

Doctoral Dissertation

Studies on Representation Learning for Retrieval
of Digital Cultural Heritage Archives

March 2022

Doctoral Program in Advanced Information Science and

Engineering

Graduate School of Information Science and Engineering

Ritsumeikan University

LI Kangying

Doctoral Dissertation Reviewed
by Ritsumeikan University

Studies on Representation Learning for Retrieval
of Digital Cultural Heritage Archives

(デジタルアーカイブ検索のための表現学
習に関する研究)

March 2022

2022年3月

Doctoral Program in Advanced Information Science and
Engineering

Graduate School of Information Science and Engineering

Ritsumeikan University

立命館大学大学院情報理工学研究科

情報理工学専攻博士課程後期課程

LI Kangying

リ コウエイ

Supervisor: Professor MAEDA Akira

研究指導教員：前田 亮 教授

Abstract

Databases of cultural resources need reliability and rely on efficient search tools. Unlike the currently released datasets for machine learning tasks, there is not sufficient data of digital cultural heritage records to train a deep learning-based model for retrieval. Research on image search and ancient character recognition by one-shot learning is attracting attention for the efficient use of digital cultural heritage records made from some rare collections such as ukiyo-e, kao (stylized signature), and collector's seal. Considering the utilization of limited digital cultural heritage resources, this dissertation aims to improve the retrieval accuracy of seal images, ancient characters and ukiyo-e records by representation learning approaches. The main contributions of this dissertation are as follows:

1) A single character segmentation method for seal images is proposed. To find better features suitable for seal image retrieval, various image features including deep features extracted from the existing pre-trained model are used for comparative experiments.

2) In order to support the reading comprehension of characters that are no longer used in modern times, such as the seal script used for seals, a meta-learning-based character recognition method is proposed to learn to represent an ancient character in suitable representation for retrieval tasks.

3) For supporting the retrieval of ukiyo-e records, a cross-modal representation learning framework is proposed by considering the utilization of textual metadata. As an application case of the proposed method, a system that can search ukiyo-e using color information expressed in word descriptions is constructed.

The performance of each proposed method was shown by conducting evaluation experiments using real-world datasets and benchmark datasets.

博士論文要旨

文化資源としての古典資料を対象としたデータベースにおいては信頼性が求められ、効率的な検索ツールの提供が期待される。現在公開されている機械学習タスク用のデータセットと異なり、文化資源においては、検索のための深層学習モデルの学習に十分な量のデジタル化されたデータが存在しない場合が多い。このため、希少な浮世絵、花押、蔵書印などを対象としたデジタルコレクションの効率的な活用に向けて、ワンショット学習による画像検索や古代文字認識の研究が注目されている。限られた量のデジタル文化資源の活用を考慮し、本博士論文では、印鑑画像、古代文字、浮世絵レコードに対して表現学習のアプローチによる検索精度の向上を目指す。本論文の主な成果は以下の通りである。

1) 印鑑画像の検索において、単一文字領域分割手法の提案を行う。印鑑画像検索に適切な特徴を見つけるため、事前学習済モデルを用いて抽出された深層特徴を含む、多種類の画像特徴を用いた評価実験を行った。

2) 印鑑に使われる篆文など、現代ではすでに使われていない文字の読解を支援することを目的として、検索タスクに適切な古代文字の表現を学習するための、メタ学習に基づいた文字認識手法を提案する。

3) 浮世絵レコード検索支援のための、メタデータテキストの活用を考慮したクロスモーダル表現学習の枠組みを提案する。提案手法の応用例として、言葉で表現する色情報で浮世絵を検索できるシステムの構築を行う。

上記の各提案手法の有効性を評価するために、実環境のデータセットとベンチマークデータセットを用いてそれぞれの評価実験を行った結果、各提案手法の有効性が示された。

Acknowledgement

I would like to convey my gratitude to everyone who has helped me during my student life since I have got involved in the PhD program. This dissertation would not be possible without the support and guidance from many incredible mentors, lab mates, friends, and family.

First of all, I would like to extend the deepest gratitude to my supervisor Professor Akira Maeda for his continuous guidance, valuable advice, and encouragement during my studies at Ritsumeikan University. He played a significant role in helping me fund the several projects and all equipment those projects needed, which was a decisive pillar throughout the research process. He kindly supported the great number of assistants in terms of getting all the data my research needed. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I am more than grateful to my doctoral dissertation committee members, Professor Hiromitsu Shimakawa and Professor Yoko Nishihara. I am honored that they gave me very detailed and valuable comments on my dissertation, whose suggestions provided new ideas on the way of my research in the future.

I would also like to express my sincere thanks to Professor Ryo Akama and senior researcher Dr. Biligasaikhan Batjargal for providing the in-depth discussion and helping me with a large number of valuable and inspirational new ideas and comments.

I tend to thank Associate Professor Lin Meng from Ritsumeikan University and Associate Professor Hui Li from Nanjing Agricultural University, China, who gave me a lot of advice on my research and helped me when I was struggling with my studies and life. I also want to show my great appreciation to the supervisors of my bachelor course, Associate Professor Xiafen Zhang from Shanghai Maritime University, China, a very

kind instructor who first introduced me to the digital humanities field.

Thanks to the Digital Library laboratory members for creating a lively and stimulating working atmosphere and supporting each other. I would especially thank Dr. Yuting Song for helping me in my daily life and study from the master program to the doctoral program.

I would like to thank JSPS for financial assistance for this research. This work is supported in part by Japan Society for Promotion of Science (JSPS) under Grant JP 21J15425. I am grateful to Momofuku Ando International Student Scholarship form 2019.4-2020.3.

Next, I would like to thank my family and friends who have brought happiness into my life.

I would like to thank Jiayun Wang, Ruishan Zhang and Xiaonan Lyu, the best friends I have met since I came to Japan. They have offered me great help and encouragement when I was fighting against depression. They have shared with me their views on all kinds of interesting events in the world as well as all kinds of delicious food, cool! Thanks to my good friend Qingqing Dong. Although we didn't study in the same country since high school, we could still talk about everything, share our dreams, and I hope that she can successfully complete her doctoral program at the University of Houston and be happy every day. Thanks to my best friends Yeying Zhang, Yi Zhang, Rongyi Tang and Vincent Hu. They have brought the happiness in my life and the motivation for my efforts, and they have provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

I would express my deepest gratitude to my parents Yuyan Zhang and Xiaolong Li. I wish to thank them all for their support during my studies and life in Japan. Thanks to

my family members, especially my aunt Yuhui Zhang and my niece Ruojun Li for listening to me about my troubles and giving me help. Thanks to my cat Tim for accompanying me and making my day not lonely.

Finally, in memory of grandfather Pingchun Zhang and grandmother Linying Yi, they are the forever soul force that I have embarked on the road of scientific research, and they are also the great courage and motivation for me to carry on in my life.

Kangying Li

Ritsumeikan University

February 2022

Contents

Abstract	I
博士論文要旨	II
Acknowledgement	III
Contents	VI
List of Figures	X
List of Tables	XIII
Chapter 1 Introduction	1
1.1 Background and Objectives	1
1.2 Contributions.....	5
1.3 Outline.....	6
Chapter 2 Data Collection, Processing, Analysis and System Implementation for Digital Cultural Heritage Archives	8
2.1 Data Collection.....	8
2.1.1 Collecting data from public dataset.....	8
2.1.2 Crawling and using APIs.....	11
2.1.3 Crowdsourcing and archiving service	12
2.1.4 Self-Archiving & Copyright arrangements	12
2.2 Data Processing.....	13
2.2.1 OCR and data cleaning.....	13
2.2.2 Labeling.....	15
2.2.3 Checking data quality.....	16
2.3 Data Analysis	17
2.4 System Implementation.....	18

Chapter 3 Fundamentals of Representation Learning and Few-shot Learning	19
3.1 Representation Learning	19
3.1.1 Representation learning in computer vision.....	19
3.1.2 Representation learning in natural language processing.....	20
3.1.3 Cross-modal representation learning.....	23
3.2 Few-shot Learning	24
3.3 Representation Learning and Few-shot Learning for Retrieval.....	25
Chapter 4 Seals Imprint Retrieval and Owner’s Relationship Extraction.....	26
4.1 Introduction.....	26
4.2 Contribution	28
4.3 Related Work.....	28
4.3.1 Seal retrieval.....	28
4.3.2 Character segmentation for OCR.....	30
4.4 Methods.....	31
4.4.1 Ukiyo-e artist’s seal retrieval and artists’ relationship extraction	31
4.4.2 Character segmentation for collector’s seal retrieval.....	39
4.5 Experiments and Results	48
4.5.1 Evaluation experiments on ukiyo-e artist’s seals retrieval.....	48
4.5.2 Character segmentation results.....	49
4.5.3 Character retrieval results.....	53
4.6 Summary	61
Chapter 5 Ancient Character Recognition.....	64
5.1 Introduction.....	64
5.2 Related Work and State-of-the-Art.....	65

5.2.1 Ancient character recognition.....	66
5.2.2 Sketch-based image retrieval based on shape-matching.....	66
5.2.3 Meta-learning and metric-based method.....	68
5.3 Main Contributions of This Study.....	70
5.4 Methods.....	71
5.4.1 Data-augmentation-based method.....	71
5.4.2 Metric-learning-based method.....	74
5.5 Experiments and Results.....	81
5.5.1 Evaluation on data augmentation-based method.....	81
5.5.2 Evaluation on data metric learning-based method.....	82
5.6 Summary.....	95
Chapter 6 Cross-Modal Retrieval for Japanese Ukiyo-e Prints.....	96
6.1 Introduction.....	96
6.2 Contributions.....	101
6.3 Related Work.....	102
6.4 Methodology.....	103
6.5 Demo Implementation.....	111
6.6 Experiments.....	112
6.7 Summary.....	116
Chapter 7 Conclusions and Future Work.....	117
Bibliography.....	119
Publication List.....	130
Journal Papers.....	130
International Conferences.....	130

Domestic Conferences	131
Other publications	131
Appendix A	133
Appendix B	134
Appendix C	136

List of Figures

Figure 1.1 ‘Digital humanities’ in Google Ngram Viewer (Screenshot of the tool from Google Ngram site, https://books.google.com/ngrams).....	1
Figure 1.2 Retrieval interface of ARC Japanese Woodblock Prints Database	3
Figure 1.3 Detailed page of ARC Japanese Woodblock Prints Database	3
Figure 1.4 User interface of Shirakawa-font retrieval system	4
Figure 1.5 The data of 'ukiyo-e' and 'ancient characters' from Google Ngram Viewer (Screenshot of the tool from Google Ngram site, https://books.google.com/ngrams).....	5
Figure 2.1 Types of humanities data	9
Figure 2.2 An example of processing the scanned old newspaper information extraction	14
Figure 2.3 An example of OCR post-correction	15
Figure 2.4 The main issues that need to be checked and verified on the completed dataset.....	16
Figure 2.5 System implementation, deployment and release environment.....	18
Figure 3.1 Image and engineering knowledge	20
Figure 3.2 Convolutional neural network and autoencoder	20
Figure 3.3 Joint representation and coordinated representation in cross-modal representation learning	23
Figure 3.4 A case of few-shot learning on character recognition.....	24
Figure 3.5 Three types of data mainly used in representation learning	25
Figure 4.1 Examples of two types of seals.....	26
Figure 4.2 Complete system framework and application of search results.....	32
Figure 4.3 Matching process of the proposed method	32
Figure 4.4 Seal retrieval on whole seal image	33
Figure 4.5 An example of a seal overlaps with handwritten words	33
Figure 4.6 Three-dimensional representation of color information	34
Figure 4.7 Tree structure and random sampling for quick search and normal search.....	36
Figure 4.8 Obtain the ranking result base on similarity calculation.....	36
Figure 4.9 Extending the candidate results	37
Figure 4.10 Examples of the candidate results.....	37
Figure 4.11 Artists relationship extraction	38
Figure 4.12 Visualization of clustering results under different bandwidths.....	41
Figure 4.13 Obtaining the segmentation results.....	42

Figure 4.14 Clustering results under adjacent bandwidth	44
Figure 4.15 Images converted from a font file.....	44
Figure 4.16 Extraction of CNN features	45
Figure 4.17 Visual expression of feature map in max pooling layer.....	46
Figure 4.18 Skeleton map of a character.....	46
Figure 4.19 Matching process	47
Figure 4.20 A part of experimental data.....	49
Figure 4.21 Experiment on feature and distance calculation method selection	54
Figure 4.22 Calculating the distinction score of each character	57
Figure 4.23 Evaluation on seal imprints character recognition.....	60
Figure 4.24 An incorrect result caused by pre-processing of query image.....	60
Figure 4.25 Some difference in the current version of the reference book.....	61
Figure 5.1 Similar characters in Odia number, a, b, and c are three different characters.....	68
Figure 5.2 GAN-based zi2zi model	72
Figure 5.3 An example of training data pair.....	73
Figure 5.4 Training data with morphological transform	73
Figure 5.5 Generated training samples.....	73
Figure 5.6 Our proposed character-recognition framework.....	74
Figure 5.7 Typeface image pre-processing.....	75
Figure 5.8 An overview of our proposed model	75
Figure 5.9 Architecture design of the function $f_{loc}()$ in our method	77
Figure 5.10 Ten completely different characters randomly selected from OMNIGLOT.....	77
Figure 5.11 An example of HOG feature map	77
Figure 5.12 An example of prototypical networks classification in a few-shot case	78
Figure 5.13 A voting classifier	81
Figure 5.14 Skeletonization process	87
Figure 5.15 Demo application implementation.....	92
Figure 5.16 Utilization of the proposed method for recognition of oracle bone characters from historical documents.....	92
Figure 5.17 The proposed method applied to approximate query matching for retrieval-based character recognition.....	95
Figure 6.1 $(H,L)=(0,40)$ color group	96
Figure 6.2 Beer color and tea color.....	97
Figure 6.3 Results for ‘yellowish blue’ using text-image search and image-image search	98
Figure 6.4 Obscure colors in ukiyo-e.....	99

Figure 6.5 The same work from the databases of different institutions	100
Figure 6.6 Multitask fine-tuning using a pre-trained CLIP model that learned a large amount of cross-modal knowledge from a large number of real life images.....	101
Figure 6.7 Architecture of our cross-modal multitask fine-tuning representation learning framework	103
Figure 6.8 Structure of space sampler module	104
Figure 6.9 Example sketch image output from AODA Net	105
Figure 6.10 An example of HOG feature map	105
Figure 6.11 IDF calculation.....	108
Figure 6.12 Colors with the top 6 highest frequencies.....	108
Figure 6.13 Example of color names and their Scorecolor name	109
Figure 6.14 Example of training data of cross-modal fine-tuning task.....	110
Figure 6.15 Target user groups and requirements	112
Figure 6.16 Demo application implementation.....	112

List of Tables

Table 2.1 A part of European digital humanities projects	10
Table 2.2 Examples of digital humanities resources opened for machine learning tasks	11
Table 2.3 Examples of digital humanities resources APIs	11
Table 2.4 Some problems that may be encountered when scanning documents with general scanners	13
Table 2.5 Some existing open platform examples for image labeling	16
Table 2.6 Tools for data analysis	17
Table 2.7 Examples of data analysis in the Seal Script Dataset	17
Table 3.1 Examples of representation learning for natural language processing	22
Table 4.1 The main tasks and utilizations	28
Table 4.2 Extracting a seal area from an ukiyo-e print	35
Table 4.3 XML document design for tree search-based matching process	35
Table 4.4 The shapes of collector’s seals	39
Table 4.5 Seals outside the scope of this research	40
Table 4.6 Seals of the 10 ukiyo-e artists	49
Table 4.7 Retrieval accuracy	49
Table 4.8 Segmentation results in square seal images	51
Table 4.9 Segmentation results for seal imprint with different shapes	52
Table 4.10 Performance of our proposed method	53
Table 4.11 Regular map and skeleton map used in matching experiments	55
Table 4.12 Experiment on feature and distance calculation method selection	55
Table 4.13 Evaluation of features extracted from pre-trained model by ImageNet	56
Table 4.14 Evaluation of features extracted from model trained on handwritten Chinese characters dataset	56
Table 4.15 <i>Scorefeature_i</i> in different situations. (a) and (b) show the distinction score of regular image and skeleton map in ImageNet pre-trained model. (c) and (d) show the distinction score in pre-trained model by CASIA Online and Offline Chinese Handwriting Databases	58
Table 4.16 <i>Scorefeature_i</i> in different situations	58
Table 4.17 Features used in seal imprint retrieval	59
Table 4.18 Part of the data used in retrieval experiments	59
Table 4.19 Examples of poor results	61
Table 5.1 Related work on ancient character recognition	67
Table 5.2 Experimental results on data augmentation-based method	82

Table 5.3 The loss value in training domain and target domain at 1000 epoch. The orange line represents the training domain and the blue line represents the target domain, the vertical axis represents the loss value, and the horizontal axis represents the number of epochs.	84
Table 5.4 Comparison on five-way (one-shot) classification task	84
Table 5.5 Classification ability comparisons on each classifier under different settings of sample numbers in the support set and different feature dimensions	85
Table 5.6 The performance of the retrieval task that uses the features extracted from the model trained on forty-way (5-shot) task and 1000 epochs	86
Table 5.7 Ablation experiments of our proposed model	87
Table 5.8 Performance evaluation of pre-training features on retrieval tasks.....	88
Table 5.9 Application performance on MERO_HIE hieroglyphics font and Aboriginebats font datasets	90
Table 5.10 Results for OMNIGLOT→EMNIST five-shot classification	90
Table 5.11 Performance comparative evaluation of pre-training features on retrieval tasks.....	91
Table 5.12 Comparison between this research and other exiting methods in different tasks.....	94
Table 6.1 Description of the color name for the same color in different databases	97
Table 6.2 Similarity between image pairs in IMGonline and DeepAI.....	100
Table 6.3 Example of color information extraction	107
Table 6.4 Example of training data of structural-feature-based triplet fine-tuning task	111
Table 6.5 Vocabulary differences between CN and HCN	113
Table 6.6 Example of color name for the same color in different databases	113
Table 6.7 Example of fine-tuning task progresses	114
Table 6.8 Cross-modal performance	115
Table 6.9 Examples of search results	115

Chapter 1 Introduction

1.1 Background and Objectives

‘Digital Humanities is born of the encounter between traditional humanities and computational methods’ was said by Burdick et. al in MIT Press [1]. Digital humanities integrates important insights from language and literature, history, music, media and communication, computer science and information research, and combines these different methods into a new framework. Recently, with the rapid development of engineering related fields, such as machine learning, data science and artificial intelligence technologies, digital humanities will occupy a more and more important position in humanities research. As a part of digital humanities research, some new methods such as computer-based statistical analysis [2], search and retrieval [3], 3D modeling [4] and data visualization [5] attracted attention from all walks of life.

The Google Ngram Viewer is an online search engine that charts the frequencies of any set of queries using a yearly count of n-grams found in resources printed between 1500 and 2019 from text corpora in several languages. The search result of ‘digital humanities’ is shown in Figure 1.1.

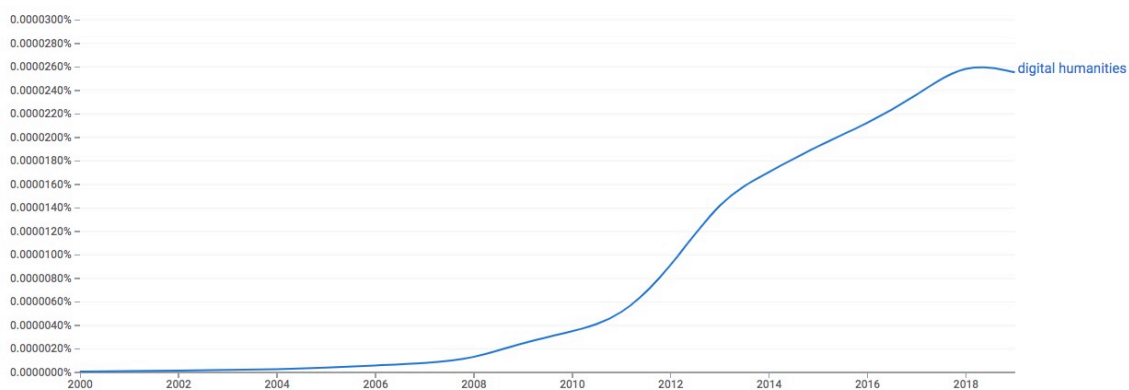


Figure 1.1 ‘Digital humanities’ in Google Ngram Viewer (Screenshot of the tool from Google Ngram site, <https://books.google.com/ngrams>)

It can be seen that since 2006, attention in this field has gradually increased. Digital humanities has shown obvious advantages in the fields of linguistics and literature. At the same time, its research results in the field of history are increasing rapidly, while it only accounts for a small proportion in other humanities fields. In order to meet the needs of the development of the times, more and more museums, art galleries, expert institutions, and academic institutions have begun to pay attention to the construction of digital humanities and the promotion of digital humanities research. Especially in Japan, many humanities researchers said that with the increase in the utilization of digital images, and metadata of historical records, the effectiveness of search results has brought more convenience to humanities research work. The National Diet Library of Japan [6] has been trying to process more than 300,000 publications that have been digitalized since decades ago, and recently they have announced that most of these publications will be made available to the public. The Toyama Library of Waseda University released Edo-period and Japanese Literature Collection (from Japanese and Chinese Classics) [7] online. This project work started in 2005 with the aim of creating data (bibliography/images) of about 300,000 classic books held by the Waseda University Library, and continues to this day. Kyoto University set up a working group called Kyoto University Digitization Hub of the Humanities, Social and Cognitive Sciences (KUDH) from 2021 [8], scholars active in the field of digital humanities will be invited to share their research results and academic views. The cultural resources handled by this group are wide-ranging, including the Kanji cultural sphere, Western ancient times, ancient Egypt, and ancient Maya.

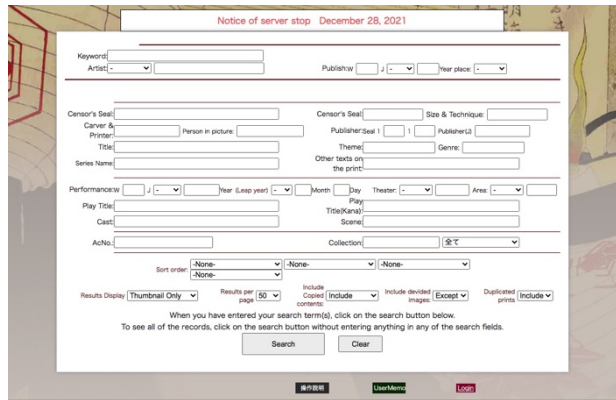


Figure 1.2 Retrieval interface of ARC Japanese Woodblock Prints Database

Ritsumeikan University Art Research Center [9] has also released a lot of digital cultural heritage archive databases including The Early Japanese Books Database, Donated and Entrusted Books Database, The Modern Japanese Book Database, Exhibition Catalogues Database, Movie Brochures Database, Shibai Banzuke (Kabuki Playbills) Browsing System, Japanese Woodblock Prints, Fujii Eikan Bunko Database and so on. As shown in Figure 1.2, Japanese Woodblock Prints database provides a very detailed and convenient retrieval interface, and users can retrieve and view the hidden data from 206,671 items.

A retrieval case of ukiyo-e prints is shown in Figure 1.3. Users can quickly obtain information such as ‘title’, ‘series’ and ‘artist’ from the detailed page corresponding to the entered query, and can easily find similar records to the current ukiyo-e print.

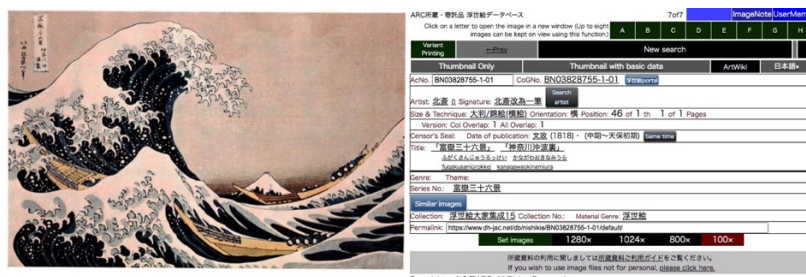


Figure 1.3 Detailed page of ARC Japanese Woodblock Prints Database

The Shirakawa Shizuka Institute of East Asian Characters and Culture, Ritsumeikan University has developed a search system for ‘Shirakawa font’ [10], which can be used freely by converting kanji characters from the modern calligraphic style to the old calligraphic styles (oracle bone script, bronze script, and seal script). About 4,400 ancient characters of ordinary and personal names are able to be searched. The total number of ancient characters recorded is 4,391 characters (oracle bone script 681, bronze script 1,084, seal script 2,593, kobun 3, and chubun 3).

The system is possible to search by two methods, ‘search by text string’ and ‘search by characters’. As shown in Figure 1.4, each ancient character is made as a vector image with smooth edges, and each search result can be enlarged to see the details. These well-archived images have also been made as font packages and released on the website.

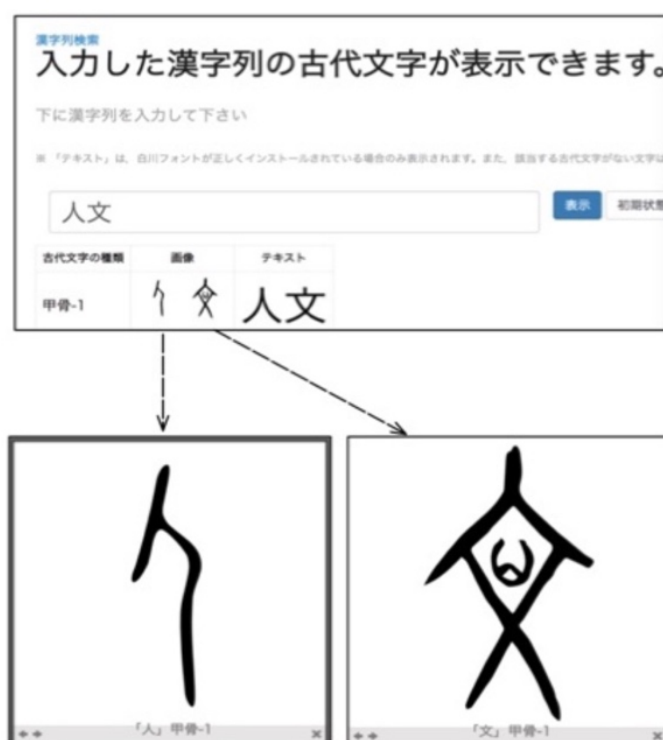


Figure 1.4 User interface of Shirakawa-font retrieval system

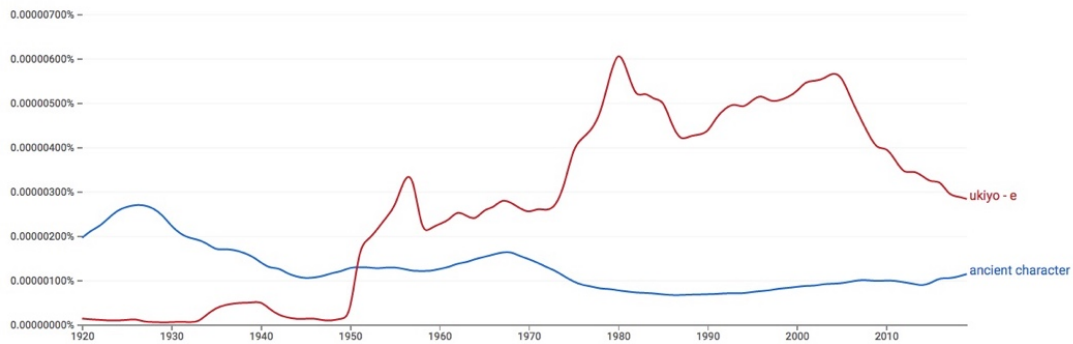


Figure 1.5 The data of 'ukiyo-e' and 'ancient characters' from Google Ngram Viewer (Screenshot of the tool from Google Ngram site, <https://books.google.com/ngrams>)

It is a challenge to use new technical means to increase the attention of these digitized cultural resources. Figure 1.5 shows the data of 'ukiyo-e' and 'ancient character' on Google Ngram Viewer. It can be seen that in different periods, utilization of cultural resources such as ukiyo-e and ancient characters has not increased.

The purpose of this dissertation is to propose methods that focus on representation learning and image processing, using the data of three kinds of cultural resources and applying them to retrieval tasks. From the perspective of flexible utilization of digital humanities data resources of Ritsumeikan University, these three types of data mainly include: ukiyo-e, collector's seals, and ancient characters. The improvement of retrieval accuracy and providing users with a friendly demonstration including data visualization are the goals of this research.

1.2 Contributions

The main contributions of this dissertation can be considered as follows:

(1) From a technical aspect, focusing on the digital humanities resources that are difficult to expand, various methods were proposed to solve the problems of limited training data. The details are introduced in each chapter.

(2) The proposed methods are applied to retrieval and visualization tasks, providing a new perspective for the flexible use of low-resource data.

1.3 Outline

The remainder of the dissertation after this introductory chapter is organized as follows:

In Chapter 2 of this dissertation, cultural heritage archives data collection in digital humanities and data processing cases are introduced. Focusing on the experimental objects in this dissertation, the challenges when collecting data are also described. In Section 2.4, the case of system implementation in Chapters 4, 5, and 6 and the development tools used are also briefly introduced.

In Chapter 3, The basics of representation learning and few-shot learning are briefly summarized. The cases of these fundamental knowledge and their application in information retrieval tasks are also presented in Chapter 3.

Chapters 4, 5, and 6 introduce the studies on collector's seal imprints, ancient characters, and ukiyo-e prints data in the field of digital humanities research, respectively.

In Chapter 4, first, the basic methods of collector's seal imprints image pre-processing are introduced. Second, a method for character segmentation is introduced, and comparative experiments with other character segmentation methods are presented. Finally, the experiment and demo system implementation of the retrieval task of collector's seal imprint images is described. An attempt at isolated character recognition using deep features is presented in this study. In this chapter, the initial experiment results were introduced and became the foundation of the study in Chapter 5.

In Chapter 5, based on the research introduced in Chapter 4, for some ancient characters that are difficult for non-experts to interpret on collector's seal imprints, several

studies related to ancient character recognition and the corresponding experimental results are introduced. The methods mentioned in this chapter mainly focus on one-shot learning and representation learning, which lays a foundation for the research of cross-modal representation learning in Chapter 6.

In Chapter 6, based on the research on the representation learning method of ancient character images with distinctive geometric features in Chapter 5, the research are carried out to the representation learning of artwork images and the corresponding metadata. This chapter focuses on the exploration of cross-modal representation learning methods. From the perspective of image metadata and image color words-based description, the utilizations of pre-trained language model and cross-modal model are introduced.

In Chapter 7, the conclusions and future work of this dissertation are presented.

Chapter 2 Data Collection, Processing, Analysis and System Implementation for Digital Cultural Heritage Archives

2.1 Data Collection

Data collection of digital cultural heritage archives is an important task for the research of digital humanities. There are usually the following four methods of obtaining data:

(1) From public dataset

(2) Using APIs for downloading data from databases or building a crawler to extract data from websites

(3) Asking for crowdsourcing or archiving service

(4) Self-archiving

This section introduces some commonly used methods of obtaining data and some things needed to pay attention to when publishing datasets.

Some tools, processes and products are also presented in this section.

2.1.1 Collecting data from public dataset

As shown in Figure 2.1, there are many types of data that digital humanities researchers frequently engage with.

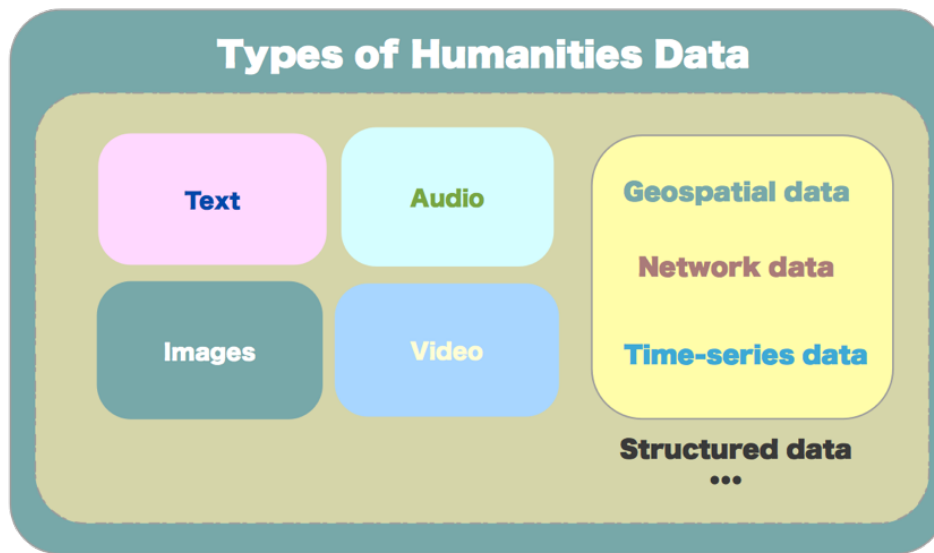


Figure 2.1 Types of humanities data

These data can be flexibly used in many different tasks, such as: text mining, natural language processing, computational linguistics, image generation, game design, etc. We can easily obtain some data that can be used for machine learning tasks from some publicly available digital humanities projects. Table 2.1 shows a part of European digital humanities projects. Data include colonial zapotec texts, the open-access bibliographical corpus of medieval Catalan literature, biblical references found in both Western and Eastern Christian literature, the Digital Resource of documents and their constituent letter-forms for Palaeography and so on.

