Character decomposition to resolve class imbalance problem in Hangul OCR

Geonuk Kim¹  Jaemin Son²  Kanghyu Lee²  Jaesik Min¹
¹42dot.ai  ²Hyundai Motor Group
{geonuk.kim, jaesik.min}@42dot.ai  {woalsnd, kanghyulee}@hyundai.com

Abstract

We present a novel approach to OCR (Optical Character Recognition) of Korean character, Hangul. As a phonogram, Hangul can represent 11,172 different characters with only 52 graphemes, by describing each character with a combination of the graphemes. As the total number of the characters could overwhelm the capacity of a neural network, the existing OCR encoding methods pre-define a smaller set of characters that are frequently used. This design choice naturally compromises the performance on long-tailed characters in the distribution. In this work, we demonstrate that grapheme encoding is not only efficient but also performant for Hangul OCR. Benchmark tests show that our approach resolves two main problems of Hangul OCR: class imbalance and target class selection.

1. Introduction

Optical Character Recognition (OCR) technology aims to recognize texts in the form of pixels in an image, which has long been studied in the field of computer vision. Most research efforts [8, 10, 12, 21, 24, 27] have been dedicated to the recognition of English characters which consists of 26 classes of the alphabet and few efforts [2, 6, 25] address other languages. However, the techniques applied to English character recognition cannot be directly applied to Hangul where the total number of recognition targets induces a severe class-imbalance.

Unlike English, each Hangul character is described as a combination of 2 or 3 graphemes as shown in Figure 1: either [first consonant + vowel] or [first consonant + vowel + last consonant]. The total number of characters amount to 11,172, which may easily exhaust softmax layers in representational capability. Hence, the current Hangul OCR systems utilize only a subset of characters, maintaining about a thousand most frequently used ones. Therefore, the encoding choice naturally excludes characters that would appear in reality. Moreover, the dataset is severely disproportionate in terms of the appearance of characters, which aggravates the class-imbalance issue. Previous approaches attempt to generate virtual images to provide evenly-distributed-characters [14, 15, 20], however, they cause extra training time for learning characters that may not be critical to target domains.

To address this issue, we propose to decompose the character into graphemes. Since consonants and vowels are at each position with fixed rules in Hangul ([first consonant, vowel] or [first consonant, vowel, last consonant]), changing the recognition target from characters to graphemes can replace the problem of classifying 11,172 classes to 52 classes. This decomposition has two main advantages: first, the class-imbalance problem is significantly alleviated as we let the model learn to classify the graphemes in the majority class and minority class of characters into the same grapheme class. Second, unlearned characters can also be recognized through the combination of previously learned consonants and vowels, enabling character recognition for the entire characters of Hangul. We construct a benchmark for Korean text recognition task and experimentally verified the effectiveness of the proposed method.

2. Related Work

Scene text recognition task, as a subtask of the entire OCR system, aims to recognize texts in a cropped image that primarily contains text objects. The techniques are widely used in various real world applications ranging from

Figure 1. Innate structure of the Hangul, Korean character. A character is composed of 2 or 3 graphemes either [first consonant + vowel] or [first consonant + vowel + last consonant]. The total number of legitimate characters amounts to 11,172.
license plate recognition, cursive recognition, to credit card number recognition [18].

Canonical methods modeled the recognition task as a sequence prediction - visual features are extracted from a CNN backbone such as resnets [9] and fed into RNN layers such as LSTM [11] and GRU [3] to be trained with connectionist temporal classification (CTC) loss [8]. Recent approaches [7, 13] make use of more robust backbone such as vision transformer (ViT) [5] that models interactions between image patches, inspired by attention mechanisms in Transformer [23] developed in natural language processing.

The selection of which of the 11,172 characters of Hangul to train has been handled from different angles in Korean text recognition. Previous studies bifurcate into 1) selecting only frequently used characters [15, 20] or 2) including all of the 11,172 characters [14]. Both of the methodologies have obvious limitations: in the former approaches, characters outside the chosen set can never be recognized. The latter requires much more data containing a wider range of characters, in order to classify characters that rarely appear in real-world situations. Theoretically, the optimal sets of characters can be searched for all specific domains, yet, it is resource-intensive and still unclear whether the trained models are transferable between different domains.

