

Received 2 January 2024, accepted 27 January 2024, date of publication 31 January 2024, date of current version 8 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360482

RESEARCH ARTICLE

Definition and Automatic Extraction Performance Analysis of Stroke Elements in the English Alphabet

SOON-BUM LIM¹, YUJIN LEE¹, AND YOOJEONG SONG^{1,2}

¹Department of IT Engineering, Sookmyung Women's University, Seoul 04310, South Korea

²School of Computer Science, Semyung University, Jecheon 27136, South Korea

Corresponding author: Yoojeong Song (yjsong@semyung.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2021R1I1A4A01059550 and Grant RS-2022-00165818.

ABSTRACT Fonts are a critical element that determines the perception of any medium. To ensure consistent and culturally appropriate font selection across diverse language groups, a multilingual font matching system is currently in development. This research focuses on leveraging the latest advancements in machine learning and computer vision to deeply understand font characteristics and enhance the accuracy of multilingual font matching. Utilizing the 'stroke elements' of fonts is crucial for this matching, building upon the successful development of a method to calculate similarity between Korean fonts in previous studies. We have applied this approach to the English alphabet, defining distinctive 'stroke elements' and developing a deep learning model for their automatic extraction. Additionally, we evaluate the performance of this stroke element extraction model and discuss strategies to further improve extraction accuracy. This groundwork establishes the basis for multilingual font matching and enables the recommendation of similar fonts using the 'stroke elements' of the English alphabet.

INDEX TERMS Artificial neural networks, deep learning, diverse font styles, fonts, object extraction model.

I. INTRODUCTION

The font has a significant impact on the overall impression of the medium in which it is used. Therefore, it is crucial to convey the same feeling as the original font when replacing fonts used in various media with fonts from different language groups. As a result, research is being conducted on a multilingual font matching system that recommends similar-looking fonts that suit different language groups [1], [2], [3]. The correlation between font aesthetics and cultural preferences adds another layer of complexity to the font matching process. Cultural nuances often influence how fonts are perceived, making the task of recommending suitable fonts across diverse languages even more intricate.

However, recent advancements in machine learning and computer vision have paved the way for more sophisticated font analysis techniques. These techniques enable a deeper

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

understanding of the intricate details that contribute to font similarity, enhancing the accuracy of the multilingual font matching system.

Multilingual font matching is the task of recognizing and matching fonts while considering diverse languages and cultures. This is because fonts in different languages often have distinct shapes, sharpness, thickness, and other attributes, which are related to the characteristics of the respective languages and cultures. For example, Korean and English have different writing systems, where Korean characters combine to form syllables, whereas English consists of individual alphabet characters. These differences necessitate that multilingual font matching systems are capable of handling various writing systems and font styles. Additionally, cultural nuances influence how each language and culture perceive and use specific fonts. For instance, in Korean culture, vertically elongated fonts are commonly used, reflecting cultural preferences. In contrast, horizontally wider fonts may be more prevalent in English. These cultural

disparities impact font recognition, and thus, multilingual font matching systems need to account for such cultural subtleties to cater to diverse languages and cultures. We have developed an algorithm in previous research that calculates the similarity between Korean fonts and recommends fonts based on this similarity [4]. We also applied this method to Chinese fonts (Hanja) to conduct research on similar font mappings between Korean and Chinese characters [5]. The method involved comparing the fonts' similarity using distinctive 'stroke elements' in characters, and it showed excellent performance. Stroke elements, which help distinguish between font groups, possess characteristic visual appearances, such as the presence or absence of serifs in serif and sans-serif fonts, aiding in differentiation. In prior research, these characteristic visual elements were directly defined and validated to establish stroke elements. A total of 8 stroke elements were defined, utilized to calculate the distance between fonts, and determine their similarity. Building on this success, we defined 'stroke elements' specific to the English alphabet and developed a model to extract these elements from various fonts.

Here are the contributions of this paper: Extracted stroke elements play a pivotal role as a groundwork for prospective multilingual font matching. Given the English alphabet's distinct morphological characteristics compared to Korean, defining unique stroke element attributes specific to the alphabet becomes essential. Our approach employs an image-based object detection model for the automated extraction of these defined stroke elements, ensuring a high level of accuracy. This paper concentrates on delineating the stroke element features inherent to the English alphabet, facilitating font recommendations based on these features, and validating the efficacy of stroke element extraction. Additionally, we delve into strategies aimed at enhancing extraction performance. The structure of this paper is as follows: In Section II, we provide an overview of previous research and related studies that form the basis of this research. Section III elaborates on the definition of stroke elements for the English alphabet and the stroke element detection model. In Section IV, we explain font classification based on visual differences among fonts and the performance of stroke element extraction for extracting stroke elements from English alphabet fonts. Finally, in the conclusion, we summarize the results of this research and outline directions for future studies.

