

Estimating the number of chases used for printing books with movable metal types

Dongkeun Kim¹, Maaz Ahmed¹, and Kang-Sun Choi¹

¹Future Convergence Engineering Major, KOREATECH, South Korea

Abstract

Many books were printed with movable metal types during the Joseon Dynasty (A.D. 1392-1897), and even early books are older than the Gutenberg Bible which was the earliest book printed using movable metal type in Europe (A.D. 1455). However, there is little information about the printing technology, such as the scale of metal type at that time and the management system. In this paper, we propose an algorithm to estimate the number of chases simultaneously used to print a book with movable metal types, which can provide not only insights into the level of printing technology at that time but also important clues for estimating the number of movable metal types. In contrast to previous studies of chase count estimation based on the subjective comparison, we propose an algorithm to estimate the number of chases using the similarity of character spacing distribution in the central part of the printed sheets, assuming that the metal types for the central part were set once and reused without replacement during the production of the book. The central part of each sheet in one book is cropped. After rotational deviations of all the central part images are removed using the principal component analysis, the images are binarized. Morphological operations are performed to remove noise and facilitate character analysis. The profile histogram of the binarized image is used as a probability distribution for character spacing, and the Wasserstein distances among the profile histograms of all the central part images in the book are calculated to obtain a similarity matrix of the central part images. By performing an eigenvalue analysis using spectral clustering with the similarity matrix, we determine the number of clusters of the central part images, which indicates the number of chases simultaneously used in printing the book. When applying the proposed algorithm to the images of "Neungumgyeoung" Volume 4 (A.D. 1462), "Seokbosangeol" Volume 6 (A.D. 1446), and "Worin cheongangjigok" Volume 1 (A.D. 1447), it was estimated that 3, 6, and 4 chases were used, respectively. To verify the reliability of the results, subjective classification of the core part based on the size or style of the characters was performed for "Neungumgyeoung" Volume 4, and it was classified into 3 groups, consistent with the results of the proposed algorithm. This proposed method can be utilized for the analysis of various metal movable chases.

1. Introduction

Metal chases refer to the use of metal typefaces, which are important tools in the printing process. Metal types are typically made of copper. They are used to transfer ink onto printing media, such as paper. Metal chases are primarily used in the printing industry and were historically employed for typeset fonts and characters. In addition to printing purposes, metal types have been used to create new documents or to indicate ownership. Metal types are usually small blocks that represent the shape of each character. It is still used for its historical value, in artistic works, or in specialized printing tasks. Metal movable type in Korea, known as "Gap-inja" [Kim09] or "Eulhaeja" [LY07] was primarily used during the Joseon Dynasty. It played a significant role in the development of printing culture by facilitating the production and distribution of books and documents. Metal movable type in Korea was predominantly made of copper. Each individual character was meticulously crafted and assembled. To print, ink was applied to the type, and pressure was exerted to transfer the ink onto the paper. Compared

to traditional woodblock printing, metal movable type in Korea offered greater precision and efficiency, enabling mass production. It played a vital role in the widespread dissemination of knowledge and culture. Today, Korean metal movable type is preserved as a cultural heritage, representing a significant milestone in the history of printing.

In metal letterpress printing, small pieces of metal (usually copper) are used as typefaces. Each typeface represents the shape of a letter or symbol and is used to convey the content of the document. The process of metal letterpress printing involves the following steps [Yoo22]:

Metal typeface production: Firstly, metal typefaces corresponding to each character or symbol are produced. This is usually done by carving the desired shapes into pieces of cut metal. The typefaces need to have engraved characters or symbols.

Typesetting: Typesetting determines the layout and format of the document to be printed. It involves preparing the typefaces for the

desired composition of the document. Typefaces are selected, combined to form sentences or paragraphs, and adjusted with respect to height and spacing.

Inking: Inking involves applying ink to the typefaces using an inked printing plate. The ink only covers the character portions of the typefaces.

Printing: After inking the printing plate, the typefaces transfer the inked characters onto the printing substrate (typically paper) using the pressure of a printing press. The pressure from the printing press ensures that the typefaces transfer the ink and print on the surface.

Drying and post-processing: Once the printing is complete, time is allocated for the ink to dry. Afterward, any necessary post-processing tasks can be performed on the document. This may include folding, trimming, binding, and other finishing techniques. There are several important parts of Korean chases which were used in the Joseon Dynasty that give us information about the printed book. An example of typesetting and a complete example of typesetting is shown in Figure 1.

The outer lines surround the four sides of the book's pages, specifying the size of the printed portion and specifying the text area. The center column is the folded center section when the book is folded to divide it into two sides, and it contains the page, number of volumes, and book title. A dividing line is a line drawn to separate each line in the book body, and in the case of metal type, it is a part that fits inside the dividing line and allows you to control the slope and position of the letters.

Estimating the number of chases used to print a metal-type book provides a good indication of the state of the art of printing at the time and provides important clues to the size of the presses in operation. As a side project, we are working on comparing typefaces to estimate the type and number of each typeface used to produce a single metal type book. A companion piece to this research is the study of type count estimation. However, it is difficult to estimate the number of chases used in the printing process when there is no record of who produced the metal type. While many books were produced using metal types, there are few records of how many chases or typefaces were used.

