




Tracing the origins of incunabula through the automatic identification of fonts in digitised documents

Javier Lacasta¹  · Javier Noguerras-Iso¹ · F. Javier Zarazaga-Soria¹ · Manuel-José Pedraza-Gracia²

Received: 2 February 2021 / Revised: 27 January 2022 / Accepted: 3 April 2022 /

Published online: 14 May 2022

© The Author(s) 2022

Abstract

Incunabula are the texts printed mainly during the second half of 15th century that are a key cultural element in a revolutionary period of the history and evolution of the book and the printing. In these books, the identification of their origin largely affects its academic, cultural, patrimonial, and economical value. This paper proposes a process to automate the identification of the origin of a digitised incunabula document using the Proctor/Haebler method, a commonly established procedure in the field. This process has been validated with a selected dataset obtained from the incunabula collection at the digital repository of the University of Zaragoza.

Keywords Font identification · Machine learning · Neural networks · Incunabula

1 Introduction

The word *incunabula* refers to books, pamphlets and broadsides printed in Europe with metal types in the first times of the printing press up to the approximate date of 1st January 1501 (books printed between 1501 and 1520 are referred as *post-incunabula*). They are a key cultural element in a revolutionary period of the history and evolution of the book and the printing. They have been frequently studied by researchers to know their history and features but, depending on their preservation status, finding information about them may be complex. About a third of them do not contain any information about their imprint process, such as the printer, date or edition and are known as *sine notis* incunabula [7]. Many others have lost the cover (front page) or colophon, which contained such information, or are too deteriorated (*mutilus* incunabula). Finally, in an important number of cases only a few page fragments remain from the original book and lack imprint information (*membra disjecta*).

✉ Javier Lacasta
jlacasta@unizar.es

¹ Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, Zaragoza, Spain

² Institute of Heritage and Humanities (IPH), Universidad de Zaragoza, Zaragoza, Spain

The identification of the printing place and printing date of works, editions, variants and technical characteristics of documents whose origin was initially unknown has a direct impact on the increase of their academic, cultural, patrimonial, and economical value. Traditionally, one way to do this identification task has been the study and comparison of the fonts used in the documents with the ones used in incunabula with known origin. The origin of this idea is that the types representing each letter or symbol were cast from handmade matrix moulds, making them unique. Tracing such fonts to the printer who used them makes possible to identify the printing place and an approximate printing date of the documents. In addition, this font analysis has also a high relevance for the further text processing of digitised documents. Text recognition performed by modern Optical Character Recognition (OCR) tools requires training with the fonts used in the documents to have a high accuracy [21].

Henry Bradshaw was the first to consider incunabula fonts as a true digital fingerprint of the editions and, in 1870, published a comprehensive font classification of such documents [2]. Currently, the most used methodology for incunabula font identification tasks is the Proctor/Haebler system [17]. Upon the work of Bradshaw, this methodology was developed at the end of the XIX century by Robert Proctor (1898-1903) at the British Museum (currently British Library) and the Bodleian Library of Oxford [15]; and later by Konrad Haebler (1905-1924) at the Berlin Royal Library (currently Berlin State Library). Proctor classified the incunabula fonts by the height of 20 lines of the unique handcrafted letter casts of each font. Some years later, Haebler completed the method adding the identification of a prototypical letter as an additional parameter. He identified that the most different letter for the Gothic fonts was the “M” and for the Roman fonts it was the “Q”. Therefore, he proposed that once the Proctor measure was taken, the shape of the uppercase “M” or “Q” could be used to provide a control and confirmation measure of the font. As result of his work, between 1905 and 1924, Haebler published the *Typenrepertorium der Wiegendrucke* (TW), an exhaustive description of known fonts that includes information about the printing offices, printers and books using these fonts [9]. Nowadays, TW is constantly maintained by the Berlin State Library and accessible through a web site [14], which is also linked to an associated incunabula bibliographic index known as *Gesamtkatalog der Wiegendrucke*. Proctor/Haebler classification is not exempt from criticism and other alternative classifications methods for fonts have been developed over the years, but none has been successful in replacing the Haebler TW work [18].

Figure 1 illustrates how to use the Proctor/Haebler classification with an incunable page fragment. The Proctor measure is computed with 20 text lines, but the left paragraph is smaller than that. In this case, the height of 5 lines has been measured and multiplied by 4 to obtain a Proctor height of 134 mm (134G). Apart from the imprecision of this measure, this value is not enough to identify the corresponding font annotated in TW or similar databases: there are 9 Gothic fonts with a height of 134 mm, and 45 Gothic fonts if a height range between 132 mm and 136 mm is considered. Therefore, the Haebler “M” classification is needed to filter the correct font. Looking at Haebler classification, the shape of the “M” in the paragraph corresponds with M89 type. A search in TW for a font compatible with these measures allows us to identify the font as the one used by Paul Hurus in Zaragoza (Spain) between 1485 and 1499.