II. RELATED WORKS

In research [4] representative stroke elements, phrases, and fonts were initially defined for generating training data and for validation. Regarding fonts, they were categorized into three types: structured fonts with significant variations in stroke thickness, semi-structured fonts with similar characteristic elements to structured fonts but with greater variations in stroke thickness, and unstructured fonts with many stroke elements that deviate from the typical stroke shapes. For each type, character images were extracted and used to

train the model. The stroke element detection performance was evaluated by comparing the results with ground truth data. The stroke element detection model exhibited a 99% detection accuracy when detecting individual characters. However, in experiments involving character combinations, it showed a 90% detection accuracy. This variance was due to changes in character structure and appearance resulting from the combination of characters. Moreover, while the detection was robust for structured fonts, there was a significant drop in performance for semi-structured and unstructured fonts. Hence, there is a need for the development of a model with high detection accuracy for both structured and semi/unstructured fonts. Research on finding similar typefaces encompasses a wide range of languages, including English, Chinese, and Korean [6], [7], [8], [9], [10], [11]. Among them, research on finding similar typefaces for English fonts often relies on simple image comparison methods. However, utilizing stroke elements for comparison offers the advantage of assessing more nuanced similarities [5]. The research utilizing stroke elements to recommend similar fonts has been limited to Hangul and Chinese characters. Currently, there is no study defining and detecting stroke elements for the English alphabet and applying them to compare and recommend similar fonts.

Other related study [12] presents a novel method for recognizing printed English characters from multiple fonts. The approach utilizes neural networks to achieve accurate character recognition. This study addresses the challenges posed by variations in font styles and sizes, aiming to improve the overall performance of character recognition systems. The main differences between our research and the referenced study lie in their approach and objectives. The referenced study primarily focuses on recognizing characters printed in various fonts, aiming for precise character recognition by utilizing neural networks. Conversely, our study examines the utilization of stroke elements to compare and recommend similar fonts. This involves analyzing and recommending fonts by understanding the distinctive stroke patterns and structures of each character, providing an advantage in evaluating more nuanced similarities. Therefore, our research emphasizes the detailed assessment of similarities between fonts, with the goal of contributing to the field of multilingual font matching.

Meanwhile, there is existing research that proposes a deep learning-based method for identifying fonts from images of English alphabets [13]. This paper addresses the problem of recognizing and identifying fonts depicted in images. In other words, the goal is to automatically determine the type of font used in a given image. Font identification is closely related to font recommendation based on similarity, but it has certain limitations. Fonts used in real-world environments can often be distorted or altered. Developing a robust model to handle such font variations can be challenging, and the paper may have shortcomings in addressing these variations. Therefore, we believe that utilizing a method that recognizes fonts and recommends similar fonts based on stroke elements,

rather than analyzing the entire image, can potentially yield better performance. The study on Hangul font clustering and recommendation is conducted by [14]. The study proposes a methodology that combines Convolutional Neural Networks (CNN) and font clustering techniques to achieve accurate and efficient recognition of Hangul characters across a wide range of fonts. The approach aims to address the challenges associated with the large diversity of Hangul fonts by leveraging the power of deep learning and clustering algorithms. The results demonstrate the effectiveness of the proposed system in achieving high recognition accuracy and adaptability for various Hangul font styles and sizes. Research studies like the ones mentioned above are all related to recommending similar fonts between different languages or character recognition tasks. Additionally, most of these studies involve calculating the similarity between characters represented as images. However, according to research conducted by [4] and [5], using stroke elements in characters enables a more detailed comparison of similar fonts. This insight has inspired us to develop a stroke element detection model for matching similar fonts between Korean and English.

III. EXPERIMENTAL DESIGN AND ANALYSIS

A. EXPERIMENTAL DESIGN

In this experiment, we defined stroke elements specific to the English alphabet for the purpose of extracting stroke elements. We also selected the target text and fonts from which to extract these stroke elements. The definition of stroke elements for the English alphabet began by listing the alphabets in order of their highest usage frequency. We selected the top 9 alphabets [15], which are ‘E’, ‘A’, ‘T’, ‘I’, ‘S’, ‘N’, ‘O’, ‘R’ and ‘L’ as candidates. Then, we examined the stroke elements that could be extracted from each of these alphabets.