These public data can be collected after determining the authority to use the data. There are also many digital humanities resources that are specifically open for machine learning tasks, Table 2.2 shows some examples of them.

Digital humanities researchers can flexibly use these published data to create some meaningful machine learning tasks and provide new ways to access historical and cultural resources.

Table 2.1 A part of European digital humanities projects

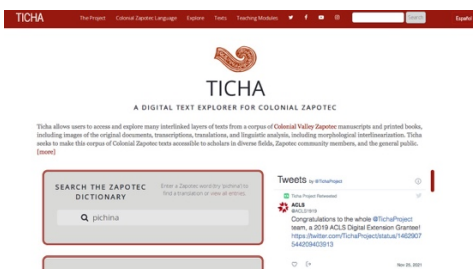




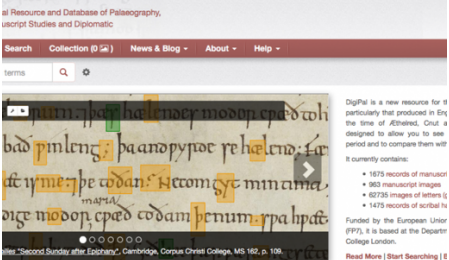
Text	Image
	
<p>Ticha: Online, digital explorer for a corpus of Colonial Zapotec texts. (https://ticha.haverford.edu/en/about/)</p>	<p>Artefacts: Gathers data from publications, museums and private collections to offer images of object's forms. (http://artefacts.mom.fr)</p>
	
<p>Bilicame: Open-access bibliographical corpus of medieval Catalan literature. (http://bilicame.iifv.net/cat/)</p>	<p>Forschung: Providing access to a digital corpus of historical sources containing texts as well as images. (https://www.uni-weimar.de/de/medien/professuren/medienwissenschaft/theorie-medialer-welten/forschung/)</p>
	
<p>Biblindex: An index of biblical references found in both Western and Eastern Christian literature. (http://www.biblindex.org)</p>	<p>The Digital Resource of documents and their constituent letter-forms for Palaeography. (http://www.digipal.eu/)</p>

Table 2.2 Examples of digital humanities resources opened for machine learning tasks

Dataset	Description	Ref.
KMNIST Dataset	Contains 70,000 28x28 grayscale images spanning 10 classes.	https://github.com/rois-codh/kmnist
ARC Ukiyo-e Faces Dataset	A large-scale (>10k paintings, >20k faces) ukiyo-e dataset with coherent semantic labels and geometric annotations.	https://github.com/rois-codh/arc-ukiyo-e-faces/
Seal Script Dataset	A machine learning-friendly dataset of "Tensho" character images to be used for the interpretation of seals.	http://codh.rois.ac.jp/tensho/book/
Large Time Lags Location (LTLL) dataset	Ancient images of historical locations dating back to the period 1850s-1950s have been provided.	http://users.cecs.anu.edu.au/~basura/beeldcanon/

2.1.2 Crawling and using APIs

Some open source websites do not publish packaged machine learning datasets. For such cases, there is a way to collect these data using web crawlers or APIs.

In order to integrate collection records with modern technology, many museums and companies have published APIs that facilitate researchers to obtain collection information. Table 2.3 shows several API interfaces that can obtain data of digital humanities records.

Table 2.3 Examples of digital humanities resources APIs

API name	Institution & Company	Ref.
The Metropolitan Museum of Art Collection API	The Metropolitan Museum	https://metmuseum.github.io/
Google Arts & Culture	Google	https://github.com/googleartsculture
Wikidata	Wikipedia	https://www.wikidata.org
The National Museum of Australia collection API	The National Museum of Australia	https://www.nma.gov.au/about/our-collection/our-apis

After checking the license and data copyright, the collection data of these institutions can be used in projects. If it is some institutional database that does not provide API, after consulting and confirming the right to use the data, web crawlers can be used to collect data.

2.1.3 Crowdsourcing and archiving service


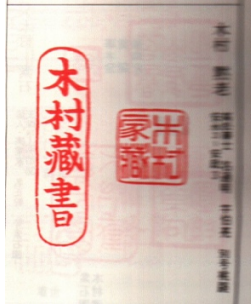

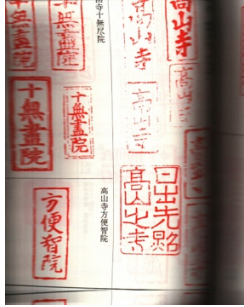

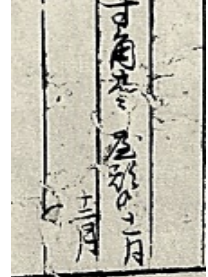
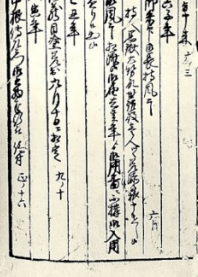

If some historical and cultural resources need to be archived by individuals, scanning and tagging become tasks to face. Establishing metadata and scanning requires expert knowledge, which is a consumption of time and human resources. There are many studies and reports on the current status and case studies of crowdsourcing services in the field of digital humanities [11][12]. Some companies also provide scanning and archiving services, such as [13][14]. These companies often have professional equipment and staff to scan these historical documents in archives.

2.1.4 Self-Archiving & Copyright arrangements

Applying for crowdsourcing or archival services requires heavily funded support, so self-archiving has become a method of making digital humanities resources datasets. When archiving historical documents, the original record's proper preservation and operation need to be paid attention to. For example, some historical books need to be disassembled, scanned, and rebound, and the scanning light may damage some paintings. At the same time, it is necessary to choose a suitable and affordable scanner. When the normal scanner is used to scan books, there usually be some problems. Table 2.4 shows the issues that appeared when scanning documents with two types of scanners. Since the book cannot be damaged during scanning, the two commonly used scanners can complete the entire scanning process in the case of long-distance scanning and partial hand-held scanning.

The scanner shown above in Table 2.4 is a portable office scanner.

Table 2.4 Some problems that may be encountered when scanning documents with general scanners

Scanner	Blurred	Distorted	Exposed
			
			

It is necessary to confirm the copyright and other information of the document. After re-confirmation, the scanned information can be processed in the next step.

2.2 Data Processing

There are corresponding processing methods for different archived documents. This section introduces the basic data processing methods of documents based on images and texts.

2.2.1 OCR and data cleaning

If the scanned document is in the form of images, then extracting the content of the text in the document is the first step in data processing. The scanned document represented by modern characters such as old newspapers, old letters, etc., can be automatically

processed by existing high-precision OCR tools.

As shown in Figure 2.2, these tools can automatically detect the layouts of the document and extract the text and images, such as LayoutParser [15], SimpleHTR [16] and Document layout analysis [17].

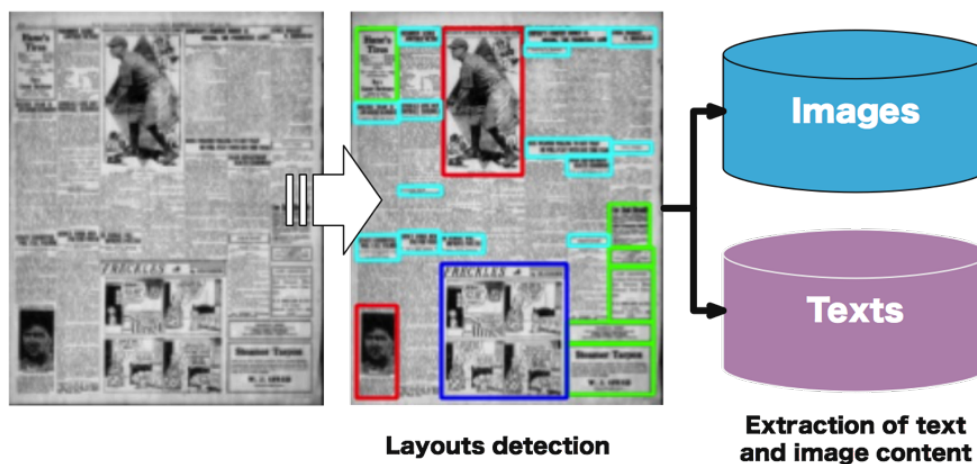


Figure 2.2 An example of processing the scanned old newspaper information extraction

For documents represented by modern characters, tools such as Ochre [18] and TopOCR [19] can be used to automatically correct the outputs of OCR. An example of correction is shown in Figure 2.3. Some basic word processors and spell checkers can also assist in the correction of text data.

For documents mainly based on ancient characters, special text detection methods and character recognition methods need to be considered. Chapter 5 introduces processing and classification methods for unlabeled ancient character documents.

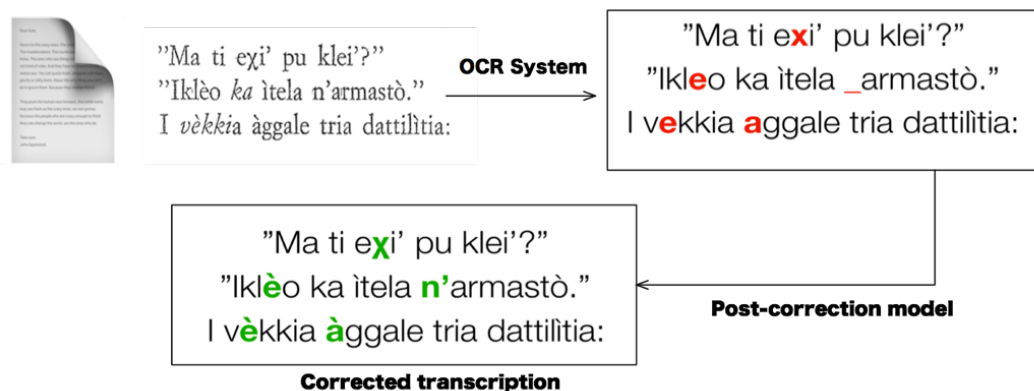


Figure 2.3 An example of OCR post-correction

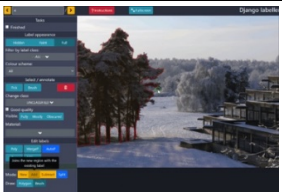
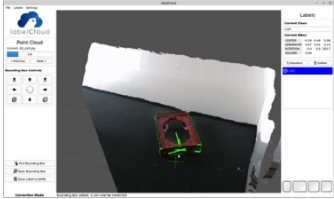
2.2.2 Labeling

After the data has been cleaned, it is necessary to label the data. For different target tasks, corresponding data labeling methods are required. Image data is mainly divided into image data based on characters and image data based on other non-character objects. There are many existing open platforms for image labeling.

Table 2.5 shows some examples of open platform for image or 3D object labeling. LabelFlow is an open platform for image labeling, source is entirely available and developed with a friendly user interface. Django-labeller is a light-weight image labelling tool, it is compatible with Django, Flask and Qt and support polygon, box, point and oriented ellipse annotations. labelCloud is a labeling tool for 3D objects, it also developed light-weight and suitable for extremely large data.

The original data of cultural heritage includes not only 2D scanning data, but also 3D cultural heritage data, such as 3D scanning data of historical relics, etc. Flexible use of these open source projects can reduce the time and money consumption of the project.

Table 2.5 Some existing open platform examples for image labeling

Tool	Interface	Ref.
LabelFlow		https://labelflow.gitbook.io/labelflow/
Django-labeller		https://github.com/Britefury/django-labeller
labelCloud (for 3D object)		https://github.com/ch-sa/labelCloud

There are also many tools for labeling text data for tasks such as natural language processing and named entity recognition such as Small-text [20] and Rubrix [21].

2.2.3 Checking data quality

After the task of labeling the data, in order to improve the reliability of the data, the quality of the data needs to be checked.

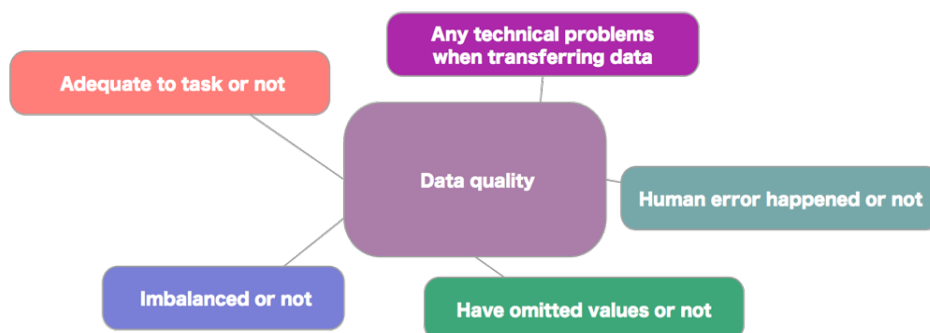


Figure 2.4 The main issues that need to be checked and verified on the completed dataset

2.3 Data Analysis

Data analysis is a necessary process before starting a machine learning task, whether the dataset is a self-made or downloaded.

Table 2.6 shows the data analysis tools we commonly use in our research tasks. Take the Seal Script Dataset mentioned in Table 2.2 as an example. By using these tools can easily check the samples quality and the number of samples in each category.

Table 2.6 Tools for data analysis

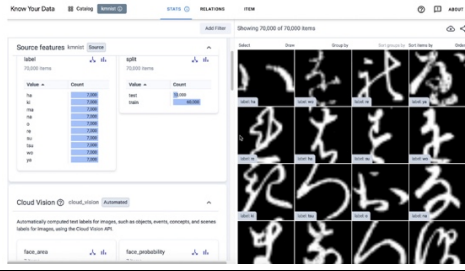
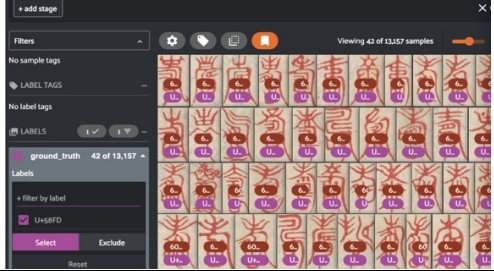
Know Your Data	Voxel51
	
<p>https://knowyourdata.withgoogle.com/</p>	<p>https://voxel51.com/</p>

Table 2.7 Examples of data analysis in the Seal Script Dataset

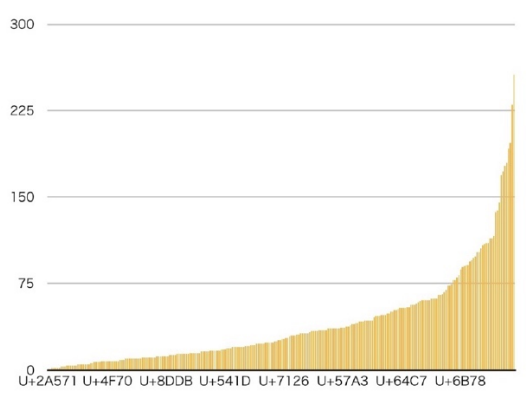
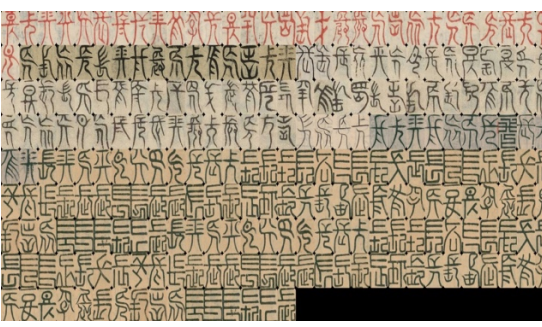
	
<p>Visualization of the distribution of the number of samples corresponding to each category</p>	<p>Preview of all samples under the label labeled '「長」 (U+9577)'</p>

Table 2.7 shows part of the analysis results. It can be known that this is an imbalance

dataset, and it has the characteristics of large differences between samples under the same label. In this research, different task proposals and experiments are carried out according to different analysis results.

2.4 System Implementation

There are many ways to deploy the implemented algorithm to the website. For example, a python-based package scikit-learn can be simply deployed using Flask and Docker container [22]. In this dissertation, the environment mainly used are shown in Figure 2.5 for system implementation, deployment and release.

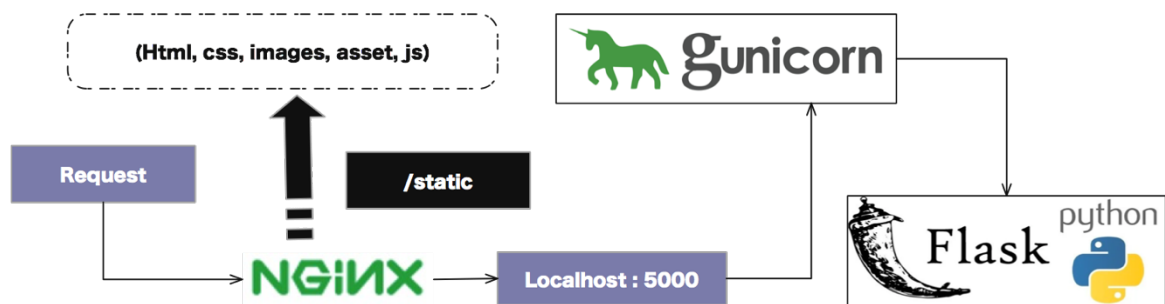


Figure 2.5 System implementation, deployment and release environment

Chapter 3 Fundamentals of Representation Learning and Few-shot Learning

3.1 Representation Learning

Deep learning has recently brought many impressive empirical advantages in a wide range of applications, including image and speech recognition. Good data features can enable better performance on different tasks of deep learning. Representation learning is to make the computer learn better features to use in a variety of different tasks.

The research of representation learning is very critical in the fields of supervised learning and unsupervised learning.

3.1.1 Representation learning in computer vision

Representation learning with images as objects can be simply regarded as transforming images into more appropriate engineering knowledge according to different tasks. As shown in Figure 3.1, images can be represented by engineering knowledge such as some feature vectors. Because the input image object does not have obvious color and texture information, engineering knowledge that focuses on geometric features is more conducive to use. The engineering knowledge obtained by representation learning can be utilized in computer tasks such as data visualization, classification, and clustering.

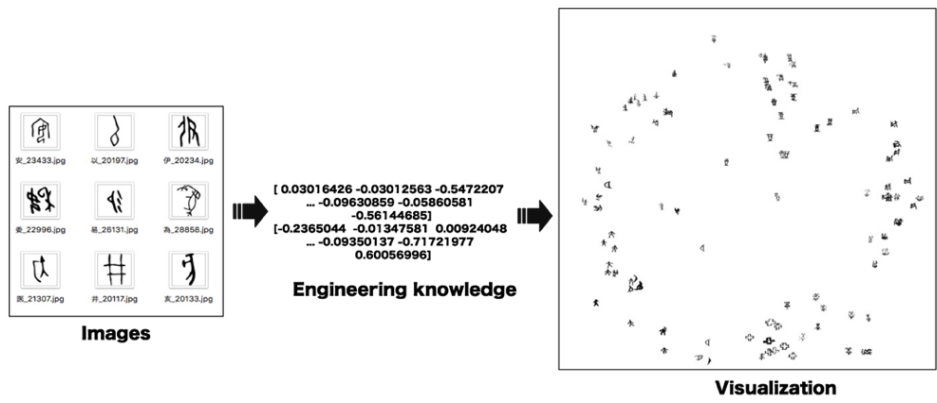


Figure 3.1 Image and engineering knowledge

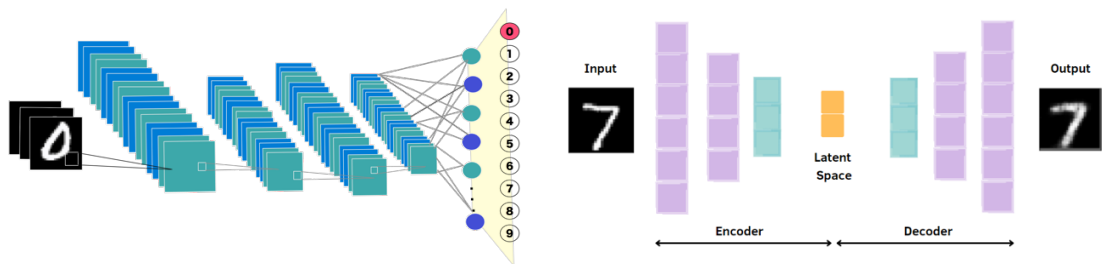


Figure 3.2 Convolutional neural network and autoencoder

In Figure 3.2, the left is an example of a convolutional neural network and the right is an example of an autoencoder. They represent an example of supervised visual representation learning and unsupervised visual representation learning respectively. The representation learned by the former has the corresponding information with the label, or the mapping information of the sample corresponding to the same label in the vector space. The latter is towards the low-dimensional representation of the original input image, which can be used as a method to reduce the dimensionality of the input data.

3.1.2 Representation learning in natural language processing

Different from the computer task of vision, representation learning based on human language is mainly centered on the understanding and processing of symbolic information and logical information. Human language is difficult to process and understood by

computer due to different environments, different backgrounds, different cultural environments, and different information contexts.

Table 3.1 Examples of representation learning for natural language processing

Category	Examples	Models	Utilizations
Word Representation	(1) One-hot Word Representation	(-)	Multilingual Word Representation, Task-Specific Word Representation, Time-Specific Word Representation
	(2) Distributed Word Representation	Latent Semantic Analysis [23], Word2vec [24], GloVe [25]	
	(3) Contextualized Word Representation	ELMo [26]	
Sentence & Document Representation	(1) One-hot Sentence Representation	(-)	Text Classification, Relation Extraction
	(2) Probabilistic Language Model	(-)	
	(3) Neural Language Model	Feedforward Neural Network [27], Convolutional Neural Network [28], Recurrent Neural Network [29]	
	(4) Transformer Language Model	Transformer [30], Transformer-Based PLM	
World Knowledge Representation & Network Representation	(1) Knowledge Graph & Multisource Knowledge Graph Representation	TransE [31], ManifoldE [32], Structured Embeddings [33], RESCAL [34], HOLE [35]	Entity Typing, Information Retrieval, Link Prediction, Community Detection
	(2) Network Representation	Deep walk [36], Structural Deep Network Embedding [37]	
	(3) Graph Neural Network	Spectral Network [38], Neural FPs [39]	

Table 3.1 shows some examples of representation learning for natural language processing. In Chapter 6, this dissertation introduces some word and sentence

representation applications in information retrieval and comparative experimental results.

3.1.3 Cross-modal representation learning

Cross-modal representation learning aims to extract engineering knowledge using information from multiple modalities including texts, images, videos and sounds, etc. As shown in Figure 3.3, the most basic cross-modal representation learning is divided into learning joint representation and coordinated representation from data. The joint representation usually project modalities to their own coordinated spaces. The learned cross-modal representation can be used for image zero-shot recognition, image caption generation, cross-media retrieval and other tasks. Considering the coordinated representation can be used when only one of the modalities is present during test-time. This research selected the coordinated representation based method to implement the proposed training method.

This dissertation introduces frameworks of extracting embeddings from text and image information and applied to the retrieval of ukiyo-e prints. I tried to integrate metadata and ukiyo-e images into unified embeddings, and also tried to build embeddings for different modalities in a common semantic space and apply it to word-based ukiyo-e retrieval based on human sense. The details are introduced in Chapter 6.

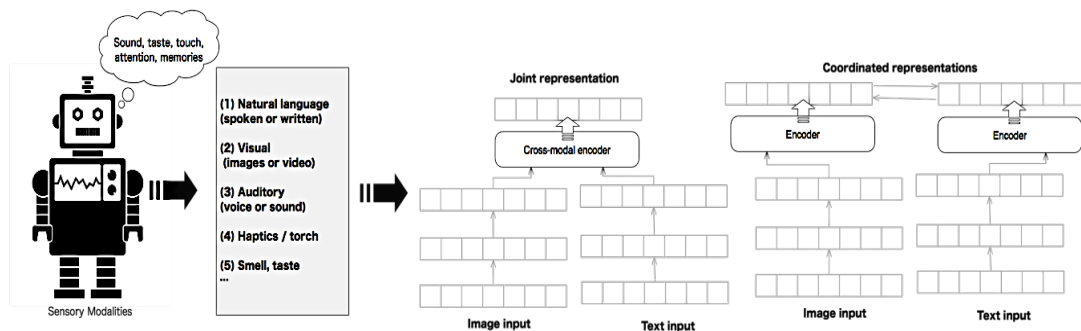


Figure 3.3 Joint representation and coordinated representation in cross-modal representation learning

3.2 Few-shot Learning

Few-shot learning makes predictions based on a limited number of samples from a dataset. It aims not to let the model recognize the images in the training set and then generalize to the test set. The goal of few-shot learning is to ‘learn to learn’. For example, training on different samples from each category and performing the learned distinguishability to the classification task on the unseen dataset.

Meta-learning is a common method to solve few-shot based tasks. Among them, metric-based methods mostly focus on learning the representation from data to be applied on the few-shot based machine learning tasks.

As shown in Figure 3.4, because we don't have enough data, one solution can be considered to learn from sufficient data sources. In this dissertation, a few-shot learning method for character recognition is introduced and applied to the character scarcity task of solving ancient character recognition tasks.

In Chapter 5, the challenges and problems of few-shot learning for character recognition are described. Some attempts to solve these problems and the details of the corresponding experimental results are introduced.

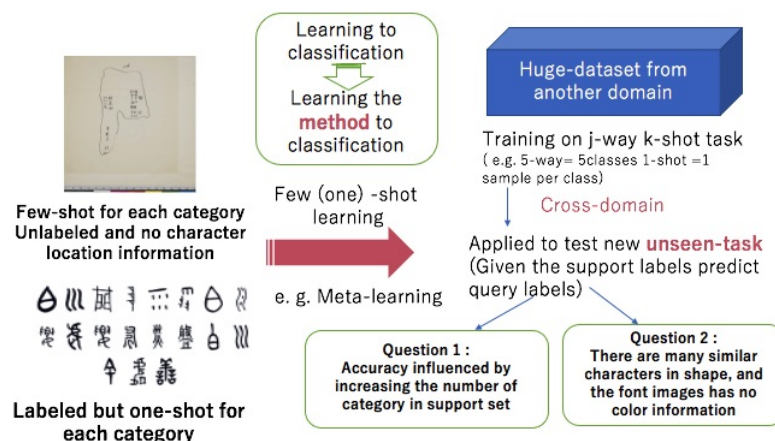


Figure 3.4 A case of few-shot learning on character recognition

3.3 Representation Learning and Few-shot Learning for Retrieval

Due to the amount of digital humanities data that is difficult to expand, this dissertation is mainly oriented to and combines the knowledge of representation learning and few-shot learning, and applies them to the retrieval task of digital humanities data. Figure 3.5 shows the three types of data mainly used in this dissertation, which are ukiyo-e prints, collectors' seals and ancient characters. The introduced contents include data processing, representation learning and proposed few-shot learning methods, and their application in other machine learning tasks.

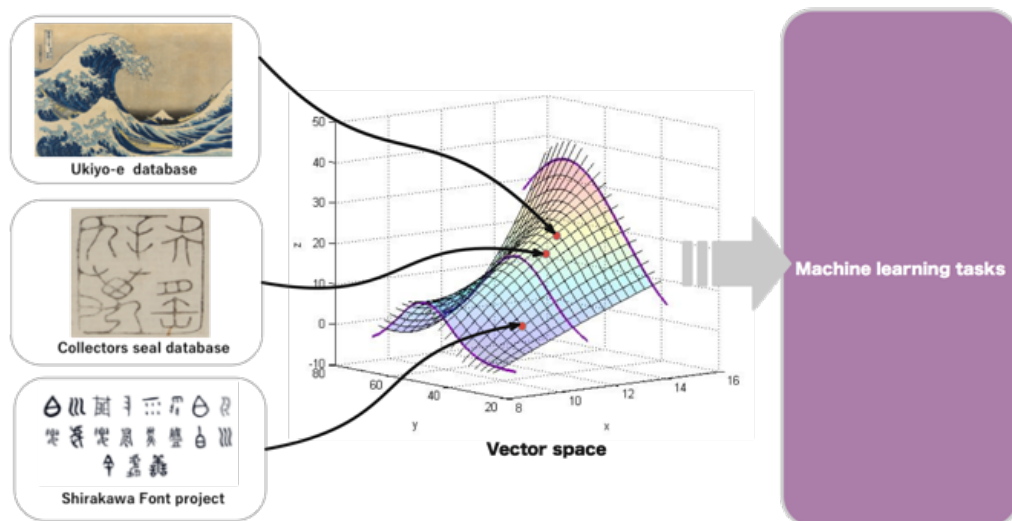


Figure 3.5 Three types of data mainly used in representation learning

Chapter 4 Seals Imprint Retrieval and Owner's Relationship

Extraction

4.1 Introduction

The seal imprints are considered to be very important information for research in the digital humanities field. Various calligraphic and artworks such as ukiyo-e works are stamped with the artist's seal.

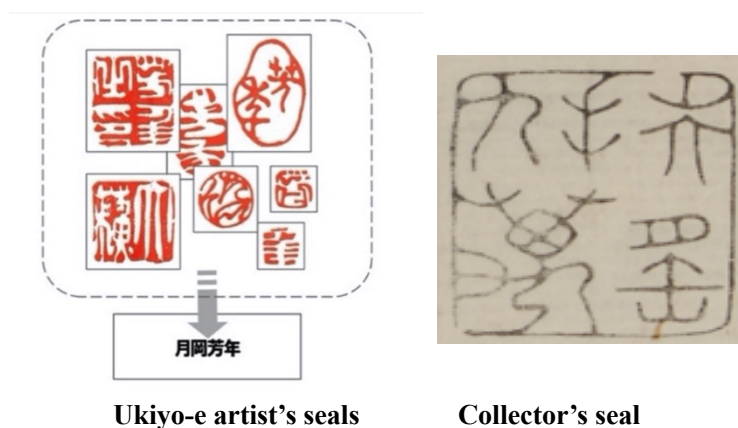


Figure 4.1 Examples of two types of seals

Figure 4.1 show the Examples of two types of seal. As shown on the left side of Figure 4.1, different seals may be used for different works drawn by the same ukiyo-e artist, and may include elements such as designs as well as kanji characters. Expertise is required to identify the characters and design information contained in the seal, and to identify the artist and the work. Determining the production date can be difficult. Likewise, the collector's seal is a stamp for the ownership of the book. An example can also be seen on the right side of Figure 4.1. The collector's seals contain much important information, which is the essential element of historical materials. They show the possession, the relation to the book, the identity of the collectors, the expression of dignity,

etc. Through collector's seals in libraries, can find out the collective experience and the source of inheritance of a book. In Asian countries where kanji characters are used, ancient characters usually make collector's seals. Also, as time passes, the shape of kanji may be changed, and multiple variations of a character might be created. There are also characters derived from kanji characters or those that look like kanji characters. A system that automatically recognizes these characters and retrieves the seal images can help enthusiasts and professionals better understand the background information of these seals more efficiently. For the application of search results, the extraction of the personal relationship network of the seal owners helps researchers to obtain more relevant information of entered query.

When implementing the retrieval system, consider the following issues:

(1) If people consider using the deep learning method, seal imprint dataset is an imbalanced dataset with only one sample for many categories. Besides, some samples include blur, distortion, noise, etc.

(2) On the seals, there are not only the design of patterns, but also text information. These characters are generally meaningful, but they are often expressed as ancient characters that are difficult to understand by non-expert users. Using existing resources to identify these characters is a challenge.

(3) As shown in the examples in Table 2.7 and Figure 4.1, the individual samples of the same category are quite different.

(4) The application of retrieval results and improving the speed of retrieval in large-scale databases is a challenge.

This chapter introduces research based on the above problems and challenges.

