Published online on the journal's webpage: **http://journal.ittelkom-pwt.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 3 No. 2 (2023) 59 - 64 E-ISSN: 2809-8064

Minimalist DCT-based Depthwise Separable Convolutional Neural Network Approach for Tangut Script

Agi Prasetiadi[1*], Julian Saputra[2], Imada Ramadhanti[3], Asti Dwi Sripamuji[4], Risa Riski Amalia[5]

[1*,2,3,4,5]Faculty of Informatics, Institut Teknologi Telkom Purwokerto

[1*]agi@ittelkom-pwt.ac.id, [2]19102008@ittelkom-pwt.ac.id, [3]19102003@ittelkom-pwt.ac.id, [4]19102006@ittelkom-pwt.ac.id, [5]19102079@ittelkom-pwt.ac.id

Abstract

The Tangut script, a lesser-explored dead script comprising numerous characters, has received limited attention in deep learning research, particularly in the field of optical character recognition (OCR). Existing OCR studies primarily focus on widely used characters like Chinese characters and employ deep convolutional neural networks (CNNs) or combinations with recurrent neural networks (RNNs) to enhance accuracy in character recognition. In contrast, this study takes a counterintuitive approach to develop an OCR model specifically for the Tangut script. We utilize shorter layers with slimmer filters using a depthwise separable convolutional neural network (DSCNN) architecture. Furthermore, we preprocess the dataset using a frequency-based transformation, namely the Discrete Cosine Transform (DCT). The results demonstrate successful training of the model, showcasing faster convergence and higher accuracy compared to traditional deep neural networks commonly used in OCR applications.

Keywords: Tangut script, Optical character recognition, Depthwise separable convolutional neural network, Discrete cosine transform

1. Introduction

The Tangut script is a logographic writing system that was created in the 11th century by the Tangut people, a non-Han ethnic group that lived in what is now Gansu and Ningxia provinces in China. The Tangut script was based on the Chinese writing system, but it was significantly modified to reflect the sounds and grammar of the Tangut language [1]. The Tangut script was used for writing official documents, religious texts, and other materials until the fall of the Tangut empire in 1227 [2]. After that, the Tangut script fell into disuse and was eventually forgotten.

The Tangut script is a very complex writing system with over 6,000 characters. The characters are made up of a combination of strokes, which can be arranged in different ways to represent different sounds. The Tangut script is written top-to-bottom in lines and from right to left, and the characters are usually arranged in columns [3]. The characters are pronounced using a system of tones, which are similar to the tones used in Chinese. The Tangut language also has a number of unique grammatical features that distinguish it from Chinese.

The Tangut language is a tonal language, which means that the meaning of a word can be changed by changing the tone of its pronunciation. There are four tones in Tangut: high, rising, falling, and level. The Tangut language is also an agglutinative language, which means that words are formed by adding prefixes, suffixes, and infixes to a root word. This is in contrast to Chinese, which is a fusional language [4][5][6].

The reading of Tangut script involves analyzing the characters in terms of their semantic and phonetic components. The semantic component provides clues about the meaning of the character, while the phonetic component offers insights into its pronunciation. The grammar of the Tangut language is distinct from standard Chinese, featuring its own syntactic rules, word order, and grammatical structures [7][8].

The Tangut script exhibits distinctive features compared to the contemporary Chinese script. It is known for its complex characters, often comprising a greater number of strokes than those found in the standard Chinese scripts of the time. The Tangut characters are more elaborate and intricate, featuring a combination of strokes, curves, and intricate shapes. The visual

Received: 18-06-2023 | Revised: 14-07-2023 | Published: 01-08-2023

complexity of the Tangut script poses unique challenges for optical character recognition systems due to the intricacies involved in accurately recognizing and distinguishing these characters [9][10].

