

Proof of Concept: Automatic Type Recognition

Vincent Christlein,¹ Nikolaus Weichselbaumer,² Saskia Limbach,³ Mathias Seuret¹

Abstract: The type used to print an early modern book can give scholars valuable information about the time and place of its production as well as its producer. Recognizing such type is currently done manually using both the character shapes of ‘M’ or ‘Qu’ and the size of the total type to look it up in a large reference work. This is a reliable method, but it is also slow and requires specific skills. We investigate the performance of type classification and type retrieval using a newly created dataset consisting of easy and difficult types used in early printed books. For type classification, we rely on a deep Convolutional Neural Network (CNN) originally used for font-group classification while we use a common writer identification method for the retrieval case. We show that in both scenarios, easy types can be classified/retrieved with a high accuracy while difficult cases are indeed difficult.

Keywords: type recognition; type classification; type retrieval; deep learning

1 Introduction

Type recognition is one of the central methods of analytical bibliography [Sc18, p. 42–70]. It is used to date printed books and identify both the printer and the publication place. Type recognition is traditionally done manually by using a combination of the type size (measured over 20 lines) and the characteristic shape of the letters ‘M’ or ‘Qu’ to identify the correct type in the *Typenrepertorium der Wiegendrucke* (TW), a reference work for all known incunabula types [Ha05].⁴ This method is slow and requires specific skills. Most importantly, it relies on the existence of a reference work that is difficult and time-consuming to compile, making it next to impossible to use type recognition for material beyond the incunabula period. For this approach, pattern recognition methods are extremely helpful as they increase the speed and ease of type recognition and thereby widen the scope of material that it can be applied to. For book historians, using type recognition for books printed in the incunabula period and beyond helps to answer important questions. In the early modern period, many books appeared without any indication about when and where the book was printed. This was often done when authors and printers feared political prosecution, but not limited to such cases. Type recognition would enable us to identify the producers of

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Pattern Recognition Lab, Martensstr. 3, 91058 Erlangen, Germany, vincent.christlein@fau.de, mathias.seuret@fau.de

² Johannes Gutenberg-Universität Mainz, Gutenberg-Institut für Weltliteratur und schriftorientierte Medien, Abt. Buchwissenschaft, Jakob-Welder-Weg 18, 55128 Mainz, Germany, weichsel@uni-mainz.de

³ Università degli Studi di Milano, Department of Economics, Management and Quantitative Methods, Via Conservatorio 7, 20122 Milano, Italy, dr.saskia.limbach@gmail.com

⁴ The *Typenrepertorium der Wiegendrucke* is available online: <https://tw.staatsbibliothek-berlin.de>

these books. On top of that, we would gain a better understanding of the material used in a given print shop, which could tell us more about the economic background of the printer. This paper explores the effectiveness of existing pattern recognition methods from writer identification and font group recognition.

2 Related Work

We want to investigate type recognition in two different ways: classification and retrieval.

Classification An early work for font classification [Wa15] – working on modern computer fonts – built a large dataset of real and mostly synthetic images of about 5000 classes and 2 mio. images. The proposed classifier achieves about 80 % accuracy on the test set. Closer related are the competitions and datasets for cursive script type classification [C116; C117], where the best method can differentiate between 12 script classes with an accuracy of about 85 % [Ch19b; C117]. In contrast, font group classification [Se19], seems to be an easier task achieving accuracies of about 98 % [We20]. Note that we strive to classify types, which are much more challenging than font groups since the differences are often much smaller.

Retrieval Type retrieval is closely related to other image retrieval tasks, such as writer retrieval for historical data. The current writer identification performance is well represented in the last image retrieval competitions [Ch19a; Fi17]. These competitions involved large datasets containing 3600 [Fi17] and 20 000 test images [Ch19a]. The accuracies vary widely (74 % to 97 %) depending on the data source and image quality. The current state-of-the-art approach for historical writer identification is given by Lai et al. [LZJ20]. They propose “pathlet” features, which they combine with SIFT [Lo04] and encode it in a novel bagged version of VLAD encoding [Jé12]. They achieve about 90 % Top-1 accuracy and outperform the previous unsupervised deep learning-based approach by Christlein et al. [Ch17b]. In this study, we evaluate the performance of a baseline writer identification method [Ch18] based on SIFT descriptors.