3. Method

For scene text recognition in Hangul, the input is an image and the output is the text in Hangul. As shown in Figure 2, we employ an encoder-decoder structure which is widely used in previous studies [7, 17, 24, 27]. In encoder, CNN layer $\phi$ and transformer-unit $\psi$ are employed to learn the context between graphemes that define a character. For the input image $x$, we have:

$$f_e = \psi(\phi(x)) \in R^{H \times W \times N}$$ (1)

where $N$ is the feature dimension and $H, W$ are the size of $x$.

Then the decoder learns to attend feature corresponding to each grapheme in the given rich feature map $f_e$. As shown in Figure 1, each grapheme exists in certain geometric prior. Thus, we express the positional relationship among each set of [first-middle-last] graphemes on the feature map by using positional encoding. With the position encodings of the character orders $Q \in R^{L \times C}$ and length of the total sequence $L$,

$$f_d = \text{softmax}(\frac{QK^T}{\sqrt{N}})V \in R^{L \times N}$$ (2)

where $K \in R^{H \times W \times N}$ is reconstructed feature map with U-net structure.

Finally, in the Grapheme-based Predictor, each grapheme is classified with linear classifier $\varphi$ as follows:

$$g = \varphi(f_d) \in R^{L \times C}$$ (3)

where $C$ is the number of graphemes in the Hangul. We use cross entropy loss for the individual grapheme classification. During the inference, the predicted graphemes are merged into the final output in a form of characters as follows:

$$s = h(g^*)$$ (4)

where $g^* \in R^{L \times 1}$ is top-1 classification output. Note that the length of the $s$ is $\frac{L}{3}$.

There is an exceptional type of characters where the last grapheme doesn’t exist. To handle this case, we add a virtual class of “no-last-consonant” to unify the two different types of characters of [first consonant, vowel] and [first consonant, vowel, last consonant] structure into a single structure of [first consonant, vowel, last consonant].

4. Experiments

4.1. Datasets

We newly present a benchmark for Hangul OCR which has the intractable number of classes to recognize. This benchmark reveals the class-imbalance and target-class selecting issues in Hangul OCR. The benchmark dataset is available at https://github.com/mandal4/HangulNet.
**AI Hub** It consists of about 100,000 images of Hangul characters supported by an AI integrated platform operated by the Korea Intelligent Information Society Agency [4]. A total of 674,110 text areas are extracted to evaluate the performance of character recognition, excluding character detection. Of these, 10,000 are separated into the test set, and the rest are used as a training data.

**MLT-h** MLT dataset [19] was introduced as a part of Robust Reading Competition in ICDAR to resolve the problem of multi-lingual text detection and script identification. We exploit only the Hangul text regions in the MLT17 test-set for the evaluation, and name it as MLT-h. Note that since we have found many annotation errors in this data set, we rectified those noisy labels.

**Standard Foreign Words (SFW)** In order to objectively evaluate whether the proposed model effectively alleviates the class-imbalance problem in Korean character recognition, we have synthesized a new dataset containing a large number of minority classes using SynthTiger [26]. The dataset contains a total of 18,831 standard foreign words [1] that are registered in the National Institute of the Korean Language, which is used only as the test set.

**Unseen Characters** From the aforementioned SFW dataset, we have selected 72 characters that could not be represented with a common character encoding, and generated an image per character. Then, we measure the performances of various models on this dataset to compare the robustness on the unseen characters.

### 4.2. Implementation Details

We employ 5 transformer layer for the encoder. $C$ is set to 52. Grapheme Merger is simply implemented with UTF-8 unicode. We trained all of the benchmark models for 800K iterations with a learning rate of 0.001, and a batch size of 300. The input images are resized into 32 x 128. All experiments are conducted on two Nvidia V100s.

### 4.3. Experimental Settings

We compare our performance with the Tesseract [22], EasyOCR [16] and more competing methods [24, 27]. We reproduce the methods [24, 27] and obtain the model with the same training dataset. According to the previous studies, most commonly used 1,549 Hangul characters are chosen as the baseline encoding. In addition, we conduct an ablation to verify the capability of the transformer in modeling the context between consonants and a vowel that defines a character.