4.2 Contribution

Table 4.1 The main tasks and utilizations

Data	Retrieval object	Proposed methods	Utilizations
Ukiyo-e artist's seals	Whole seal image	A feature search algorithm based on tree structure is proposed. [44][42]	The demo system that can search the entire seal and isolated characters. According to the search results, the seal owner's relationship network can be extracted automatically. [44]
Collector's seals	Character	Character segmentation method based on unsupervised clustering is proposed. [40][41][43]	

In this research, seal imprint data is targeted and proposed methods for the retrieval task based on entire seal and character level. Table 4.1 shows the main tasks and utilizations of this study. According to different experimental objectives, different are tried methods and the implementation of a demo system. As the utilization of search result, information of seal owners and their work shown in the search results is extracted from multiple data sources, and relationship information of owners such as the teacher-student relationship between the ukiyo-e artists is extracted for the visualization.

4.3 Related Work

4.3.1 Seal retrieval

Many studies have already been proposed on the seals retrieval and identification of ukiyo-e authors [45][46]. Hirose et al. [45] prepared a dictionary by preprocessing the segment character images for the characters in ukiyo-e and using the weighted direction

index of the characters as a feature. The weighted direction of the character is pretreated by the preprocessing of the cut character image. Identification of ukiyo-e artists by obtaining pseudo Mahalanobis distance between query images and dictionaries. Oohara et al. [46] proposed a method for extracting ukiyo-e seal strings. On the other hand, as a system for automatic analysis of the seal imprints, a seal retrieval technique that differs from related technologies in that it is specifically for Chinese antique document images is proposed by Fujitsu [47]. A color separation technique is used to separate the seal from the background image and then a two-level hierarchical matching method based on global feature matching is applied.

In recent years, research on image retrieval systems using deep features has been actively conducted. The construction of a search system that utilizes the deep features of the seal plays an important role in humanities research. Onizuka et al. [48] proposed a retrieval method of symbolic Kaou character with the feature extracted by the convolutional autoencoder. Aoike et al. [49] proposed a method that automatically extracts the illustration area of the material image including the seal image. Su et al. [50] proposed a seal imprint verification system featuring edge difference that uses support vector machine (SVM) to classify the proposed feature. This system is aimed at the entire content of a seal, so the retrieval scope depends on the existing seal features in the database. Sun et al. [51] proposed a Chinese seal image character recognition method based on graph matching. In this method, a skeleton feature extracted from each character is used to construct a graph, where the nodes of the graph are typically branch points, turning points, and endpoints of the character strokes. The strokes are represented by the edges of the graph. The most similar reference character is selected by calculating the matching score between two graphs. A single character database is used in this process,

but the segmentation method has not been explained in detail.

4.3.2 Character segmentation for OCR

Most seals are personal assets and consist of meaningful, separate, and unique characters, so it makes sense to extract a single character from a seal and then analyze it individually.

There have been several studies on character segmentation in historical documents.

Zahan et al. [52] proposed a segmentation process for a printed Bangla script. It has a good performance in segmenting characters with topologically connected structures; however, the target script is quite different from the target script in this research. Nguyen et al. [53] proposed a two-stage method to segment Japanese handwritten texts that utilizes vertical projection and stroke width transformation for coarse segmentation and bridge finding and Voronoi diagrams for fine segmentation. While this method performs very well for the segmentation of Japanese handwritten characters, seal characters usually have irregular positions or distributions and notable differences in character size, which may have a negative impact on the segmentation result when using this method. Liu et al. [54] also introduced a character segmentation method for Chinese seals that focuses on two types of circular seals. However, since the collector's seal data in the present study have irregular edges and irregular character distribution, concrete means to segment and extract a single character from various seals with different boundaries is needed. Ren et al. [55] have put forward several character segmentation methods for circular and elliptical Chinese seal imprints. One of these methods fits the contour of a seal image into a geometrical shape (e.g., a circle or an ellipse), then uses a mathematical transformation to convert it into a rectangular region, and finally obtains a single character according to the calculation of the horizontal distribution.

4.4 Methods

This section introduces proposed methods on seal image analysis. Section 4.4.1 introduces the retrieval method based on the complete image of the seal, as well as the automatic extraction of the relationship network of ukiyo-e artists. Section 4.4.2 shows the single character extraction method based on the collector's seal image. It also discusses the retrieval-based character recognition method for isolated characters.

4.4.1 Ukiyo-e artist's seal retrieval and artists' relationship extraction

This research focuses on automatic identification of signatures and seals in ukiyo-e collections, and use open data to provide scholars with a new perspective to ukiyo-e collections. Figure 4.2 shows the implementation framework of the algorithm introduced in this Section.

Based on the logic of the hierarchical structure tree [56], a tree-based seal image retrieval method was proposed that uses the whole seal image to find the most similar image from a database of 20,000 seal images.

As shown in Figure 4.3, according to the characteristics of seal design, we have set up two retrieval modes in the system. An approximate search method was proposed, this method uses a pre-trained neural network model to extract deep features [57] from a large-scale image data and these data are arranged into a tree structure using distance calculation [58].

Due to the large number of images in the retrieval range, the deep feature extracted from the existing model has the problem of relatively large vector size. For this problem, an approximate-based retrieval method is proposed.

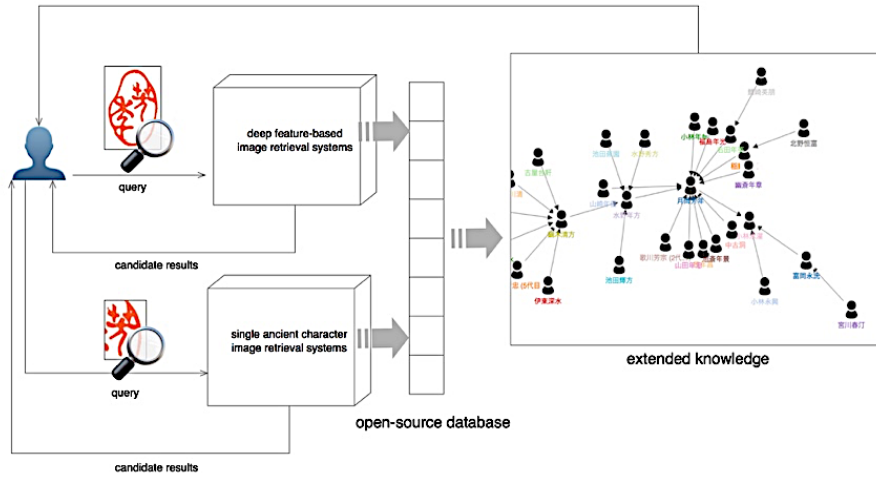


Figure 4.2 Complete system framework and application of search results

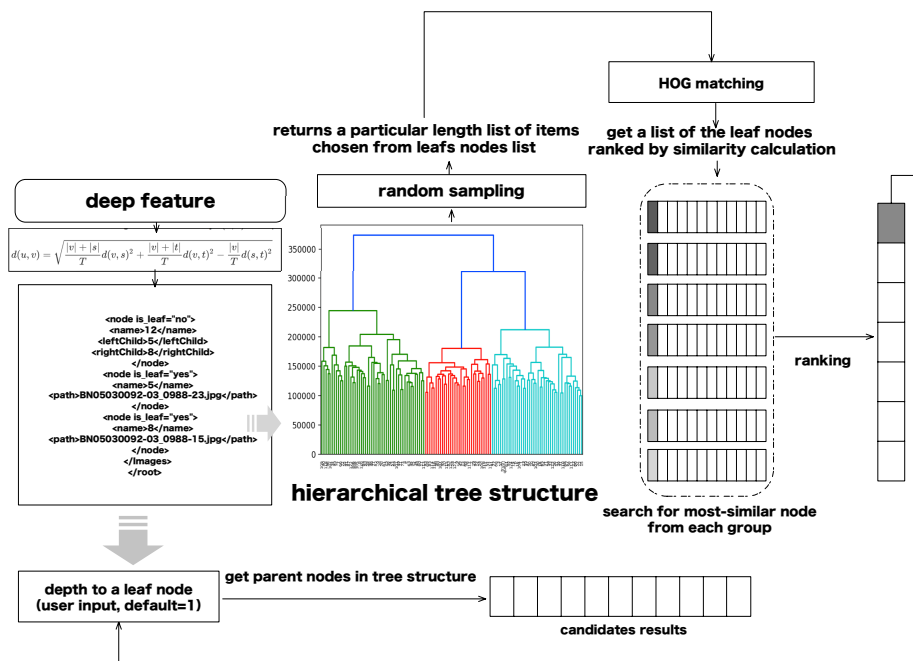


Figure 4.3 Matching process of the proposed method

This method consists of the following steps, sampling calculation to filter the clustered leaf nodes to the groups, extracting the detailed features [59][60] and comparing with a query to sort the sample leaf node groups, and finally predicting the location of the most similar image. As users need to find more similar images, bottom-up search is used

to expand the candidate images.



Figure 4.4 Seal retrieval on whole seal image



Figure 4.5 An example of a seal overlaps with handwritten words

It consists of three steps for searching the hierarchical tree. As shown in Figure 4.4, the search for the entire seal stamp uses 1) user selection of the seal stamp area, 2) extraction of the seal stamp area, and 3) extraction of deep feature and HOG (Histograms of Oriented Gradients) feature.

As shown in Figure 4.5, there are many images in which the stamp area and handwritten characters overlap in the ukiyo-e image data. Therefore, it is an important task to first extract the stamp area from the handwritten characters. Based on the color information of the image, the seal area is extracted by k-means clustering [61] to cluster

image color information. As shown in Figure 4.6, the information of three RGB channels of an image are projected into a three-dimensional space.

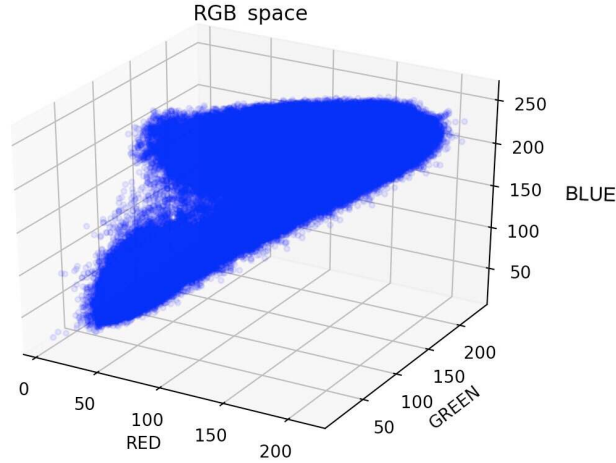


Figure 4.6 Three-dimensional representation of color information

Hence, the Euclidean distance is used to represent the color relationship between two pixels, which is defined by:

$$D_{rgb} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}, \quad (4.1)$$

where $R_1, R_2, G_1, G_2, B_1, B_2$ represent RGB values for pixels 1 and 2, respectively, and D_{rgb} is used to cluster pixels with similar colors. According to the principle of the k-means algorithm, this task is regarded as extracting K groups of pixels with similar colors from images. Proposed system automatically extracts areas with more red components.

HOG features are extracted from the ukiyo-e artist's seal image data, and structural data is generated by calculating the Ward variance minimization distance. The generated tree structure is automatically saved in the XML file. The image data appears in the leaf node, and if the tag is "is leaf = yes", it indicates that the node is a leaf node.

Table 4.2 Extracting a seal area from an ukiyo-e print

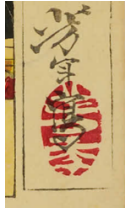
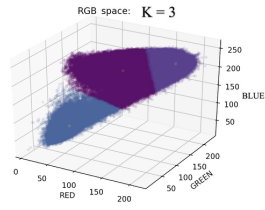



Input image	Clustering results	Pixel group 1	Pixel group 2	Pixel group 3
				

Table 4.3 XML document design for tree search-based matching process

Items	Tags
Node ID	<id></id>
ID of the child node to the left of the node	<leftChild> </leftChild>
ID of the child node to the right of the node	<rightChild></rightChild>
Node	<node is_leaf="(yes/no)"></node>

In this research, a method is proposed for speeding up the search by random sampling. As shown in Figure 4.7, if the user selects the rapid search function "quick search", which divides the entire leaf node equidistantly, the system samples the leaf node for each section based on the configurable probabilities, then proceeds to image similarity calculation using deep features.

Figure 4.8 shows how to calculate the similarity between the leaf node group acquired by sampling and the query image.

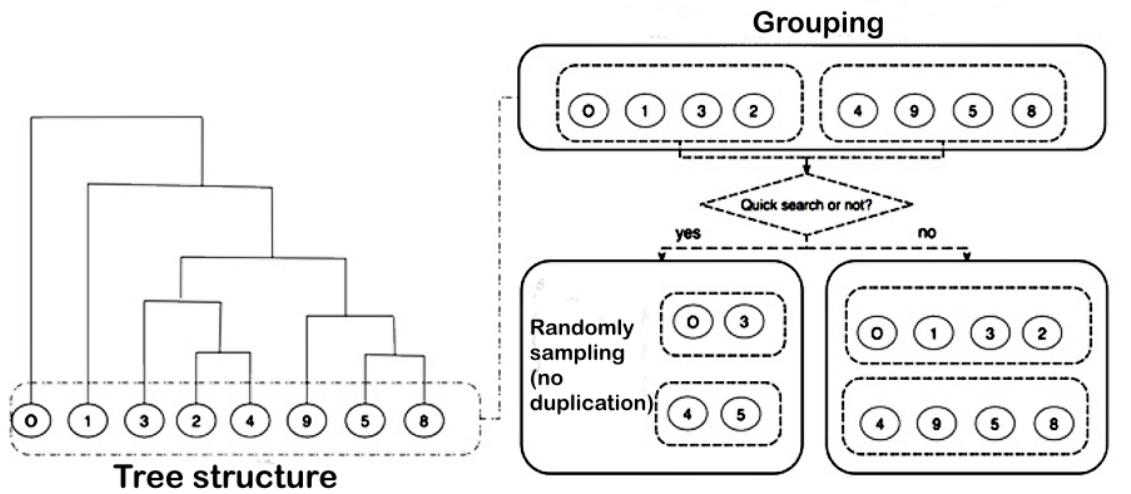


Figure 4.7 Tree structure and random sampling for quick search and normal search

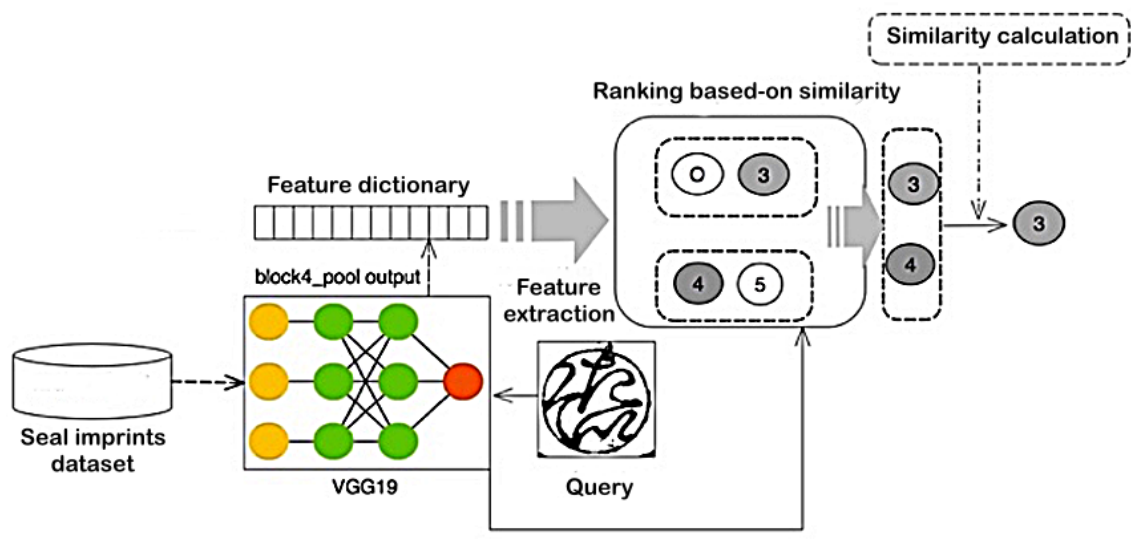


Figure 4.8 Obtain the ranking result base on similarity calculation

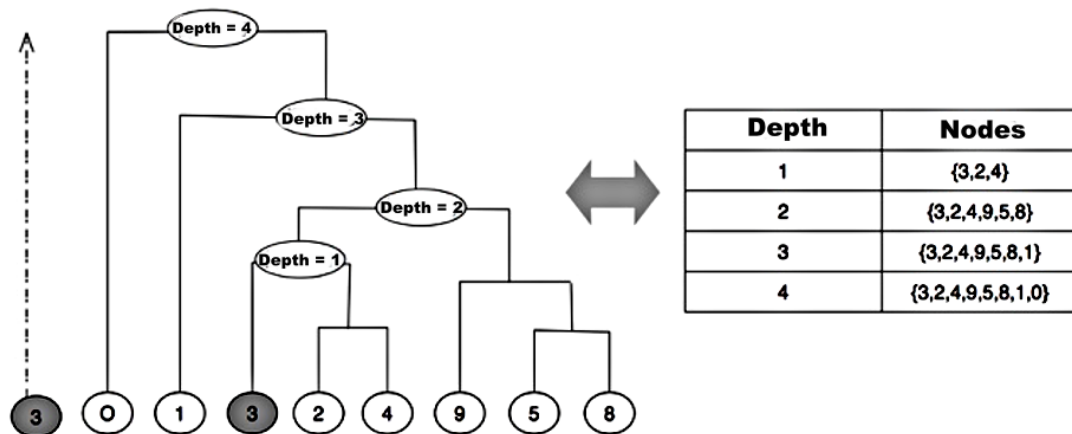


Figure 4.9 Extending the candidate results

Using the trained model of VGG19 [62] as a feature extractor, the cosine similarity calculation of the VGG19 intermediate layer output of the query image and the leaf node image is performed.

The image with the highest degree of similarity is extracted from each group, and the node representing it is searched for.

As shown in Figure 4.9, the search depth is set according to the number of candidate results specified by the user, and candidate images with high local area similarity of the image are extracted. Figure 4.10 shows an example of the search results by the proposed system.



Figure 4.10 Examples of the candidate results

From the retrieval results, to respond to the user’s specific interests in ukiyo-e artists, some additional information could be explored as a list of recommended candidate artists by linking artist names to open data. The current implementation of the proposed method is shown in Figure 4.11, for instance, ukiyo-e artists could be influenced by their masters and ancestors.

The relationships between ukiyo-e artists can be visualized by using “student” and “instructor” links in Wikidata items, and a simple data visualization prototype system is developed, as shown in Figure 4.11. A simple data visualization sample is available on GitHub [63].

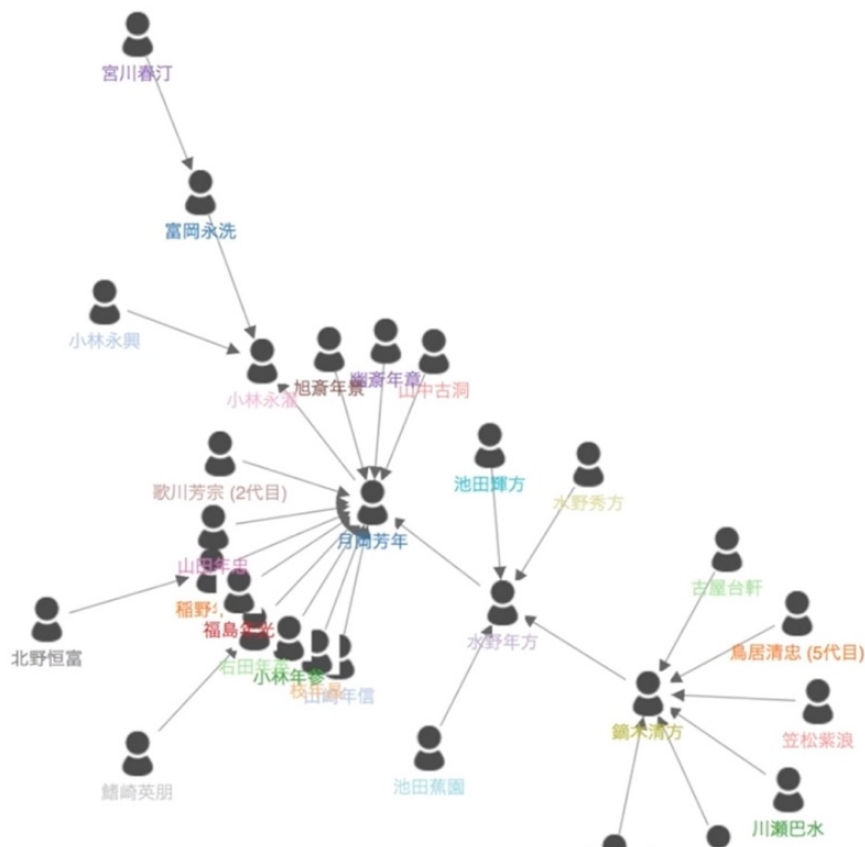





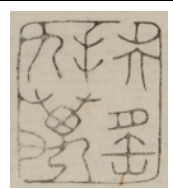


Figure 4.11 Artists relationship extraction

4.4.2 Character segmentation for collector's seal retrieval

Character segmentation task is mainly for collector's seal images. The main shapes of the seal image this study focus on are shown in Table 4.4.

Table 4.4 The shapes of collector's seals

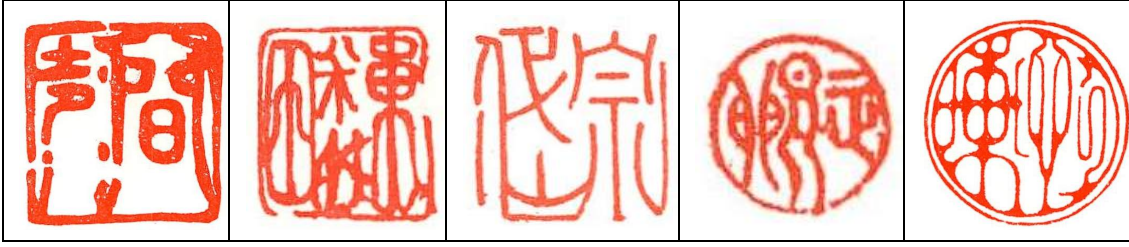
(a)	(b)	(c)	(d)	(e)	(f)
					

These shapes include (c) rectangles, (a)(d)(f) squares, (b) circles, and (e) seals with irregular edges. The proposed method is suitable for seal imprints in which non-experts can identify the boundary of each character. People could not recognize the segmentation ground truth of some of the seals because they featured irregular characters and images, sometimes even mixed together, and only experts would be able to identify the segmentation ground truth.

In this study only seals that have an easily recognizable layout are examined. Examples of some seals that are not included in this experiment are provided in Table 4.5.

To determine a single character, clustering can be used to segment each character. Because kanji characters are independent and balanced in structure, we regard every character as a module, and each module has its centroid.

Table 4.5 Seals outside the scope of this research



By considering these centroids, clustering can be used to extract character fields. Since the coordinates of each pixel of the seal areas can be seen as known information, the density information of each pixel can be obtained by kernel density estimation. The values are calculated by:

$$\hat{f}_{bandwidth(X,Y)}(x,y) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2\pi bandwidth(X,Y)} \exp\left(-\frac{((x-X_i)^2+(y-Y_i)^2)}{2\pi bandwidth(X,Y)^2}\right), \quad (4.2)$$

where X and Y are the vectors $X = \{x_0, x_1, \dots, x_n\}$, $Y = \{y_0, y_1, \dots, y_n\}$ extracted from the foreground pixel set $\{(x_0, y_0), (x_1, y_1) \dots (x_n, y_n)\}$, n is the number of elements in vector X or Y , X_i is the i_{th} element in vector X , and Y_i is i_{th} element in vector Y . $bandwidth(X, Y)$ refers to the optimal bandwidth values.

The mean-shift clustering [64] is used to cluster the pixels of an image. The clustering results can be optimized by adjusting the bandwidth to consider different variables. Figure 4.12 shows the clustering results under different bandwidth settings when the input is not a normalized image.

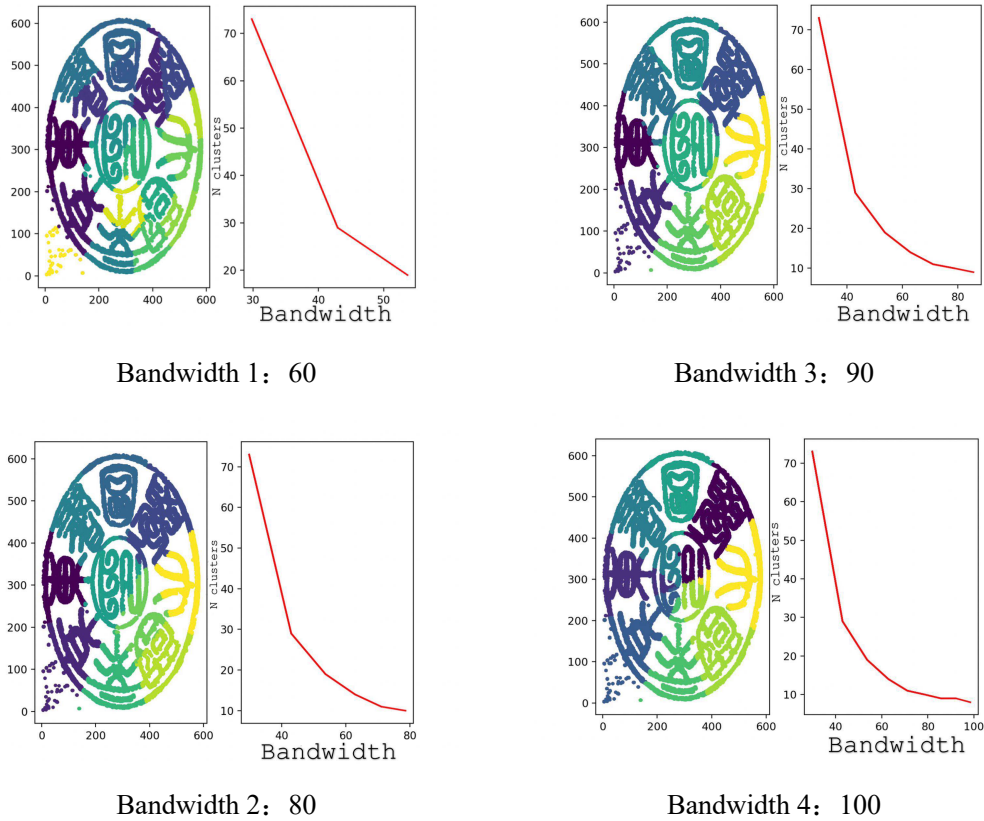


Figure 4.12 Visualization of clustering results under different bandwidths

In each unit, the graph on the left side is the clustering result of each pixel of the seal, and each color represents a different cluster. For the graph on the right side, the X-axis is the bandwidth value and the Y-axis is the number of clusters.

The results are selected as shown in Figure 4.13. When the change rate of the total number of clusters becomes steady—for example, when the bandwidth is about 90—each cluster in the result are treated as a candidate result of the character segmentation.

The bandwidth interval is calculated, and the bandwidth value is obtained equidistantly in the interval. These bandwidth values are then used to obtain the segmentation candidates. The algorithm is shown in Algorithm 1.

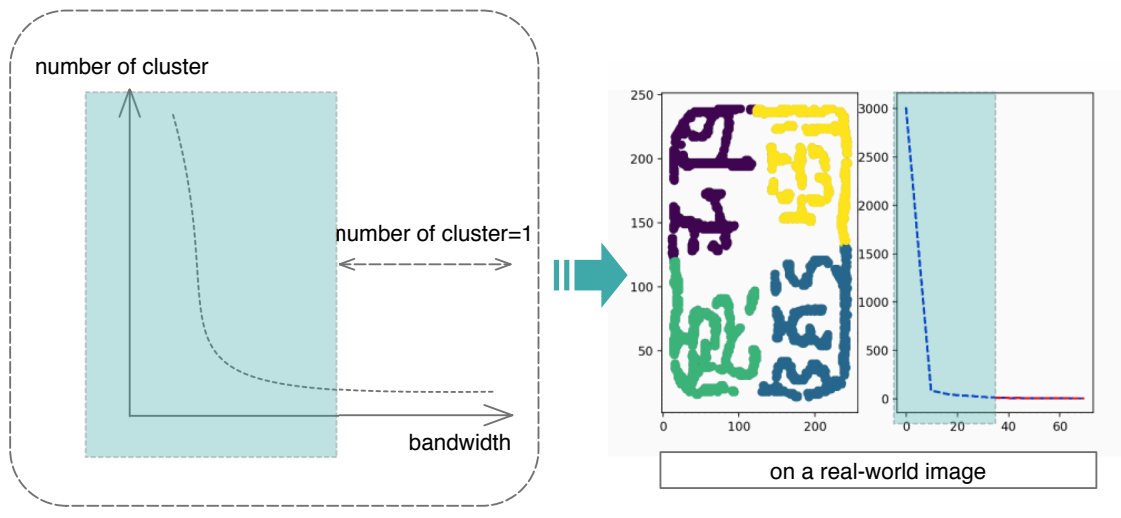


Figure 4.13 Obtaining the segmentation results

Algorithm 1

Input:

F: coordinate set $\{(x_0, y_0), (x_1, y_1) \dots (x_n, y_n)\}$ from foreground pixel obtained by K-means clustering

R: Randomly generated sorted set $\{r_1, r_2, r_3, r_4 \dots r_i\}, r_i \in [0.0, 1.0]$ for bandwidth estimation, default $i=100$

Output: Set of character location hypotheses **U**

- 1: Compute the length of **F**, $len_F = len(\mathbf{F})$
 - 2: Initialize set of estimate_bandwidth **Bandwidth** = { }
 - 3: **For each** $r_1 \in \mathbf{R} \{r_1, r_2, r_3, r_4 \dots r_i\}$ **do:**
 Set the number of neighbors = $r_i * len_F$, compute the nearest neighbor for each coordinate in **F**
 For each coordinate in **F**, compute farthest neighbor and distance between them, get a set of farthest distance
 $Distance_{farthest} = \{dis_{farthest}(x_0, x_0), dis_{farthest}(x_1, x_1) \dots dis_{farthest}(x_n, x_n)\}$
 Compute the average of **Distance**_{farthest} get
 $Average_{(r_i, farthest\ neighbour)} = \frac{\sum_0^i dis_{farthest}(x_0, x_0)}{len_F}$
 Add $Average_{(r_i, farthest\ neighbour)}$ into **Bandwidth**
 - 4: For each $\{b_1, b_2, b_3, b_4 \dots b_i\}$ in **Bandwidth**, get the result of number of clusters $\{Nclusters_{b_1}, Nclusters_{b_2}, Nclusters_{b_3}, \dots Nclusters_{b_i}: Nclusters_{b_i} \in f_{meanshift_{clustering}}(b_i)\}$
 - 5: Fit a polynomial $Q(b)$ with set $\{(b_1, Nclusters_{b_1}), (b_2, Nclusters_{b_2}) \dots (b_i, Nclusters_{b_i})\}$
 - 6: Get the second derivative $Q''(b) = \frac{d^2 Nclusters_{b_i}}{db_i^2}$ of $Q(b_n)$
 - 7: Initialize set of Descent_bandwidth **Descentband** = { }
 - 8: **For each** $\{b_1, b_2, b_3, b_4 \dots b_i\}$ in **Bandwidth**,
IF $Q''(b_i) < 0$
 Add b_i into **Descentband**
 - 9: Get sorted **Descentband** = $\{b_{Descent_1}, b_{Descent_2}, \dots b_{Descent_n}\}$
 - 10: Set an interval $Interval_{descentband}, default = 5$
 - 11: Group **Descentband** by $Interval_{descentband}$, get set
 $G_{interval} = \{Group_{descentband1}, \dots Group_{descentbandn}, (\{b_{Descent_{n-Interval_{descentband}}}, b_{t_{n-Interval_{descentband}+1}}, \dots\} \in Group_{descentbandn})\}$
 - 12: Compute the standard deviation of each group in $G_{interval}$, get $STD_{group} = \{std_{group1}, std_{group2} \dots std_{groupn}\}$
 - 13: Compute the minimum value of STD_{group} , get the group **Candidates** from $G_{interval}$ corresponding to the minimum value
 - 14: Initialize set of character location hypotheses **U**
 - 15: **For each** $\{b_{candidate1}, b_{candidate2}, b_{candidate3}, \dots b_{candidaten}\}$ in **Candidates**,
 Compute the result of clustering $f_{meanshift_{clustering}}(b_{candidaten_i}), i \in n$
 Classify the **F** use the output labels of clustering, add the set
 $Character_{candidate_i} = \{(x_{labeli_0}, y_{labeli_0}), (x_{labeli_1}, y_{labeli_1}) \dots (x_{labeli_n}, y_{labeli_n}): (x_{labeli_n}, y_{labeli_n}) \in \mathbf{F}\}$, with same label i into **U**
-

As shown in Figure 4.14, the segmentation results under adjacent bandwidth may have a large area of overlap.

