The Automatic Classification of Noh Chant Books with Machine Learning

Suzuka Kokubu

Music Technology Area

Department of Music Research

Schulich School of Music

McGill University, Montreal

August 2022

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Arts

Abstract

Noh is a traditional Japanese performance art form that has developed its unique style over the course of more than 600 years and was designated as an Intangible Cultural Heritage by UNESCO in 2008. The music of Noh is composed of several musical instruments in addition to vocals, with the chant being written in dedicated Noh chant books. Within these books, pitches and rhythms are specified with symbols called *fushi*, which are written alongside the text.

One of the challenges in the study of Noh is the difficulty in finding Noh chant books in libraries due to cataloguing issues. Because it is difficult to distinguish Japanese musical notations, Noh chant books are often miscatalogued. Even in the national legal depository of Japan, the National Diet Library, Noh chant books are often catalogued with other books with distinct types of musical notation. This is a difficult task especially for non-specialized cataloguers to distinguish among different music genres. To aid in this cataloguing task, this study focuses on the automatic classification of Noh chant books using a unique feature of a document image, *fushi* notation, to aid the human cataloguers.

Three machine learning models were used to identify these components and accomplish the document classification. First, a dataset consisting of Noh chant and non-Noh-chant books was obtained from the same class at the National Diet Library. Then, using document analysis models, all musical notations were extracted from document images. These were further processed with connected component analysis to separate them, and each component was classified using a symbol classification

model. The last process is binary classification, which uses the outcome of the symbol classification model to identify Noh chant books using a decision tree classifier.

Résumé

Le nô est une forme d'art de la scène traditionnel japonais qui a développé son style unique pendant plus de 600 ans et a été désigné patrimoine culturel immatériel par l'UNESCO en 2008. La musique du nô est composée de plusieurs instruments de musique ainsi que du chant, celui-ci étant écrit dans des livres de chant nô. Dans ces livres, les hauteurs et les rythmes sont indiqués par des symboles appelés *fushi*, écrits à côté du texte.

L'un des défis de l'étude du nô est la difficulté de trouver des livres de chant nô dans les bibliothèques en raison de problèmes de catalogage. En effet, parce qu'il est difficile de distinguer les notations musicales japonaises, les livres de chant nô sont souvent mal catalogués. Même dans le dépositaire légal national du Japon, la Bibliothèque nationale de la Diète, les livres de chant nô sont souvent catalogués avec d'autres livres avec des types différents de notation musicale. Il est difficile, en particulier pour les catalogueurs non spécialisés, de distinguer les différents genres musicaux. Pour faciliter cette tâche de catalogage, la présente étude porte sur la classification automatique des livres de chant nô en utilisant une caractéristique unique d'une image de document, la notation *fushi*, pour aider les catalogueurs humains.

Trois modèles d'apprentissage automatique ont été utilisés pour identifier ces composants et classifier des documents. Tout d'abord, un ensemble de données composé de livres de chant nô et de livres de chant autre a été obtenu auprès de la même classe à la Bibliothèque nationale de la Diète. Ensuite, à l'aide de modèles

d'analyse de documents, toutes les notations musicales ont été extraites des images de documents. Celles-ci ont ensuite été traitées avec une analyse des composants connectés afin de les séparer, et chaque composant a été classé à l'aide d'un modèle de classification des symboles. Le dernier processus est la classification binaire, qui utilise le résultat du modèle de classification des symboles pour identifier les livres de chant nô à l'aide d'un classificateur d'arbre de décision.

Acknowledgement

I would like to thank all of my colleagues and friends who helped me in writing this thesis. I would not have been able to complete this work without the guidance and support of all the members of the Distributed Digital Music Archives and Libraries (DDMAL) Laboratory. I would like to thank Francisco J. Castellanos for helping me work with the layer analysis module and Martha E. Thomae for helping me work with Rodan and Compute Canada. I would also like to thank Sevag Hanssian and Geneviève Gates-Panneton for helping to translate my abstract into French.

I would like to thank Ichiro Fujinaga, who guided me throughout this research from its inception to its completion. He was instrumental in coming up with many of the ideas used for this research. His wealth of experience was very helpful for the literature review and learning all the utilized methodologies. All the comments and feedback were invaluable for both conducting the research as well as writing it up.

I would like to thank for my partner for supporting me with love and compassion, in addition to revising my thesis. I would also like to thank all my family for supporting me throughout this research. I also would like to thank all my colleagues at IBM Japan for their understanding and supporting me in the completion of my thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Noh is a traditional performing art of Japan that has gained international recognition and been added as an Intangible Cultural Heritage by UNESCO in 2008. The characteristic of Noh is its simplicity. Unlike modern Western theater, all Noh stages have a similar, simple, and small stage design, as shown in Figure 1-1. Similarly, unlike Western opera, which is performed with an orchestra, Noh music is performed with only a few singers and a few instruments, shown in Figure 1-2: a flute shown in Figure 1-3, a small drum (*kozutsumi*), a large drum (*ozutsumi*) shown in Figure 1-4, and a *taiko*, also known as *shime daiko*, which is similar to the large drum with a wider body. Noh plays have been performed for a millennium, but they were not written down until about the mid-fourteenth century (Pinnington 2019, 21). The transition to a scripted format of Noh, especially with the inclusion of the chanting part of Noh, allowed for the accurate transmission of the contents of plays.

Figure 1-1 An image of a contemporary Noh stage (Artanisen 2019).

Figure 1-2 An image of group of instrument performers in Noh (Fujinami 1975).

Figure 1-3 An image of Japanese flutes. *Nokan*, which is used for Noh, is the first one from the bottom in the picture (Sano 2005).

Figure 1-4 An image of Japanese hand drums. The left is a large drum(*ozutsumi*), and the right is a small drum (*kozutsumi*) (Dopehat39 2018).

One of the challenges in the study of Noh is the difficulty in finding Noh chant books in libraries due to cataloguing issues. Because it is difficult to distinguish Japanese musical notations, Noh chant books are often miscatalogued. The National Diet Library, the largest library in Japan with more than 45 million items, is the national legal depository, analogous to Library and Archives Canada and the Library of Congress (National Diet Library 2021). Even at this library, Noh chant books are often catalogued with other books with distinct types of musical notation. This is a difficult task especially for non-specialized cataloguers to distinguish among different music genres. To aid in this cataloguing task, this study focuses on the automatic classification of Noh chant books using a unique feature of a document image, *fushi* notation, to aid the human cataloguers.

The goal of this thesis is to automatically classify Noh chant books using the characteristic appearance of its notation system. Noh chant books contain musical symbols, *fushi*, and other elements such as text, the role of the performer, or frontispieces, as shown in Figure 1-5. While text, Figure 1-5 (a), denotes the words

used in the chant, *fushi*, Figure 1-5 (b), indicates how the text should be sung in terms of pitch and rhythm. The text appears in Noh chant books as standard Japanese characters without any special symbols. The *fushi* notation, however, is distinctive to those who have domain knowledge and comprises dot-like signs alongside the Japanese characters that indicate how each syllable should be sung. Thus, these symbols allow for books to be identified as Noh chant books. In other words, detecting the existence of *fushi* notation can be used for automatic document classification. However, Figure 1-6 shows examples of non-Noh-chant books, where (a) has text written in a similar style as Figure 1-5, and (b) and (d) contains dot-like symbols which could be mistaken as *fushi* notation, complicating the manual classification of books as Noh chant or non-Noh-chant. Therefore, machine-aided classification may be useful.

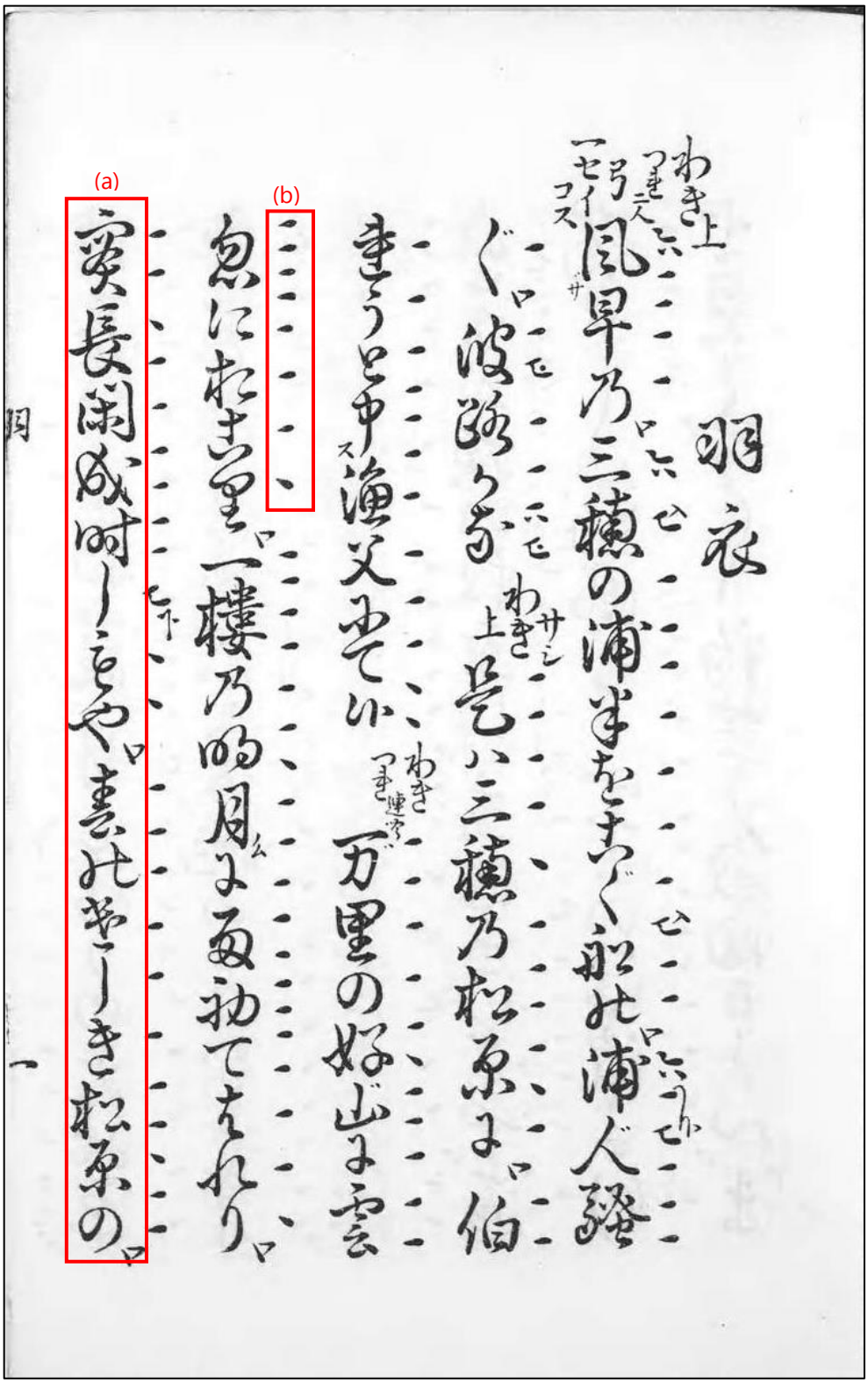
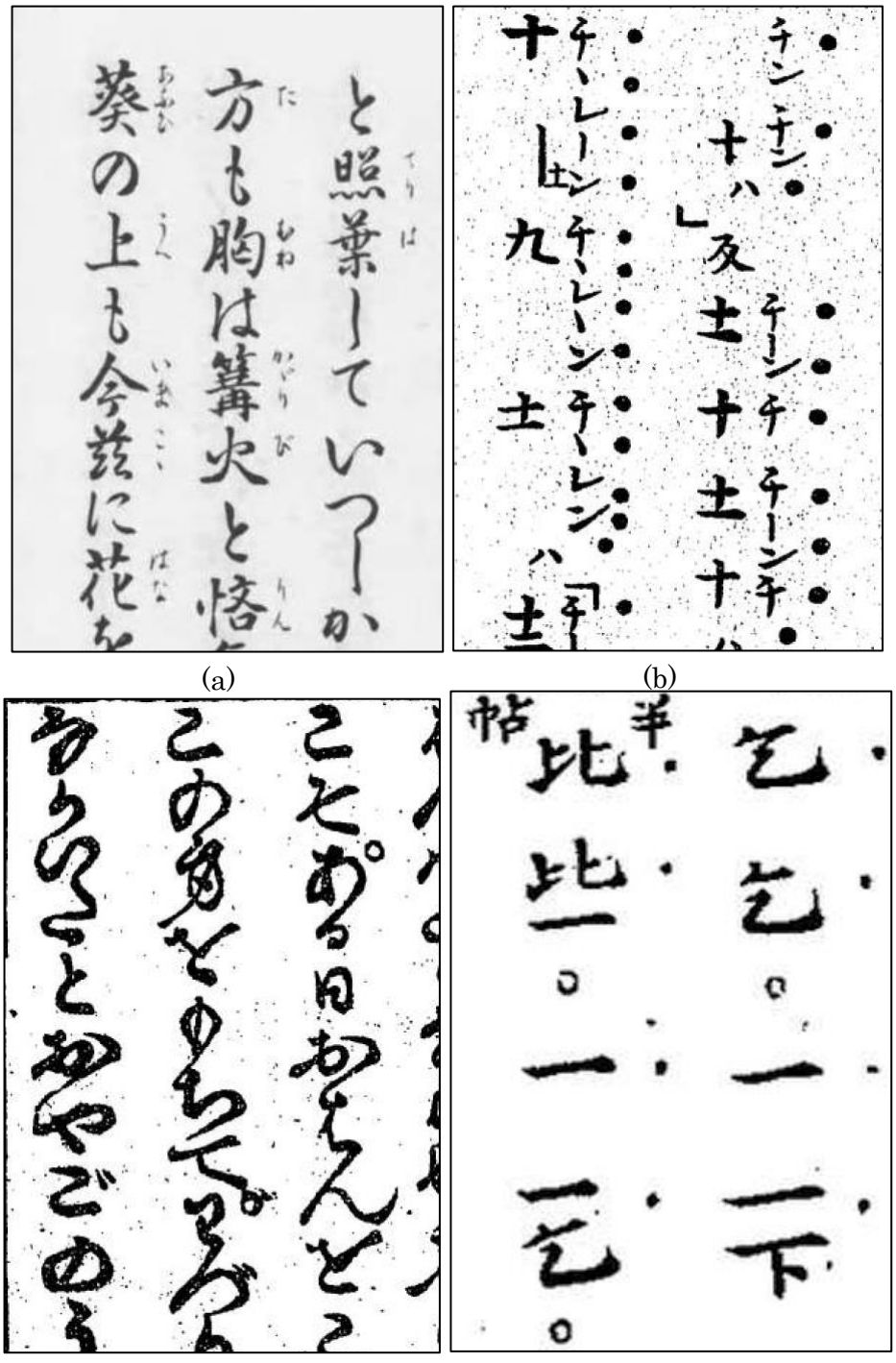


Figure 1-5 An example of Noh chant book (Hosho 1936, 3). The red box marked (a) contains some chant text and the red box marked (b) contains some *fushi* notation.



(a) (b)

(c) (d)

Figure 1-6 Examples of other Japanese books (non-Noh-chant). (a) (Teihon tokiwazu zenshu kankoukai 1943, 20), (b) (Tokuhiro 1899, 16), (c) (Tan 1883, 5), and (d) (Tougi 1894, 26)

1.1 Brief History of Noh

For the most of its early existence, Noh was a collection of various performing arts, often with religious elements. Noh was not acknowledged as a unified performing art genre until the fourteenth century, when Noh performers earned sponsorship from the military government (Nogami 2005).

Even though Noh plays have been performed for centuries, they were not scripted until the mid-fourteenth century (Pinnington 2019, 21). Prior to this time, it is likely that they were planned orally and put together by combining dialogues and songs. Noh is thought to have evolved from *sangaku*, which originally came from China during the eighth century and combined with ancient Japanese comedies, called *sarugaku*. *Sarugaku* gained in popularity from the ninth to fourteenth centuries; its subject matter was comedic and included skits, acrobatics, and magic. During this time, the content of sarugaku also evolved and separated into two elements, Noh and Kyogen: Noh had a more serious and dramatic tone while Kyogen had more comic dialogue. Noh itself was known as *sarugaku* from the fifteenth to nineteenth centuries. Those scripts were written down as records of already established performance practice intended to define the repertoire long after the inception of the plays (Pinnington 2019, 26).

Around the turn of the fourteenth century, the first performance corresponding to today's Noh plays appeared (Pinnington 2019, 26). Although there had been a long tradition of performances before then, the contents of such performances were not always clear. For example, the play *Okina* is one of the Noh performances still

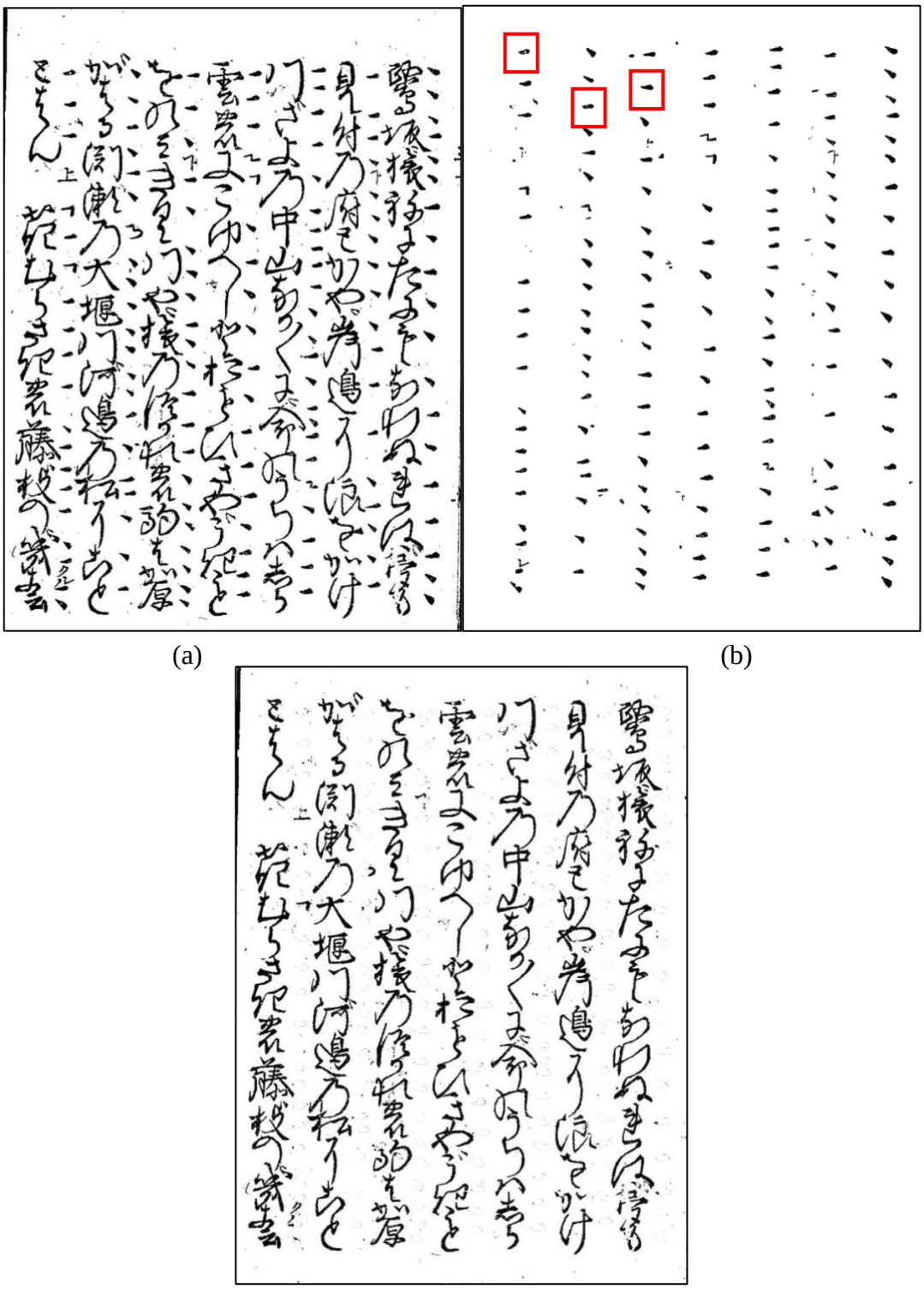
performed in the present day, and the forerunner of this piece was often performed at shrines during rituals around this time. In the following generation, *sarugaku* became a key element in entertainment, especially in aristocrat and warrior society in Kyoto, which are described and written by Kanze Zeami (1363–1443). Since then, Noh was able to develop with the support of warlords (Pinnington 2019, 91).

While Noh was established as a form of entertainment, it also expanded as music and dance for private amateur performances starting in the sixteenth century. In the seventeenth century, Noh spread to many more people including the lower merchant classes, and Noh chants were printed and widely used throughout Japan (Nishiyama 1997). They were also used in primary education to teach about geography, history, literature, and culture, which shows that Noh progressed from just entertainment to also encompassing education and training (Takanori and Tokita 2008, 128–129).

1.2 Project Overview

The goal of this thesis is to develop machine learning models that can automatically classify Noh chant books among other Japanese books with a similar writing style. In order to classify Noh chant books, the initial step is data acquisition since there are no pre-existing labeled datasets available. 16 Noh chant and 15 other Japanese books were manually selected from the National Diet Library to create datasets. These document images were processed to extract *fushi* notations using Rodan, which is a web-based workflow engine for optical music recognition (OMR) (Fujinaga 2019). There are three machine learning models that were applied: layer

separation, symbol classification, and binary classification. The document images are first processed to extract a Noh notation layer using layer separation models, which classify regions of pixels containing Noh notation to one layer and everything else to the other layer, as shown in Figure 1-7. Then, those extracted layers are processed with a symbol classification model to classify each connected component within a Noh notation layer. Three examples of connected components are enclosed with red rectangles in Figure 1-7 (b). The last process is binary classification, which determines whether a document is a Noh chant or another document class, shown in Figure 1-8, using the results from the symbol classification process.



(a) (b) (c)

Figure 1-7 A result from the layer separation (Terada 1885). (a) is the original image, (b) is the Noh notation layer, and (c) is the background layer. A red rectangle in (b) shows a connected component in the layer.

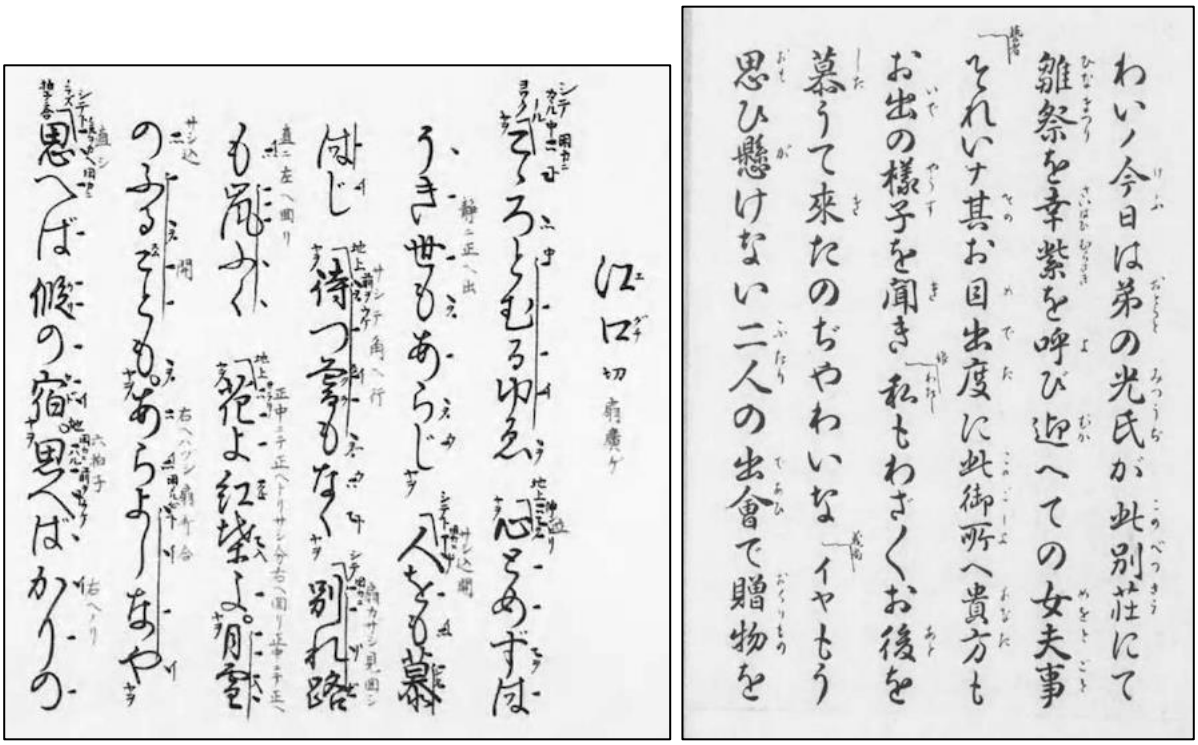


Figure 1-8 A comparison of Noh chant book and non-Noh-chant books. The right image is Noh chant book (Kanze 1923), and the left image is non-Noh-chant book (Teihon tokiwazu zenshu kankoukai 1943).

1.3 Thesis Organization

There are five chapters in this thesis. This first chapter included an introduction to Noh history as well as an outline of this research. The description of Noh notations, the classification of Noh chant books using a standard Japanese library cataloguing system, a review of related studies, and finally an explanation of the Rodan system are all covered in Chapter 2. The methodology for data collection as well as classification of Noh chant books, which contains layer separation, symbol classification, and binary classification of Noh chant from a dataset containing Noh chant and non-Noh-chant books, are presented in Chapter 3. The creation of the datasets, the results from experiments, and additional analysis of data outliers are

covered in Chapter 4. Finally, Chapter 5 is the conclusion that contains a summary of this study as well as future works.

Chapter 2 Background

In this chapter, we will go over all the background concepts necessary for this research. Section 2.1 describes major Noh notations in two different chanting styles. Then, Section 2.2 explains how Noh chant books are categorized within the Japanese standard library classification system and raises potential cataloging issues with this system. Section 2.3 introduces related technologies, such as book classification, document classification, script identification, semantic segmentation, and decision tree classifiers. Finally, Section 2.4 describes the Rodan system that was used in this research.

2.1 Noh Chant and its Notation

Noh chant books contain musical symbols known as *fushi*, as well as other elements such as text, the role of each performer, and frontispieces. *Fushi* indicates musical speech to be performed with a rhythm and melody and describing relative intervals. Figure 2-1, Figure 2-2, and Figure 2-3 are the first three pages of a Noh chant document titled *Aoinoue*, which is the name of a Noh play. *Fushi* is a dot-like symbol that starts to appear in Figure 2-2 (e). Here, it is important to mention that Japanese sentences must be read from top to bottom and right to left. As you can see, these document images include a lot more than just text and *fushi* notation. The information on the right side on the first page, shown in Figure 2-1 (a), is about costumes that each character wears on stage. The small image that appears above

the text, shown in Figure 2-1 (b), describes the movement of those characters. The role of character is also specified at the beginning from each phrase, with the protagonist referred to as *shite* in Figure 2-3 (f), antagonist referred to as *waki,* and the companions of *shite* and *waki* are referred to as *tsure* in Figure 2-2 (d) and *wakitsure* in Figure 2-1 (c), respectively. Along with the text, there is *furigana*, which is a reading aid for some kanji characters written in syllabic characters as shown in Figure 2-3 (g). Here, two slightly different *furigana* are written both right and left sides of the text. The musical and rhythmical expressions are shown in Figure 2-3 (h).

The vocal music in Noh performance can be divided into two separate modes: *utai* and *kotoba*, which indicate chanting and speech, respectively (Serper 2000). *Utai* is a vocal part characterized by a melody determined by relative pitch. *Kotoba*, on the other hand, is a dialogue with a unique tune, yet it does not have a defined melody. In Noh chant, *utai* is a section with *fushi* notation alongside the text, while *kotoba* is a section without such notations. These can be seen in Figure 2-1 as well: the document begins with the *kotoba* section and continues until line five on the second image in Figure 2-2, where the *utai* section begins and continues for the rest of the example.

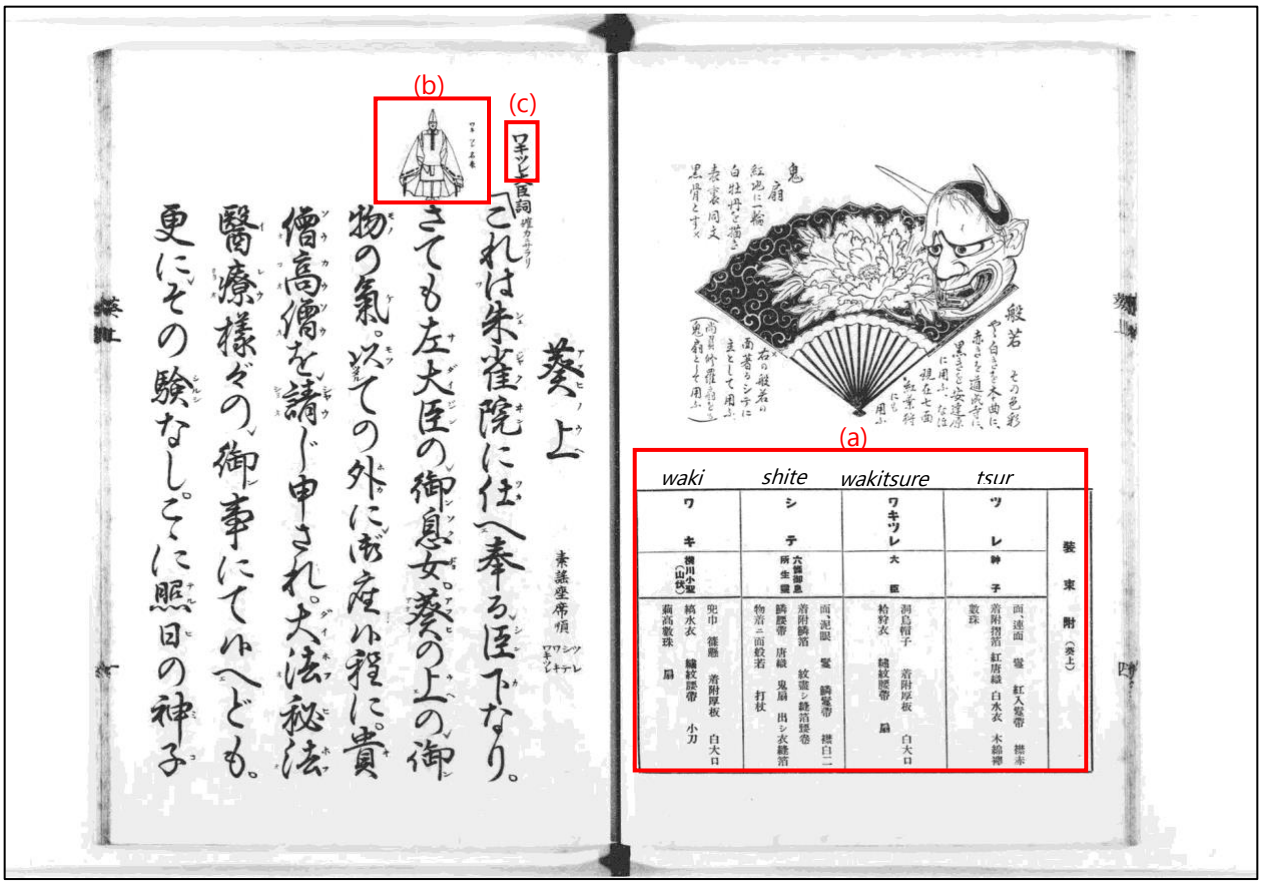


Figure 2-1 Document image from a Noh chant book. The first page from a performance script, called *Aoinoue* (Kanze 1934, 59). (a) describes each character's costumes. (b) shows character's movement. (c) specifies which character to chant the text right below the red rectangles.

