TOKYO UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

DEPARTMENT OF ELECTRONIC AND INFORMATION ENGINEERING

****

An Attempt to Recognize Historical Documents of Chinese Origin

中国語起源の歴史文書認識の試み

Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electronic and Information Engineering

Student : NGUYEN Cong Kha

ID: 17834303

Supervisor: Prof. Nakagawa Masaki

Email: congkhanguyen@gmail.com

Tokyo, 11/2019

Acknowledgements

First, the author would like to thank his advisor **Prof. Masaki Nakagawa** of the Department of Computer and Information Sciences at Tokyo University of Agriculture and Technology (TUAT) for the continuous support of the author's Doctoral study and related research, for his patience, motivation, and immense knowledge. The door to **Prof. Masaki Nakagawa** office was always open whenever the author ran into a trouble spot or had a question about the author's research or writing. His guidance helped the author in all the time of research and writing of this thesis. The author could not have imagined having a better advisor and mentor for his Doctoral study.

Besides the author's advisor, he would like to thank his senior, **Dr. Phan Van Truyen**, who handed his research over to the author completely when he graduated from TUAT in 2015 and helped the author so much whenever the author was stuck with the research. **Dr. Truyen** also suggested the author new research directions, told the author the remaining problems needed to solve in each research, and collaborated to improve the current Japanese text handwriting system.

The author thanks his fellow lab-mates **Dr. Nguyen Tuan Cuong** and **Mr. Ly Tuan Nam** for the collaboration of participating in Japanese Kuzushiji recognition contest, writing academic journals, and conference papers, for the stimulating discussions on the state-of-the-art techniques in machine learning, deep learning as well as pattern recognition, for helpful suggestions to find out the best solutions, and for all the fun they have had in the last three years.

The author would also like to express his deepest gratitude to the Japan Science and Technology Agency, Ilabo company, the Nara National Institute for Cultural Properties (Nabunken), and the National Library of Vietnam (NLV), for financially supporting the study, for providing research materials and for helping the author to transfer his research to practical applications. Working with professional employees in Ilabo company and Nabunken are his honor. This is the wonderful time of the author's life and he will never forget.

Finally, the author must express his very profound gratitude to his family for providing him with unfailing support and continuous encouragement throughout his years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Once again thank you.

Author
NGUYEN Cong Kha

Table of contents

List of figures

List of tables

Chapter 1. Introduction

There are two topics the author has researched on: off-line Japanese handwriting recognition (recognition of characters on images ready captured by devices such as cameras, scanners), and apply image-processing methods, handwriting recognition and deep learning for historical document processing.

1.1. Background of the research

The research on Optical Character Recognition (OCR) has a long history [1], [2]. It started as printed character recognition, was extended to handwritten character recognition and printed document recognition, and then finally evolved into handwritten document recognition. The largest success was achieved in postal address recognition and form recognition of fixed formats.

When format and vocabulary are unrestricted, however, the optical handwritten text recognition (OHTR) is still a big challenge because of various distortions due to handwriting (variations of character size, unstable gaps between characters, touching of characters, irregular spacing between lines and so on). For Japanese OHTR, the problem is more difficult, because characters appear in various shapes and sizes (Kanji of Chinese origin, Kata-kana and Hira-gana of phonetic characters, alphabets, numerals, Greek letters punctuation marks and so on) in a document without any word spacing as shown in **Figure 1.1**. Another common problem is that Japanese always mixes the Chinese characters with phoentic hiragana and katakana chracters. Moreover, many Chiniese characters are composed of radicals, which themselves can be Chinese characters. Hira-gana used as conjugation parts are often written many times smaller than Kanji characters.

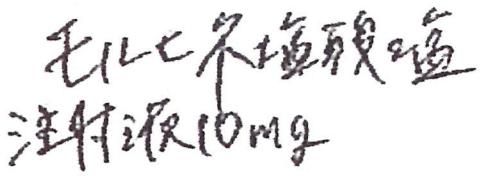


Figure 1.1. Japanese handwritten text including Kanji, Kana, numerals and alphabet

After achieving success with Postal code and address recognition, research and development spread to problems where format, vocabulary and/or layout were not restricted. Postal code and address recognition and some commercially successful applications were well-designed compromises between technologies and demands. Making a step ahead to the format-free and vocabulary-free handwritten Japanese text

recognition, however, the challenge has not been met yet. Due to this difficulty, manual typing in low labor-cost countries is continuing even today.

The diversity of character shapes, and distortion due to handwriting make deterministic segmentation and recognition difficult, and require efficient optimization strategies to segment and recognize Japanese handwritten text. This is the so-called over-segmentation approach, where over-segmented parts are combined and recognized by the best-path search for text recognition with character recognition, considering the linguistic and geometric contexts [3],[4],[5]. The linguistic context restricts the relation of characters with adjacent characters in daily context while the geometric context represents geometric relation between character patterns and within a character pattern.

Another approach to handwritten text recognition is the segmentation-free approach such as Messina et al.[6]. This approach has become feasible for handwritten Chinese or Japanese text due to the progress of deep neural networks. For practical applications, however, problems remain in recognition speed, difficulty of analyzing misrecognitions, and sensitivity to large aspect ratio as well as shape variabilities.

Moreover, other bottom-up approaches based on semantic segmentation get the attention of many researches. In these methods, text pages are firmly segmented into character patterns, then recognized by OCRs, and combined into contexts to get final results. Semantic segmentation is to classify the different parts of a visual input into real-world meaning classes. In case of character segmentation, we want to classify every pixels of a character pattern into its corresponding category and separate them with others.

For historical document processing, the author was working on two topics to preserve two heritages of Japan and Vietnam: Nara Hejoukyou mokkan and Chu Nom.

Mokkan is the Japanese name of wooden tablets, or wooden pieces, used as documents in ancient periods in Japan. They were used to write or carve characters on as shown in **Figure 1.2**. Until now, hundred millions of mokkans have been excavated over Japan. Some of them have been decoded by archeologists and proves invaluable properties about the culture, economy, and politics of that era in Japan. Therefore, the mokkan decoding is essential. Due to the severe impacts of outer conditions, however, most of the excavated mokkans are damaged, stained and often broken into pieces so that character images are rarely kept completely and hard for even veteran archeologists to decode.

Recently, due to the development of Information Technology (IT), especially the rapid progress of machine learning and pattern recognition, optical character recognitions (OCRs) are experimentally applied to recognize mokkans' characters, but due to the lack of training patterns, machine recognition is still a big challenge. Therefore, the author's research focus on improving the visual quality of mokkan character images. The author apply a generative adversarial network (GAN), trained by the normal adversarial loss and the hierarchical character attention loss to improve them so that OCRs and archeologists are able to recognize them.

Figure 1.2. A glass plate image of mokkan.

Nom is an ancient script used in Vietnam until the current Latin-based Vietnamese alphabet became common as shown in **Figure 1.3** . From the tenth century to the twentieth century, all of the documents in Vietnam were recorded by Nom, so that tens of thousands of Nom documents are stored in families, pagodas, churches, and libraries. We face a high risk that the invaluable Vietnamese history would be lost and could not be accessed by the next generations because Nom documents are gradually degrading, most of them have not yet been digitized, and the number of scholars who can read Nom documents is getting smaller.

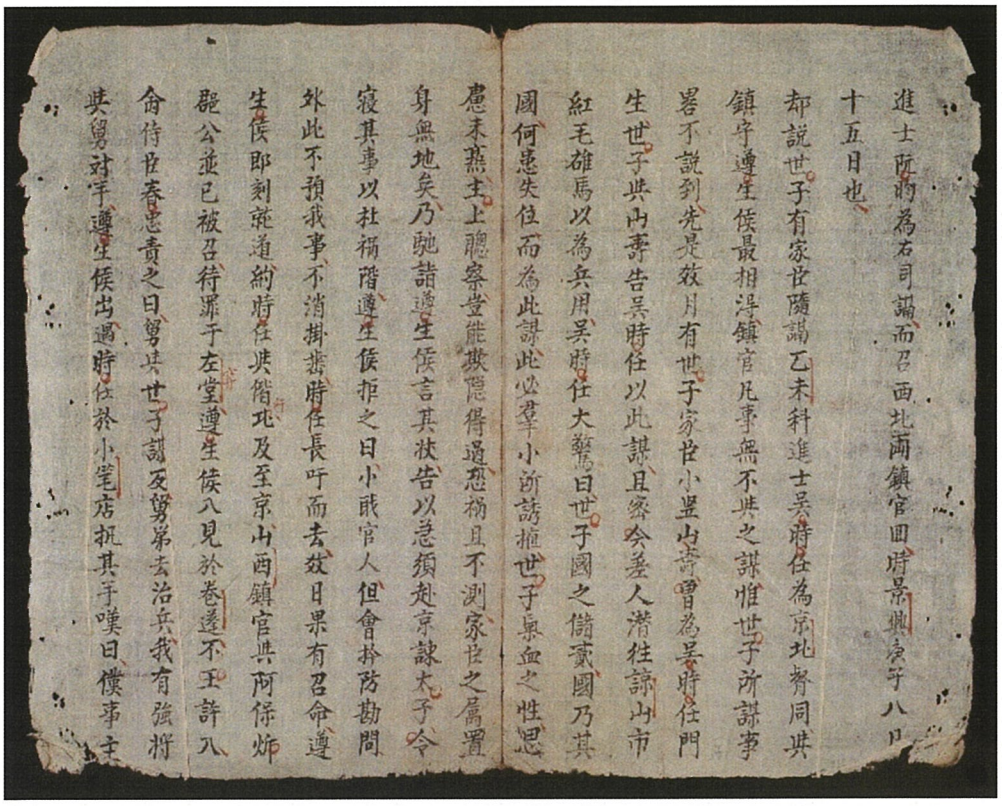


Figure 1.3. Nom text page

Recently, several institutes and libraries started projects to digitalize Nom documents for reserving this heritage such as the National Library of Vietnam [7], the General Library of Thua Thien Hue and the Temple University Library [8], the Tue Quang wisdom light foundation [9] and so on. Nevertheless, they share the common drawback that they do not have highly accurate OCRs for recognizing Nom characters after scanning Nom documents, so that the digitalizing process mostly depends on the interpretation by Nom experts. Now all over the world, however, the number of experts who can comprehend Nom script is less than 100, and most of them are aged.

Nom has a large character set of tens of thousands of characters. It has Chinese origin and new characters were composed of characters or their radicals [10]. About 60% of Nom characters were invented by ancient Vietnamese people. This proportion increased while characters of Chinese origin decreased, as the documents were made later.

Previously, we developed a system to recognize Nom characters on the documents scanned by the National Library of Vietnam [11]. In the system, Nom documents are firstly preprocessed and binarized, then segmented by the method based on projection profiles and the Voronoi diagram. The segmented patterns are recognized by OCRs using generalized learning vector quantization (GLVQ) and modified quadratic discriminant function (MQDF). Finally, the system provides a GUI for users to revise the recognition results and save the results to text files. Although we applied the histogram analysis method to remove boundary lines and rules lines in the Nom documents, many noises still remain, so that the above segmentation method often fails and requires user's corrections. Another problem is that the recognition rates of the OCRs are still low due to the lack of adequate training patterns for a large number of the Nom categories. The OCRs have not yet been combined with a language model for correcting wrong recognition results. Although the system allows users to automatically convert scanned Nom pages into text files, they have to check the recognized results again. This makes the digitalization process still consuming a huge human resource.

To resolve the above problems, the author first propose a character extraction method based on the U-Net structure [12]. He train the models with synthetic data from Nom fonts. To create artificial Nom pages, he generate single character patterns from fonts and paste them in different scales to empty pages. The ground-truths of character regions are convex-hulls of character patterns instead of rectangle boxes to reduce the overlap between character regions. Because the trained models sometimes produce touching character regions, the marker watershed method is applied to separate them.