Although the Proctor/Haebler method is commonly used by experts in the field, to our knowledge there are no works in the literature automating its application. The main contribution of this paper is to propose a process to automate the identification of the origin of a document using the Proctor/Haebler methodology on Gothic fonts simplifying the identification of “sine notis” and “membra disjecta” printed matter. The performance and

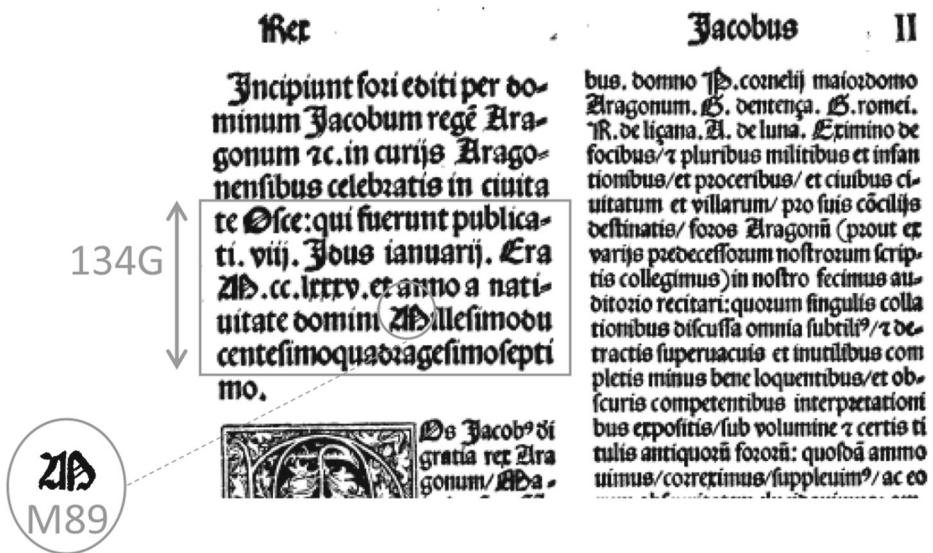


Fig. 1 A page fragment of “Fori regni Aragorum”, printed in 1496 by Paul Hurus at Saragosse (Spain) [24]

limitations of the different steps in the proposed method are also indicated. The paper focuses on Gothic fonts as a first approach to the problem, which can be generalized in the future to deal also with Roman fonts identified by a “Q”. Our process starts by splitting the document scanned image into pages (if it contains more than one) and cleaning the contained noise. Then, the bounding boxes of the text lines are extracted and measured according to the Proctor method. Next, the text lines are processed to identify the contained “M”s and classify them into the Haebler categories. Finally, the font is selected by matching the Proctor/Haebler measure with those in the TW.

The rest of this paper is structured as follows. Section 2 describes the works related to our proposal. Then, the proposed method is explained in Section 3. Section 4 compiles the experiments performed to validate the proposal with a selected dataset of page images obtained from the incunabula collection available at the digital repository of the University of Zaragoza [24]. Finally, Section 5 ends with some concluding remarks and an outlook on future work.

2 State of the art

There are multiple works in the literature that process images or text documents to extract their features and classify them. For instance, Galdekallu et al. [6] use PCA and the Whale optimisation algorithm for extraction and selection of features in images. Hakak et al. [10] use an ensemble of a decision tree, a random forest and an extra tree classifier for the classification of fake news. In addition, nowadays, convolutional neural networks are frequently used for image analysis given their excellent performance. For example, Liao et al. [13] use convolutional neural networks for the automatic learning of image features to detect manipulation in a forensic context. Liao et al. [12] also use a convolutional neural network to validate their steganography proposal.

However, there are few works in the literature focused on processing text images for the automatic identification of text fonts. Christlein et al. [4] propose the identification of fonts in the TW catalogue by training a deep neural network with sample paragraphs of each font. As proof of concept, they test if they are able to identify the fonts of some text fragments after training the network with other ones. A more general approach is the one proposed by Seuret et al. [19], whose objective is to identify the family to which a font belongs, such as Gothic, Textura or Bastarda. They present a dataset for the recognition of font groups and compare a set of neural network architectures to determine the best model for this task. Haraguchi et al. [11] describe a system for font comparison. Their objective is to identify if two different letters belong to the same or different font. They do it by training a neural network with letters of a wide variety of fonts. Similarly, Baluja [1] proposes another neural network model that not only identifies if some characters are from different fonts, but also is able to generate a complete font from a sample of characters. Table 1 shows a comparison of these works in terms of their purpose, type of processed data, and the neural network architecture that has been applied.

In the context of text recognition, there are several works focused on making OCR of incunabula or historical documents. For this task, they have to deal with additional problems such as heterogeneous page compositions, the use of mixed hand-made typefaces (including ligatures), varied concentration of ink, the logical degradation of paper and the poor quality of the digitised page images [8, 22]. Vijayarani and Sakila [25] make a comparison of some general OCR tools in terms of features and performance. However, although these tools can work properly with modern fonts, they are not trained to deal with incunabula Gothic fonts. Reul et al. [16] describe a tool suitable for incunabula OCR that includes all the steps from page segmentation to word recognition, but it has to be previously trained with the fonts of the book to process. To solve this lack of trained models, Springmann et al. [23] provide a dataset for OCR containing pairs of original images extracted from historical documents (dated between 15th and 19th centuries) and their corresponding manual transcriptions, which can be used to train models for character recognition. However, it only focuses on a specific set of documents of different periods that do not cover all the existent incunabula fonts available at the TW catalogue.

This work has the same purpose as the work of Christlein et al. [4], i.e. the identification of the font of a given text. However, the approach used is different. Instead of training a neural network with previously annotated text paragraphs, our work aims at detecting automatically blocks of lines and at classifying the shape of “M” characters as a reference for font identification, which is the proposed approach to mimic the Proctor/Haebler methodology. In this sense, our work is closer to Haraguchi et al. [11] and Baluja [1], which also study character shapes although with a different purpose of font identification. With respect to the network used for the “M” recognition, the two VGGs that have been used are similar to the one proposed by Seuret et al. [19], but adapted to the specific recognition task in this paper.