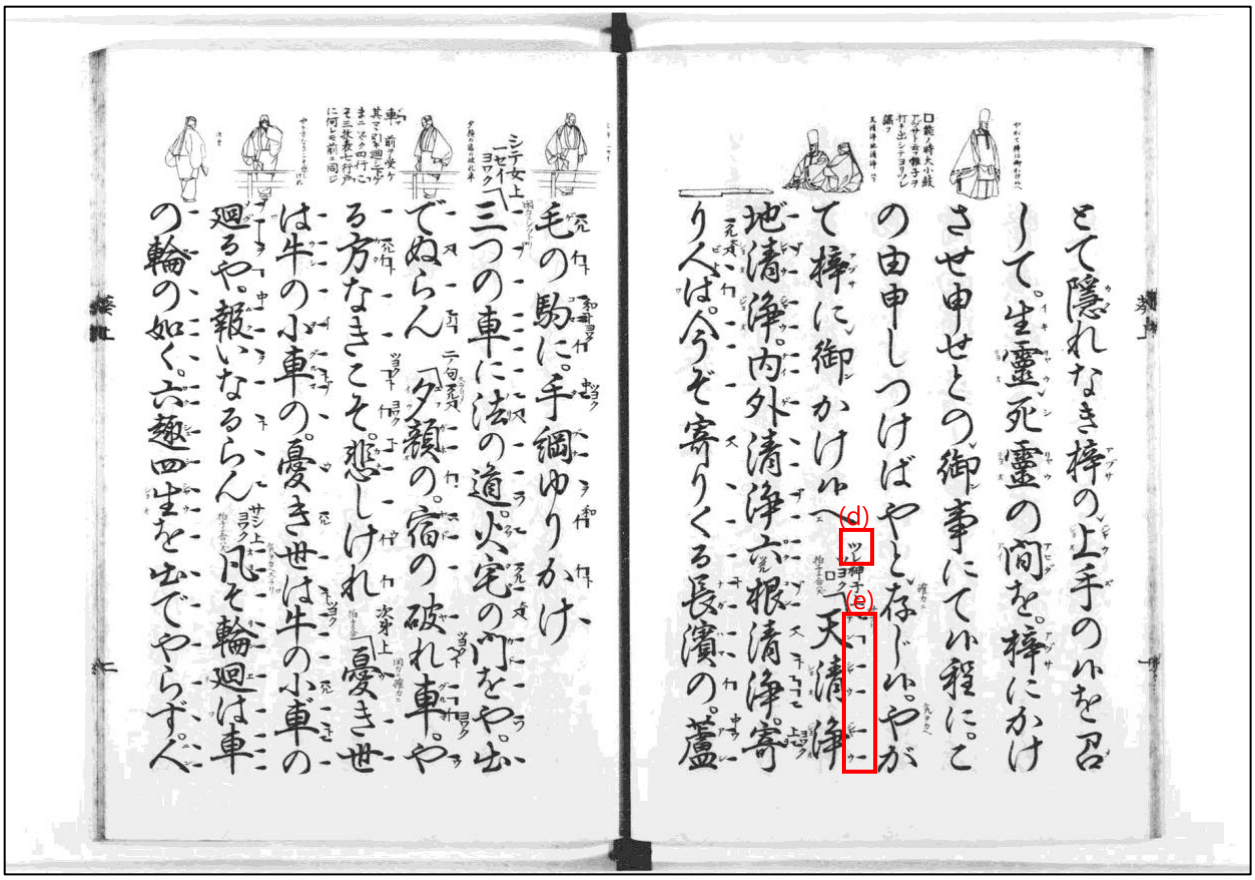


Figure 2-2 Document image from a Noh chant book. The second page from a performance script, called *Aoinoue* (Kanze 1934, 60). (d) specifies which character to chant the text right below the red rectangles. (e) is *fushi* notation.

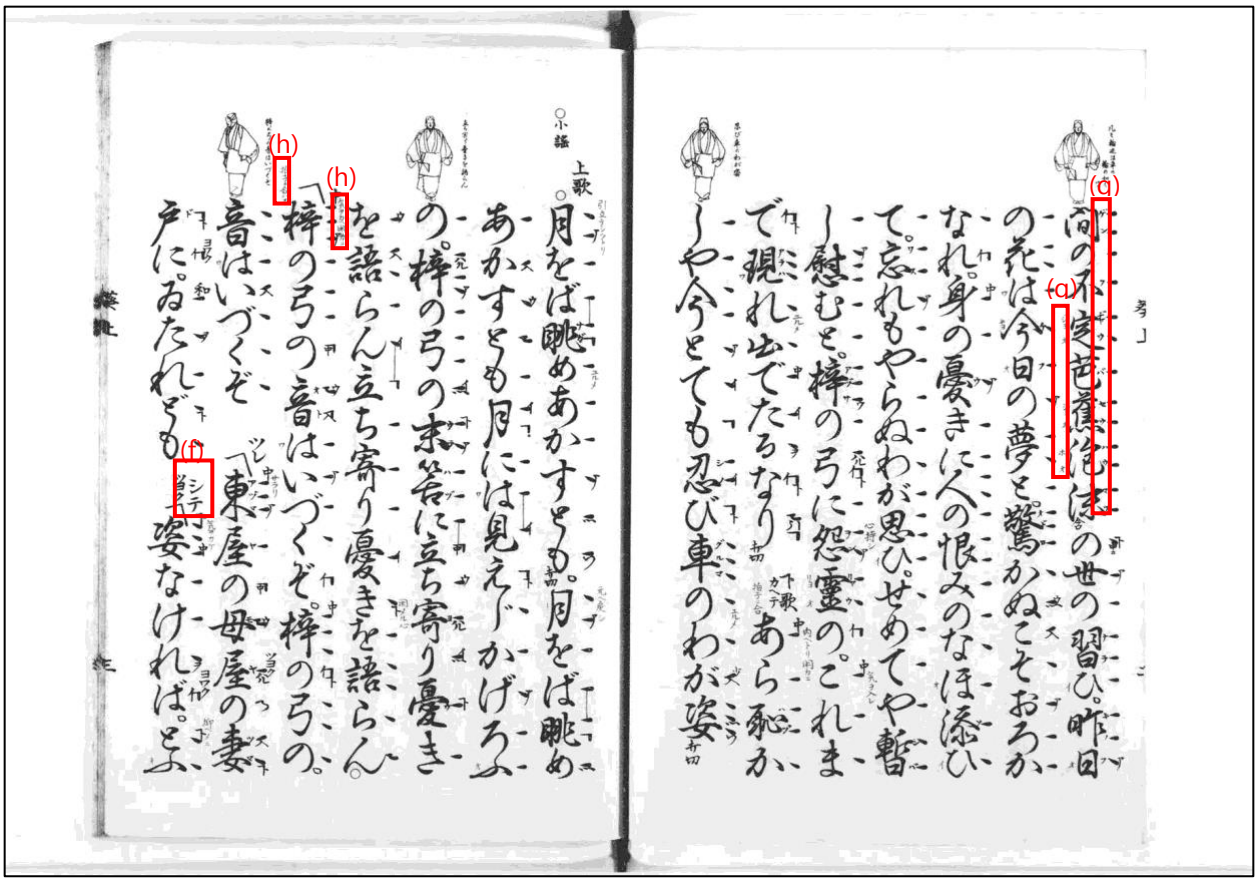


Figure 2-3 Document images from a Noh chant books. The third page from a performance script, called *Aoinoue* (Kanze 1934, 70). (f) specifies which character to chant the text right below the red rectangles. (g) shows *furigana* of *kanji* characters. (h) shows musical and rhythmical expressions.

2.1.1 Fushi

Fushi is a notation used for the *utai* section and composed of a variety of symbols. Even the same notation can be sung in a number of ways depending on a variety of elements, such as the chant style. In an *utaibon*, the Japanese term for a Noh chant book, the sung text is annotated with dot-like symbols that indicate how it should be sung (Kanzeryu Yokyoku Kenkyukai 1929, 22). Figure 2-4 is called *aya fushi*, which

is the most fundamental notation, and numerous examples appear also in Figure 2-2 and Figure 2-3.

Figure 2-4 An example of *aya fushi* notation (Miyake [1951] 2005, 5)

There are many other additional types of *fushi* notations, which are often a combination of this *aya fushi* notation and other symbols, representing varying pitch as well as musical expressions. These are also known as *goma fushi*, and examples can be found in Figure 2-5.

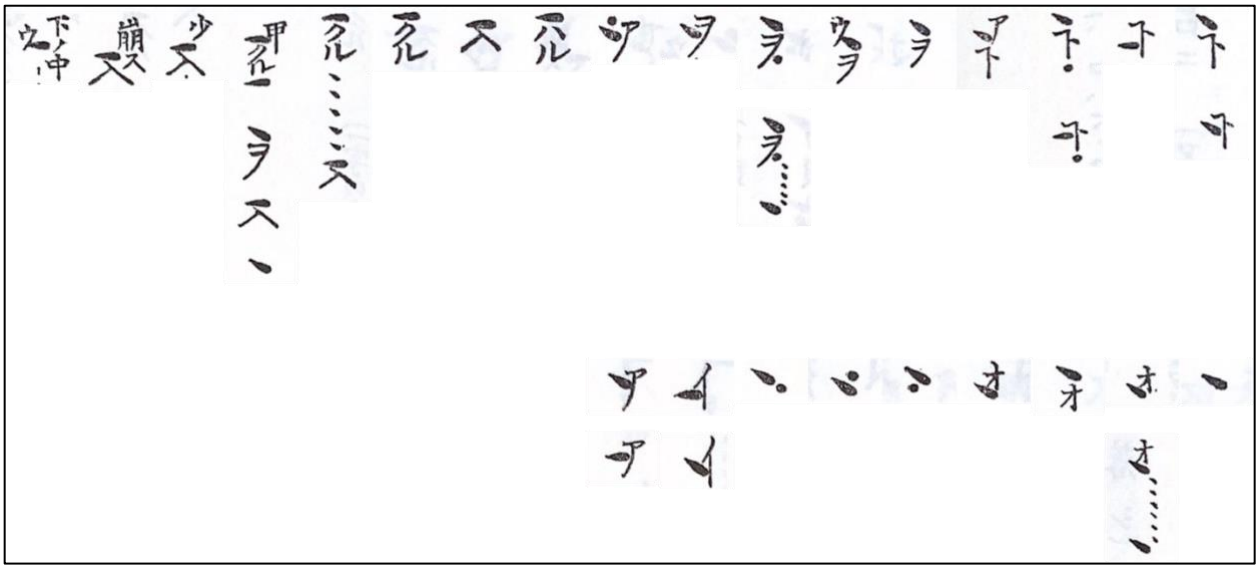


Figure 2-5 Examples of *goma fushi* notations (Miyake [1951] 2005, 2–5)

2.1.2 The Difference in Chant Styles

There are two different ways of chanting with same *fushi* notation, *yowagin* and *tsuyogi*n. The *yowagin* is translated as "weak singing" in English and is chanted by a soft and melodious voice, which is used to express moods, such as elegance and sadness. The *tsuyogin*, on the other hand, is translated as "strong singing" and is generally chanted with the strong impression, varying more of intensity than melodic range. The fundamental difference between *yowagin* and *tsuyogin* is that *tsuyogin* has a much simplified melody since it contains fewer sung pitches and the level of intensity and rhythm are more important for *tsuyogin*. (Miyake [1951] 2005, 3–8).

2.1.2.1 Yowagin

Yowagin has five pitches spanning two octaves: very low (*ryo-on*), low (*ge-on*), middle (*chu-on*), high (*jo-on*), and very high (*kuri-on*) (Miyake [1951] 2005, 15). In melodies, consecutive pitches do not leap more than two pitch levels, which indicates that a pitch can move from low to high, but not from low to very high without passing high. Low, middle, and high are three fundamental pitches, each with a relation of more or less a perfect fourth (Fujita, Kapuściński, and Rose 2019). Very low and very high are less frequently used compared to those fundamental pitches but appears sometimes to enrich the expression: very low appears after low, or very high appears after high. In fact, transitions between pitches follow a number of patterns and rules. In addition to the direct pitch transition (e.g., low to high), there is another transition that includes a slightly ascended pitch from an initial pitch in the middle of transition.

This middle pitch is called *ukion*, and this only occurs at the transition between middle and high. During this transition, the middle note will be slightly ascended from the initial note before reaching the last note: from middle to middle *ukion* to high, or from high to high *ukion* to middle. This type of transition, on the other hand, never occurs in a transition between low and middle, and instead these have direct transition: from low to middle, or from middle to low.

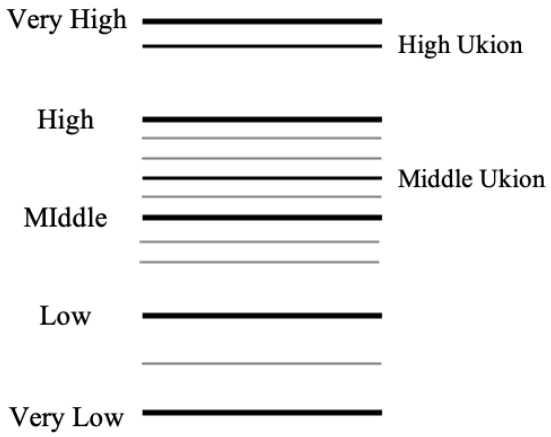


Figure 2-6 Yowagin pitches

The following summarizes the basic rules associated with transitions between fundamental pitches:

1. The pitch always ascends or descends in the order of low, middle, and high, and the melody does not jump more than two pitch levels at one time.
2. A slightly ascended pitch before a transition, *ukion*, occurs when transitioning from middle to high or vice versa. For example, the *ukion* appearing in the middle of ascending transition from middle to high would

be middle - middle *ukion* - high, while descending transition from high to middle would be high - high *ukion* - middle.

3. For a transition between middle and low, a pitch is always directly jumped to without having *ukion*.

As the above rules state, any transition associated with low occurs directly, and *ukion* only appears in the transition between middle and high.

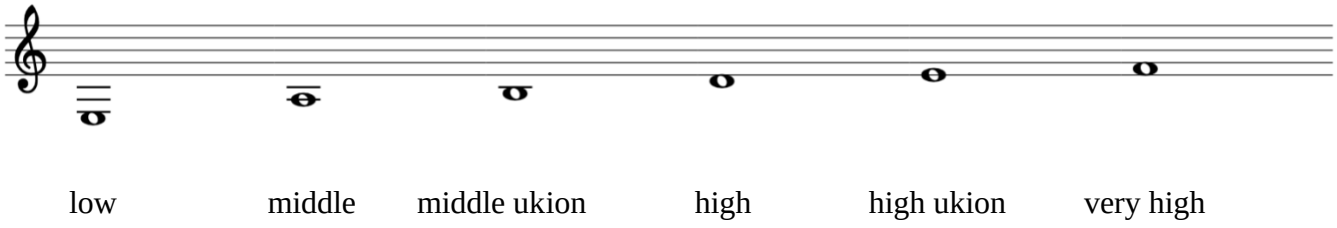


Figure 2-7 Musical scale of *yowagin*

2.1.2.2 Tsuyogin

The *tsuyogin* has a similar scale and rules as *yowagin*, however there are fewer sung pitches. In addition to low, middle, and high, similar to *yowagin*, it also includes low-mid (*ge-no-chu*). While *yowagin* has three fundamental pitches, *tsuyogin* has four fundamental pitches: low, low-mid, middle, high. However, due to the following rules, actual sung pitches are two:

1. A sung pitch should stay the same when transitioning between middle and high.
2. A sung pitch should stay the same when transitioning between low and low-

mid.

Therefore, *tsuyogin* has four pitches that are defined and even written in Noh chant but has two fundamental pitches to be sung, which has the relation of a minor third (Miura 1998). Since there is no difference in pitch for low to low-mid and middle to high, it is hard to distinguish those by ears. However, these are written differently in a chant book. *Tsuyogin*, unlike *yowagin*, does not contain *ukion*, a slightly ascended pitch that appears during some transitions to enhances expressiveness for *yowagin*. There are several techniques in *tsuyogin* to emphasize the melody, although they are not expressed in Noh chant books (Miyake [1951] 2005, 91–94).

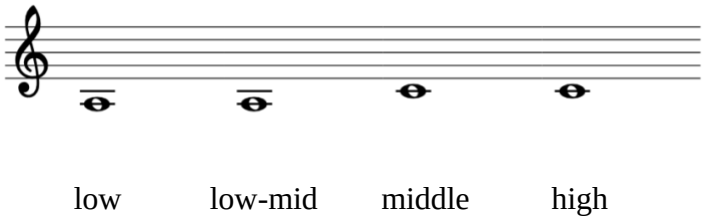


Figure 2-8 Musical scale of *tsuyogin*

2.1.3 Notations

2.1.3.1 Yowagin

There are two ways of describing a pitch, either writing the pitch directly or expressing it indirectly. For the former approach, a pitch is written using letters for low, middle, and high. Examples for these notations are shown in Figure 2-9 with annotation of red rectangles. When these notations appear, this pitch will be kept

until the next notation that changes the pitch appears. Table 2-1 shows several common notations for the expression of pitch transitions in *yowagin* (Miyake [1951] 2005, 16–17).

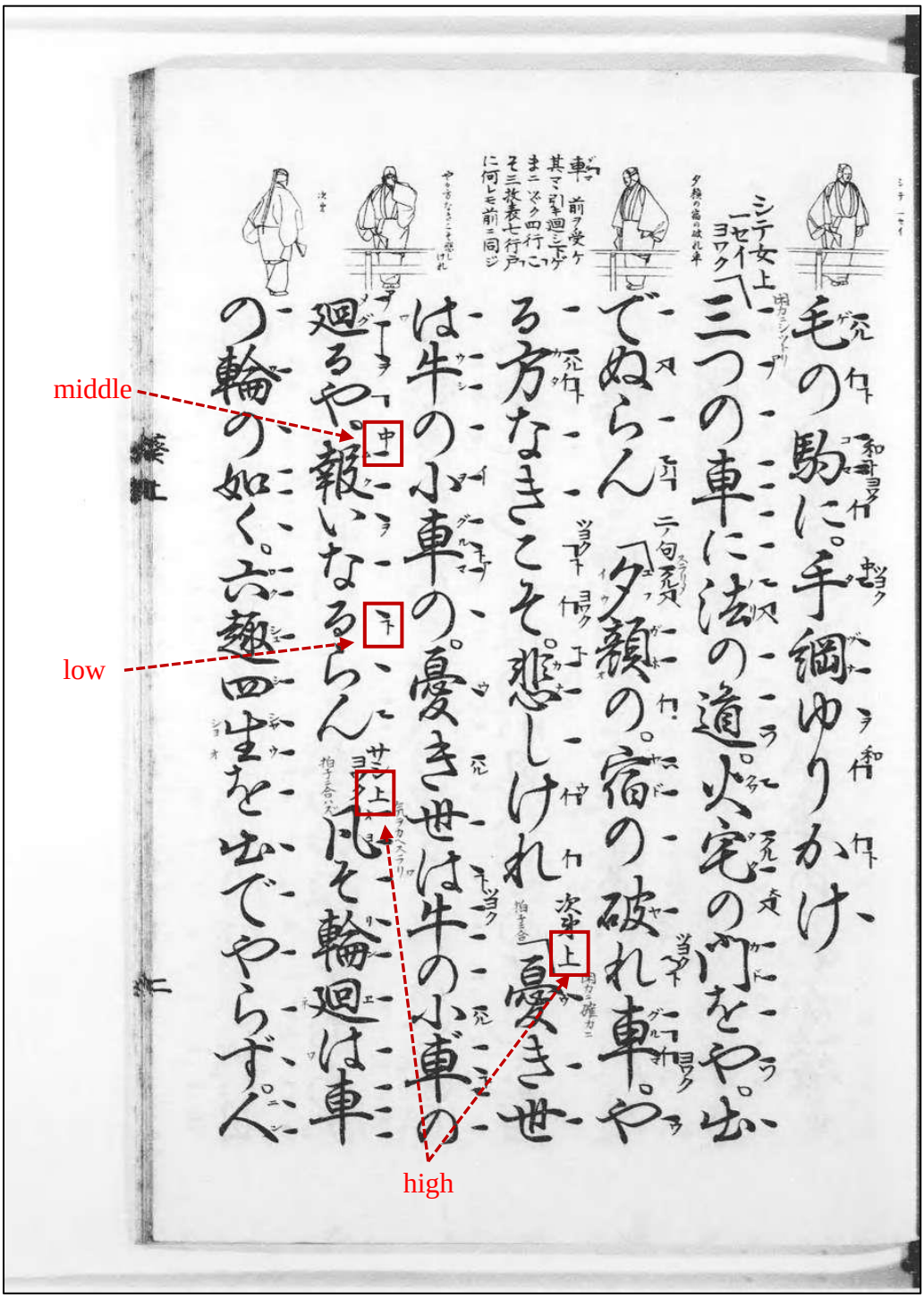


Figure 2-9 Document images from a Noh chant book of *Aoinoue* (Kanze 1934, 60). Red rectangles show written pitches of low, middle, and high.

Table 2-1 Common notations for transitions between fundamental pitches in *yowagin* (Miyake [1951] 2005, 1–3)

Name	Notation Image	Description
low (*ge)*		This notation means to sing at a low pitch if this notation appears at the beginning of sentence.
middle (*chu*)		This notation means to sing at a middle pitch.
high (*jo*)		This notation means to sing at a high pitch.
haru or *hari*		This is another way to describe high pitch. This notation appears when transitioning from middle to high and denotes shifting a pitch from middle to high.
sage		This notation appearing at the middle of a sentence means to lower one pitch. However, the notation appearing while singing at middle pitch indicates that more than two syllables are lowered, then back to the middle.
otoshi or *osae*		This notation also means that a pitch is lowered similar to *ge*, but only one syllable will be lowered, and the next syllable will be back to the original pitch.
iri		Reverse of the above, this notation means that one syllable from a middle of sentence will be in the higher pitch.
uki		This notation means *ukion*, which appears before a transition between middle and high according to the Rule 2 stated above.

Figure 2-10 shows the usage of *yowagin* notation in the play called *Tomoe*. At (a), one can see the notation of high, specifying the pitch to be sung. A notation of *uki* at (b) transitions the pitch from high to *ukion* and continues until the *sage* notation at (c), which lowers the pitch from *ukion* to middle. From (d), the new sentence starts

from *ukion*, which was denoted with notations of *haru* and *uki*. Lastly, the notation of *sage* at (e) transitions the pitch from *ukion* to middle.

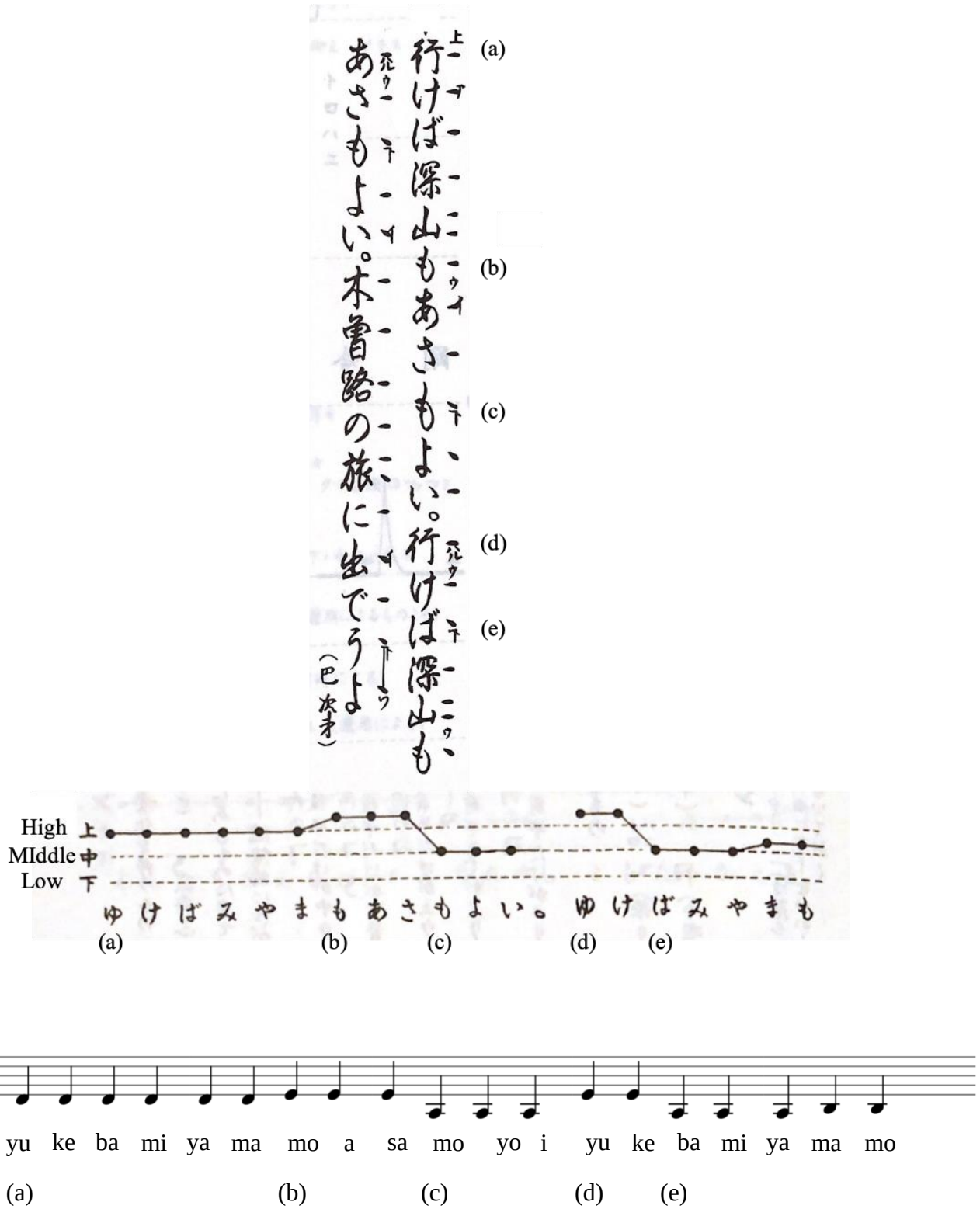


Figure 2-10 An example of yowagin notations for transitions between fundamental pitches (Miyake [1951] 2005). (a) shows that the phrase starts with high. (b) An *uki* transitions from high to ukion, (c) A *sage* shifts the pitch to middle. (d) A new phrase begins at *ukion* with *haru* and *uki* notations. (e) A *sage* notation lowers a pitch to middle.

Transitions of pitches involving nonfundamental pitches are described in Table 2-2. For any transition from high to very high, *ukion* always appears before going to very high. On the other hand, a transition from very high to high can occur directly. These transitions, involving very high, are often used to enhance the melody and expressiveness, and its notations are shown in the first two rows in Table 2-2 (Miyake [1951] 2005, 18–19). The very low is one step lower than the low, usually appearing at the end or the beginning of a phrase (Miyake [1951] 2005, 19). There are several different notations used to shift a pitch to very low depending on where it appears.

Table 2-2 Common notations for transitions between nonfundamental pitches in *yowagin* (Miyake [1951] 2005, 1–5)

Name	Notation Image	Description
kuru - iri		This notation means that a pitch is ascended to very high from high where *kuru* appears and continues until the *iri*. This *iri* notation here means the end of the very high pitch, unlike the general usage mentioned in Table 2-1.
iri		This single notation in the middle of the high indicates that only one syllable will ascend to very high, and then return to high.
ryo		This notation means descending a pitch to very low.
ryo		Only the first syllable will be lowered to very low pitch if this notation appears at the beginning of a phrase at low pitch.

Figure 2-11 shows the usage of *kuru* and *iri*. At (a), the phrase starts with high, which is specified by *jo* notation. The notation of *uki* ascends the pitch from high to *ukion* at (b). The very high pitch starts from the *kuru* notation at (c) and continues until *iri* notation appears at (d). Another example begins at high pitch denoted by *haru* notation, and the pitch is ascended to *ukion* at the next syllable (e). The *iri* notation appearing at (f) further ascends the pitch to very high for only one syllable, then back to the original pitch of high. At (g), a pitch is transitioned to *ukion*.