Furthermore, considering that different alphabets may have different shapes even with the same stroke element name, individual stroke elements were given different names to be used separately. Finally, the selected representative stroke elements for English were ‘BowlR’, ‘BowlA’, ‘TailR’, ‘TailT’, ‘Serif’, ‘Spur’, ‘Apex’, ‘Spine’, ‘ArmL’, ‘ArmE’, ‘Shoulder’ and ‘Terminal’ totaling 12 elements. The representative English phrase was chosen as ‘LARGE trains’, which combines uppercase and lowercase alphabets, including the representative stroke elements. The illustrations of each stroke element can be found in Figure 1, and detailed descriptions for each stroke element are provided in Table 1, as follows.

Fonts come in various shapes and styles, each with its own classification system. Typically, English fonts are classified using the Vox classification method [16]. In this paper, we reclassified fonts into three categories for stroke element extraction experiments. These three categories are ‘Structured,’ ‘Semi-Structured,’ and ‘Unstructured’ fonts. In this paper, we focused on structured and semi-structured



FIGURE 1. Positions of 12 representative stroke elements in the phrase ‘LARGE trains’.

TABLE 1. Description of each stroke element.

Stroke Element Name	Description
Bowl	A round curve that creates a closed space
Tail	A downward stroke
Serif	A short stroke at the end of a stroke
Spur	A small projection from the main stroke
Apex	The top junction where the two strokes meet
Spine	The main part of a curved stroke in S
Arm	One or both ends of the stroke are connected based on the vertical stroke. Or a line that rises to the right
Shoulder	the rounded part of h, m, and n
Teardrop Terminal	the end of a stroke that resembles a tear

fonts for stroke element extraction. This decision was made to exclude the heterogeneous stroke element shapes found in graphic and script fonts, which belong to the unstructured font category and could hinder the recommendation of similar fonts. Unstructured fonts typically include handwritten or highly embellished fonts, and further research may be needed to develop additional methods for recommending similar fonts and defining stroke elements for this category.

We utilized Vox’s classification method, as illustrated in Table 2, to categorize English structured fonts. Following this approach, we selected a total of 18 representative fonts, with 2 fonts from each of the subcategories—Classicals and Moderns. For semi-structured fonts, these fonts have higher freedom compared to structured fonts but do not include embellishments and are not handwritten. We selected 2-3 fonts from each of the subcategories within the ‘fancy’ category of a prominent commercial free English font website, dafont [17], which includes typewriter, Old School, Western, Stencil, Groovy, and Retro. In total, we selected 18 representative English semi-structured fonts. These 36 fonts were used to measure the detection performance of the stroke element detection model.

Based on the aforementioned experimental design, the experiment was conducted in the following sequence. First, image files were generated for each alphabet using the

TABLE 2. 18 representative English structured fonts selected according to Vox's classification system.

Classification	Subcategories	Structured representative fonts		
Classicals	Humanist	Centaur	Cloister	
	Garalde	Garamond	Bembo	
	Transitional	Baskerville	Times Roman	
Moderns	Didone	Bodoni	Walbaum	
	Mechanistic	Clarendon	Bitter	
		Grotesque	Franklin Gothic	News Gothic
	Lineal	Neo-grotesque	Helvetica	Univers
		Geometric	Futura	Century Gothic
		Humanist	Gill Sans	Optima

English representative phrase, 'LARGE trains'. These image files were then labeled to create a dataset with a total of 12 stroke elements. The model was trained using this dataset. Next, using the trained model, the 12 stroke elements were detected from the image files created using the English structured representative fonts and English semi-structured representative fonts. Performance evaluation was carried out using the mean Average Precision (mAP) metric [18]. Figure 2 illustrates the overall process of this study.

B. DEEP LEARNING MODEL FOR STROKE ELEMENT EXTRACTION

This section provides an overview of the deep learning model utilized in our research for the automatic extraction of eight stroke elements from letter images. We employed the Faster R-CNN (Region-based Convolutional Neural Network) among various deep learning models. Determining the similarity of images can be approached in various ways. Typically, the full text image is used to assess font similarity. However, once stroke elements are extracted, a more efficient method of measuring similarity is to use the stroke element image [19]. Based on this idea, we aimed to extract and detect stroke elements for fonts with irregular shapes beyond the conventional standard fonts. Previously, we attempted stroke element detection using a Support Vector Machine (SVM) [20] for character image analysis. However, the accuracy significantly decreased during the detection process. Consequently, we opted for a deep learning-based object detection model for stroke element detection, as it employs a deeper neural network. We fine-tuned the Faster R-CNN Inception-V2 model [21] among deep learning-based object detection models. Faster R-CNN consistently demonstrated the highest accuracy compared to other models, especially excelling in the detection accuracy of small-sized objects. Furthermore, Faster R-CNN was adopted due to its superior stroke element detection performance compared to widely used YOLO [28] and SSD [29] models in image recognition. The structure of Faster R-CNN is illustrated in Figure 3.