One way to estimate the number of chases is to compare the type used to produce metal-type books. However, estimating how many typefaces were used to print metal types is another challenge. In this study, we estimate the number of typefaces by analyzing "Pan-Sim", the central part of a page in a metal type book. The "Pan-Sim" is the part of the book that contains the name, volume, and page information. It is the only part of the book that uses the same Chinese characters on all the pages. We also assume that the information about the book's title and number of volumes, except for the page number, would have been fixed and not separated after the type was fixed. Using these features and analyzing the similarity of letter spacing, it is possible to estimate the number of chases used to produce the book. The entire process including the network creation was performed using MATLAB.

2. Related Work

2.1. Wasserstein Distance

Wasserstein distance, also known as Earth Mover's distance (EMD) [RTG00] or Kantorovich-Rubinstein distance, is a mathematical metric used to measure the dissimilarity between two probability distributions [ACB17] [EC23]. It takes into account the underlying structure of the distributions and provides a measure of the minimum "work" or effort required to transform one distribution into another. The Wasserstein distance is computed by finding the minimum cost of moving the mass from one distribution to another, where the cost is defined as the distance times the amount of mass being moved. This transportation problem is solved using linear programming techniques or optimal transport algorithms. The Wasserstein distance has several desirable properties. It is a true metric, satisfying the properties of non-negativity, symmetry, and triangle inequality. Unlike other distance measures, such as the Euclidean distance or Kullback-Leibler divergence (KL divergence), the Wasserstein distance considers both the geometric and probabilistic structure of the distributions, making it particularly useful in applications involving probability distributions or image analysis.

2.2. Bhattacharyya Distance

Bhattacharyya distance is a measure of similarity between two probability distributions [Kai67]. It was developed in Bayesian statistics and is used to assess the degree of overlap between two distributions as a measure of similarity. Bhattacharyya distance is used to measure the similarity between probability density functions (PDFs) of two distributions. It calculates the degree of overlap between the two distributions, considering differences in covariance and variance. When two distributions completely overlap, the Bhattacharyya distance is 0, and when they have no overlap, the distance reaches its maximum value. Bhattacharyya distance is calculated using equation 1:

$$D_B(p, q) = -\ln(BC(p, q)), \quad (1)$$

where BC is the Bhattacharyya coefficient.

The Bhattacharyya coefficient is defined as:

$$BC(p, q) = \int \sqrt{p(x)q(x)} dx, \quad (2)$$

where $p(x)$ and $q(x)$ are the probability density functions of the two distributions being compared. D_B represents the Bhattacharyya distance. Bhattacharyya distance returns values greater than or equal to 0, with smaller values indicating greater similarity between the distributions. It is widely used in pattern recognition, computer vision, statistical analysis, bioinformatics, and other fields for comparing and classifying probability distributions [TCXH22].

2.3. KL Divergence

KL divergence is a measure of the difference between probability distributions [Shl14]. Given two probability distributions, P and Q ,

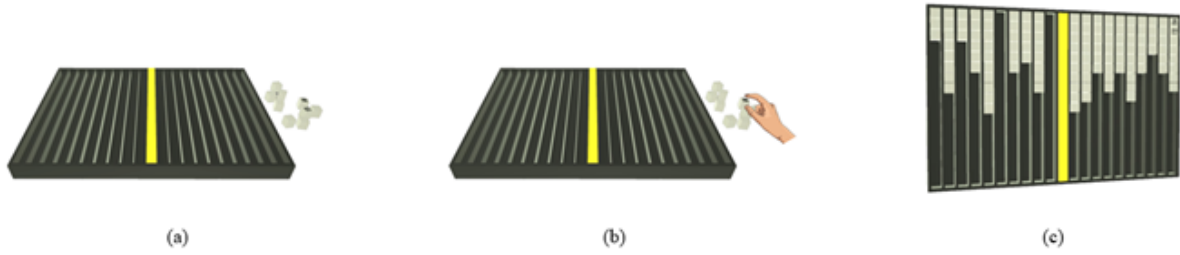


Figure 1: Typesetting process. (a) A chase and movable metal types. The Pan-Sim part where the book title and the sheet number are placed is indicated in yellow. (b) Typesetters find appropriate metal types and insert them into the chase. (c) Completion of typesetting. After this process, inking, printing, and binding work are performed.

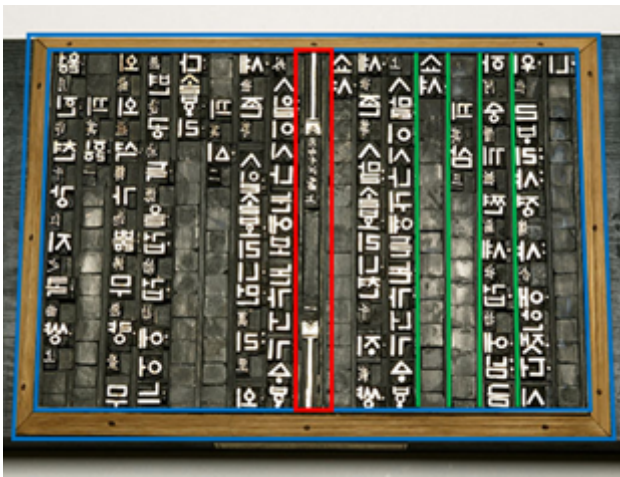


Figure 2: An example of typesetted chase. The blue part is the outer lines, the red box is the center column (Pan-Sim), and the green lines are dividing lines.