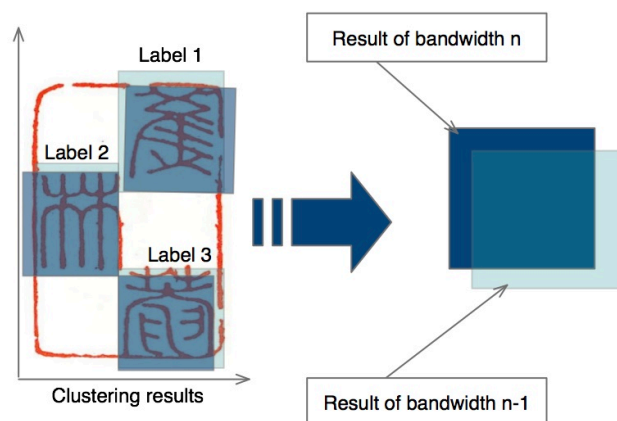


Figure 4.14 Clustering results under adjacent bandwidth

We only keep the larger of the two candidate areas that overlap more than 90% of the area of any one of them. The algorithm implementation is available on GitHub [65] as a reference.

Based on the result of character segmentation, The preliminary attempt of retrieval-based character recognition task is applied by using Shirakawa font typeface file [66] which was strictly made under the supervision of experts. The typeface images extracted from the font file are shown in Figure 4.15.

Modern commonly used characters	‘人’	‘文’	‘科’	‘学’
“Shirakawa font” Typeface				

Figure 4.15 Images converted from a font file

As there are many variations of ancient characters, even a slight change of the

structure will affect the extraction of geometric features. The typeface images are normalized with the maximum and minimum coordinates of the black pixels and then standardize them to the size of 225×225 .

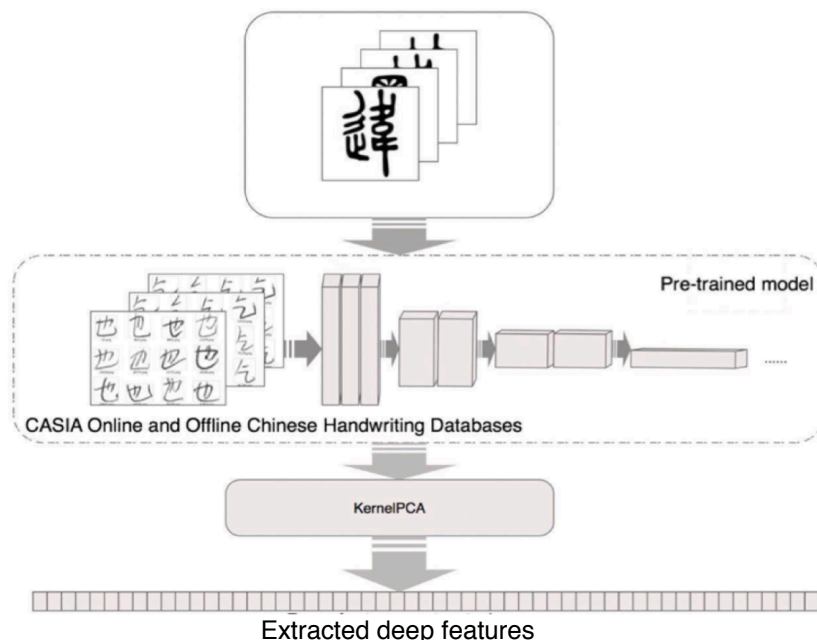


Figure 4.16 Extraction of CNN features

Pre-trained model based on convolutional neural network is used to extract the deep features of fonts, with the aim of subtracting minor changes in a character's structure, to enhance the robustness of search results.

As shown in Figure 4.16, to give transfer knowledge to make the extracted character feature more focused on its geometric particularity, besides the pre-trained model which was trained by ImageNet and CASIA Online and Offline Chinese Handwriting Databases [67], in which the characters have some shape features in common with ancient kanji characters, as the training data to train the model using VGG16 [68].

The visual expression of the feature map in the max pooling layer of the pre-trained model is shown in Figure 4.17. KernelPCA [69] to reduce the dimensions of the output

from the middle layer.

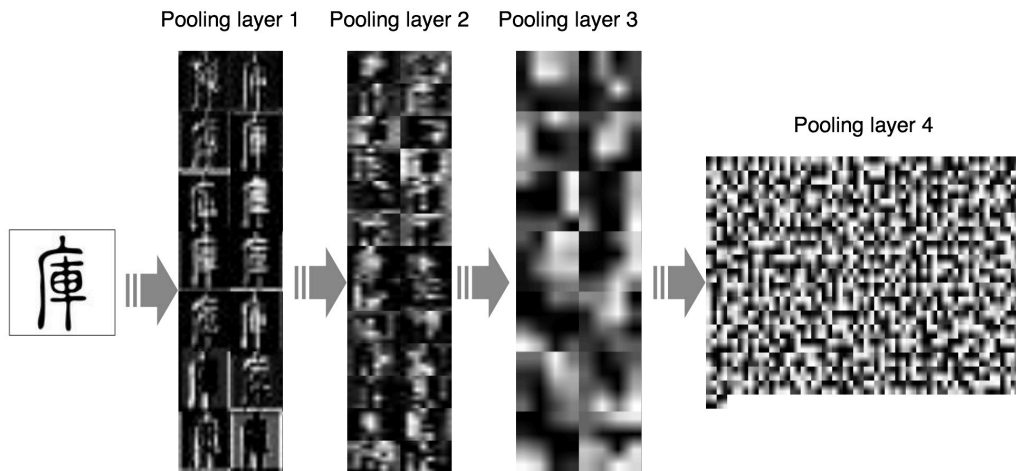


Figure 4.17 Visual expression of feature map in max pooling layer

A character's skeleton map is obtained by using the method of Zhang et al. [70], as shown in Figure 4.18. In contrast to a general thinning method, Zhang's method ignores the existence of the stroke width. It can obtain a skeleton map without noises to represent each stroke by a unique corresponding continuous single pixel.

Then we use the Harris corner [71] to obtain the coordinates of the intersection of each stroke. The coordinate points of the skeleton map and of the stroke intersections are then stored in the database as one of the representations of geometric features.

The matching process is shown in Figure 4.19. The same feature extraction method is applied to the segmented user query image and the typeface image to extract the corresponding features, including CNN and geometric features.

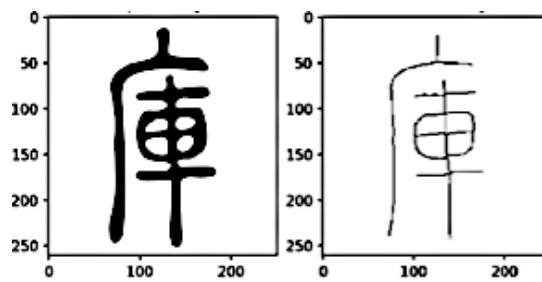


Figure 4.18 Skeleton map of a character

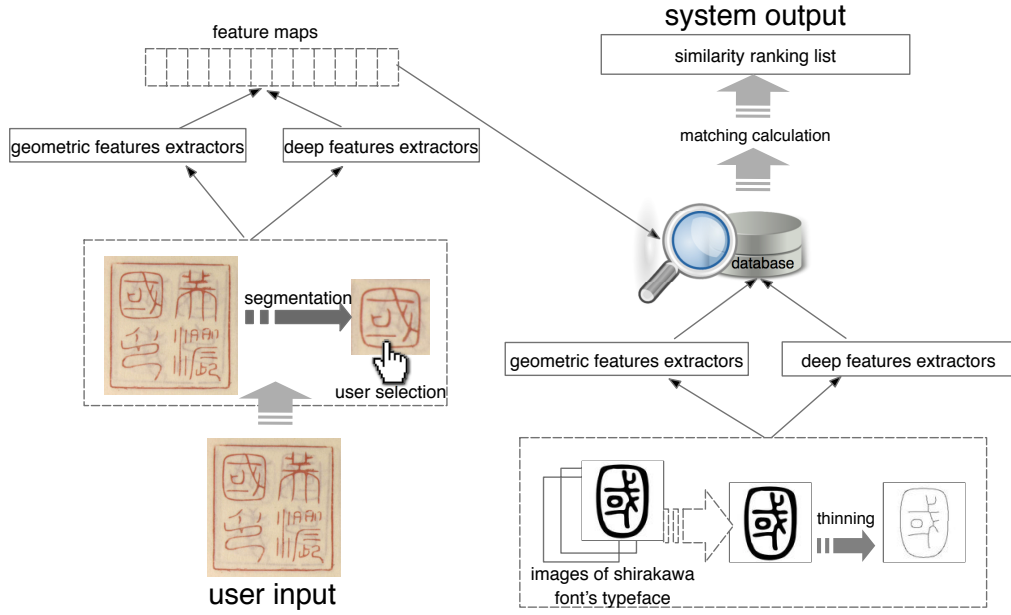


Figure 4.19 Matching process

Due to the different training data used, the numerical benchmark of the learned feature space is different. We therefore normalize the results of each similarity calculation into scores, and each feature is then given a different weight corresponding to its score.

The results of distance calculation are normalized to values in the range [0, 1] by using min-max normalization, and the function is defined by:

$$S_{distance_i} = \frac{distance_i - \text{Min}(\{distance_1, distance_2, \dots, distance_n\})}{\text{Max}(\{distance_1, \dots, distance_n\}) - \text{Min}(\{distance_1, \dots, distance_n\})} \quad (4.3)$$

where $i=0, 1, \dots, n$, n is the number of images in the database, $S_{distance_i}$ is the similarity score between image i and the query image, and $distance_n$ is the distance between image i and the query image calculated by using different distance formulas. The multi-feature similarity score is calculated by:

$$S_{total} = \frac{W_{cf} S_{cnnFeature} + W_{gf} S_{geometricFeature}}{W_{cf} + W_{gf}} \quad (4.4)$$

where S_{total} is the total score of similarity, $S_{geometricFeature}$ is the sum of similarities of geometric features calculated using distance formulas, and W_{cf} and W_{gf}

are the weights of the similarity score of CNN feature $S_{cnnFeature}$ and geometric feature $S_{geometricFeature}$. Finally, we use the total score of similarity to predict the input image's category.

4.5 Experiments and Results

We conducted evaluation experiments on the study described in Section 4.4.1 and Section 4.4.2 respectively. Section 4.5.1 describes the experimental results of ukiyo-e artist seal retrieval, and Section 4.5.2 describes the experimental results of the proposed method of character segmentation and recognition on collector's seal data. Section 4.5.3 shows character retrieval results.

4.5.1 Evaluation experiments on ukiyo-e artist's seals retrieval

To evaluate the search accuracy of the system, we collected 100 query images from the subjects. All of these images are selected by experiment participants from the Ritsumeikan University Art Research Center Japanese woodblock prints database and include the seals of the 10 ukiyo-e artists shown in Table 4.6. A part of the test data is shown in Figure 4.20.

Table 4.6 Seals of the 10 ukiyo-e artists

Ukiyo-e artists	
北尾政美 (Kitao Masayoshi)	月岡芳年 (Tsukioka Yoshitoshi)
小林永濯 (Kobayashi Eitaku)	歌川豊国 (Utagawa Kunisada)
尾形月耕 (Ogata Gekko)	鳥居清長 (Torii Kiyonaga)
月岡耕漁 (Tsukioka Kogyo)	西川祐信 (Nishikawa Sukenobu)
月岡雪鼎 (Tsukioka Settei)	祇園井特 (Seitoku Gion)



Figure 4.20 A part of experimental data

As an evaluation scale of the search result, the ratio of the query containing the correct answer, which was the ratio of the search query containing the correct answer in the search result with the depth of ‘tree search = 1’, was taken. The experimental results were shown in Table 4.7.

Table 4.7 Retrieval accuracy

Ukiyo-e artists	Number of queries	Accuracy
北尾政美	6	16%
小林永濯	5	100%
尾形月耕	11	81%
月岡耕漁	9	22%
月岡雪鼎	1	0%
祇園井特	5	80%
西川祐信	10	50%
鳥居清長	10	70%
歌川豊国	22	77%
月岡芳年	21	76%
Total/Average	100	66%

4.5.2 Character segmentation results

For the evaluations of character segmentation, we used several different types of seals to

test our proposed segmentation algorithm. 100 seal print samples were selected from the Collectors' Seal Database for evaluating the proposed character segmentation method. The results are shown in Table 4.8. Our method achieved good results for processing images with irregular character distribution.

However, as indicated by Result 3, larger characters were segmented into the candidate areas of other characters, so further research on dealing with images with different character sizes and close character spacing is required. The segmentation results for seals of different shapes are shown in Figure 4.9.

Table 4.8 Segmentation results in square seal images





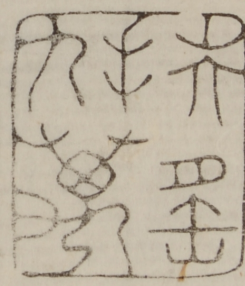




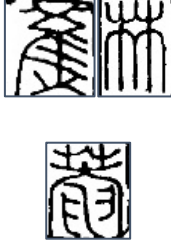






1: Regular seal	Result 1	2: Characters with irregular glyph	Result 2
			
<p>3: Irregular character distribution</p>	<p>Result 3</p>	<p>4: With noisy background and overlaps with handwritten words</p>	<p>Result 4</p>
			

Table 4.9 Segmentation results for seal imprint with different shapes

1: Rectangle	Result 1	2: Circle	Result 2
			
3: Irregular shape (a)	Result 3	4: Irregular shape (b)	Result 4
			

These results demonstrate that our proposed method can deal with the seal imprints of different shapes.

Among several different evaluation metrics of character segmentation such as [72] [73], etc., we choose the method proposed in the ICDAR contest [74] to calculate the segmentation score. We evaluate the $\text{Score}_{\text{Segmentation}}$ as:

$$\text{Score}_{\text{Segmentation}}(G, R) = \frac{\text{Area}(G \cap R)}{\text{Area}(G \cup R)} \quad (4.5)$$

where R is the segmentation results and G is its ground truth.

Characters were segmented correctly when $\text{Score}_{\text{Segmentation}} \geq 0.75$, where TP (true positive) is the number of characters segmented correctly. FP (false positive) is the segmentation results which do not overlap with corresponding ground truth. FN (false negative) is the number of ground truth characters that remain unsegmented. Precision is calculated by:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{TP}{\text{all segmentation results}} \quad (4.6)$$

Recall is calculated by:

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{TP}{\text{all ground truths}} \quad (4.7)$$

Our method is based on a set of random initial values. Table 4.10 shows the results of the calculations based on the results generated by the best set of initial values.

Table 4.10 Performance of our proposed method

	Precision	Recall
Square shapes	0.73	0.81
Irregular shapes	0.85	0.89

It can be seen that more candidate areas and lower recall scores were caused when character spacing was not prominent. Because the proposed method is limited in terms of data, it can be seen from the Table 4.10 that the accuracy on square shapes need to be improved. The reason is that the interval between the seal characters of square shapes is narrow, and they are difficult for proposed method to detect the interval of character units.

4.5.3 Character retrieval results

Because the font typeface dataset keeps only one sample for each category, it is unfavorable to select the best features and method for distance calculation to classify characters with a different shape. For selecting features and distance calculation methods, we performed experiments on a standard dataset and selected Japanese Hiragana data from the Omniglot dataset [75], in which the character shapes are different from Chinese characters. There are 2,586 images for ‘Tenbun’ typeface in the Shirakawa font dataset, so our retrieval scope includes all 2,586 images in different categories.

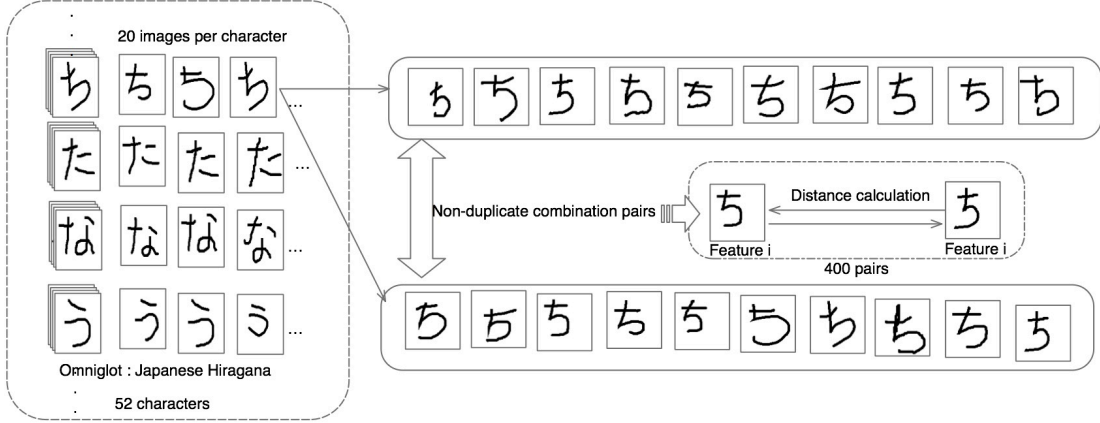


Figure 4.21 Experiment on feature and distance calculation method selection

As shown in Figure 4.21, the experiments on feature selection were done by calculating the average similarity from the same character. We defined the average similarity of the same character q as Sim_{ave_q} . There were 20 images for one character in the Omniglot dataset. We defined $P_{character_q}$ as a set of all the generated non-duplicate combination pairs from 20 images, and Sim_{ave_q} can be obtained by calculating the average similarity of $P_{character_q}$. Hence, we evaluated the different features by:

$$S_{average_similarity} = \frac{\sum_q^{L_{characterSet}} f_{feature_i}(Sim_{ave_q})}{L_{characterSet}} \quad (4.8)$$

where $S_{average_similarity}$ is the average similarity score for each feature, $f_{feature_i}(Sim_{ave_q})$ is the average similarity of character q in $feature_i$, and $L_{characterSet}$ is the total number of character categories.

We needed to choose the best image performance under geometric features and the best layer and pre-trained model to extract deep features. The original image and the skeleton map, shown in Table 4.11, were used in this experiment.

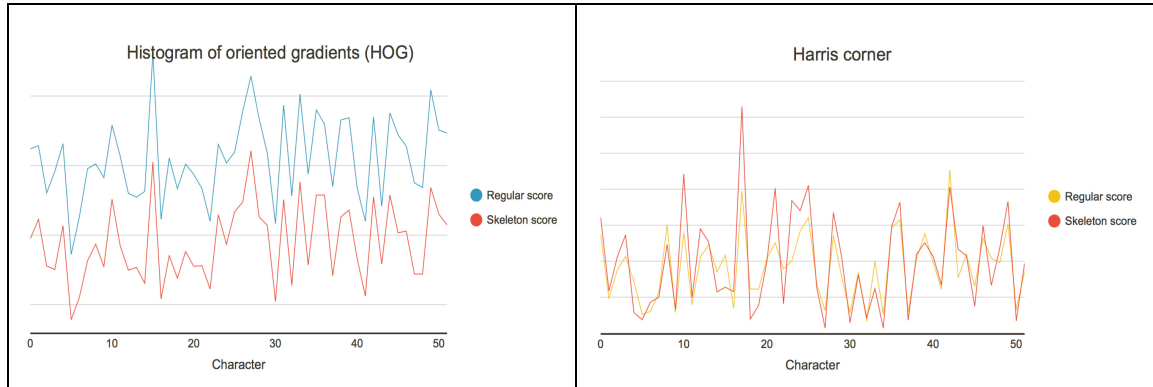
Table 4.11 Regular map and skeleton map used in matching experiments

Regular map	Skeleton map

We first experimented on geometric features. The histogram of oriented gradients (HOG) is one of the most efficient descriptors to extract specific and invariant local and global features. It enables the difference between similar characters to be distinguished by strokes, and the intersection of strokes is also an important feature. We used the Harris Corner Detector [76] to extract the difference between strokes.

Table 4.12 shows the Sim_{ave_q} of each character, where the horizontal axis is the index of the character and the vertical axis is the average similarity of each character.

Table 4.12 Experiment on feature and distance calculation method selection



The results show that better results can be obtained by using the regular image in the HOG feature and the skeleton map to obtain the Harris corner feature.

In order to select suitable layer and training domain to extract deep features from CNN-based model, we utilized the VGG19 [62] pre-trained model trained by ImageNet and the VGG16 pre-trained model trained by the handwritten Chinese characters dataset [67] as feature extractors. Features were extracted using the intermediate layer from

‘pool3’ to ‘fc2’ of these pre-trained models. Tables 4.13 and 4.14 show the $S_{average\ similarity}$ in deep features extracted from the model trained by ImageNet and by the handwritten Chinese characters dataset, respectively.

Table 4.13 Evaluation of features extracted from pre-trained model by ImageNet

	Regular map	Skeleton map
VGG19_fc1	0.79	0.82
VGG19_fc2	0.82	0.84
VGG19_pool3	0.58	0.59
VGG19_pool4	0.40	0.43
VGG19_pool5	0.53	0.58

Table 4.14 Evaluation of features extracted from model trained on handwritten Chinese characters dataset

	Regular map	Skeleton map
VGG16_fc1	0.979	0.997
VGG16_fc2	0.975	0.996
VGG16_pool3	0.992	0.999
VGG16_pool4	0.995	0.999
VGG16_pool5	0.978	0.997

The results show that the features extracted from the skeleton map had the highest similarity between characters. The skeleton map reduces the classification ability of the features extracted from the model.

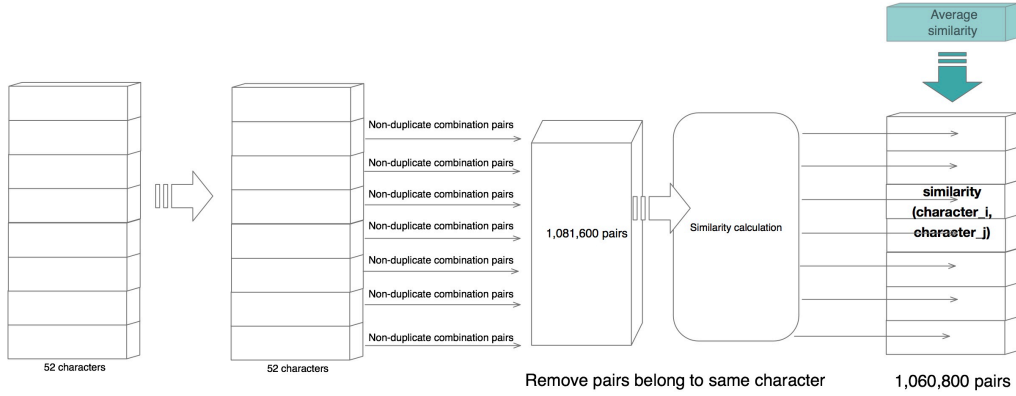
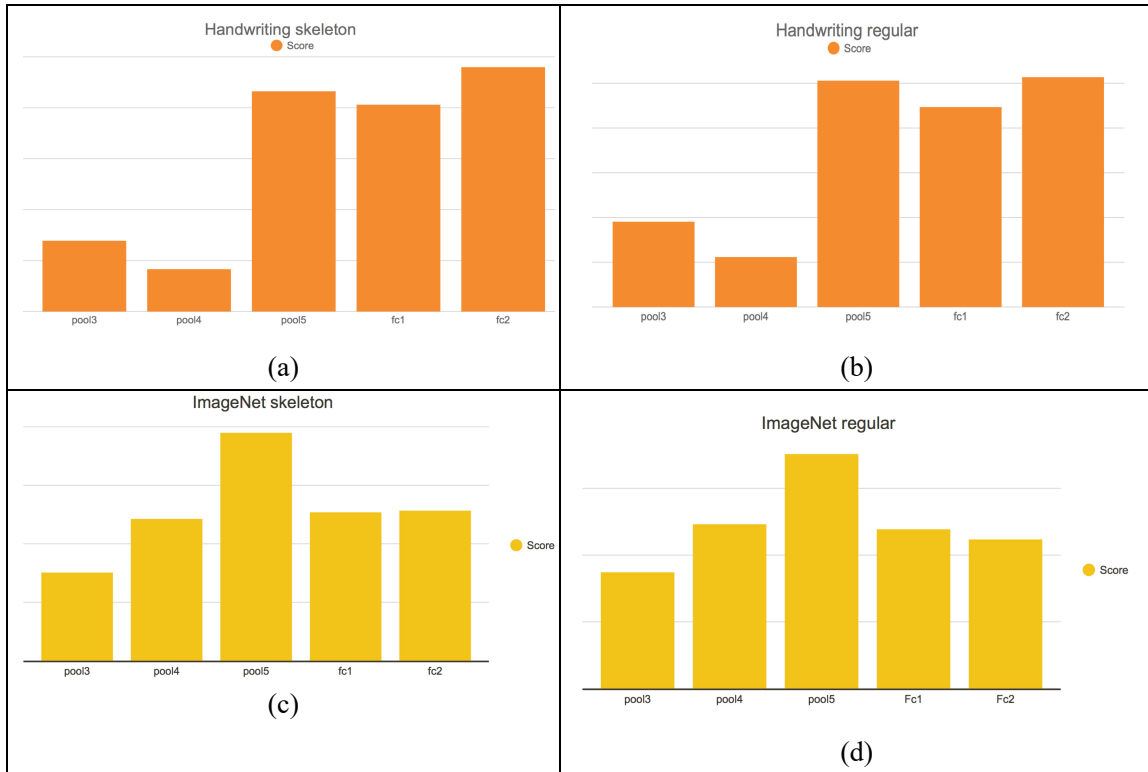


Figure 4.22 Calculating the distinction score of each character

The retrieval task needs to determine whether a certain feature can be distinguished from other characters. In Figure 4.22, we introduce the calculation of distinction score $Score_{distinction}$. Non-duplicate combination pairs were generated from all images from hiragana character data. We calculated the average similarity between pairs where neither element of the pair belonged to the same character. Finally, we calculated the average results to represent $Score_{distinction}$. The evaluation of a feature i is done by:

$$Score_{feature_i} = \left| S_{average_similarity} - Score_{distinction} \right| \quad (4.9)$$

Table 4.15 $Score_{feature_i}$ in different situations. (a) and (b) show the distinction score of regular image and skeleton map in ImageNet pre-trained model. (c) and (d) show the distinction score in pre-trained model by CASIA Online and Offline Chinese Handwriting Databases



The results in Table 4.15 show that the global image feature, which was extracted by the pool 5 layer, had the best performance.

Table 4.16 $Score_{feature_i}$ in different situations

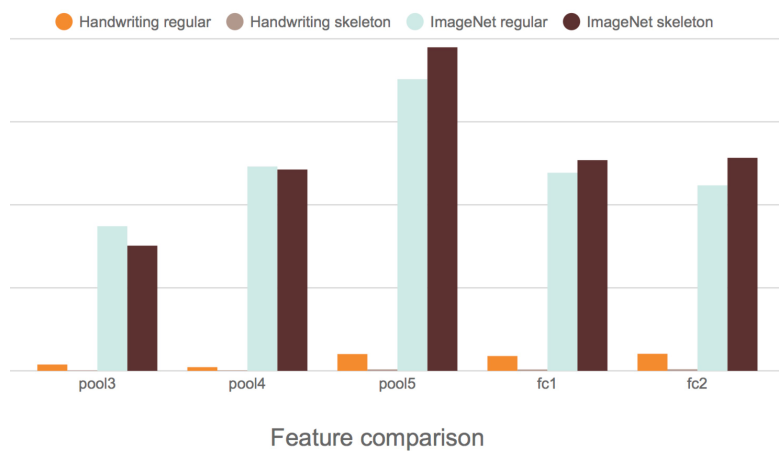


Table 4.16 shows a comparison of the $Score_{feature_i}$ results in different situations. It can be seen that the features extracted from the ImageNet pre-trained model with skeleton maps had the best discrimination ability. Based on the above experimental results, we choose the features shown in Table 4.17 to perform character retrieval experiments.

Table 4.17 Features used in seal imprint retrieval

Feature	Input image
Harris Corner Detector	Regular map
Histogram of oriented gradients	Skeleton map
VGG19_pool5(ImageNet)	Skeleton map

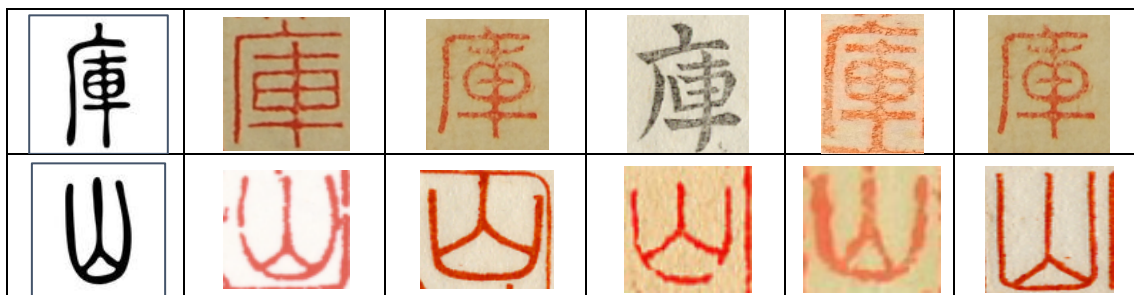
Here, we show the results of the 10 characters with the highest frequency in the printed text, which were calculated by Mean Reciprocal Rank (4.10):

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (4.10)$$

where Q is the total number of images retrieved, i is the image number, and $rank$ is the ranking order.

For each character, we tested with 20 images, and examples of the data used in experiment are shown in Table 4.18. The results are shown in Figure 4.23.

Table 4.18 Part of the data used in retrieval experiments



Most of the test data had a performance in the top-50 ranking results. Characters

with minor deformation were in the top-5 ranking results.

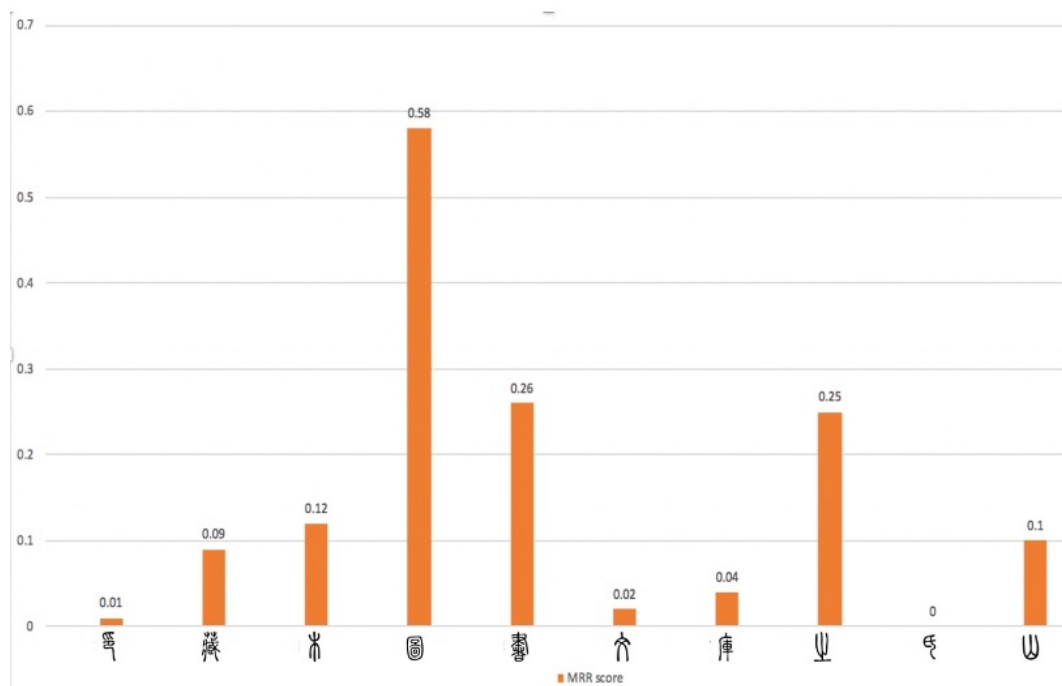


Figure 4.23 Evaluation on seal imprints character recognition

We have not found a suitable solution for the situation shown in Figure 4.24 and Figure 4.25, there are some incorrect result caused by pre-processing of query image, also there are some difference in the current version of the reference book. We will confirm and further organize the data in future research.