C. TRAINING THE MODEL FOR ENGLISH STROKE ELEMENT EXTRACTION

For structured fonts, 178 font files were prepared, and for semi-structured fonts, 175 font files were prepared. The font data was collected manually and consists of both free

and commercially paid fonts. The Python image processing library, Pillow [22], was used to convert each alphabet of the representative phrase 'LARGE trains' into separate image files. Then, the labeling tool, Labelling [23], was used to designate the areas with the 12 stroke elements in each character image as labeled bounding boxes and annotate them with the stroke element names. The labeled data was saved in XML format. Through this process, the structured font group generated 1,958 data, and the semi-structured font group generated 1,925 data. Each group was divided into 80% train data and 20% test data. We had to gather and label the data manually because there was no existing dataset available for stroke element images. This allowed us to obtain a stroke element dataset, and we intend to keep collecting data continuously. The Faster-RCNN-Inception-V2-COCO model [24] was used in this study. The TensorFlow Object Detection API [25], a deep learning framework, was used to fine-tune the pre-trained model with the newly generated data. Table 3 summarizes key parameters and hyper-parameters of the employed model, facilitating a deeper understanding of its architecture and training configurations.

TABLE 3. Fine-tuned object detection model hyperparameters.

Model Used	Faster-RCNN-Inception-V2-COCO
Layers	Convolutional, RPN, Fully Connected
Activation Function	ReLU
Learning Rate	0.0002
Iterations	Structured Font: 49,694 / Semi-Structured Font: 54,034
Batch Size	64
Optimizer	Momentum

Faster-RCNN-Inception-V2-COCO model consists of Convolutional Layers, a Region Proposal Network, and Fully Connected Layers. The activation function used throughout the model is ReLU. The learning rate was set to 0.0002 for model training. The model was fine-tuned separately for the structured and semi-structured font groups. Training was conducted for 49,694 iterations for the structured font group and 54,034 iterations for the semi-structured font group, continuing until the loss rate dropped below 0.1. The batch size used during training was set to 64. The optimization was performed using the Momentum optimizer. The following Figure 4 and Figure 5 represent examples of structured and semi-structured fonts, respectively. These fonts with their distinctive visual characteristics were collected and used for training and stroke element extraction. In this study, images of size 256×256 were used for model training, and True Type Font (ttf) [26] files were converted to JPG format before use. The model's output consists of images containing the detected portions of the 12 predefined stroke elements. Other detailed parameters of the model were set the same as the original Faster-RCNN-Inception-V2-COCO model's parameters.