Table 1 Font processing works

	Purpose	Training data	Network architecture
Christlein et al. [4]	Font identification	Text paragraphs	DenseNet
Seuret et al. [19]	Font identification	Text pages	Residual, VGG, DenseNet
Haraguchi et al. [11]	Font equivalence	Characters	Two stream CNN
Baluja [1]	Font equivalence	Character sets	Fully connected

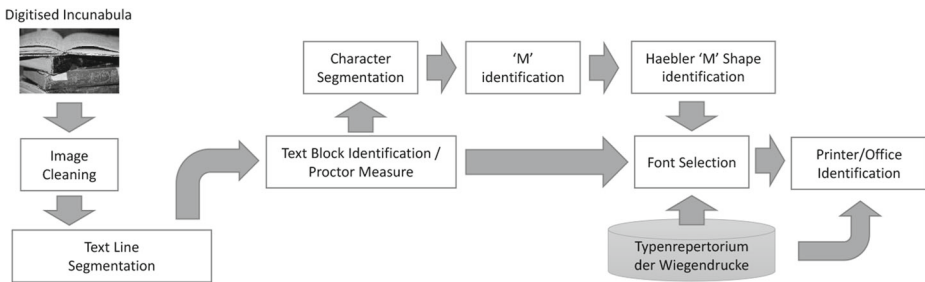


Fig. 2 Process for the identification of fonts and the associated printer/office

3 Identification process

The process developed to identify incunabula origin is described in Fig. 2. This process uses classic OCR cleaning and text/character segmentation methods as preprocessing steps. The novelty of the approach lies in applying automatically the Proctor/Haebler method to the obtained text blocks and characters to identify the font. The Proctor height is obtained by identifying blocks of consecutive segmented lines. Proctor blocks are processed in search of “M”s and those “M”s found are classified according to the different types of “M” shapes catalogued by Haebler. This “M” shape classification is performed with a neural network trained with the alphabet images associated with the fonts catalogued in TW. Finally, a font with the identified features is searched in the TW, and the details of the printer office can be retrieved.

The following subsections describe in detail the different parts of this process.

3.1 Preprocessing and text segmentation

The first step in the process is the cleaning of the input image to facilitate segmentation. It focuses on removing stains, discolorations, and correcting tilted lines since they have proven to produce more segmentation errors. In the documents analysed in the experiments, a simple black and white conversion followed by the opening morphological transformation has proven to be enough to remove most of the noise. With respect to line tilting, it mainly occurs when the image includes two consecutive pages of a book. In these cases, book owners have not forced the book spine when scanning (nor used a modern scanning device), so the scanned lines are tilted. This is solved by dividing the images in the two contained pages (by the spine) and correcting the tilt of each page separately. These transformation operations have been performed making use of the functions provided by the OpenCV library.¹

The identification of the lines and characters in the image is done using the Tesseract OCR text segmentation library.² The provided segmentation algorithms are based on the analysis of the image composition and the separation between the elements. This has worked well for line segmentation, but it produces many errors when segmenting characters. The use of ligatures in some fonts and the ink spreading in the printing process makes the characters frequently connected and because of this are not correctly segmented. Although this

¹<https://opencv.org/>

²<https://github.com/tesseract-ocr/tesseract>

problem is not so common with uppercase letters (they are bigger and have more space around), most of the classification issues in the experiments are caused by this problem. This has led us to make the system flexible enough to allow the manual selection of text blocks and “M”s in cases where the segmentation fails.

3.2 Identification of text blocks and computation of proctor measure

The text lines identified by Tesseract are aggregated in blocks of consecutive lines to compute the Proctor height measure. It is considered that two lines are consecutive and belong to the same text block if they are reasonably horizontally aligned; have a similar width, and height; and the vertical gap between lines is limited. According to the initial tests with incunabula documents, the general value used to establish the threshold for height/width similarity, horizontal alignment and vertical gap has been set experimentally to 5.5 mm. This has allowed us to identify the end of a text block when a distorted line is found (due to noise or margin handwriting), or when a new paragraph starts (with the same font or a different font).

The height of these blocks is measured in pixels and transformed into mm using the image resolution. If the obtained text blocks contain more than 20 lines, only these blocks are considered to scale the height of 20 lines, i.e. the calculation of the Proctor measure. Furthermore, only these blocks of 20 lines are considered to continue with the process of identifying “M”s. In case there are not blocks of at least 20 lines, an equivalent process is done with blocks of 10 or 5 lines and their height is again scaled to estimate the height of 20 lines.

3.3 Identification of “M”s and Haebler “M” shapes

To get the Haebler “M”s classification of each text block detected along with the computation of the Proctor measure, their lines are segmented with Tesseract to extract the characters inside. These characters are submitted to a convolutional neural network that separates “M”s from other characters. The found “M”s are then submitted to a second neural network that identifies the type of “M” according to the Haebler classification. The identification is done in two steps to facilitate the font identification when the automatic character segmentation fails. Thanks to this, if an image is incorrectly segmented, the user can manually select the “M” to analyse and identify its Haebler type with the second neural network. Obviously, if the text block does not contain an “M”, the font identification is only done with the Proctor measure. In this case, all the alternative fonts with the same Proctor height are returned, and the user has to study them individually to identify the correct font.