Previous studies have employed Deep Convolutional Neural Networks (DCNN) with four convolution layers to develop a Tangut character recognition system. This system was trained on a dataset containing over 100,000 labeled Tangut images and focused on recognizing the first 1,000 high-frequency Tangut characters [11]. The DCNN network was also utilized to automatically extract Tangut characters, enabling automated script annotation without manual intervention [12]. Other researchers have explored the use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), as well as their derivatives such as Region-based CNN (RCNN) [13], Faster RCNN [14], and Bidirectional Long Short-Term Memory (Bi-LSTM) [15], for the recognition of more commonly encountered Chinese characters, including standard Chinese [16] and handwritten forms [17]. Furthermore, previous investigations have suggested that wider convolution networks with larger filters and shallower layers may lead to improved performance [18].

In this study, a different approach to optical character recognition (OCR) for Tangut script is proposed. Instead of increasing the depth of the neural network, a strategy involving shorter and slimmer layers is employed, accompanied by preprocessing the dataset. The dataset consists of three types of preprocessed input data: raw data, Fourier-based transformed data, and Discrete Cosine-based transformed data. The comprehensive set of Tangut characters is obtained from Google Noto fonts, and each character is augmented with 20 different variations. Annotation of each character class is performed based on the associated Unicode code. Surprisingly, the experimental results demonstrate outstanding performance, as the minimalist model surpasses the performance of deeper networks.

2. Research Methods

2.1. *Tangut Script Dataset*

The Tangut characters are assigned a specific Unicode block, U+17000–U+187F7, encompassing a total of 6,136 characters [19]. The inclusion of Tangut script in the Unicode Standard occurred in June 2014, coinciding with the release of version 7.0 of the standard, highlighting the growing recognition and importance of this ancient script in digital contexts.

To facilitate the training and evaluation of the OCR model, a dataset comprising Tangut script characters was prepared. All the characters used in this research were extracted from the Google NotoSerifTangut-Regular font. In order to enhance the model's robustness and generalization, all characters were augmented to create 20 different variations. This augmentation involved applying random distortions, rotations, and zoom levels to the characters. The dataset was split evenly, with half of the augmented characters used for training the OCR model and the remaining half for testing, enabling accurate evaluation of the model's performance on unseen Tangut script characters.

2.2. *Depthwise Separable Convolutional Neural Network*

The Depthwise Separable Convolutional Neural Network (DSCNN) is a refined CNN layer that optimizes the convolutional operation by decomposing it into two smaller convolutions: depthwise convolution, which operates on individual input channels independently, and pointwise convolution, which combines the resulting output channels. This technique effectively accelerates the convolution process, enhancing computational efficiency [20].

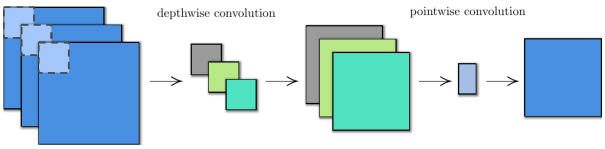

Figure 1. Depthwise Separable Convolutional Neural Network calculation step.

Instead of performing the conventional full-fledged convolution calculation, where a matrix $\mathbf{M}$ is convolved directly with a kernel k resulting in $\mathbf{M}' = \mathbf{M}*k$, the depthwise separable convolution approach divides the convolution process into separate steps. As illustrated in Figure 1, the first step involves depthwise convolution, where three distinct kernel matrices are convolved individually with their assigned channels. This results in three separate channels. Subsequently, the values of each channel are mixed using a vector kernel in the pointwise convolution, resulting in a single matrix. This process can be scaled up to produce multiple diverse results based on different combinations of kernels. We can summarize this process using the following formula: $\mathbf{M}' = [\mathbf{M_R}*k_R, \mathbf{M_G}*k_G, \mathbf{M_B}*k_B]*k_p$, where k_R, k_G, k_B, and k_p represent the kernel matrices for the red, green, and blue channels, as well as the overall pointwise convolution process, respectively.