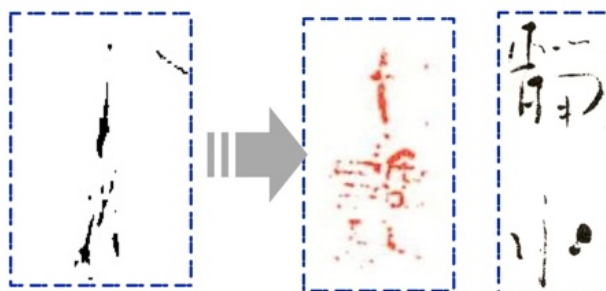


Figure 4.24 An incorrect result caused by pre-processing of query image

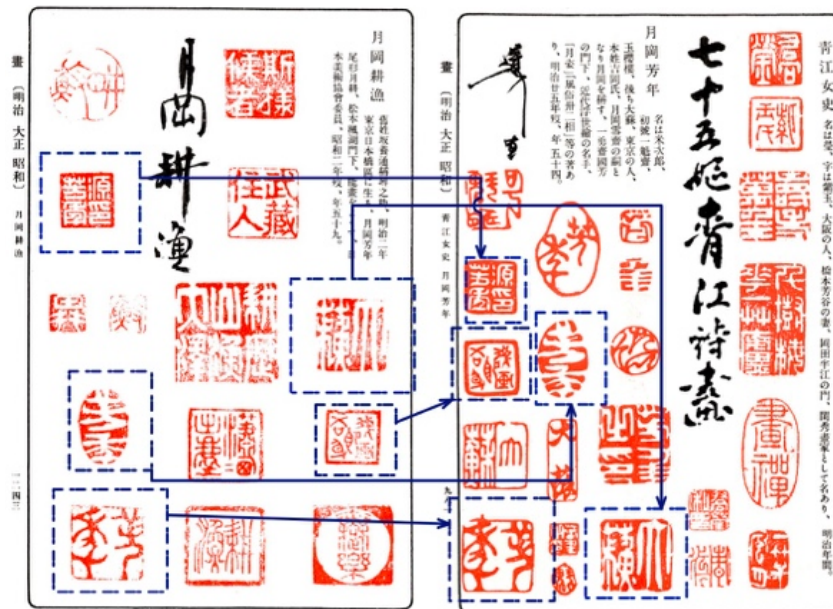

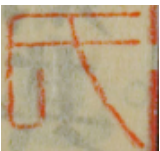



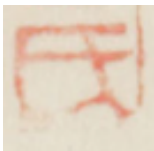


Figure 4.25 Some difference in the current version of the reference book

However, as shown in Table 4.19, the results of the similarity calculation were negatively affected when the query was significantly different from our standard typeface.

Table 4.19 Examples of poor results

Standard typeface	Query				
					

4.6 Summary

In this study, we attempted to construct a seal search system using multiple data sources. For seals of ukiyo-e artists with fewer characters, we propose a tree structure-based image retrieval method using deep features extracted from convolutional neural network.

In order to improve the search accuracy, it is possible to introduce the trained Siamese Network [77] for the calculation of image similarity. In the future, we will consider expanding the search range of seals to those owned by cultural figures such as writers, calligraphers, and painters, and high-speed search methods that support large-scale image data. In addition, it is a future study to automatically extract the personal relationship from the text of the person introduction sentences by machine learning or deep learning, and to examine the calculation method of the weight of the personal relationship using the person's work.

For the collector's seals which include some characters information, a new unsupervised-based character segmentation method is proposed to extract the single character from seal images for the collector's seal, which includes some characters information. We used two clustering algorithms to pre-process seal images and obtained good results in this work. In contrast to training a neural network, a clustering algorithm extracts part of the required information from an image without consuming computational resources. We used a combination of deep features and geometric features to retrieve ancient kanji characters by calculating similarities. This reduces the time it takes to re-train a model when adding a new category of characters and enables flexible usage of the intermediate output of neural networks. We can determine which seal belongs to which famous person by using the recognition results, which allows us to learn more about this person's habits from their collection of information. Exploring this further will be the next target of our research.

In future work, we will demonstrate the supervised-learning-based seal detection method on other newly publicized datasets such as the NDL-DocL dataset [78] by using labeled data to optimize for dealing with poor results caused by input seal images in

similar colored characters. We will also investigate how to use a single typeface image better and try new methods based on hierarchical matching and reinforcement learning techniques to optimize the matching results.

Chapter 5 Ancient Character Recognition

5.1 Introduction

The Greek philosopher Aristotle explained things, concepts, and symbols in his ‘On Interpretation.’ He emphasized the writing system and the complicated relationship among things, concepts, languages, and cultures. There are different ancient characters in the world. For example, the ancient Egyptian hieroglyphs lost their meaning in the fourth century AD and became a mysterious writing system. Protecting existing records of language and writing systems has become the goal for humanities studies. If these scarce records can be archived, the system that provides retrieval or identification can be accessed publicly. More scholars can see clearer vectorized glyph resources, and more people will be able to taste these fading historical–cultural ambiances.

As described in Chapter 4, the seal script is one of the widely used ancient Chinese typefaces nowadays. However, most people today cannot read the seal script unless they have the expert knowledge to read the glyph form. Moreover, although a new dataset [79] has been released recently, the existing OCR system is very limited. As described in Section 2.3, the utilizations of the imbalanced dataset are still a challenge. In addition to the above dataset, font typeface resources may be a good way to protect the ancient character records; for example, many ancient character fonts are made strictly based on ancient characters or symbols that are no longer used [80][81][82]. However, some of the ancient characters are represented by historians and archived with just one sample for each character. Although many character-recognition techniques have been developed, while most of them have been created for a specific domain, and each category of characters has sufficient training data [83][84][85], determining how to use the few

sample resources effectively is still a challenge.

As mentioned in Chapter 1 and Chapter 4, the Shirakawa Shizuka Institute of East Asian Characters and Culture, Ritsumeikan University, has developed a search system for the ‘Shirakawa font’ [10] which can be used freely by converting kanji characters from the modern calligraphic style to the old calligraphic scripts (oracle bone script, bronze script, and seal script). The total number of oracle bone script characters recorded was 681. Each ancient character was made as a vector image with a smooth edge, and users can enlarge each search result to see the details. These font typeface images have also been bundled as font packages and released on the website. Our research considers the efficient use of font typeface image resources and implemented an oracle bone script offline handwriting recognition framework based on the oracle bone script in ‘Shirakawa font’.

In this chapter, we attempt to continue the preliminary experiment conducted in Chapter 4. This study provides a flexible perspective on the use of low-resource ancient script datasets and attempts to evaluate the performance of our proposed approach on existing datasets and tasks. The goal is to obtain features with better generalization capabilities from font typeface data and apply them to character retrieval tasks.

5.2 Related Work and State-of-the-Art

In the following sections, we discuss character-recognition techniques applied to ancient character datasets and compare their differences with the content-based image retrieval based on shape-matching.

5.2.1 Ancient character recognition

Digitization of ancient manuscripts and inscriptions is a very complicated and difficult process [86]. Many efforts have been made to overcome the lack of resources and the deformations caused by the handwritten script. As shown in Table 5.1, number of techniques have been proposed and applied to various ancient scripts. Several of them focus on character features in the object domain, extract some geometric features and obtain well results on the targeted dataset.

As can be seen from Table 5.1, and in contrast to previous studies on ancient character recognition, the main difficulty of our work is that the number of total characters to be classified is 681, and there is only a single reference sample for each of them. We aim at finding a general method that uses public and sufficient data resources from other domains to perform the same retrieval task for public ancient character scripts.

5.2.2 Sketch-based image retrieval based on shape-matching

Many content-based image retrieval approaches use different representations of images, such as color, object shape, texture, or a combination of them [89]. A shape is one of the most common and determinant low-level visual features, which contains an object's geometric structural characteristic. Sometimes, we could assume a character as a symbol with a shape.

Table 5.1 Related work on ancient character recognition

Script	Method	Task & Data availability
Devanagari script	Naive Bayes, RBF-SVM, and decision tree using HOG and DCT features [83]	33 classes and 5484 characters for training; dataset is unavailable
Tamil character	Fuzzy median filter for noise removal, a neural network including 3 layers [87]	Total class number is unknown; dataset is unavailable
Batak script	K-Nearest Neighbors [88]	Total class number is unknown; dataset is unavailable
Vattezhuthu character	Image Zoning [84]	237 classes and 5000 characters for training; dataset is unavailable
Odia numbers	LSTM [91]	10 classes and 5166 characters for training; dataset is unavailable

There are two main classes of sketch-based shape matching approaches: (1) the learning-free method and (2) the learning-based method. Like traditional image retrieval methods, learning-free methods focus on selecting representation, extracting global [90] or local [91] features, and accomplishing retrieval based on a nearest-neighbor search in a descriptor space. The learning-based method [92] usually requires sufficient annotated data. It tends to work better than learning-free methods but usually involves massive manual-annotation effort. Shape-matching can usually transform one shape and use some similarity metric to measure the similarity to another shape. The problem is how to evaluate the degree of similarity. Some feature descriptors do not pay much attention to gradient orientation in localized portions of an image, for example, features extracted by the VGG pre-trained model [93]. Moreover, some proposed methods do not control the limitation of rotation when augmenting the training data.

As shown in Figure 1, in Odia, letters a, b, and c are three different characters. In

cases where the gradient orientation in localized characters is ignored, the similarities of (a, c) and (a, b) might become closer. Hence, it is difficult to find a suitable space to represent the original data. When queries (see query 1, query 2, and query 3) arrive, it is challenging for some learning-free methods to obtain a better result by simply ranking based on similarity only. Our approach focuses on correcting the distortion and deformation of characters by considering the gradient-orientation feature and aims to find a better representation that can distinguish a character from other characters with similar structures. In Section 6, we compare the image features extracted by our pre-trained model and current existing pre-trained models, including the Vision Transformer [94].

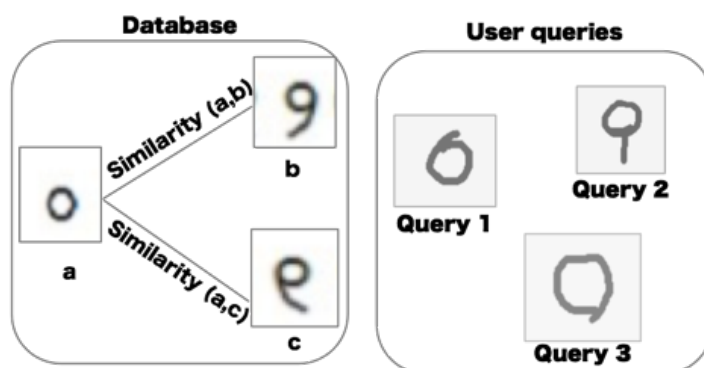


Figure 5.1 Similar characters in Odia number, a, b, and c are three different characters

5.2.3 Meta-learning and metric-based method

Meta-learning, also known as ‘learning to learn’, aims to learn models as a new method to adapt to new environments and solve several problems, such as when a classifier trained on one specific domain can tell whether a given image contains an object from another domain after seeing a handful of images. There are three common approaches to meta-learning: metric-based, model-based, and optimization-based.

Although there are many cross-domain meta-learning methods for image classification, it should be clear that the state-of-the-art will depend on the type of data.

Considering the geometric characteristics and the lack of color information in character images, the methods that had results reporting character classification are selected for comparison.

For metric-learning-based methods, RelationNet [95], MatchingNet [102], and Siamese network [103] pay more attention to learning how to compare the relationship between a pair of images, which can lead to similar problems in shape-matching. It is better to learn to embed characters with similar features to the closer position in the feature space. A typical example might be prototypical networks (ProtoNet) [104], which aim to learn an embedding function to encode each input into a feature vector. Our proposed method optimizes the performance of prototypical networks to extract geometric features and obtains better results on the benchmark datasets.

For model-based meta-learning methods, Finn et al. [105] provided a meta-learning-based method, MAML (Model-Agnostic Meta-Learning), that can be applied to cross-domain character recognition and obtain good performance. To improve the cumbersome, unstable Bayesian framework, Deep Kernel Transfer (DKT) was proposed by Patacchiola et al. [106]. The Gaussian process approach is used by learning a deep kernel across tasks, and it has maintained the best record of five-way (one-shot) OMNIGLOT→EMNIST cross-domain characters recognition.

Publicly available benchmark datasets and implementation of DKT and MAML provided from Patacchiola et al. [106] are used as state-of-the-art baseline methods for comparison experiments of OMNIGLOT→EMNIST cross-domain character recognition tasks; MatchingNet [102], ProtoNet [104], and RelationNet [95] are selected as basic metric-learning-based methods to compare the performance of feature extraction; the results are introduced in Section 5.5.

5.3 Main Contributions of This Study

In this study, we tried data augmentation based on the generative adversarial network (GAN) model and conducted experiments using generated data as training data. According to the experimental results, we choose the adjustment method and propose a character recognition method based on metric-learning.

The main contributions of this study can be considered as follows:

(1) From a technical aspect, we present a new model structure based on metric learning to use a low-resource ancient character typeface dataset. It is a new attempt to apply generic handcrafted features combined with few-shot metric learning to the model, which works in low-resource data. Thus, the proposed method can obtain more low-latitude features that are conducive to the retrieval of symbols and ancient characters. Comparing with other existing metric-learning based methods, the features extracted from our proposed method have better advantages in finding the highest-ranked relevant item.

(2) Training on other sufficient public character datasets such as OMIGLOT and testing on font typeface-based datasets. This method can learn to represent the typeface images into a vector space more appropriate to their geometric properties. The method proposed in this paper also performs better than several other metric-learning based few-shot learning methods on the cross-domain task using the benchmark datasets.

(3) The calculation of the dynamic mean vector is imported to enhance the robustness of prototypical networks. In the mini-batch training process, we not only select a unique prototypical center but also adapt to the deformation of the support set from test data.

(4) Our proposed framework consists of several components, i.e., spatial

transformer network, feature fusion, dynamic prototype distance calculator, and ensemble-learning-based classifier. Each component has reasonable contributions to the classification task.

(5) We apply the proposed method to retrieving ancient characters that support handwritten input, providing a new perspective for the flexible use of low-resource data. This is an innovative effort in terms of one-shot-based ancient character recognition by utilizing the metric-learning method. It provides a reference for the application of ancient font typeface resources in digital humanities and other fields in the future.

5.4 Methods

This section introduces our attempt to use font typeface to build text recognition applications. Section 5.4.1 introduces data-augmentation-based method, which is a preliminary attempt after referring to common methods to solve the problem of lack of data. Section 5.4.2 introduces the proposed method based on metric-learning, which is a deep learning model based on one-shot data utilization.

5.4.1 Data-augmentation-based method

One of the commonly used methods to solve the problem of the lack of data is data augmentation. Ghosh et al. [96] conducted experiments to generate a dataset of MNIST handwritten digits, and the results showed the effectiveness of GAN in the character recognition area. Creswell et al. [97] conducted research on applying data extensions to Merchant Marks images using DCGAN, and used these data to provide search support for the Merchant Marks database. Tran et al. [98] used GAN to generate images of humans with shooting angle conversion and to provide data generation support in human

face recognition.

In this study, the 2,590 characters of the “Shirakawa font” are used and converted into an image format. These seal script images are used as the generation target images of the generation model. In the "Master Ideographs Seeker for CNS 11643 Chinese Standard Interchange Code" [99] in Taiwan, there are 6,721 characters created in Unicode based on the oldest radical-specific Chinese character dictionary "Sentence Kanji". The open source font “Uniform Kanji True Type Character Type” [99] including the above is released. Since it can be used free of charge for research purposes, this seal script font is used, converted to an image format, and input to the model as a source image.

There was some consistency in the structure of the images generated from the zi2zi model [100] shown in Figure 5.2, and relatively high consistency was found in attributes such as stroke spacing for the same characters.

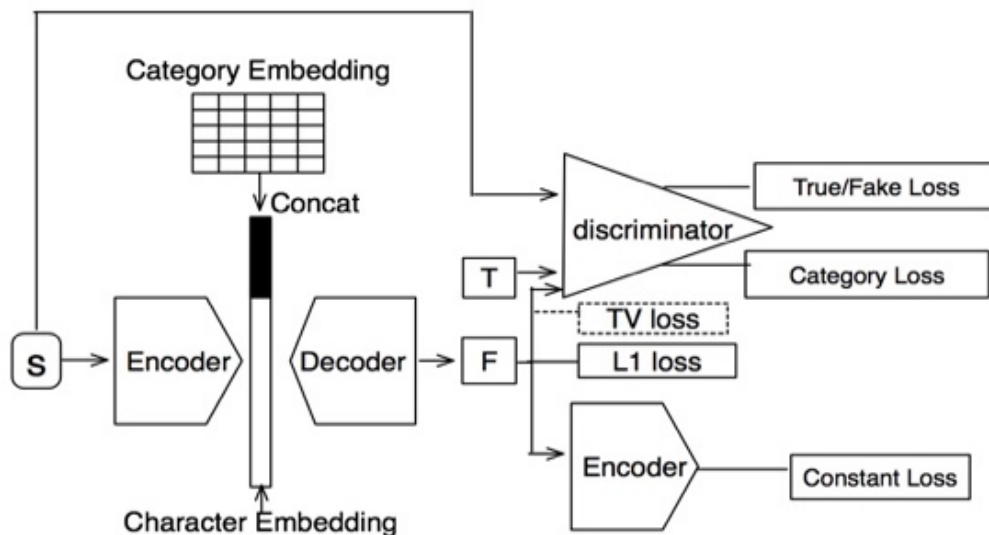


Figure 5.2 GAN-based zi2zi model

An example of training data pair is shown in Figure 5.3. The left side is the target image generated from the Shirakawa font, and the right side is the source image extracted

from the "Uniform Kanji". The purpose of optimizing the model is to generate new style characters by learning character styles.

In addition, in this study, by adding some random image processing, the morphological structure is changed, the effects of the generated images are combined, and an image close to the characteristics of the handwritten text is generated, and then it is used as training data.



Figure 5.3 An example of training data pair



Figure 5.4 Training data with morphological transform

Some of the generated characters are shown in Figure 5.4.



Figure 5.5 Generated training samples

In the case of actual historical document, in order to obtain a binarized image that is usually severely damaged and has relatively little noise, the original image is often subjected to multi-distance imaging, smoothing, binarization, expansion and contraction processing, etc. Based on the generated images, we further add random noise to enhance the robustness of training. Also, we applied random processing using Gaussian blur technology to change the difference in pixel quality of the image. The output images are shown in Figure 5.5. We used the most basic LeNet to test the basic performance of augmented data for classification tasks. Section 5.5 introduces our experimental results.

5.4.2 Metric-learning-based method

In this section we introduce the metric-learning-based approach. Figure 5.6 shows the structure of our ancient-character-recognition framework.

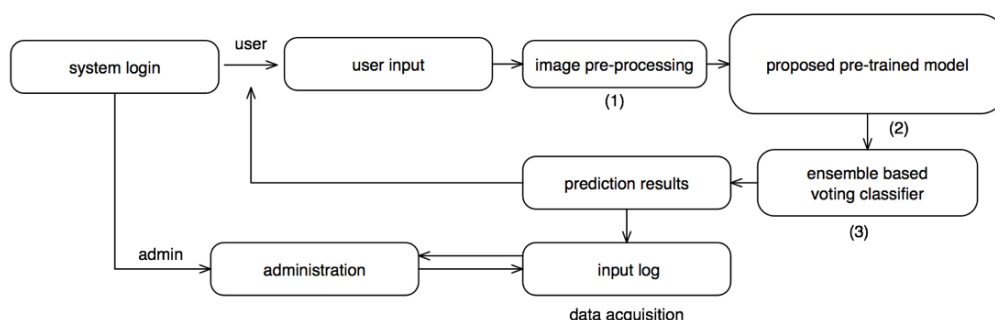


Figure 5.6 Our proposed character-recognition framework

Considering the difference in character position and stroke width of typeface images, we process the training data and user input as shown in Figure 5.7.

As shown on the left in Figure 5.7, we use a horizontal histogram projection to obtain the position of the characters. The idea is to add up the columns or rows of the image to obtain a projection whose minimum value allows us to segment the characters.

The architecture of our proposed recognition model is shown in Figure 5.8.

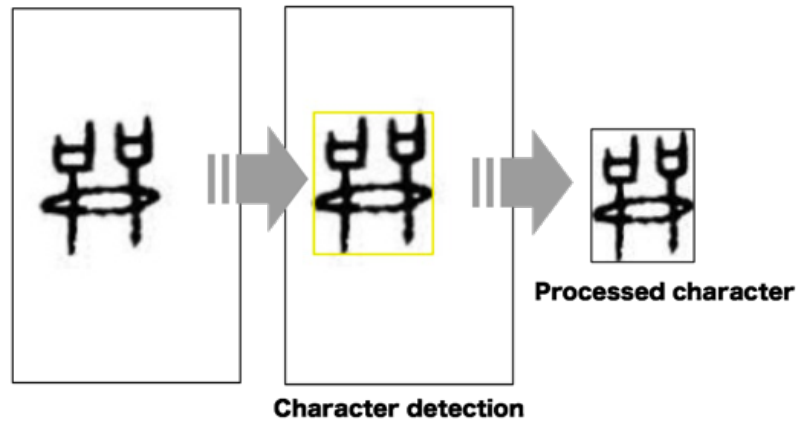


Figure 5.7 Typeface image pre-processing

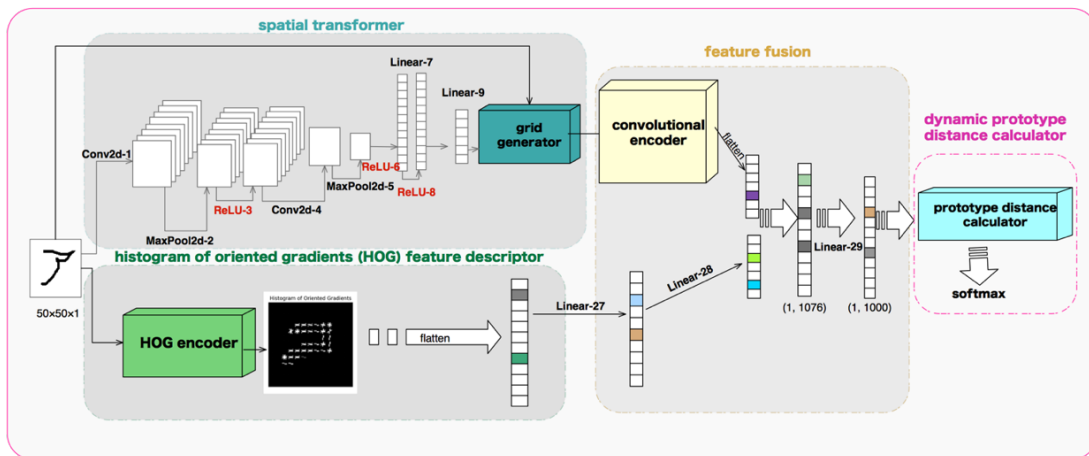


Figure 5.8 An overview of our proposed model

It consists of four modules: (1) spatial transformer module, (2) histogram of oriented gradients (HOG) feature descriptor module, (3) feature fusion module, and (4) dynamic prototype distance calculator module.

Affine transformation helps to modify the geometric structure of an image, and it preserves collinearity and ratios of distances. It can be used in machine learning and deep learning for image processing and image augmentation [108]. It is also used to correct

geometric distortions and deformations and is useful in image processing of satellite images [109]. There is a study that performs handwritten symbol classification in the presence of distortions modeled by affine transformations [110]. To solve geometric distortions and deformations problems, we utilize the spatial transformer, which is conceived in spatial transformer networks [111], in our method. The spatial transformer can reduce the influence caused by translation, scale, rotation, and more generic warping and has a better performance in handwritten character classification. It consists of the localization net and grid generator. The localization net takes the input feature map $U_{input} \in \mathbb{R}^{H \times W \times C}$, where H , W , and C represent the width, height, and channels of input images, respectively. The output of the localization net is represented by θ and calculated by $f_{loc}(U)$. The function $f_{loc}()$ can take any form, but for affine transformation, θ needs to be 6-dimensional. Hence, we designed the function $f_{loc}()$ as shown in Figure 5. From the output, we can obtain transformation parameter θ as the input of the grid generator.

The grid generator creates a sampling grid by using the calculated transformation parameters. The output of the parameterized sampling grid is represented by $V_output \in \mathbb{R}^{(H' \times W' \times C)}$, where H' and W' are the same as H and W of the input feature map. We utilized the output of spatial transformers as one part of the input of our proposed feature fusion module.

The HOG is an efficient way to extract the gradient-orientation feature in localized portions of an image. The distribution of directions of gradients is used as features to represent an input image. The character structure has a strong orientation feature; as shown in Figure 5.10, ten completely different characters are randomly selected from OMNIGLOT dataset. Though many pairs are only slightly different from each other, they can be distinguished effectively by the feature extraction of the HOG extractor.

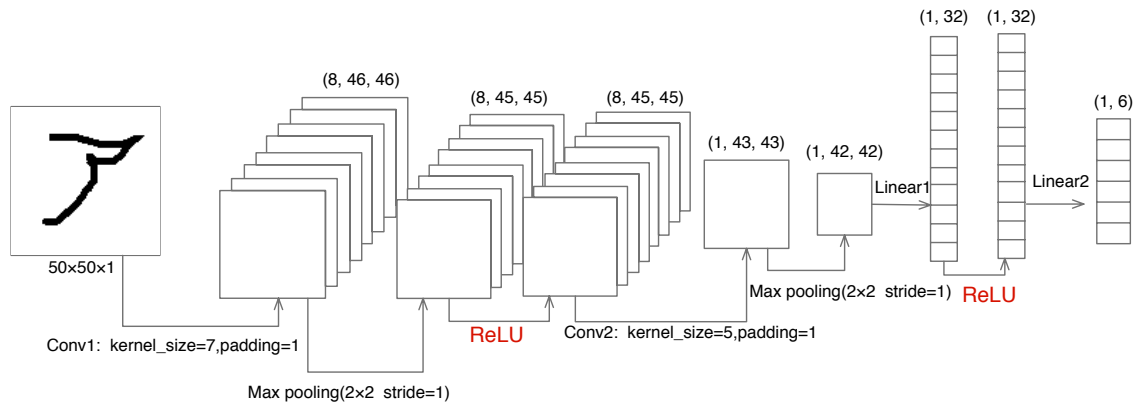


Figure 5.9 Architecture design of the function $f_{loc}()$ in our method



Figure 5.10 Ten completely different characters randomly selected from OMNIGLOT

Additionally, a study [112] showed that handwritten character recognition performance could be improved by using the HOG feature as an input of a neural network. Therefore, in our work, feature extraction based on the HOG feature descriptor was applied to extract features from ancient characters.

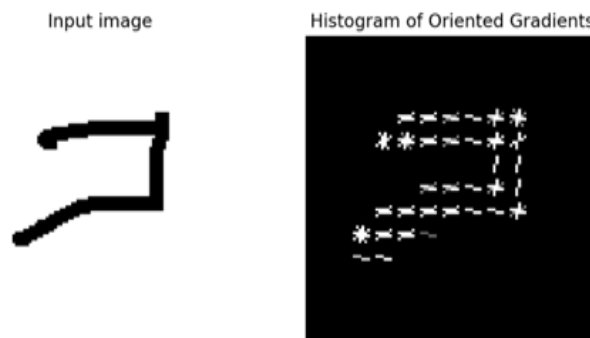


Figure 5.11 An example of HOG feature map

Figure 5.11 shows an example of a feature map we obtained. The input images were resized to (50×50) , we set the number of orientation bins as 8, the size of a cell as $(6, 6)$, and number of cells in each block as $(2, 2)$. We used a flattened HOG feature vector as the input to the ‘Linear-27’ calculation, as shown in Figure 5.8.

A naive approach to combine multiple features is to concatenate the feature sets together. However, the vector may consume significant space and cannot show the combined characteristics of the object. Many studies [113][114] have learned the common representation of data from different domains by using feature-fusion technology. We use a feature fusion module that adaptively weighs and combines these representations based on local features extracted from the HOG feature descriptor module and features extracted from the convolutional neural network architecture. The structure of the convolutional encoder is shown in Appendix A.

As shown in Figure 5.12, prototypical networks [104] learn a metric space and require only a small amount of training data with limited information.

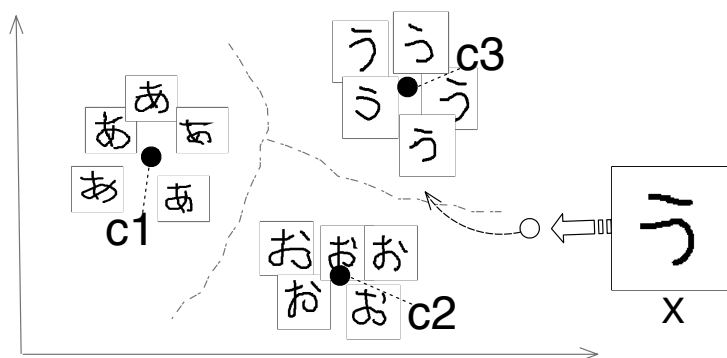


Figure 5.12 An example of prototypical networks classification in a few-shot case

It represents each class by the mean of its examples in a representation space. The predicted probabilities of a given test input X could be calculated by (5.1):

$$P(y = c|X) = \frac{\exp(-d_\varphi(f_\theta(X), V_c))}{\sum_{c' \in C} \exp(-d_\varphi(f_\theta(X), V_{c'}))} \quad (5.1)$$

where f_θ shows the embedding function, d_φ is the distance function, and V_c represents the mean vector of the embedded support data samples in this class, which is calculated by (5.2):

$$V_c = \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} f_\theta(x_i) \quad (5.2)$$

where S is defined as a set of embedded support data samples. The loss function is defined as (5.3) in the training stage:

$$\mathcal{L}(\theta) = -\log P_\theta(y = c|x) \quad (5.3)$$

In our method, we define d_φ as Euclidean distance calculation and predict the input data as (5.4):

$$P(y = c|X') = \text{softmax} \left(-\frac{\sum_{i \in \{0,1,\dots,z\}} d_\varphi(f_\theta(X'), V_{ci})}{z} \right) \quad (5.4)$$

where V_{ci} in our training step is defined as a dynamic mean vector of a subset extracted by random sampling from the support set. The dynamic prototype distance calculator is used to calculate the loss value, and the algorithm to compute the loss $\mathcal{L}(\phi)$ for a training episode is provided in Algorithm 2.

Algorithm 2 The algorithm to compute the loss $\mathcal{L}(\phi)$ in our network. Training set contains the support set S and query set Q. K is defined as the number of classes per episode. S_q is the number of queries of each class in Q. $\text{RANDOMSAMPLE}(S, n)$ denotes a set of n elements chosen uniformly at random from the set S, without replacement. f_θ is the embedding function, and d_ϕ is the distance function.

Input:

S: support set $\{(x_{k0}, y_{k0}), (x_{k1}, y_{k1}) \dots (x_{kn}, y_{kn}), \}$ from random selection from training set T, $k \in \{1, 2, 3, 4 \dots K\}$, n is the number of support examples per class

Q: query set $\{(x'_{k0}, y'_{k0}), (x'_{k1}, y'_{k1}) \dots (x'_{kz}, y'_{kz}), \}$ from random selection from training set T, there is no repetition with S, z is the number of query examples per class

Output: The loss $\mathcal{L}(\phi)$ for each training episode

Compute the length of each class in S, $Len_S = \{len_{s1}, len_{s2}, len_{s3} \dots len_{sk}\}$

For each class k in S generate a random integer $N_{sk} \in \{N_{s1}, N_{s2} \dots N_{sk}\}$ such that $0 \leq N_{sk} \leq Len_{sk}$,

For k in $\{1, 2, 3 \dots K\}$ do

For (x'_{ki}, y'_{ki}) in $\{(x'_{k0}, y'_{k0}), (x'_{k1}, y'_{k1}) \dots (x'_{kz}, y'_{kz}), \} = Q$, $i \in \{0, 1, \dots, z\}$ do:

$S_{ki} \leftarrow \text{RANDOMSAMPLE}(len_{sk}, n)$

$c_{ki} \leftarrow \frac{1}{N_{sk}} \sum_{(x_{kj}, y_{kj}) \in S_{ki}} f_\theta(x_{kj}), \quad j \in \{0, 1 \dots N_{sk}\}$

$d_{ki} \leftarrow d_\phi(x'_{ki}, c_{ki})$

end for

$$\mathcal{L}(\phi) \leftarrow \mathcal{L}(\phi) + \frac{1}{K} \left[\frac{\sum_{i \in \{0, 1, \dots, z\}} d_{ki}}{z} \log \left(\sum_{k'} \exp \left(-\frac{\sum_{i \in \{0, 1, \dots, z\}} d_{ki}}{z} \right) \right) \right]$$

end for

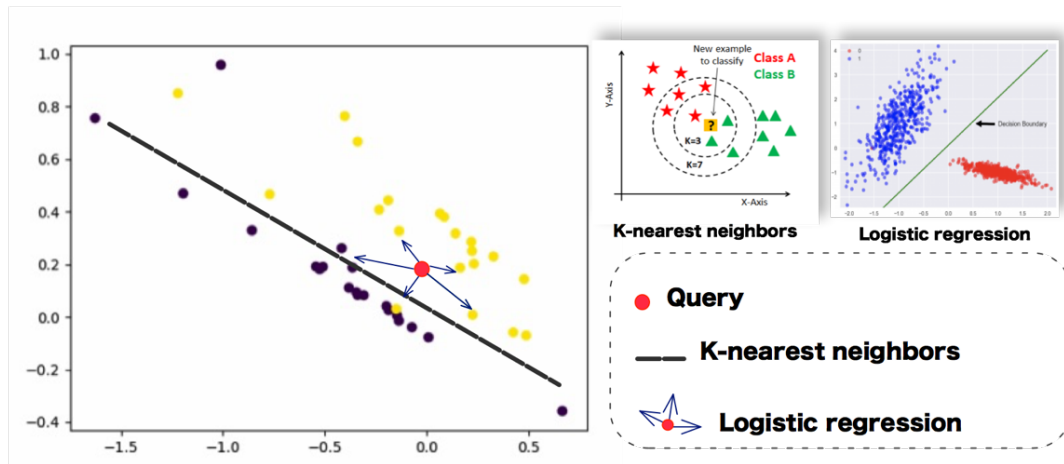


Figure 5.13 A voting classifier

As shown in Figure 5.13, to improve the prediction results, we use an ensemble approach to the voting classifier. As the prediction domain was not fine-tuned, we used a K-nearest neighbor classification method based on the local distribution of the data and a logistic regression method based on the overall distribution of the data to predict queries. We describe the results by using a voting classifier combined with K-nearest neighbor and logistic regression in Section 5.5.