Multiple variations of the VGG-16 architecture [20] have been tried to select suitable convolutional network configurations for the two classifiers in the system. Figure 3 shows the main components in these kinds of networks. Their input is a collection of images scaled to the neural network input layer size. These images go through a set of processing blocks consisting of a few ReLU convolutional layers ending with a max-pool layer. The last processing block is connected to a stack of fully connected ReLU layers that ends in a SoftMax classifier. This general model has been adapted to the specific features of the desired tasks of “M”, and “M” type identification.

For the distinction between “M”s and non “M”s, the classification speed is vital. A text page may contain thousands of characters to check and, with a deep neural network, such as the original VGG-16, this process takes several minutes in a production server without GPU. To reduce classification time to a few seconds, the network dimensions have been reduced

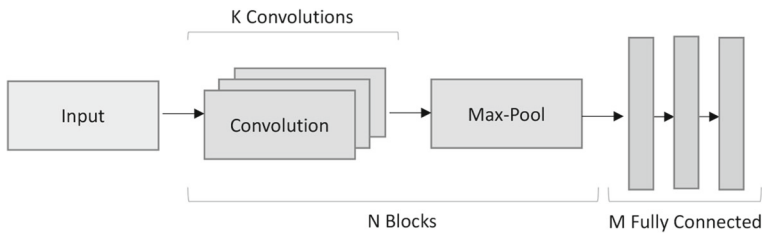


Fig. 3 Structure of VGG convolutional neural networks

as much as possible, searching for a compromise between time and performance. The final configuration shown in Fig. 4 has an input layer of a quarter of the biggest alphabet letter in the processed fonts so the images have to be scaled to these pixel dimensions. Additionally, it only contains two processing blocks consisting of a convolutional (3x3) and a max-pool (2x2) layer, a fully connected layer and the final SoftMax classifier that indicates if the input is an “M” or not. To increase the performance, the fully connected layer receives the original millimetre dimensions of the character image as an additional input. This input facilitates the classification because uppercase “M”’s have dimensions and ratios quite different from many of the other upper and lowercase characters. In the experiments, if the characters have been correctly segmented, this configuration has proven to generate very few false positives and almost none false negatives.

To identify the type of an “M” according to Haebler classification, the focus has been put on the precision of the network. The graphical features that allow distinguishing between types of “M”’s are many times subtle and can be easily mistaken. Additionally, since the number of “M”’s in text is reduced, the classification speed is not so relevant, and it is not a problem if it takes a few seconds to identify each one. For this task, the VGG-16 layer configuration used includes an input layer with the pixel dimensions of the biggest alphabet letter in the training collection.

The training of the two networks has been done using the process described in Fig. 5. As training data, a collection of alphabet images of each font provided in the TW has been used. If an alphabet image is not available for a given font it is ignored as the networks cannot be trained to recognise it. The alphabets are segmented into characters using Tesseract, and the segmentation has been manually reviewed to correct segmentation errors and annotate the alphabet “M”’s.

The neural network for “M” identification uses as training data the characters in the alphabet images divided between “M”’s and not “M”’s. Since the two categories are very

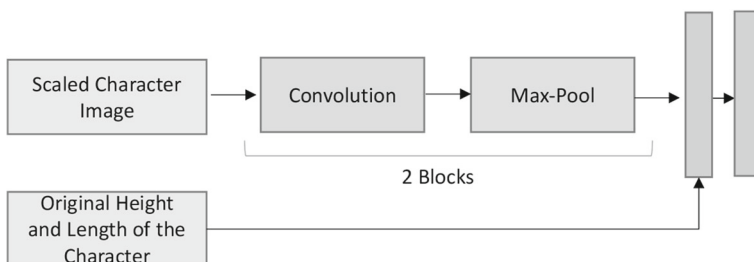


Fig. 4 “M”’s detector convolutional neural network

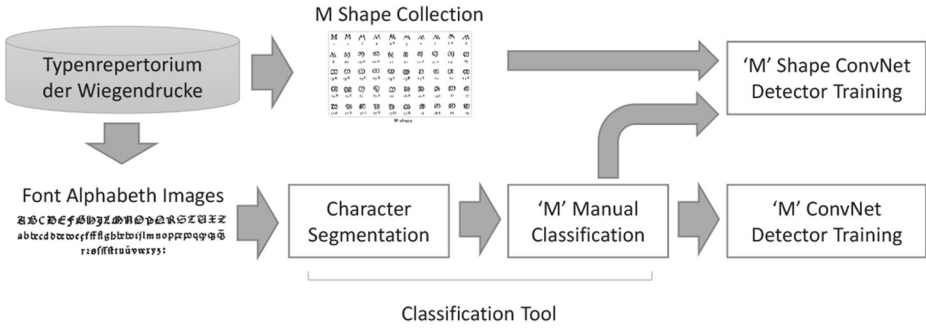


Fig. 5 Training process of convolutional neural networks for Haebler classification of “M”s

unbalanced (there are many more non “M”s than “M”s), the “M” samples have been multiplied by adding rotation noise, so the classifier is less dependent on a precise horizontal alignment of the characters. The neural network that performs the Haebler classification uses as training data the “M” characters of each font organized by their Haebler type. In this case, there are few samples of each type of “M”, so they are also multiplied with rotation noise. The lack of training samples may lead to overfitting, but in this specific context it is not a problem. Overfitting makes a network not to deal well with elements different from the ones used in training. However, in this context, there is almost no variety. Since printed “M”s of a font are all based on the same movable types, all occurrences are almost equal in all the documents containing them.

3.4 Font and printer identification

The final step in the process is to use jointly the obtained Proctor height and the Haebler classification to search for compatible fonts in the TW catalogue.