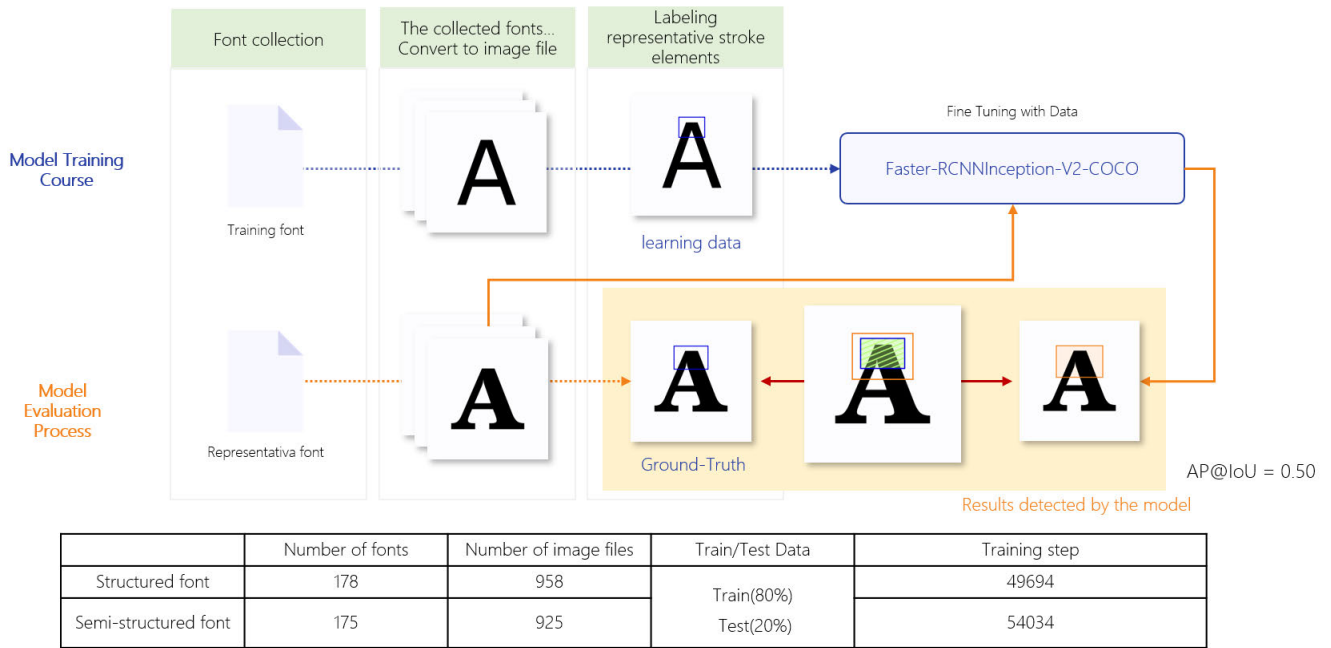


FIGURE 2. Overview of research progress.

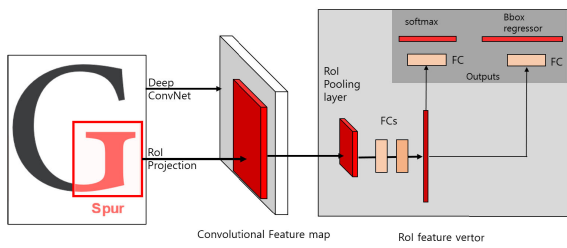


FIGURE 3. The structure of faster R-CNN.

Structured font representative font (18 types)	
Centaur	Cloister
Garamond	Bembo
Baskerville	Times Roman
Bodoni	Walbaum
Clarendon	Bitter
Franklin Gothic	News Gothic
Helvetica	Univers
Futura	Century Gothic
Gill Sans	Optima

FIGURE 4. Examples of structured fonts.

D. STROKE ELEMENT DETECTION PERFORMANCE EVALUATION AND IMPROVEMENT

The performance of the stroke element automatic detection model was evaluated by comparing it with the Ground Truth.

Semi-structured font representative font (18 types)	
GROBOLD	atwriter
<u>JackstayDemoRegular</u>	Bluetea
Magical Story	BOHEMIAN TYPEWRITER
Minercraftory	CHICKEN PIE
OMEGLE	CT ProLamina
Ruler Stencil Regular	DACTYLOG
Springtime Romance	FORCED SQUARE
StencilGothic	FuturaHandwritten
tommy font	Green London

FIGURE 5. Examples of semi-structured fonts.

Ground Truth was prepared by labeling the representative stroke elements in the 'L', 'A', 'R', 'G', 'E', 't', 'r', 'a', 'i', 'n', 's' character image files generated with untrained English representative fonts. The model was then used to detect the stroke elements in the same images. The results detected by the model and the Ground Truth were both indicated by bounding boxes on the character images. If the overlap area between the two bounding boxes measured by IoU (Intersection over Union) [27] was above 0.5, the prediction result was considered as a correct detection (True Positive). The detection performance of each stroke element in the model was quantitatively evaluated based on precision and recall, using the AP (Average Precision) metric. The overall

performance of the model in extracting the 12 stroke elements was evaluated using the mAP (mean Average Precision) value, which is the average AP of each stroke element. The reason for using mAP (Mean Average Precision) as an evaluation metric in object detection models is to quantitatively measure and compare the accuracy and performance of the models. Here are the main reasons for using mAP. Object detection models perform the task of predicting the location and class of objects in images. mAP is a critical metric to assess how accurately the model detects objects and estimates their positions, providing a reliable measure of model performance. Also, Object detection models can detect multiple objects simultaneously. mAP is useful for evaluating such multi-object detection tasks, reflecting the model’s ability to accurately detect and distinguish multiple objects. In the object detection models, commonly used performance metrics include the Precision-Recall Curve, F1-Score, and mAP (Mean Average Precision) [30], [31]. This paper utilizes mAP as its chosen metric due to the fact that mAP considers both the accuracy of object localization and individual object-level performance. By employing this metric, it becomes possible to simultaneously evaluate the model’s ability to estimate the positions of multiple objects. Furthermore, mAP is widely recognized as a standard evaluation metric for comparing the performance of various object detection models, enabling a fair and consistent comparison among different models. The formula for Mean Average Precision (mAP), which is the performance evaluation metric for the stroke element detection model, is as follows:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{1}$$

N : The total number of classes or categories.

AP_i : The Average Precision for each individual class or category.