3 Dataset

tw lists about 6000 different types used between 1450-1500. Yet, it is impossible to use all of these types in this dataset. The sheer size of tw already presents a challenge, but it is mainly its design and original purpose which prevents us from using all of it. tw only lists the type used in a given book, but gives no indication if this type is the only type used in the book and – more importantly in this context – on which pages the type is used. Most books were printed with more than one type. Additional types were used in order to emphasize words, highlight headlines, etc. As of now the only way to select training data is to manually

label every image. This made us limit our dataset to 8 examples which were selected to illustrate how easy and how difficult it can be to differentiate between various types.

The easy examples, cf. Fig. 1a, consist of types with very distinctive shapes. *tw ma00131* is a Rotunda used by the Augsburg printer Anton Sorg between 1475 and 1478 in at least 19 editions of which we have a total of 21 digital copies. The type is very recognizable because of its unique decorative upper case characters. Note, the red highlights were later added by a contemporary hand and not printed.

tw ma00967 is a Textura used by an unknown printer from Salamanca from 1481 to 1490. While there is not as much surviving material for this type as for others (only 8 editions and 4 digital copies), the type is still easy to recognize for a human expert. The type is of rather poor quality and has a characteristic jagged look.

tw ma02771 is a Bastarda used by Jean du Pré in Lyon between 1489 and 1491. Relatively little material survives – *tw* lists 8 editions and only 2 digital copies –, but the type is rather large at 119 mm over 20 lines and with its looped ascenders and flourished upper case characters it uses very complex shapes that are often easier to attribute correctly than the regular shapes of e. g. a well-cut Textura.

tw ma04614 is a Textura with some unusual letter shapes used by Arnold ter Hoernen in Cologne from 1474 to 1482. It survives in 11 editions of which 5 copies are available as scans. The type is unusual in that most of its letter shapes follow the model of a Textura, but ‘f’ and ‘f’ have descenders and the lower-case ‘a’ is of the cc-type, not the uncial type that is more common for Textura types.

To represent the other end of the spectrum we selected two pairs of types that we assumed to be particularly challenging. *tw ma07487* and *tw ma07488* were both used by Bartholomäus Kistler in Straßburg, *ma07487* from 1498 to 1499 (2 editions, 2 digital copies) and *ma07488* from 1499 to 1501 (6 editions, 7 digital copies). The types share identical upper case Rotunda characters, but combine them with Bastarda lower case (*ma07487*) and Rotunda lower case (*ma07488*). Presumably, *ma07487* was a temporary fix as the production of the lower case was not yet finished. This kind of combination appears fairly often and poses particular challenges as parts of the types are not similar, but completely identical.

The second pair of difficult types, *tw ma07721* and *tw ma07718* were successively used by Johann Schäffler in Ulm from 1492 to 1494 (5 editions, 2 digital copies) and from 1496 to 1500 (22 editions, 11 digital copies). Both types are Upper Rhine Bastardas of very similar size and design. They do however not share any perfectly identical characters. *tw ma07718* only replaced *tw ma07721* after Schäfflers attempt to establish a print shop in Freising. It was apparently made to replace the older type [Am79, p. 370–371]. These types can be told apart by a human expert, but only by methodical comparison. At a cursory examination they can easily be mistaken for one another.



(a) Easy cases – tw ma00131 (top left), tw ma00967 (top right), tw ma02771 (bottom left), tw ma04614 (bottom right).



(b) Difficult case 1 – tw ma07487 (left) and tw ma07488 (right).



(c) Difficult case 2 – tw ma07721 (left) and tw ma07718 (right).

Fig. 1: Type dataset examples.

To create training data we used tw and its sister catalogue the Gesamtkatalog der Wiegendrucke (GW) [Ge5–]. For a given type we looked up all editions that are known to contain this type in tw. Via GW we searched for and downloaded all accessible scans of these editions. In the next step, we selected pages that contained only the respective type and in some cases cropped images in order to delete headlines, woodcuts and other irrelevant material. The dataset consists of 9083 labeled images and is publicly available.⁵

4 Classification

As a first step to approach this problem, we train a classifier to recognize the selected types. We use the same methodology as in [Se19]. The baseline method of this work consists of a Convolutional Neural Network (CNN) to classify the text of document images as belonging to different font groups, such as Antiqua, Fraktur or Textura.