To facilitate the identification task, the information provided by the Berlin State Library through the TW web site has been transformed into a semantic repository, i.e. an RDF triple store. In January 2018, TW contained information about 13,419 distinct materials, 1,990 printing offices and 2,174 printers (persons). To illustrate the information provided by TW, Fig. 6 shows an example of its content. The figure contains the materials linked to the font “134G:M89” used in the example of Fig. 1. A material is an abstract concept for referring to fonts, initials, printer stamps, paragraph marks and title woodcuts. The font type report contains the identifier of the font, the height and the shape of the “M”. This is connected with the description of the printing office that has used this font and the person in charge of the office. This information allows dating a book if the used font is identified. Additionally, the TW provides an image with the alphabet of the font. That is, this alphabet image contain an image of each printed character belonging to the font. This alphabet image is used for training the neural networks for detecting “M”s and “M”s shapes.

Figure 7 shows the proposed model for the RDF-based representation of classes and properties associated with fonts, offices and incunabula books stored as resources in the semantic repository. The terms with the prefix *incunabula* belong to the *incunabula* vocabulary defined. However, whenever it has been possible, properties from well-known vocabularies such as the DCMI Metadata Terms (terms with *dct* prefix in Fig. 7) [5] or the FOAF Vocabulary (terms with *foaf* prefix in Fig. 7) [3] have been reused. This RDF-based representation has two main advantages. On the one hand, it is flexible enough to include

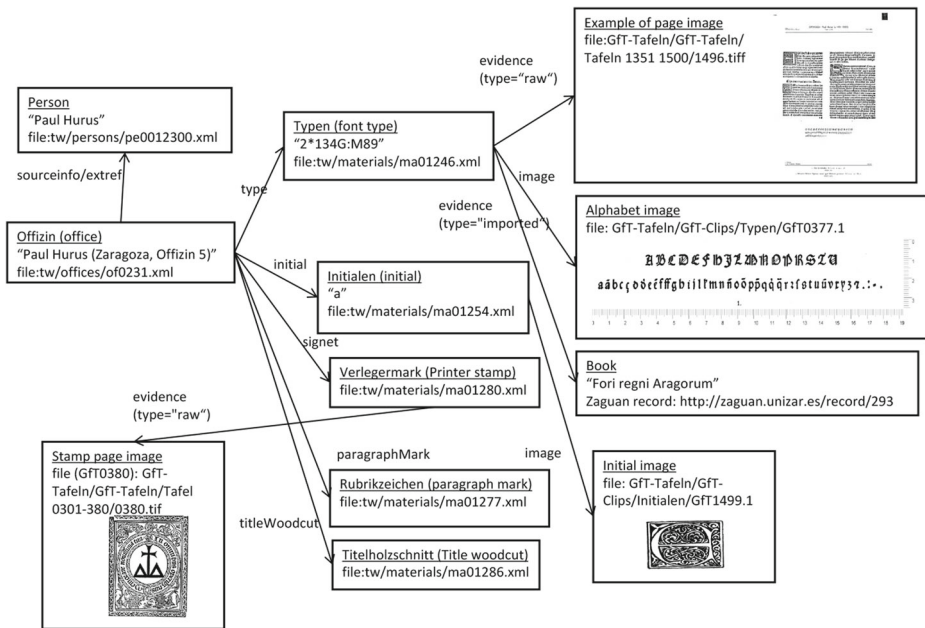


Fig. 6 A schematic example of the font-related information provided in TW

more properties to describe our target resources, if required. On the other hand, it can be accessed in a standardized way through an SPARQL end-point by any researcher interested in querying the content.

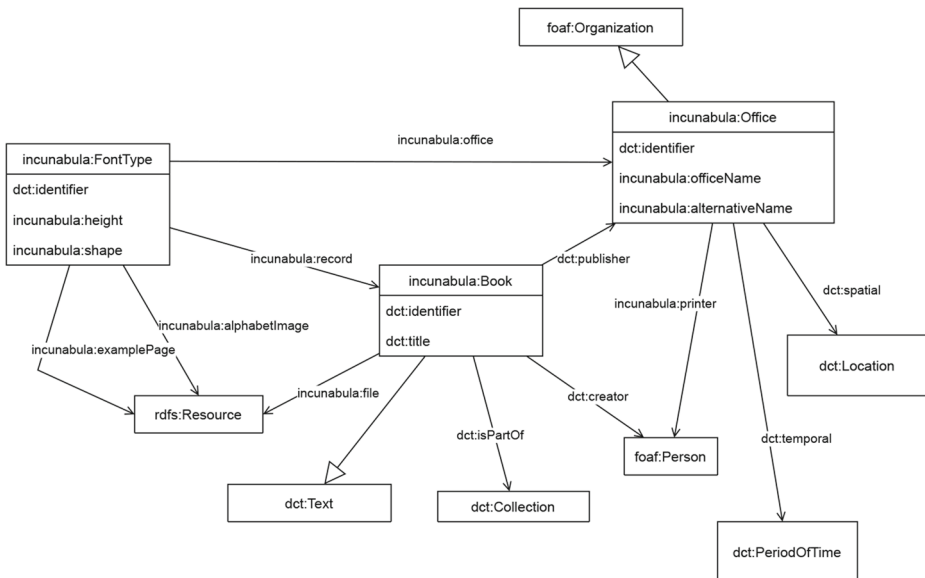


Fig. 7 Proposed model for the RDF-based representation of fonts, offices, and associated incunabula books

In the case of our proposed method, this semantic repository is used to look for the fonts having the height and “M” shape previously identified. In cases where there are not valid fonts with the identified height, a margin error, experimentally set to 2 mm, is taken into account for the Proctor measure. Last, after having identified the font, the information about the printing office and printer can be retrieved.