### 4.4. Quantitative Results

For the quantitative performance comparison, we evaluate the character-level accuracy and the word-level accuracy. Previous studies [1] assess the performance only on the selected target-classes, yet, we set up an additional experimental setup to simulate real-world scenarios where characters not included in the training set occasionally appear. For SFW* in Table 1, the accuracy is calculated on the whole range of characters covered in SFW dataset. The results from Tesseract [22] and EasyOCR [16] are not reported as their target-classes are not verifiable.

As shown in Table 1, our approach seamlessly outperforms the existing baselines in all datasets. In particular, the proposed method achieves 4.5% higher word-level accuracy than the strong baseline [27] in SFW where a number of minority classes exist. This shows that our framework is able to successfully classify the graphemes that belong to the minority class. For MLT-h and AI Hub, the accuracy gap between our framework and the strong baseline is smaller than that of SFW because the imbalance in these test sets is significant. Note that the accuracy of minority class is not emphasized when the test-set is imbalanced.

Table 1 also shows that excluding transformer design in the encoder degrades the word-level accuracy 3.5%, 2.5% and 8.2% in the MLT-h, AI Hub and SFW, respectively. This reveals that transformer captures the context between consonants and a vowel that defines each character. In SFW*, the word accuracy of the strong baseline [27] is

| Model          | MLT-h | AI Hub | SFW | SFW* |
|---------------|-------|--------|-----|------|
| Tesseract [22]| 13.0  | 9.4    | 12.3 | -    |
|              | 11.5  | 8.9    | 7.2  | -    |
| EasyOCR [16]  | 71.9  | 73.8   | 63.7 | -    |
|              | 55.5  | 55.6   | 27.7 | -    |
| Wang et al [24]| 91.6 | 96.5   | 87.7 | 86.3 |
|              | 84.5  | 92.8   | 67.8 | 66.7 |
| SRN [27]      | 93.2  | 97.7   | 89.7 | 88.3 |
|              | 89.2  | 95.4   | 72.9 | 71.8 |
| Ours (w/o Transformer) | 95.8 | 98.1   | 94.0 | 94.0 |
|              | 85.9  | 93.0   | 69.2 | 69.2 |
| Ours          | 96.1  | 98.8   | 96.1 | 96.1 |
|              | 89.4  | 95.5   | 77.4 | 77.4 |
1.1% lower than the SFW counterpart, but the performance of our framework does not get degraded at all. This is because our grapheme encoding enables recognizing characters of all cases in Hangul.

Table 2. Performance comparison on the Unseen Characters dataset. Character-encoding methods \cite{24, 27} inevitably fail in classifying any characters that are not included in the training process.

| Model                  | Encoding unit | Accuracy |
|------------------------|---------------|----------|
| Wang et al \cite{24}   | character     | 0.0      |
| SRN \cite{27}          | character     | 0.0      |
| Ours (w/o Transformer) | grapheme      | 52.8     |
| Ours                   | grapheme      | 73.6     |

On Unseen Characters dataset, the proposed method shows notable recognition performances. Table 2 shows that our framework achieves 73.6% character-level accuracy. On the other hand, the conventional common character-encoding methods \cite{24, 27} can never recognize any unseen characters.

4.5. Further Analysis

To verify the effectiveness of our approach on addressing the class-imbalance problem, we plot the frequency of the characters and the graphemes with AI Hub dataset which consists of large-scale real-world Hangul images. As shown in Figure 4, there is a huge difference in frequency between the majority character classes and the minority character classes. The imbalance rate of the characters, which is represented with Gini Index, is 0.8644. On the other hand, when substituted characters with graphemes, the class-imbalance is substantially alleviated and the Gini Index is lowered to 0.6687 (Figure 5).

Additionally, we visualize the attention weight map to validate whether the recognition of each grapheme is effectively performed through positional encoding. Figure 6 shows that the model infers Hangul while following the character construction rule of \textit{[first consonant, vowel, last consonant]}. Note that the last row represents attention map for the class “no last consonant”.