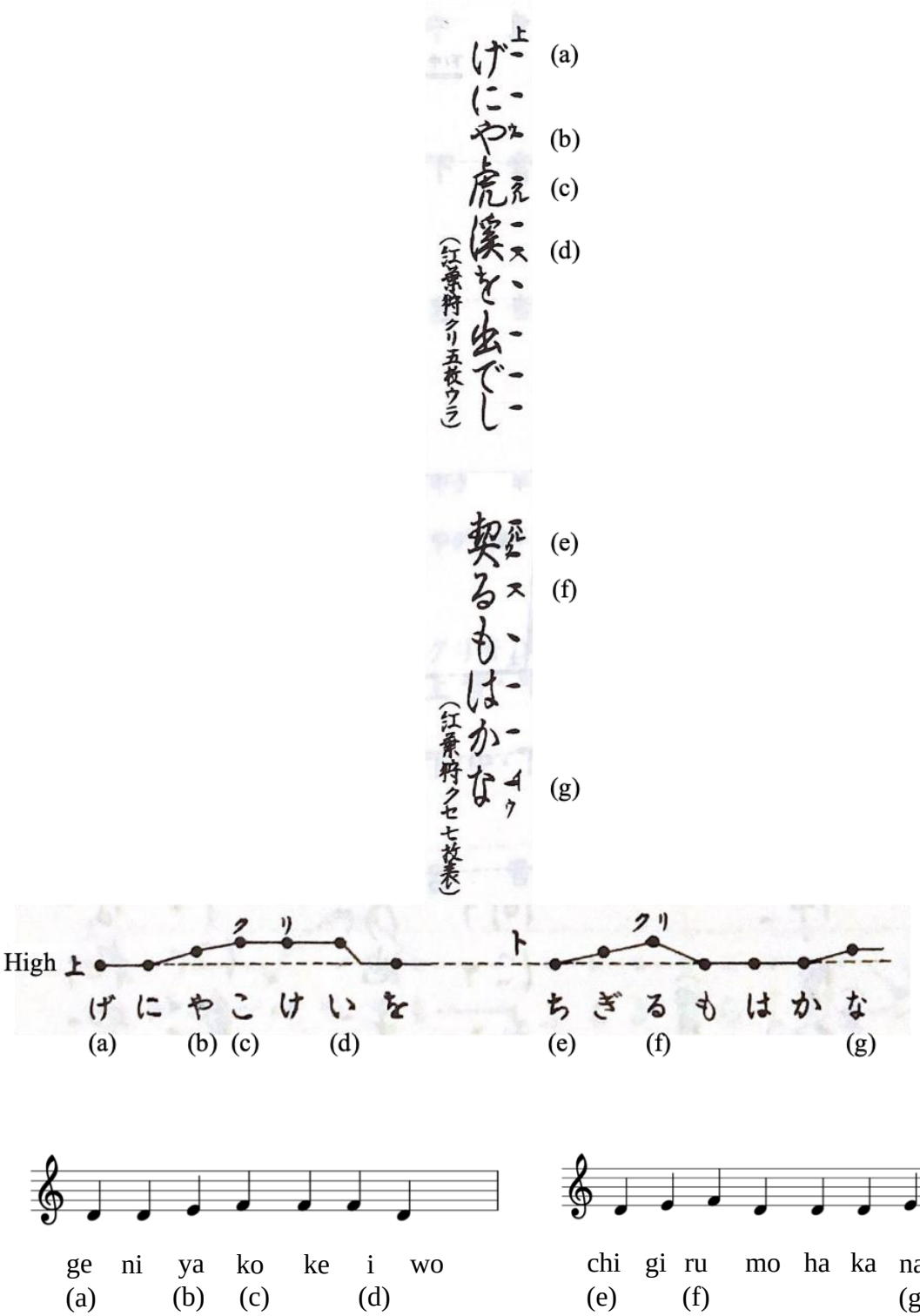
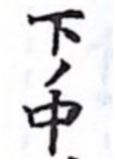


Figure 2-11 Examples of *yowagin* notations for transition with very high (Miyake [1951] 2005). (a) shows that the phrase starts with high. (b) An *uki* transitions from high to ukion. (c) The *kuru* notation shifts the pitch to very high. (d) *iri* notation lower the pitch from very high to high. (e) the new phrase starts with high, and the second syllable becomes *ukion* with *uki*. (f) *iri* notation shifts the pitch to very high for one syllable. (g) the pitch is transitioned to *ukion.*

2.1.4 Tsuyogin

Table 2-3 shows several common notations for the expression of pitch transitions in *tsuyogin*. Although some notations have the same meaning as *yowagin*, others have completely different meanings. Figure 2-12 shows an example of *tsuyogin* from a Noh play called Tamura. This passage starts at low-mid at (a) and continues until the transition to low at (b), yet the sung pitch stays same according to rule 2. The pitch is then ascending to middle at (c) before descending to low-mid at (d) and low at (e).

Table 2-3 Common notations for transitions in *tsuyogin* (Miyake [1951] 2005, 1–3)

Name	Notation Image	Description
low (*ge)*		This notation means to sing at low if this notation appears at the beginning of sentence.
low-mid (*ge-no-chu*)		This notation means to sing at low-mid, which is same pitch as low.
middle (*chu*)		This notation means to sing at middle.
high (*jo*)		This notation means to sing at high, which is the same pitch as middle.
haru or *hari*		This is another way to describe high pitch. This notation appears when transitioning from middle to high and denotes shifting a pitch from middle to high.

sage		This notation appearing at the middle of a sentence means to lower one pitch. However, the notation appearing while singing at middle pitch indicates that more than two syllables are lowered, then back to middle.
sage		In the transition from middle to low-mid to low in *tsuyogin*, this notation can be used to denote low-mid if the low-mid occurs for only one syllable. Then, the next low would be expressed using *otoshi* notation.
otoshi or *osae*		Unlike *yowagin*, this notation can be used for several different ways. 1. This notation appearing in middle pitch means to lower it to low-mid for two syllables. 2. This notation appearing in middle pitch at the end of phrase means to lower it to low-mid for the last syllable. Then, the next starting phrase also continues at low-mid. 3. As stated in the description of *sage*, this notation can be used to denote low after low-mid in the transition from middle to low-mid to low.
iri		This notation means that one syllable will be in the higher pitch.
uki		In *tsuyogin*, this notation can be used to transition from low to low-mid.

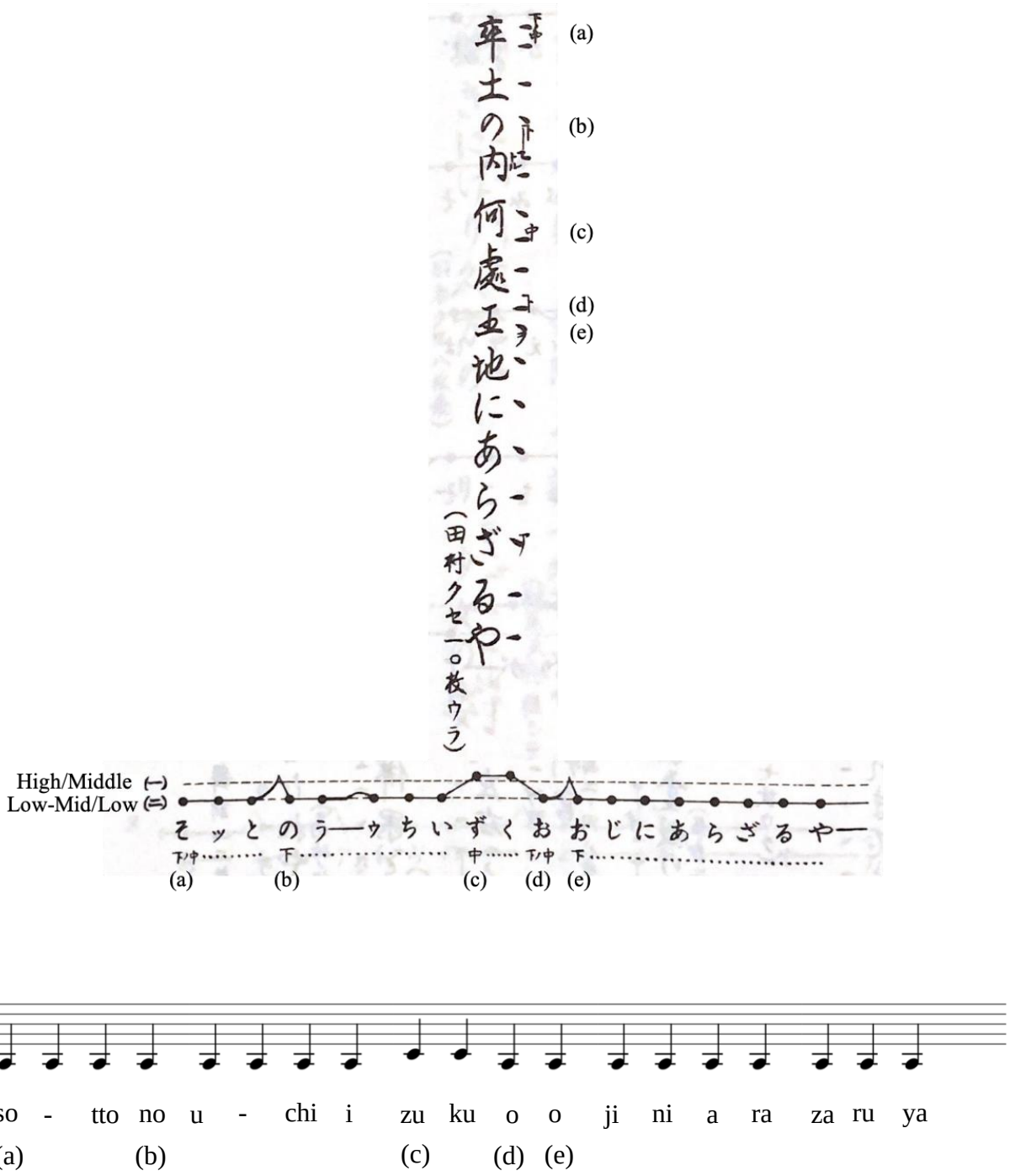


Figure 2-12 An example of *tsuyogin* notations (Miyake [1951] 2005). (a) the phrase starts at low-mid with *ge-no-chu* notation. (b) *sage* notation lowers from low-mid to low, but the sung pitch stays the same. (c) The pitch ascends to middle. (d) sage notation lowers the pitch from middle to low-mid. (e) *otoshi* notation further lowers to low, but the sung pitch stays the same.

2.2 Library Cataloging

Noh chant books are often stored in a library, and a classification system at a library can be helpful when searching for these books. A library catalog is a system, which manages the collections of all bibliographic items that exist in a library, where items include many types of information media, such as books, computer files, graphics, and music. To manage all resources, a library uses a classification system that organizes items systematically to help retrieval of requested resources by a user. This system often forms a hierarchical structure with each class consisting of items from the same genre.

The Nippon Decimal Classification (NDC) is a Japanese standard library classification, used by most libraries in Japan. This system is the most commonly used in Japanese libraries: 99% of public libraries and 92% of university libraries in Japan follow this system. (Omagari 2010). This system was originally established in 1929 and has been maintained by the Committee of Classification of the Japan Library Association since 1948 (Fujikura, Hammarfelt, and Gnoli 2020). The latest is the 10th edition (NDC10) published in 2014.

The reason for its popularity can be traced back to the postwar period following World War II. Almost all Japanese libraries followed their own classification system before NDC started to appear. The system was reconstructed in the postwar period as NDC gained its popularity by adaption to school libraries and usage for the classification of Japanese books at the National Diet Library, which is the legal depository of books in Japan. Since then, the number of Japanese libraries to adapt

their system to NDC increased significantly, and currently, almost all Japanese libraries use the NDC system (Fujikura, Hammarfelt, and Gnoli 2020).

For the classifications with NDC, three-digit numerals with or without a decimal point are used. The first level numeral is called the Main class, which represents the main category of a book as shown in Table 2-4, according to the newest NDC10 (Japan Library Association n.d.).

Table 2-4 the Main classes in NDC classification system (Japan Library Association n.d.).

Main class numeral	Class
0xx	General works
1xx	Philosophy
2xx	History
3xx	Social Sciences
4xx	Natural Sciences
5xx	Technology
6xx	Industry
7xx	Arts
8xx	Language
9xx	Literature

The second and the third levels are called the Division class and the Section class, respectively, and these further classify the Main class into more detailed sections. Table 2-5 shows that the division of the Arts category, which is category 7xx of the Main class.

Table 2-5 the Division classes for the class 7xx in NDC classification system (Japan Library Association n.d.).

Division class numeral	Class
70x	Fine Art
71x	Sculpture: Plastic Art
72x	Painting: Pictorial Art
73x	Engravings
74x	Photography and Photographs
75x	Industrial Art
76x	Music
77x	Theater
78x	Sports and Physical Training
79x	Accomplishments and Amusements

At the National Diet Library, Noh chant books are often classified into NDC: 768 with this classification as shown in Table 2-6, which is designated to Japanese music within the Music category (760). On the other hand, some chants appear to be categorized into NDC: 773 as shown in Table 2-7, which category means Noh or Kyogen within the Theater category (770). The boundary between the categories is quite ambiguous since Noh has both musical and theatrical aspects. The category 768 includes many Japanese music-related books, such as *koto*, *biwa*, and even Kabuki chant. On the other hand, category 773 consisting of any books related to Noh and Kyogen theater includes not only chant books, but also many other categories, such as a book for dancing, a guidebook for watching Noh performance, and even Noh-

related research. Therefore, with these classes, the categories of NDC are not narrow enough to find only Noh chant books.

Table 2-6 the Section classes for the class 76x in NDC classification system (Japan Library Association n.d.).

Section class numeral	Class
760	Music
761	Musicology
762	History of music
763	Musical instruments. Instrumental music
764	Instrumental ensembles
765	Religious music. Sacred music
766	Dramatic music
767	Vocal music
768	Japanese music
769	Theatrical dancing. Ballet

Table 2-7 the section classes for the class 77x in NDC classification system (Japan Library Association n.d.).

Section class numeral	Class
770	Theater
771	Stage. Direction. Acting
772	History of theater
773	Noh play and Noh comedy
774	Kabuki play
775	Other theaters. Stage
776	N/A
777	Puppetry
778	Motion pictures
779	Public entertainments

In addition, some Noh chant books are found to be classified into NDC: 768.4 at the National Diet Library, which is a class for Noh chants, as shown in Table 2-8. However, although this class is supposed to be the designated class for Noh chant, many misclassifications occurred for this specific class. For example, Figure 2-13 is a piece of Joruri, which should be classified into the class of Joruri, NDC: 768.5. However, this item is found in the class of Noh chant, NDC: 768.4, as shown in Figure 2-14 (National Diet Library 2019). It was possible to detect this misclassification since I reviewed each item in the class of Noh chant one by one, with dozens of examples of such misclassifications. It is possible that similar misclassifications have occurred for Noh chant books and that there may be some found in classes other than 768, 768.4,

or 773. However, even if this were not the case and all Noh chant books could be found in those three classes, it would still be difficult to find them because of the high number of non-Noh-chant books within these classes.

Table 2-8 the subsection classes for the class 768.x in NDC classification system (Japan Library Association n.d.).

Section class numeral	Class
768.1	Japanese musical instruments
768.2	Gagaku
768.3	Biwa songs
768.4	Noh chants
768.5	Joruri, Shamisen, Gidayu-bushi
768.6	Koto pieces
768.7	N/A
768.8	Hayashi
768.9	Rogin

Figure 2-13 A page from a Joruri book (Maeda 1913)

Figure 2-14 A misclassification of a Joruri book from Figure 2-13 into NDC:768.4 (National Diet Library 2019).

Due to the lack of a single class into which all Noh chants could be correctly classified within the NDC system as well as the presence of non-Noh-chant items and even outright misclassifications within the correct classes, there are difficulties in finding Noh chant books. In order to find Noh chant books automatically from multiple classes or even classify them automatically instead of doing it manually, this research can be applied to help those processes.

2.3 Technological Literature Review

This section introduces technologies related to this research. First, literature on automatic book classification is reviewed, then automatic document classification as well as a subcategory, script identification, will be reviewed. Then, the research area called semantic segmentation will be briefly introduced and used in the subsequent section on document analysis for musical documents (Brazil, Yin, and Liu 2017). The framework used for musical symbol classification, *Gamera*, will be introduced in the symbol classification of musical documents. Gamera is an open-source framework for building document analysis applications written in C++ and Python, which enable a user to process document, such as symbol segmentation and classification, without formal technical background (Droettboom, MacMillan, and Fujinaga 2003). Lastly, I will discuss decision tree classifier, which will be used to determine whether a page belongs to a Noh chant book or not.

2.3.1 Automatic Book Classification

Automatic book classification and document classification have received increased interest in recent years, especially in a library environment (Desale and Kumbhar 2013). Chiang et al. first implemented convolutional neural networks to analyze book covers as well as perform natural language processing on the title (Chiang et al., 2015). Iwana et al. attempted to classify books using only visual aspects with AlexNet analyzing and classifying 57,000 samples from 30 genres, and achieved 40.3%

accuracy (Iwana et al., 2017). Another approach was done based on the same dataset introduced by Iwana et al, which used more powerful image recognition models, such as NASNet, Inception ResNet v2, and ReNet-50 to achieve a highest accuracy of 55.7% (Lucieri et al., 2020). Although the classification of books by the cover can be applied to modern literature, it is not sufficient for ancient and medieval texts which in addition to being in various states of deterioration can have a much more varied set of formats such as scrolls.

2.3.2 Document Classification

Document classification is a field of research that is concerned with the automated identification of a document to its corresponding categories, such as technical papers, business letters, and magazines. The document to be classified can take many forms of expression, such as text, images, or music scores. Critical to the classification of documents are underlying features: identifiable characteristics observed in a document that can take the form of image, structural, or textual features (Chen and Blostein 2007).

2.3.2.1 Features

Image features are visual elements within a document either extracted directly from an image or after an image has been segmented. Examples of image features are, for instance, the density of black pixels in a region for the former approach and the number of horizontal lines in a segmented block for the latter approach. In

general, there are two types of image features: global image features and local image features. While global image features are visual elements extracted from a whole image, local image features are extracted from regions of an image. An example of the use of image features is Shin, Doermann, and Rosenfeld (2001), who worked on the classification of documents based on visual similarities to retrieve documents, using image features, such as column structures, the density of content areas, and connected components. Similarly, in the research by Bagdanov and Worring (2001), various image features, such as global image features, zone features, and text histograms were used to classify business documents from trade journals and product brochures.

Structural features, which are global features, consider the structure of document images, such as relationships between objects on a page. These features can be obtained from logical or physical layout analysis. For example, XY-Tree representation has been applied to several works using physical layout analysis (Diligenti, Frasconi, and Gori 2003; Baldi, Marinai, and Soda 2003; Cesarini et al. 2001; Appiani et al. 2001), in which a whole document is split into several regions where each region is associated with a tree node.

The textual features include frequency, weights of keywords, or index terms, which can be used alone or together with image features. These textual features can be extracted directly from an image (Shin, Doermann, and Rosenfeld 2001) or by using the output of Optical Character Recognition (OCR) (Ittner, Lewis, and Ahn 1995). Shin, Doermann, and Rosenfeld (2001) used unsegmented bitmap images to

directly extract image features, including the density of content area, statistics of features of connected components, column and row gap, and relative point sizes of fonts. For the extraction of document image features from OCR results, the OCR results can be noisy; there are a few studies that have been done to compensate for these OCR errors, such as n-gram-based text categorization (Cavnar and Trenkle 1994) and morphological analysis (Junker and Hoch 1997).

After obtaining the features from a document, class models can be trained using those extracted features, where class models define the characteristics of the document and can take many different forms, such as grammars, rules, and decision trees (Chen and Blostein 2007).

2.3.2.2 Classification

There are several types of classification, including statistical pattern, structural pattern, and knowledge-based classification. In this section, each of those types will be briefly discussed.

For the statistical pattern classification, many classification techniques have been developed, such as decision trees, Neural Networks, and Hidden Markov Models (HMM). A decision tree is a supervised learning model which consists of many nodes at which simple decisions are made, leading to a tree structure, which technique was applied for document classification by Shin et al. (2001). Neural Networks have been used for recognizing document patterns in research by Cesarini et al. (2001) and

Heroux et al. (1998). Hidden Markov Model is a technique for sequence modeling, and this has been used by Hu et al. (1999) for document classification.

Recently, many document classification studies have applied convolutional neural networks (CNN). CNN was applied to a document classification study by Kang et al. (2014) for document classes defined by the structural similarity, and they reported that error rates were reduced in comparison to traditional feature-based approaches. Harley et al. (2015) introduced Ryerson Vision Lab Complex Document Information Processing (RVL-CDIP), a dataset of 400,000 documents from 16 classes, and applied the AlexNet CNN model to achieve 90% accuracy for document classification. Later, Das et al. (2018) followed the work by Harley et al. but changed their CNN model to VGG-16 architecture, resulting in obtaining higher accuracy.

Structural pattern classification considers physical document components, such as titles, tables, and sections. In earlier work, Geometric Trees were used to classify business letters with their physical layout (Dengel and Dubiel 1995). Similarly, in the work by Baldi et al. (2003), k-Nearest Neighbor (kNN) was used to classifying documents in which the similarity of documents was computed using tree-edit distance.

Knowledge-based document classification requires experts on classifying documents using a set of rules a hierarchy of frames, which can be constructed either manually or automatically where the manual approach only performs in a way that they are programmed to do with knowledge from domain experts (Taylor et al. 1995; Esposito et al. 2000).

2.3.3 Script Identification

In order to accurately process a document with a suited model, the type of document needs to be determined. A subdomain of document image classification is script identification. This is a process to identify a type of script before using a document analysis algorithm or character recognition, and it can be used to determine a suited algorithm for further document analysis (Ghosh et al. 2010). For example, OCR is one of the oldest and most-investigated research fields. However, as Ghosh et al. (2010) mention in their paper, OCR features vary from script to script consisting of different characteristics of structural properties and style, resulting in a difficulty of recognizing different script characters with a single OCR module. As such, features for recognizing English alphabets are generally different from features for Chinese logograms. Therefore, script identification plays an important role in such situations to identify a type of script to process further document analysis with a suited model, such as the OCR model. Script identification has become an important field of research and applied to many problems including automatic storage, document image retrieval, video indexing and retrieval, and document sorting (Ghosh et al. 2010). This field of research was started by Spitz (1994) who conducted on extensive research on automatic script identification. Also, early review papers were written by Ghosh and Shivaprasad (2000) and Pal (2006).

There are mainly two broad categories in script identification: one approach is called visual-appearance-based technique and the other is called structure-based technique. The former approach analyzes the visual appearance of a script without

analyzing its character patterns. Scripts usually have different appearances due to many factors, including script types, the shape of individual characters, and how those characters are formed into words and sentences; such differences are noticeable even at a glance by a casual observer. One early work was done in Wood et al. (1995), which used both vertical and horizontal projection profiles on a script image to determine printed scripts written in three different languages: the Roman alphabet, Chinese text, and Arabic text. Tan (1998) used Gabor function-based texture analysis for script identification for printed Chinese, Latin Greek, Russian, Persian, and Malayalam characters where texture features were extracted from text blocks using a 16-channel Gabor filter. However, there was a disadvantage to this method; extracted text blocks did not have consistent character spacing. To overcome this issue, Peake and Tan (1998) further continued their research by introducing preprocessing steps to obtain text blocks with uniform spacing, which include text-line location, outsized text-line removal, spacing normalization, and padding. In their research, grey-level co-occurrence matrices (GLCMs) were used to extract features in addition to a Gabor filter. Pan, Suen, and Bui (2005) proposed utilizing steerable Gabor filters to identify scripts in machine-printed texts with the advantage of reducing the computational cost in comparison to previous research and extracting rotation-invariant features that can discriminate scripts containing characters with similar shapes. This utilization was limited to machine-printed documents only since various discrepancies in handwritten text, such as writing style, character size, and spacings make recognition challenging. For recognition of documents with

handwritten texts, the preprocessing step before utilization of the Gabor filter was found to be helpful (Singhal, Navin, and Ghosh 2003).

While all the research described above works at a page level, the implementation of script identification at the word level was achieved in several works (Jaeger, Ma, and Doermann 2005; Dhanya, Ramakrishnan, and Pati 2002; Dhanya and Ramakrishnan 2002; Pati et al. 2004; Pati and Ramakrishnan 2006). Ma and Doermann (2003) applied the Gabor filter to extract features from each word in bilingual scripts and applied several classifiers, whose architectures were based on support vector machine (SVM), k-nearest neighbors (KNN), weighted Euclidean distance, and gaussian mixture modeling (GMM).

While the recognition approach differs depending on the region level in documents as mentioned above, there are also differences depending on the type of document. The majority of studies on the topic of script identification focuses on materials that are either printed or handwritten scripts, but some research addresses hybrid documents, which is a combination of printed and handwritten scripts (Ubul et al. 2017). There are various ways to approach analysis on printed and handwritten documents. The analyses for printed documents can be conducted at the page level (Hochberg et al. 1997; Ding, Lam, and Suen 1997; Chaudhuri and Pal 1997; Peake and Tan 1997), text-line level (Pal and Chaudhuri 1999; Chanda, Pal, and Kimura 2007; Padma and Vijaya 2009; Ferrer, Morales, and Pal 2013), word level (Dhanya, Ramakrishnan, and Pati 2002; Dhandra et al. 2007; Jaeger, Ma, and Doermann 2005), and character level (Pal and Sarkar 2003; Rani, Dhir, and Lehal 2013). In the

research conducted by Chanda et al. (2007), an SVM-based model was successfully applied to identify printed languages at text-line level, such as English and Japanese. Later, the word-wise identifications were achieved for English, Sinhala, and Tamil (Chanda et al. 2008), and English, Devnagari, and Bengali (Chanda et al. 2009).

Likewise, handwritten research can also be conducted from broader ranges, such as page level (Hochberg et al. 1997; Hiremath et al. 2010; Ghosh and Shivaprasad 2000) and text-line level (Namboodiri and Jain 2002), to narrower ranges, such as word level (Roy, Pal, and Chaudhuri 2005; Roy and Majumder 2008; Zhou, Lu, and Tan 2006; Dhandra and Hangarge 2007; Roy, Alaei, and Pal 2010; Roy, Das, and Obaidullah 2011; Obaidullah, Roy, and Das 2013) and character level (Pal et al. 2007; Razzak, Hussain, and Sher 2009). For the research by Pal et al. (2007), a modified quadratic classifier was proposed for the recognition of handwritten numerals in six different Indian languages. Handwritten documents are more challenging in comparison to printed documents since there is much more variance in characters (Ubul et al. 2017).

2.3.4 Semantic Segmentation

Semantic segmentation is an image analysis method, which is used in many fields, including self-driving vehicles (Tseng and Jan 2018), pedestrian detection (Brazil, Yin, and Liu 2017), and computer-aided diagnosis (Zhu et al. 2016). Although many approaches for semantic segmentation were proposed before deep learning, such as random forest and visual grammar, the usage of deep learning, Fully Convolutional

Network (FCN) proposed by Long, Shelhamer, and Darrell (2015) increased the accuracy for segmentation compared to any previous approaches. Figure 2-15 shows different image analysis methods, which was obtained from the work by Hao, Zhou, and Guo (2020), and the original image is shown in Figure 2-15 (a). As Figure 2-15 (c) shows, an image classification algorithm can recognize objects in an image and predict the corresponding label, such as a car or road in this case. An object detection algorithm can recognize objects in an image as well, however, it also estimates the locations of each object. As Figure 2-15 (d) shows, generally detected objects are annotated with bounding boxes. Semantic segmentation goes one step further and partitions each region of the object accurately at pixel level by delineating their boundaries, as shown in Figure 2-15 (b). There are also two recent approaches derived from semantic segmentation, which are instance segmentation and panoptic segmentation. The goal of instance segmentation is to recognize each object in an image as a distinct entity, and thus every object is labeled differently, each of which represents an instance in Figure 2-15 (e). Panoptic segmentation predicts both a semantic label and an instance label at every pixel level, as shown in Figure 2-15 (f).

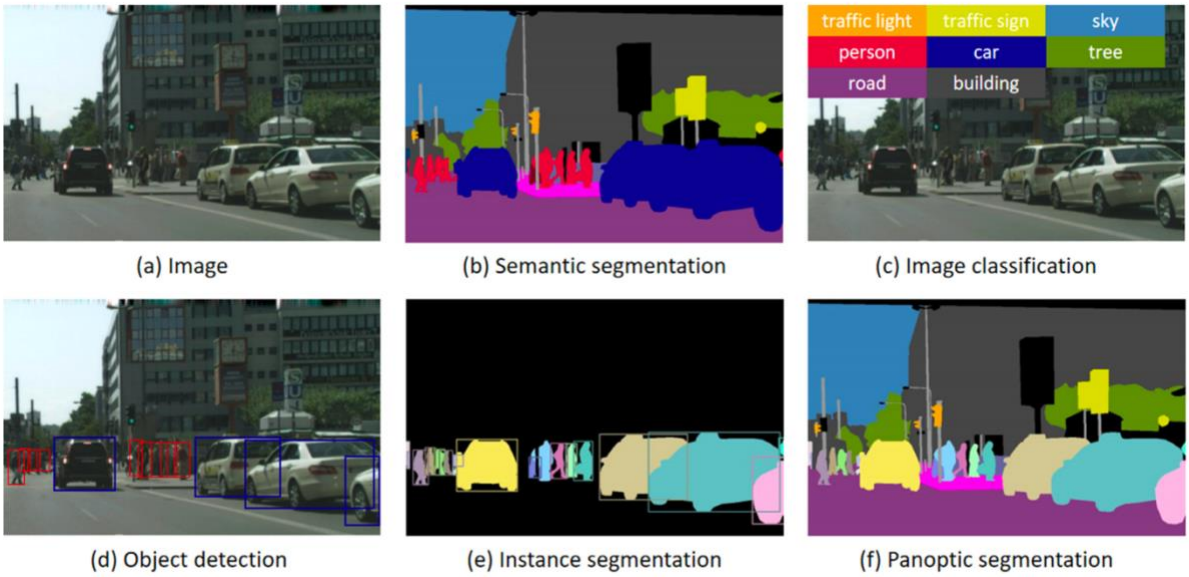


Figure 2-15 A comparison of image analysis tasks. (Hao, Zhou, and Guo 2020, 303)

2.3.5 Document Analysis of Musical Documents

Optical Music Recognition (OMR) is a field of study regarding the automatic identification and encoding of musical content from scanned image data, with document analysis playing an important role in the development of OMR systems (Castellanos et al. 2018). Because of the complexity of musical notation and the variety of information included within musical documents, such as staff lines, musical notation, lyrics, or artwork, processing these images can be difficult, especially when compared to related domains of research, such as OCR. Many OMR pre-processing approaches have been proposed to address these challenges, including binarization (Howe 2013), which separates the background from the foreground of an image, text region segmentation (Burgoyne et al. 2009), and staff-line removal (Montagner, Hirata, and Hirata 2017).

Calvo-Zaragoza et al. (2018) worked to separate different zones of an image before applying identification of musical content. This was achieved with the use of CNN to perform classification at the pixel level, meaning that each pixel in an image was classified into categories such as background, note, text, or staff-line. For the input of the CNN model, they used a rectangular region from an original image, centered at the pixel of interest, where the size of window varies depending on the size of image. This is because the neighborhood provides contextual information that can be utilized during the classification. They were able to use the results of the pixel classification to separate different layers from the image. Since this classification occurs at pixel level, even small components were able to be identified correctly. Although the above method achieved high performance for the classification, the cost of its computation was high since it needed to classify every pixel on an image. In order to overcome this issue, Castellanos et al. (2018) incorporated the use of deep selectional autoencoders to classify an image patch by classifying many pixels simultaneously. A conventional autoencoder is an unsupervised neural network that usually consists of two parts: encoder and decoder. The encoder takes an original image input and reduces the dimension by compressing it into an encoded representation, while the decoder takes the encoded representation of an image and reconstructs the original input. In their work, instead of reproducing an input image, they created a pair of encoding and decoding for each layer, such as background, staff-line, musical notation, text. In this way, each layer was able to be reproduced from an original image input to create pixel

annotations. This approach was also able to achieve similar accuracy to the previous approach, pixel-wise classification.

2.3.6 Musical Symbol Classification

Gamera is an open-source document analysis framework for domain experts who have a deep understanding of certain documents without having a technical background (Droettboom, MacMillan, and Fujinaga 2003). There are five main tasks, including pre-processing, document segmentation and analysis, symbol segmentation and classification, syntactical or structural analysis, and output. The pre-processing includes common image processing, such as noise removal, blurring, de-skewing, contrast adjustment, sharpening, binarization, and morphology. Document segmentation and analysis includes analysis of the document structure. Syntactical or structural analysis is the process of creating a semantic representation of the symbols from a document. The classification results are formatted as XML in the output.