4.1 Methodology

We use a DenseNet-121 [Hu17], a deep CNN with densely-connected blocks, to classify overlapping patches over the whole surface of the input image, and average the results. We

⁵ <https://doi.org/10.5281/zenodo.3923638>

Tab. 1: Confusion matrix containing the classification results. To enhance readability, the diagonal is bold. Rows correspond to types, and columns to classification results. The first four rows depict the easy cases while the last four rows show the difficult cases.

prediction →	ma00131	ma00967	ma02771	ma04614	ma07487	ma07488	ma07718	ma07721
ma00131	160	0	0	0	0	0	0	0
ma00967	0	115	0	0	0	0	0	0
ma02771	0	0	711	0	0	0	0	0
ma04614	2	0	0	225	0	0	0	0
ma07487	1	0	0	0	0	6	225	0
ma07488	0	0	0	0	0	124	11	0
ma07718	0	0	0	0	0	0	2	0
ma07721	0	0	0	0	0	6	1012	0

took the pre-trained one provided by Seuret *et al.*⁶ [Se19], and replaced its last layer by a new one with eight outputs. As this network has been trained for font groups classification, we expect it to have learned features useful for type classification. Note that we train only the last layer, leaving the other ones frozen. In earlier tests, which we conducted, we realized that fine-tuning the whole CNN makes it over-fit quickly.

The training is done as follows: First, we create a training set consisting of 5000 patches of 300×300 pixels for each type, uniformly distributed over the available training images. Then, during the training of the CNN, we apply data augmentation on these patches and extract 224×224 pixels center crops. We used the following data augmentation strategies: random rotations between]-15, 15[degrees, shearing of an angle between]-3, 3[, re-scaling by a factor in the range of]0.9, 1.1[, color jittering (PyTorch settings: 0.7, 0.7, 0.3, 0.03). Additionally, we add JPG artifacts with quality factors in the range of [2, 100[, and a low binarization probability with Otsu (5 %) or Sauvola (2.5 %).

The network is trained for 10 epochs, with an initial learning rate of 0.0005, a weight decay of 0.00001, and a momentum of 0.9. After each epoch, the learning rate is decreased by 5 %. Due to the small amount of images for some types, no validation set is used.

4.2 Results

For the evaluation, we split the dataset into a training and a test set. The training set consists of 6472 samples while the test set contains 2600 document images. We made sure that there is no document overlap in the subsets to guarantee document-independent testing.

A confusion matrix presenting the classification results is shown in Tab. 1. We can see that the system reaches an overall classification accuracy of 51.4 % and an average accuracy of

⁶ https://github.com/seuretm/ocrd_typegroups_classifier

73.9%, which already indicates that some classes are well recognized in contrast to others. The overall accuracy is significantly lower than the accuracy obtained on the training data (over 80% of the patches), which implies that we are running into over-fitting despite the rather aggressive augmentation approach.⁷

While the classification of the easy types was successful, several documents of the difficult types could not be detected properly. What makes the training for type ma07721 especially difficult is the fact that there is only one single training sample available, even when cropping a sufficient number of patches, the script variance might be too low for a reliable training. This suggests the use of a retrieval scenarios where the learning of a good embedding is in focus.

5 Retrieval

In addition to classification, we test the retrieval scenario. That means, we want to retrieve the most similar types given a query image.

5.1 Methodology

We make use of the general writer identification framework by Christlein [Ch18]. It consists of a sampling step, where we evaluate two strategies: SIFT keypoints [Lo04] computed at (a) the original images, (b) contours extracted by means of the well-known Canny edge detector [Ca86]. For the latter approach, we set the two hysteresis thresholds automatically [WMB15].

Afterwards, SIFT descriptors [Lo04] are computed at the keypoint locations. They are Dirichlet-normalized and PCA-whitened following Christlein [Ch18]. Afterwards, the local descriptors are encoded using VLAD [Jé12] encoding using 100 clusters for the codebook. For improving the VLAD embedding [CM18], we employ Generalized Max-Pooling (GMP) [Mu16] with $\lambda = 1000$ in combination with power normalization (power of 0.5) and ℓ^2 -normalization, i. e. normalizing the global descriptor such that its norm equals one. This process is repeated five times, the resulting global descriptors are concatenated and jointly PCA-whitened and dimensionality reduced to 6400 components and ℓ^2 -normalized again.

