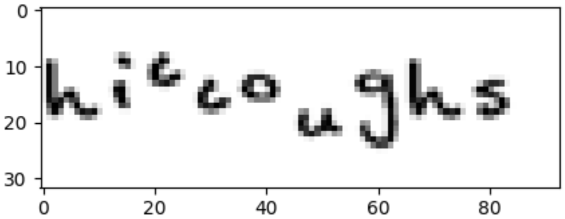

Fig. 2. Generated image using handwritten-like fonts

To get handwritten document image from the iOS electronic tablet, a function to print the input from the Apple Pencil[4] on the white canvas according to the path of the pen has been implemented.

For training the model, Google Colaboratory environment [5] has been used. All the implementation is published[6].

IV. Results & Discussion

- Describe your results in a clrear and understandable way.
- Clearly differentiate between what you have achieved and what you have build upon.
- Ideally add some sort of visual representation of your result that underlines the progress you have made during the research project.
- Make sure that the results are reproducible by your reader if needed
- Critically discuss your results.
- Did you achieve what you set out to do?
- What are the strengths and weaknesses of your research?

V. Future Work & Conclusion

- Summarize your thoughts and state your final conclusion about the work you have performed.
- Describe possible future work in the field that is realted to you work.
- Detail improvements that could be done to your work in a following project.
- Identify the importance of your work and create an arch to the related work and problem defined in the previous chapters.

References

[1] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[2] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," 2018.

[3] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5676–5685.

[1] https://www.python.org/downloads/release/python-360/

[2] https://www.tensorflow.org/

[3] https://apple.github.io/coremltools/

[4] https://www.apple.com/apple-pencil/

[5] https://colab.research.google.com/notebooks/welcome.ipynb

[6] https://github.com/koukyo1994/iOS-note-v2

[4] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–83.

[5] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[9] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 785–792.

[10] T. Q. Phan, P. Shivakumara, B. Su, and C. L. Tan, "A gradient vector flow-based method for video character segmentation," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1024–1028.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[12] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 67–72.

[13] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Reading scene text with attention convolutional sequence modeling," *arXiv preprint arXiv:1709.04303*, 2017.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2018.