KL divergence quantifies the information loss when approximating P with Q. KL divergence is defined as follows:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log_b \left(\frac{P(x)}{Q(x)} \right) \quad (3)$$

In equation 3, P and Q represent the two probability distributions, and x represents the random variable. KL divergence is the weighted sum of the logarithm of the ratio between P(x) and Q(x), weighted by P(x). KL divergence can be interpreted as the distance or difference between two probability distributions. However, it is an asymmetric measure, so $D_{KL}(P \parallel Q)$ and $D_{KL}(Q \parallel P)$ generally yield different values.

2.4. Spectral Clustering

Clustering is a technique for grouping data points with similar characteristics together. It is used to identify the inherent structure of

data and categorize data based on similarities. Clustering is widely used in various fields for data analysis and pattern recognition. Clustering algorithms generally vary in how they measure the similarity between data points and determine the centers of clusters. Common clustering algorithms include k-means [NXY10], hierarchical clustering [PSJ15], and DBSCAN [Den20]. Spectral clustering is a machine learning algorithm used for clustering or grouping data points based on their similarity or proximity [NJW01]. It is particularly effective when dealing with data that doesn't have well-defined clusters or when the clusters have complex shapes. The spectral clustering algorithm works by transforming the data into a lower-dimensional space and then applying a clustering technique, typically K-means, to the transformed data. Spectral clustering leverages the spectral properties of the Laplacian matrix to capture the underlying structure of the data, even when the clusters are not linearly separable in the original feature space. By utilizing the lower-dimensional space and spectral techniques, it can discover meaningful clusters in complex datasets. Overall, spectral clustering is a powerful technique for unsupervised learning tasks where the goal is to discover hidden patterns or groups in the data. It has been widely used in various fields such as image segmentation, community detection, and document clustering [WNWL19].

2.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction and noise removal in data [JC16]. PCA focuses on finding the most important underlying factors that best describe the data. It is useful for reducing the dimensionality of data or explaining variability [SSA*14]. By using fewer principal components than the original variables, PCA allows for visualization and computational tasks with high-dimensional data. It can also be used to refine data by removing noise or irrelevant information. PCA is commonly applied in data analysis, pattern recognition, image processing [LCVM09], finance, and various other fields as a statistical tool.

2.6. Previous Research

Previous studies have used character similarity by human impressions of the letters and their surrounding outlines instead of computerized analysis [Lee07].

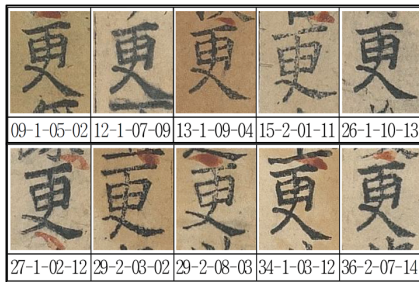


Figure 3: Examples of characters extracted from *Jikji*. It is all done by the camera.

In the case of metal type, which is used to print metal-type books, a single letter was printed using a variety of metal typefaces. When analyzing characters by human impressions, we need to manually extract all the characters one by one, and there are 14,000 characters in *Jikji*. To analyze this, the authors estimated the number of typefaces used in a book by manually extracting and analyzing features such as the size of the letters and the curvature of the letters. Typical methods include comparing and categorizing the printing shapes of the outline and analyzing the letters in the center column Pan-Sim as shown in Figure 4. However, even if the horizontal and vertical frames that make up the outline are separated, there are problems that make the judgment unreliable, such as the effect of surface tension while inking, which causes the print to stick.

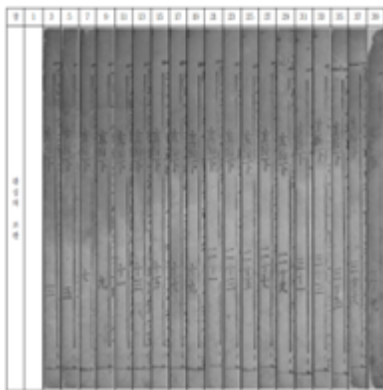


Figure 4: Used Pan-Sim area in previous research.

3. Data Collection

In this research, we used high-resolution scanned images of "*Neungumgyeoung*" Volume 4 (Ne-V4), "*Seokbosangjeol*" Volume 6 (Se-V6), and "*Worin cheongangjigok*" Volume 1 (Wo-V1). Also, different from the previous research, we use only characters from Pan-Sim instead of all of the parts of the Literature.

3.1. Used scanner for making datasets

In order to scan the books, we used a scanner that has an optical resolution of 600 dpi useful for generating high-resolution datasets.