2.3. *Frequency Domain*

The frequency domain serves as a means of representing a signal by its constituent frequency components. Analyzing signals in the frequency domain involves converting them from the time domain to the frequency domain using mathematical techniques like the Fourier Transform and the Discrete Cosine Transform. It provides valuable insights into the frequency composition of a signal, which may not be readily

discernible from its representation in the time domain. This conversion enables the extraction of valuable information about the signal's frequency content and distribution. Unlike the time domain representation, where signals are described in terms of their amplitude variations over time, the frequency domain reveals the specific frequencies present in the signal and their respective magnitudes.

The Fourier Transform is a mathematical operation that decomposes a signal into a sequence of sine and cosine functions of different frequencies. Formula (1) shows the calculation of Fourier Transform. By applying the Fast Fourier Transform (FFT), we can obtain a sequence of complex numbers representing the signal's frequency domain. However, this complex number representation poses certain challenges. Firstly, it doubles the amount of data, increasing computational requirements. Additionally, deep learning networks didn't differentiate between the real and imaginary components of complex numbers. To address these issues and enhance simplicity, a specific scheme is employed. After the data is converted into its Fourier representation, only the real values of the converted Fourier Transform are utilized.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i \cdot k \cdot n / N}, \quad \text{for } k = 0, 1, \ldots, N-1 \quad (1)$$

In this formula, x represents the input signal, X represents the transformed signal (complex numbers), k represents the frequency bin index ranging from *0* to *N-1*, n represents the time-domain sample index ranging from *0* to *N-1*, and N represents the length of the input signal.

The Discrete Cosine Transform (DCT) is a member of the Fourier Transform family, specifically designed to decompose a signal into a sequence of cosine functions. The calculation of the DCT can be represented by Equation (2). Unlike the Fourier Transform, the DCT produces a representation that consists entirely of real numbers. This characteristic makes the DCT representation more straightforward and easier to interpret. Notably, the structure of the converted DCT data is distinctive, typically beginning with a larger magnitude coefficient followed by mostly smaller coefficients.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right), \quad k = 0, 1, 2, \ldots, N-1 \quad (2)$$

In this formula, x_n represents the input signal, X_k represents the k-th coefficient in the resulting DCT sequence, k represents the frequency bin index ranging from *0* to *N-1*, n represents the time-domain sample index ranging from *0* to *N-1*, and N represents the length of the input signal.

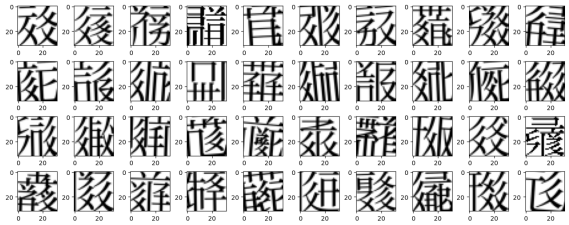

Figure 2. Random Sampling of Tangut Script Characters from Dataset.

Figure 2 displays a random sampling of the augmented Tangut script characters extracted from the dataset. In Figure 3, the Tangut script characters are converted into their Discrete Cosine domain representation. Notably, the upper left portion of the converted data exhibits significantly higher magnitude values in comparison to the remaining portions.

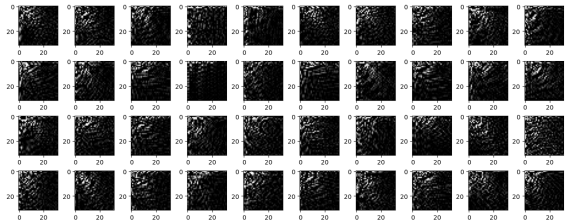

Figure 3. Random Sampling of Tangut Script Characters Converted into Discrete Cosine Domain using DCT.

2.4. *Architecture*

In this research, we constructed a model consisting of two distinct types of architecture: a multi-layer design, and a minimalist layer design. Common intuition suggests that deeper architectures tend to exhibit superior performance in object recognition tasks. While this holds true for complex objects present in natural images, it may not necessarily apply to the task of optical character recognition (OCR). Unlike natural objects, characters in OCR typically consist of simple monochromatic strokes in black and white.