Since Nom script includes tens of thousands of categories, and he do not have real training patterns for Nom, the OCRs using simple features like directional features are ineffective. The author proposes a combination of a coarse classifier and a fine classifier to predict an input category by their output probabilities. The author names this coarse and fine combined classifier. The coarse classifier calculates the probability of an input to be in a super category (described in section 6.3) while the fine classifier calculates the output probability for each fine category (character category). The coarse classifier and the fine classifier share the same feature extractor.

To recognize a Nom page, which was written in the vertical direction from top to bottom, left to right. The author first segments the Nom page into character regions, then he groups them into text lines based on their positions. He applies the coarse and fine

combined character recognizer to each character region in a text line and concatenate the recognition candidates for each character into a sequence for the text line. Then, he applies the beam search decoder with a language model of Nom for revising the final recognition results.

1.2. Organization of the thesis

The rest of this thesis is organized as follows. Chapter 2 shows a survey on historical document processing. Chapter 3 presents an over-segmentation method for recognizing Japanese text. Chapter 4 shows the improvement from the previous Japanese handwriting recognition system. Chapter 5 presents a basic system for digital archiving of Nom historical document. Chapter 6 describes Nom document digitalization by deep convolution neural networks. Chapter 7 shows a character attention generative adversarial network for degraded mokkan document restoration. Chapter 8 draws a conclusion and the remaining work of the research.

Chapter 2. Survey on Historical Document Processing

Chapter 3. Segmentation and recognition of handwritten characters

Chapter 4. Improvement by Deep Neural Networks Restructure

In the section, the author describes the methods for off-line Japanese handwriting recognition and the improvement from the previous system. The author perform two main improvements: adding segmentation score and recognition score by CNN based models, and fixedly segmentation by the semantic segmentation.

4.1. Related work

Over-segmentation methods had been popular for handwriting recognition before deep learning based approaches came. The methods start with segmenting the text-line into unconnected components. Some parts of one character pattern may be segmented into small components, so we call them over-segmented methods. Then, pre-trained OCRs are used to recognize separated components and combined components. Some unconnected components are combined together based on the geometric features. The recognition result will be combined with a linguistic context and a geometric context to get final results. Zhu et al. [13] proposed a robust over-segment model for on-line handwritten Japanese text recognition. The method utilizes off-strokes of characters, but it can be applied for off-line handwriting recognition by replacing off-strokes with unconnected components. One of the most important problems in over-segmentation method is the segmentation of touching components. This problem, however, was resolved in the author's previous work in the author's Master course [AP9]. The remaining problems in the over-segmentation method is using geometric features to determine segmentation points (none combination) and non-segmentation points (combination) between unconnected components and over-segmentation causes the reduction of recognition rate when combined with contexts in lattice diagram. The more over-segmentation the system makes, the harder to find best paths. This problem can be resolved by the next two methods.

Sematic segmentation based methods is methods that firmly segment text pages into isolated character patterns, use OCR to recognize each of them and combine with a linguistic context to improve recognition results. Until now there are many method have proposed. J. Long et al. [14] are the first group which proposed Fully Convolutional Network (FCN) for image segmentation. They modify some types of convolution neural network such as AlexNet, VGG16, GoogLeNet to accept a non-fixed size input and replace all the fully connected layers by convolutional layers. They add an up-sampling

layer known as a deconvolution layer to create outputs having the same size as the inputs. The network is trained with a pixel-wise loss function. Skip connections are used to pass features from the convolution layers to the deconvolution layer. The model, however, loses the global context of the image in its deep layers, so W. Liu et al. [15] propose ParseNet that can keep the global information. A global pooling layers is utilized to convert feature maps of an early convolutional layers to a global vector. The vector is normalized by the L2 Norm and un-pooled to feature maps with the same size as the initial ones. Finally, they concatenate them. Their experiment results show that the network can segment objects more stable without suddenly category change pixels as the previous methods. H. Noh et al.[16] use a model that down-samples by pooling layers and up-sampling by un-pooling layers. O. Ronneberger et al. [12] extend the FCN method known as U-Net for segmenting objects in biology microscopy images. The convolution layers and deconvolution layers are center-symmetrical correspondent because each pair has the same size of output feature map. The feature of the convolutional layers is concatenated to its corresponding deconvolution layers to avoid losing pattern information knows as skip connections. The network can be trained with a small labelled dataset. In 2017, K. He et al. published the Mask R-CNN based on Faster R-CNN [17]. Basically, a Faster R-CNN produce two output: bounding box of objects and the category of the object. They add a branch for predicting segmentation masks for each Region of Interest (RoI) to make Mask R-CNN. The mask segmentation branch is a small FCN that predict segmenting class for each pixel on RoIs.

Recently, many character segmentation-free methods have been proposed and proven to be very powerful for text recognition. B. Shi et al. [19] proposed an end-to-end trainable model combining a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) to recognize printed scene text. A. Grave et al. [20] combined Multi-Dimensional LSTM (MDLSTM) and CTC to build an end-to-end trainable model for offline handwritten Arabic recognition. Ly et al. [21] also presented the combination of a pre-trained CNN and an LSTM with CTC for recognizing offline handwritten Japanese text. Besides, an attention-based sequence to sequence model has been successful for recognizing multiple text lines image in Japanese historical documents [22]. The methods can achieve very high recognition in the same style dataset, but they are still unstable for large aspect ratios and shape variability datasets.

4.2. Powerfully segmentation output model

In the previous system, the author proposed a core to fine segmentation method. The coarse stage is for segmenting non-touching and singly touching components and the fine

stage is for multiply touching pairs. After the coarse segmentation and the fine segmentation, all the segmentation candidates are classified by an SVM model into three groups: *S* (segmentation), *NS* (non-segmentation), and *U* (undecided) using seven geometric features. The SVM model is trained so that *S* must include true segmentation boundaries but may include false segmentation boundaries, while *NS* must exclude true boundaries as much as possible. *U* is treated as either *S* or *NS*. *U* provides robustness for recognition but slows recognition speed. *NS* is removed and not considered any more. The *NS* deletion reduces the execution time as well as increases the recognition rate. A segment bounds by two neighbor *S* or *U* boundaries forms a candidate character pattern. Moreover, a sequence of consecutive segments delimited by *U* may be combined to form a candidate character pattern. Concatenation of segments is limited by their total lengths. Each candidate character pattern is recognized by the MQDF based OCR into candidate classes. The combination of all candidate character patterns and candidate classes of each character pattern is represented by a segmentation and recognition candidate lattice as shown in **Figure 4.1**. Here, each path of the diagram is evaluated by combining the scores of the candidate classes with their scores in the linguistic context and the scores of geometric features. Each score is weighted with a parameter and optimized using the Genetic algorithm. Finally, the Viterbi algorithm is used to find the path with the highest score [23], which will become the final recognition result of the text line. That, however is not so strong with the arbitrary in shapes, and distances between individual components.

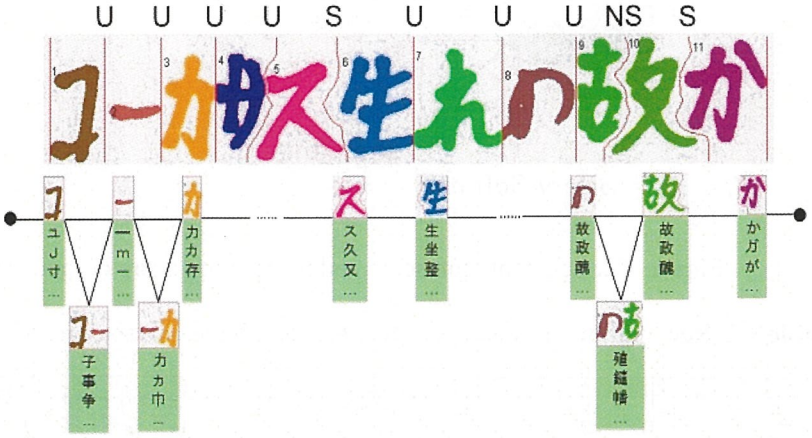


Figure 4.1. Segmentation and recognition candidate lattice.

To improve the recognition rate of the MQDF based OCR and segmentation by SVM, the author first train CNNs model with three data sets as shown in **Table 4.1** to recognize single character patterns. The datasets are combined by all data sets in Nakagawa

laboratory (ETL, iLabo collected data, Font data, HP, Nakayoshi & Kuchibue, NTT). The recognition results for recognizing single characters are shown as **Table 4.2**. The recognition rates of CNN based OCRs are outperformed the MQDR based model because the deep features extracted by convolutional layers are strong compared with the gradient features used in the MQDF model.

Table 4.1. Training dataset for single character recognition OCRs

Datasets	# Categories	#Train Patterns	# Testing Patterns
Jis 1 Kanji (A)	2965	4677681	521959
A and Non-Kanji (B)	3464	5434418	605538
B + Jis 2 Kanji	7214	8233022	914770

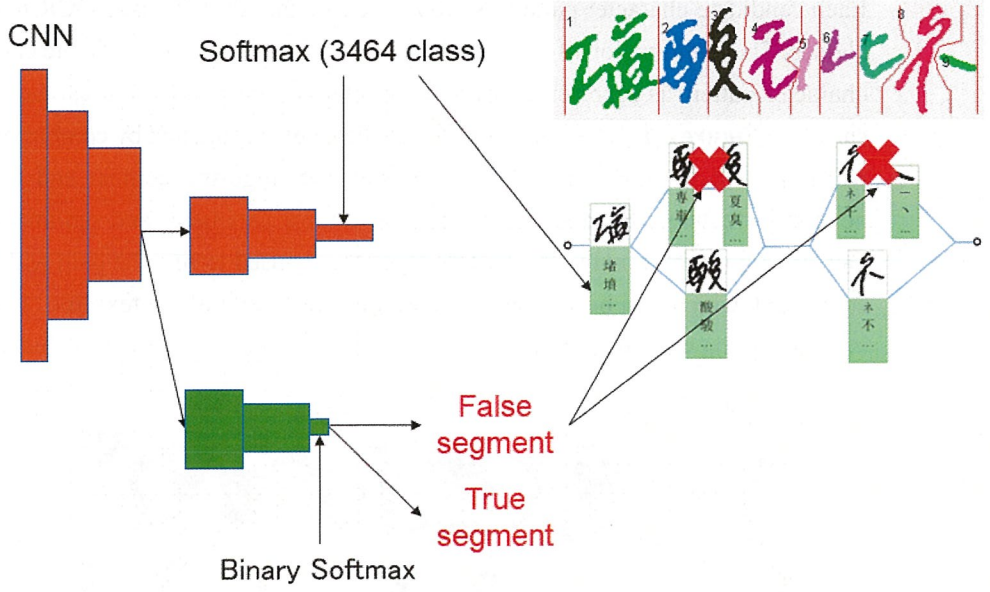


Figure 4.2. Segmentation score and recognition score outputting model

Table 4.2. Recognition rates on the testing sets for single character recognition OCRs

Models	Time execution per pattern	Recognition rate on dataset A	Recognition rate on dataset B	Recognition rate on dataset C
MQDF	3~4ms	96.72	94.11	92.46
VGG16	14~18ms	98.96	98.11	96.93
InceptionResNetV2	31~35	99.24	98.72	98.38

From the fact that CNN based OCRs can produce good scores (probability almost one) for components with the shape similar to patterns existing in the training data of OCR while they yield very bad score for others, the author pre-trained a model to be able to produce two outputs: character segmentation score and recognition score as shown in **Figure 4.2**. The part of model outputting character segmentation score is to replace for segmentation by SVM with geometric features while the remaining part has the role as the MQDF based OCR. To create that model, the author takes the best CNN based model (InceptionResNetV2) for the dataset B, freezes the shallow layer (high level feature layers) and re-train the remaining part of the model to produce segmentation score outputs. The data for training and testing segmentation is extracted from the Kondate text line dataset (90 persons for training and 10 persons for testing). Because the Kondate data set cannot cover 3464 (only 1396 classes), the author also generates text line data from Asahi and Nikkei corpuses and the Nakayoushi and Kuchibue dataset. The author use segmentation part in the previous system to segment the Kondate text lines, generated text lines into unconnected components. If unconnected components are a character pattern, they are put into the TRUE segmentation class. If the unconnected components are under segmentation or over segmentation, they are placed in the FALSE class. Patterns in the TRUE segmentation groups and FALSE segmentation groups are shown in **Figure 4.3**. In the FALSE group, the first column is over segmentation components. (Inside a character patterns); the second column is segmentation patterns that are combined by several character patterns (TRUE components), the third column is segmentation patterns that are combined by several inside character patterns (over-segmentation components). All of them are considered in the previous system in the lattice, however, that makes the best path search heavy and reduce the recognition rate significantly.