5.5 Experiments and Results

In this section we introduce evaluation experiments based on data augmentation and metric-learning methods.

5.5.1 Evaluation on data augmentation-based method

We used the generated data as a training model for training data and conducted a character recognition experiment for handwritten data. 1,000 characters of "Shirakawa font" were randomly selected, and 300 characters were randomly selected from them, and they were used for 300 kinds of classification learning. The number of training data for each

classification learning is about 300 to 500 images.

Table 5.2 Experimental results on data augmentation-based method

Methods	Accuracy (%)
Without pre-processing	60.00
Affine transformation	73.33
Addition of background	63.60
Affine transformation + addition of background	77.00

Table 5.2 shows the experimental results. From the experimental results, it can be seen that affine transformation plays an important role in data augmentation.

5.5.2 Evaluation on data metric learning-based method

Since there is only one sample for each character in the Shirakawa font typeface dataset, a robust system framework is expected to extract features from one-shot character samples and apply these features to retrieval tasks, and there must be enough test data to evaluate the performance of the feature extraction. Hence, we evaluated our proposed method using the standard benchmark handwritten character dataset OMNIGLOT [115] for the training task, which included 1623 characters from different languages, not including oracle bone characters, and 20 samples for each character. For the test task, we used EMNIST [116], which included 47 characters based on a mix of letters and digits and 2400 samples for each character. We use these two datasets on a regular five-way (five-shot), cross-domain, few-shot learning task. We used model weights trained on OMNIGLOT to extract the features from the Shirakawa oracle bone font typeface [10] to evaluate the performance of the proposed model. We manually collected 40 queries from the website [117], used an automatic segmentation algorithm to extract all oracle bone

characters on the OMNIGLOT webpage, and labeled all the segmented characters according to the modern character annotations of the OMNIGLOT webpage. We performed feature extraction and retrieval experiments on 681 oracle bone characters using the pre-trained model. To show the scalability in different font retrieval, two font families, MERO_HIE hieroglyphics font [81] and Aboriginebats font [82], were used to display the vector visualization and the example of the retrieval results. We used Adamax [118] as the optimizer, and the learning rate was set to 0.001.

To test the ability of feature extraction learned by the system on unseen input data, we evaluated the proposed method on the ‘OMNIGLOT→EMNIST’ task, which means training on OMNIGLOT, extracting features on EMNIST for classification testing, and not performing fine-tuning on EMNIST. Each epoch randomly selects 5 categories of data from OMIGLOT, and each category includes 1 support sample and 15 queries. Table 5.3 shows the losses on the training domain and the target domain under different batch size settings. The results show that a model with higher accuracy in the target domain can be obtained under a smaller batch size setting. However, in this case, the classification performance of the trained model in the target domain is unstable.

Table 5.3 The loss value in training domain and target domain at 1000 epoch. The orange line represents the training domain and the blue line represents the target domain, the vertical axis represents the loss value, and the horizontal axis represents the number of epochs.

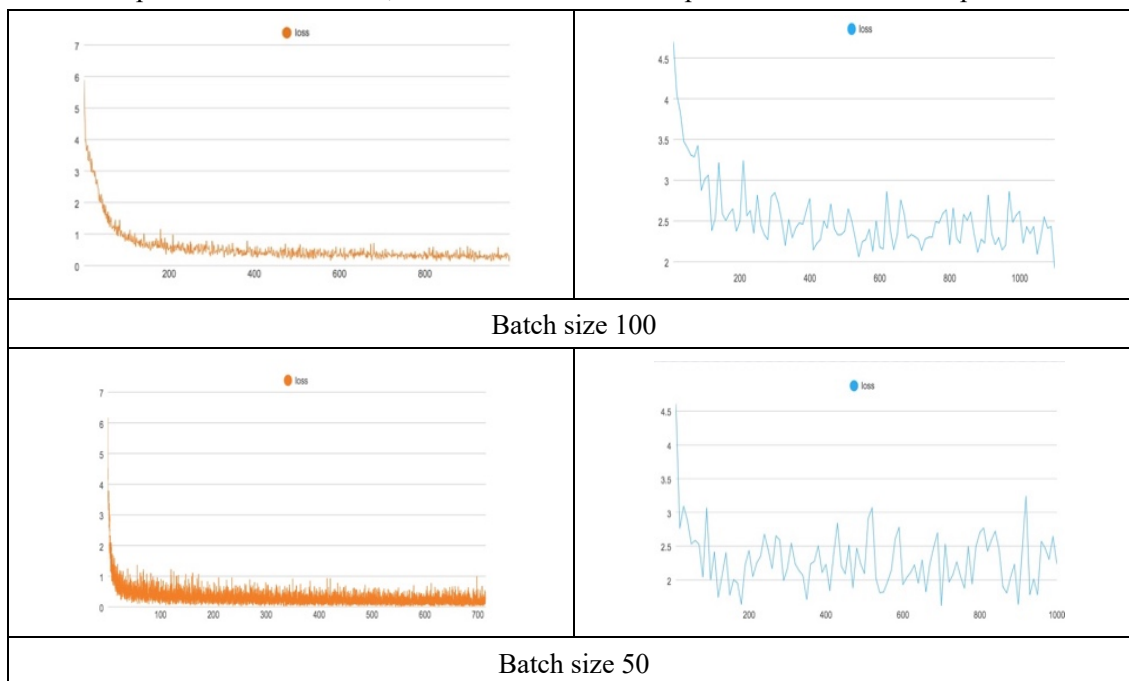


Table 5.4 shows the comparative experimental results of the five-way (one-shot) classification task between the primary metric learning baseline methods and our method. The results show that our proposed method has learned better representation from characters than ProtoNet [104] in five-way (one-shot) classification task. It has a good performance in the best sampling situation, but compared to MatchingNet [102] and RelationNet [95], the average classification accuracy of the five-way task is lower.

Table 5.4 Comparison on five-way (one-shot) classification task

Method	Average
MatchingNet (Vinyals et al., 2016) [102]	75.01 \pm 2.09
ProtoNet (Snell et al., 2017) [104]	72.77 \pm 0.24
RelationNet (Sung et al., 2018) [95]	75.62 \pm 2.00
Our proposed method	74.00 \pm 6.00

To evaluate the performance of the classifier used on the target domain and compare the performance of the ensemble-learning-based classification method, we tested the classification ability of each classifier under different settings of sample numbers in the support set and different feature dimensions. The results are shown in Table 5.5.

Table 5.5 Classification ability comparisons on each classifier under different settings of sample numbers in the support set and different feature dimensions

Methods	Number of Samples in Support Set	Average Score		
		50 Dimensions	500 Dimensions	1000 Dimensions
K-nearest neighbors	one shot	0.74	0.5	0.66
	five shots	0.78	0.48	0.8
	ten shots	0.80	0.84	0.82
Logistic regression	one shot	0.64	0.48	0.68
	five shots	0.78	0.74	0.92
	ten shots	0.94	0.72	0.90
Ensemble learning (proposed method)	one shot	0.68	0.64	0.74
	five shots	0.76	0.72	0.83
	ten shots	0.88	0.82	0.86

The experimental results indicate that logistic regression has a good classification ability in the five-shot and ten-shot classification tasks with more samples in the support set. Compared to the one-shot classification task with fewer samples in the support set, using low-dimensional features in 50 dimensions, K-nearest neighbors have robust distance calculation and classification capabilities. Still, our proposed ensemble learning method has a better performance in the case of increased feature dimensions and has the best performance on the one-shot task.

Our test target domain EMNIST has more than 40 characters; thus, we set the number of categories trained on OMNIGLOT to 40. We tried the forty-way (five-shot)

training task; 40 categories of characters were sampled for training on the training domain for each epoch, and each category had 15 queries for training.

Table 5.6 shows the performance of the features obtained from the trained model utilized in the classification task with different category settings of the target domain. For each category in the target domain, we use only one-shot support data; thus, we regard it as a retrieval task and use precision at 1, 5, and 10 ($P@1$, $P@5$, $P@10$) to evaluate the performance.

It can be seen from the results that the accuracy of the top one decreases after expanding the number of classification categories, but the obtained features are still suitable for retrieval tasks that show 10 results as the results in $P@10$.

Table 5.6 The performance of the retrieval task that uses the features extracted from the model trained on forty-way (five-shot) task and 1000 epochs

Number of Classification Categories	$P@1$	$P@5$	$P@10$
20	0.55	0.85	0.9
30	0.3	0.83	0.9
47*	0.28	0.65	0.76

* On the test domain, we used all the 47 categories of data from EMNIST.

Since the expected target ancient character font typefaces usually include more than 40 categories, we conducted ablation experiments of our proposed model on the forty-way training task.

Skeletonization was used as one of the comparison targets of the ablation experiments. Skeletonization is a process of reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels. Skeleton maps, as shown in Figure 5.14, provide an efficient shape descriptor in many applications,

such as content-based image-retrieval systems and character-recognition systems. We used the skeletonization method of Zhang et al. [70] and set a comparative experiment with the original image for training.

It can be seen from Table 5.7 that the spatial transformer plays a significant role in our model. If the model does not use the dynamic prototype distance calculator, the accuracy will be affected, even lower than if the HOG encoder module is not used. As shown in the results, for current data, the skeletonization method does not improve the results.

The visualization of the training process introduced in this section can be found in Appendix B, Figure A.1, and Table A.2.

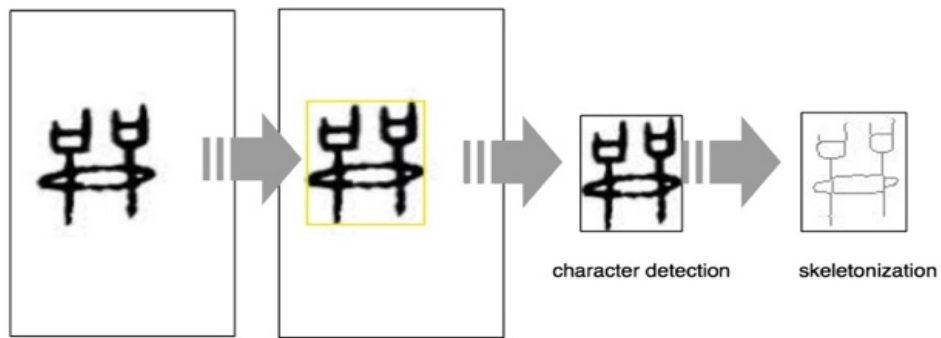


Figure 5.14 Skeletonization process

Table 5.7 Ablation experiments of our proposed model

No.	HOG encoder	Spatial transformer	Dynamic	Skeletonization	P@1	P@5	P@10
1	✓	✗	✓	✗	0.19	0.43	0.63
2	✗	✓	✓	✗	0.24	0.61	0.73
3	✓	✓	✗	✗	0.17	0.52	0.6
4	✓	✓	✓	✗	0.28	0.65	0.76
5	✓	✓	✓	✓	0.20	0.26	0.41

In actual use, there is only one image for each character of each font resource. We

compared the performance of the features extracted from the existing pre-training models and our proposed model in the retrieval task.

We prepared 40 oracle bone characters obtained from the OMNIGLOT website [117] that eliminated non-corresponding data, which means characters completely different from those recorded in the Shirakawa font. Examples of non-corresponding data are shown in Appendix C and Table A3.

The search range is 681 oracle bone characters from Shirakawa font [10]. We chose the VGG19 [62] and ResNet50 [119] models that are most commonly used in image-retrieval studies and the transformer-based model ViT [94] for comparison. Since we used a total of 681 oracle characters from the Shirakawa font, the index for retrieval was increased to 681.

We used the evaluation metric MRR (Mean Reciprocal Rank) to evaluate the system. Each query has only one corresponding correct result. We set the experiments using different models to extract features and applied them to retrieval tasks to prove the performance of our method of feature extraction.

Table 5.8 Performance evaluation of pre-training features on retrieval tasks

Model	Feature Dimension	MRR Score
VGG19 [62]	1000	0.0580
ResNet50 [119]	1000	0.0563
ViT [94]	1000	0.0586
Our proposed method	1000	0.1943

Table 5.8 shows that since our model is not trained on a dataset like ImageNet, the results shown in Table 5.8 indicate that the features extracted by our model are more suitable for use in character retrieval without color information. It can be seen from the

results that our system has a better performance in finding the highest-ranked relevant item.

Our method can also be generalized to other font datasets. Table 5.9 shows the vector visualization of our pre-trained model applied in MERO_HIE hieroglyphics font and Aboriginebats font datasets [81][82].

A simple stroke of a bird is used as a query. It can be seen that the features extracted by our model are more sensitive in shape-matching, and better results can be obtained. In the visualization figure, the red box shows the most similar image based on cosine similarity calculation, and the yellow box represents the top five search results.

Table 5.9 Application performance on MERO_HIE hieroglyphics font and Aboriginebats font datasets

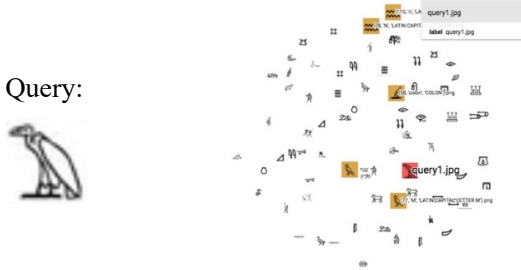
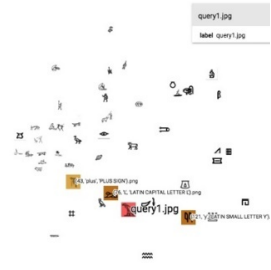


Models:	Our proposed method	Pre-Trained ResNet50
Query:		
Dimensions:	500	1000
Top retrieval result:		

Table 5.10 Results for OMNIGLOT→EMNIST five-shot classification

Method	One-Shot		Five-Shot	
	Reported	Our Re-tested Results	Reported	Our Re-tested Results
MAML (Finn et al., 2018)	72.04 ± 0.83	71.12 ± 0.84	88.24 ± 0.56	88.80 ± 0.24
DKT + BNCosSim (Patacchiola et al., 2020)	75.40 ± 1.10	74.90 ± 0.71	90.30 ± 0.49	90.11 ± 0.20
DKT + CosSim (Patacchiola et al., 2020)	73.06 ± 2.36	76.00 ± 0.42	88.10 ± 0.78	89.31 ± 0.19
Our proposed method	(-)	73.33 ± 2.67	(-)	83.31 ± 0.87

Since the data used in our utilization has only one sample per category, comparison with state-of-the-art methods is based on the same ‘OMNIGLOT→EMNIST’ task. The current state-of-the-art methods based on meta-learning in the same ‘OMNIGLOT→EMNIST’ task were selected as the DKT [106] and MAML [105] to conduct a comparative experiment.

Because there is a difference in the total number of categories of the EMNIST dataset compared to the DKT experimental dataset setting [107] (Patacchiola et al., 2020), we

used the implementation contribution and datasets provided by Patacchiola et al. [106] to conduct comparative experiments.

The results on five-way one-shot and five-shot tasks are shown in Table 5.10. Compared with the test data we used in Table 5.4, the results are improved. On the one-shot task, the result from our proposed method is unstable but can obtain better maximum accuracy. The DKT proposed by Patacchiola et al. has a good performance on five-way classification tasks.

MAML and DKT do not support feature extraction for retrieval. Metric-learning-based methods MatchingNet, ProtoNet, and RelationNet were used for comparison as feature extractors. We retrain these three architectures, and the ‘Conv4’ mentioned in [106] is used as the backbone network training and feature extraction. The results are shown in Table 5.11. From the results, it can be seen that the features extracted from our proposed method have better advantages in finding the highest-ranked relevant item.

Table 5.11 Performance comparative evaluation of pre-training features on retrieval tasks

Model	Feature Dimension	MRR Score
RelationNet (Sung et al., 2018)	1600	0.1022
MatchingNet (Vinyals et al., 2016)	1600	0.0605
ProtoNet (Snell et al., 2017)	1600	0.0817
Our proposed method	1000	0.1943

A demo application is implemented for character recognition based on our proposed method. Figure 5.15 shows the user interface of the application implementation.

Each stroke sketched by a user is recorded by the system, and the search results are updated when the user completes the input of a stroke. Each user input stroke is converted into an image for a prediction. Each prediction does not rely on trajectory information of

the user input. Because most of the oracle bone characters correspond to a modern kanji character, the retrieval results are shown as pairs of ancient characters and their corresponding modern characters. Ancient characters are not represented by images, but they are represented directly by Shirakawa font.

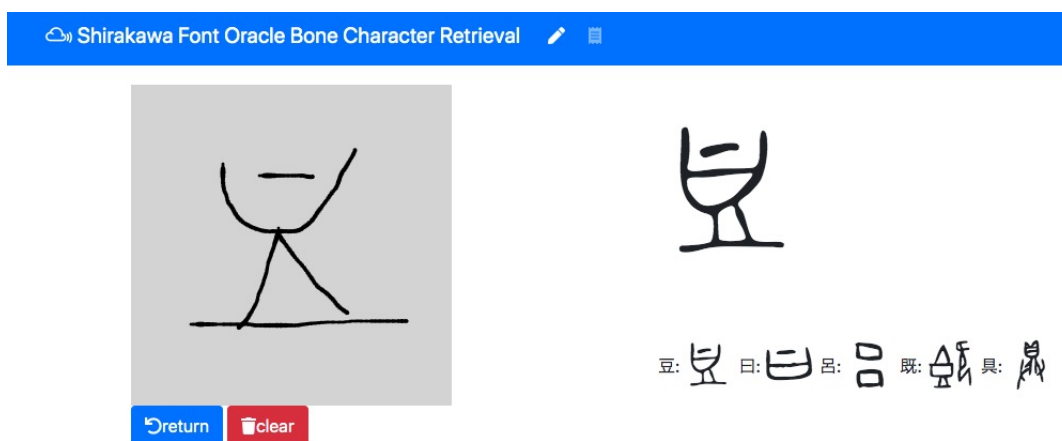


Figure 5.15 Demo application implementation

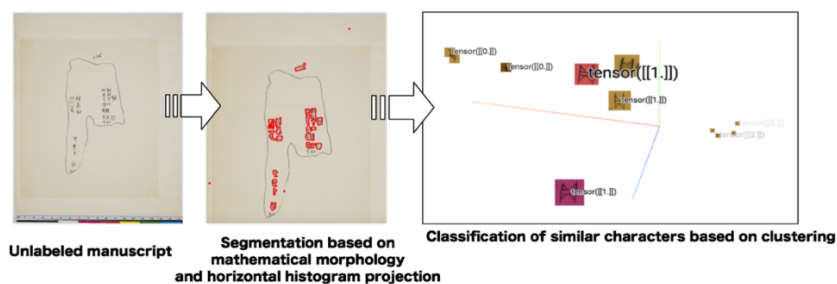


Figure 5.16 Utilization of the proposed method for recognition of oracle bone characters from historical documents

Our method can also be applied to unlabeled valuable historical documents. Figure 5.16 shows a case of utilization of the proposed method for sorting out unlabeled manuscripts [120] of oracle bone script. The features extracted from our model are used for unlabeled image clustering, and the visualization of actual results is shown in Figure

5.16.

From the results of experiments and analysis, it can be seen that the existing deep-learning models trained on large-scale data have limitations in extracting features from images with solid geometric characteristics.

In our previous work, we conducted a comparative experiment on the features of different layers extracted from the VGG19 pre-trained model. The goal of this study was to utilize the one-shot font typeface images to achieve a good retrieval performance. We found that different from the retrieval of artwork images [93], typeface images with weaker color and texture features have some issues that need to be solved, such as dealing with geometric deformation and detailed geometric features.

Comparative details are given in Table 5.12. Compared with the existing approaches, our work that uses only one reference sample for each category to retrieve more than 500 characters is a novel attempt. However, solving the problem of different character variation problems mentioned in Appendix C is a challenge for future studies and attempts.

Table 5.12 Comparison between this research and other exiting methods in different tasks

Task	Our Proposed Method	Model-Based Meta-Learning Methods	Metric-Learning-Based Methods	CNN Based and Transformer Based Pre-Trained Models
Five-way classification tasks on benchmark dataset	The performance is not as good as the model-based meta-learning method, which is relatively unstable, but has relatively good best results.	Bayesian framework Deep Kernel Transfer (DKT) proposed by Patacchiola et al. [106] has the best results but consumes more training resources.	RelationNet (Sung et al., 2018) has good results, but it is not as good as the model-based meta-learning method on the cross-domain character image classification task.	It is not a common method of few-shot learning and has not been evaluated by this research.
Used as feature-extractor for character images	Due to the focus on geometric feature processing and feature extraction, our model has a better performance on character data when used as a feature-extractor.	Due to insufficient descriptions and cases for feature extraction, this research did not conduct evaluation experiments using such methods.	Due to the use of pair images for learning, methods such as RelationNet and MatchingNet are more suitable for use as feature-extractor for the identification of the authenticity of handwritten characters. ProtoNet (Snell et al., 2017) is more suitable for use as a feature-extractor for retrieval, but if it is character data, some optimizations that focus on the use of geometric features are worth recommending.	Used large-scale datasets such as ImageNet for training, which is very effective for real-life image feature extraction with texture and color features. However, when extracting character data with obvious geometric features, performance needs to be improved.

5.6 Summary

In this chapter, we proposed a feature fusion spatial transformer structure combined with a prototypical network, which focuses on achieving font-typeface-based ancient character script handwriting recognition and correcting the distortion and deformation of handwritten characters to improve the accuracy of recognition in low-resource datasets.

The proposed method could be useful to researchers looking for new perspectives for taking advantage of scarce ancient characters or symbol typeface records.

Figure 5.17 shows that our method can also be applied to approximate query matching for retrieval-based character recognition, and it is very convenient to combine with other model architectures. In future work, we will focus on reducing the unstable problems of our proposed model. Additionally, we will aim to improve the generalization of the features extracted by our model while improving the prediction accuracy of the classifier.

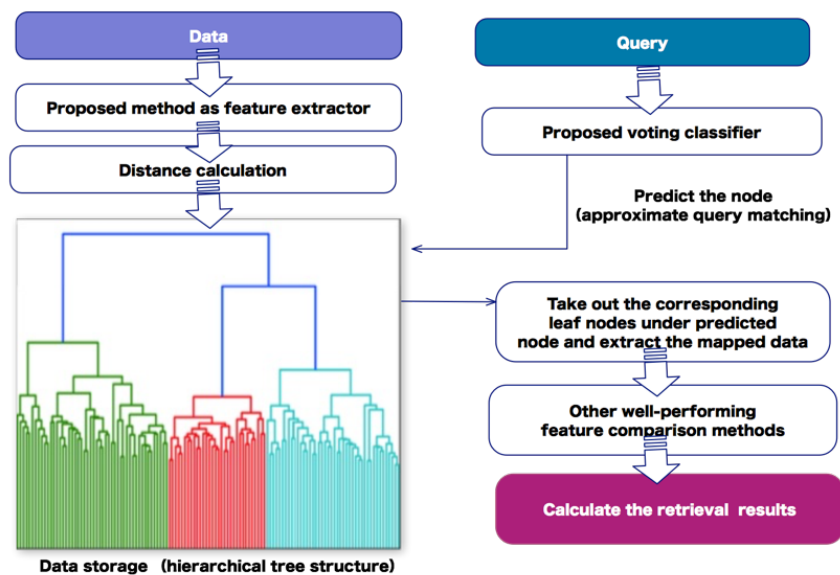


Figure 5.17 The proposed method applied to approximate query matching for retrieval-based character recognition

Chapter 6 Cross-Modal Retrieval for Japanese Ukiyo-e Prints

6.1 Introduction

The classification methods of colors are complicated. According to physical characteristics or according to luminosity and brightness, different classification methods have different color characteristics. These color space models are based on different color systems. They are mainly based on one-to-one color information description, such as assigning a specific value or vector to a particular color. Using only a simple case, it may be challenging to find images corresponding to similar colors based on human senses by simply selecting RGB or HEX values. For example, if colors are ordered by hue (rounded to nearest 30) and luminance (rounded to nearest 20), it can be seen that the color group in the range of $(H, L) = (0,40)$ tends to be pink, as shown in Figure 6.1.

It can be seen from the figure that most of the colors in this range are close to the pink of the human senses, but the colors in the red boxes may tend to be described more as gray.

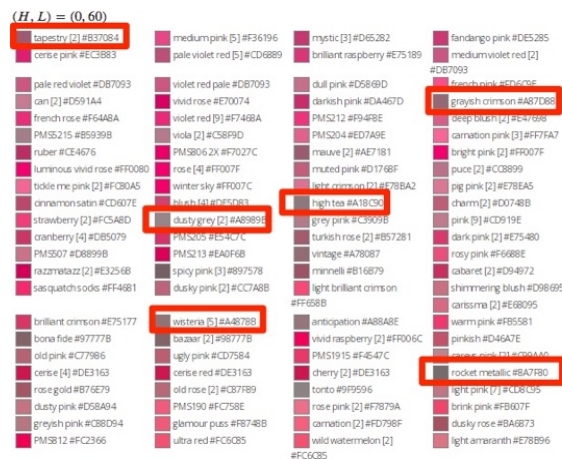


Figure 6.1 $(H,L)=(0,40)$ color group



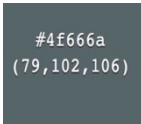


Beer color Tea color

Figure 6.2 Beer color and tea color

Therefore, how to find works with similar colors based on human senses is a challenge. In addition, describing a color from the metaphor of objects that exist in everyday life may also be a way to find color information, for example, by describing a certain color as ‘beer color’ or ‘tea color’.

As shown in Figure 6.2, since these descriptions and colors vary based on individuals, the colors one may want to look for are not the same according to the different senses of each person. The same problem may also occur to users who want to find the name of the pigment by description, as there may be cultural differences between countries in the description of objects and colors. In several different color name databases [121][122][123], the phenomenon shown in Table 6.1 can be found. There are considerable differences in color description between different language groups, even in different databases of the same language.

Table 6.1 Description of the color name for the same color in different databases

Japanese [123]: 濁った緑みの青 (Turbid green)	English 1 [121]: Stormcloud	English 2 [122]: Sumatra Chicken
		

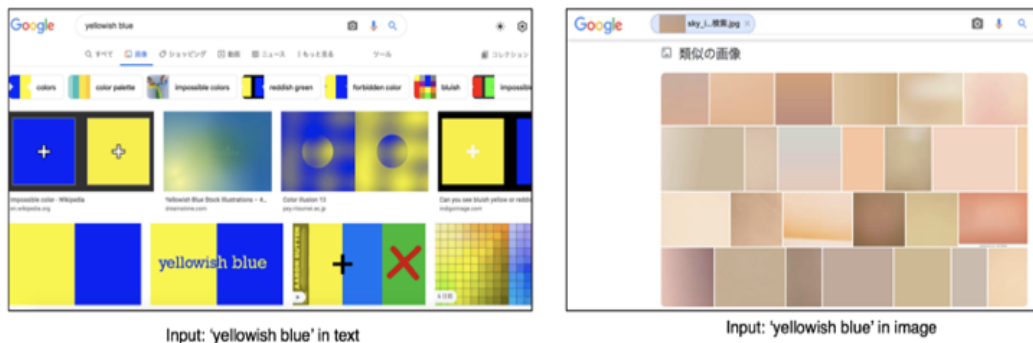


Figure 6.3 Results for 'yellowish blue' using text-image search and image-image search

In addition, a more complex challenge is to retrieve the impossible color. Newall [124] shows that some people describe the color of 'sky illuminated by the sun' as 'yellowish blue'. In fact, this color does not exist in the structure of color spaces. However, through the adjustment of light and other elements, it is possible to simulate this seemingly natural color, and this research aim to show examples of this in order to fit the requirement of researcher or designer who is focusing on the studies related to color and artworks. Using similar image retrieval and text-image retrieval from Google, as can be seen in Figure 6.3, the search results show the given color example and the text 'yellowish blue'.

Hence, it is possible to find more similar colors that fit human senses through language description, and therefore how to map different colors to natural language descriptions is a challenge for searching artworks based on colors.

Search for obscure colors in artworks, as shown in Figure 6.4, taking Japanese ukiyo-e as an example, some colors that are less used on the picture often attract more attention. Many artists use rare pigment materials and toning methods in their paintings, in many cases, these rare color schemes do not appear on a large area of the paintings. If the retrieval method is based on common feature extraction methods of color space, these

less noticeable color schemes are easily ignored. In addition, it is also a challenge to learn an image feature that does not ignore rare colors for artwork retrieval.



Figure 6.4 Obscure colors in ukiyo-e

Sketch is also the basis of painting, and also an important way to study the characteristics of image composition and an artist's style. Differing from ordinary image retrieval, the separation of color and sketch for feature extraction is meaningful for the retrieval of artworks. When ukiyo-e prints are digitized, the color information may be affected by factors such as light, or angle. This is one of the major challenges in the retrieval of artworks. If just simple color features such as RGB to describe the artwork are used, it can be affected by light, thereby the color information of each artwork is very unequal in the collection process, which will also greatly affect the accuracy of artwork retrieval. The same work from different institutions after digitizing are shown in Figure 6.5.

We do not know whether differences between different versions of the same work is caused by color difference, angle changes, or some blemishes.



Input image 1 Input image 2 Input image 3

Figure 6.5 The same work from the databases of different institutions

Table 6.2 Similarity between image pairs in IMGonline and DeepAI

Input pair	IMGonline [125]	DeepAI [126]
Input image (1,3)	Similarity _{1,2} : 81.80 %	Distance _{1,2} : 6
Input image (2,3)	Similarity _{2,3} : 90.05 %	Distance _{2,3} : 2
Input image (1,3)	Similarity _{1,3} : 87.11 %	Distance _{2,3} : 6

Table 6.2 shows the similarity calculation results of three images in Figure 6.5 by two different images similarity calculation websites [125][126], since they belong to commercial applications, the detailed calculation methods are not explained. It can be seen that there is a big difference between the two calculation methods for the pair of the input image 2 and the input image 3 under the two calculation methods, even though they are the same image. Hence, extracting an image sketch to obtain structural features, extracting image color information and representing it in an appropriate way are indispensable considerations for the retrieval of artworks. Paying attention to the similarity of the structure helps to retrieve the works with different versions in some databases or even the works with fading phenomenon.

To attack the problems and challenges mentioned above, we propose a framework for color-based artworks retrieval. The proposed framework uses the pre-trained model of CLIP (Contrastive Language-Image Pre-Training) [127] for multitasking fine-tuning, as shown in Figure 6.6, the pre-trained CLIP model has learned a lot of text-image

correspondence knowledges, which can link the image color information in real life with the text descriptions. Our framework allows the model to learn more about the cross-modal information of the artworks and color language description, and also optimizes the ability to learn to encode images with similar structures into near vector spaces.