3.2. Scanning methods

The metal-type books are printed on a large sheet of paper and folded in half to create double-sided pages. These pages are then bound together with string to form a book as shown in Figure 7. Due to the structure of these books, there are two ways to scan a book: disassemble the book completely, remove the double-sided pages, and scan them, or scan the book directly, page by page, while it is still assembled. In the case of Ne-V4 and Se-V6, the method of scanning each page directly without disassembling the book was utilized whereas in the case of Wo-V1, the method involving disassembling the book and scanning both sides at once was used.

3.3. Scanned Books

Ne-V4 (100 pages), Se-V6 (47 pages), and Wo-V1 (71 pages) were scanned. All three books were made with metal types during the Joseon Dynasty and are now a cultural heritage designated as a national treasure of Korea.

4. Methodology

In this chapter, the process for analyzing the similarity of character spacing using a high-resolution scanned dataset is described. Pan-Sim is extracted from the dataset and binarized to clearly separate the letters from the background. Afterward, the noise generated by the binarization process is removed, with the image resized to fit into the network, and then the slope of the image is unified. For the similarity measure of character spacing analysis, the profile histogram is used as the probability distribution of letters, and then the similarity of character spacing is obtained using various distance techniques, and finally, the number of clusters is estimated using Spectral Clustering. The final number of clusters is the number of chases used to print the metal type book. The entire process is illustrated in Figure 6.

4.1. Extraction of the Pan-Sim from the Datasets

The process of extracting Pan-Sim presents challenges depending on the dataset. In the case of Ne-V4, and Se-V6, which were scanned as one page per scan, where the Pan-Sim part is located at the edge of the book. To tackle this problem, we create two-page scanned data to one-page data with coordination as represented in Figure 5. After this preprocessing step, we extract the portion of the Pan-Sim part that needs to be used from the scanned high-resolution dataset. This can be done using Photoshop as demonstrated in Figure 8. In the Pan-Sim section, the edges of the books that have been used for a long time, some of the printing has been erased due to wear and tear. If the two pages are matched and combined in this state, problems such as duplication and loss of the Chinese characters of the judgment part may occur in the created dataset. In this case, the next step, binarization of the input image, is highly noisy. Analyzing the similarity of characters using such an inaccurate dataset can lead to unreliable results. (Figure 9) Therefore, in this study, we analyze the similarity of character spacing instead of character similarity.

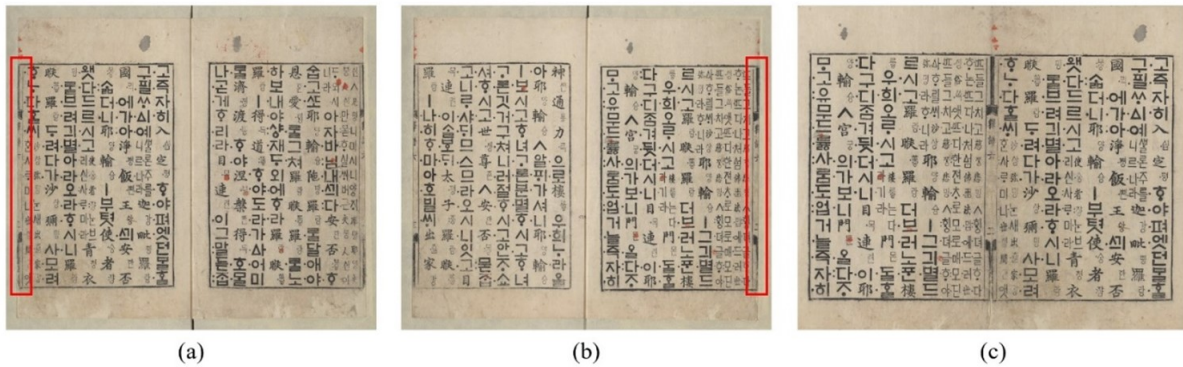


Figure 5: Figures (a) and (b) are examples of datasets scanned from a book spread wide open, showing Pan-Sim at the edges of the red boxes. To use it correctly, the left page in (a) and the right page in (b) must be coordinated like in Figure (c).