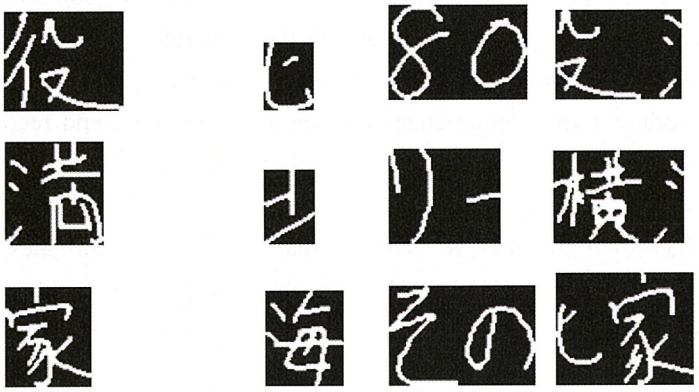


Figure 4.3. Segmentation training data

Table 4.3. Recognition rates on the testing sets for single character recognition OCRs

# train, test components in TRUE segmentation group	# train, test components in FALSE segmentation group	Geometric feature based SVM model (%)	Binary classification rate CNN model (%)
908842, 66674	957852, 15261	91.23	95.79

Table 4.3 shows the binary classification rate for TRUE segmentation patterns and FALSE segmentation patterns. The combined model by CNN is outperformed the geometric feature SVM based models. Moreover, as we can see from the confusion matrixes, while the CNN model produce balanced result between two classes, the SVM model cannot classify FALSE category well.

	F	T
F	10462	2387
T	4799	64287

SVM model

	F	T
F	14676	2865
T	585	63809

CNN model

Figure 4.4. FALSE and TRUE segmentation classification

The author finally uses the two output CNN to replace for MQDF OCR and segmentation based SVM and geometric context and fine-tunes parameters for the whole model. The recognition of the system improve significantly as shown in Fi. The character error rate and sequence error rate are used the same as in

Table 4.4. Recognition rate of previous system and improve system

Model	Character error rate	Sequence error rate
MQDF + geometric feature SVM (previous system)	81.32	46.28
Recognition and segmentation score outputting CNN	88.98	54.72

4.3. Semantic character segmentation

Although the CNN based model with two outputs improves the recognition rate of the over segmentation method significantly, the improved system is still far from demands compared with the free-segmentation method. In this section, the author describes the semantic character segmentation method that is expected to produce the better recognition rate than free-segmentation method.

There are many methods for semantic character segmentation as describe in the related work section, but in the author has just completed the U-Net based method as the character extraction for Nom document presented in Chapter 6. The encoder of the U-Net model is ResNet50. The U-Net is to classify each pixel in text line images into three classes: center marker of character patterns, convex-hull of character patterns and fore-ground pixels as shown in **Figure 4.5**. The author makes ground truth for two data set:

Kondate data set (90 persons for training), generated text line data from Asahi&Nikkei corpus (90% for training).

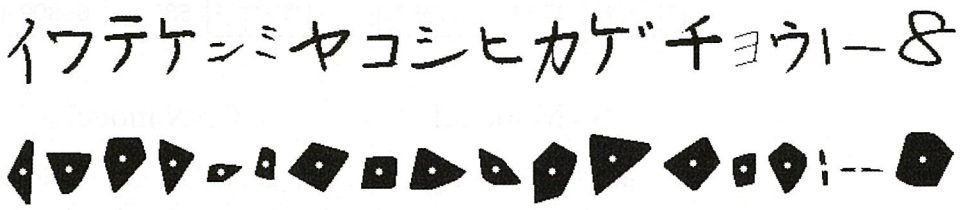


Figure 4.5. Ground-truth for training semantic segmentation method

The testing results for 10 persons in the Kondate dataset and 10% of the generated dataset are shown in **Table 4.5** . The *IoU* matrix is defined in Chapter 6. The predicted result for each image sometimes includes touching convex hull bounding box, but they can be separated by the Watershed method with markers are center points of convex-hull regions.

Table 4.5. Testing results of U-Net for semantic character segmentation.

	Pixel level accuracy	IoU
Kondate testing set	99.65	99.21
Generate testing set	99.87	99.43

Chapter 5. A basic system for Nom historical document recognition for digital archiving

Chapter 6. Nom Document Digitalization by Deep Neural Networks

6.1. Related work

Usually, there are three approaches for digitalizing documents to text: the segmentation-based approach, the text location and recognition approach and the end-to-end text sequence recognition approach with Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). In the segmentation-based approach, documents are segmented to character regions, and an OCR is applied to recognize them. In the text localization and recognition approach, CNNs are used to extract feature maps of input images and character regions are proposed on the maps by a Region Proposal Networks (RPN). Then each character region is recognized by a classifier. In the third approach, the feature maps extracted by CNNs can also be fed into RNNs and a transcription decoder is used to decode time step outputs of the RNNs to text sequences. The second approach usually fails when the number of classes is more than ten thousand and both of the second and third approaches require a large number of patterns for training. They are also very sensitive to the change of handwriting styles. In this paper, we focus on the first approach, but we also employ a transcription decoder with a language model for correcting the recognition results.

In the segmentation-based approach, text pages are segmented into text lines based on projection profiles, the Hough transform, smearing [38] and then separated into single characters by analyzing the fore and background of text line images [39]. Another approach is based on local features [40]. The authors generated template images and extracted SIFT features for multiple-size sliding windows and matched to template images. A voting and geometric verification algorithms are used to decide final results. Baek et al. (2019) applied the U-Net based network for character region detection. They utilize not only character regions but also the affinity between characters [41]. Due to the lack of real training data, they trained the model for synthetic images first, then used it to estimate character regions for real images.

As for large category classification, many methods group objects into coarse categories before classifying them into fine categories, which is known as coarse-to-fine classification. Cevahir et al. (2016) combined deep belief nets and deep auto-encoder neural network models to firstly classify large-scale e-commerce data into five super classes, and then classify them into 28,338 fine categories [42]. They utilized all textual contents such as titles and descriptions but ignored image contents of products. Yan et al.

(2015) applied a hierarchical deep convolution neural network (HD-CNN) for large scale visual recognition [43]. They first pre-trained a CNN model for a fixed number of coarse categories (5, 9, 14 and 19). Then, they pre-trained a model for each coarse category. After both the coarse category network and the fine category networks for the coarse categories were properly pre-trained, they fine-tuned the complete HD-CNN. Because the number of parameters in the fine category networks grew linearly with the number of coarse categories, they compressed the parameters of the fine category networks by K-mean clustering. Pre-training a large number of fine category networks, however, is time-consuming, so that it is hard to apply for recognizing Nom.

Jie et al. (2017) considered coarse categories and fine categories for weakly supervised learning [44]. They had more training data labeled with coarse categories but fewer data labeled with fine categories. From such training data, they trained a model that could classify a new image into one of the fine categories. This is not the combination of a coarse classifier and a fine classifier, but we are inspired by the method of how coarsely labeled data can help fine label classification.

As for text-sequence decoding, there are many text sequence decoding algorithms proposed so far. The basic algorithm is best path decoding that takes the most likely candidate at each time step. Graves et al. (2009) proposed the token passing algorithm to search the most probable sequence from the output matrix in a dictionary word [45]. The algorithm constrains the output to a context dictionary, so it cannot handle arbitrary character sequences. In 2012, he also introduced the beam search decoding method that integrated a language model for arbitrary character sequences [46]. The beam search algorithm expands all possible next steps and keeps the k most likely, known as the beam width.

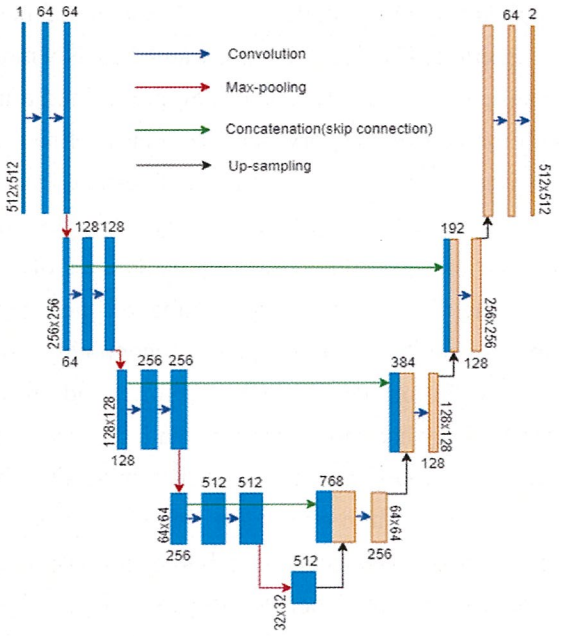


Figure 6.1. Character region extraction network with the VGG-16 decode.

6.2. Character extraction method

To extract character images, we follow the structure of U-Net to classify each pixel in Nom pages into background pixels and pixels in character regions. The architecture of the network includes an encoder to capture context and a symmetric expanding decoder to relocate precise locations. The network learns to predict the character regions as polygons instead of rectangles so that it reduces the touching between character segmentation results.

Although we follow the U-Net architecture, we consider a few different encoders such as VGG-16, Resnet 50 or Inception-ResNet V2 to down-sample training images to the small sizes of feature maps which contain the information of character regions. Assuming the encoder here is a VGG-16 based structure, the corresponding decoder up-samples the feature maps and produces the segmentation maps. **Figure 6.1** shows the character extraction network, including the down-sampled feature maps (boxes in sky-blue) by the encoder, the up-sampled feature maps (boxes in yellow) by the decoder, and the network operations (colored arrows). The input size of the network is set to 512×512. Following each convolution layer is batch normalization layer, which computes the mean and the standard deviation of all the output feature maps and then normalizes them. That makes

all feature maps have the same range and zero mean, so that helps the training process of the next layer not have to learn offsets of data, which is known as the covariate-shift problem. The information of the former convolutional layers in the encoder is passed to the up-sampling layers in the decoder, which can avoid the vanishing gradient problem. In detail, the features by each multi-level down-sampling layer in the encoder are concatenated to those of the same size by the corresponding multi-level up-sampling layer in the decoder as shown in the green arrows in **Figure 6.1**. This concatenation is known as the skip connection.

Character regions produced by trained models, still sometimes touch each other, especially at the confusing boundary pixels of convex-hull character regions. The produced segmentation maps are finally segmented by the marker-based watershed algorithm. Using the erosion morphology, we can locate the sure centers of character regions (markers). We apply the watershed algorithm with decided markers to segment touching character regions.