The main components of the Gamera system are segmentation, feature extraction, and symbol classification, which consists of two types of classifiers: interactive classifiers and non-interactive classifiers. Interactive classifiers allow a user to add training examples by annotating a symbol and then testing the results immediately. Non-interactive classifiers do not allow a user to annotate interactively but allows for highly optimized classification. The k-nearest neighbor (KNN) algorithm is used for these classifications where weights are optimized using a genetic algorithm (GA).

KNN is a supervised machine learning algorithm originally developed by Fix and Hodges (1951), which learns from labeled training data to produce labels for unlabeled data. In the training phase, the training data, which is a feature vector, is only stored in a multidimensional feature space with class labels. Given a k value, which is a user-defined value, a new unlabeled data is classified into the class that occurs most frequently among the k nearest training samples. For the calculation of distance, Euclidean distance is a widely used distance measure for continuous variables, as shown below. Another measure, such as Hamming distance, which directly count differences in two strings, could be used for discrete variables, such as text categorization.

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

GA is an algorithm proposed by Holland (1975) and commonly used for optimization and search problems, which idea is based on natural selection and genetics. The key elements of this algorithm are the chromosome representation, selection, crossover, mutation, and fitness function computation. First, an initial population of chromosomes is created with random bitstrings, each of which is computed with a fitness function to obtain fitness values. Then, a pair of chromosomes are selected from the population using their fitness values, which will be used for creating the next generation with a crossover operator and mutation operator. There are several strategies for the selection techniques, including roulette

wheel, rank, tournament, Boltzmann, and stochastic universal sampling (Katoch, Chauhan, and Kumar 2021).

2.3.7 Decision Tree Classifier

Decision tree is a classifier originally developed by Quinlan (1986) and is a widely used approach for various classifications by researchers from fields such as machine learning and statistics. This classifier will be used to determine whether a page belongs to a Noh chant book or not. It has a tree-like structure consisting of nodes, and each node has an if-then rule to split data. The classifier performs prediction by passing a new data from the initial node, called a root node, down to the last node, called a leaf node, which will be the class. Decision trees are binary trees where each non-leaf node will have a function to determine which child node to progress to, as shown in Figure 2-16. For example, in $x = .6$ is passed in, it would reach the internal node as it is it would be greater than .5, and then classified as B as it is less than .7. In the training phase, the split point of the best feature is selected to maximize information gain.

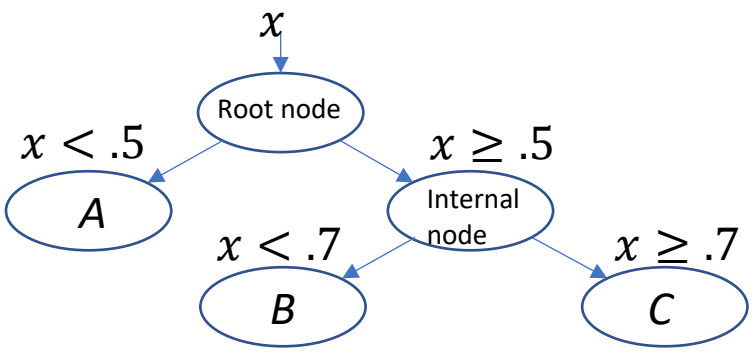


Figure 2-16 An example of a decision tree

2.4 Rodan

Rodan is a web-based workflow engine, which enables a user to interactively perform many tasks for optical music recognition (OMR). This software was developed under the Single Interface for Music Score Searching and Analysis (SIMSSA) project.[1] A large government-funded project to make musical scores searchable online and enables workflows such as that shown in Figure 2-17 (Fujinaga 2019). This project focuses on the digitization of manuscripts with neume notation using two standardized formats of file; the International Image Interoperability Framework (IIIF)[2] and the Music Encoding Initiative (MEI).[3] The digitized manuscript in IIIF format can be fed into the SIMSSA workflow and processed in several stages to obtain OMR output encoded in the format of MEI. The workflow of SIMSSA consists of the following: document analysis, symbol classification, music reconstruction and encoding, and symbolic score generation and correction, analogous to the four processes in Figure 2-18 (Vigliensoni, Calvo-Zaragoza, and Fujinaga 2018). These entire processes are hosted at the Rodan server, and one can use all tasks needed for performing OMR through a web interface.

[1] https://simssa.ca/
[2] https://iiif.io/
[3] https://music-encoding.org/

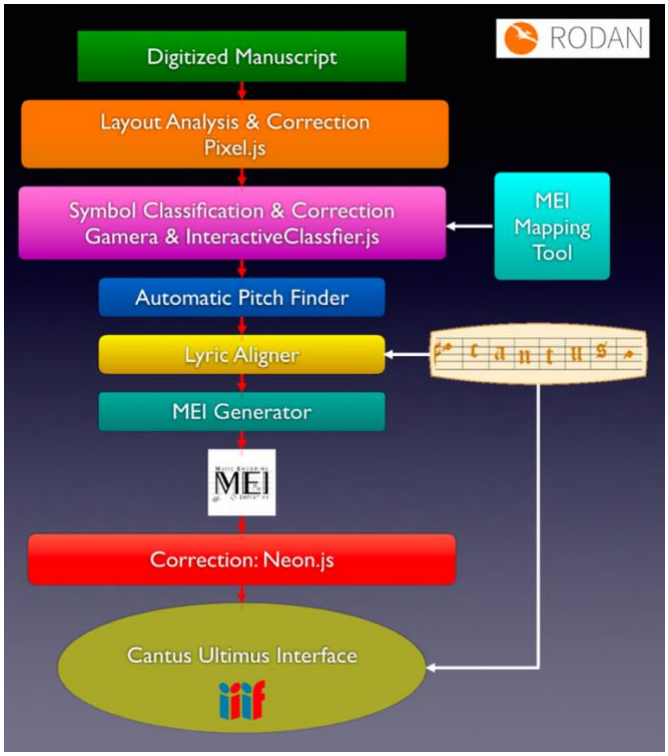


Figure 2-17 An example of optical neume recognition workflow (Fujinaga 2019).

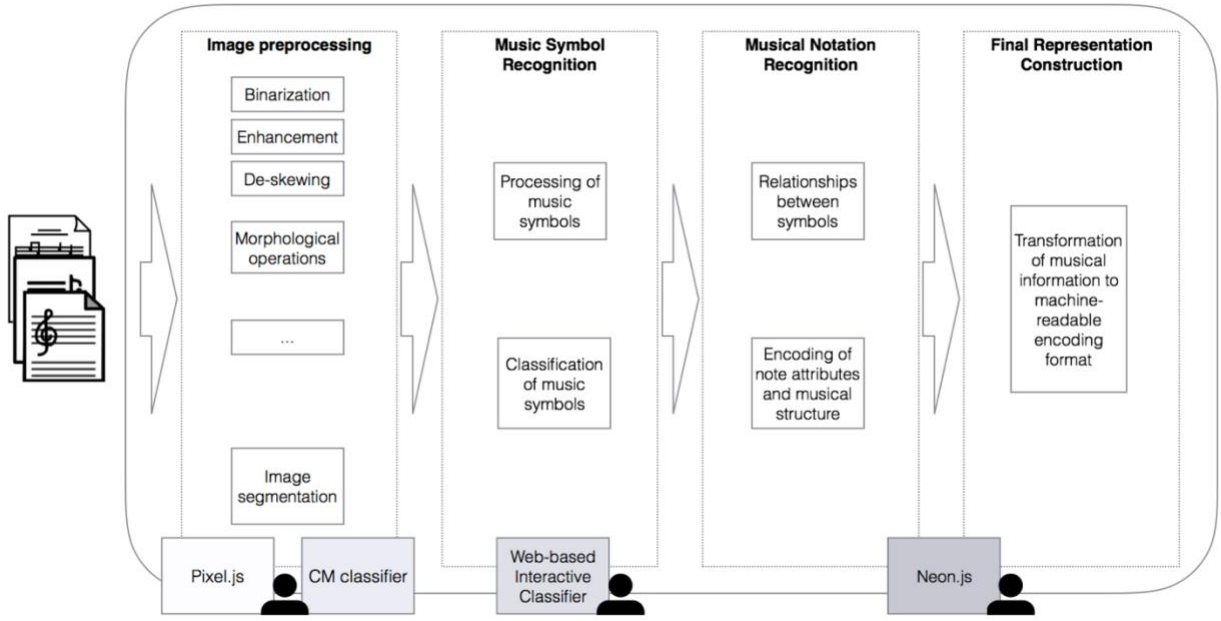


Figure 2-18 An example of end-to-end OMR workflow (Vigliensoni, Calvo-Zaragoza, and Fujinaga 2018). The human icons indicate places require human intervention.

2.4.1 How Rodan is structured

There are four basic concepts in Rodan, which are project, workflow, resource, and job. A project means a group of resources, workflows, workflow runs, and run jobs at the Rodan server. Here, the workflow run is an instance of a workflow, and the history of the workflow's execution can be obtained from the workflow runs. The run job is an instance of a job, and the run jobs shows an execution history of jobs. To use Rodan, a user first need to create a project with a user-defined name. To perform an OMR task, one needs to create their own workflow structure on the interactive console window by serializing jobs. A job is a module to perform a specific task, and there are many types of jobs available at Rodan, such as converting an image format and performing an image analysis to recognize a specific shape on an image. The workflow describes a sequence of jobs with associated resources passing through where input and output of each job are connected; it also measures how the output from each job passes down to the next job. The resource includes all files that are necessary for a project, and these can be either imported to Rodan or automatically generated from an execution of a workflow. The types of files include, but are not limited to, image, classification model, and MEI. The resource can also be assigned user-defined tags for easier categorization and sorting purposes.

2.4.2 Jobs in Rodan

There are mainly four groups of tasks in Rodan: document analysis, symbol classification, music reconstruction and encoding, and symbolic score generation and

correction. The document analysis includes several image analyses tools from generic process, such as image resizing and de-speckling, one of noise reduction process, to more complicated process, such as layer separation. There is a job called Pixel.js, which allows a user to interactively select a region from an image and categorize into classes, as shown in Figure 2-19. For example, a Western music score has musical notes and symbols printed on top of staff lines and text indicating musical expression or lyric if it is a vocal score. In this case, pixels on this image can be categorized into four classes: music elements, text, staff lines, and background. This categorization can be done manually using Pixel.js job in Rodan. The other important jobs are called Patchwise Trainer and Pixelwise Classifier jobs. The former is a job for training a layer separation model while the latter is for performing layer separation using a trained model. If one wants to perform document analysis task on an entire book or manuscript, manually separated layers created with Pixel.js job can be used to train a model with the Patchwise trainer, and the trained model can be used to perform layer separation on different pages from the same book.

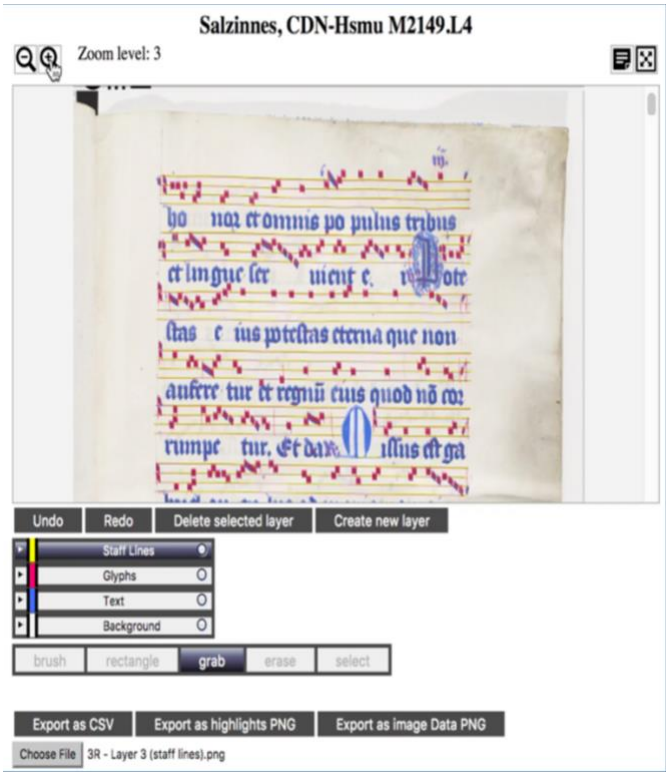


Figure 2-19 A interface of Pixel.js where a manuscript is color coded with different colors. (Fujinaga 2019)

The symbol classification uses layers separated in the document analysis, and those are taken apart into connected components or glyphs and classified into symbols or score elements. This process can be done manually or automatically using jobs called Interactive Classifier or NonInteractive Classifier. Figure 2-20 shows an interface of Interactive Classifier, which allows a user to interactively classify musical symbols. If prepared training data exits or musical symbols already classified manually, one can use those as training data for automatic classification.

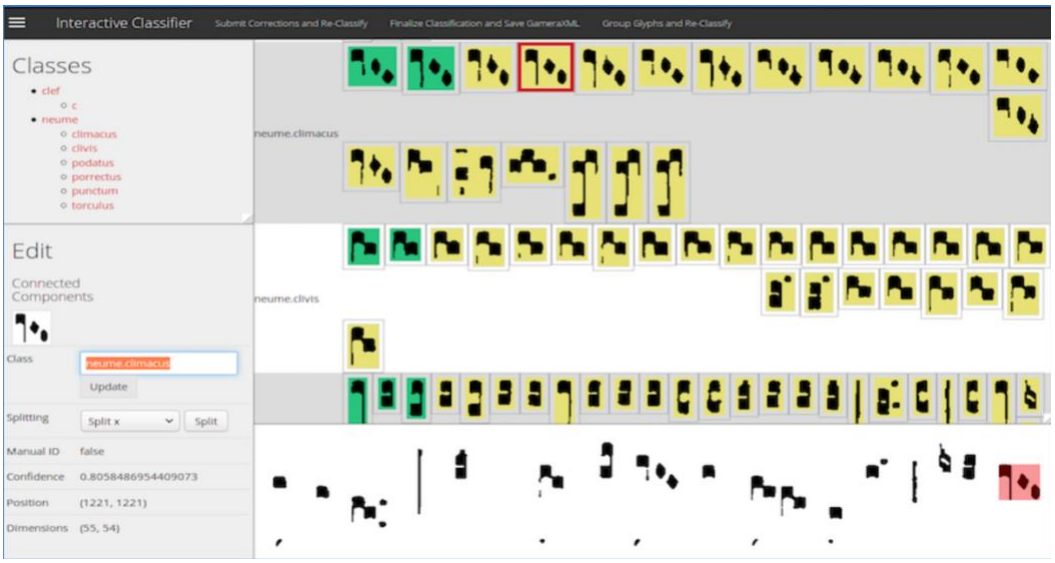


Figure 2-20 An interface of Interactive Classifier. Manually classified neumes are shown in green boxes, while unclassified components are shown in yellow (Fujinaga 2019).

The music reconstruction and encoding can be applied to interpret meaning of musical symbols once all symbols on an image are classified. This includes jobs, such as Miyao Staff Finding to find staves, Heuristic Pitch Finding to find pitch of musical symbol, and Text Alignment to align plain text with optical character recognition (OCR) result.

The symbolic score generation and correction includes job called MEI Encoding, which takes output from the music reconstruction and encoding to obtain results in MEI format. For square-notation score, mistakes occurring on those MEI results can be corrected with a job called Neon.

2.4.3 Models in Rodan

Rodan has many machine learning models to facilitate all functions. For the layer analysis, the Pixelwise Classification job uses Convolutional Neural Network to

classify a region of interest at the pixel level (Calvo-Zaragoza, Vigliensoni, and Fujinaga 2017). The Patchwise Trainer uses Selection Auto-Encoder (SAE) configured with a Fully Convolutional Network (FCN) (Castellanos et al. 2018). An FCN is a network consisting of filters, such as convolution. There are two parts in an SAE model, encoding and decoding. The former consists of a series of convolutional and pooling layers while the latter consists of a series of convolutional and up-sampling layers. The last layer in the decoding has sigmoid activation that predicts a value between 0 and 1 where this selectional level is close to 1 if a model is more confident about the classification of a pixel. The training stage requires the ground truth data to train models for every class, such as background, note, and staff lines.

The jobs for the symbol classification are called Interactive Classifier and Non-Interactive Classifier. Both are functions based on Gamera, using k-nearest neighbor classifier, and used for grouping connected components or classifying musical symbols to user-defined classes ("Gamera Classifier API" 2018). The Interactive Classifier is used during the training stage since manually classified symbols can be added to the training sample for automatic classification in real-time. Afterwards, Noninteractive classifier takes the completed training set from the Interactive Classifier and outputs an XML file that will be used for the decision tree classifier.

Chapter 3 Methodology

In this chapter, the methodology for Noh Chant classification will be explained. The goal of this study is to classify Noh chant books from other Japanese books published around the same era. The key to achieving this classification is *fushi* notation, which is a musical symbol, as discussed in Section 2.1.1. This *fushi* notation is a feature that appears only in Noh chant; other types of Japanese music do not employ the same notation to indicate how to sing. Therefore, machine learning models are used to enable this classification by identifying *fushi* notation in this research. In this methodology, there are three steps for achieving the classification, including data acquisition for Noh chant books, pixel segmentation as well as classification of *fushi* on each page from the books, and lastly, classification of Noh chant books.

The first step, data acquisition, was done to collect document images of Noh chant books as well as other books related to traditional Japanese music. The purpose of this step is to create a balanced dataset containing both classes. A balanced dataset has all classes in the same ratio. In machine learning, it is important to create a balanced dataset to achieve the right accuracy since this ratio directly affects the predictions of a model. This step of dataset creation is necessary since this type of dataset does not previously exist and needs to be created.

Once the dataset is obtained, we will move to the next step, which is the pixel segmentation. In this step, only pixels for musical symbols were extracted from an

acquired document image and separated from the rest of the document components, such as text, picture, and background. These were processed using Rodan, which is a web-based workflow engine for optical music recognition (OMR) (Fujinaga 2019). Rodan includes several OMR processes, but a user may select only those that are required and design a unique workflow with them. This separation was performed using machine learning models on Rodan. Therefore, ground truth data was required and needs to be created for training a model. The trained models were then used to perform extraction of a Fushi notation layer, and those layers were further processed to perform the classification of connected components in the next step.

Each of the connected components in an image was then classified into four categories, *fushi* notation, Noh notation other than *fushi,* text, and noise. The main purpose of this process is to determine how many *fushi* notations appear on a page. While many *fushi* symbols are predicted to be identified in Noh chant books, most of the components in other Japanese books are expected to be classified into classes other than *fushi* and Noh notation. This difference in results was considered when classifying Noh chant books.

3.1 Data Acquisition

The first step is to collect data. Here, the same number of Noh chant data and other Japanese books were acquired from each of NDC classes, which contains Noh chant books. This process is necessary since no pre-existing labeled datasets are available. I ensured that data from each NDC class was sampled uniformly from each

published period while collecting data from each NDC class. To do so, ranges for published years for the NDC classes were subdivided into smaller ranges, and data was sampled evenly between Noh chant and other books from each of the subranges.

A total of 30 books, including 15 Noh chant books and 15 other Japanese books, was obtained from the National Diet Library. The total number of pages was 2,483 pages, consisting of 1,423 pages of Noh chant books and 1,060 pages of other Japanese books. In order to find Noh chant books, there were a few classification numbers and keywords specified when conducting the search. Most Noh chant books are classified with either the code: 768 or the code: 773 with Nippon Decimal Classification (NDC) system (See Section 2.2). NDC: 768 is the class for Japanese music within the Music category, and NDC: 773 is the class for Noh or Kyogen within the Theater category.

From the National Diet Library webpage,[4] each NDC category number was used as a key to obtain books categorized into these classes. Under the Advanced Search tab, in addition to these NDC numbers, Accessibility was specified as "Via the Internet", as shown in Figure 3-1, since we are interested in books available online.

[4] https://ndlonline.ndl.go.jp/

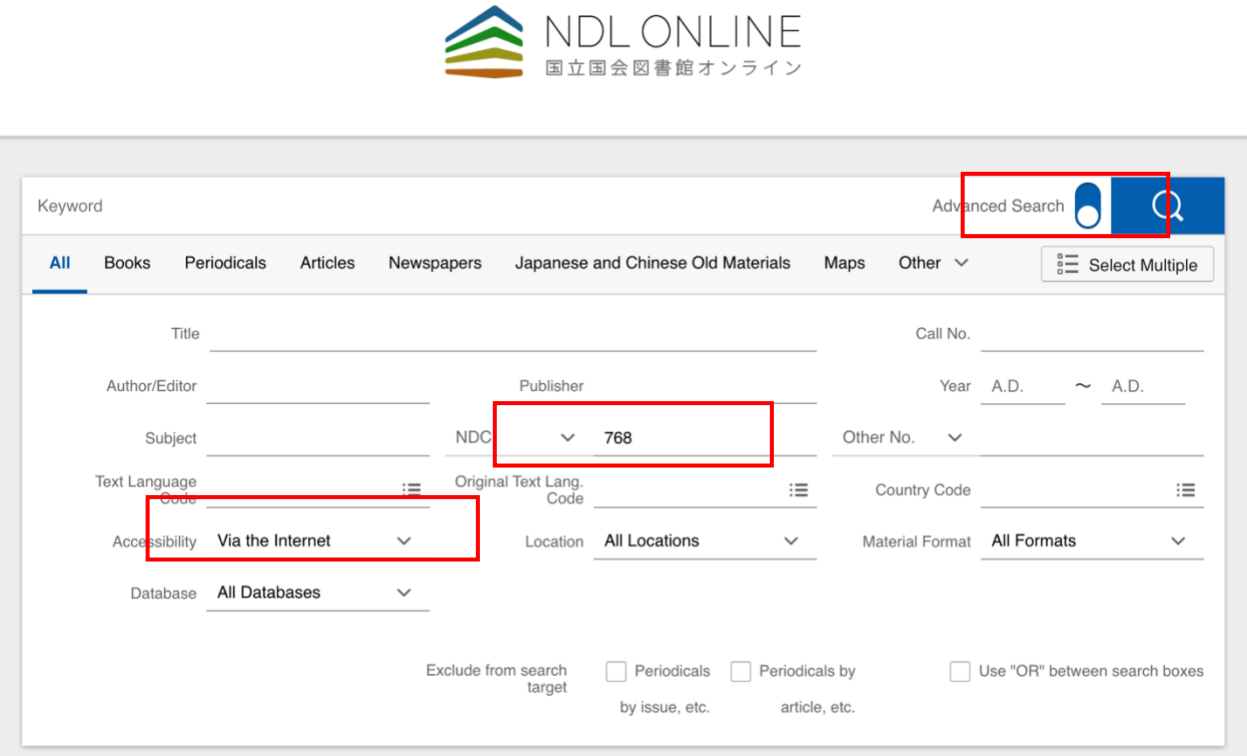


Figure 3-1 The advanced search menu on National Diet Library Online (National Diet Library n.d.).

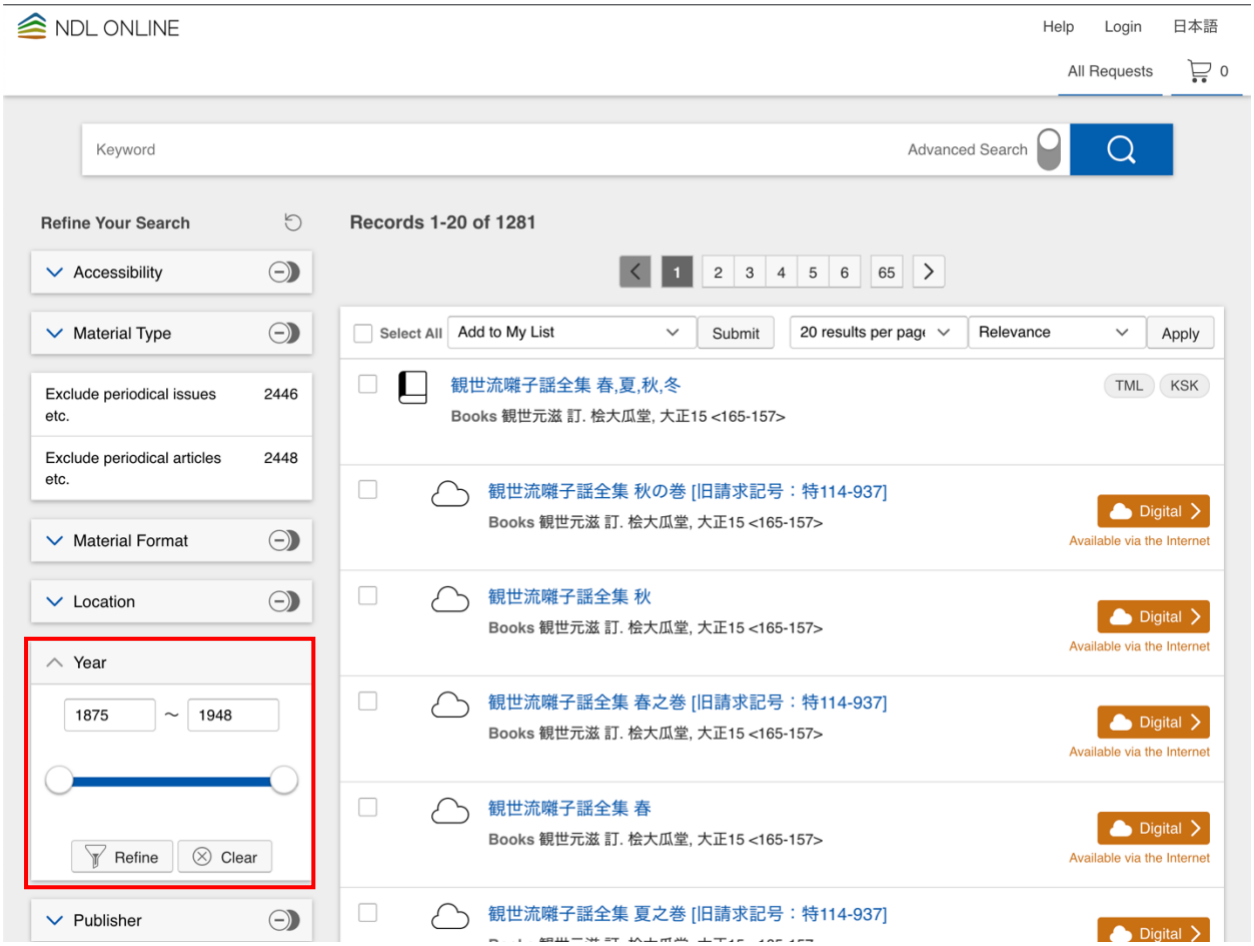


Figure 3-2 The search results on National Diet Library Online (National Diet Library n.d.).

Figure 3-2 shows the search results from Figure 3-1, and the year under the left navigation column was used to specify the time range. For NDC: 768, 1281 books published between 1875 and 1948 were found. The total of 73 years was divided into subranges with 10-year spans; each document for Noh chant and other Japanese books were sampled from each subrange. If no books were found from a specific subrange, data was instead sampled from another subrange. For NDC: 773, 294 documents published between 1700 and 1948 were found. Since only one book was found between 1700 and 1875, the same subranges were used from NDC: 768, each

having 10-year spans, to sample data where the earliest subrange is from 1700 and 1885.

So far, the classification results in the specific NDC category were narrowed with the NDC number, accessibility, and time range. However, even those results contained books other than Noh chants, related to Japanese music or Noh theater. Thus, I also used different keywords, such as publisher or Noh performance names, to further filter the search results to obtain Noh chant books. If today's well-known publishers for Noh chant books, such as Wanya Shoten, Hinoki Shoten, and Nohgaku Shorin, were present in the search result, these publishers were selected from the left navigation column to obtain Noh chant books. However, there were some subranges containing none of these publishers, specifically those before 1915. For those ranges, names of Noh performance were obtained from the Noh program database[5] and used as a keyword to search. If still no Noh chant books were found, each list of books was manually checked one by one to see if any *fushi* notation appeared in a document image. Of course, because only two categories were checked using this approach, these strategies will still overlook many Noh chant books in the library. The books that I was able to find were utilized to create ground truth data for training machine learning classifier.

[5] (Nohgaku kyokai (公益社団法人 能楽協会) n.d.)

3.2 Document Classification

3.2.1 Layer Separation

After the data collection, the pixel segmentation of *fushi* notation was processed using Rodan, a web-based workflow engine for optical music recognition (OMR) (Fujinaga 2019). Rodan contains many jobs used for the OMR process, and a user can select and use only processes needed for one's workflow. In this section, there are three components that I need to achieve results with this application:

- Creating ground-truth data
- Training a layer separation model
- Using the trained model to extract the fushi layer

The model trainings and executions were performed cyclically and incrementally in this process, while continuously increasing the number of training data. The workflow is shown in Figure 3-3, showing the method of creating one page of training data, learning a model using the data, and creating additional training data using the model. The creation of ground-truth data can be achieved using Pixel.js, which is a job with a web-based graphical interface at Rodan. This job enables manual layer separations by selecting each pixel in a document image and classifying it into a distinct class defined by a user. The initial document image was manually annotated using a brush tool and separated into the Noh notation layer and the background as shown in Figure 3-4. By selecting the Noh notation layer and manually masking all Noh notations in a document image, pixels can be selected and registered as the Noh

notation layer, and non-selected pixels were automatically classified as the background layer.