4 Experiments

The method has been validated with a selected dataset of page images obtained from the incunabula collection available at the digital repository of the University of Zaragoza, called Zagan [24]. This is a remarkable collection containing 384 different incunabula items with 413 digitised documents (some books have several digitised exemplars), which were originated by 165 different printing offices in Europe. Figure 8 shows a map with the printers and the location of their printing offices. Of them, 103 incunabula items (and their associated 116 digitised versions) are also referenced in TW as examples of 146 distinct fonts.

From the set of Zagan incunabula documents with known font in TW, a corpus consisting of 38 pages selected from 38 distinct documents has been created for testing experiments. The criteria for the selection of these documents and pages has been to check that they had been digitised with a 1:1 scale, that they were using a Gothic font, and that this font had a corresponding alphabet image available in TW. The Zagan documents used for testing are not used during training of the networks. This guarantees the validity of the results, as it shows that the training networks are able to work with information different from the one used in the training.

The neural network models for “M” recognition and “M” shape detection have been trained with the alphabet images of 79 different fonts. For this task, the following parameter configurations in both networks have been used. The training uses 1000 epochs with a batch-size of 32 and a validation split of 30% of the images. Since the alphabet images contain a



Fig. 8 Printers and location of printing offices originating the incunabula collection maintained in Zagan

Table 2 Experiment results

	Precision	Recall	F1 score
Proctor	0.87	1.00	0.93
Haebler	0.98	0.45	0.61
Proctor+Haebler	0.98	0.45	0.61

single occurrence of each font character, each character image has been multiplied 20 times with a random image generator. This image generator flips the character images vertically with a random rotation angle in the range of 0–4 degrees. This facilitates the identification of those occurrences that are not completely straight.³

Table 4 in the Appendix shows the results of our proposed system on these 38 pages. The tables indicate: how to locate the images in Zagan collection through the columns *Rec.* (record number in Zagan collection), *Title, File* (name of the PDF file containing the digitised document) and *Page* (page number together with an optional L/R to indicate if the image corresponds to the left or right part of the page in case of double page digitisation); the font used in these images through the columns *font* (TW font identifier), *H* (20-line height of this font) and *M* (M shape of this font); and features of the font discovered through columns *F.H.* (found 20-line height) and *F.M* (found M shape).

With respect to the Proctor measure, our system has found at least one text block in all pages suitable to compute the Proctor measure, and 87% of the images have been assigned the correct height of the font taking into account a margin error of 2 mm. The few errors in the detection of the Proctor measure are mainly due to the low quality digitisation of some documents or small skews in text columns.

Related to the correct identification of “M” shapes, the system detects “M”s in 45% of the images. As already indicated in Section 3.1, this lower percentage is due to the incorrect segmentation of characters performed by Tesseract in some cases. However, whenever the system has been able to detect an “M”, the system has detected the correct “M” shape in 98% of cases: only 1 of the 2 different “M”s detected in record 156 was wrong. Taking into account both the Proctor measure and the Haebler M classification, the system has identified the correct font in 45% of the page images.

A summary of the quality of the results obtained is shown in Table 2. It shows the precision, the recall and the F1 score of the Proctor and Haebler parts of our automated process, and the results for the whole process. In this context, the recall is the rate of pages in the corpus, where we have identified either Proctor line blocks (line blocks whose height can be measured) or “M” characters. Once line blocks or “M” characters have been detected, the precision is the rate of these items that have been correctly measured with respect to the expected Proctor height, or correctly classified with respect to the expected “M” shape. The precision and recall of the whole process is equivalent to the measures obtained in the row corresponding to the Haebler measures: only in the cases where an “M” is detected, our process provides a full decision. In all cases where an “M” was detected, the height was correct and the precision was dependent on the correct classification of “M” shape.

Focusing on the performance of the testing part of our automatic font identification approach, our purpose was that the neural network models obtained after training, together with the complementary software, could be executed in a computer without expensive hardware requirements. Therefore, we checked how it performed in a computer with an i5-4590

³The software used in the experiments can be found at <https://doi.org/10.6084/m9.figshare.18708032>.

Table 3 Mean time of document identification steps

Task:	Load	Segment	Proctor measure	M Search	Haebler classification	TW Search
Seconds:	1.4	4.6	0.01	6.1	1.8	0.01

processor without GPU. The result obtained has been that the process has taken a time of 13.9 seconds per document. This cost hinders its application to check all the pages in a book, but it is suitable for validation of selected pages. Table 3 shows a summary of the time spent in each step. It can be observed how the slower steps are the document segmentation and the “M” search. This is normal because each page contains around 1900 characters, on average, which must be segmented and classified. However, proportionally, the slower step is the Haebler classification, since it takes almost two seconds, on average, to process a very small number of “M”s in each page. This is caused by the dimensions of the used network and the lack of a GPU for making the prediction.

5 Conclusions and future work

This paper has described an automatic system to identify the fonts used in incunabula documents according to the Proctor/Haebler methodology. Using a database of font types and their associated resources modelled as a semantic repository, it has been possible to provide additional details of the fonts and the printing offices that used these fonts.