6.3. Coarse and fine combined classifier

We present a novel architecture of combining a coarse and fine classifier for a huge number of categories (denoted by CF_Combined). We apply a coarse classifier to classify an input pattern into a super category and a fine classifier to classify it into a particular class known as the fine category. Then, we multiply the coarse category probability and the fine category probability to predict the fine category probability. The coarse classifier is also used to guide the backpropagation for training the fine classifier. We will show that this architecture is better than just a single fine classifier for a large category set or a simple sequential architecture of the coarse-to-fine classifier.

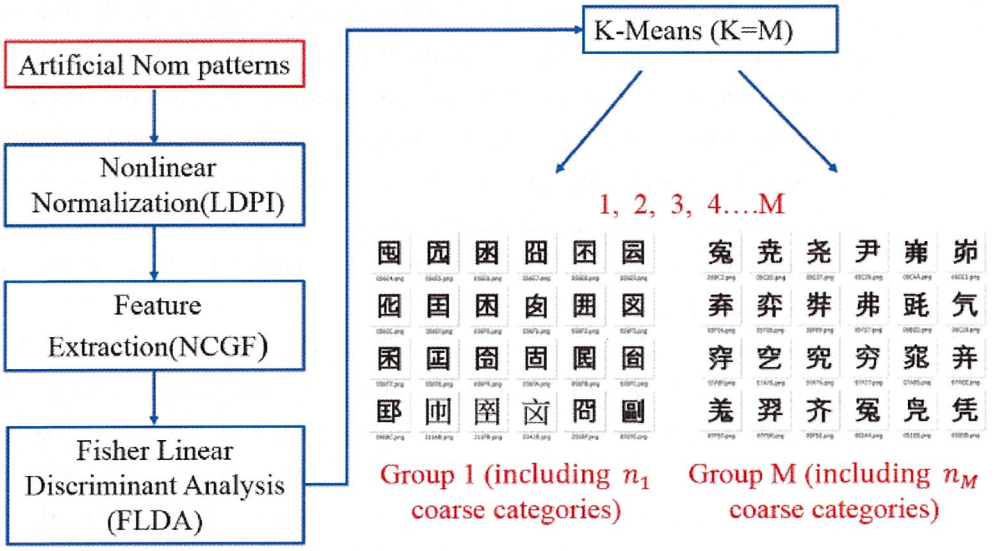


Figure 6.2. Coarse category formation.

6.3.1. Coarse category formation and category labeling

When Truyen et al. (2016) made the first attempt to make Nom OCRs, they pointed out that Nom script includes at least 32,695 categories based on studies on Nom fonts and publications from the Vietnamese Nom Preservation Foundation [11].

To create super groups (called coarse categories), we group the 32,695 Nom categories (called fine categories) by the K-means algorithm. Since Nom is composed of Chinese characters or their radicals, we consider the number of clusters K to be more or less of the number of radicals in Nom, i.e., 304 [10]. However, if we group Nom categories by radicals, some characters in a group have very complicated structures while others have simple structures, with the result that coarse category classification may not correspond to the radical groups. Therefore, we use just this number to guide us to find the best number.

We follow the same process for feature extraction as the previous work [11]. We normalize Nom character patterns by non-linear line density projection interpolation (LDPI). Then, we extract directional features of 512 dimensions by the normalization-cooperated gradient feature (NCGF) method. To make features be more discriminative so that the clustering process by K-means is more effective, we reduce the dimension of original features to 160 by the Fisher Linear Discriminant Analysis (FLDA). The clustering process to create coarse categories is shown in **Figure 6.2**.

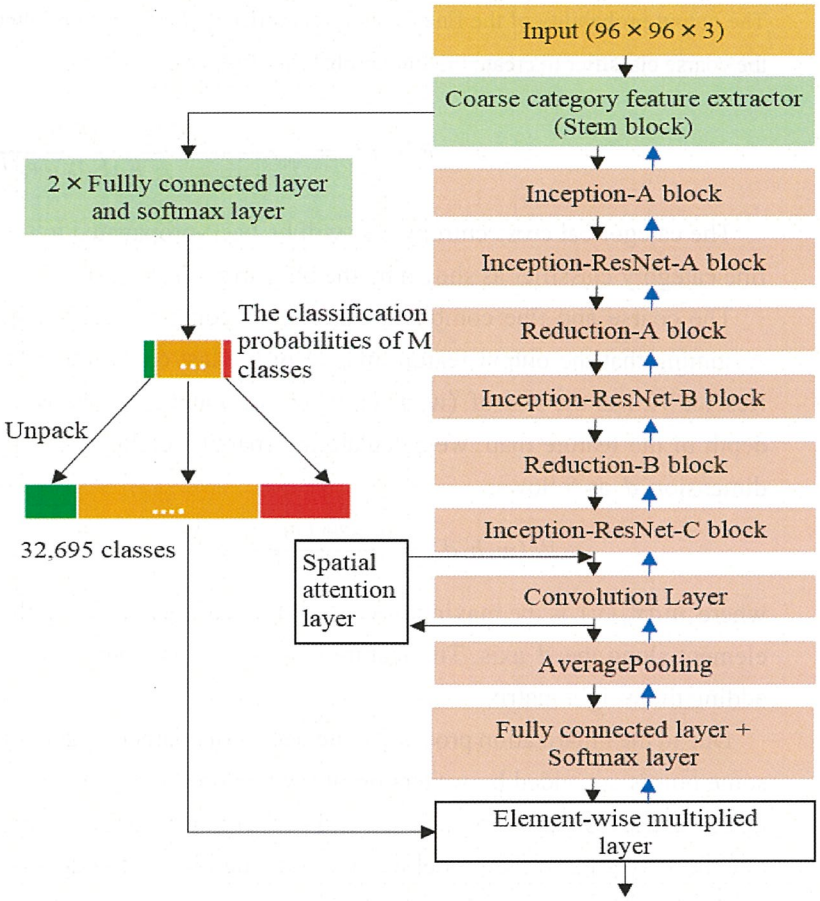


Figure 6.3. Coase and fine combined classifier.

6.3.2. Coarse and fine combined classifier

The stem block in the Inception-ResNet V2 [33] which is a VGG-16 liked structure, is combined with two fully connected layers and a softmax layer to form the coarse classifier as shown in the green color blocks in **Figure 6.3**. The probability of an input pattern (x) assigned to a coarse class $C_m, m \in \{1, \dots M\}$, by the coarse category classifier after the softmax layer is $P_c(C_m|x)$. The probability of a fine category $F_n, n \in \{1, \dots N = 32695\}$ corresponding to the coarse category C_m is $P_c(F_n|C_m, x)$:

$$P_c(F_n|C_m, x) = P_c(C_m|x) \; if \; F_n \in C_m \tag{1}$$

The coarse category feature extractor is frozen its weights and is used to extract features for the fine category classifier as shown in the carrot color blocks in **Figure 6.3.** The output probability of the fine category classifier $P_f(F_n|x)$ is multiplied with the output from the coarse classifier to create the final probability $P(F_n|x)$:

$$P(F_n|x) = \frac{P_f(F_n|x) * P_c(F_n|C_m,\ x)}{\Sigma_{n=1,m=1,F_n \in C_m}^{N,M} P_f(F_n|x) * P_c(F_n|C_m,\ x)} \quad (2)$$

The categorical cross entropy loss will be back-propagated to the shallow layer in the fine category classifier as shown by the blue arrow in **Figure 6.3.**

The coarse and fine combined classifier is combined with a spatial attention layer. Assuming that the output feature map y of the last convolution layer in the Inception-ResNet V2 has the size of (w, h, d), where w, h and d are the width, the height and the depth of the feature map, we calculate the spatial weight matrix W along feature deep dimension d as follows:

$$W(w, h, d) = \left[\frac{e^{y(w,h,d_k)-max_d(y)}}{\Sigma_k e^{y(w,h,d_k)-max_d(y)}}\right] \quad (3)$$

where $max_d(y)$ is the maximum value of the feature map along the axis d, d_k is the k^{th} element along the d axis. The feature map of the last convolution layer is weighted by adding the weight matrix.

Due to the binarization process, some details of character patterns have vanished while some noises are added by writers or later when preserving. To make the training dataset noised as the testing set, we add salt and pepper noises to the training images. The spatial attention layer helps the model to pay more attention to the details of character patterns rather than the noises on character regions.

6.4. Beam Search decoder with a language model

To find the best path through the sequence of recognition candidates for each character pattern in a text line, we apply a beam search decoder with a language model to decode the text. We use the unigram and the bigram for the beam search decoder. Assuming that $P(c_1)$ is the unigram probability of the candidate c_1 and $P(c_2|c_1)$ is the bigram probability of candidatec_2 , the probability $P_{LM}(c_1,c_2 \dots c_n)$ of a candidate sequence $c_1, c_2 \dots c_n$ is shown in the following equation:

$$P_{LM}(c_1,c_2 \dots c_n) = P(c_1) \times P(c_2|c_1) \times P(c_n|c_{n-1}) \quad (4)$$

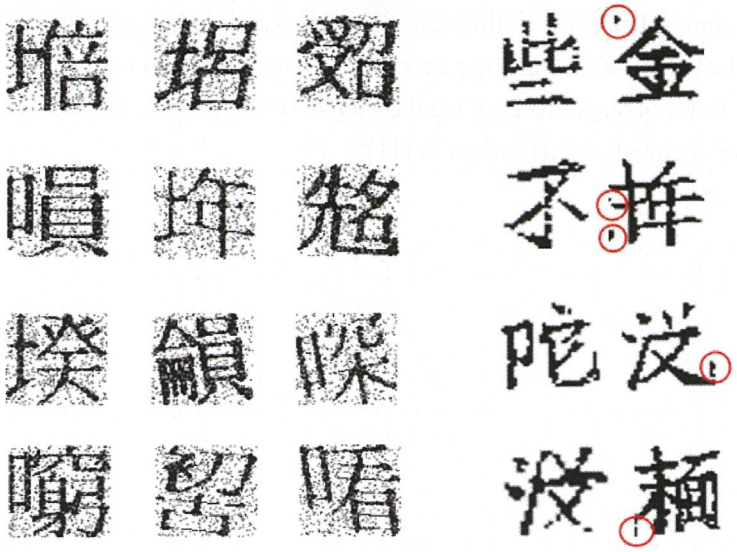
The score $P(b)$ of the beam b will be multiplied with its context probability $P_{LM}(b)$ to create the overall score P and take the best sequence:

$$P = P(b) \times P_{LM}(b) \quad (5)$$

6.5. Experiment

In this section, we present the training and testing datasets for the OCRs and the character region extraction models. Then, we show their performances on the prepared datasets. The results of the beam search decoder with a Nom language model is also shown at the end of this section.

6.5.1. Training dataset and testing dataset



(a) Artificial patterns for training (b) Real patterns for testing.

Figure 6.4. Training and testing patterns for OCRs.

To train the OCRs for single characters, we prepared artificial Nom character patterns since we do not have a sufficient number of real patterns. We made about 1000 artificial patterns as shown in **Figure 6.4** (three left columns) for each fine category, using image deformation methods of the affine, rotate, shear, shrink, and prospective methods from 27 Nom fonts such as Nom Na Tong, Nom Khai, Nom Minh and so on. In total, we have 28,035,360 artificial training patterns for 32,695 categories. We also added salt and pepper noises to the above-generated images to make training patterns, which is the key factor for improving the recognition rate on the testing dataset as analyzed later.

To train the models for character region extraction, we create empty page images of fixed size: 512×512, choose character patterns in the 28,035,360-character dataset randomly and paste the patterns in different scales to the page images as shown in **Figure 6.5**. We find a convex hull boundary of each character pattern to make its ground-truth. The convex hull boundaries are to reduce the touching between characters. In fact, the

testing Nom documents do not have many touching cases, but they are written closely. The rectangle bounding box may make the touching among regions of characters. In total, we generate 58,150 Nom page images and their ground-truths. We separate them in a ratio of 0.8:0.2 for training and evaluating, respectively.