The stroke element detection was conducted for both English regular fonts and English semi-regular fonts, and Table 4 provides the AP values for each stroke element and the mAP value of the model. The mAP value of the stroke element automatic extraction model trained on English regular fonts was 95.88%, while the model trained on English semi-regular fonts yielded a relatively lower mAP value of 75.01%.

In Figure 6, it can be observed that the model trained on English regular fonts successfully detected all stroke elements except for ‘TailT’ and ‘Serif’. However, when examining Figure 7, it can be seen that the model trained on English semi-regular fonts had incorrect detection for most of the stroke elements.

Upon analyzing the incorrectly detected results, three types of errors, as presented in Figure 8, were identified. The most common type, Type 1, involved the incorrect detection of stroke elements that should not be present in the respective characters. Additionally, Type 2 and Type 3 errors occurred, where the same stroke element was detected multiple times.

TABLE 4. mAP and AP value of stroke element detection model for structured and semi-structured fonts.

	Structured font	Semi-structured font
BowlR	100	77.78
BowlA	100	88.89
TailR	100	72.22
TailT	100	51.52
Serif	100	63.15
Spur	100	88.99
Apex	100	54.04
Spine	100	83.33
ArmL	100	74.44
ArmE	100	93.86
Shoulder	100	972.64
Terminal	100	79.23
mAP	95.88%	75.01%

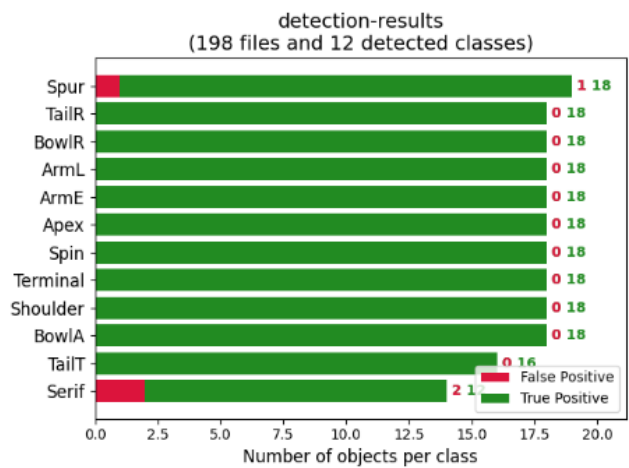


FIGURE 6. Detection accuracy of stroke element detection models for English structured fonts.



FIGURE 7. Detection accuracy of stroke element detection models for English semi-structured fonts.

To improve the performance of the stroke element extraction model for semi-regular fonts, the types of incorrect detection identified in Table 4 were taken into account. To reduce Type 1 detection errors, the character image file names were utilized to restrict the detection to only the

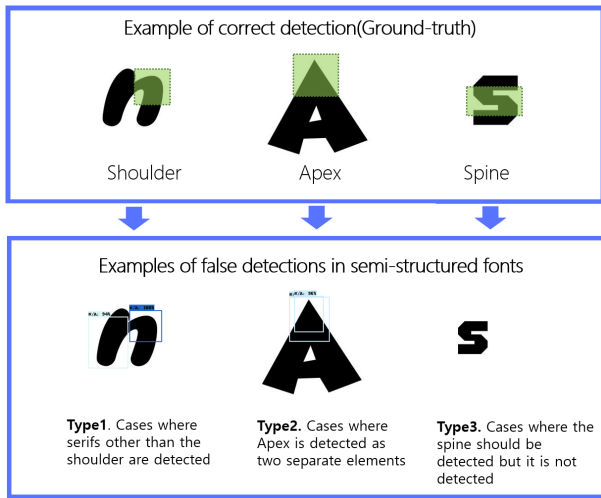


FIGURE 8. Performance improvement results of the stroke element detection model for English semi-structured fonts, including AP and mAP scores.

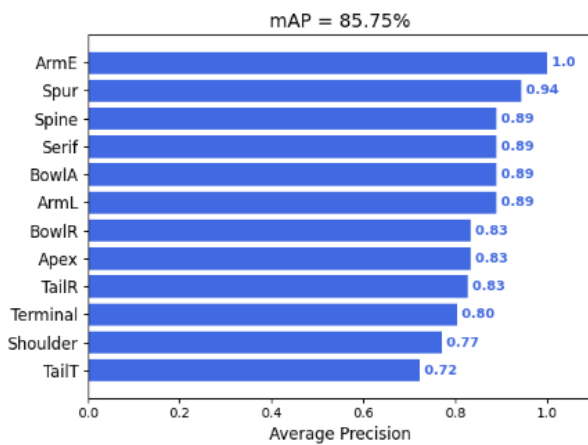


FIGURE 9. Performance improvement results of the stroke element detection model for English semi-structured fonts, including AP and mAP scores.