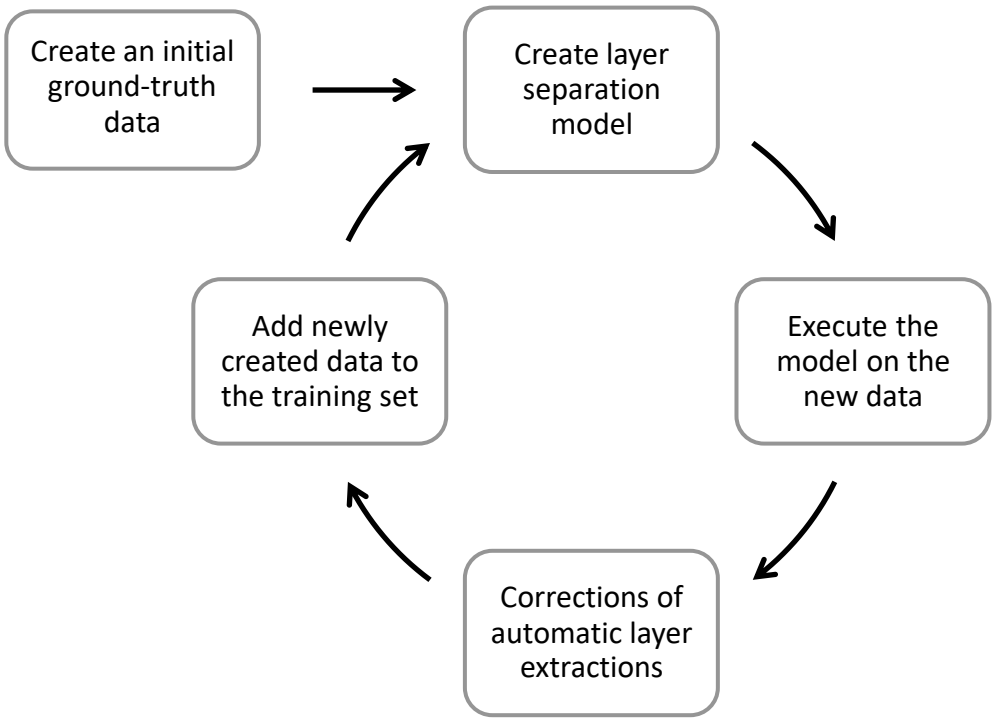
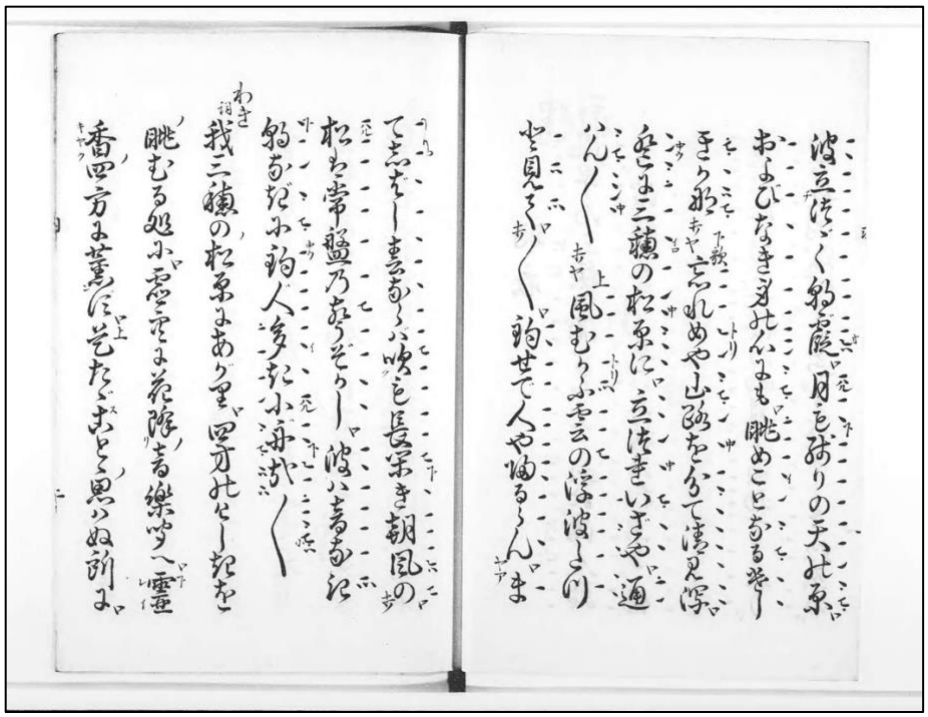
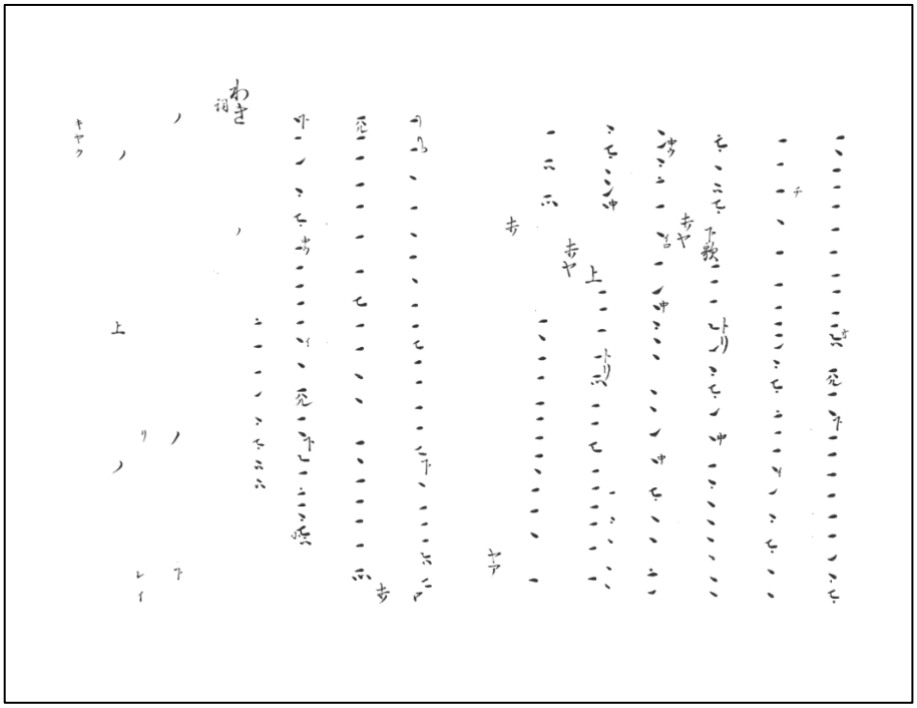


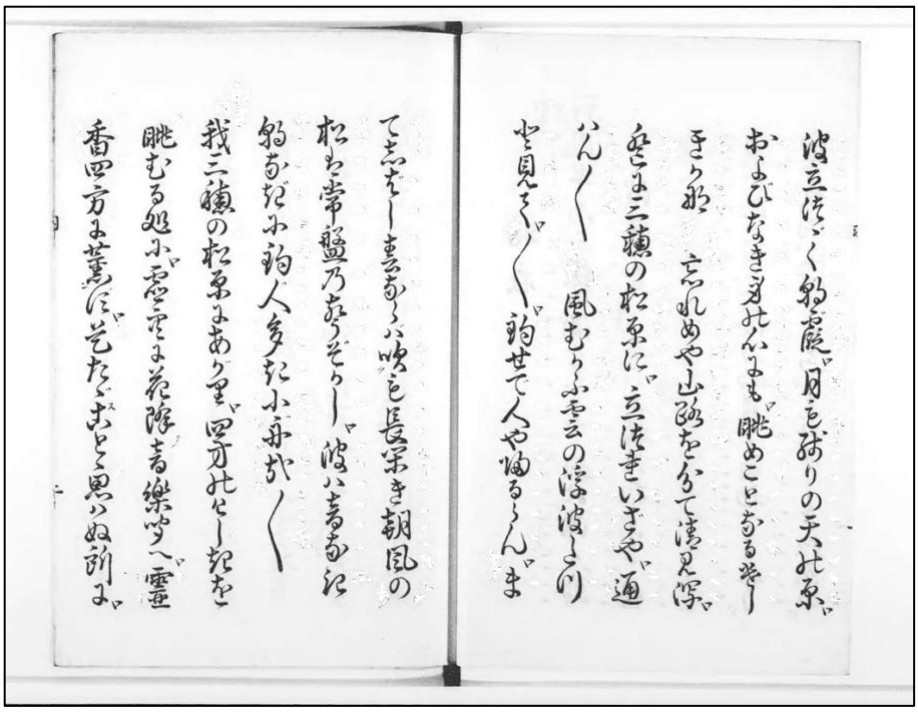
Figure 3-3 The workflow for layer separation



(a)



(b)



(c)

Figure 3-4 The separated layers of a document image of Noh chant. (a) is the original image, (b) is the Noh notation layer, and (c) is the background layer.

Those data were then used for training layer separation models. For each training, two models were created: one for the Noh notation layer and another for the background layer. Afterwards, the newly trained models were used to conduct layer separation on another document image from the sampled data. This results in the separation of a single image into two layers, with often several misclassifications in between, as shown in Figure 3-5. The Pixel job was used to manually rectify these misclassifications.

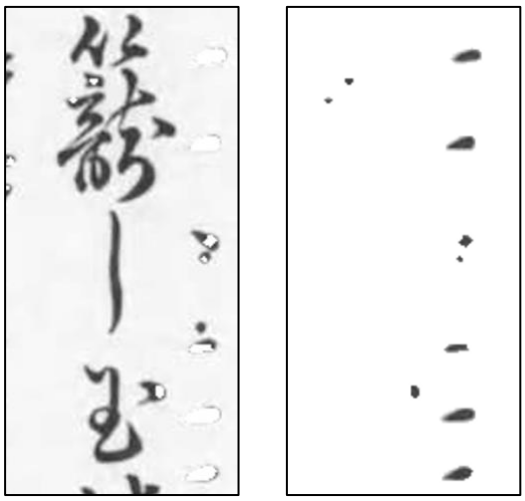


Figure 3-5 An example of misclassification of pixels with Layer Separation model. The left side shows a background layer, and the right side shows a Noh notation layer. This example was obtained during the training process and is not a result from the final model.

The layer separation was performed to extract Noh notation as well as *furigana*, which is a reading aid written in smaller syllabic characters, or other descriptions written in the same region as *furigana*. From Noh chant books, Noh notation as well as *furigana* or other descriptions in the same region were extracted while symbols similar to Noh notation as well as *furigana* or other descriptions in the same region were extracted from other books. In Noh chants, *furigana* is printed next to chant text as shown in Figure 3-6. Similarly, musical symbols and notation are also written in the same region and is often difficult to distinguish from *furigana* even for a human reader. Therefore, the extraction of both *furigana* and musical notation can reduce the complexity for a computer to distinguish between these since *furigana* and musical expression can be similar as shown in Figure 3-6. For books other than Noh chant books, *furigana* also often appeared next to text, which were also extracted as

a Noh notation layer. Figure 3-7 shows non-Noh-chant books with *furigana* where *furigana* is annotated with blue color.

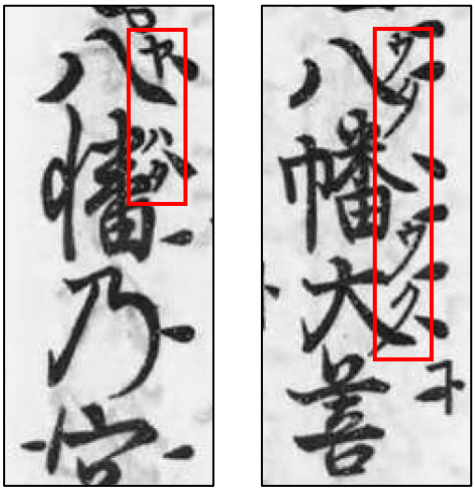
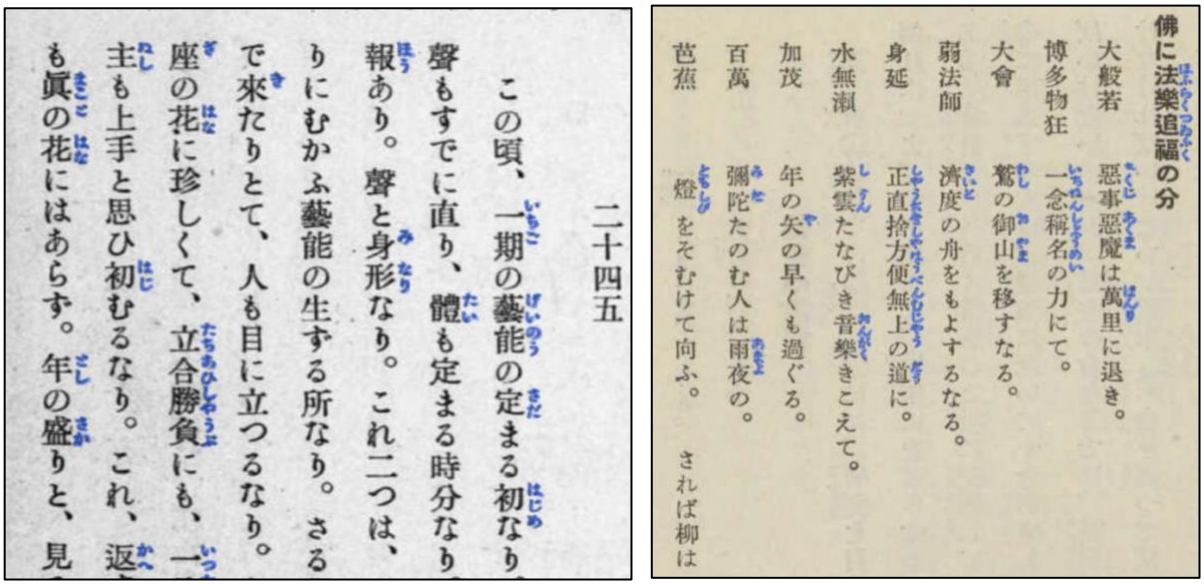


Figure 3-6 The difference between *furigana* and musical notation. The left side is an example of *furigana*, and the right side is an example of musical notations.



二十四五

この頃、一期の藝能の定まる初なり。聲もすでに直り、體も定まる時分なり。報あり。聲と身形なり。これ二つは、りにむかふ藝能の生ずる所なり。さるで來たりとて、人も目に立つるなり。座の花に珍しくて、立合勝負にも、一主も上手と思ひ初むるなり。これ、返も眞の花にはあらず。年の盛りと、見

佛に法樂追福の分

大般若　惡事惡魔は萬里に退き。
博多物狂　一念稱名の力にて。
大會　鷲の御山を移すなる。
弱法師　濟度の舟をもよするなる。
身延　正直捨方便無上の道に。
水無瀬　紫雲たなびき音樂きこえて。
加茂　年の矢の早くも過ぐる。
百萬　彌陀たのむ人は雨夜の。
芭蕉　燈をそむけて向ふ。

されば柳は

3.2.1.1 Pixel.js

As discussed in the previous section, the Pixel job allows a user to manually annotate pixels in a document image. This job takes the following two inputs:

- An original document image
- Automatically generated Noh notation layer by a machine learning model (Optional)

These two inputs must all come from the same document image. The second input is optional, and classification errors in the second input can be corrected while referencing the original document image on the Pixel job. Only the first input above was assigned to annotate Noh notations manually in the initial creation of the ground-truth data. The machine learning model was incorporated afterwards, which outputs extracted layers requiring manual corrections; the second input was used from the second attempt, as shown in Figure 3-8.

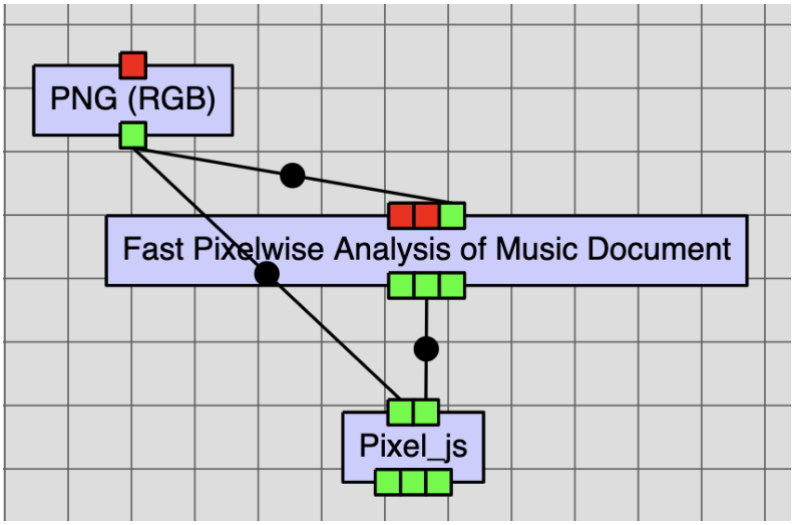


Figure 3-8 A Rodan workflow for the layer extraction with Fast Pixelwise Analysis of Musical Document and Pixel.js.

3.2.1.2 Fast Pixelwise Analysis of Music Document

A job named Fast Pixelwise Analysis of Music Document can be utilized for layer separation since it employs a Convolutional Neural Network to identify and classify each pixel in an image. A region of interest defined within a document image, such as extracted windows, is classified as either the region that belongs to Fushi or background class. Because local information within a single window must be adequate to identify pixels successfully, the extracted windows must be properly sized. Often, a CNN model may be applied to recognize pixel patterns without the need for human feature extraction. Therefore, even though the model was originally designed to process western manuscripts, it is also suitable for analyzing Noh chant books.

This job takes three types of inputs:

- A document image, whose pixels are to be classified

- A model for extracting the Noh notation layer
- A model for extracting the background layer

There are three types of output:

- A Noh notation layer, which contains only pixels of Noh notations as well as *furigana* from the document image
- A background layer, which contains all pixels that were not included the Noh notation layer
- a log file for the process of Fast Pixelwise Analysis of Music Document (Optional)

3.2.1.3 Training model for Patchwise Analysis of Music Document

For the extraction of each layer automatically by Fast Pixelwise Analysis of Music Document job, machine learning models need to be trained. This can be achieved by a job, called the Training model for Patchwise Analysis of Music Document. The machine learning models assigned as inputs for the Fast Pixelwise Analysis of Music Document job can be trained using this job.

This job takes four types of inputs:

- An original document image
- A Noh notation layer image
- A background layer image
- Selected regions

This fourth input, selected region, can be specified in the Pixel job to identify a manually annotated area in a document image. In my case, all notations on each document page were annotated, therefore both the first and fourth inputs were assigned a whole document image. The job will output the following after training layer separation models:

- A layer separation model for the Noh notation layer
- A layer separation model for the background layer
- a log file

This Rodan job is a wrapper that is running the underlying Calvo classifier. As I needed changes in the Calvo classifier that were part of a new branch that has not yet been merged to develop,[6] I used a Python script to directly run the new branch to train my model.

3.2.2 Symbol Classification

After the creation of each manuscript layer using the Fast Pixelwise Classifier and the machine learning models created with the Training model for Patchwise Analysis of Music Document, the next step is to identify *fushi* notation. There are a few steps to accomplish this. First, we check the details of whether or not there are any *fushi* notations in the Noh notation layer using a job called Interactive Classifier. Interactive Classifier is a web-based interactive graphical interface of the Gamera Classifier and can be used to train a model for symbol classifications (Droettboom,

[6] https://github.com/DDMAL/Calvo_classifier/tree/sample_generator

MacMillan, and Fujinaga 2003). For this classification, four classes were defined, which are *fushi*, other (non-fushi) Noh notation, text, and skip. *Fushi* notation, which is the key component for detecting Noh chants, was classified into the designated *fushi* class. Other notations appearing only in Noh chants other than *fushi* notation were classified into the other Noh notation class. Noh chant or other Japanese books contained various text, including *furigana*, and these were classified into the text class. Any noise pixels, which occur due to the misclassification of pixels in the layer separation, were classified into the skip class. There were image pre-processing steps before Interactive Classifier, which were achieved by using the following Rodan jobs; Convert to one-bit, Despeckle, and CC Analysis. For every page, a few connected components were first annotated with a label, and other were automatically classified based on the labeled data. Then, correct results as well as some corrected misclassifications were added to the training set until all connected components are classified.

3.2.2.1 Preprocessing steps

In prior to the Interactive Classifier, there are preprocessing steps, using the following jobs: Convert to one-bit, Despeckle, and CC Analysis, as shown in Figure 3-9.

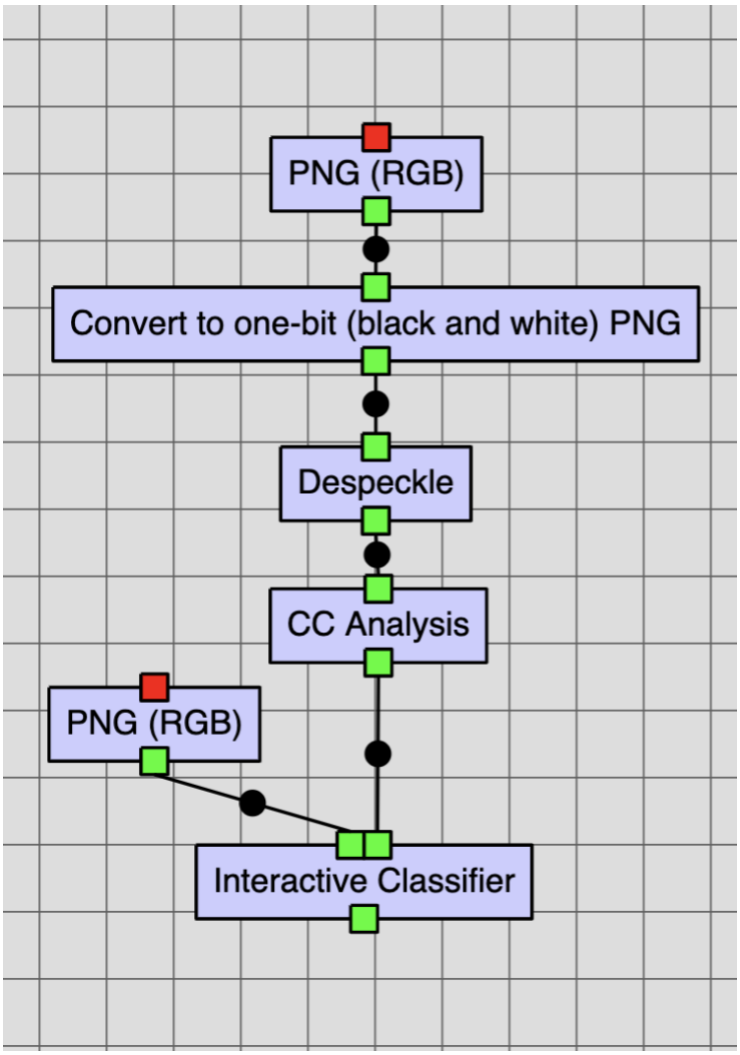


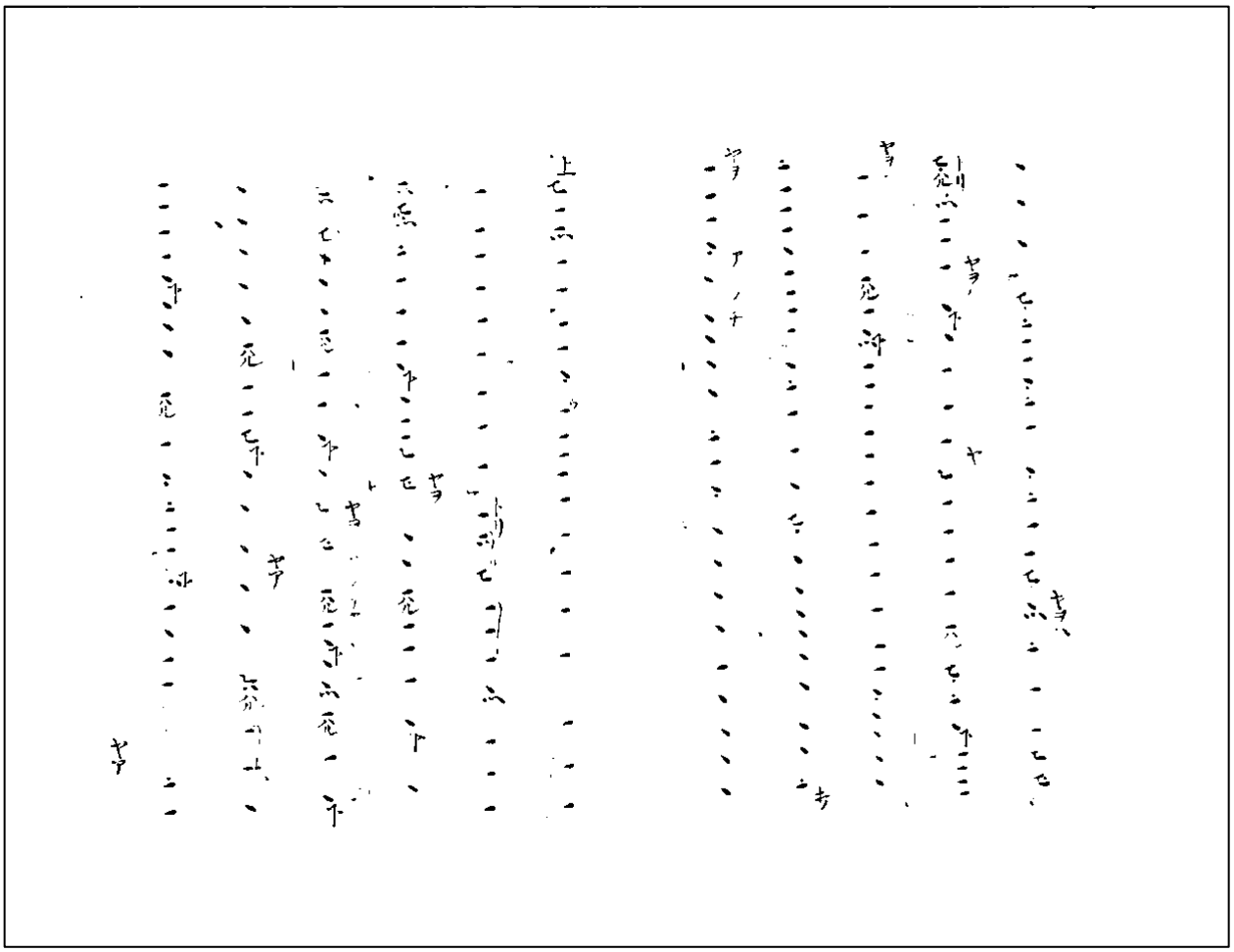
Figure 3-9 A Rodan workflow for the symbol classification with Interactive Classifier

The Convert to one-bit job converts an input image into grey-scale color as shown in Figure 3-10 (c), which takes a single image for input and output in PNG format. The Despeckle job removes any connected components smaller than a size specified in the setting, which was specified to 1 pixel in this research. The result from this job is shown in Figure 3-10 (d). The input and output ports take a single image in PNG format. The CC Analysis job performs connected component analysis on an input image, which considers more than two pixels together as a component. This job takes

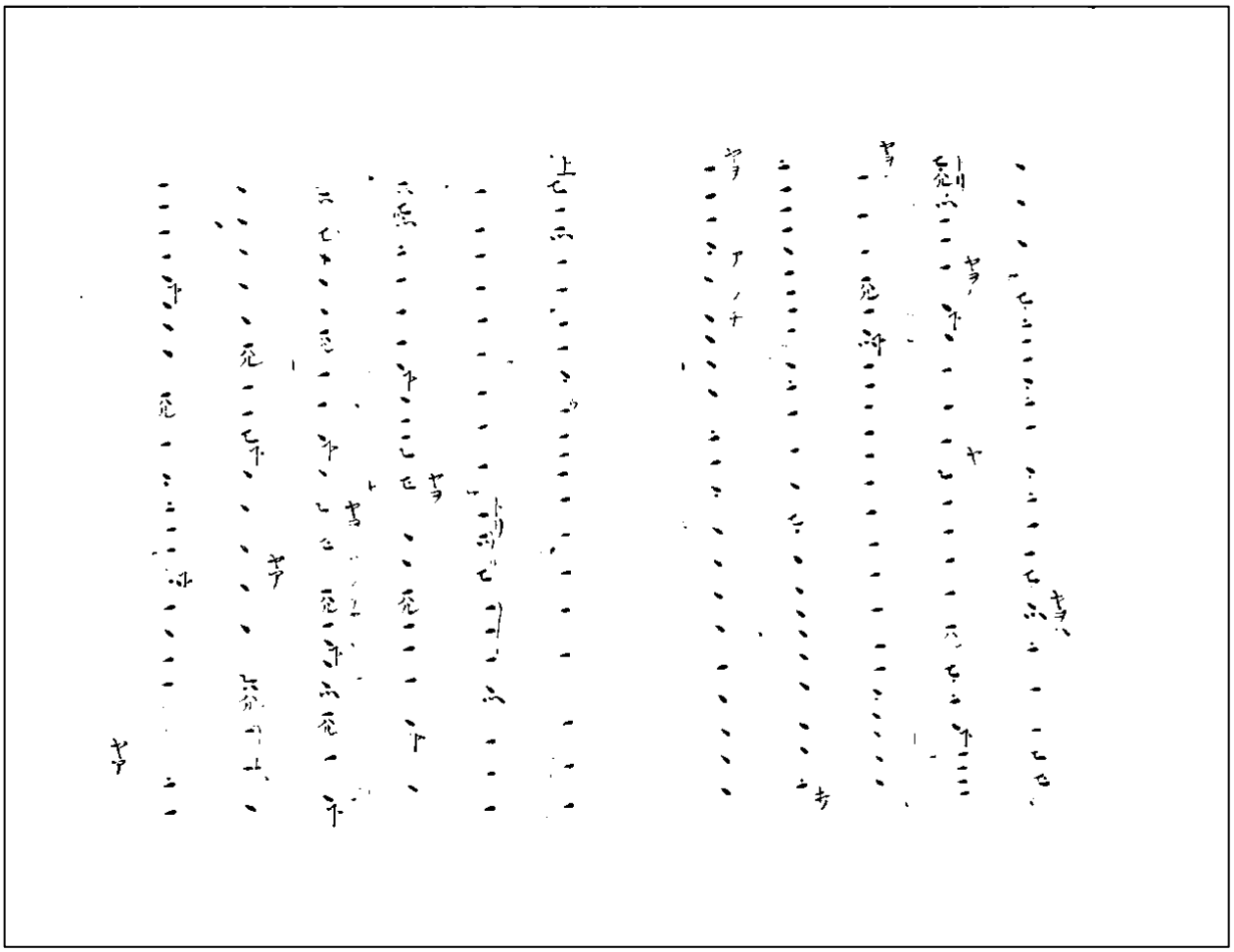
a single image in PNG format as an input and output connected components data in XML format as shown in Figure 3-10 (e).

(a)

(b)



(c)



(d)

(e)

Figure 3-10 Outputs from preprocessing jobs for the symbol classification (Hosho 1936). (a) is an original image, (b) is a result from the layer separation, (c) is a result after processing Convert to one-bit, (d) is a result after processing Despeckle, and lastly (e) is a result after the CC Analysis.

3.2.2.2 Interactive Classifier

The Interactive Classifier recognize each symbol in the Fushi notation layer as a distinct element. There are two stages involved in this process: the automatic classification stage and the manual classification stage. The former stage allows a user to manually classify glyphs to accumulate training data, while the latter approach allows an automatic classification based on the training data. In order to

achieve the classification, we need to define the classes for each musical symbol, which can be constructed in a hierarchical order.

For this classification, four classes were defined as mentioned in Section 3.2.2, which are *fushi*, other (non-*fushi*) Noh notation, text, and skip. All *fushi* notations were classified into the *fushi* class. Because there are numerous musical expressions other than *fushi* notation in Noh chant, other class was also defined, which is a category for storing musical expressions other than *fushi* notation. There may be some noise pixels on the separated *fushi* notation layer due to misclassification of a pixel, which may result in being identified as one connected component. This type of pixels needs to be ignored during the symbol classification since it does not belong to Fushi class or other musical symbol class. Rather than manually removing these connected components, these can be automatically selected to be filtered out during the automatic classification step by using the skip category. Otherwise, all connected components are assigned to one of three classes: *fushi*, Noh notation, or text, regardless of their confidence level for the prediction. The confidence level is certainty of prediction result from an automatic classification.