The testing dataset for the proposed OCRs was collected from 47-page images from ten real Nom documents which include 13,111 character rectangle bounding boxes in which 11,669 patterns are decoded to 2,538 categories as shown in **Figure 6.4** (two right columns). The 2,538 testing categories are scattered in the 32,695 training categories. We collected these real testing patterns by segmenting them based on projection profiles and the Voronoi diagram, applying the MQDF+GLVQ OCR, and then inspecting and revising errors manually as described in [11].

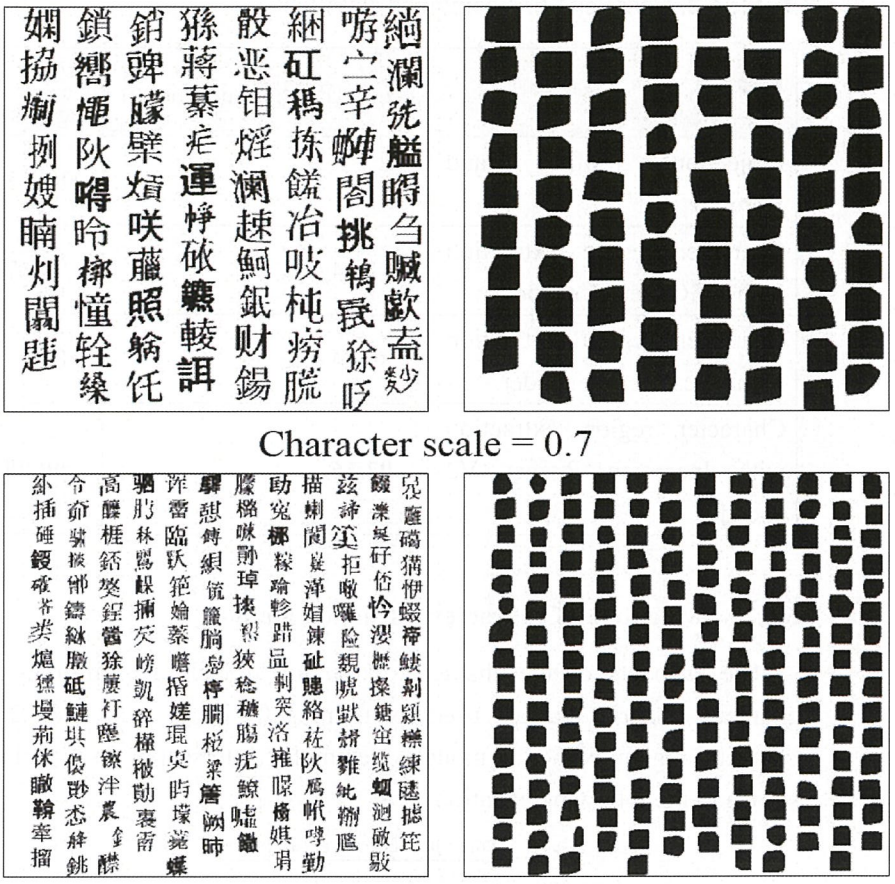


Figure 6.5. Training patterns from different scale characters and convex-hull ground-truths.

Since pages in a single document usually have similar backgrounds, we choose one page in each document for fine-tuning the character region extraction models which have been trained with the artificial dataset. We use the remaining 37 pages for evaluating the performance of the character region extraction models.

Table 6.2. Coase and fine combined classifier.

Methods	*IoU* (Polygonal regions)	*IoU* (Rectangular regions)
Projection profiles and Voronoi	–	81.23
Character region extraction with VGG Net 16 encoder	81.14	79.67
Character region extraction with ResNet 50 encoder	86.54	83.17
Character region extraction with Inception Resnet V2 encoder	**92.16**	**90.08**

6.5.2. Evaluation of character extraction method

The encoders of the character extraction models are pre-trained with generated character patterns that are used for training OCRs in section 6.5.3. To evaluate the character region extraction models, we employ Intersection over Union (*IoU*) metric as shown in the following equation:

$$IoU = \frac{Total\ pixels\ in\ overlap\ areas}{Total\ pixels\ in\ union\ areas} \quad (6)$$

Because the segmentation regions of the method in [11] are rectangular boxes, after getting the polygonal regions of character we also calculate the IoU metric on the rectangles of the polygonal regions and the ground-truths on 37 Nom real pages. The character region extraction models are trained in five epochs with the batch size of 2 and the Adam optimizer [48]

Table 6.1 shows the pixel level accuracies and the IoU metrics of the character region extraction with the different encoders on the evaluating dataset. Since the deeper encoders can take more information about the shapes of characters, they have produced better results.

Table 6.2 shows the character extraction results by different models on the real Nom dataset of 37 pages. The performance is reduced on the real dataset compared to the evaluating dataset. Increasing the number of scales of characters on Nom pages may help to improve the results. The second column is the evaluation in the polygonal ground-truths while the third column is in the rectangular regions. The applied method is almost ten percent point better than the projection profile and Voronoi diagram based method.

At the end of this paper, we show some segmentation results on real Nom pages. The first column is the segmentation by the combination of the projection profile and the Voronoi diagram. The second column is the polygonal segmentation maps by the fine-tuned models, and the last column displays the character regions after applying the marker-based watershed.

6.5.3. Performance of proposed OCRs

Table 6.3 shows the training accuracy, the top-one accuracy, and the top-ten accuracy on the testing set. CNN-based models such as VGG-16, Inception-ResNet V2, CF_Combined with M = 100, 304 and the model with the spatial attention layer were trained in ten epochs, with the Adam optimizer [48] using the batch size of 128. MQDF+GLVQ, VGG-16, Inception-ResNet V2, CF_Combined (M = 100, 304) models are trained with generated patterns without noises. The proposed CF_Combined (M =100, 304) produced the best top-one accuracy, outperformed the single fine classifiers of VGG16 and Inception-ResNet V2. CF_Combined models also outperformed the coarse-to-fine model of MQDF+GLVQ. The model by MQDF+GLVQ was over-fitted so that it could not produce a good result for the real dataset. Previously, we trained VGG-16 without the batch normalization but the network could not converge. Therefore, we used the batch normalization for the VGG-16 model. The three CF_Combined models which use coarse classifiers to guide the training for fine classifiers achieve the better recognition rates than any single fine category classifiers of VGG16 or Inception-ResNet V2 after ten epochs. In the proposed architecture, the coarse classifiers also help the fine

Table 6.3. Training accuracy and testing accuracy of OCRs.

OCR models	Training accuracy (%)	Top-one testing accuracy (%)	Top-ten testing accuracy (%)
MQDF+GLVQ	96.36	69.08	86.03
VGG16	96.47	76.09	91.42
Inception-ResNet V2	97.53	79.93	92.13
CF_Combined (M=100)	97.89	82.15	94.17
CF_Combined (M=304)	**98.16**	82.26	94.04
CF_Combined (M=304) with the attention layer and trained with noised data (BEST)	97.62	**85.07**	**94.76**

Table 6.4. Beam search decoder for the output sequences of the best CNN model.

Beam width	BEST model	Total time execution (s)
k=5	85.18	177
k=10	85.22	191

classifiers converge quickly as shown in **Figure 6.6**. Moreover, the CF_Combined model (M = 304) with the added attention layer and trained with the noised patterns improved the top-one accuracy on the real test set. The improvement on the test set rather than the training set shows that the attention layer and added noises help the network reduce overfitting. Without adding noises to the training data, the CF_Combined model (M = 304) with the attention layer does not show its effectiveness. The green and red lines in **Figure 6.6** show the losses of the CF_Combined (M=304) and the CF_Combined model (M = 304) with the attention layer, trained without noised patterns, in which the training losses of two models are almost similar. We achieved the best recognition rate of 85.07% for Nom on the real testing dataset by the CF_Combined model (M = 304) with the added attention layer and trained with the noised patterns (called BEST model).

6.5.4. Evaluation of the performance of the beam search decoder

The language model is built from a dictionary of 26,063 Nom characters in the famous Nom poem named "the tale of Kieu". The content of experimented text-pages is not included in the language model. We set the beam width to 5 or 10. **Table 6.4** shows the improvement of the beam search decoder combined with the language model for the recognition rate of the best proposed OCR The increase of the beam width improves the recognition rate a little, but it also incurs a longer time for the decoding process.

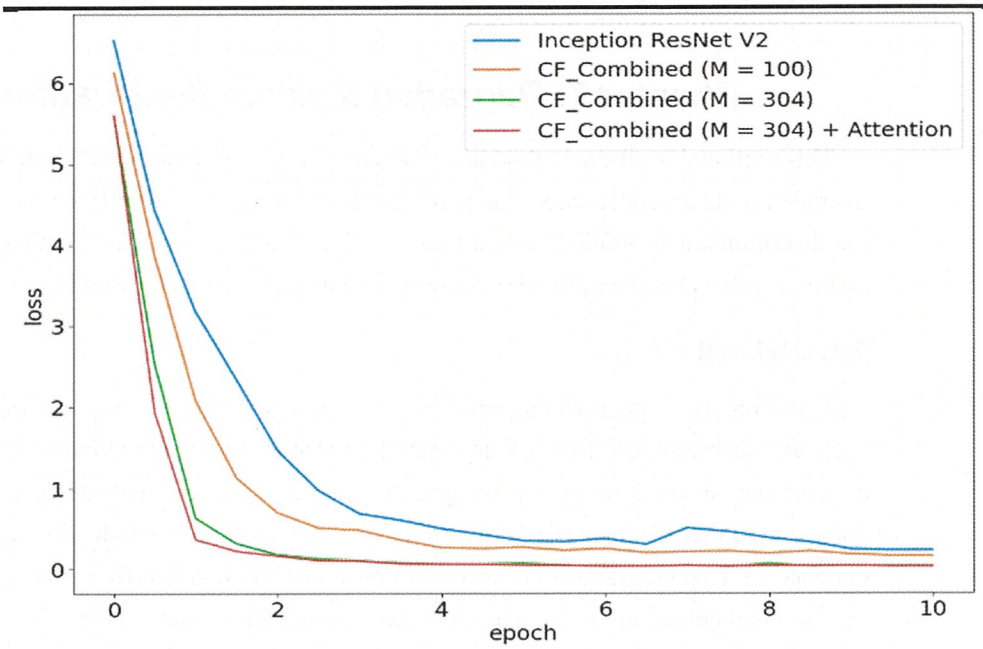


Figure 6.6. Training losses of classifiers.

Chapter 7. Degraded Mokkan Restoration

This section describe a method based on Generative Adversarial Network for restoring degraded mokkan documents. The generator in the model is U-Net based network while the discriminator is VGGnet based network. The model is trained with adversarial loss and perceptive character attention losses calculated at three deep level layers.

7.1. Related work

Traditionally, many methods have been proposed to enhance the readability of images such as the brightness/contrast adjustment methods, the binarization methods which convert the image into binary images, the noise reduction methods and so on. For enhancing mokkan images, Takakura et al. [24] used noise reduction and contrast enhancement techniques, some thresholding and Gaussian blurring methods, and the similar pixel detection on the deterrent color spaces. Those methods may be effective for archeologists to read characters on mokkans, but still hard for OCRs to recognize those characters.