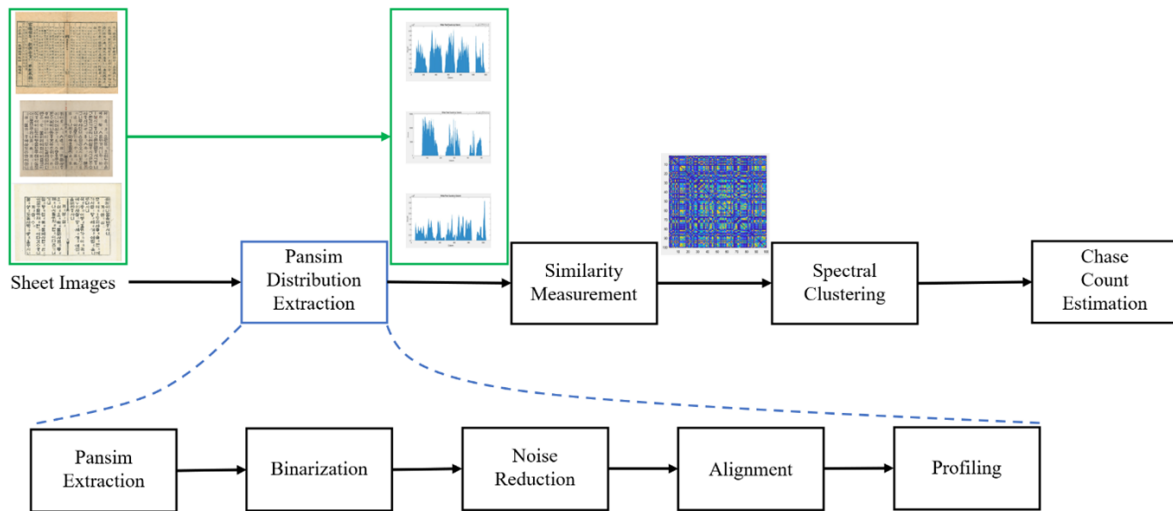


Figure 6: The entire process of the proposed method.

4.2. Binarization and noise reduction

After, extracting the Pan-Sim from the metal type book, the binarization process is initiated. The reason for the binarization process is to clearly distinguish the text and background areas of the created dataset by making the values 0 (black) and 1 (white) respectively. The results with simple binarization contain some noise due to the paper material of the dataset. To remove this noise, we adjust the threshold value appropriately or alternate morphology (erosion and dilation) [HSZ87].

4.3. Image Alignment Using PCA

Aligning the gradient of the image through PCA as described in Section 2.5. The main reason for unifying the gradients of the images is that the gradient of the data varies depending on the way people scan the book, and using it directly for the network gives inaccurate results. When PCA is run on each binarized image, it

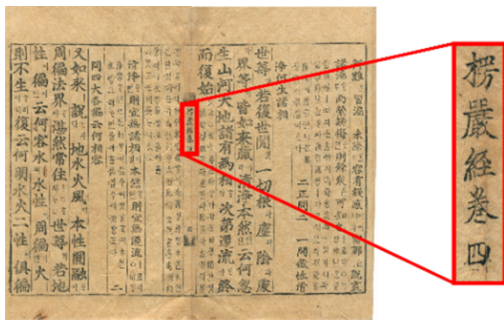
stores the location and information of all the 0 and 1 regions in the image. By extracting the regions of 1s and using the arc tangent, we get an idea of how much the character placement is skewed within a single image. If we rotate the resulting slope in the opposite direction, we observe see that the slope of the letter placement is nearly zero, which is better for analyzing letter spacing similarity.

4.4. Profile Histogram

After completion of the preprocessing steps discussed, we then start the process of analyzing character spacing similarity. In order to use different distances, we first need to create a probability distribution table. The probability distribution table we use is the character profile histogram. While a normal histogram analyzes the color of every pixel from 0 to 255 and shows how many of each color there are, a profile histogram counts how many pixels have a user-specified color horizontally when an image is inserted. Since the



Figure 7: Green area shows the part bounding the book using a red string.



(a)



(b)

Figure 8: (a) Pan-Sim part which is used in this research, (b) examples of Pan-Sim from Ne-V4, Se-V6, and Wo-V1 from the left.

input image consists of 0s and 1s, the number of 1s is counted horizontally and plotted on a graph. The distribution table of 1s is used as the probability distribution table for this study.

4.5. Calculation of similarity using several Distances

Various distance equations used are used to obtain character spacing similarity values for deep learning. Typical distance equations used are the Bhattacharyya distance and the Wasserstein distance. We use the profile histogram created from a probability distribution, but we need to perform another step before that. In our dataset,

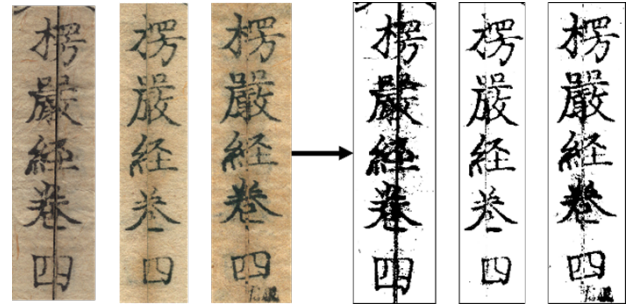


Figure 9: Problems with binarizing images in which the image is duplicated.