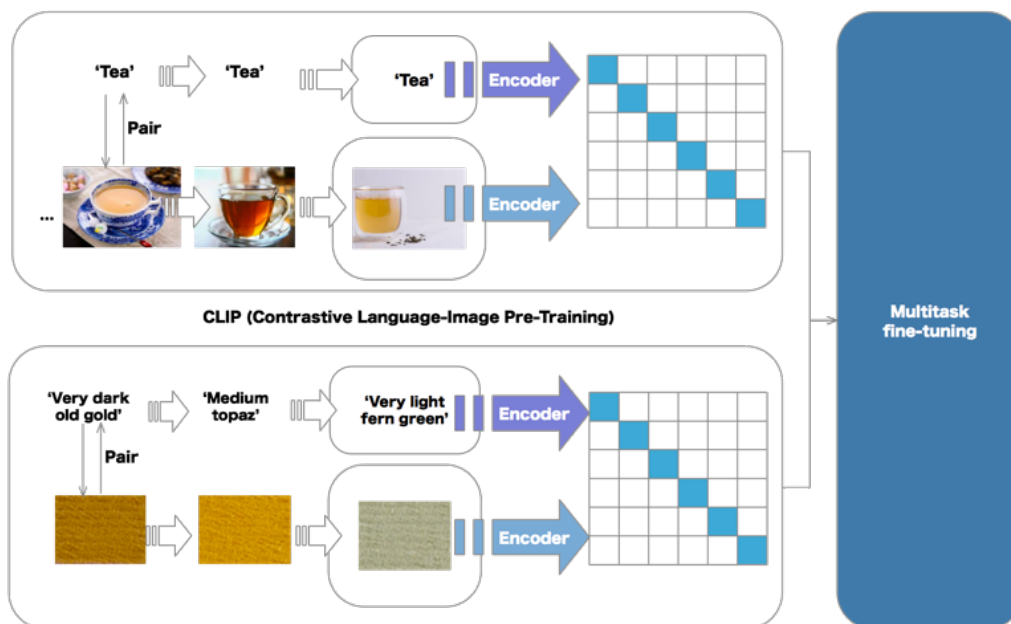


Figure 6.6 Multitask fine-tuning using a pre-trained CLIP model that learned a large amount of cross-modal knowledge from a large number of real life images

6.2 Contributions

The main contributions of our work are summarized as follows:

For the retrieval of artworks, we propose a new retrieval framework for word-based color retrieval of artworks.

1. We propose a new artwork color descriptor, transform the color information into the text feature space for obtaining the similar color based on human senses by using textural semantic space.

2. We apply the IDF (inverse document frequency) method commonly used in document retrieval to extract image color information, proposed a label generation

method for finding obscure colors.

3. A training data sampling method using sketch structure is proposed, images with the same structure can learn more similar feature vector representation.

We apply the proposed method to retrieving ukiyo-e prints by two ways colors selections, direct main color retrieval using colors of palette and search of obscure colors using text descriptions. By modifying the training setting, the main color can also be retrieved by the language description.

6.3 Related Work

There are already many retrieval systems for artworks in museum collections [128][129]. Artrieval et al. [130] proposed an effective way to search the paintings by exploiting the color to express human visual memory, and they provide an interface for non-expert users, where users can search by drawing graphics on the interface. NBS-kmeans algorithm was proposed for color clustering, and Hierarchical-LMNN (HLMNN) was proposed to calculate the distance between query and images in the database. Wang et al. [131] proposed a framework based on sketch, in which user's drawing style can be inferred by analyzing contour features, and the results are calculated by reranking K-nearest sketches stored in the user's historical recordings. Zhao et al. [132] compared art classification performances of seven different models when either using or not using transfer learning. The models include ResNet and its variants (e.g., RegNet, ResNeXt, Res2Net, ResNeSt and EfficientNet). For the task of classifying paintings by artist, style or genre, the color information from the paintings was used by the CNNs to perform classification. Companioni-Brito et al. [133] proposed an image retrieval framework on art paintings using shape, texture and color features. Locality Sensitive Hashing (LSH) method was

used for image indexing. However, all of these researches are for monomodal retrieval. Conde et al. [134] fine-tuned descriptions and images of artworks on CLIP (Contrastive Language-Image Pre-Training) model. However, in addition to paintings, training data also includes photos of museum collections. This research mainly focuses on learning the multimodal representation of artwork images and their description, but does not pay attention to the learning of the spatial representation of color information and structure information of artwork images.

6.4 Methodology

Figure 6.7 shows the architecture of our proposed method. The space sampler is used to sample the image triplet pairs from an image dataset for fine-tuning the weights of the image encoder. As shown in Figure 6.8, this module contains three processes: (1) image sketch extraction, (2) histogram of oriented gradients (HOG) feature extraction and (3) triplet data sampling. This module contains image processing, but these processed images are only used to sample the image data based on the perspective of geometric features, and the processed images are not used in fine-tuning training on CLIP model.

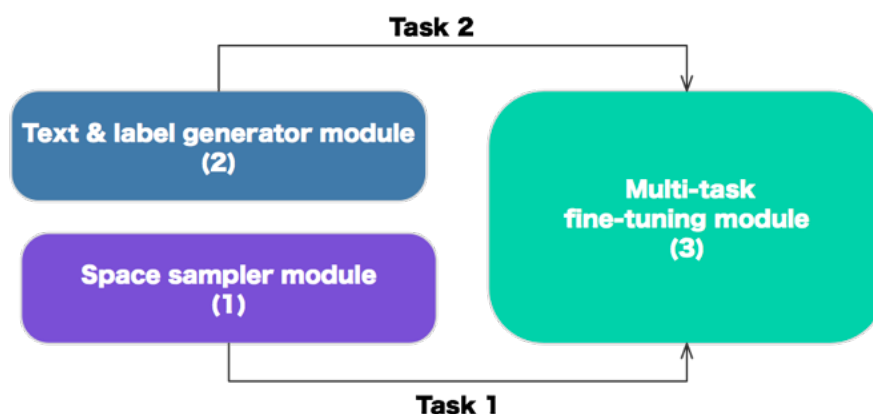


Figure 6.7 Architecture of our cross-modal multitask fine-tuning representation learning framework

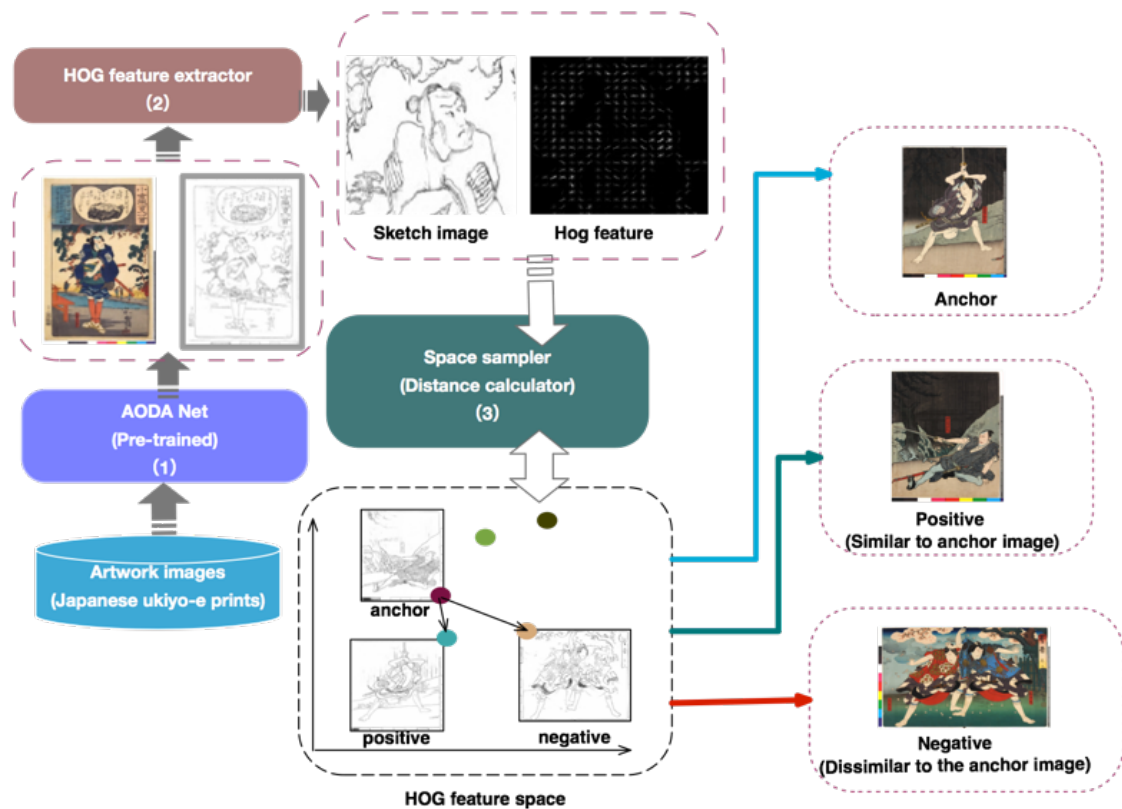


Figure 6.8 Structure of space sampler module

AODA Net (Adversarial Open Domain Adaption Network) is an open-domain sketch-to-photo translation framework which was proposed by Xiang et al [135]. A sketch image closer to the hand-drawn style can be obtained, by utilizing the pre-trained weights trained on large scale sketch-photo dataset.

As shown in Figure 6.9, we applied AODA Net and its pre-trained weight to sketch all the images and get their structural feature.

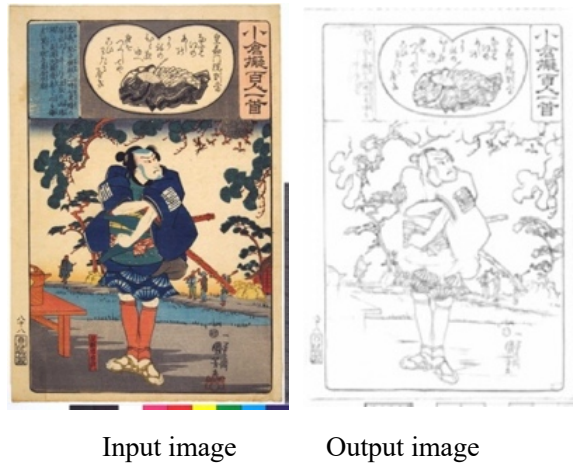


Figure 6.9 Example sketch image output from AODA Net

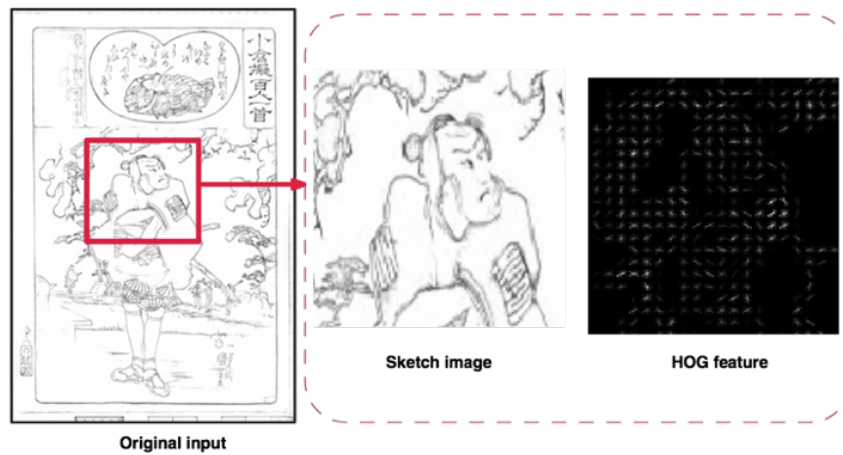


Figure 6.10 An example of HOG feature map

As introduced in Chapter 4 and 5, the Histogram of oriented gradients (HOG) feature is an efficient way to extract the gradient orientation feature in localized portions of an image. The distribution of directions of gradients are used as features to represent an input image. Figure 10 shows an example of HOG feature map. It can be seen that HOG feature can obtain the gradient feature of the sketch edge and not influenced by information such as the color of the original image.

Features obtained from the HOG feature extractor are used for sampling the triplet data, which are used as triplet pairs for fine-tuning task. We randomly sample three

images from the training data of the image each time, randomly select one as the anchor image, and use distance calculation to calculate which of the remaining two images is closer to the anchor image and which one is farther away. Definition of the sampler is (6.1):

$$\begin{cases} x_i = x_{anchor} & \text{selected as } x_i, \\ x_j = x_{positive} & \text{if } D(x_i, x_j) < D(x_i, x_k) \\ x_k = x_{negative} & \text{if } D(x_i, x_k) > D(x_i, x_j) \end{cases} \quad (6.1)$$

where x_i, x_j and x_k are the triple of random choices from the training data, function $D ()$ stands for the distance function. This is set as regular cosine similarity function, and other similarity metrics like Euclidean distance, etc. also can be used for distance calculation. Labels *anchor*, *positive* and *negative* are calculated by the distance between x_i, x_j and x_k . x_{anchor} , $x_{positive}$ and $x_{negative}$ are used as one of the training pairs of the triplet training task on CLIP model.

In order to train cross-modal information, the training data of ‘text-image’ pair needs to be prepared. In this section, a text and label generator is proposed to help the pre-trained model learn more ‘color description-image’ information. The generator includes two processes: (1) color information extraction and (2) IDF (inverse document frequency) calculation.

We applied colorgram [136] to extract color information from images. Results and their proportion are shown in Table 6.3.

Table 6.3 Example of color information extraction

	Color proportion
	Desert Sand : 34%
	Wenge : 16%
	Pale Taupe : 13%
	Deep Space Sparkl : 7%
	Yankees Blue : 7%
	Bourbon Spice : 6%
	Bistre : 3%
	Shadow Blue : 2%
	Azul Petróleo : 1%
	Jet : 1%
	Bugman's Glow : 1%
	Wenge : 1%

12 main colors and their proportions are extracted. The corresponding color descriptions are extracted from Handpicked color names dataset (HCN) [122] based on RGB value. For the RGB values that are not recorded in HCN, we use nearest neighbor search (NNS) to find the RGB value recorded in HCN dataset that is most similar to a given RGB value.

We extract the color-proportion index and color description document from image data. In the color description document, the name of each color is treated as a word unit.

As shown in Figure 6.11, we convert the extracted color information into a textual document for IDF label generation and a color-proportion index for the main color search by using a palette interface.

By calculating the frequency of the color name words that appear in the ‘document’ extracted from all images, colors with the top 6 highest frequencies of the entire training dataset are shown in Figure 6.12.

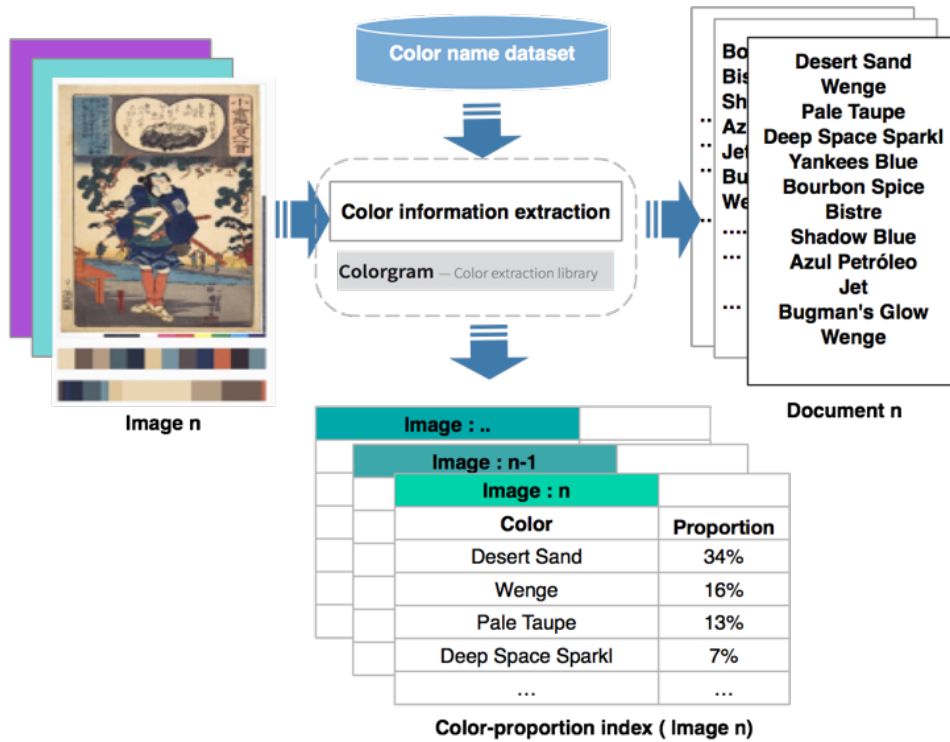


Figure 6.11 IDF calculation

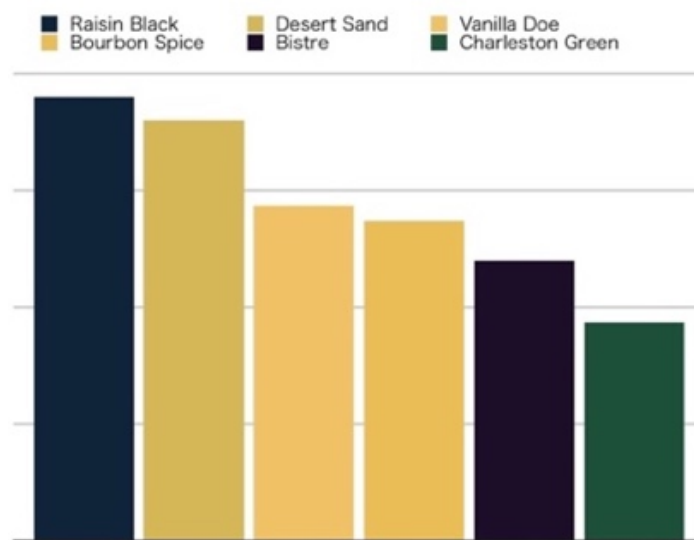


Figure 6.12 Colors with the top 6 highest frequencies

It can be seen that these are the colors of the background or the color of the lines in ukiyo-e. If the task needs to find obscure colors in the image, the importance of these

colors is considered in relatively low values. We use IDF (inverse document frequency), a common method in text retrieval, to calculate the low frequency color names that appear in all image color name documents. $Score_{color\ name}(x^i)$ is used to indicate the importance of each color name and is calculated as (6.2):

$$x_{min-max}^i \left(\frac{\text{sigmoid}\left(\log\left(\frac{N}{df_x}\right)\right)^i - \mu_{X_{sigmoid}}}{\sigma_{X_{sigmoid}}} \right), \quad (6.2)$$

where df_x shows the number of image color documents containing color name x_i , N means the total number of image color documents. $\mu_{X_{sigmoid}}$ and $\sigma_{X_{sigmoid}}$ show the sample mean and sample standard deviation of all the result from $\log\left(\frac{N}{df_x}\right)$ from all the color name of all image color documents. The function $x_{min-max}^i()$ show the Min-Max normalization.

The function $\text{sigmoid}(z)$ represents the calculation (6.3):

$$\frac{1}{1+e^{-z}}. \quad (6.3)$$

	Gruyère Cheese	Wenge	Pale Taupe	Deep Space Sparkl	Yankees Blue	Bourbon Spice	Bistre	Shadow Blue	Azul Petróleo	Jet	Bugman's Glow	Wenge
color												
score	0.71	0.77	0.76	0.87	0.60	0.05	0.65	0.99	0.95	0.85	0.87	0.77

Figure 6.13 Example of color names and their $Score_{color\ name}$

Figure 6.13 shows the $Score_{color\ name}$ of the color names from the image in Table 6.3. It can be seen that blue, which many art researchers focus on, gets a high score. If the task requires the result to be biased towards extracting the dominant color, the opposite value can be used on the training loss function setting.

We set two tasks to fine-tune the CLIP model: (1) cross-modal contrastive fine-

tuning task for learning the cross-modal knowledge between color name descriptions and images, (2) fine-tuning task on image encoder with triplet loss for learning the geometric similarity feature of the image sketch structure.

As shown in Figure 6.14, the training data used for cross-modal fine-tuning is a pair of images and texts of color names. We extracted a total of 12 colors from each image, so it means each image corresponds to 12 sets of training data pairs.



Figure 6.14 Example of training data of cross-modal fine-tuning task

The similarity function is defined as (6.4):

$$\text{Cosine}_{I_{\text{image}} \sim T_{\text{colorname}}} = \frac{\sum_1^n I_{\text{image}(i)} T_{\text{colorname}(i)}}{\sqrt{\sum_1^n I_{\text{image}(i)}^2} \sqrt{\sum_1^n T_{\text{colorname}(i)}^2}}, \quad (6.4)$$

where $\sum_1^n I_{\text{image}(i)} T_{\text{colorname}(i)}$ is the dot product of image and text representation I_{image} and $T_{\text{colorname}}$ learned from the model. Loss function is defined as (6.5):




$$\mathcal{L}_{\text{Pair}_n}(I_{\text{image}}, T_{\text{colorname}}, y) = \left\| y - \text{Cosin}_{\text{Pair}_n}(I_{\text{image}} \sim T_{\text{colorname}}) \right\|, \quad (6.5)$$

where $\mathcal{L}_{\text{Pair}_n}(I_{\text{image}}, T_{\text{colorname}}, y)$ represents the loss value generated when inputting an image - color name pair.

In order to enhance the learning of representation that focuses on image structure, triplet loss is used to improve the performance of image encoding.

Table 6.4 shows one of the input pairs for the structural-feature-based triplet training task.

Table 6.4 Example of training data of structural-feature-based triplet fine-tuning task

Task	Fine-tuning for learning image representation focusing on similarity of sketch structure		
Image-label pair	label: anchor 	label: positive 	label: negative 

6.5 Demo Implementation

A demo system was built to show the applicability of our proposed method. Examples of user groups targeted by the system are shown in Figure 6.15 :

As shown in Figure 6.16, we have implemented two modes to search for the color of ukiyo-e prints: (1) search artworks by word-based color description (2) By selecting color from a color palette.

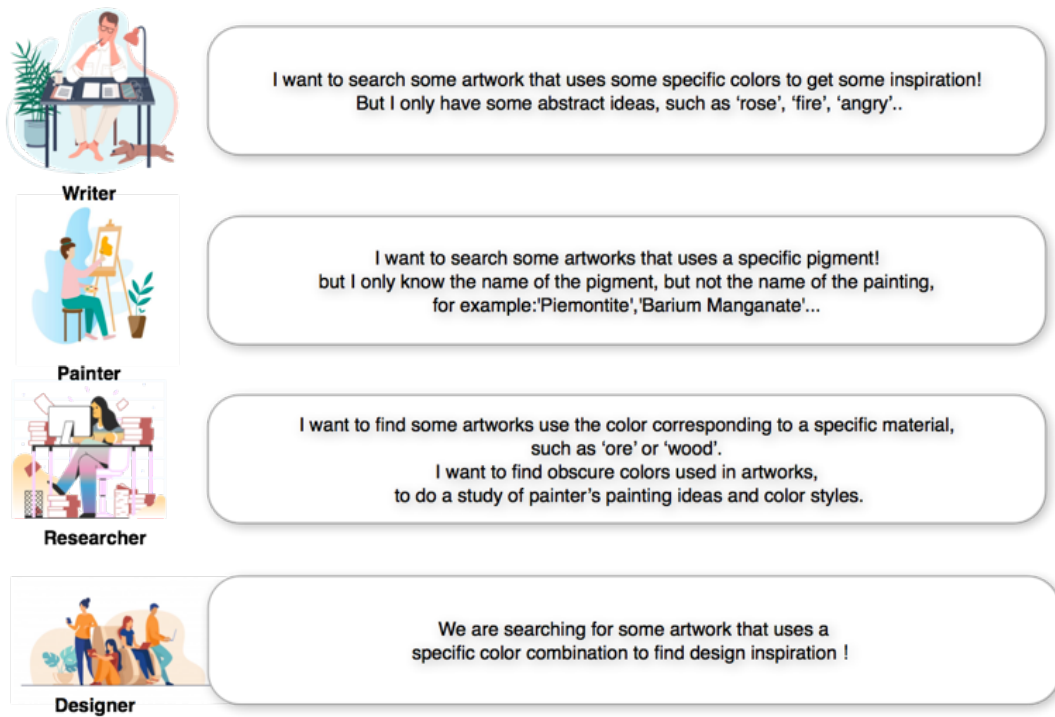


Figure 6.15 Target user groups and requirements

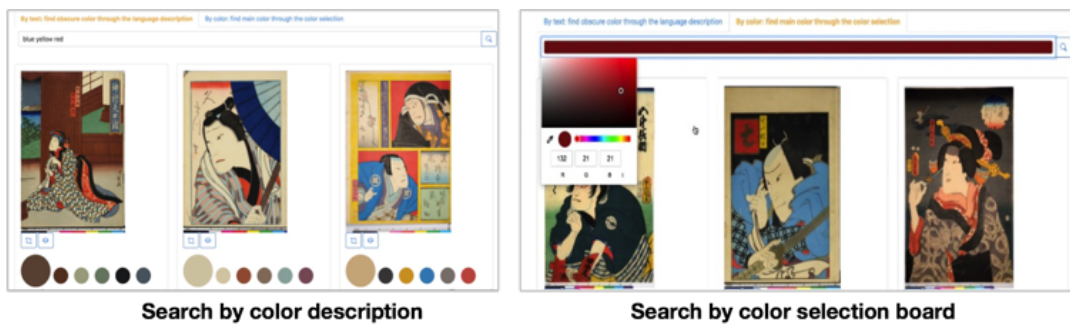


Figure 6.16 Demo application implementation

Some search results of the system are shown in the experimental section.

6.6 Experiments

In this section, we introduce the datasets used in our proposed framework and evaluation experiments.




All ukiyo-e images are from [9], and due to the high cost of calculation, we only selected 1,000 (images ID numbered 0-1000 in dataset) for training.

Two color name datasets in English are used for fine-tuning and evaluation experiments of the model: (1) Handpicked color names dataset (HCN) for training and testing, (2) Color-names (CN) dataset for evaluation. Handpicked color names dataset (HCN) is a handpicked list of 29,205 unique color names from various sources and thousands of curated user submissions. Color-names (CN) is a color name dataset including over 1,200 color names and corresponding RGB values. We extracted the ‘color name-RGB value’ pair data corresponding to the RGB values recorded in the two datasets. As shown in Table 6.5, the vocabulary differences between the two datasets are obvious, and the totally different color names account for almost 48%. Color names examples from HCN and CN are shown in Table 6.6.

Table 6.5 Vocabulary differences between CN and HCN

	Totally different	Overlap(differ in alphabetic case)	Exactly the same
Proportion	721(48%)	570(37%)	223(15%)

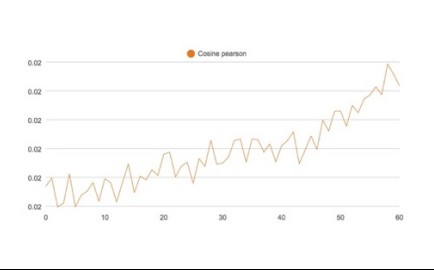
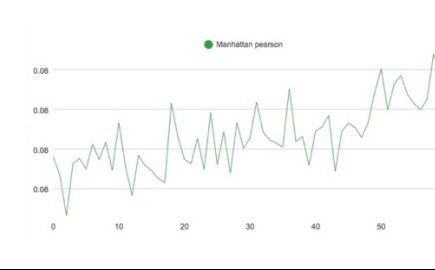
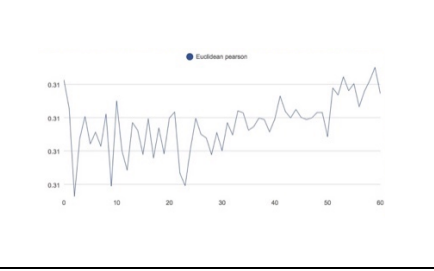
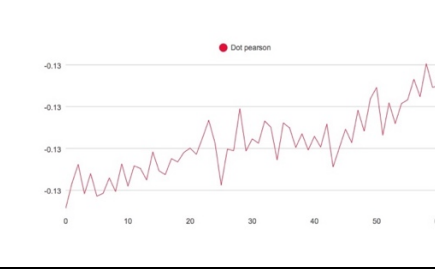
Table 6.6 Example of color name for the same color in different databases

	For training and testing: HCN	For evaluation: CN	sample
1	Manganese Red	Amaranth	
2	Hiroshima Aquamarine	Aquamarine	
3	Wet Ash	Ash grey	

20% of the training data is selected as the testing data during training process. Cosine pearson, Euclidean pearson, Manhattan pearson and pearson correlation are chosen to evaluate the training process. The Pearson correlation values calculated at 60 epochs are shown in Table 6.7.

It can be seen from Table 6.7 that as the fine-tuning task progresses, the corresponding relationship between text embedding and image embedding is increasing.

Table 6.7 Example of fine-tuning task progresses


























	Pearson correlation		Euclidean pearson
1		3	
2		4	

CN dataset and corresponding RGB color cards are used for model evaluation. Table 6.8 shows the results. Table 6.9 shows part of the image-image search results. It can be seen that the image features extracted by the model after fine-tuning cannot only be similar in color, but also have better structural similarity than other models.

Table 6.8 Cross-modal performance

Pre-trained model	Pearson correlation coefficient score
CLIP	0.7990
Fine-tuned CLIP	0.7996↑

Table 6.9 Examples of search results

Model	Query	Top 1	Top2	Top 3	Top 4
CLIP					
Cross-Vision Transformer					
VGG19					
HOG					
Ours					

6.7 Summary

This work started from the perspective of mining the color information of artworks. Taking the color psychology application of the cross-modal pre-training model as the ultimate goal, the multi-task fine-tuning structure design and preliminary evaluation of the trained model were carried out. The main innovation is that this is an attempt to link the color information space of artworks with language space, which can make color-related retrieval close to human senses. The experimental results show that a small amount of data can improve the correspondence between text description and color information, but a small amount of training data and insufficient training time are not enough to significantly improve the performance of the model. Further experimental evaluations on user queries were not conducted in the end of this research. What kind of queries users will have is a topic that needs to be considered in this research in the future.

We will investigate how to map multimodal information into a suitable space. We will also develop an end-to-end embedding model in a future study, as well as considering the use of knowledge graph to represent metadata.

Chapter 7 Conclusions and Future Work

This dissertation introduced the studies based on three types of digital cultural heritage archives data. Unlike typical image retrieval tasks, the data for this dissertation carries more distinct characteristics and language-level information than images in daily life, but lacks the common features of generalization of objects in real life such as more colors and textures. Like person identification based on personal signatures, the retrieval objects targeted by this research have distinct personalization characteristics based on geometric characteristics. There is a great deal of personal preference and uniqueness in the design of the seal: the handwritten characters are like individual signatures, and there are significant differences in the handwriting fonts of different people, as well as in the artistic styles, colors and compositions of artists, which are highly contextual and individual. Another challenge for this research is the limitation of the number of data and sample clarity. Most of the data used in this study comes from historical documents, and most of which are impossible to expand. There are a few samples or only one sample exists, and due to the passage of time, these samples themselves have been worn to a certain extent before being digitized. With the help of experts in the fields of history, humanities, etc., most of the data can be annotated, but there is still a part of the data that is not annotated, and it is a challenge for this study to apply this kind of data.

As a summary, for the seal data, this dissertation introduced some thoughts on collecting seal data, and introduced possible problems and challenges in data pre-processing, such as the overlap of handwritten characters and seals. Some pre-processing methods for seal images have been introduced, and a tree-structure-based seal image retrieval framework is proposed. At the same time, a character segmentation method based on unsupervised learning is proposed. Since the extracted characters are mostly

ancient characters based on unused kanji characters in modern times, we made a preliminary attempt to recognize ancient characters based on retrieval, and conducted our research on ancient character recognition.

For the study on ancient character recognition. In order to improve the problem of low-resource data, we tried from the perspective of data-augmentation and the perspective of metric-learning. In the metric-learning model we proposed, we introduced the conclusion that affine transformation in the data-augmentation attempt has improved the character classification effectiveness, and applied the spatial transformer into the deep learning framework, combined with the improved metric-learning-based model ProtoNet. Experimental results show that our model has advantages over the baseline and state-of-the-art methods in the feature extraction of cross-domain and multi-category classification tasks.

For the study on cross-modal retrieval for Japanese ukiyo-e prints, we have proposed a multi-task fine-tuning method for cross-modal encoder-decoder based model. We applied the fine-tuned model to the ukiyo-e retrieval task, which can find out the ukiyo-e images based on human sense color name description.