The Interactive Classifier requires connected components as input, which means that a preprocessing step to analyze each connected component from the extracted Fushi notation layer is required in advance to the use of Interactive Classifier. In the Rodan system, a job, called CC Analysis, can take on the role of the preprocessing step. Therefore, the workflow shown in Figure 3-9 was constructed for the purpose of the *fushi* notation identification.

The following are the inputs used for Interactive Classifier:

- An original document image
- A file containing all connected components
- Training data (Optional)

There are three outputs for Interactive Classifier:

- Training data, which includes the classification results from both current iteration and previous runs (Optional)
- Class Names, which is names of classes and subclasses used during the classification (Optional)
- Classified Glyphs, which is the classification result from the current iteration and is the only required output

3.2.2.3 NonInteractive Classifier

The results from Symbol Classification are exported as a file in XML format, which contains various information regarding every component, as shown in Figure 3-11. All information related to components are surrounded with a tag <glyphs></glyphs>, and inside the tag, a tag <glyph></glyph> is used to enclose each component. Each contains various information, including location, predicted class with confidence value, whether the classification was automatic or manual, and various features. Noninteractive Classifier was used to create the data for the binary classification, which employs the same classification engines as Interactive Classifier but only includes an automatic classification stage without a manual correction stage.

Noninteractive Classifier was chosen over Interactive Classifier to acquire confidence values for each classification, where the confidence values were used to eliminate predictions with low confidence. Interactive Classifier retrieve such values since those are always set to 1.0 after the manual correction stage. Therefore, all data for this Binary Classification was prepared using Noninteractive Classifier.

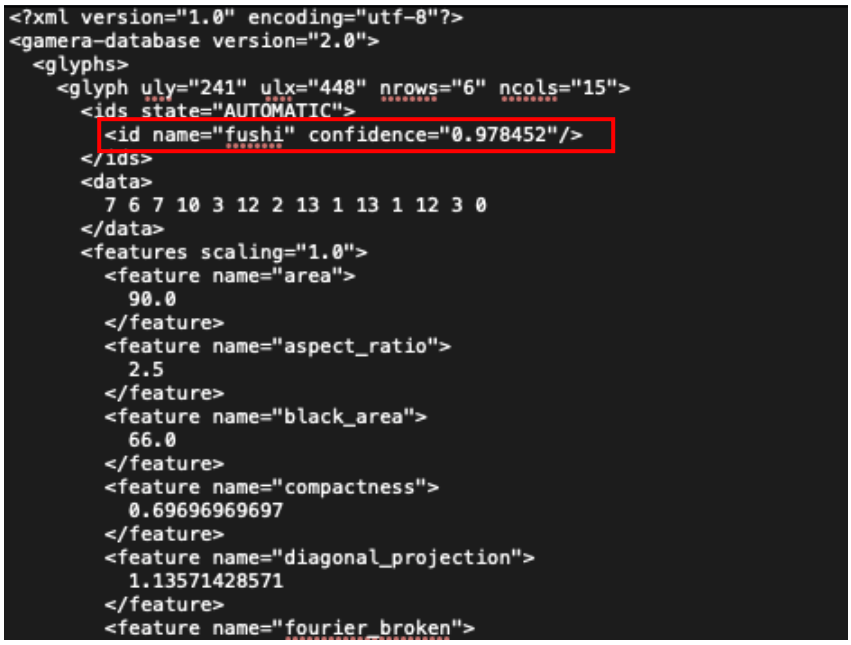


Figure 3-11 The result of the Noninteractive Classifier in xml file format where each symbol is surrounded by <glyph> tag. The red rectangle shows the data used for the following binary classification.

3.2.3 Binary Classification

After symbol classification, the next step is to classify Noh chant books from the dataset containing both Noh chant books and other Japanese books (non-Noh-chant) using the symbol classification results. Those results were parsed using a library

called pandas,[7] which is Python library for data analysis and manipulation (McKinney 2010). The class name and its confidence values were extracted for every glyph, shown in Figure 3-11 with a red rectangle, and the sum of each class as well as the ratio of each class with respect to the total number of glyphs as computed per XML file.

I used a decision tree classifier for the binary classification with a library called scikit-learn, which is machine learning library in Python (Pedregosa et al. 2011). The function in the scikit-learn library has various parameters, including criterion, splitter, max_depth, min_sample_split, min_sample_leaf, min_weight_fraction_leaf, and max_features ("Sklearn.Tree.DecisionTreeClassifier" n.d.). The first parameter criterion is related to calculating information gain, which is a measure of how well the child nodes are able to classify the data compared to the parent node. In the scikit-learn library, either "gini" or "entropy" can be chosen, both equations are shown below (Hastie, Tibshirani, and Friedman 2001).

$$Gini\ Index = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

$$Entropy = -\sum_{k=1}^{K} p_{mk} \cdot log p_{mk}$$

The splitter is related to how the features and the threshold value used for each node are selected, with two possible values: "best" or "random". The "best" selects the most important feature, while the "random" takes the feature randomly. The max_depth

[7] https://pandas.pydata.org/

is the maximum depth of a tree, and any integer values or "None" can be selected. If "None" is selected, a tree will be extended until all leaf nodes become pure and contain only data with the same class, or all leaf nodes contain data less than a value specified in min_samples_split. The min_sample_split is the minimum number of data points for a node to perform split, and any integer or float values can be selected. The min_weight_fraction_leaf is the minimum weighted fraction of the total weights for a leaf node, expressed as a float. Lastly, the max_features is the maximum number of features to consider when looking for the optimal split, which takes any integer values, float values, "auto", "sqrt", "log2", or None.

$$auto\ or\ sqrt: \quad \max \text{feature} = \sqrt{number\ of\ features}$$

$$log2: \quad \max \text{feature} = \log_2 number\ of\ features$$

$$None: \quad \max \text{feature} = \text{number of features}$$

GridSearchCV is a function in scikit-learn software library, allowing one to find the best performing parameters from selected ranges of parameters. This is achieved by trying out all possible combinations of those parameters in the ranges; this strategy was applied to the decision tree classifier using the parameters described above. A nested cross validation was constructed, where the inner loop performs parameter searches with two-fold cross-validated grid search and the outer loop is five-fold cross validation.

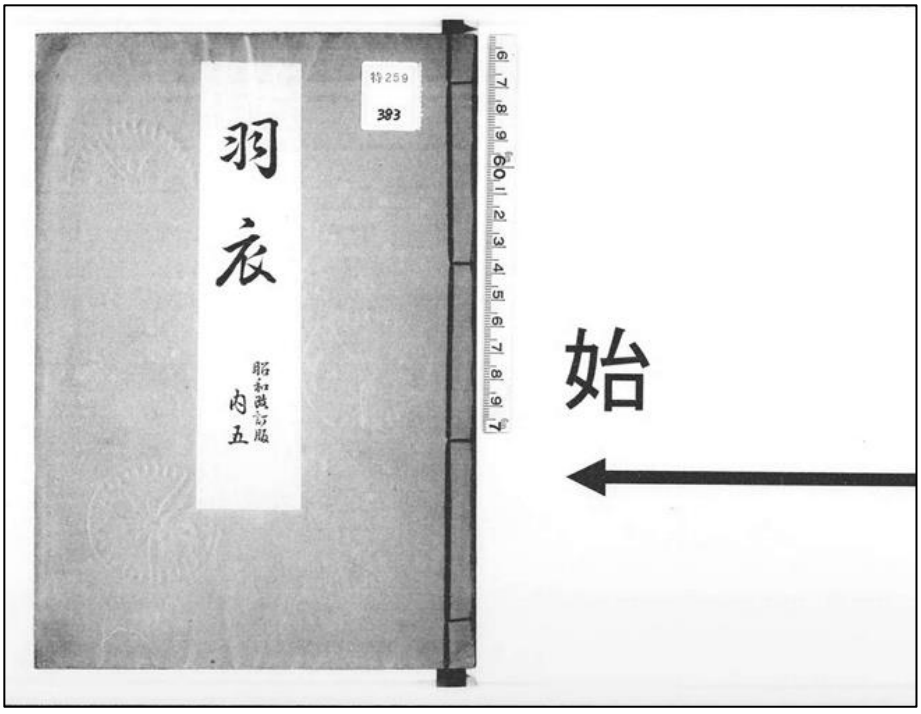
Chapter 4 Experiment

In this chapter, based on the methodology discussed in Chapter 3, the performance and results of each of the processes executed will be evaluated. There are three machine learning models that were applied: layer separation, symbol classification, and binary classification. Layer separation is a process of pixel classification of a given image; this model was used to produce two layers from an original image, the Noh notation layer and the background. In other words, a single document image is split into two layers: one containing all the Noh notation and the other containing everything else. The *fushi* notations were successfully and accurately extracted with this process, although the layer separation model appeared to have more difficulty classifying other notation, such as text or other Noh notation. After layer separation, the next step is symbol classification, which classifies each connected component within a Noh notation layer. The execution result was examined using the Interactive Classifier's console interface, and despite a few misclassifications, the majority of the components were accurately classified. The last process is binary classification, which determines whether a document is a Noh chant or another document class using the results from the symbol classification process; the accuracy of 0.950 was achieved using a decision tree classifier.

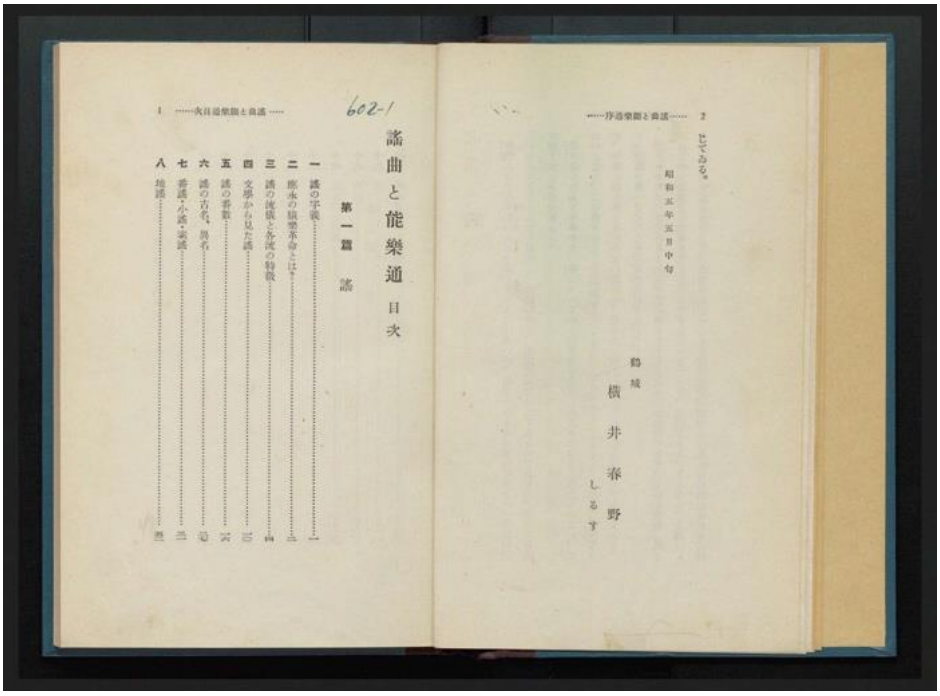
The methodology for data sampling is discussed in Section 4.1, the results from three machine learning models is discussed in Section 4.2, and lastly, discussions on data analysis and obstacles in the training are in Section 4.3.

4.1 Data

Data was collected from the National Diet Library, consisting of 16 Noh chant books and 15 other Japanese books. These were selected from two NDC (Nippon Decimal Classification) categories - NDC:768 and NDC:773. From the collected data, I sampled four pages from each book to be used for training machine learning models, and there were some pages that I intentionally avoided including in the training dataset. Often, a book contains a front and back cover within data, and occasionally blank pages or explanation pages as well, as shown in Figure 4-1. These irregular pages were different from the majority of the content within a book and thus were intentionally omitted for sampling data.



(a)



(b)

Figure 4-1 Irregular pages excluded for sampling. (a) is the front cover of a Noh chant book (Hosho 1936, 1). (b) is the table of contents for a book related to Noh (Yokoi 1930, 10).

The other type of data exclusion was only applied to Noh chant books. A Noh chant consists of two different modes, *utai* and *kotoba*, which means chanting and words respectively as discussed in Section 2.1 (Serper 2000). While *utai* contains various musical symbols including *fushi* notation alongside the text, *kotoba* has no such symbols. This study intends to classify documents based on those musical symbols, so I also intentionally excluded pages containing only such *kotoba* sections from the sampling data.

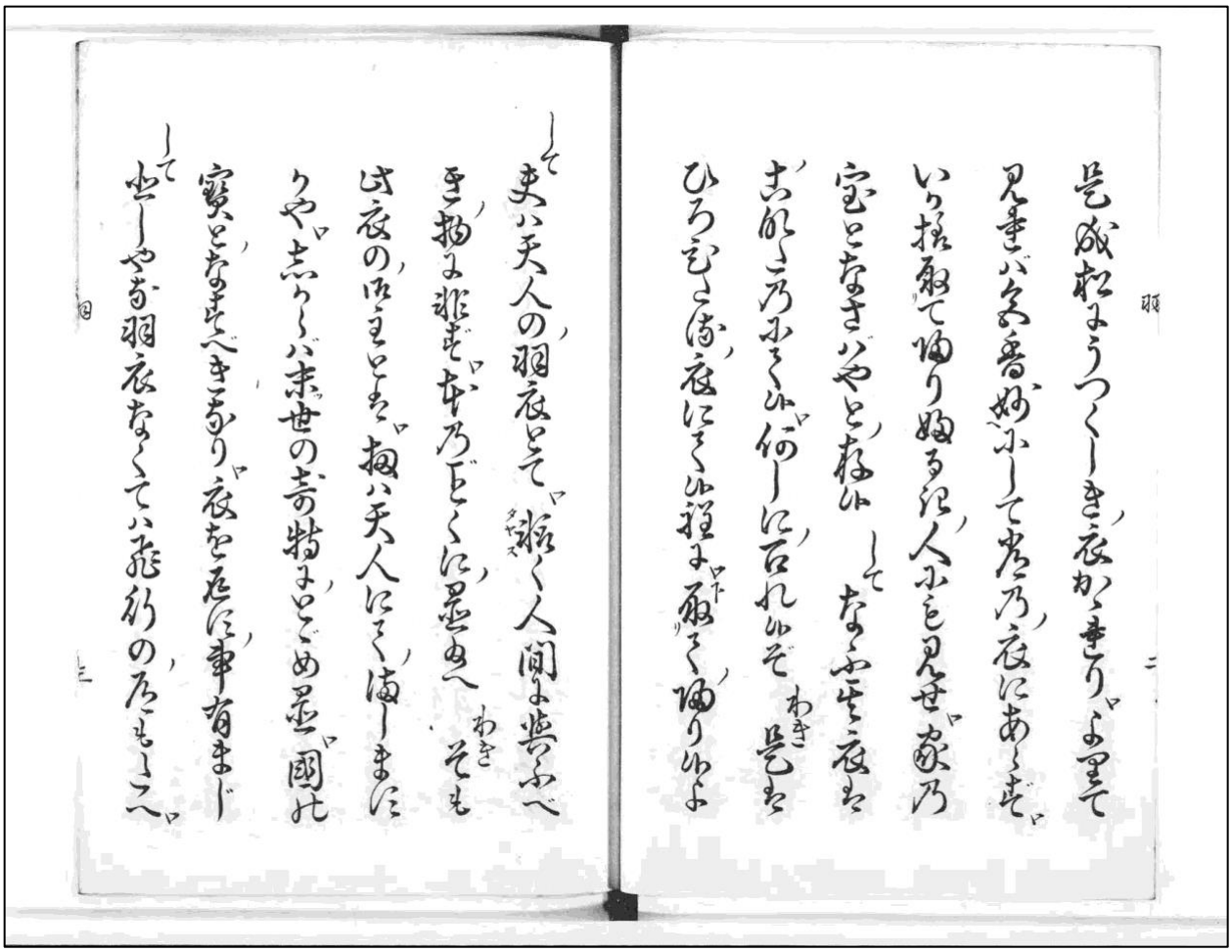


Figure 4-2 A page of *kotoba* section in Noh chant book without any *fushi* notation (Hosho 1936, 5).

A total of 124 pages were sampled, consisting of 64 pages of Noh chant books and 60 pages of non-Noh-chant books; I created training datasets corresponding to the three types of machine learning models applied in this research: layer separation, symbol classification, and binary classification. The 31 pages, with 16 pages of Noh chant books and 15 pages of non-Noh-chant books, were sampled for the layer separation, 31 pages, 16 pages of Noh chant books and 15 pages of non-Noh-chant books, were sampled for the symbol classification, and lastly 60 pages consisting of

30 pages of Noh chant books and 30 pages of non-Noh-chant books were sampled for the binary classification.

The symbol classification performs component-wise classification, therefore every component in each page of training data has annotation of its class, such as *fushi* or text. Often, the shape of fushi notations were observed to be similar among different Noh chant books as shown in Figure 4-3. Therefore, due to the commonality of *fushi* notation and other components for each class, only 12 pages from the initially collected dataset were used to train the symbol classification model, resulting in a total of 6,088 components of training samples.

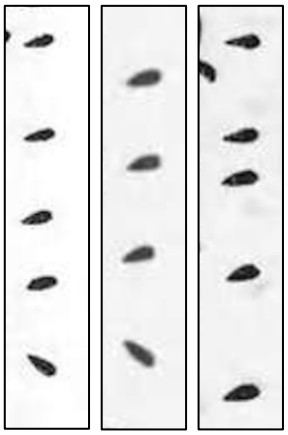


Figure 4-3 Examples of fushi notation in similar shape from different Noh chant books. (a) (Hosho 1936), (b) (Hosho 1937), (c) (Ookita 1914).

4.2 Results

4.2.1 Layer Separation

Although there are some differences in shapes of *fushi* notation in Noh chants, it has become possible to extract various shapes successfully as shown in Figure 4-4.

Figure 4-4 (a) has different *fushi* shapes from Figure 4-4 (d), for example, yet each *fushi* notation was extracted using the same model. For an issue with *fushi* notation extraction, sometimes the *fushi* existing on the first line of the page was not extracted as shown in Figure 4-5, even though most of the training examples contained *fushi* notation from the first line of a document page. One of the training examples for Noh chant is shown in Figure 4-6. This chant contains a variety of *fushi* shapes on a single document page. Some *fushi* symbols are long and similar to the common type often seen in other Noh chant books, whereas others are small and almost resemble noise-like components due to faded pixels. Although some noise was captured, most of the faded *fushi* notation was also able to be successfully extracted.

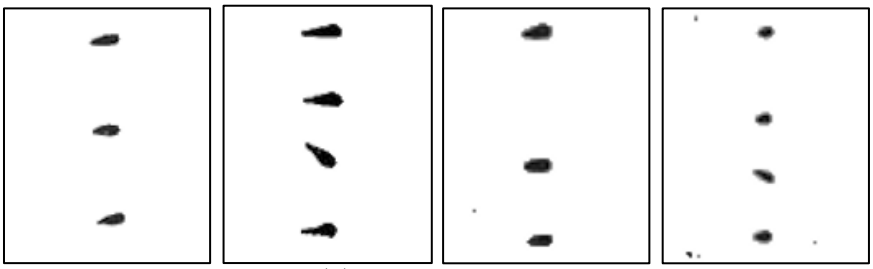


Figure 4-4 The successful extractions of *fushi* notation from various books. (a) (Hosho 1936, 10), (b) (Kanze 1881, 6), (c) (Kongo 1931, 21), (d) (Kongo 1932, 33)

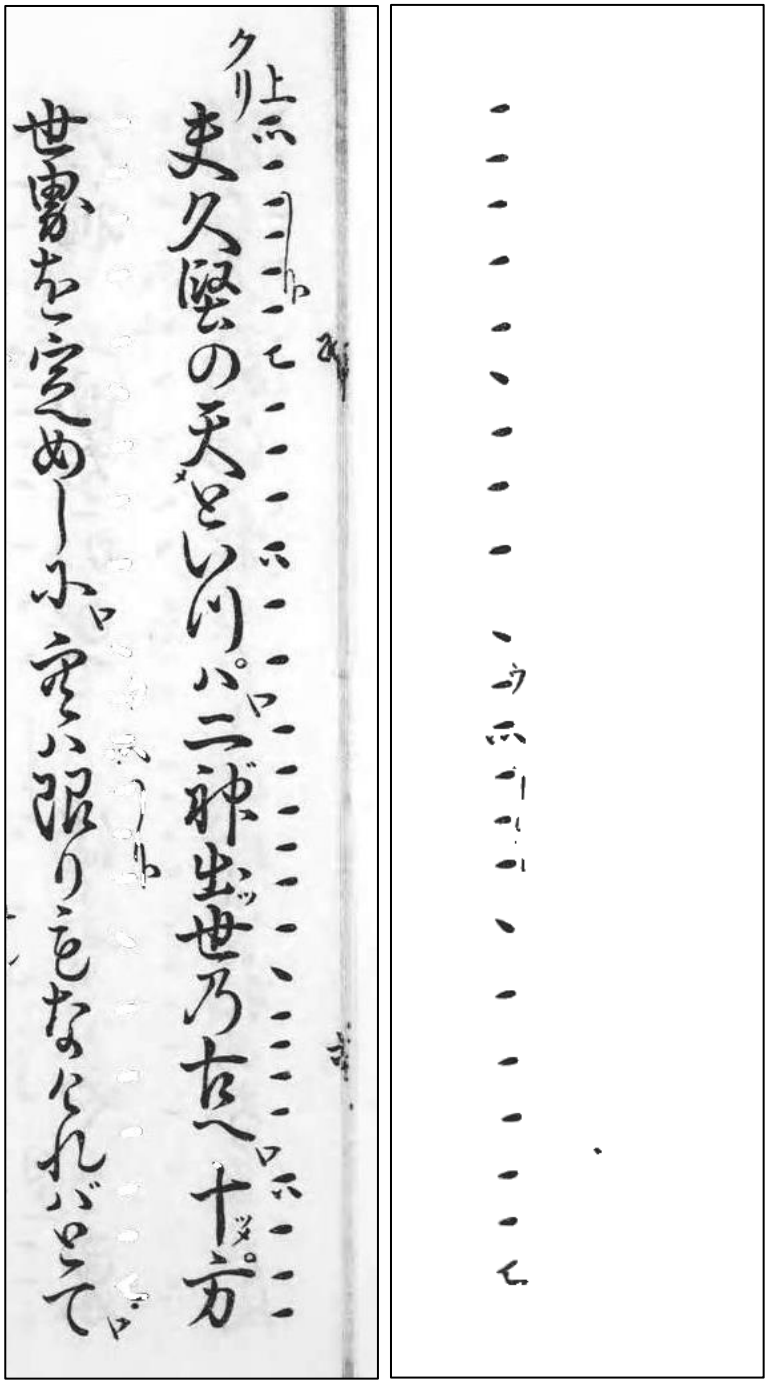


Figure 4-5 An issue detecting *fushi* notation on the first line. The left side shows the background layer, and the right side shows the Noh notation layer (Hosho 1936, 9).

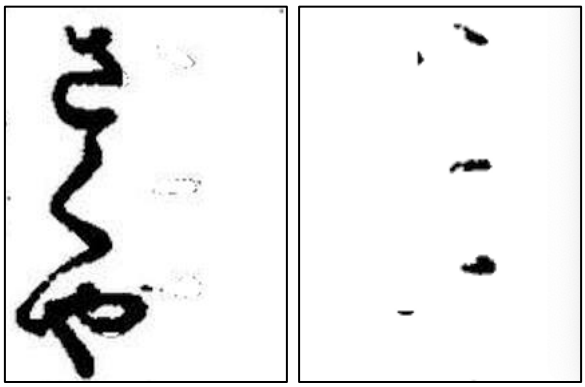


Figure 4-6 An example of Noh chant with faded fushi notation. The left side shows an original image, and the right side shows a Noh notation layer (Kon 1886, 6).

Other than *fushi*, there were further variances in Noh notation depending on Noh chants due to differences in description and writing style. Therefore, the extraction resulted in slightly worse outcome in comparison to *fushi* notation. In Noh, for example, the protagonist is referred to as *shite*, whereas the antagonist is referred to as *waki*. Noh chant also has descriptions such as *shite* and *waki* to specify a role for singing in a book, often appearing at the beginning of sentence. Some books use *hiragana* while others use *katakana*, both of which are different types of Japanese phonetic systems, as shown in Figure 4-7. As a result of such a disparity, it appears that extracting Noh symbols other than *fushi* is more challenging. Still, there were many cases where symbols were successfully extracted as shown in Figure 4-8, however, there were also other cases that involved partial or no extractions as shown in Figure 4-9.

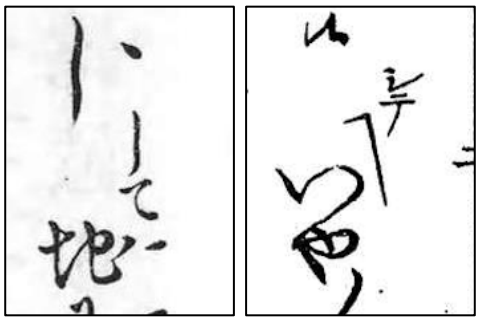


Figure 4-7 The difference in writing style of Noh chant. Both means *shite*. The left side is written in *hiragana* (Hosho 1936, 6), while the right side is written in *katakana* (Kanze 1881, 4).

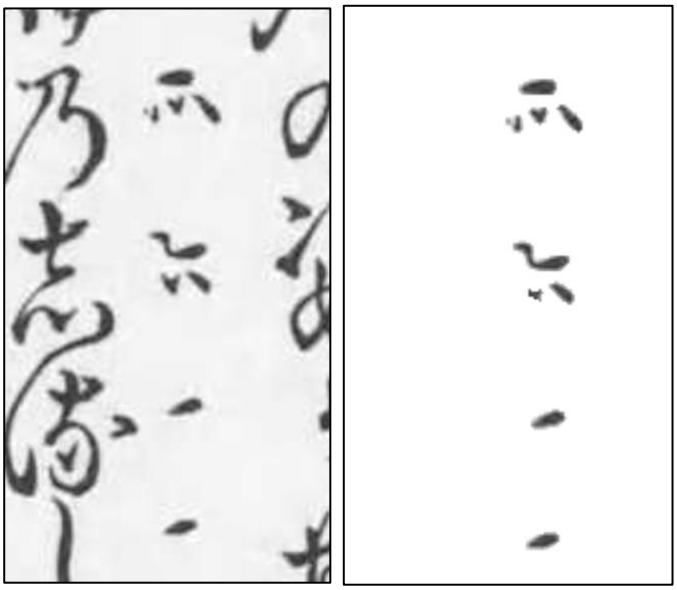


Figure 4-8 An example of Noh notation extraction. The left is an original image, and the right is Noh notation layer (Hosho 1937, 10).

Figure 4-9 An example of Noh notation extracted partially. The left is a background layer, and the right is Noh notation layer (Kanze 1921, 10).

Furthermore, among the main text, which should be classified as the background in Noh chant, those with components similar to *fushi* notation were occasionally retrieved into a Noh notation layer. In Figure 4-10, the *dakuten* is recognized as Noh notation. The *dakuten* is a pair of double dot-like symbols added to *hiragana* to shift consonants, such as *fu* (ふ) to *bu* (ぶ). Similarly, in Figure 4-11, a text containing a component similar to *fushi* notation was extracted in some cases.

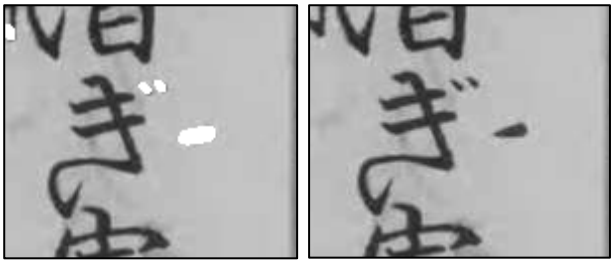
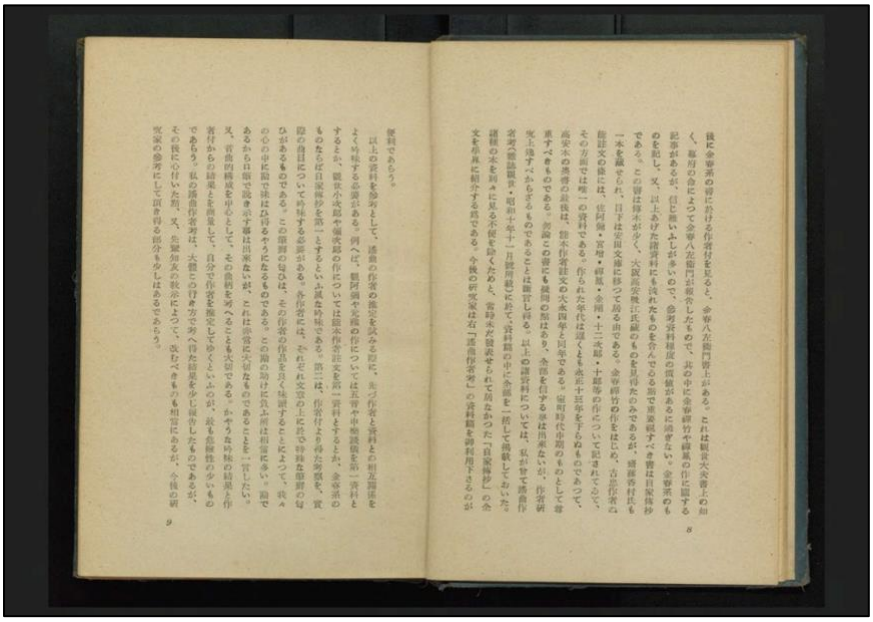


Figure 4-10 An example of incorrect extraction of dakuten. The left is an original image, and the right is Noh notation layer (Kanze 1921, 21).

Figure 4-11 An example of incorrect extraction of partial text (Kanze 1881, 4). The left is an original image, and the right is Noh notation layer.

The extraction results varied depending on the book in the case of non-Noh-chant books. Some, such as Figure 4-12, do not contain any *fushi*-like symbols or *furigana*, and all were successfully classified as background.



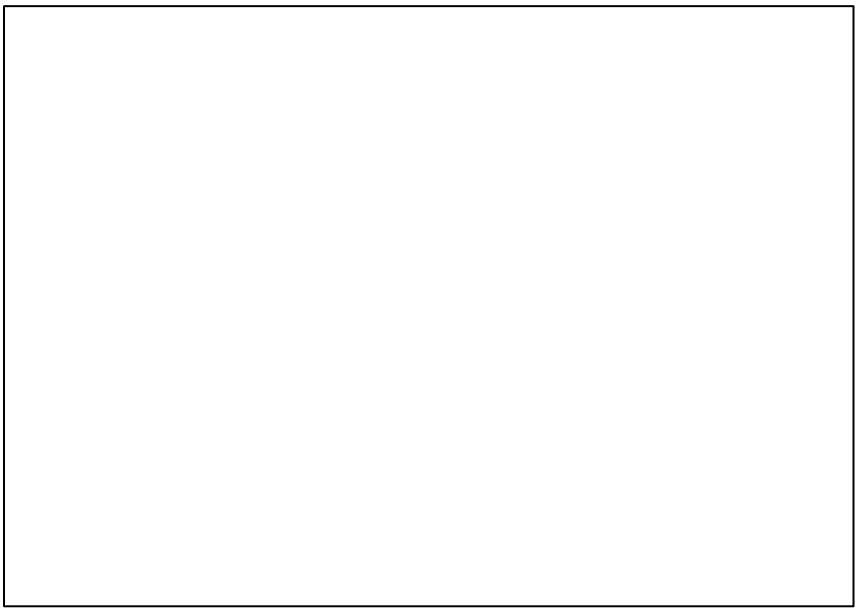


Figure 4-12 An example of a non-Noh-chant book without any pixels on Noh notation layer (Nose 1940, 11). The top is the background layer, and the bottom is the Noh notation layer, which should be completely blank as shown.