Recently, GANs have become popular for image restoration. They are deep neural net architectures comprised of two nets: generator (G) and discriminator (D) which pit one against the other (adversarial) [25]. The discriminator tries to classify an input, i.e., given the features of the input, it predicts a label or category to which the input belongs. On the other hand, the generator attempts to predict the features given a certain label. In short, the discriminator learns the boundary between classes while the generator models the distributions of individual classes. We assume that z is a noise vector with the distribution of $P_{noise}(z)$ and x is a real data variable with the distribution $P_{data}(x)$. G generates samples from z while D aims to distinguish between samples from the real data distributions and the generator's distributions. The training process is a min-max game as the following expression:

$$min_G max_D\, V\,(D,G) = E_{x\sim P_{data(x)}}\,[log\, D(x)] + E_{z\sim P_{noise}(z)}\Big[log\Big(1 - D\big(G(z)\big)\Big)\Big] \quad (7)$$

Image restoration can be broken into three sub-problems: de-noising, super-resolution, and in-painting. Yang et al. [26] proposed a method using a GAN with the Wasserstein distance and the perceptual loss to de-noise CT images. Ledig et al. [27] proposed a GAN for generating the super-resolution images from low-resolution images. Each training sample was a pair of a low-resolution image and a high-resolution image. They utilized residual blocks to build the generator. The training loss was combined by the usual adversarial loss and the content loss by VGG net which was trained with the ImageNet data set [27]. Yu et al. [29] employed a GAN with contextual attention to reconstruct images with lost parts of the white masks. They introduced a two-stage coarse-to-fine

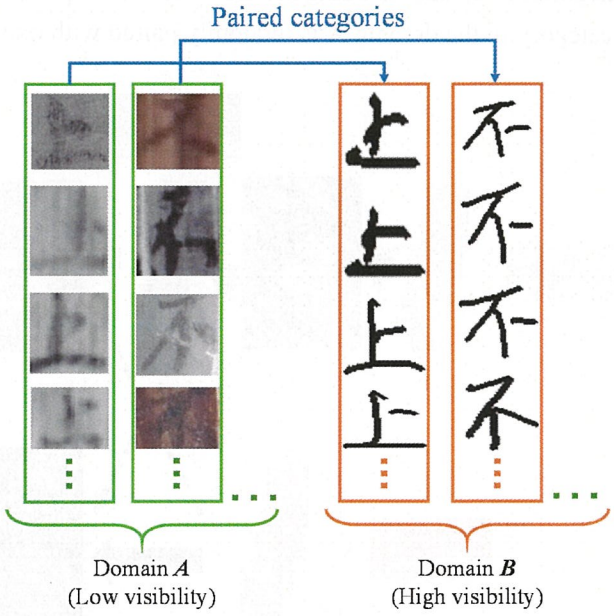


Figure 7.1. Weakly paired domain training.

network architecture where the first stage makes an initial coarse prediction, and the second one takes the coarse image with refined results. In the fine reconstruction stage, they proposed a contextual attention layer which helps the fine network learn to borrow or copy feature information from known background patches to generate missing patches.

7.2. Proposed method

We propose a character attention generative adversarial network (CAGAN) to create high visibility images I^{HV} from severely degraded or low visibility input images I^{LV}. To do that, we separate available input images into two spaces: one domain includes degraded mokkan patterns (domain A) and other consists of high visibility patterns from the TUAT Kuchibue and Nakayosi datasets (domain B) [30]. These datasets are sets of single online Japanese character patterns, but we transformed them into offline patterns. The Kuchibue dataset, written by 120 writers, includes 3,356 categories while the Nakayosi dataset, written by 163 writers, contains 4,438 categories. We limit the number of categories to 2,965, including the most common Japanese Kanji characters. Each image in a category in the domain A is randomly paired with one of its corresponding category

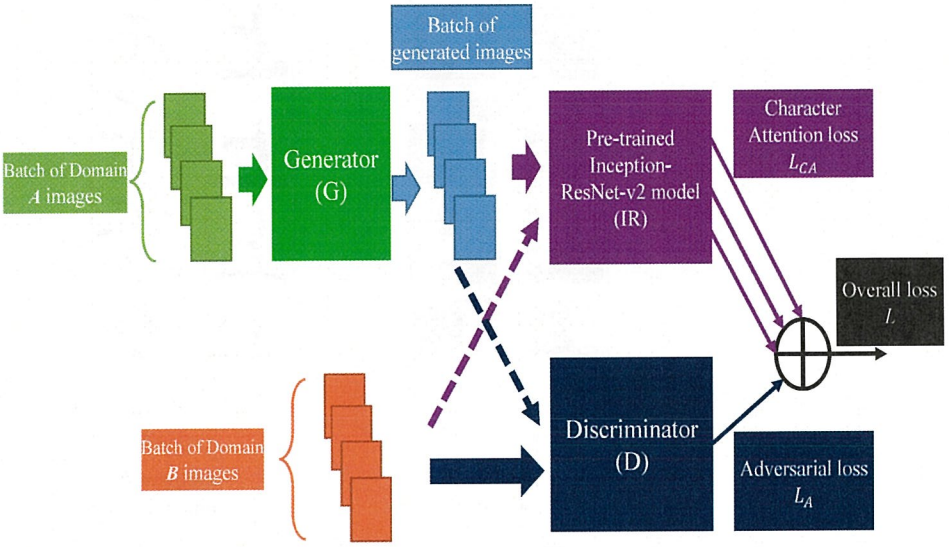


Figure 7.2. CAGAN architecture.

in the domain B. We call the training method "weakly paired domain training" as shown in **Figure 7.1**. The architecture of CAGAN is shown in **Figure 7.2**, including the generator and the discriminator. The generator takes batches of degraded images in domain A and the discriminator is fed with batches of images in domain ***B***. A pre-trained Inception-Resnet-v2 model by the TUAT Kuchibue&Nakayosi-mixed dataset of 867,475 patterns for 2,965 categories is to calculate the character attention loss at the three different depth levels of convolution layers.

7.2.1. Generator (G)

The generator is built like the U-Net structure with skip connections as shown in **Figure 7.3**. The strides of convolution layers in the encoder are 2 while those in the decoder are 1. Skip connections are to pass information of the former convolutional layers to the de-convolutional layers which can help to avoid the vanishing gradient problem. In detail, we concatenate the input from a down-sampling layer to its center symmetrical up-sampling layer. Each down-sampling block consists of a convolution layer with the leaky ReLU activation function and the batch normalization layer while an up-sampling block includes a de-convolutional layer, a convolution layer without down-sampling, and a batch normalization layer.

7.2.2. Discriminator (D)

The discriminator is constructed by seven down-sampling blocks like in the generator and finally followed by two fully connected layers. Normally, a discriminator is ended by a one-unit fully connected layer with sigmoid activation, representing the probability of the input sample to be real (values in [0, 1]). However, this likely causes the gradient vanishing problem. As the advantage of the WGAN [31], we build the output as linear without any activation function. We train the discriminator with the Wasserstein loss

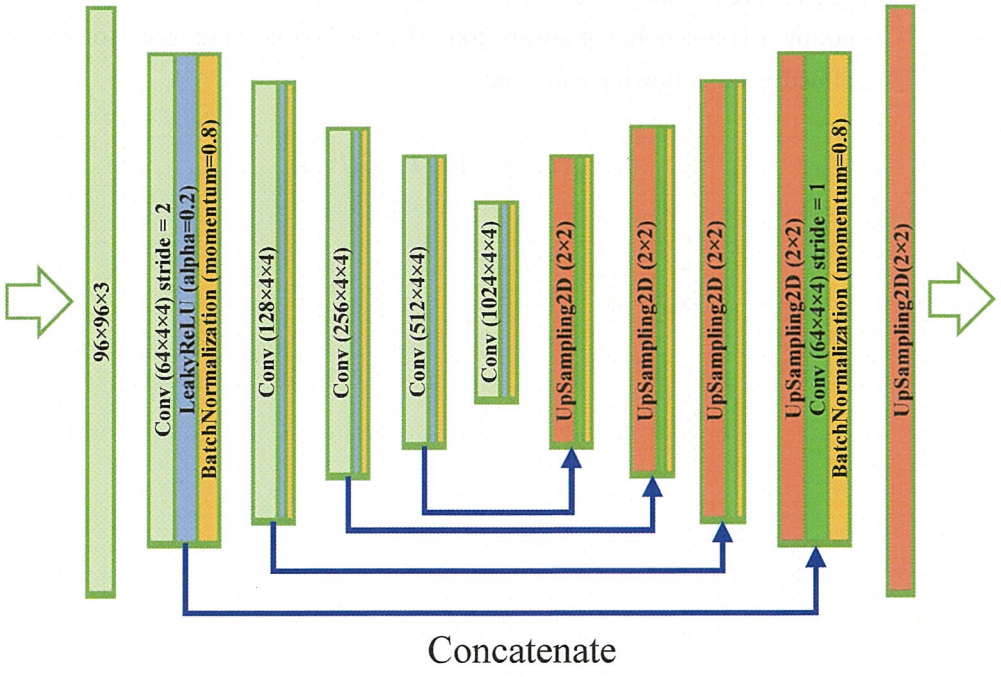


Figure 7.3. U-Net architecture of the generator with skip connections.

function. We assume that the output of the discriminator is y_{pred} and the target is y_{true}. Those two terms (y_{pred} and y_{true}) are vectors with the size of the training batch size with each element having a value in [-1, 1]. The generated images are labeled -1 while the real images are labeled to 1. The Wasserstein loss function is simply defined as the following equation:

$$D_{loss} = mean(y_{pred} * y_{true}) \tag{8}$$

where $*$ is the element-wise multiplying operation. The loss makes the discriminator to maximize the distances between its outputs for real patterns and those for generated samples as large as possible.

7.2.3. Training loss

The training loss L is combined by the common adversarial loss L_A and the character attention loss L_{CA}. The adversarial loss L_A is to encourage the generator to fool the discriminator as much as possible and help the generator to transfer global features of the training domain, such as background, locations of characters and the shape of characters, to the target domain, while the character attention loss L_{CA} forces the generator to learn the detail features of character patterns, so the generator can reconstruct the character patterns even if they miss some strokes. The adversarial loss L_A is the binary cross entropy loss function, but always trained with the true class label, so the loss function is defined as the following equation:

$$L_A = -log\, P_D(G(i^{LV})) \tag{9}$$

where $G(i^{LV})$ is a generated image by the generator and $i^{LV} \in I^{LV}$ is a low visibility image in domain $\boldsymbol{A}$, P_D is the probability that the discriminator D thinks an input image to be real.

To build the character attention loss function, we train the Inception-ResNet-v2 model [33] by the training data set in domain $\boldsymbol{B}$, which includes 2,965 of the first standard Shift-JIS single characters. This model needs less computation cost and a smaller size than VGG net or others, but it can extract very deep features.

As convolution neural networks can extract less complex features at the shallow layers and more complex features at the deeper layers as shown in **Figure 7.4**, we calculate the character attention loss at the three deep levels of the Inception-Resnet-v2 network. The first loss is calculated at the concatenation layer of the inception-A block (C_1), the second loss is computed at the concatenation layer of the reduction-A block (C_2) and the third loss is at the last layer of the inception-ResNet-B block (C_3). We assume $IR_j(i)$ is the feature map of an image $i \in I$ by the j^{th} layer of the Inception-Resnet-v2 model. The character attention loss function is defined by the following equation:

$$L_{CA} = \sum_{j \in \{C_1, C_2, C_3\}} IR_j(i^{HV}) * log \frac{IR_j(i^{HV})}{IR_j(G(i^{LV}))} \tag{10}$$

This is the Kullback-Leibler divergence which is a measure of how the probability distribution of the feature map with the input i^{HV} is different from the probability distribution of the feature map with the generated image $G(i^{LV})$. Traditionally, some

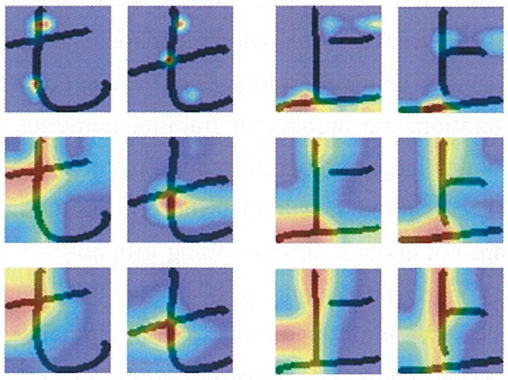


Figure 7.4. Heat map of the features extracted at the various depth layers of the Inception-ResNet-v2 model [32]. The top images are features of character patterns, extracted at the shallow layers of the Inception-ResNet-v2 while the bottom images show the features at the deep layers.

methods which introduce context losses or texture losses usually utilize the Mean Square Error (MSE) or the Mean Absolute Error (MEA) functions to calculate the loss between two feature maps. These loss functions are associated with over-smoothed textures since they are element-wise loss functions, so they are just effective in cases of paired pattern training such as problems [27], [34], [35]. On the other hand, the Kullback-Leibler divergence tries to minimize the general difference of the distribution between two feature maps.