For future research tasks, we will work for the improvement of retrieval accuracy and the release of application demonstration. From the improvement of retrieval accuracy to the consideration of the individual needs of users is still a big challenge. It is the constant goal of this research to accurately extract rare knowledge from historical documents and provide experts and users with concise and convenient information tools.

Bibliography

- [1] Burdick, A., Drucker, J., Lunenfeld, P., Presner, T., & Schnapp, J. (2016). *Digital humanities*. Boston, MA: MIT Press.
- [2] Hai-Jew, S. (Ed.). (2017). *Data analytics in digital humanities*. Cham: Springer.
- [3] Zhang, B., Kumpulainen, S., & Keskustalo, H. (2021). *A review on information retrieval in the historical digital humanities domain*. Nagoya, Japan: IAFOR (International Academic Forum).
- [4] Ciula, A., & Eide, Ø. (2017). Modelling in digital humanities: Signs in context. *Digital Scholarship in the Humanities*, 32(Suppl_1), i33–i46.
- [5] Bradley, A. J., El-Assady, M., Coles, K., Alexander, E., Chen, M., Collins, C., ... & Wrisley, D. J. (2018). Visualization and the digital humanities. *IEEE Computer Graphics and Applications*, 38(6), 26–38.
- [6] Akiyama, T. (2014). *Struggles of the national diet library in collecting online publications in Japan*. Paper presented at the IFLA WLIC 2014 - Lyon - Libraries, Citizens, Societies: Confluence for Knowledge in Session 87 - Information Technology with Preservation and Conservation and National Libraries, Lyon, France. Retrieved from <http://ifla-test.eprints-hosting.org/id/eprint/886>
- [7] Waseda University Japanese and Chinese classics database. (n.d.). *About the database*. Retrieved from <https://www.wul.waseda.ac.jp/kotenseki/about.html>
- [8] Kyoto University Digitization Hub of the Humanities. (n.d.). *Social and cognitive sciences*. Retrieved from <https://www.bun.kyoto-u.ac.jp/events/kudh2021/>
- [9] Ritsumeikan University. (n.d.). *ARC database*. Retrieved from <http://www.arc.ritsumei.ac.jp/j/database/>
- [10] The Shirakawa Shizuka Institute of East Asian Characters and Culture. (n.d.). *Shirakawa Font project*. Retrieved from <http://www.ritsumei.ac.jp/acd/re/k-rsc/sio/shirakawa/index.html>
- [11] Carletti, L., Giannachi, G., Price, D., McAuley, D., & Benford, S. (2013). *Digital humanities and crowdsourcing: An exploration*. Retrieved from <http://hdl.handle.net/10871/17763>
- [12] Terras, M. M. (2016). Crowdsourcing in the digital humanities. In: S. Schreibman & R. Siemens (Eds.), *Companion to digital humanities II* (pp. 420–439). Oxford: Wiley-Blackwell.
- [13] ILMCORP. (n.d.). *Historical documents & archive scanning service*. Retrieved

- from <https://www.ilmcorp.com/services/document-scanning/historical-documents/>
- [14] Scans America. (n.d.). *Historical archive scanning*. Retrieved from <https://www.scansamerica.com/scanning-services/historical-preservation/>
- [15] Github. (n.d.). *LayoutParser*. Retrieved from <https://github.com/layout-parser/layout-parser>
- [16] Github. (n.d.). *SimpleHTR*. Retrieved from <https://github.com/githubharald/simplehtr>
- [17] Github. (n.d.). *Document layout analysis*. Retrieved from <https://github.com/bobld/documentlayoutanalysis>
- [18] Github. (n.d.). *Ochre*. Retrieved from <https://github.com/kbnlresearch/ochre>
- [19] TopOCR. (n.d.). *Homepage*. Retrieved from <https://www.topocr.com/>
- [20] Schröder, C., Müller, L., Niekler, A., & Potthast, M. (2021). *Small-text: Active learning for text classification in Python*. arXiv preprint arXiv:2107.10314. Retrieved from <https://arxiv.org/abs/2107.10314>
- [21] Rubrix. (n.d.). *Welcome to Rubrix*. Retrieved from <https://rubrix.readthedocs.io/en/stable>
- [22] Github. (n.d.). *Sklearn-flask-docker*. Retrieved from <https://github.com/chrisalbon/sklearn-flask-docker>
- [23] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2–3), 259–284.
- [24] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781. Retrieved from <https://arxiv.org/abs/1301.3781>
- [25] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Stroudsburg PA: Association for Computational Linguistics.
- [26] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations*. arXiv preprint arXiv:1802.05365. Retrieved from <https://arxiv.org/abs/1802.05365>
- [27] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 1137–1155.
- [28] Pham, N. Q., Kruszewski, G., & Boleda, G. (2016). Convolutional neural network language models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1153–1162). Stroudsburg PA: Association for Computational Linguistics.

- [29] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech* (pp. 1045–1048). Chiba: International Speech Communication Association.
- [30] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need*. arXiv preprint arXiv:1706.03762. Retrieved from <https://arxiv.org/abs/1706.03762>
- [31] Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2* (pp. 2787–2795). New York, NY: Association for Computing Machinery.
- [32] Xiao, H., Huang, M., & Zhu, X. (2016). From one point to a manifold: Knowledge graph embedding for precise link prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (pp. 1315–1321). New York, NY: Association for Computing Machinery.
- [33] Bordes, A., Weston, J., Collobert, R., & Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 25, No. 1, pp. 301–306). Boston, MA: AAAI Publications.
- [34] Nickel, M., Tresp, V., & Kriegel, H. P. (2012). Factorizing yago: Scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web* (pp. 271–280). New York, NY: Association for Computing Machinery.
- [35] Nickel, M., Rosasco, L., & Poggio, T. (2016). Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1, pp. 1955-1961). Boston, MA: AAAI Publications.
- [36] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 701–710). New York, NY: Association for Computing Machinery.
- [37] Wang, D., Cui, P., & Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1225–1234). New York, NY: Association for Computing Machinery.
- [38] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). *Spectral networks and deep locally connected networks on graphs*. arXiv preprint arXiv:1312.6203. Retrieved from <https://arxiv.org/abs/1312.6203>
- [39] Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs

for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2* (pp. 2224–2232). New York, NY: Association for Computing Machinery.

[40] Li, K., Batjargal, B., & Maeda, A. (2018). Ownership stamp character recognition system based on ancient character typeface. In *International Conference on Asian Digital Libraries* (pp. 328–332). Cham: Springer.

[41] Li, K., Batjargal, B., & Maeda, A. (2019). Character segmentation in collector's seal images: An attempt on retrieval based on ancient character typeface. In *Proceedings of the 5th International Workshop on Computational History (HistoInformatics)* (pp. 40–49). Porto, Portugal: CEUR-WS.

[42] Li, K., Batjargal, B., Maeda, A., & Akama, R. (2020). Toward exploring artist information from seal images in ukiyo-e collections. In *Conference Abstracts of Digital Humanities* (c13). Ottawa, Canada: Alliance of Digital Humanities Organizations.

[43] Maeda, A., Batjargal, B., & Li, K. (2021). Character segmentation in Asian collector's seal imprints: An attempt to retrieval based on ancient character typeface. *Journal of Data Mining & Digital Humanities*.

[44] Li, K., Batjargal, B., Maeda, A., & Akama, R. (2019). A seal retrieval system for ukiyo-e collections: Toward exploring artist information from retrieval results. *Proceedings of Symposium on Computers and the Humanities (JINMONKON 2019)*, 2019(1), 261–266.

[45] Hirose, S., Yoshimura, M., Hachimura, K., & Akama, R. (2008). Authorship identification of ukiyoe by using rakkan image. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems* (pp. 143–150). New York, NY: IEEE.

[46] Oohara, K., Yoshimura, M., & Hachimura, K. (2009). Automatic Extraction of Rakkan character string from ukiyoe. *Proceedings of Symposium on Computers and the Humanities (JINMONKON 2009)*, 2009(16), 41–48.

[47] Fujitsu Research and Development Center. (2016, March 30). *Seal retrieval technique for Chinese ancient document images*. Retrieved from <https://www.fujitsu.com/cn/en/about/resources/news/press-releases/2016/frdc-0330.html>

[48] Onitsuka, Y., Ohyama, W., Yamada, T., Inoue, S., & Uchida, S. (2018). Convolutional feature extraction for Kaou image retrieval. *Proceedings of Symposium on Computers and the Humanities (JINMONKON 2018)*, 2018, 257–262.

[49] Aoike, T., Satomi, W., & Kawashima, T. (2018). Automatic extraction of illustration from images of documents and image retrieval. In *Proceedings of Symposium on Computers and the Humanities (JINMONKON 2018)*, 2018, 97–102. Tokyo: Information Processing Society of Japan.

- [50] Su, Y. C., Ueng, Y. L., & Chung, W. H. (2019). Automatic seal imprint verification systems using edge difference. *IEEE Access*, 7, 145302–145312.
- [51] Sun, B., Hua, S., Li, S., & Sun, J. (2019). Graph-matching-based character recognition for Chinese seal images. *Science China Information Sciences*, 62(9), 1–14.
- [52] Zahan, T., Iqbal, M. Z., Selim, M. R., & Rahman, M. S. (2018). Connected component analysis based two zone approach for bangla character segmentation. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1–4). New York, NY: IEEE.
- [53] Nguyen, K. C., & Nakagawa, M. (2016). Text-line and character segmentation for offline recognition of handwritten japanese text. *IEICE Technical Report*, 115(517), 53–58.
- [54] Liu, H., Lu, Y., Wu, Q., & Zha, H. (2007). Automatic seal image retrieval method by using shape features of Chinese characters. In *2007 IEEE International Conference on Systems, Man and Cybernetics* (pp. 2871–2876). New York, NY: IEEE.
- [55] Ren, C., Liu, D., & Chen, Y. (2011). A new method on the segmentation and recognition of Chinese characters for automatic Chinese seal imprint retrieval. In *2011 International Conference on Document Analysis and Recognition* (pp. 972–976). New York, NY: IEEE.
- [56] Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2019). Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4), 1–42.
- [57] Babenko, A., & Lempitsky, V. (2015). Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1269–1277). New York, NY: Association for Computing Machinery.
- [58] Murtagh, F., & Legendre, P. (2014). Ward’s hierarchical agglomerative clustering method: Which algorithms implement Ward’s criterion? *Journal of Classification*, 31(3), 274–295.
- [59] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05)* (Vol. 1, pp. 886–893). New York, NY: IEEE.
- [60] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37). New York, NY: JMLR W&CP.
- [61] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108.

- [62] Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556. Retrieved from <https://arxiv.org/abs/1409.1556>
- [63] Li, K. (n.d.). *Ukiyo-e artists relationship extraction*. Retrieved from <https://github.com/timcanby/DrawArelationGraphFromCSV>
- [64] Comaniciu, D., & Meer, P. (1999). Mean shift analysis and applications. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1197–1203). New York, NY: IEEE.
- [65] Li, K. (n.d.). *Seal character segmentation*. Retrieved from https://github.com/timcanby/collector-s_seal-imageprocessing
- [66] The Shirakawa Shizuka Institute of East Asian Characters and Culture. (n.d.). *Shirakawa Font project*. Retrieved from <http://www.ritsumeai.ac.jp/acd/re/k-rsc/sio/shirakawa/index.html>
- [67] Liu, C. L., Yin, F., Wang, D. H., & Wang, Q. F. (2011). CASIA online and offline Chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition* (pp. 37–41). New York, NY: IEEE.
- [68] Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556. Retrieved from <https://arxiv.org/abs/1409.1556>
- [69] Mika, S., Schölkopf, B., Smola, A., Müller, K. R., Scholz, M., & Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems* (Vol. 11, pp. 536–542). Cambridge, MA: MIT Press.
- [70] Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236–239.
- [71] Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference* (Vol. 15, No. 50, pp. 147–151). London: The Plessey Company Plc.
- [72] Choudhary, A., Rishi, R., & Ahlawat, S. (2013). A new character segmentation approach for off-line cursive handwritten words. *Procedia Computer Science*, 17, 88–95.
- [73] Chitrakala, S., Mandipati, S., Raj, S. P., & Asisha, G. (2012). An efficient character segmentation based on VNP algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 4(24), 5438–5442.
- [74] Gatos, B., Antonacopoulos, A., & Stamatopoulos, N. (2007). Handwriting segmentation contest. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* (Vol. 2, pp. 1284–1288). New York, NY: IEEE.

- [75] Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2019). The Omniglot challenge: A 3-year progress report. *Current Opinion in Behavioral Sciences*, 29, 97–104.
- [76] Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference* (Vol. 15, No. 50, pp. 147–151). London: The Plessey Company Plc.
- [77] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37). New York, NY: JMLR W&CP.
- [78] National Diet Library Digital Collections. (n.d.). *NDL-DocL dataset*. Retrieved from <https://github.com/ndl-lab/layout-dataset>
- [79] Center for Open Data in the Humanities. (n.d.). *Seal script dataset*. Retrieved from <http://codh.rois.ac.jp/tensho/>
- [80] Anglia Ruskin University. (n.d.). *ETRVSCA Sans-Font typeface*. Retrieved from <https://arro.anglia.ac.uk/id/eprint/705654/>
- [81] Dafont. (n.d.). *MERO_HIE Hieroglyphics font*. Retrieved from <https://www.dafont.com/meroitic-hieroglyphics.font>
- [82] Dafont. (n.d.). *Aboriginebats font*. Retrieved from <https://www.dafont.com/aboriginebats.font>
- [83] Narang, S. R., Jindal, M. K., & Sharma, P. (2018). Devanagari ancient character recognition using HOG and DCT features. In *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 215–220). New York, NY: IEEE.
- [84] Vellingiriraj, E. K., Balamurugan, M., & Balasubramanie, P. (2016). Information extraction and text mining of Ancient Vattezhuthu characters in historical documents using image zoning. In *2016 International Conference on Asian Language Processing (IALP)* (pp. 37–40). New York, NY: IEEE.
- [85] Das, A., Patra, G. R., & Mohanty, M. N. (2020). LSTM based Odia handwritten numeral recognition. In *2020 international conference on communication and signal processing (ICCSP)* (pp. 0538–0541). New York, NY: IEEE.
- [86] Jayanthi, N., Indu, S., Hasija, S., & Tripathi, P. (2016). Digitization of ancient manuscripts and inscriptions-a review. In *International Conference on Advances in Computing and Data Sciences* (pp. 605–612). Singapore: Springer.
- [87] Rajakumar, S., & Bharathi, V. S. (2011). Century identification and recognition of ancient Tamil character recognition. *International Journal of Computer Applications*, 26(4), 32–35.

- [88] Romulus, P., Maraden, Y., Purnamasari, P. D., & Ratna, A. A. P. (2015). An analysis of optical character recognition implementation for ancient Batak characters using K-nearest neighbors principle. In *2015 International Conference on Quality in Research (QiR)* (pp. 47–50). New York, NY: IEEE.
- [89] Yue, J., Li, Z., Liu, L., & Fu, Z. (2011). Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling*, *54*(3–4), 1121–1127.
- [90] Chalechale, A., Naghdy, G., & Mertins, A. (2004). Sketch-based image matching using angular partitioning. *IEEE Transactions on Systems, Man, and Cybernetics-part a: Systems and humans*, *35*(1), 28–41.
- [91] Das, A., Patra, G. R., & Mohanty, M. N. (2020). LSTM based Odia handwritten numeral recognition. In *2020 international conference on communication and signal processing (ICCSP)* (pp. 0538–0541). New York, NY: IEEE.
- [92] Bhattacharjee, S. D., Yuan, J., Hong, W., & Ruan, X. (2016). Query adaptive instance search using object sketches. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 1306–1315). New York, NY: Association for Computing Machinery.
- [93] Zhao, W., Zhou, D., Qiu, X., & Jiang, W. (2021). Compare the performance of the models in art classification. *Plos One*, *16*(3), e0248414.
- [94] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv preprint arXiv:2010.11929. Retrieved from <https://arxiv.org/abs/2010.11929>
- [95] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1199–1208). New York, NY: IEEE.
- [96] Ghosh, A., Bhattacharya, B., & Chowdhury, S. B. R. (2017). Handwriting profiling using generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1, pp. 4927–4928). Association for the Advancement of Artificial Intelligence.
- [97] Creswell, A., & Bharath, A. A. (2016). Adversarial training for sketch retrieval. In *European Conference on Computer Vision* (pp. 798–809). Cham: Springer.
- [98] Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1415–1424). IEEE Computer Society.

- [99] 國家發展委員會. (n.d.). *True type character type*. Retrieved from <http://www.cns11643.gov.tw/downloadlist.jsp?id=2>
- [100] Tian, Y. C. (n.d.). *zi2zi Master Chinese Calligraphy with Conditional Adversarial Network*. Retrieved from <https://github.com/kaonashi-tyc/zi2zi>
- [101] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- [102] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 3637–3645). New York, NY: Association for Computing Machinery.
- [103] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37). New York, NY: JMLR W&CP.
- [104] Snell, J., Swersky, K., & Zemel, R. S. (2017). *Prototypical networks for few-shot learning*. arXiv preprint arXiv:1703.05175. Retrieved from <https://arxiv.org/abs/1703.05175>
- [105] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning* (pp. 1126–1135). York: MLR Press.
- [106] Patacchiola, M., Turner, J., Crowley, E. J., O’Boyle, M., & Storkey, A. J. (2020). Bayesian meta-learning for the few-shot setting via deep kernels. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 33, 16108–16118.
- [107] GitHub. (n.d.). *Pytorch implementation of DKT*. Retrieved from <https://github.com/BayesWatch/deep-kernel-transfer>
- [108] Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)* (pp. 117–122). New York, NY: IEEE.
- [109] Ono, T., Hattori, S., Hasegawa, H., & Akamatsu, S. I. (2000). Digital mapping using high resolution satellite imagery based on 2D affine projection model. *International Archives of Photogrammetry and Remote Sensing*, 33, 672–677.
- [110] Golubitsky, O., Mazalov, V., & Watt, S. M. (2010). Toward affine recognition of handwritten mathematical characters. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (pp. 35–42). New York, NY: Association for Computing Machinery.
- [111] Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). *Spatial transformer networks*. arXiv preprint arXiv:1506.02025. Retrieved from

<https://arxiv.org/abs/1506.02025>

- [112] Iamsa-at, S., & Horata, P. (2013). Handwritten character recognition using histograms of oriented gradient features in deep learning of artificial neural network. In *2013 international conference on IT convergence and security (ICITCS)* (pp. 1–5). New York, NY: IEEE.
- [113] Dou, T., Zhang, L., Zheng, H., & Zhou, W. (2018). Local and non-local deep feature fusion for malignancy characterization of hepatocellular carcinoma. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 472–479). Springer, Cham.
- [114] Huang, S., Xu, H., Xia, X., Yang, F., & Zou, F. (2019). Multi-feature fusion of convolutional neural networks for Fine-Grained ship classification. *Journal of Intelligent & Fuzzy Systems*, 37(1), 125–135.
- [115] Omniglot. (n.d.). *The encyclopedia of writing systems and languages*. Retrieved from <https://omniglot.com/index.htm>
- [116] TensorFlow. (n.d.). *Emnist*. Retrieved from <https://www.tensorflow.org/datasets/catalog/emnist>
- [117] Omniglot. (n.d.). *Oracle Bone Script*. Retrieved from <https://omniglot.com/chinese/jiaguwen.htm>
- [118] Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980. Retrieved from <https://arxiv.org/abs/1412.6980>
- [119] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). New York, NY: IEEE.
- [120] Shirakawa, S. (2008). Kokotsu kinbungaku ronso. In *The collection of Shirakawa Szhezuka*. Tokyo: Heibonsha.
- [121] Ranatunga, D., & Gadoci, B. (n.d.). *Color-names*. Retrieved from <https://data.world/dilumr/color-names>
- [122] Meodai. (n.d.). *Handpicked color names*. Retrieved from <https://github.com/meodai/color-names>
- [123] Ironodata. (n.d.). *配色の見本帳 / キーカラーで選ぶ配色パターン*. Retrieved from <https://ironodata.info/>
- [124] Newall, M. (2021). Painting with impossible colours: Some thoughts and observations on yellowish blue. *Perception*, 50(2), 129–139.
- [125] IMGonline. (n.d.). *Homepage*. Retrieved from <https://www.imgonline.com.ua>
- [126] DeepAI. (n.d.). *Image similarity API*. Retrieved from <https://deepai.org/machine-learning-model/image-similarity>

- [127] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). *Learning transferable visual models from natural language supervision*. arXiv preprint arXiv:2103.00020. Retrieved from <https://arxiv.org/abs/2103.00020>
- [128] Goodall, S., Lewis, P. H., Martinez, K., Sinclair, P. A., Giorgini, F., Addis, M. J., ... & Stevenson, J. (2004). Sculpteur: Multimedia retrieval for museums. In *International Conference on Image and Video Retrieval* (pp. 638–646). Springer, Berlin, Heidelberg.
- [129] Sharma, M. K., & Siddiqui, T. J. (2016). An ontology based framework for retrieval of museum artifacts. *Procedia Computer Science*, 84, 169–176.
- [130] Kim, N., Choi, Y., Hwang, S., & Kweon, I. S. (2015). Artrieval: Painting retrieval without expert knowledge. In *2015 IEEE International Conference on Image Processing (ICIP)* (pp. 1339–1343). New York, NY: IEEE.
- [131] Wang, F., Lin, S., Luo, X., Zhao, B., & Wang, R. (2019). Query-by-sketch image retrieval using homogeneous painting style characterization. *Journal of Electronic Imaging*, 28(2), Article 023037.
- [132] Zhao, W., Zhou, D., Qiu, X., & Jiang, W. (2021). Compare the performance of the models in art classification. *Plos One*, 16(3), Article e0248414.
- [133] Companioni-Brito, C., Mariano-Calibjo, Z., Elawady, M., & Yildirim, S. (2018). Mobile-based painting photo retrieval using combined features. In *International Conference Image Analysis and Recognition* (pp. 278–284). Springer, Cham.
- [134] Conde, M. V., & Turgutlu, K. (2021). CLIP-Art: Contrastive pre-training for fine-grained art classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3956–3960). IEEE Computer Society.
- [135] Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., & Allebach, J. P. (2021). *Adversarial open domain adaptation for sketch-to-photo synthesis*. arXiv preprint arXiv:2104.05703. Retrieved from <https://arxiv.org/abs/2104.05703>
- [136] Darosh. (n.d.). *Colorgram*. Retrieved from <https://github.com/obskyr/colorgram.py>

Publication List

Journal Papers

1. Kangying Li, Biligsaikhan Batjargal, and Akira Maeda. Character Segmentation in Asian Collector's Seal Imprints: An Attempt to Retrieval Based on Ancient Character Typeface. *Journal of Data Mining and Digital Humanities*, Vol. HistoInformatics, pp. 1-19, Jan. 2021.

2. Kangying Li, Biligsaikhan Batjargal, and Akira Maeda. A Prototypical Network-Based Approach for Low-Resource Font Typeface Feature Extraction and Utilization. *Data*, 6(12), Dec. 2021.

International Conferences

1. Kangying Li, Biligsaikhan Batjargal, Akira Maeda, and Ryo Akama, Artwork Information Embedding Framework for Multi-source Ukiyo-e Record Retrieval. In *Proceedings of the 22nd International Conference on Asia-Pacific Digital Libraries (ICADL2020)*

2. Kangying Li, Biligsaikhan Batjargal, Akira Maeda, and Ryo Akama, Toward Exploring Artist Information from Seal Images in Ukiyo-e Collections. *Conference Abstracts of Digital Humanities 2020*.

3. Kangying Li, Biligsaikhan Batjargal, and Akira Maeda: Ownership Stamp Character Recognition System Based on Ancient Character Typeface. In *Proceedings of the 20th International Conference on Asia-Pacific Digital Libraries (ICADL2018)*, Hamilton, New Zealand, pp. 328-332, Nov. 2018.

4. Kangying Li, Biligsaikhan Batjargal, and Akira Maeda. Character Segmentation in Collector's Seal Images: An Attempt on Retrieval Based on Ancient Character Typeface. In *Proceedings of the 5th International Workshop on Computational History*

(HistoInformatics 2019), pp. 40-49, Oslo, Norway, Sep. 2019.

Domestic Conferences

1. 李 康穎, Biligsaikhan Batjargal, 前田 亮, 赤間 亮. 落款印および関連情報の検索システムの構築：人物情報と人物関係ネットワークの自動抽出に向けて. 人文科学とコンピュータシンポジウム論文集, pp.261-266 Dec. 2019.

2. 李 康穎, Batjargal Biligsaikhan, 前田 亮. 古代文字フォント字形の特徴抽出に基づく蔵書印の検索支援. 人文科学とコンピュータシンポジウム論文集, pp. 123-128, Dec. 2018.

3. 李 康穎, Batjargal Biligsaikhan, 前田 亮. 古代文字フォントの画像データに基づく手書き篆書体文字の検索支援. 人文科学とコンピュータシンポジウム論文集, pp. 125-130, Dec. 2017.

Other publications

1. Li Kangying, Batjargal Biligsaikhan, 前田 亮, 赤間 亮. 浮世絵レコードのクロスモーダル多言語横断検索に向けて：Multilingual-BERT による作品情報の特徴埋め込み抽出の試み. 第10回知識・芸術・文化情報学研究会, Feb. 2021.

2. 前田 亮, バトジャルガル ビルゲサイハン, 李 康穎. 古代文字のデジタル化とその活用の可能性. 第五十一回 日本古文書学会大会研究発表要旨, pp.7-8, Sep. (2018)

3. Li Kangying, Batjargal Biligsaikhan, 前田 亮. 生成モデルによる篆書体の文字認識手法の提案. 第10回データ工学と情報マネジメントに関するフォーラム(DEIM2018)論文集, Mar. (2018)

4. 李 康穎, Batjargal Biligsaikhan, 前田 亮. 篆書体による蔵書印の文字認識の試み. 第7回知識・芸術・文化情報学研究会, Feb. (2018)

5. 李 康穎, バトジャルガル ビルゲサイハン, 前田 亮. 白川フォントの画

像データに基づく手書き篆書文字検索支援. 第 8 回横幹連合コンファレンス論文集, Dec. (2017)

6. 李 康穎, Batjargal Biligsaikhan, 前田 亮. 古代文字検索のためのフォントからの字形特徴量の抽出および活用可能性の検討. 第 11 回データ工学と情報マネジメントに関するフォーラム(DEIM2019)論文集, Mar.2019.

Appendix A

The architecture of the convolutional encoder mentioned in Section 5.4.2 and Figure 5.8 is shown in Table A.1.

Table A.1 Architecture of the convolutional encoder

Layer	Settings
Conv2d-10	kernel_size = (3, 3), stride = (1, 1), padding = (1, 1)
Conv2d-11	kernel_size = (3, 3), stride = (1, 1), padding = (1, 1)
ReLU-13	(-)
MaxPool2d-14	kernel_size = 2, stride = 2, padding = 0, dilation = 1
Conv2d-15	kernel_size = (3, 3), stride = (1, 1), padding = (1, 1)
BatchNorm2d-16	eps = 1e-05, momentum = 0.1
ReLU-17	(-)
MaxPool2d-18	kernel_size = 2, stride = 2, padding = 0, dilation = 1
Conv2d-19	kernel_size = (3, 3), stride = (1, 1), padding = (1, 1)
BatchNorm2d-20	eps = 1e-05, momentum = 0.1
ReLU-21	(-)
MaxPool2d-22	kernel_size = 2, stride = 2, padding = 0, dilation = 1
Conv2d-23	kernel_size = (3, 3), stride = (1, 1), padding = (1, 1)
BatchNorm2d-24	eps = 1e-05, momentum = 0.1
ReLU-25	(-)
MaxPool2d-26	kernel_size = 2, stride = 2, padding = 0, dilation = 1

Appendix B

As shown in Figure B1, we select three classes of characters from OMNIGLOT, which are not related to the test domain in the structure of graphics, as the training domain.

We set the output size of the ‘Linear-29’ layer as two and used two classes of other characters for testing.

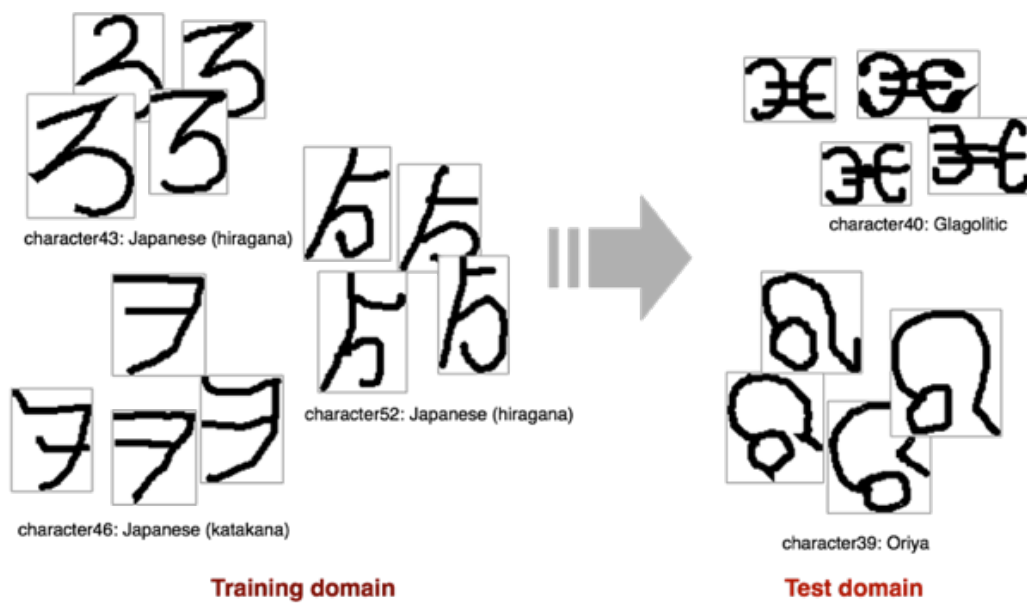
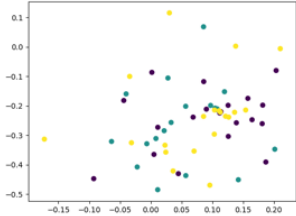
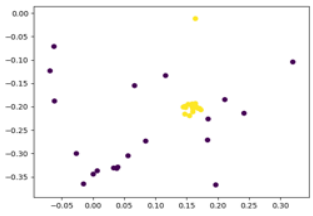
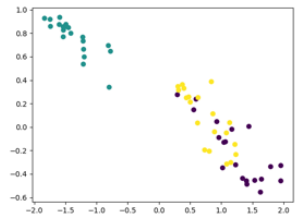
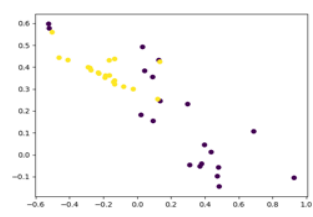
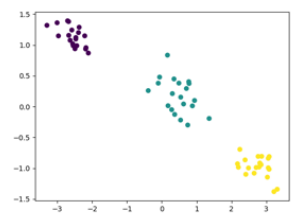
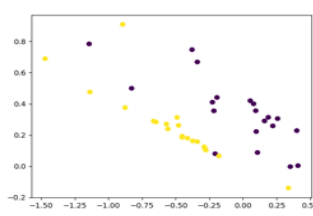
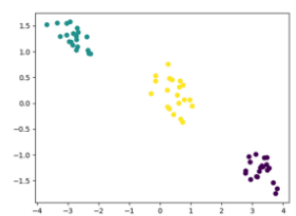
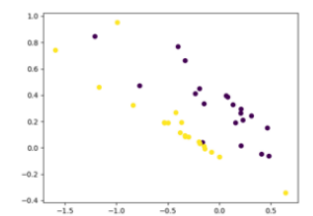


Figure B.1 Training domain and test domain

The training process is shown in Table B1. The scatter points of different colors in the figure represent the coordinate information output by different classes of characters from the ‘Linear-29’ layer of our model.

Table B.1 Training domain and test domain







Epoch	Training Domain	Test Domain
1		
50		
200		
500		

As can be seen from Table B1, although we did not fine-tune our model on the test domain, the output of the ‘Linear-29’ layer of our model in the test domain is gradually becoming linearly separated. It becomes adapted to the voting classifier in our proposed framework.

Appendix C

Queries removed from the test data are shown in Table C.1. As shown in the table, there is a significant difference between the query from OMNIGLOT and our font data from the Shirakawa font. This may be caused by different variations of the same character; hence, in our experiment, we did not use all 21 characters. Table C1 shows only four examples.

Table C.1 Examples of queries removed from the test data

Label Japanese (English)	Shirakawa Font [10]	OMNIGLOT [117]
魚 (fish)		
馬 (horse)		
妻 (wife)		
光 (light)	