FIGURE 10. Detection accuracy improvement of the stroke element detection model for English semi-structured fonts.

stroke elements originally present in the respective alphabet. For Type 2 and Type 3 errors, where multiple instances of the same stroke element were detected, the model was modified

to only retain the more accurate detection. Subsequently, the performance of the model was evaluated again.

Upon reviewing Figure 9 and Figure 10, it can be observed that the mAP value of the model trained on English semi-regular fonts increased from 75.01% to approximately 85.75%, indicating an improvement of around 10%. Additionally, the visibly noticeable incorrect detection were significantly reduced.

IV. CONCLUSION

In this paper, we implement a multilingual similar font matching system, preliminary experiments were conducted to extract stroke elements from English fonts. When conducted on regular fonts, the majority of stroke elements were successfully detected. However, when applied to semi-regular fonts, the model exhibited a high frequency of incorrect detection. Two additional measures were taken to improve the performance of the model trained on semi-regular fonts. However, these methods are only applicable when the font file is provided as input by the user, which corresponds to the first input method in the multilingual similar font matching system. When the input data is in the form of an image, it may require a complex process to recognize the alphabets within the image. Therefore, further consideration is needed on how to improve the performance in such cases. Additionally, the results of the performance improvement are still insufficient, requiring further exploration for enhancing the performance. In future research, stroke element extraction will be attempted on irregular fonts to assess the overall performance across all three types of English fonts. Furthermore, it is anticipated that the methods employed to improve the performance of the model on semi-regular fonts will also aid in enhancing the performance of the model on irregular fonts. To further advance the multilingual similar font matching system, preliminary experiments were conducted to extract stroke elements from English fonts. While most stroke elements were successfully detected when applied to regular fonts, issues arose with the model’s performance when dealing with semi-regular fonts, resulting in a notable frequency of incorrect detection. In order to address these challenges, two additional measures were taken to enhance the performance of the model trained on semi-regular fonts. However, it’s worth noting that these methods are only applicable when users provide the font file as input, corresponding to the first input method in the multilingual similar font matching system. In cases where input data is in the form of images, intricate processes may be required to recognize the alphabets within the image. Thus, careful consideration is required in devising strategies to improve performance in such image-based scenarios. Furthermore, despite progress in enhancing performance, the results remain insufficient, indicating the need for further exploration to achieve substantial performance improvement. In future research endeavors, attempts will be made to extend stroke element extraction to irregular fonts to comprehensively assess overall performance across all three types of English fonts. Additionally, it’s anticipated

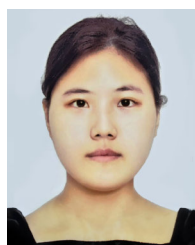
that the methods employed to bolster the model's performance on semi-regular fonts will also contribute to enhancing the model's proficiency with irregular fonts.