5. Conclusion

In this paper, we propose a grapheme-level Hangul text recognizer with an encoder-decoder structure. With our grapheme-level classification approach, the class-imbalance problem is substantially alleviated by learning to classify the graphemes regardless of whether they are majority or minority characters. Extensive experiments demonstrate the robustness of our approach even in unseen characters that are difficult to handle with conventional methods. Moreover, we newly present a benchmark for Hangul OCR to reveal class-imbalance and target-class selecting issues in Hangul OCR.
References

[1] Korean standard foreign word. Retrieved from https://stdict.korean.go.kr/main/main.do, 2020. Last accessed on July 13, 2022. 3
[2] Jingye Chen, Bin Li, and Xiangyang Xue. Zero-shot chinese character recognition with stroke-level decomposition. arXiv preprint arXiv:2106.11613, 2021. 1
[3] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014. 2
[4] S.J. Kim. Al Hub Korean Text-In-The-Wild Dataset. Retrieved from https://aihub.or.kr/, 2020. Last accessed on July 13, 2022. 3
[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 2
[6] Agam Dwivedi, Rohit Saluja, and Ravi Kiran Sarvadevahala. An ocr for classical indic documents containing arbitrarily long words. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 560–561, 2020. 1
[7] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7098–7107, 2021. 2
[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning, pages 369–376, 2006. 1, 2
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016. 2
[10] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiangao Tang. Reading scene text in deep convolutional sequences. In Thirtieth AAAI conference on artificial intelligence, 2016. 1
[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997. 2
[12] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Ziping Lin. Gtc: Guided training of ctc towards efficient and accurate text recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 11005–11012, 2020. 1
[13] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4593–4603, 2022. 2
[14] Ga-Hyeon Kang, Ji-Hyun Ko, Yong-Jun Kwon, Na-Young Kwon, and Seok-Ju Koh. A study on improvement of korean ocr accuracy using deep learning. In Proceedings of the Korean Institute of Information and Communciation Sciences Conference, pages 693–695. The Korea Institute of Information and Commucation Engineering, 2018. 1, 2
[15] In-Jung Kim and Xiaohui Xie. Handwritten hangul recognition using deep convolutional neural networks. International Journal on Document Analysis and Recognition (IJ- DAR), 18(1):1–13, 2015. 1, 2
[16] Rakpong Kittinaradorn. Easyocr. Retrieved from https://www.jaided.ai/easyocr/, 2020. Last accessed on July 13, 2022. 3
[17] Junyeop Lee, Sungae Park, Jeonghun Baek, Seong Joon Oh, Seonghyeon Kim, and Hwalsuk Lee. On recognizing texts of arbitrary shapes with 2d self-attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 546–547, 2020. 2
[18] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). IEEE Access, 8:142642–142668, 2020. 2
[19] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 1, pages 1454–1459. IEEE, 2017. 3
[20] Sun-Woo Park. A study on the ocr of korean sentence using deeplearning. In Annual Conference on Human and Language Technology, pages 470–474. Human and Language Technology, 2019. 1, 2
[21] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39(11):2298–2304, 2016. 1
[22] Ray Smith. An overview of the tesseract ocr engine. In Ninth international conference on document analysis and recognition (ICDAR 2007), volume 2, pages 629–633. IEEE, 2007. 3
[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017. 2
[24] Peng Wang, Lu Yang, Hui Li, Yuyan Deng, Chunhua Shen, and Yanning Zhang. A simple and robust convolutional-attention network for irregular text recognition. arXiv preprint arXiv:1904.01375, 6(2):1, 2019. 1, 2, 3, 4
[25] Wenchen Wang, Jianhua Zhang, Jun Du, Zi-Rui Wang, and Yixing Zhu. Denseran for offline handwritten chinese character recognition. In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 104–109. IEEE, 2018. 1
[26] Moonbin Yi, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. Synthtiger: Synthetic text image generator towards
better text recognition models. In *International Conference on Document Analysis and Recognition*, pages 109–124. Springer, 2021.

[27] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020.