The experiments performed have proven the suitability of the process to study incunabula documents with unknown origin, or even to detect problems in documents with known origins. In the case of documents with known origin, the disagreements between expected fonts and discovered fonts may be due to two main reasons. First, the digitisation of documents had a bad quality and it did not maintain a 1:1 scale between the hard-copy version and the digitised version. Second, the document is probably using an additional font that has not been catalogued in TW yet, or whose alphabet image is not available.

With respect to the performance of the proposed process, it presents an important issue related to the document segmentation. Solving the segmentation problems is a challenge by itself. Images could be cleaned better so that the characters can be more easily identified. Nevertheless, the existing noise is very heterogeneous in its nature (e.g., stains, handwriting, ink fading, bad quality scanning. . .) and this may require an ad-hoc cleaning procedure for each document. Removing ligatures and ink spreading that merges characters is even more complex. It would need to know the character shape, but that is completely font dependant.

A main objective for future work will be to improve the quality of the results. For this task, it will be studied how to enhance the cleaning of documents and the development of a segmentation component focused on Gothic fonts features. As these are the main causes of problems in the current system, its improvement will have an important effect on the quality of the results. In addition, we want to focus on other complementary areas of work. Firstly, we want to enrich the contents of TW by documenting the use of additional fonts in incunabula, and providing new alphabet images. The lack of alphabet images for some fonts precludes recognising them, and therefore reduces the system coverage and its usability for identifying unknown documents. Then, we also want to extend the available document collection used for testing with documents from other libraries. Finally, we want to extend the system to apply OCR once the fonts are identified. An OCR system trained with the correct fonts could be used for automatic text transcription.

Appendix: Results of classification

Table 4 Results of classification with a selected dataset of incunabula images

Rec.	Title	File	Page	Font	H	M	F.H.	F.M	
11133	Quaestiones in quattuor libros...	L_292	72L	ma02525	66	M87_3	68		
112	Sermones quadragesimales de po...	L_260	25L	ma10244	76	M49	78		
139	Scotus pauperum super IV sente...	L_130	28	ma01162	71	M49	70		
141	Sermones de laudibus sanctorum...	L_132	25	ma12762	63	M50_1	63		
149	De patientia ...	L_139	21R	ma03339	65	M18	65	M18	
156	Summa universae theologiae siv...	L_147	15	ma02050	82	M87_1	82	M7,M87_1	
163	De anima. De intellectu et de ...	L_155	30	ma01123	84	M98_2	84		
219	Confessionale "Defecerunt" (en...	L_168	27L	ma01252	84	M22	85	M22	
223	Summa de casibus conscientiae ...	L_171	123L	ma03192	58	M99	58	M99	
236	Rationale divinatorum officiorum...	L_179	49	ma02501	92	M88_1	91		
238	Sermones de tempore et de sanc...	L_183	13R	ma02063	83	M88_1	83		
24	De civitate Dei ...	L_25	63	ma01185	84	M49	85	M49	
242	De proprietatibus rerum (en ca...	L_187	12R	ma04078	101	M22	101	M22	
246	Sermones de evangelio aeterno ...	L_192	18R	ma02383	82	M88_1	81	M88_1	
253	Theoremata de corpore Christi ...	L_199	23L	ma08182	75	M49	75	M49	
258	De exterioris et interioris ho...	L_204_A	8L	ma01526	98	M93	98		
261	Determinationes magistrales co...	L_206	65R	ma00923	92	M88_1	92	M88_1	
262	Mariale, seu Sermonarium de ex...	L_207	23R	ma07432	74	M49	75	M49	
268	De regimine principum (en cast...	L_212	12R	ma01526	98	M93	97		
272	De officiis, ...	L_216	24R	ma03067	69	M16	70	M16	
275	Moralia, seu Expositio in Job ...	L_219	60L	ma01123	84	M98_2	87		
291	De consolatione philosophiae. ...	L_232	67	ma04067	80	M7	80		
316	Conclusiones super quattuor li...	L_292	73L	ma02525	66	M87_3	68		
319	Opera ...	L_258	81	ma03127	69	M49	70	M49	
320	Sermonarium de peccatis per ad...	L_259	15	ma10173	75	M4	74		
321	Sermones de adventu ; Sermo de...	L_260	11L	ma10244	76	M49	78		
335	Vita et transitus S. Hieronymi...	L_275	52L	ma13487	104	M17E	107		
348	Legenda aurea sanctorum ...	L_283	34R	ma02920	76	M18	80		
349	Compendium theologiae veritat...	L_284	32L	ma03264	68	M91_1	70	M91_1	
354	Fortalium fidei ...	L_290	66R	ma01225	74	M72_2	76		
365	Super sapientiam Salomonis ...	L_073	21L	ma05473	80	M47	80		
371	De regimine principum ...	L_080	52L	ma03843	88	M38	88	M38	
4	Expositio super toto Psalterio...	L_2015	20	ma12011	104	M17E	105	M17E	
41	Sermones de tempore super Evan...	L_039	262R	ma02812	65	M21	62		
43	Expositio officii Missae sacri...	L_076	6R	ma12011	104	M17E	104	M17E	
65	Consolatio theologiae ...	L_064	42L	ma07182	88	M3	88		
79	Compendium theologiae veritat...	L_089	33R	ma02715	77	M38	76	M38	
9	Expositio evangeliorum dominic...	L_010	109L	ma10271	94	M38	88		
file count		38	correct count					33	17
distinct font type count				33					

Acknowledgements The authors wish to acknowledge *Staatsbibliothek zu Berlin* for facilitating the direct access to the electronic materials published at *Typenrepertorium der Wiegendrucke* web site, and the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work is part of the projects T59_20R and S65_20D supported by the Regional Government of Aragon (Spain).