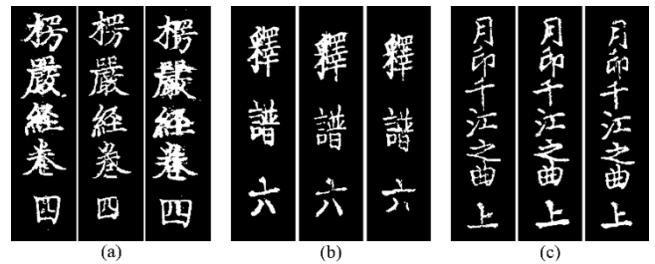


Figure 10: Images after binarization and noise reduction. (a) Ne-V4, (b) Se-V6, (c) Wo-V1.

the images are uniform in size and slope, but the letters have different starting points in each image. In the case of the Profile histogram, we can observe that the horizontal and vertical axes of the graph are different depending on the size of the image, and the start points of the letters are different, as shown in Figure 12a. To calculate the correct distance, we need to unify the start points of the letters in one dataset. If the user cuts and modifies the video arbitrarily, it would take more time to create the dataset. However, in our case, we compare the profile histogram of an image with the profile histogram of another image by moving it by 50 pixels to the left and right, respectively, and saved the result that shows the highest value of the DISTANCE result through a total of 100 movements as the final result to increase the accuracy of the results. In addition, the similarity result is used in the next step of spectral clustering, so all images are compared with each other (Figure 14). This process was performed for Ne-V4, Se-V6, and Wo-V1, respectively, and the results can be seen in Figure 12. To increase the reliability of the Distance results only results with a similarity score of 0.97 or higher were saved.

4.6. Spectral Clustering

The character spacing similarity score obtained in the previous step is used to perform spectral clustering, which divides the clusters based on the values near zero or jumps within the values. For example, if the number of zeros and jumps in a total of 10 indexes is 5, the number of clusters is estimated to be 5.

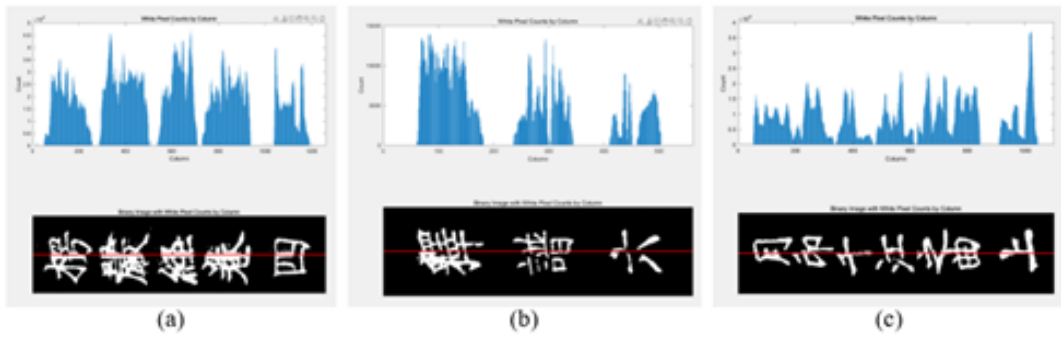


Figure 11: Results of profile histogram for each Pan-Sim. (a) Ne-V4, (b) Se-V6, (c) Wo-V1.

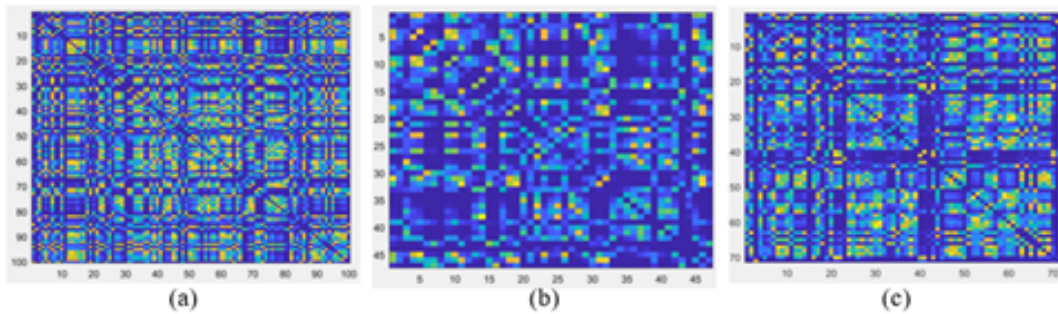


Figure 12: Figure shows the similarity results when using Wasserstein distance. The lighter the color, the higher the similarity between the two images; the darker the color, the lower the similarity. (a) Ne-V4, (b) Se-V6, (c) Wo-V1



Figure 13: Example images after unifying gradients. The red lines show the gradient is straight.

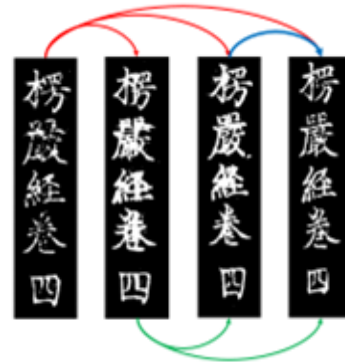


Figure 14: To get the distance for the entire input images, we compare images 1 with images 2 to 100 and 2 with images 3 to 100. This process is done with all images.

5. Experiment Results

The spectral clustering from all the processes is shown in Table 1 and Table 2. However, when using Wasserstein Distance, the presence or absence of a cluster can be clearly discerned by the number of zeroes or non-zeroes, as shown in Table 2, but when using Bhattacharyya Distance, there is a possibility of different analysis depending on the threshold at which the numbers jump, as shown in Table 1. This suggests that Wasserstein Distance is a more appropriate distance tool for studies estimating chases. We can see that there are three zeroes or jumps in Ne-V4, six in Se-V6, and four

in Wo-V1. From these results, we can estimate that the number of chases used to create each book is three, six, and four, respectively. To increase the reliability of the results, we first conducted an impressionistic critique of Ne-V4, which directly analyzed the font style or characteristic features of the characters the same way as section 2.6, and the results were the same as those obtained from the network, suggesting that three chases were used.