Wide convolutional layers have the potential to accelerate convolutions by utilizing fewer, but thicker layers with an increased number of kernels per layer. This technique proves effective as the addition of more convolutional layers can significantly slow down the model. Interestingly, it is possible to further expedite this process by employing slimmer kernels. Therefore, in this study, we systematically reduced the number of layers in the model, one at a time, to investigate their impact on accuracy. Additionally, we closely monitored the convergence rate of each model to assess the effectiveness of these layer adjustments.

Table 1. Model's architectures and specifications.

Layer(s)	Number Layer	Input	Pooling	Combination	Model Number
Multi	3	Raw	Yes	SC2PSC4P	1

				SC16P	
			No	SC2SC4SC16	2
	2			SC2SC4	3
Minimalist	1	Raw	No	SC2	4
		Fourier (Raw)			5
		DCT			6
		DCT (Low-Freq)			7
		DCT (High-Freq)			8

In this study, we employed a specific naming convention to represent the configurations of the models. The abbreviation "SC" denotes the Depthwise Separable Convolutional Neural Network (DSCNN) layer, while "P" refers to the Pooling layer. Therefore, a label such as "SC2PSC4P" indicates a model with one layer of DSCNN using 2 filters, followed by a Pooling layer, another layer of DSCNN using 4 filters, and finally another Pooling layer. All models in this research were connected to a flattened layer and converged directly into softmax dense nodes, representing the probabilities of different character classes.

Table 1 shows the model's architectures and specifications utilized in this study. The first and second models utilized a more extensive layer design, consisting of three layers, with one incorporating a Pooling layer and the other without it. The third model was a simplified version, comprising only two layers without a Pooling layer. These three models can be categorized as multi-layer models.

On the other hand, the remaining models were minimalist models, consisting of a single layer with only two filters. The distinction among these models lies in the type of input they were fed: raw data, Fourier-based representation data, or Discrete Cosine-based representation data. The fourth model is referred to as "SC2," while the fifth model is known as "Real Fourier-based SC2" or "RFSC2." The sixth, seventh, and eighth models were fed with Discrete Cosine-based representation data. The sixth model received input from all frequencies of the DCT-transformed data, hence referred to as "DCT All frequency-based SC2" or "DASC2." The seventh model utilized low-frequency components and is referred to as "DCT Low frequency-based SC2" or "DLSC2." Lastly, the eighth model utilized high-frequency components and is denoted as "DCT High frequency-based SC2" or "DHSC2."

3. Results and Discussion

In the results section of our study, we observed that the inclusion of a pooling layer in the architecture slowed down the convergence speed towards achieving small loss. Interestingly, architectures without a pooling layer reached lower loss faster than those with a pooling layer. We believe this effect may be attributed to the size of the Tangut script, which has already been resized to a point where anti-aliasing is minimized. Further resizing could lead to aliasing issues and potential similarity collisions. Figure 4 shows overall loss performance of our models. The models undergo training for a limited number of epochs, and the training data is divided into two separate batches.

Furthermore, our findings revealed that reducing the number of layers positively impacted the loss performance. Models with fewer layers demonstrated improved loss compared to models with a larger number of layers. Additionally, models with fewer layers reached lower loss levels at a faster rate. This finding is particularly advantageous as it allows for the training of OCR models for broader character recognition across various languages in a shorter period.

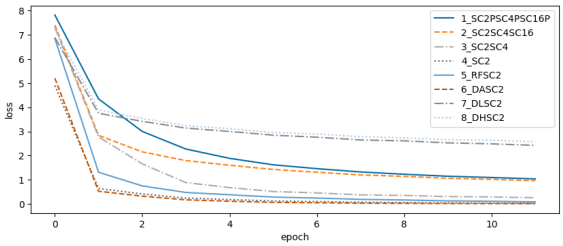

Figure 4. Comparative Performance Analysis of 8 Models based on Loss.

In the context of architectures with minimalist layers, it is worth noting that the most effective preprocessing transformation for script image datasets is the Discrete Cosine Transform (DCT), followed by the raw (non-transformed) data, and then the real-valued Fourier Transform. The performance of DCT-based and raw-based training methods is comparable, while Fourier-based training is slightly slower. This difference in speed may be attributed to the utilization of only the real values of the Fourier Transform, disregarding the imaginary part.