REFERENCES

- [1] D. Spohr, L. Hollink, and P. Cimiano, "A machine learning approach to multilingual and cross-lingual ontology matching," in *Proc. Semantic Web-ISWC 10th Int. Semantic Web Conf.*, Bonn, Germany, Berlin, Germany: Springer, Oct. 2011, pp. 665–680.
- [2] S. Rapee and T. Igarashi, "Example-based automatic font generation," in *Proc. Int. Symp. Smart Graphics*. Berlin, Germany: Springer, 2010, pp. 127–138.
- [3] K. Kim and J. Byun, "An implementation of optimized hangul font set for multilingual IM engine," in *Proc. 6th Int. Conf. Adv. Lang. Process. Web Inf. Technol. (ALPIT)*, Aug. 2007, pp. 200–205.
- [4] J. Ja-Yeon, "Automatic extraction of hangul stroke element using faster R-CNN for font similarity," *J. Korea Multimedia Society*, vol. 23, no. 8, pp. 953–964, 2020.
- [5] S.-B. Lim, J. Lee, X. Zhao, and Y. Song, "Detection model of hangul stroke elements: Expansion of non-structured font and influence evaluation by stroke element combinations," *Electronics*, vol. 12, no. 2, p. 383, Jan. 2023.
- [6] G. Chen, J. Yang, H. Jin, J. Brandt, E. Shechtman, A. Agarwala, and T. X. Han, "Large-scale visual font recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3598–3605.
- [7] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1192–1200, Oct. 2001.
- [8] P. O'Donovan, J. Libeks, A. Agarwala, and A. Hertzmann, "Exploratory font selection using crowdsourced attributes," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–9, Jul. 2014.
- [9] S. Dawn and B. Chaparro, "Perception of fonts: Perceived personality traits and appropriate uses," *Digital Fonts Reading*. World Scientific, 2016, ch. 13, pp. 226–247. [Online]. Available: https://doi.org/10.1142/9789814759540_0013
- [10] C. Hsien-Chih, M. Ma, and Y. C. Feng, "The features of Chinese typeface and its emotion," in *Proc. Int. Conf. KANSEI Eng. Emotion Res.*, Mar. 2010, pp. 1–11.
- [11] X. Kong and Z. Liu, "Reaserch on monolingual font recommendation based on target perception," in *Proc. Int. Conf. Intell. Design (ICID)*, Dec. 2020, pp. 94–97.
- [12] N. Samadiani and H. Hassanpour, "A neural network-based approach for recognizing multi-font printed English characters," *J. Electr. Syst. Inf. Technol.*, vol. 2, no. 2, pp. 207–218, Sep. 2015.
- [13] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, "DeepFont: Identify your font from an image," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 451–459.
- [14] J. Yang, H. Kim, H. Kwak, and I. Kim, "HanFont: large-scale adaptive hangul font recognizer using CNN and font clustering," *Int. J. Document Anal. Recognit. (IJДАР)*, vol. 22, no. 4, pp. 407–416, Dec. 2019.
- [15] G. Grigas and A. Juškevičienė, "Letter frequency analysis of languages using Latin alphabet," *Int. Linguistics Res.*, vol. 1, no. 1, p. p18, Mar. 2018.
- [16] H.-Y. Kim and S.-B. Lim, "Standardizing design-based font classification system for hangul font services," *Comput. Standards Interfaces*, vol. 55, pp. 47–54, Jan. 2018.
- [17] D. Schander, "Digital signage: A new tool in your arsenal of knowledge," *AALL Spectrum*, vol. 17, no. 5, p. 7, 2013.
- [18] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 237–242.
- [19] J. Ja-Yeon, "CNN-based similar font recommendation techniques using hangul stroke elements information," M.S. thesis, Dept IT Eng., Sookmyung Women's Univ., Seoul, South Korea, 2021.
- [20] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.
- [21] R. Shaoqing et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [22] A. Clark, "Pillow (PIL fork) documentation," *Readthedocs. Io*, vol. 620, pp. 1–199, Jun. 2015.
- [23] A. Yakovlev and O. Lisovychenko, "An approach for image annotation automatization for artificial intelligence models learning," *Adapt. Syst. Autom. Control*, vol. 1, no. 36, pp. 32–40, 2020.
- [24] J. Singh and B. Bhushan, "Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Oct. 2019, pp. 347–352.
- [25] P. Mustamo, "Object detection in sports: TensorFlow object detection API case study," Master's thesis, Dept. Faculty Sci., Univ. Oulu, Oulu, Finland, 2018.
- [26] H. D. Roger, "Font rasterization: The state of the art," in *From Object Modelling to Advanced Visual Communication*. Berlin, Germany: Springer, 1994, pp. 274–296.
- [27] W. Shengkai, J. Yang, X. Wang, and X. Li, "IoU-balanced loss functions for single-stage object detection," *Pattern Recognit. Lett.*, vol. 156, pp. 96–103, Apr. 2022.
- [28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [29] L. Wei, "SSD: Single shot multibox detector," in *Proc. Comput. Vis.-ECCV 14th Eur. Conf.*, Amsterdam, The Netherlands, Springer, Oct. 2016, pp. 21–37.
- [30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
- [31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.



SOON-BUM LIM received the B.S. degree from Seoul National University, South Korea, in 1982, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1983 and 1992, respectively. From 1989 to 1997, he was the Engineering Director for the Font Technology and Printer Division, Human Computer Inc, and Trigem Computer Inc. Since 1997, he has been an Assistant Professor with the Department of Computer Science, Konkuk University, South Korea. Since 2001, he has been a Professor with the Department of IT Engineering, Sookmyung Women's University, South Korea. His main research interests include web and mobile multimedia contents, user interface, assistive technology, computer graphics, and electronic publishing, such as font, eBook, and XML documents.



YUJIN LEE is currently pursuing the degree with the Department of IT Engineering, Sookmyung Women's University. Her research interests include deep learning, UI/UX, and computer graphics.



YOOJEONG SONG was born in Jeju, South Korea, in 1993. She received the B.S. degree in multimedia science and the Ph.D. degree in IT engineering from Sookmyung Women's University, in 2015 and 2021, respectively. She is currently an Assistant Professor with Semyung University, Jecheon, South Korea. Her research interests include artificial intelligence, system software, statistical analysis, and computational finance.

• • •