Similarity Clustering / Bhattacharyya Distance			
	Ne-V4	Se-V6	Wo-V1
1	3.88×10^{-26}	-4.31×10^{-17}	-3.88×10^{-26}
2	0	2.79×10^{-17}	-3.41×10^{-17}
3	-1.62×10^{-17}	5.24×10^{-25}	0.2321
4	0.0723	6.81×10^{-18}	0.3402
5	0.0986	3.57×10^{-13}	0.6772
6	0.1174	8.61×10^{-25}	0.7569
7	0.2437	0.9780	0.8467
8	0.3661	0.9964	0.8587
9	0.4121	1.0134	0.8904
10	0.4291	1.0166	0.9023
Cluster	3	6	3 ~ 4

Table 1: Results from spectral clustering using Bhattacharyya distance.

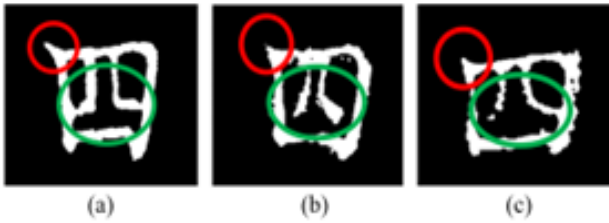


Figure 15: A clustered character from Ne-V4 (a) cluster 1, (b) cluster 2, (c) cluster 3. The upper left stroke is shown in the red box and the center stroke is shown in the green box.

Figures 16 through 18 show some examples of results categorized into clusters. If we look at the details of the clustering results, we can see that although the clusters were classified based on letter spacing similarity, the letter features are also similar. If we compare the last character (四) of the Ne-V4 Pan-Sim classified into different clusters in Figure 15, we can see the difference between the upper left stroke and the center stroke, and you can see that the Pan-Sims in each cluster have these features in common. As with the analysis using character spacing similarity, we can see that the results are

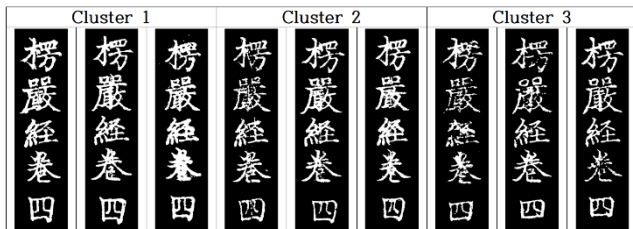


Figure 16: Example of results for clustered "Neungumgyeoung" Volume 4 after spectral cluster.

Similarity Clustering / Wasserstein Distance			
	Ne-V4	Se-V6	Wo-V1
1	-7.17×10^{-18}	4.72×10^{-17}	-2.89×10^{-17}
2	0.0685	1.03×10^{-25}	2.58×10^{-26}
3	0.0924	3.88×10^{-26}	0
4	0.2565	2.58×10^{-26}	0
5	0.2791	0	0.6964
6	0.3342	-1.29×10^{-26}	0.8151
7	0.4163	1.0046	1.0025
8	0.4628	1.0047	1.0025
9	0.5464	1.0050	1.0026
10	0.5945	1.0051	1.0026
Cluster	3	6	4

Table 2: Results from spectral clustering using Wasserstein.

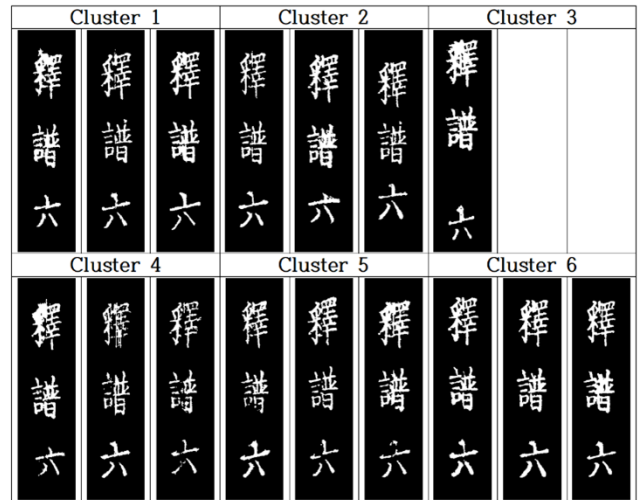


Figure 17: Example of results for clustered "Seokbosangjeol" Volume 6 after spectral cluster.

clustered into groups with similar character spacing. For example in Figure 17, the separation of the character in cluster 3 Bo(譜) and Six(六) is really different compared to other clusters. These outputs prove that the proposed algorithm produces reliable results.