Comparing different training approaches, the real-valued Fourier-based training outperforms both the low-frequency and high-frequency DCT-based training methods. Moreover, the model with pooling layers exhibits superior performance compared to these individual frequency-based approaches, such as low-frequency and high-frequency only dataset. Therefore, it can be inferred that including all frequencies in the training process is beneficial for OCR models, as some writing systems contain subtle differences between characters that may appear visually similar at first glance.

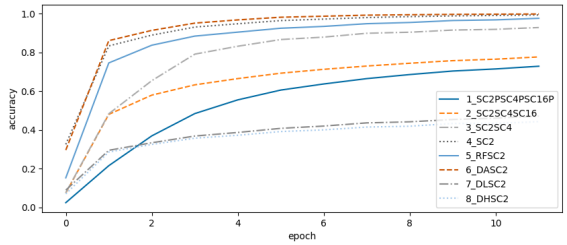

Figure 5. Comparative Performance Analysis of 8 Models based on Accuracy.

The overall accuracy performance of all models is depicted in Figure 5. Notably, only two models exhibit an initial accuracy higher than 20\% during their first epoch: namely the minimalist model with raw input data and the model trained on DCT transformed data. In contrast, the models with multiple layers demonstrate an initial accuracy performance close to 0\%. Interestingly, all models consistently display a similar trend in their accuracy performance as observed in their respective loss performance.

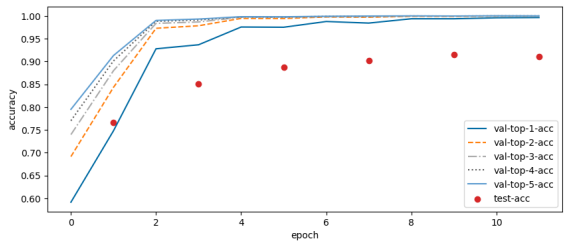

Figure 6. Top K Accuracy of 6th Model, DASC2.

In order to assess the detailed performance of our best model, DASC2, we examined its performance against the validation data, which accounted for 10\% of the training data. Figure 6 presents the results obtained from conducting the Top K Accuracy test on the model.

The Top K Accuracy testing methodology allows us to evaluate the model's performance not only based on the most probable class prediction but also by considering the next k-th most probable classes. For instance, in the case of top 3 accuracy, we assess the model's ability to correctly predict the class with the highest probability as well as the second and third highest probabilities. This technique provides insights into the model's effectiveness in distinguishing between different script classes. Furthermore, we examined the model's capability to read the testing dataset, and it was observed that the model exhibited signs of overfitting during the training process. This implies that diversifying the writing styles within each script category in the dataset would enhance the model's ability to generalize and accurately recognize different script variations.

As final note, the result also suggests that the construction of an optical character recognition model does not necessarily require a wider layer format, where the number of filters is increased within smaller layers. Surprisingly, slimmer layers with only two filters yielded exceptionally high accuracy models in our study. This indicates that focusing on the simplicity of the architecture, rather than the width of the layers, can lead to excellent performance. However, it is important to note that there are drawbacks associated with this technique. The burden of decision-making is primarily placed on the dense layer. Overall, our study highlights the impact of pooling layers, the benefits of reducing layer complexity, and the importance of considering simplicity in the design of optical character recognition models.

4. Conclusion

In this study, we have trained and tested deep learning architectures with multi and minimalist layers for reading Tangut script. From our findings, it can be concluded that for script reading purposes, the absence of a pooling layer may result in better performance in terms of loss and accuracy compared to models with a pooling layer. Additionally, having fewer layers in the architecture also appears to enhance the training performance. In the case of the minimalist model, employing slimmer layers led to improved classification results. However, it is important to note that preprocessing of the character image dataset is essential. In our experiments, applying the Discrete Cosine Transform (DCT) as a preprocessing step yielded better performance and faster convergence compared to using the raw data. Despite achieving promising results, our best model still exhibited overfitting, as it achieved only around 90\% reading accuracy on the testing dataset. To address this, further efforts should be made to diversify the script variation in the dataset, allowing the model to generalize better. Moreover, future studies should explore the potential of various frequency-based transformations as a means to enhance the performance of the models.