For another case in Figure 4-13, there are dot-like symbols along with text, showing emphasis on characters, similar to underlining in English text. These were also extracted due to its similar structure with *fushi* notation.

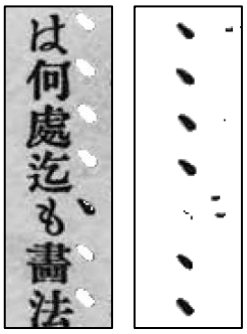


Figure 4-13 An example of extraction of emphasis dots from non-Noh-chant books (Sakamoto 1914, 17). The left is the background layer, and the right is the Noh notation layer.

In some circumstances, however, non-Noh-chant books appeared to have more noise than Noh chant. Figure 4-14, for example, was found to be relatively noisy. The training data used for this specific book is shown in Figure 4-15, which exclusively extracts *furigana*. However, in addition to this *furigana*, the top portion of the image is also extracted as a result. It is possible that the text at the top region was extracted since it is written in a smaller font than the main text of the page and is comparable in size to *furigana*.

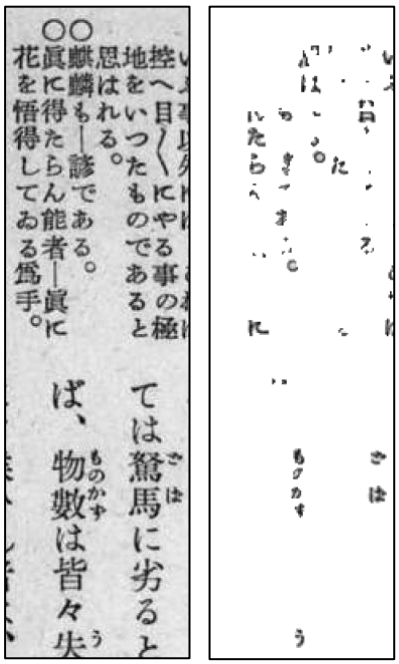


Figure 4-14 An example of noisy data for non-Noh-chant books (Zeami and Nose 1947, 17). The left is an original image, and the right is the Noh notation layer.

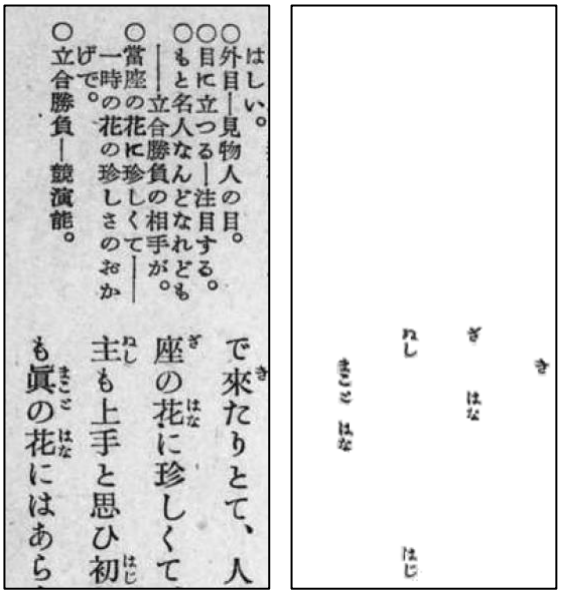
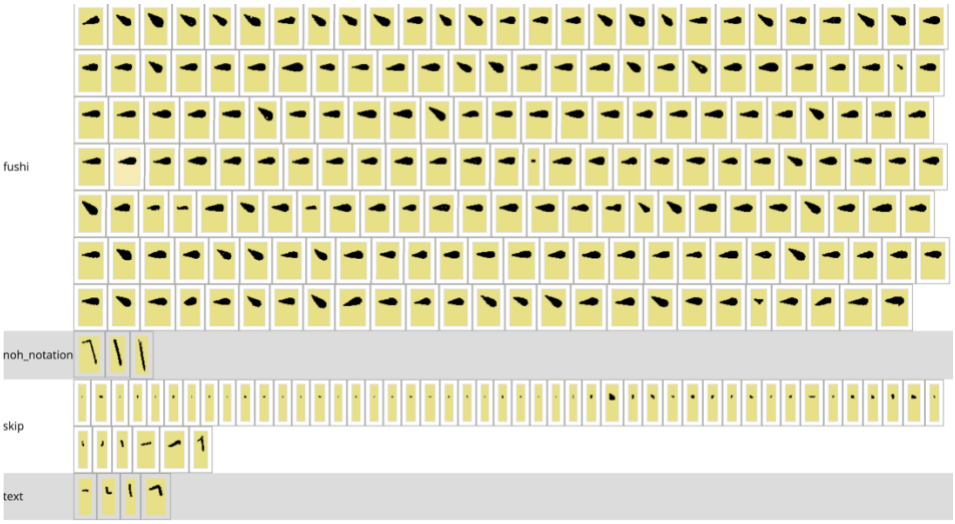


Figure 4-15 The training sample from the same book as Figure 4-19 (Zeami and Nose 1947, 15). The left is an original image, and the right is the Noh notation layer.

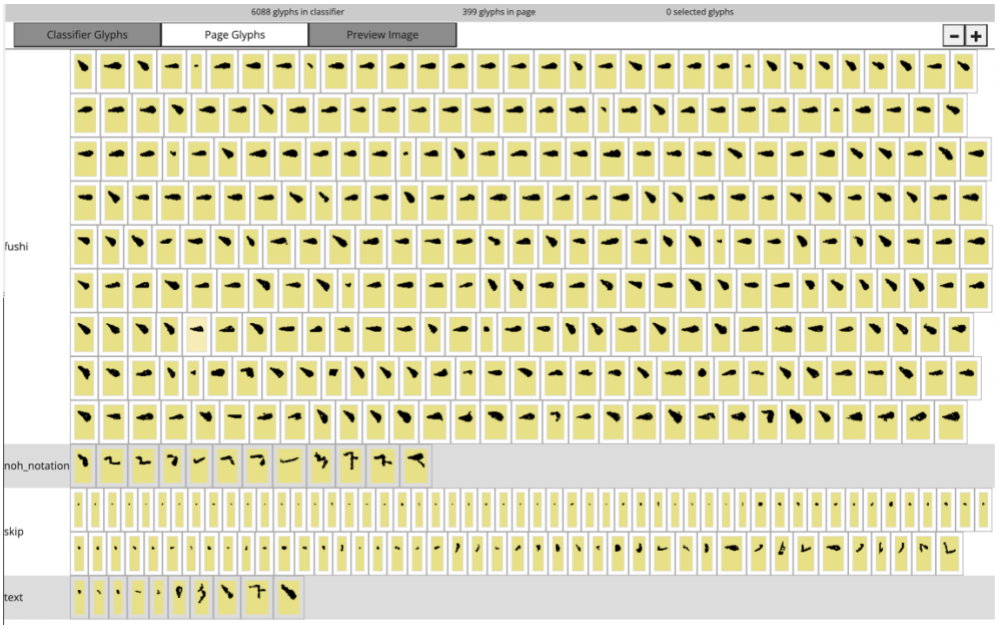
Although many components were successfully recognized, especially in the case of *fushi* notation, other components, such as text and other Noh notation, appeared to be more challenging to identify.

4.2.2 Symbol Classification

Here are the results of extracting each Connected Component from the Layer Separation results and classifying those symbols. In Layer Separation, noises appeared depending on the document, but those were categorized into one class "skip" in the symbol classification. The outcomes of the execution are provided below. These are the results of running Interactive Classifier on the test data and images were captured on its console screen.



(a)

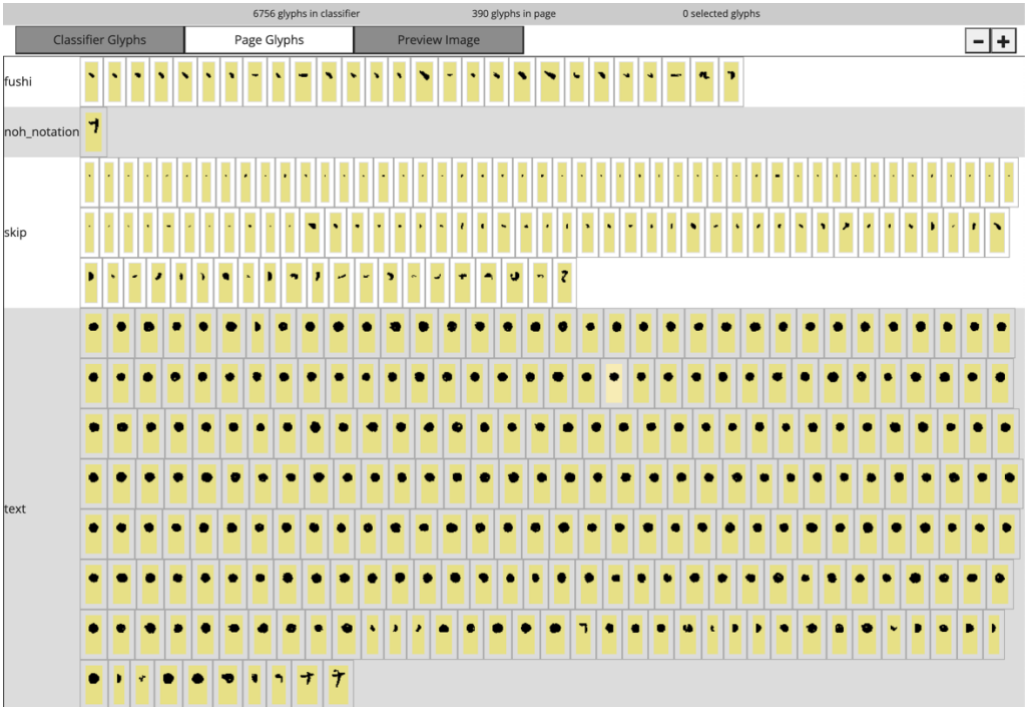


(b)

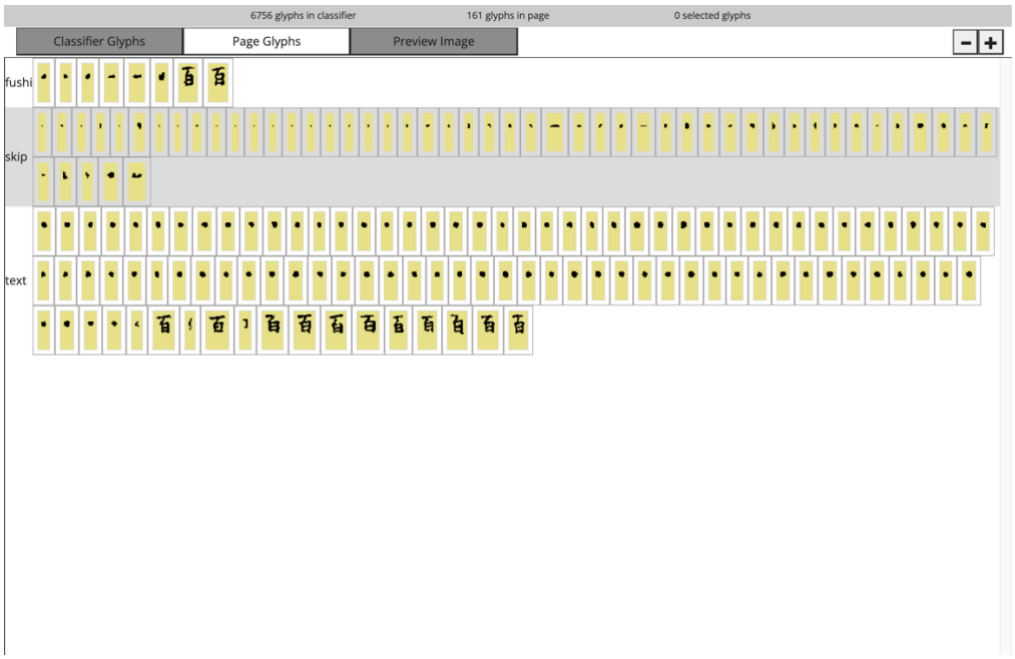
(c)

Figure 4-16 The classification results of the symbol classification for Noh chant books using Interactive Classifier. (a) (Kanze and Kanze 1894, 7), (b) (Terada 1885, 9), and (c) (Kongo 1932, 202)

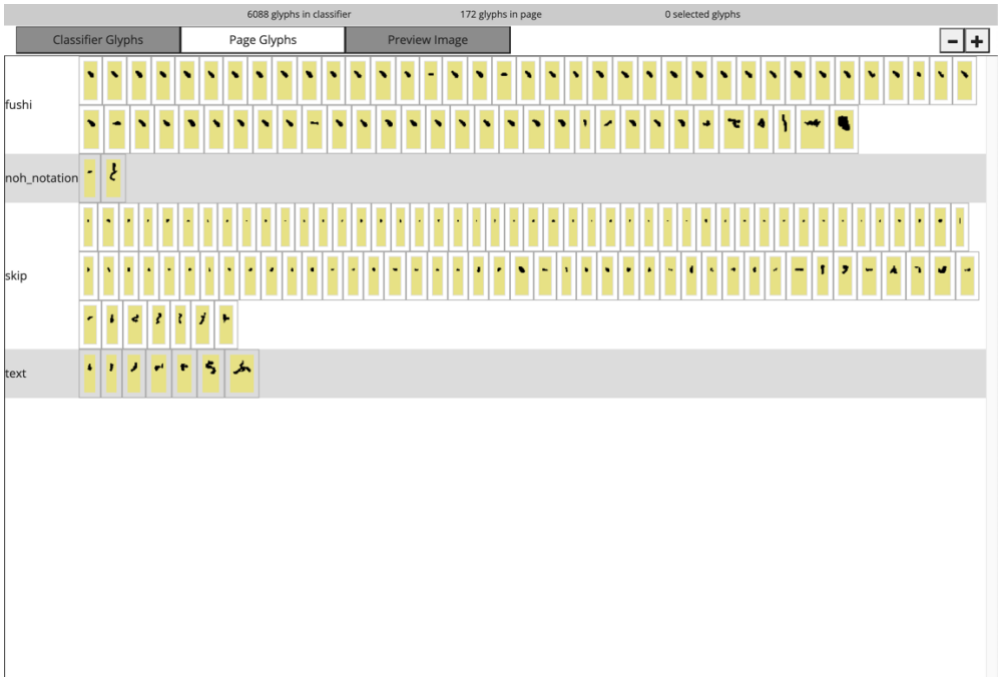
Figure 4-16 (a) shows that all of the *fushi* notations was correctly classified into the *fushi* class. However, there are some misclassifications also appearing in the *fushi* class. Most of the *fushi* notation was classified into the *fushi* class in Figure 4-16 (b) as well; however some misclassifications were observed. Non-*fushi* components are included in the *fushi* class, owing to the fact that other document images have relatively small and similar forms to these misclassified components. Figure 4-16 (c) contains smaller *fushi* notation than previous examples. In this document, there are symbols other than *fushi*, whose shape is similar to "*ノ*" , in the *fushi* class.



(a)



(b)



(c)

Figure 4-17 The classification results of the symbol classification for non-Noh-chant books using Interactive Classifier. (a) (Tokuhiro 1899, 13), (b) (Tougi 1894, 19), (c) (Sakamoto 1914, 18).

Figure 4-17 (a) is an example from books other than Noh chant, but the document image contains *fushi*-like component. These appear to have been detected as *fushi* notation. Likewise, some components have been classified as *fushi* in Figure 4-17 (b), but the majority have been classified successfully. Figure 4-17 (c) also contains *fushi*-like components, which were misclassified into *fushi* class.

Therefore, although few misclassifications were observed in each class, most components were generally classified successfully, and many more components were classified into the *fushi* class particularly for Noh chant.

4.2.3 Binary Classification

4.2.3.1 Data Processing

For data processing, predicted classes and their confidence values were extracted from each XML file, and any prediction with a confidence value of less than 90% was discarded since most of the correct classifications achieved higher confidence values. The number of components for each class as well as the ratio of those to the sum of all components were calculated for each document image using an xml file.

Table 4-1 and Table 4-2 shows the statistics of the results from symbol classification for Noh chant data and non-Noh-chant books. Noh chant books have a significantly higher average number of *fushi* notation and *fushi* ratio than other books. Non-Noh-chant books, on the other hand, have a substantially higher average number of text and text ratio than Noh chant books.

Table 4-1 Data statistics of the results from symbol classification for Noh chant books

	fushi	noh notation	text	skip	fushi ratio	noh notation ratio	text ratio	skip ratio
mean	185.813	4.594	22.531	127.125	0.560	0.012	0.060	0.368
std	62.194	4.302	19.319	68.047	0.137	0.008	0.033	0.118
min	43.000	0.000	3.000	50.000	0.264	0.000	0.011	0.200
max	335.000	20.000	72.000	333.000	0.772	0.033	0.149	0.693

Table 4-2 Data statistics of the results from symbol classification for non-Noh-chant books

	fushi	noh notation	text	skip	fushi ratio	noh notation ratio	text ratio	skip ratio
mean	16.833	1.467	151.567	146.933	0.086	0.006	0.246	0.628
std	16.423	2.389	358.292	160.925	0.096	0.012	0.226	0.238
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
max	67.000	10.000	1559.000	700.000	0.393	0.052	0.685	1.000

4.2.3.2 Classification

Classification was performed using a decision tree classifier with k*l-fold nested cross validation, as discussed in Section 3.2.3. The processed data was split into 5 folds with equal distributions of Noh chant and non-Noh-chant data, each fold containing 12 document pages consisting of 6 pages of Noh chant and 6 pages of non-Noh-chant. Each set was split into two folds, each fold containing 3 pages of Noh

chant and 3 pages of non-Noh-chant, to be cross validated with GridSearchCV to find the best performing hyperparameters, as discussed in Section 3.2.3. The following are the parameters for decision tree classifier and evaluated with five folds cross validation; criterion, splitter, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, and max_features. The hyperparameters include “gini” and “entropy” for criterion, “best” and “random” for splitter, between 2 and 5 in integer increments for max_depth, between 2 and 10 in integer increments for min_samples_split, between 1 and 5 in integer increments for min_samples_leaf, between 0 and 0.5 in 0.1 increments for min_weight_fraction_leaf, “auto”, “sqrt”, “log2”, and None for max_features. Table 4-3 shows that the results from this classification, and the mean accuracy of 0.9500 was obtained.

Table 4-3 The results of decision tree classifier with nested cross validation

fold	test score
1	0.8333
2	0.9167
3	1.0000
4	1.0000
5	1.0000
mean	0.9500

4.3 Discussion

4.3.1 Data Analysis of the symbol classification

In symbol classification, there were a few outliers in the data observed in the results. For example, some Noh chant data contained relatively small number of *fushi* notation. On the other hand, some non-Noh-chant data were determined as containing some *fushi* notation even though *fushi* notation is not present in those documents. In fact, the minimum number of *fushi* notation in Noh chant in Table 4-1 is smaller than the maximum number of *fushi* notation in Table 4-2. Therefore, the results from the symbol classification were sorted in the following criteria, and the details of the top five data were manually inspected.

1. Noh chant with a small number of *fushi*
2. Noh chant with a low *fushi* ratio
3. Non-Noh chant books with many *fushi* notations
4. Non-Noh chant books with a high *fushi* ratio

Table 4-4 shows a list of Noh chant books with a small number of *fushi* notation. All the examples here were checked manually and found to have only few *fushi* notations in each image. For example, the first and the second data were contained 30 and 66 *fushi* notations respectively, which was counted by eye. Table 4-5 shows a list of Noh chant books with low *fushi* ratio, and those with a low *fushi* ratio were found to be either those that originally have a small number of *fushi* notation similar to the

previous criteria, or those that contain many other components, such as text, which resulted in a low *fushi* ratio.

Table 4-4 Noh chant data with low number of *fushi* notations

fushi	Noh notation	text	skip	fushi ratio	Noh notation ratio	text ratio	skip ratio
43.000	3.000	4.0	113.0	0.264	0.018	0.025	0.693
71.000	0.000	10.0	58.0	0.511	0.000	0.072	0.417
88.000	4.000	11.0	120.0	0.395	0.018	0.049	0.538
106.000	2.000	11.0	128.0	0.429	0.008	0.045	0.518
112.000	0.000	13.0	59.0	0.609	0.000	0.071	0.321

Table 4-5 Noh chant data with low *fushi* ratio

fushi	Noh notation	text	skip	fushi ratio	Noh notation ratio	text ratio	skip ratio
43.0	3.000	4.000	113.000	0.263804	0.018405	0.024540	0.693252
158.0	10.000	72.000	333.000	0.275742	0.017452	0.125654	0.581152
205.0	11.000	67.000	263.000	0.375458	0.020147	0.122711	0.481685
88.0	4.000	11.000	120.000	0.394619	0.017937	0.049327	0.538117
178.0	14.000	42.000	198.000	0.412037	0.032407	0.097222	0.458333

Table 4-6 Non-Noh-chant books with high number of *fushi* notations

fushi	Noh notation	text	skip	fushi ratio	Noh notation ratio	text ratio	skip ratio
67.000	1.000	13.000	140.000	0.303167	0.004525	0.058824	0.633484
64.000	0.00000	4.00000	95.000	0.392638	0.000000	0.024540	0.582822
33.000	1.000	1306.000	576.000	0.017223	0.000522	0.681628	0.300626
28.000	2.000	192.000	109.000	0.084592	0.006042	0.580060	0.329305
27.000	1.000	248.000	110.000	0.069948	0.002591	0.642487	0.284974

Table 4-7 Non-Noh-chant books with high *fushi* ratio

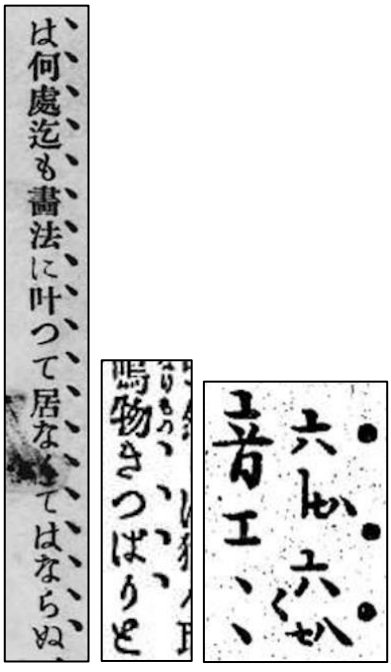
fushi	Noh notation	text	skip	fushi ratio	Noh notation ratio	text ratio	skip ratio
64.000	0.000	4.000	95.000	0.392638	0.000000	0.024540	0.582822
18.000	0.000	4.000	32.000	0.333333	0.000000	0.074074	0.592593
67.000	1.000	13.000	140.000	0.303167	0.004525	0.058824	0.633484
7.000	0.000	5.000	38.000	0.140000	0.000000	0.100000	0.760000
7.000	0.000	2.000	50.000	0.118644	0.000000	0.033898	0.847458

Table 4-6 shows a list of Non-Noh-chant books with a high number of *fushi* notation, and these were mainly due to the presence of *fushi*-like symbols in a document. For example, the first and second data in Table 4-6 contained dot-like symbols, shown in Figure 4-18 (a), which were incorrectly classified as *fushi* notation.

Similarly, the third data in Table 4-6 also contained symbols similar to *fushi* notation, shown in Figure 4-18 (b). For the fourth and fifth data in Table 4-6, symbols are shown in Figure 4-18 (c). While Noh chant contains many *fushi* notations along text, these misclassifications were infrequent and spread across an image if they existed.

Table 4-7 shows a list of Non-Noh-chant books with high *fushi* ratio, and these data was found to fall into two categories; one, *fushi*-like symbols are misclassified similar to previous criteria and affecting the ratio, or other, the number of other components is low and affecting the ratio as well.

Therefore, these anomalous data were found to be caused by the document's characteristics, including the number of *fushi* notation, and the number of other components, and the presence of a component similar to *fushi* notation.



(a) (b) (c)

Figure 4-18 Misclassified *fushi*-like symbols in Non-Noh-chant documents. (a) (Sakamoto 1914, 17), (b) (Toundo 1892, 7), (c) (Tokuhiro 1899, 13)

4.3.2 Challenges in the training process

There were few challenges and issues I encountered during the training process of the symbol classification. The training was performed page by page, and all previously annotated symbols were used as training data to classify new unlabeled components. There was a slight difficulty of this classification when the shape of *fushi* notation differed significantly from previous samples. For example, the *fushi* notation in Figure 4-19 (a) appeared to be a common type in the whole dataset, each of which contained a sharp edge on the left side and thicker right side. On the other side, the *fushi* notation in Figure 4-19 (b) is much smaller. When classifying Figure 4-19 (b) using only training data from Figure 4-19 (a), *fushi* notations in Figure 4-19 (b) were often mistaken as noise components. Therefore, even though *fushi* notation is present in the data, those could be misclassified if the training data did not contain similar shapes of *fushi* notation. This means that it is important to include various shapes of *fushi* notations in the training dataset, especially in the case where this classifier can be applied to unseen data.

For Figure 4-20, there were many texts written alongside *fushi* notations, and these included many *kanji* characters. Because many *kanji* characters consist of independent connected components, they had to be manually grouped during the training process. As an example, the letter "𠂢" was identified as two components, as shown in Figure 4-20 (a). Since the right component is relatively small, it was first categorized as noise and had to be reclassified. Although the emphasis for *kanji*

recognition has been decreased because this study focuses on *fushi* notation detection, this grouping would be a challenging if a *kanji* character needs to be precisely recognized precisely.

(a) (b)

Figure 4-19 The difference in shape of *fushi* notation from Noh chant document. (a) (Hosho 1936, 6), (b) (Kongo 1932, 212). A *fushi* notation in (a) is slightly larger than (b).

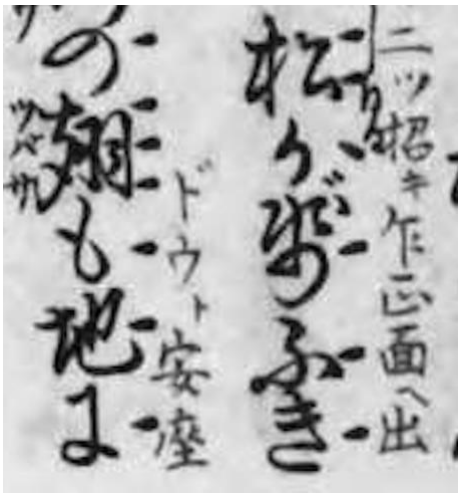
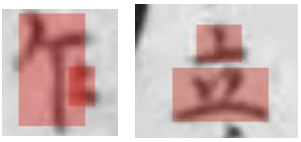


Figure 4-20 An example of training sample for Noh chant document with many texts written alongside *fushi* notations (Kongo 1931, 21)



(a) (b)

Figure 4-21 Examples of a single character detected as multiple components by automatic classification on Interactive Classifier, which needed to be manually grouped together (Kongo 1931, 21). (a) a character is split into three components. (b) a character is split into two components.

In this chapter, I have described dataset creation for three machine learning models, and their utilization in the execution of each model. First, the performance of the layer separation models as well as misclassifications were discussed. While several components, especially various *fushi* notation, were successfully identified, others, such as text and other Noh notations, seemed to be more challenging to identify. Secondly, the performance of the symbol classification was discussed as well as the importance of the large difference between Noh chant and other books in the number of classified *fushi* components. Although a few misclassifications were observed, most components were generally classified successfully. Lastly, data analysis of symbol classification as well as the result from a binary classifier were discussed, and the accuracy of 0.950 was achieved using a decision tree classifier.

Chapter 5 Conclusion

This thesis presented an original approach to the automatic classification of Noh chant books using special characteristics of document images. It started with the creation of datasets; a total of 31 books were collected from the National Diet Library. From the collected data, three datasets were created to apply to each of three types of machine learning models: layer separation, symbol classification, and binary classification. For layer separation, the ground-truth data of two layers, a Noh notation layer and a background layer, were created from a dataset of 36 pages and used to train models, and various *fushi* components were successfully recognized and extracted to the Noh notation layer. The results from the layer separation were then further processed, and all connected components in each document image were classified into four classes: *fushi* notation, other Noh notation, text, and noise. 12 pages were used for training a model containing 6,608 symbols that were annotated for training samples. Although a few misclassifications were observed, most components were generally classified successfully. Even though there were many varied shapes of fushi notation in the sampled data, they were successfully recognized and classified. The last processing step was the binary classification, which determined whether a document is a Noh chant or another type of document using the results from the symbol classification process; an accuracy of 0.950 was obtained with a decision tree classifier.

Generally, the main difference was found in the number of fushi notation between Noh chant books and other Japanese books. As expected, Noh chant books contained a much higher number of fushi notation per page. However, there were a few exceptions. Some Noh chant document pages contained only a small number of fushi notation, and non-Noh-chant documents sometimes contained fushi-like symbols. Nevertheless, overall process was successful.

5.1 Future Work

For this study, the training and validation samples were collected from the same set of books, although this would not be the case in a production setting. The possible future use case for this work is to find Noh chant books regardless of their current classification even outside of musical document classes. In this case, the model needs to classify documents that have not been seen before, which may need a larger set of training examples. The model will need to be executed on many other samples from different Noh chant and other Japanese books to identify patterns of misclassification, after which more data should be collected to fine-tune the model for better performance.

Furthermore, similar to automatic glyph recognition used for Western chant manuscripts, it would be interesting to perform Noh notation recognition. This study only classified fushi as a separate class; however, as discussed in Chapter 2, there are numerous additional Noh notations that can be classified as unique classes as well. This will allow other Noh notation symbols to be detected and classified into their

own categories. In addition, it is also possible to correlate the handwritten text with Noh notation by detecting it using, for example, the Kuronet, a machine learning model for recognizing Japanese handwritten text in a cursive writing style (Lamb, Clanuwat, and Kitamoto 2020).

5.2 Contributions

This research was motivated by document analysis and musical notation recognition for Western chant manuscripts. Although many studies have analyzed Western music documents, there has been a dearth of research regarding Japanese music documents. The contribution of this thesis may be considered as a significant first step in the analysis of Noh documents so that they can be automatically classified in the future.