6. Conclusion

In this study, we proposed a method for estimating the number of chases by analyzing metal-type books printed during the Joseon Dynasty. In order to increase the reliability of the results, we conducted a direct impression critique and compared it with the results obtained through the network, and the similarity showed the reliability of the method proposed in this paper. We expect that the proposed method can also be applied to other metal-type books. In future research, we plan to estimate the number of chases by ex-

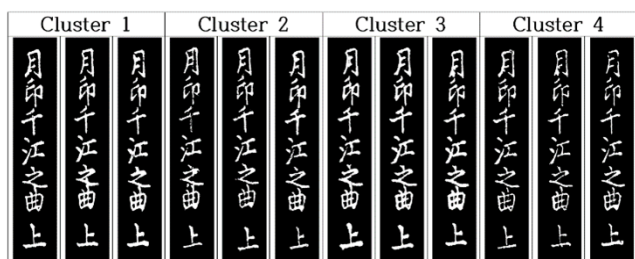


Figure 18: Example of results for clustered "Worin cheongangjigok" Volume 1 after spectral cluster.

panding the scope of Pan-Sim to include the lettering in the Pan-Sim area and the Jikji, as shown in the Jikji comparison paper (Figure 4).

Acknowledgements

This research was supported by Basic Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2022R1F1A1076305) and the BK-21 FOUR program through National Research Foundation of Korea (NRF) under Ministry of Education.

References

[ACB17] ARJOVSKY M., CHINTALA S., BOTTOU L.: Wasserstein gan, 2017. doi:10.48550/ARXIV.1701.07875. 2

[Den20] DENG D.: Dbscan clustering algorithm based on density. In 2020 7th International Forum on Electrical Engineering and Automation (IFEAA) (2020), pp. 949–953. doi:10.1109/IFEAA51475.2020.00199. 3

[EC23] ECKSTEIN S., CHERIDITO P.: Optimal transport and wasserstein distances for causal models, 2023. doi:10.48550/ARXIV.2303.14085. 2

[HSZ87] HARALICK R. M., STERNBERG S. R., ZHUANG X.: Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 4 (1987), 532–550. doi:10.1109/TPAMI.1987.4767941. 5

[JC16] JOLLIFFE I. T., CADIMA J.: Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374, 2065 (apr 2016), 20150202. doi:10.1098/rsta.2015.0202. 3

[Kai67] KAILATH T.: The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communications* 15, 1 (feb 1967), 52–60. doi:10.1109/tcom.1967.1089532. 2

[Kim09] KIM S.-S.: A research on the typesetting of movable types cast later in the shape of gabin-ja. *Journal of Studies in Bibliography (JIB)* 43 (2009), 151–176. doi:10.17258/jib.2009..43.151. 1

[LCVM09] LAPARRA V., CAMPS-VALLS G., MALO J.: Pca gaussianization for image processing. In 16th IEEE International Conference on Image Processing (ICIP) (2009), pp. 3985–3988. doi:10.1109/ICIP.2009.5413808. 3

[Lee07] LEE S. C.: An analysis of movable metal types and type-setting in jikji. *Journal of Studies in Bibliography* 38 (2007), 377–411. doi:10.17258/jib.2007..38.377. 3

[LY07] LEE J.-J., YU H.-S.: An investigation of hangeul metal printing types in the collection of national museum of korea. *Journal of Studies in Bibliography (JIB)* 37 (2007), 123–164. doi:10.17258/jib.2007..37.123. 1

[NJW01] NG A., JORDAN M., WEISS Y.: On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* (2001), Dietterich T., Becker S., Ghahramani Z., (Eds.), vol. 14, MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf. 3

[NXY10] NA S., XUMIN L., YONG G.: Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics* (2010), pp. 63–67. doi:10.1109/IITSI.2010.74. 3

[PSJ15] PATEL S., SIHMAR S., JATAIN A.: A study of hierarchical clustering algorithms. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (2015), pp. 537–541. 3

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision* 40, 2 (2000), 99–121. doi:10.1023/a:1026543900054. 2

[Sh14] SHLENS J.: Notes on kullback-leibler divergence and likelihood, 2014. doi:10.48550/ARXIV.1404.2000. 2

[SSA*14] SEHGAL S., SINGH H., AGARWAL M., BHASKER V., SHANTANU: Data analysis using principal component analysis. In *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)* (2014), pp. 45–48. doi:10.1109/MedCom.2014.7005973. 3

[TCXH22] TANG J., CHENG J., XIANG D., HU C.: Large-difference-scale target detection using a revised bhattacharyya distance in sar images. *IEEE Geoscience and Remote Sensing Letters* 19 (2022), 1–5. doi:10.1109/LGRS.2022.3161931. 2

[WNWL19] WEI Y., NIU C., WANG H., LIU D.: The hyperspectral image clustering based on spatial information and spectral clustering. In *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)* (2019), pp. 127–131. doi:10.1109/SIPROCESS.2019.8868487. 3

[Yoo22] YOO W. S.: Identification of metal type prints, recarved woodblock prints and woodblock recarving sequences through image analyses: Comparisons among six versions of jeungdoga scripts. *Journal of Conservation Science* 38, 5 (oct 2022), 404–414. doi:10.12654/jcs.2022.38.5.05. 1