References

[1] N. Tranter, "Script 'borrowing', cultural influence and the development of the written vernacular in East Asia," in Language change in East Asia, pp. 180-204, 2001.

[2] R. W. Dunnell, "Translating history from Tangut Buddhist texts," Asia Major, vol. 22, pp. 41-78, 2009.

[3] A. West, M. Everson, X. Han, C. Jia, Y. Jing, and V. Zaytsev, "Proposal to encode the Tangut script in the UCS," UC Berkeley: Department of Linguistics, 2014. [Online]. Available: https://escholarship.org/uc/item/41v0851f.

[4] Brosse, David. "The Tangut Script." In The Oxford Handbook of Writing Systems, edited by Peter T.

Daniels and William Bright, 592–606. Oxford: Oxford University Press, 2009.

[5] Fang, Hao. "The Tangut Language." In The Languages of China, edited by Graham Thurgood, 251–278. London: Routledge, 2008.

[6] Hsu, Cho-yun. "The Tangut Script." T'oung Pao, Second Series 50 (1962): 41–70.

[7] Sun, X. (2016). Basic Tangut Grammar. Language Documentation \& Conservation, 10, 158-178.

[8] Gong, H., Zhou, J., and Fu, H. (2019). Tangut Hierarchical Structure Recognition Using Subword-level Convolutional Neural Networks. In International Conference on Document Analysis and Recognition (pp. 1325-1330). IEEE.

[9] Franke, H., and Chen, Y. (2016). Tangut Manuscripts from Khara-Khoto. In Encounters between Chinese Culture and Christianity (pp. 81-87). Brill.

[10] Zhou, X., and Zuo, Z. (2018). A Tangut Character Recognition Method Based on Improved Gradient-Based Features and Convolutional Neural Networks. IEEE Access, 6, 34076-34085.

[11] G. Zhang and X. Han, "Deep learning based tangut character recognition," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, China, 2017, pp. 437-441, doi: 10.1109/ICSAI.2017.8248332

[12] G. Zhang and Y. Zhao, "Learning Radicals From Tangut Characters," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 2018, pp. 373-378, doi: 10.1109/ICSAI.2018.8599386.

[13] D. Zhu, Y. Fang, Z. Min, D. Ho and M. Q. . -H. Meng, "OCR-RCNN: An Accurate and Efficient Framework for Elevator Button Recognition," in IEEE Transactions on Industrial Electronics, vol. 69, no. 1, pp. 582-591, Jan. 2022, doi: 10.1109/TIE.2021.3050357.

[14] X. Gao, F. Yang, T. Chen and J. Si, "Chinese Character Components Segmentation Method Based on Faster RCNN," in IEEE Access, vol. 10, pp. 98095-98103, 2022, doi: 10.1109/ACCESS.2022.3206832.

[15] J. Hu, T. Guo, J. Cao and C. Zhang, "End-to-end Chinese text recognition," 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 2017, pp. 1407-1411, doi: 10.1109/GlobalSIP.2017.8309193.

[16] X. Gao, F. Yang, T. Chen and J. Si, "Chinese Character Components Segmentation Method Based on Faster RCNN," in IEEE Access, vol. 10, pp. 98095-98103, 2022, doi: 10.1109/ACCESS.2022.3206832.

[17] X. Liu, B. Hu, Q. Chen, X. Wu and J. You, "Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4637-4648, Nov. 2020, doi: 10.1109/TNNLS.2019.2956965.

[18] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint arXiv:1605.07146, 2016.

[19] Tangut (Xixia) Script and Unicode (L2/07-289 = WG2/N3307)," Unicode, May 2007. [Online]. Available: https://unicode.org/wg2/docs/n3307.pdf. [Accessed: Jun. 15, 2023]

[20] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251-1258, 2017.