Bibliography

Books, Articles, Websites, and Images

Appiani, Enrico, Francesca Cesarini, Anna Maria Colla, Michelangelo Diligenti, Marco Gori, Simone Marinai, and Giovanni Soda. 2001. "Automatic Document Classification and Indexing in High-Volume Applications." *International Journal on Document Analysis and Recognition* 4 (2): 69–83.

Artanisen. 2019. *Noh Stage*. Photograph. https://upload.wikimedia.org/wikipedia/commons/3/37/Noh-stage.jpg.

Bagdanov, Andrew D., and Marcel Worring. 2001. "Fine-Grained Document Genre Classification Using First Order Random Graphs." In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, 79–83.

Baldi, Stefano, Simone Marinai, and Giovanni Soda. 2003. "Using Tree-Grammars for Training Set Expansion in Page Classification." In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2:829. ICDAR '03. USA: IEEE Computer Society.

Brazil, Garrick, Xi Yin, and Xiaoming Liu. 2017. "Illuminating Pedestrians via Simultaneous Detection and Segmentation." In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 4960–69.

Burgoyne, John Ashley, Yue Ouyang, Tristan Himmelman, Johanna Devaney, Laurent Pugin, and Ichiro Fujinaga. 2009. "Lyric Extraction and Recognition on Digital Images of Early Music Sources." In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 723–28. Kobe, Japan.

Calvo-Zaragoza, Jorge, Francisco J. Castellanos, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. "Deep Neural Networks for Document Processing of Music Score Images." *Applied Sciences* 8 (5): 654.

Calvo-Zaragoza, Jorge, Gabriel Vigliensoni, and Ichiro Fujinaga. 2017. "Pixelwise Classification for Music Document Analysis." In *Proceedings of 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–6.

Castellanos, Francisco, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. "Document Analysis of Music Score Images with Selectional Auto-Encoders." In *Proceedings of the 19th International Society for Music Information Retrieval (ISMIR) Conference*. Paris, France.

Cavnar, William B., and John M. Trenkle. 1994. "N-Gram-Based Text Categorization." In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 161–75.

Cesarini, Francesca, Marco Lastri, Simone Marinai, and Giovanni Soda. 2001. "Encoding of Modified X-Y Trees for Document Classification." In *Proceedings of Sixth International Conference on Document Analysis and Recognition*.

Chanda, Sukalpa, Srikanta Pal, Katrin Franke, and Umapada Pal. 2009. "Two-Stage Approach for Word-Wise Script Identification." In *Proceedings of 10th International Conference on Document Analysis and Recognition*, 926–30.

Chanda, Sukalpa, Srikanta Pal, and Umapada Pal. 2008. "Word-Wise Sinhala Tamil and English Script Identification Using Gaussian Kernel SVM." In *Proceedings of 19th International Conference on Pattern Recognition*, 1–4.

Chanda, Sukalpa, Sukalpa Pal, and Fumitaka Kimura. 2007. "Identification of Japanese and English Script from a Single Document Page." In *Proceedings of 7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, 656–61.

Chaudhuri, Bidyut Baran, and Umapada Pal. 1997. "An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari (Hindi)." In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 2:1011–15.

Chen, Nawei, and Dorothea Blostein. 2007. "A Survey of Document Image Classification: Problem Statement, Classifier Architecture and Performance Evaluation." *International Journal of Document Analysis and Recognition (IJDAR)* 10 (1): 1–16.

Chiang, Holly, Yifan Ge, and Connie Wu. 2015. "Classification of Book Genres By Cover and Title," 5. Accessed August 14, 2022. https://www.semanticscholar.org/paper/Classification-of-Book-Genres-By-Cover-and-Title-Chiang-Ge/d0d0096d307a6da1332153b9cb8a72c29df38f87.

Dalitz, Christoph. 2018. "Gamera Classifier API." The Gamera Project. January 8, 2018. https://gamera.informatik.hsnr.de/docs/gamera-docs/classify.html#knninteractive.

Das, Arindam, Saikat Roy, Ujjwal Bhattacharya, and Swapan K. Parui. 2018. “Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks.” In *Proceedings of 24th International Conference on Pattern Recognition (ICPR)*, 3180–85.

Dengel, Andreas, and Frank Dubiel. 1995. “Clustering and Classification of Document Structure-a Machine Learning Approach.” In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 2:587–91.

Desale, Sanjay K., and Rajendra M. Kumbhar. 2013. “Research on Automatic Classification of Documents in Library Environment: A Literature Review.” *Knowledge Organization* 40 (5): 295–304.

Dhandra, Basanna V., H. Mallikarjun, Ravindra Hegadi, and Virendra S. Malemath. 2007. “Word-Wise Script Identification from Bilingual Documents Based on Morphological Reconstruction.” In *Proceedings of 1st International Conference on Digital Information Management*, 389–94.

Dhandra, B.V., and Mallikarjun Hangarge. 2007. “Global and Local Features Based Handwritten Text Words and Numerals Script Identification.” In *Proceedings of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, 2:471–75.

Dhanya, D., and A. G. Ramakrishnan. 2002. “Optimal Feature Extraction for Bilingual OCR.” In *Proceedings of the 5th International Workshop on Document Analysis Systems*, 25–36. DAS ’02. Berlin, Heidelberg: Springer-Verlag.

Dhanya, D., A. G. Ramakrishnan, and Peeta Basa Pati. 2002. “Script Identification in Printed Bilingual Documents.” *Sadhana* 27 (1): 73–82.

Diligenti, Michelangelo, Paolo Frasconi, and Marco Gori. 2003. “Hidden Tree Markov Models for Document Image Classification.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (4): 519–23.

Ding, Jie, Louisa Lam, and Ching Y. Suen. 1997. “Classification of Oriental and European Scripts by Using Characteristic Features.” In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 2:1023–27.

Dopehat39. 2018. *Ōtsuzumi and Kotsuzumi*. Photograph. https://upload.wikimedia.org/wikipedia/commons/c/cd/%C5%8Ctsuzumi_and_kotsuzumi.jpg.

Droettboom, Michael, Karl MacMillan, and Ichiro Fujinaga. 2003. “The Gamera Framework for Building Custom Recognition Systems.” In *Proceedings of the 2003 Symposium on Document Image Understanding Technologies*, 275–86. Greenbelt, MD.

Esposito, Floriana, Donato Malerba, and Francesca A. Lisi. 2000. “Machine Learning for Intelligent Processing of Printed Documents.” *Journal of Intelligent Information Systems* 14 (2): 175–98.

Ferrer, Miguel A., Aythami Morales, and Umapada Pal. 2013. “LBP Based Line-Wise Script Identification.” In *Proceeding of 12th International Conference on Document Analysis and Recognition*, 369–73.

Fix, Evelyn, and Joseph L. Hodges. 1951. “Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties.” 4. Randolph Field, Texas: USAF School of Aviation Medicine.

Fujikura, Keiichi, Björn Hammarfelt, and Claudio Gnoli. 2020. “Nippon Decimal Classification (NDC).” Text. Encyclopedia of Knowledge Organization. 2020. https://www.isko.org/cyclo/ndc.

Fujinaga Ichiro. 2019. “Single Interface for Music Score Searching and Analysis (SIMSSA) Project: Optical Music Recognition Workflow for Neume Notation.” In *Proceedings of the Computers and the Humanities Symposium*, 2019:281–86.

Fujinami, Shisetsu. 1975. *KANZE Sakon Okina*. Photograph. https://upload.wikimedia.org/wikipedia/commons/9/93/KANZE_Sakon_Okina.jpg.

Ghosh, Debashis, Tulika Dube, and Adamane P. Shivaprasad. 2010. “Script Recognition—A Review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (12): 2142–61.

Ghosh, Debashis, and A. P. Shivaprasad. 2000. “Handwritten Script Identification Using Possibilistic Approach for Cluster Analysis.” *Journal of the Indian Institute of Science* 80 (3): 215.

Hao, Shijie, Yuan Zhou, and Yanrong Guo. 2020. “A Brief Survey on Semantic Segmentation with Deep Learning.” *Neurocomputing* 406 (September): 302–21.

Harley, Adam W., Alex Ufkes, and Konstantinos G. Derpanis. 2015. “Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval.” In *Proceedings of 13th International Conference on Document Analysis and Recognition (ICDAR)*, 991–95.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. New York, NY: Springer New York Inc.

Heroux, Pierre, Sébastien Diana, Arnaud Ribert, and Eric Trupin. 1998. “Classification Method Study for Automatic Form Class Identification.” In *Proceedings. Fourteenth International Conference on Pattern Recognition*, 1:926–28.

Hiremath, Prakash S., Jagdeesh D. Pujari, S. Shivashankar, and V. Mouneswara. 2010. “Script Identification in a Handwritten Document Image Using Texture Features.” In *Proceedings of IEEE 2nd International Advance Computing Conference (IACC)*, 110–14.

Hochberg, Judith, Patrick Kelly, Timothy Thomas, and Lila Kerns. 1997. “Automatic Script Identification from Document Images Using Cluster-Based Templates.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (2): 176–81.

Holland, John H. 1992. “Genetic Algorithms.” *Scientific American* 267 (1): 66–73.

Howe, Nicholas R. 2013. “Document Binarization with Automatic Parameter Tuning.” *International Journal on Document Analysis and Recognition (IJDAR)* 16 (3): 247–58.

Hu, Jianying, Ramanujan Kashi, and Gordon Wilfong. 1999. “Document Classification Using Layout Analysis.” In *Proceedings of Tenth International Workshop on Database and Expert Systems Applications (DEXA)*, 556–60.

Ittner, David J., David D. Lewis, and David D. Ahn. 1995. “Text Categorization of Low Quality Images.” In *Proceedings of 4th Annual Symposium on Document Analysis and Information Retrieval*, 301–15.

Iwana, Brian Kenji, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, and Seiichi Uchida. 2017. “Judging a Book By Its Cover.” *ArXiv:1610.09204*, October.

Jaeger, Stefan, Huanfeng Ma, and David Doermann. 2005. “Identifying Script on Word-Level with Informational Confidence.” In *Proceedings of Eighth International Conference on Document Analysis and Recognition (ICDAR’05)*, 1:416–20.

Jain, Rajiv, and Curtis Wigington. 2019. “Multimodal Document Image Classification.” In *Proceedings of International Conference on Document Analysis and Recognition*, 71–77.

Japan Library Association. n.d. “NDC10 Ni Yoru Bunrui Kigojun Hyomokuhyo（ＮＤＣ１０による分類記号順標目表).” Japan Library Association (日本図書館協会). Accessed April 13, 2022. http://www.jla.or.jp/Portals/0/data/iinkai/bsh/ndc10.pdf.

Junker, M., and R. Hoch. 1997. “Evaluating OCR and Non-OCR Text Representations for Learning Document Classifiers.” In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 2:1060–66.

Kang, Le, Jayant Kumar, Peng Ye, Yi Li, and David Doermann. 2014. “Convolutional Neural Networks for Document Image Classification.” In *Proceedings of 22nd International Conference on Pattern Recognition*, 3168–72.

Kanze Sakon. 1934. *Ashikari, atsumori, tokusa, aoinoue, rinzo (芦刈・敦盛・木賊・葵上・輪蔵)*. Kanzeryu showa ban (観世流昭和版) 22. Tokyo: Hinoki shoten (桧書店).

Kanzeryu Yokyoku Kenkyukai. 1929. *Kanzeryu yokyoku fushi no zukai*. Osaka: Yoshida Yokyoku Shoten.

Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. 2021. "A Review on Genetic Algorithm: Past, Present, and Future." *Multimedia Tools and Applications* 80 (5): 8091–8126.

Lamb, Alex, Tarin Clanuwat, and Asanobu Kitamoto. 2020. "KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition." *SN Computer Science* 1 (3): 177.

Lateef, Fahad, and Yassine Ruichek. 2019. "Survey on Semantic Segmentation Using Deep Learning Techniques." *Neurocomputing* 338 (April): 321–48.

Li, Baojun, Shun Liu, Weichao Xu, and Wei Qiu. 2017. "Real-Time Object Detection and Semantic Segmentation for Autonomous Driving." In *Proceedings of Tenth International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR 2017), Automatic Target Recognition and Navigation*, 10608:167–74.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. "Fully Convolutional Networks for Semantic Segmentation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–40.

Lucieri, Adriano, Huzaifa Sabir, Shoaib Ahmed Siddiqui, Syed Tahseen Raza Rizvi, Brian Kenji Iwana, Seiichi Uchida, Andreas Dengel, and Sheraz Ahmed. 2020. "Benchmarking Deep Learning Models for Classification of Book Covers." *SN Computer Science* 1 (3): 139.

Ma, Huanfeng, and David Doermann. 2003. "Gabor Filter Based Multi-Class Classifier for Scanned Document Images." In *Proceedings of Seventh International Conference on Document Analysis and Recognition*, 968–72.

Maeda Shikanosuke. 1913. *Joruri Maruhon: dangogyou souyukabon*. Vol. 49. Osaka: Kashimaya.

Mckinney, Wes. 2010. "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, 56–61.

Miyake, Koichi. 1951. *Kanzeryu fushi no seikai (観世流節の精解)*. Tokyo: Hinoki syoten (檜書店).

Miyao, Hidetoshi. 2002. "Stave Extraction for Printed Music Scores." In *Proceedings of Intelligent Data Engineering and Automated Learning*, 562–68. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.

Montagner, Igor S., Nina S. T. Hirata, and Roberto Hirata. 2017. “Staff Removal Using Image Operator Learning.” *Pattern Recognition* 63 (March): 310–20.

Namboodiri, Anoop M., and Anil K. Jain. 2002. “Online Script Recognition.” In *Proceedings of Object Recognition Supported by User Interaction for Service Robots*, 3:736–39.

National Diet Library. 2019. “Joruri maruhon: daigogyo soyukabon ([浄瑠璃丸本] : 断五行双床本ひらかな盛衰記逆櫓松段).” National Diet Library Digital Collections. 2019.

———. 2021. “Statistics.” National Diet Library, Japan. 2021. https://www.ndl.go.jp/en/aboutus/outline/numerically.html.

———. n.d. “National Diet Library Online.” National Diet Library Online. Accessed April 12, 2022a. https://ndlonline.ndl.go.jp/#!/.

———. n.d. “Search Results.” National Diet Library Online. Accessed April 12, 2022b. https://ndlonline.ndl.go.jp/#!/search?available=ANYWHERE&ndc=768&searchCode=DETAIL.

Nishiyama, Kazuo. 1997. *Edo Culture: Daily Life and Diversions in Urban Japan, 1600–1868*. *Edo Culture*. University of Hawaii Press.

Nogami, Toyoichirō. 2005. *Japanese Noh Plays: How to See Them*. London, United Kingdom: Taylor & Francis Group.

Nohgaku kyokai (公益社団法人 能楽協会). n.d. “Kyokumoku database (曲目データベース).” The nohgaku performers’ association (能楽協会). Accessed April 12, 2022. https://www.nohgaku.or.jp/encyclopedia/program_db.html.

Obaidullah, Sk Md, Kaushik Roy, and Nibaran Das. 2013. “Comparison of Different Classifiers for Script Identification from Handwritten Document.” In *Proceedings of IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, 1–6.

Omagari, Toshio. 2010. “A survey of the use of classification schedules in Japan (わが国における図書分類表の使用状況：日本図書館協会「図書の分類に関する調査」結果より).” *Libraries Today (現代の図書館)* 2 (June): 129–41.

Padma, M.C., and P. A. Vijaya. 2009. “Monothetic Separation of Telugu, Hindi and English Text Lines from a Multi Script Document.” In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 4870–75.

Pal, Umapada. 2006. “Automatic Script Identification: A Survey.” *Vivek* 16 (3): 26–35.

Pal, Umapada, and Bidyut Baran Chaudhuri. 1999. “Script Line Separation from Indian Multi-Script Documents.” In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, 406–9.

Pal, Umapada, and A. Sarkar. 2003. “Recognition of Printed Urdu Script.” In *Proceedings of Seventh International Conference on Document Analysis and Recognition*, 1183–87.

Pal, Umapada, Nabin Sharma, Tetsushi Wakabayashi, and Fumitaka Kimura. 2007. “Handwritten Numeral Recognition of Six Popular Indian Scripts.” In *Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2:749–53.

Pan, Wumo M., Ching Y. Suen, and Tien D. Bui. 2005. “Script Identification Using Steerable Gabor Filters.” In *Proceedings of Eighth International Conference on Document Analysis and Recognition (ICDAR’05)*, 2:883–87.

Pati, Peeta Basa, S. Sabari Raju, Nishikanta Pati, and Angarai Ganesan Ramakrishnan. 2004. “Gabor Filters for Document Analysis in Indian Bilingual Documents.” In *Proceedings of International Conference on Intelligent Sensing and Information Processing*, 123–26.

Pati, Peeta Basa, and A. G. Ramakrishnan. 2006. “HVS Inspired System for Script Identification in Indian Multi-Script Documents.” In *Proceedings of the 7th International Conference on Document Analysis Systems*, 380–89. Berlin, Heidelberg.

Peake, G. S., and T. N. Tan. 1997. “Script and Language Identification from Document Images.” In *Proceedings of Workshop on Document Image Analysis (DIA’97)*, 10–17.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12 (85): 2825–30.

Pinnington, Noel John. 2019. *A New History of Medieval Japanese Theatre: Noh and Kyōgen from 1300 to 1600*. Cham: Springer International Publishing.

Quinlan, J. R. 1986. “Induction of Decision Trees.” *Machine Learning* 1 (1): 81–106.

Rani, Rajneesh, Renu Dhir, and Gurpreet Singh Lehal. 2013. “Script Identification of Pre-Segmented Multi-Font Characters and Digits.” In *Proceedings of 12th International Conference on Document Analysis and Recognition*, 1150–54.

Razzak, Muhammad Imran, Syed Afaq Hussain, and Muhammad Sher. 2009. “Numeral Recognition for Urdu Script in Unconstrained Environment.” In *Proceedings of 2009 International Conference on Emerging Technologies*, 44–47.

Roy, Kaushik, Alireza Alaei, and Umapada Pal. 2010. “Word-Wise Handwritten Persian and Roman Script Identification.” In *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition*, 628–33.

Roy, Kaushik, S. Kundu Das, and Sk Md Obaidullah. 2011. “Script Identification from Handwritten Document.” In *Proceedings of Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 66–69.

Roy, Kaushik, and Kinshuk Majumder. 2008. “Trilingual Script Separation of Handwritten Postal Document.” In *Proceedings of Sixth Indian Conference on Computer Vision, Graphics Image Processing*, 693–700.

Roy, Kaushik, Umapada Pal, and Bidyut Baran Chaudhuri. 2005. “Neural Network Based Word-Wise Handwritten Script Identification System for Indian Postal Automation.” In *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, 240–45.

Salzberg, Steven L. 1994. “C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993.” *Machine Learning* 16 (3): 235–40.

Sano, Yasuhiko. 2005. *Uta-You Shinobue and Nohkan*. Photograph. https://upload.wikimedia.org/wikipedia/commons/7/72/Uta-you_Shinobue_and_Nohkan.jpg.

Serper, Zvika. 2000. “‘Kotoba’ (‘Sung’ Speech) in Japanese Nō Theater: Gender Distinctions in Structure and Performance.” *Asian Music* 31 (2): 129–66.

Shin, Christian, David Doermann, and Azriel Rosenfeld. 2001. “Classification of Document Pages Using Structure-Based Features.” *International Journal on Document Analysis and Recognition* 3 (4): 232–47.

Singhal, V., N. Navin, and Debashis Ghosh. 2003. “Script-Based Classification of Hand-Written Text Documents in a Multilingual Environment.” In *Proceedings of Seventeenth Workshop on Parallel and Distributed Simulation*, 47–54.

“Sklearn.Tree.DecisionTreeClassifier.” n.d. Scikit-Learn. Accessed April 5, 2022. https://scikit-learn/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

Spitz, A. Lawrence. 1994. “Script and Language Determination from Document Images.” In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, 229–35. United States.

Takanori, Fujita, and Alison Tokita. 2008. *Nō and Kyōgen : Music from the Medieval Theatre*. Routledge Handbooks Online.

Tan, Tieniu. 1998. “Rotation Invariant Texture Features and Their Use in Automatic Script Identification.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (7): 751–56.

Taylor, Suzanne Liebowitz, Mark Lipshutz, and Roslyn Weidner Nilson. 1995. “Classification and Functional Decomposition of Business Documents.” In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 2:563–66.

Thoma, Martin. 2016. “A Survey of Semantic Segmentation.” *ArXiv:1602.06541*, May.

Tseng, Yu Ho, and Shau Shiun Jan. 2018. “Combination of Computer Vision Detection and Segmentation for Autonomous Driving.” In *Proceedings of 2018 IEEE/ION Position, Location and Navigation Symposium*, 1047–52. Institute of Electrical and Electronics Engineers Inc.

Ubul, Kurban, Gulzira Tursun, Alimjan Aysa, Donato Impedovo, Giuseppe Pirlo, and Tuergen Yibulayin. 2017. “Script Identification of Multi-Script Documents: A Survey.” *IEEE Access* 5: 6546–59.

Viglianseni, Gabriel, Jorge Calvo-Zaragoza, and Ichiro Fujinaga. 2018. “An Environment for Machine Pedagogy: Learning How to Teach Computers to Read Music.” In *Proceedings of the ACM IUI 2018 Workshops*, 4. Tokyo.

Wood, Sally L., Xiaozhong Yao, Kanthimathi Krishnamurthi, and Laurence Dang. 1995. “Language Identification for Printed Text Independent of Segmentation.” In *Proceedings of International Conference on Image Processing*, 3:428–31.

Zhou, Lijun, Yue Lu, and Chew Lim Tan. 2006. “Bangla/English Script Identification Based on Analysis of Connected Component Profiles.” In *Document Analysis Systems VII*, edited by Horst Bunke and A. Lawrence Spitz, 3872:243–54. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhu, Xiaofeng, Heung-Il Suk, Seong-Whan Lee, and Dinggang Shen. 2016. “Subspace Regularized Sparse Multitask Learning for Multiclass Neurodegenerative Disease Identification.” *IEEE Transactions on Biomedical Engineering* 63 (3): 607–18.

Noh chant and non-Noh-chant book data

Ezoshiya Sansaemon. 1700. *Taishokan nido no tamatori (大志よくわん二度の玉とり)*. Edo: Ezoshiya Sansaemon (ゑざうしや三左衛門).

Hosho Arata. 1936. *Hagoromo (羽衣)*. Syowa kaiteiban (昭和改訂版) 5. Tokyo: Shimogakari hoshoryu utaibon kankoukai (下掛宝生流謡本刊行会).

———. 1937. *Miwa (三輪)*. Syowa kaiteiban (昭和改訂版) 9. Tokyo: Shimogakari hoshoryu utaibon kankoukai (下掛宝生流謡本刊行会).

Hosho Kurou. 1916. *Hoshoryu utaibon tamasho (宝生流謡本 玉升)*. Tokyo: Wanya yokyoku syosi (椀屋謡曲書肆).

Kanze Kiyokado. 1897. *Yokyoku taisei (謡曲大成)*. Vol. 2. Kyoto: Hinoki Tsunenosuke (桧常之助).

———. 1909. *Kanzeryu utaibon takasago, tamura, eguchi, hanjo, ukai, naniwa, kanehira, senju, sotobakomachi, momijigari (観世流謡本 高砂・田村・江口・斑女・鵜飼・難波・兼平・千手・卒都婆小町・紅葉狩)*. Vol. 1. Kyoto: Hinoki Tsunenosuke (桧常之助).

Kanze Kiyotaka. 1881. *Fueno maki: Kanzeryu utai(笛之巻：観世流謡)*. Kyoto: Hinoki Tsunenosuke (桧常之助).

Kanze Motoshige. 1921. *Kureha, yamashi, oumu, komachi, kazuraki, taema (呉服・八嶋・鸚鵡小町・葛城・当麻)*. Tokyo: Hinoki Taikadou(桧大瓜堂).

———. 1923. *Kanzeryu shimai katatsuki: taisyo kaihan ten (観世流仕舞形附：大正改版 天)*. Vol. 10. Tokyo: Hinoki Taikadou(桧大瓜堂).

Kanze, Oribe, and Kiyokado Kanze. 1894. *Kanzeryu Utai Uchi Hyakujuban Kanyokyu (観世流謡内百拾番 34 咸陽宮)*. Vol. 34. Kyoto: Hinoki Tsunenosuke (桧常之助).

Kon Hachirouemon. 1886. *Hoshoryu banpo koutai (保生流万宝小うたひ)*. Kanazawa: Kon Hachiroemon (近八郎右衛門).

Kongo Ukyo. 1929. *Kuzu, Shokun, Shoki, Kokaji, Kasugaryujin (国栖・昭君・鐘馗・小鍛冶・春日竜神)*. Syowa kaiteiban soroibon(昭和改版揃本) 26. Tokyo: Hinoki shoten (桧書店).

———. 1931. *Kongoryu showaban shimai katatsuke(金剛流昭和版仕舞形附 第2 輯)*. Vol. 2. Tokyo: Hinoki shoten (桧書店).

———. 1932. *Kongoryu yokyoku zenshu (金剛流謡曲全集)*. Tokyo: Hinoki shoten (桧書店).

Nakao Tozan. 1908. *Haruno kyoku: shakuhachi onpu (春乃曲 : 尺八音譜)*. Osaka: Chikurinken (竹琳軒).

Nose Asaji. 1940. *Nogaku kenkyu (能楽研究)*. Tokyo: Yokyokukai hakkojo (謡曲界発行所).

Ookita Nobuhide. 1914. *Kanzeryu tokiwa utaibon bangai nakakon hokarokuban (観世流常盤謡本 番外 仲禅 他 6 番)*. Osaka: Tokiwakai (常盤会).

Oowada Tateki. 1900. *Utai to noh (謡と能)*. Nichiyo hyakka zensho (日用百科全書) 45. Tokyo: Hakubunkan (博文館).

Saitoh Yoshinosuke. 1920. *Nomen taikan jokan (能面大観 上巻)*. Tokyo: Nogaku syoin (能楽書院).

Sakamoto Seccho. 1914. *Nogaku shiron (能学私論)*. Gendai bungeisho (現代文芸叢書) 38. Tokyo: Shunyodo (春陽堂).

Tachibana Kyokuou II, and IIda Koshun. 1915. *Chikuzen biwautashu (筑前琵琶歌集)*. Tokyo: Kawagoe shoten (川越書店).

Takahama Kyoshi, and Sakurama Kintaro. 1940. *Aoniyoshi (青丹吉)*. Tokyo: Wanya shoten (わんや書店).

Tamai seibundo henshubu (玉井清文堂編輯部). 1929. *Kanadehon chushingura: keikobon koikano ishu (仮名手本忠臣蔵 : 稽古本 恋歌の意趣)*. Tokyo: Tamai Seibundo (玉井清文堂).

Tan Kabo. 1883. *Kancho warabeno satoshi daikyugo (勧懲童之論 第9 号)*. Vol. 9. Kanazawa: Kon Hachiroemon (近八郎右衛門).

Teihon tokiwazu zenshu kankoukai. 1943. *Teihon tokiwazu zenshu makino go (定本常磐津全集 巻之 5)*. Vol. 20. Tokyo: Teihon tokiwazu zenshu kankoukai(定本常磐津全集刊行会).

Terada, Kumajiro. 1885. *Sankyoku Jo (三曲 上)*. Vol. 1. Kyoto.

Tokuhiro Taimu. 1899. *Seikyodo ichigenkin fu makino san (清虚洞一絃琴譜 巻之 3)*. Vol. 3. Kyoto: Tokuhiro Jiro(徳弘時聾).

Tougi Fuminori. 1894. *Gakakushu hoshohu (雅楽集 鳳笙譜)*. Vol. 3. Tokyo: Tougi Fuminori(東儀文礼).

Toundo. 1892. *Goban tadanobu homare no tamamono kajiwara heizo homare no ishikiri kinkanban kyokaku kaoyose: kyogen sujigaki (碁盤忠信誉賜物・梶原平三誉石切・金看板侠客顔寄：狂言筋書)*. Nagoya: Toundo (東雲堂).

Yokoi Haruno. 1930. *Yokyoku to nogakutsu (謡曲と能楽通)*. tsusosho (通叢書) 32. Tokyo: Shiroku shoin (四六書院).

Zeami Motokiyo, and Nose Asaji. 1947. *Kadensho nosakusho: kochu (花伝書・能作書：校註)*. Osaka: Shin nihon tosho (新日本図書).

Glossary

aya fushi	the most fundamental fushi notation
connected component	a group of pixels that are connected with each other
dakuten	a pair of double dot-like symbols added to hiragana to shift consonants, such as fu (ふ) to bu (ぶ)
decision tree classifiers	a supervised learning method that forms a tree-like structure of decisions
document classification	the automated identification of a document to its corresponding categories, such as technical papers, business letters, and magazines
furigana	reading aid for kanji characters written in smaller syllabic characters
fushi notation	musical notation in Noh chant book. Indicates musical speech to be performed with a rhythm and melody and describes relative intervals
Gamera	an open-source framework for building document analysis applications, which enable a user to process document without formal technical background
goma fushi	another way of calling fushi notation
hiragana	phonetic lettering system used in Japanese writing
image features	visual elements within a document either extracted directly from an image or after an image has been segmented
kanji	logographic Chinese characters used in Japanese writing
kotoba	dialogue part in Noh play with a unique tune, yet it does not have a defined melody
kozutsumi	a small drum
kyogen	performing arts without chant often in comic dialogue
layer separation	a process of document analysis, which separate pixels in an image into classes.
noh	traditional Japanese performing art of Japan
ozutsumi	a large drum
Rodan	a web-based workflow engine for optical music recognition
sarugaku	comedic performance with skits, acrobatics, and magic known as the origin of Noh
script identification	a subcategory of document classificaion, and it is a process to identify a type of script before using a document analysis algorithm or character recognition
semantic segmentation	an image analysis method, which partitions each region of the object accurately at pixel level within a image by delineating their boundaries

shite	antagonist in Noh play
symbol classification	a process of document analysis, which classify connected compoenents from a separated layer
taiko	a drum with wider body
tsure	companions of shite
tsuyogin	a type of chant style; chanted with the strong impression, varying more of intensity than melodic range
ukion	a note that appears during a pitch transition
utai	chant part in Noh play characterized by a melody determined by relative pitch
utaibon	the Japanese term for Noh chant book
waki	protagonist in Noh play
wakitsure	companions of waki
yowagin	a type of chant style; chanted by a soft and melodious voice

Acronyms

CNN	convolutional neural network
FCN	fully convolutional network
GA	genetic algorithm
GLCMs	grey-level co-occurrence matrices
GMM	gaussian mixture modeling
HMM	hidden markov model
IIIF	international image interoperability framework
KNN	k-nearest neighbor
MEI	music encoding Initiative
NDC	nippon decimal classification
OCR	optical character recognition
OMR	optical music recognition
RVL-CDIP	Ryerson vision lab complex document information processing
SAE	selection auto-encoder
SIMSSA	single interface for music score searching and analysis
SVM	support vector machine