Declarations

Conflict of Interests The authors declare that they have no competing financial interests or personal relationships that could have influenced the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Baluja S (2017) Learning typographic style: From discrimination to synthesis. *Mach Vis Appl* 28(5–6):551–568
2. Bradshaw H (1870) A classified index to the XVth century books in the late M. J. de Meyer collection sold at Ghent, november 1869. McMillan, London
3. Brickley D, Miller L (2014) FOAF Vocabulary Specification 0.99. <http://xmlns.com/foaf/spec/>. Accessed 11 Jan 2021
4. Christlein V, Weichselbaumer N, Limbach S, Seuret M (2021) Proof of concept: Automatic type recognition. In: Reussner RH, Koziolok A, Heinrich R (eds) *INFORMATIK 2020*, Gesellschaft für Informatik, Bonn, pp 1307–1316. https://doi.org/10.18420/inf2020_122
5. DCMI Usage Board (2020) DCMI Metadata Terms. <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>. Accessed 11 Jan 2021
6. Gadekallu TR, Rajput DS, Reddy M, Lakshmana K, Bhattacharya S, Singh S, Jolfaei A, Alazab M (2021) A novel pca-whale optimization-based deep neural network model for classification of tomato plant diseases using gpu. *J Real-Time Image Proc* 18(4):1383–1396
7. Geldner F (1978) *Inkunabelkunde Eine einföhrung in die Welt des frühesten Buchdrucks*. Reichert, Wiesbaden
8. Gupta A, Gutierrez-Osuna R, Christy M, Capitanu B, Auvil L, Grumbach L, Furuta R, Mandell L (2015) Automatic assessment of OCR quality in historical documents. In: *Twenty-Ninth AAAI conference on artificial intelligence*, pp 1735–1741
9. Haebler K *Typenrepertorium der Wiegendrucke*, 5 vols. Rudolf Haupt, Leipzig. 1905–1924
10. Hakak S, Alazab M, Khan S, Gadekallu TR, Maddikunta PKR, Khan WZ (2021) An ensemble machine learning approach through effective feature extraction to classify fake news. *Futur Gener Comput Syst* 117:47–58
11. Haraguchi D, Harada S, Iwana BK, Shinahara Y, Uchida S (2020) Character-independent font identification. In: *Proceedings of the 14th IAPR international workshop, DAS 2020, Wuhan, China, July 26–29, 2020*, pp 497–511
12. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. *IEEE Trans Circuits Syst Video Technol* 30(3):685–696
13. Liao X, Li K, Zhu X, Liu KR (2020) Robust detection of image operator chain with two-stream convolutional neural network. *IEEE Journal of Selected Topics in Signal Processing* 14(5):955–968
14. *Preussischer kulturbesitz* (2021) Web site of *Typenrepertorium der Wiegendrucke (TW)* at *Staatbibliothek zu Berlin*. <http://tw.staatsbibliothek-berlin.de/>. Accessed 1 Jan 2021

15. Proctor R (Unknown Month 1898) Index of the early printed books in the British Museum and in the Bodleian Library of Oxford. K. Paul, Trench, Trübner, B. Quaritch, London
16. Reul C, Dittrich M, Gruner M (2017) Case study of a highly automated layout analysis and OCR of an incunabulum: ‘Der Heiligen Leben’ (1488). In: Proceedings of the 2nd international conference on digital access to textual cultural heritage, pp 155–160
17. Rial Costas B (2012) El sistema Proctor-Haebler y el estudio de las letrerías en las impresiones góticas incunables. In: *Literatura medieval y renacentista en España: líneas y pautas*, Sociedad de Estudios Medievales y Renacentistas (SEMYR), pp 855–864
18. Rial Costas B (2016) Typefaces, fonts, and types: Toward a classification of Fifteenth-Century gothic “Types”. *Cataloging & Classification Quarterly* 54(5-6):384–396
19. Seuret M, Limbach S, Weichselbaumer N, Maier A, Christlein V (2019) Dataset of pages from early printed books with multiple font groups. In: Proceedings of the 5th international workshop on historical document imaging and processing, pp 1–6
20. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Bengio Y, LeCun Y (eds) 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. arXiv:1409.1556
21. Springmann U, Lüdeling A (2017) OCR of historical printings with an application to building diachronic corpora: A case study using the RIDGES herbal corpus. *Digital Humanities Quarterly* 11(2). <http://www.digitalhumanities.org/dhq/vol/11/2/000288/000288.html>
22. Springmann U, Najock D, Morgenroth H, Schmid H, Gotscharek A, Fink F (2014) OCR of historical printings of Latin texts: Problems, prospects, progress. In: Proceedings of the First international conference on digital access to textual cultural heritage, pp 71–75
23. Springmann U, Reul C, Dipper S, Baiter J (2018) Ground Truth for training OCR engines on historical documents in German Fraktur and Early Modern Latin. *Journal for Language Technology and Computational Linguistics (JLCL)* 33(1):97–114
24. Universidad de Zaragoza (2014) Digitised incunabula collection maintained at the digital repository of the University of Zaragoza (Zaguan). <https://zaguan.unizar.es/collection/incunables?ln=en>. Accessed 31 Dec 2020
25. Vijayarani S, Sakila A (2015) Performance comparison of OCR tools. *International Journal of UbiComp (IJU)* 6(3):19–30

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.