The overall loss function is the sum of the weighted adversarial loss and the character attention loss as the following equation:

$$L = w * L_A + L_{CA} \tag{11}$$

where w is the weight, set to 10^3. We selected it experimentally. L_A is more important than L_{CA}, because it decides the overall features of generated images. We can set it to 10, 100 or others, but in our experiment, w = 1000 makes the model converge quickly and satisfies trade-off between the background features and character features.

7.3. Experiment

From the home page of the NNRICP, we collected mokkan character images of the 118 most common categories. On average, each category has about 180 images (the smallest category has just 42 images while the largest category has 1,454 images). Totally, we collected 23,193 mokkan character images (domain $\boldsymbol{A}$). We separated the dataset of each category in the domain $\boldsymbol{A}$ into two parts: one for training (20%) and the other for testing (80%). Because we randomly choose n images (batch size) of a category in domain $\boldsymbol{A}$ and n counterpart images of the category in the domain $\boldsymbol{B}$ to feed to the model at each iteration, the number of training iterations is so large. As the result, we can dispense with a large number of training patterns, which is the advantage of the proposed method. The method is suitable for processing historical documents where sample patterns are not enough and providing ground-truth is much more costly than ordinary documents since characters in historical documents are difficult to read even by archeologists. We limit the number of training iterations to 40,000.

In the experiments, we prepared two OCRs. The first OCR called Domain-$\boldsymbol{B}$-OCR, which is the Inception-ResNet-v2 trained by the whole TUAT Kuchubue&Nakayosi-mixed dataset. The second OCR called Domain-$\boldsymbol{B}$-to-$\boldsymbol{A}$-OCR, which is the Domain-$\boldsymbol{B}$-OCR then fine-tuned on domain $\boldsymbol{A}$ training samples. We test if the generated test images from the test set of domain $\boldsymbol{A}$ can be recognized by Domain-$\boldsymbol{B}$-OCR. We also shows the performance of Domain-$\boldsymbol{B}$-to-$\boldsymbol{A}$-OCR on original test samples.

The CAGAN model is trained with the batch size of 32 and the learning rate of 0.0002, used the Adam optimizer [36] . In the standard GAN, the discriminators are trained to distinguish whether the whole real images and the whole generated images are real or faked. In our built discriminator, we use the latest discriminator model named the Markovian discriminator (PatchGAN) [37]. Particularly, we separate the real images and the generate images into N ×N patch (N =12) and train the discriminator to classify whether each N ×N patch in an image is real or faked. That makes the discriminator restrict its attention to the structure of the local image patches instead of the whole of the images so that the discriminator can model the high-frequency structure of the input images.

Table 7.1. Recognition rates of the top 10 categories of the highest recognition rates and overall average of 118 categories.

Categories (Shift-JIS codes and characters)	**Domain-*B*-OCR (%)**	**Domain-*B*-to-*A*-OCR (%)**	**# testing patterns**
(9286) 中	82.9	7.26	234
(8CDC) 五	81.06	23.02	808
(8eb5) 七	78.68	4.06	591
(9573) 不	75.94	12.8	133
(906C) 人	71.26	9.32	1,159
(8FE3) 上	69.19	13.32	766
(88CA) 位	69.16	3.89	642
(8DB2) 佐	67.47	2.41	166
(985A) 六	67.32	17.50	817
(9356) 天	66.24	4.82	394
.....			
Overall	41.32	7.37	18,554

The recognition rate by each category is shown in **Table 7.1**. The first column is the Shift-JIS code and character of each category. The second and third columns show the recognition rates by Domain-B-OCR on the generated test samples and Domain-B-to-A-

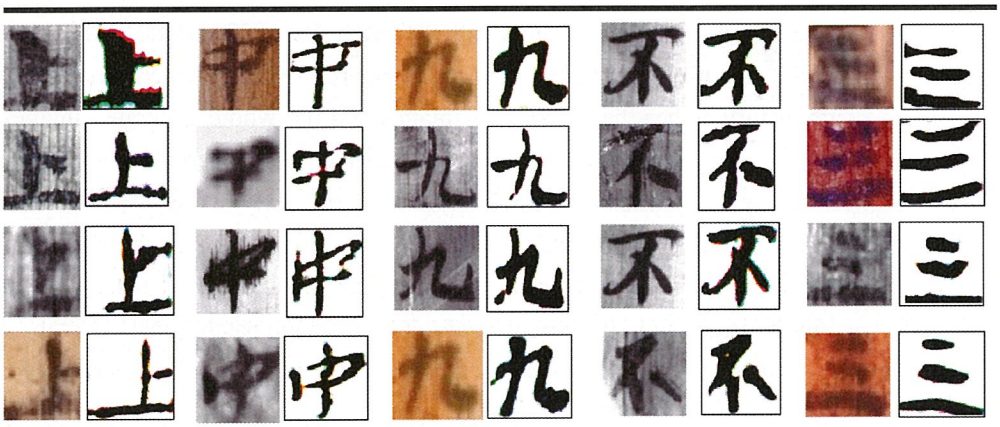


Figure 7.5. Raw character patterns and generated character patterns.

OCR on the original test samples, respectively. The last column is the number of patterns in each testing category. Here, we just show the recognition rates of the top 10 categories from the highest and the overall average of 118 categories. From the overall recognition rate, we can see that our Domain-***B***-OCR can easily recognize the generated test samples (high visibility) rather than recognizing the original test samples (low visibility) although it has been adapted to domain ***A***. Here, if the Inception-ResNetv2 is trained and tested on high visibility patterns from (Nakayosi for training, Kuchibue for testing), the model achieves the recognition rate of 98.92% for 2,965 categories.

Figure 7.5 shows the original raw images and the generated images by the generator of CAGAN. The background and noise of the degraded mokkan images are completely removed. Moreover, the missing and faded strokes or parts of the characters are also reconstructed.

Chapter 8. Remaining work and conclusion

This thesis presents the state-of-the-art methods for offline Japanese Handwriting recognition: over-segmentation methods, semantic character segregation methods, and free-segmentation methods. Free-segmentation methods are better than over-segmentation methods, but they are not robust to the change of wiring styles and character shapes. Semantic character segmentation based methods are expected to get better performance than others. In the thesis, the author have just completed the semantic segmentation based on U-Net. Using others such as FCN, Mask R-CNN should be considered. Also, they need to be combined with OCRs, linguistic contexts to evaluate the whole of the system.

Moreover, this paper also described a character attention generative adversarial network (CAGAN) for reconstructing degraded mokkan images. The generator and the discriminator are trained by combining the traditional adversarial loss and the proposed character attention loss. The character attention loss is computed from the feature maps of the pre-trained Inception-ResNet-v2 model by the TUAT Kuchibue& Nakayosi-mixed dataset. The author feeds the model by randomly choosing batches of low visibility images in one category and the counterparts of high visibility images, so we do not need a lot of data to train. The author also uses the U-Net architecture with skip connection for the generator and the Wasserstein loss function to train the discriminator. The discriminator is the latest Markovian discriminator which can learn the high-frequency structure. The recognition rate of the Inception-ResNet-v2 OCR showed that recognition from the generated images is much better than from the degraded images although we have already applied fine-tuning process for the OCR. CAGAN not only can remove the background of the degraded images but also reconstruct missing parts of the characters. From the perspective of the method, it is helpful if we can combine the method with the object detection methods to locate and recognize the characters on Heijokyo mokkan images. The author also wants to apply the method for other historical documents such as Vietnamese Nom documents and Pre-Modern Japanese Text (PMJT) published by the Center for Open Data in the Humanities (CODH) in 2016.

This thesis also presented a segmentation-based approach for digitalizing historical documents in Vietnam that have a high risk of permanent disappearance to the next generations. Character regions are extracted from the preprocessed documents by the U-Net based network with different encoders such as VGG-16, ResNet 50, and Inception-ResNet V2. After that, the character regions are recognized by the coarse and fine combined classifiers. The experiment shows that the proposed method makes the

networks converge quickly and produce better recognition rates than single fine classifiers on the dataset of a large number of categories.

The author also proposed the attention layer and added noises to the training dataset which helps the classifiers work well on the real testing dataset. The best-archived recognition rate is 85.07% for the real testing Nom dataset. We improve the recognition result by applying a beam search decoder, incorporated a Nom language model.

For remaining work, the author plans to find the optimum number of coarse categories to improve the performance of the coarse and fine combined classifiers. For character region segmentation, the author considers training the network which predicts the center of each character region as the form of Gaussian distribution to minimize the overlaps among character regions.

Bibliography

[1] Mori, S., Suen C.Y. and Yamamoto K. "Historical review of OCR research and development.", *Proceedings of IEEE, Vol. 80, Issue 7, pp. 1029-1058, (1992).*

[2] Plamondon, R. and Srihari, S.N. "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey.", *IEEE PAMI, vol. 22, No. 1, pp. 63-84, (2000).*

[3] Ikeda, Hisashi, et al. "A recognition method for touching Japanese handwritten characters." *Document Analysis and Recognition,. ICDAR 99. Proceedings of the Fifth International Conference on. IEEE, pp. 641-644, (1999).*

[4] Liu, Cheng-Lin, Masashi Koga, and Hiromichi Fujisawa. "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading.", *IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 11 pp.1425-1437, (2002).*

[5] Kha Cong Nguyen, Nakagawa Masaki. "Text-Line and Character Segmentation for Offline Recognition of Handwritten Japanese Text", *IEICE technical report, Vol. 115, No. 517, pp.53-58, (2016.3).*

[6] R. Messina and J. Louradour, "Segmentation-free handwritten chinese text recognition with lstm-rnn," *The 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 171–175, (2015).*

[7] Shih, V. J. Y., & Chu, T. L. (2004). The Han Nom Digital Library. *In The International Nom Conference, The National Library of Vietnam, Hanoi*, (12-14).

[8] Nhan, N. T., and Mai, C. (2015). An experience from Temple University Pilot Digital Project with the General Library of Thừa Thiên Huế, *A Mini-Conference in Nôm Studies, The National Library of Vietnam, Hanoi.*

[9] Khanh, T. T., Huyen, T. T. (2008). A Program to Translate the Chinese Taisho Tripitaka into English and Other Western Languages. *presentation at United Nations Vesak Day, Hanoi, Vietnam.*

[10] Dao, D. A. (1979). Chu Nom: origins, formation, and transformations. *Nhà Xuất Bản Khoa Học Xã Hội.*

[11] Truyen Van Phan, Kha Cong Nguyen, and Masaki Nakagawa. (2016). A Nom historical document recognition system for digital archiving. *International Journal on Document Analysis and Recognition (IJDAR) 19, no. 1,* (49-64).

[12] Ronneberger, O., Fischer, P., & Brox, T. U-net: Convolutional networks for biomedical image segmentation. *In International Conference on Medical image computing and computer-assisted intervention,* (234-241), 2015, October.

[13] Zhu, Bilan, and Masaki Nakagawa. "Online Handwritten Chinese/Japanese Character Recognition.", *Advance in Character Recognition, InTech, pp.51-68 (2012).*

[14] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.*

[15] Liu W, Rabinovich A, Berg AC. "Parsenet: Looking wider to see better." *arXiv preprint arXiv:1506.04579. 2015 Jun 15.*

[16] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." *In Proceedings of the IEEE international conference on computer vision, pp. 1520-1528. 2015.*

[17] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." *In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969. 2017.*

[18] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *In Advances in neural information processing systems, pp. 91-99. 2015.*

[19] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *CoRR, vol. abs/1507.05717, 2015.*

[20] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," *Proc. NIPS2008, Vancouver, Canada, pp. 545-552, 2008.*

[21] N. T. Ly, C. T. Nguyen, K. C. Nguyen and M. Nakagawa, "Deep Convolutional Recurrent Network for Segmentation-free Offline Handwritten Japanese Text Recognition", *Proc. MOCR2017, 2017.*

[22] N. T. Ly, Cuong Tuan Nguyen, Nakagawa Masaki, "An attention-based end-to-end model for multiple text lines recognition in Japanese Historical Documents", *ICDAR 2019, Sydney, Australia (September, 2019).*

[23] Jinfeng Gao, Zhu Bilan, and Masaki Nakagawa. "Development of a Robust and Compact On-Line Handwritten Japanese Text Recognizer for Hand-Held Devices." *IEICE TRANSACTIONS on Information and Systems 96.4, pp. 927-938 ,(2013).*

[24] Takakura, J., Kitadai, A., Nakagawa, M., Baba, H., & Watanabe, A. (2010, November). Techniques to enhance images for mokkan interpretation. *In 2010 12th International Conference on Frontiers in Handwriting Recognition (pp. 358-362).*

[25] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. *In Advances in neural information processing systems (pp. 2672-2680).*

[26] Yang, Qingsong, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K. Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose CT image

denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE transactions on medical imaging 37, no. 6 (2018): 1348-1357.*

[27] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W., 2017. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint.*

[28] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee, 2009.*

[29] Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. *arXiv preprint (2018).*

[30] Nakagawa, Masaki, and Kaoru Matsumoto. Collection of on-line handwritten Japanese character pattern databases and their analyses. *Document Analysis and Recognition 7, no. 1 (2004): 69-81.*

[31] Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *In Advances in Neural Information Processing Systems, pp. 5767-5777. 2017.*

[32] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *In Proceedings of the IEEE International Conference on Computer Vision (pp. 618-626).*

[33] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *In Thirty-First AAAI Conference on Artificial Intelligence. 2017.*

[34] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134. 2017.*

[35] Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint (2017).*

[36] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *In ICLR, 2015.*

[37] Li, Chuan, and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *In European Conference on Computer Vision, pp. 702-716. Springer, Cham, 2016.*

[38] Fernández-Mota, D., Lladós, J., & Fornés, A. (2014). A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition* (IJDAR), 17(3), (293-312).

[39] Zhao, S., Chi, Z., Shi, P., & Wang, Q. (2001). Handwritten Chinese character segmentation using a two-stage approach. *In Proceedings of Sixth International*

Conference on Document Analysis and Recognition, (179-183).

[40] Zheng, Q., Chen, K., Zhou, Y., Gu, C., & Guan, H. (2010). Text localization and recognition in complex scenes using local features. *In Asian Conference on Computer Vision*, (121-132).

[41] Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character Region Awareness for Text Detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (9365-9374).

[42] Cevahir, A. and Murakami, K. (2016). Large-scale Multi-class and Hierarchical Product Categorization for an E-commerce Giant. *In Proc. COLING,* (525-535).

[43] Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W. and Yu, Y. (2015). HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. *In Proceedings of the IEEE International Conference on Computer Vision*, (2740-2748).

[44] Jie, L. Zhenyu, G. and Yang, W. (2017). Weakly Supervised Image Classification with Coarse and Fine Labels. *The 14th Conference on Computer and Robot Vision (CRV)*, (240-247).

[45] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, (855–868).

[46] Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711.*

[47] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *In Thirty-First AAAI Conference on Artificial Intelligence*, (4278-4284).

[48] D. Kingma and J. Ba. (2015). Adam: A method for stochastic optimization. *In ICLR. arXiv preprint arXiv:1412.6980*

Publications

[AP1] Kha Cong Nguyen, Linh D. Truong, "FTTH Network Design with Google Map integration", *Proc. of the IEEE RIVF, Ho Chi Minh city, Vietnam, Feb-March 2012.*

[AP2] Kha Cong Nguyen, Truyen Van Phan, M. Nakagawa, "A System to Annotate and Cluster Pieces of Mokkan", *Proc of the ISPC conference, Tokyo University of Agriculture and Technology, Tokyo, Japan, May 2015.*

[AP3] Truyen Van Phan, Kha Cong Nguyen, M. Nakagawa, "A Nom Historical Document Recognition System for Digital Archiving", *a journal in International Journal on Document Analysis and Recognition, pp. 49-64, 2016.*

[AP4] Kha Cong Nguyen, Nakagawa Masaki, "Text-Line and Character Segmentation for Offline Recognition of Handwritten Japanese Text", *IEICE technical report, Vol. 115, No. 517, pp.53-58 (2016.3).*

[AP5] Kha Cong Nguyen, Nakagawa Masaki, "Enhanced Character Segmentation for Format-Free Japanese Text Recognition", *Proc. of the ICFHR2016, Shenzhen, China, pp. 138-143, October 2016.*

[AP6] Cong Kha Nguyen, Cuong Tuan Nguyen, Nakagawa Masaki, "Tens of Thousands of Nom Character Recognition by Deep Convolution Neural Networks", *Proc. of the 2017 Workshop on Historical Document and Processing, pp. 37-41, Kyoto, Japan (11.2017)*

[AP7] Hung Tuan Nguyen, Nam Tuan Ly, Cong Kha Nguyen, Cuong Tuan Nguyen, Nakagawa Masaki, "Attempts to recognize anomalously deformed Kana in Japanese historical documents", *Proc. of the 2017 Workshop on Historical Document and Processing, pp. 31-36, Kyoto, Japan (11.2017).*

[AP8] Nam Tuan Ly, Cuong Tuan Nguyen, Kha Cong Nguyen and Masaki Nakagawa, "Deep Convolutional Recurrent Network for Segmentation-free Offline Handwritten Japanese Text Recognition", *Proc. of Int'l Workshop on Multilingual OCR, pp. 5-9, Kyoto, Japan (11.2017).*

[AP9] Kha Cong Nguyen, Cuong Tuan Nguyen, Nakagawa Masaki, "A Segmentation Method of Single- and Multiple-Touching Characters in Offline Handwritten Japanese Text Recognition", *a journal of IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS Volume: E100D Issue: 12 pp. 2962-2972 (DEC 2017).*

[AP10] Nam Tuan Ly, Kha Cong Nguyen, Cuong Tuan Nguyen, Masaki NAKAGAWA, "Recognition of Anomalously Deformed Kana Sequences in Japanese Historical Documents", *a journal of IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS Volume: E102.D No.8, pp.1554-1564 (2019.8).*

[AP11] Kha Cong Nguyen, Cuong Tuan Nguyen, Seiji Hotta, Nakagawa Masaki, "A Character Attention Generative Adversarial Network for Degraded Historical Document

Restoration", *ICDAR 2019, Sydney, Australia (September, 2019).*

Appendix

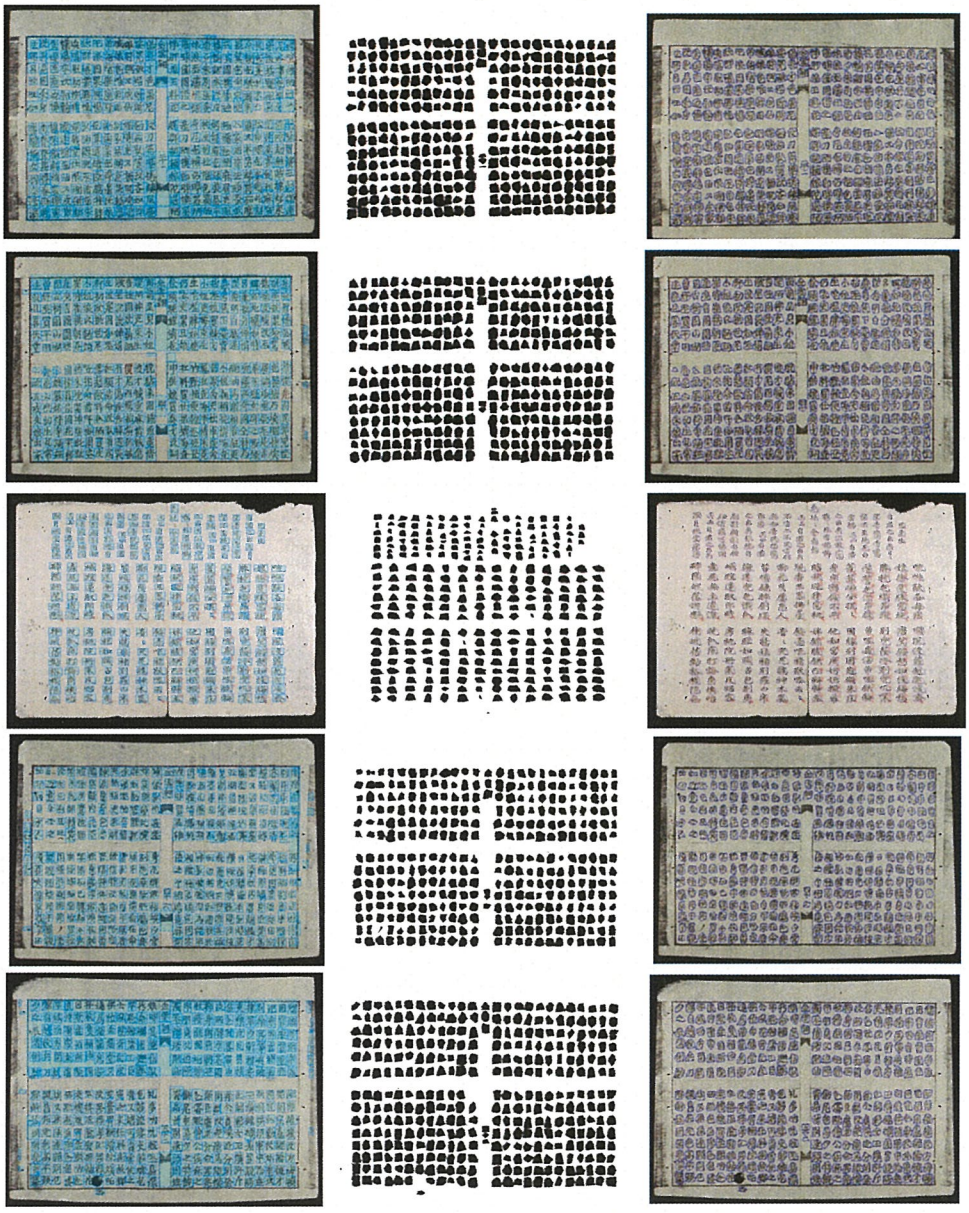


Fig. A-1. Some segmentation results on real Nom pages. First column: segmentation by the combination of the projection profile and the Voronoi diagram. Second column: polygonal segmentation maps by the fine-tuned models. Third column: character regions after applying the marker-based watershed.