Finally, an Exemplar-SVM (ESVM) is computed, which has shown to improve the writer identification results [Ch17a]. We use the ESVM as a feature transformation [Ch17b], i. e. the ℓ^2 -normalized coefficients of the ESVM are used as new feature descriptor. These descriptors are then compared using the cosine distance, which equals a dot product of the ℓ^2 -normalized descriptors.

⁷ Note that a major constraint is that the augmentation should not make a type look like another one.

Tab. 2: Retrieval results for (a) easy testset using the difficult for training and (b) difficult testset using the easy one for training. The first row denotes a document-dependent scenario, where all samples of the test set are used for each query sample. For the other experiments, retrieved samples from the same document are ignored.

		(a) Easy			(b) Difficult		
	Sampling	Top-1	Top-10	mAP	Top-1	Top-10	mAP
1 vsAll	Keypoint+SIFT	98.9	99.9	62.7	99.8	99.9	96.3
1 vsOtherDocs	Keypoint+SIFT	88.9	94.2	54.6	50.0	50.2	46.6
	Keypoint+SIFT+ESVM	93.3	95.3	56.8	50.0	50.3	47.1
	Contour+SIFT	72.8	80.8	47.1	49.9	65.4	57.3
	Contour+SIFT+ESVM	76.5	80.2	49.2	49.9	65.4	57.3

5.2 Results

For the evaluation, we split the dataset into a type-independent training and test set. For simplicity, we choose all images of the easy types (#samples: 7029) as one subset and all the images of the difficult types (#samples: 2043) as the other subset. We then evaluate the following two configurations: (1) trained with the difficult subset and tested with the easy one and (2) the other way around, i. e. trained with the easy subset and tested with the difficult one. We made sure that there is no type overlap in the subsets to guarantee type independent testing. Note that this is different from the classification scenario, where we know the classes in advance.

We report typical retrieval measures, such as Top-1 accuracy as well as mean average precision (mAP), which is a measure of the overall ranking of the relevant documents in respect to the query sample. Additionally, we give the Top-10 accuracy, i. e. the chance of finding at least one sample of the query type among the first ten ranked results.

First, we compute the typical leave-one-image-out scenario, i. e. every test sample is used as query and *all* remaining ones are ranked according to their similarity with the query. Tab. 2 (first row) shows that this works astonishingly well with rates beyond 99 %. This is only natural, since each query sample comes from a specific document, and the remaining document images are among the other samples of the test set. In other words, the algorithm most probably retrieves images from the same document.

For the remaining experiments, we evaluate the retrieval performance in a document-independent way. Therefore, we ignore images from the same document during the metric computation. This results in a drop in performance but is a much more realistic scenario. When comparing the different sampling methods (keypoint vs. contour), we see that keypoint-based sampling is superior to contour-based sampling when trained on the difficult types samples and tested on the easy ones, see Tab. 2a. The reverse behavior is shown for the difficult cases where contour-based sampling is in favor, cf. Tab. 2b. This might be

related to the quite imbalanced test document sizes; one of the difficult types has only one image in one document, thus the metrics are biased when retrieving this specific one. This might also be the reason for the bad results for the difficult subset. ESVMs show also to be beneficial for type retrieval, at least for the easy cases.

6 Conclusion

In this work, we analysed the possibilities of type recognition in the two scenarios classification and retrieval. Therefore, we adopted and evaluated two baseline systems, originally developed for font-group classification and writer identification. While the classification network achieves an overall low accuracy, this can be attributed to the class imbalance of the training set. In the case of type retrieval, a careful document-independent evaluation reveals that very similar-looking types are problematic for a common retrieval pipeline. For future work, we would like to investigate networks, trained by deep metric-learning methods, e. g. with the use of contrastive or triplet loss, which might enable to differentiate also very similar-looking types.

References

- [Am79] Amelung, P.: *Der Frühdruck im deutschen Südwesten. 1473–1500. Bd. 1*, Ulm. Hiersemann, Stuttgart, 1979, ISBN: 3777279293.
- [Ca86] Canny, J.: *A Computational Approach to Edge Detection*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8/6, pp. 679–698, Nov. 1986.
- [Ch17a] Christlein, V.; Bernecker, D.; Hönig, F.; Maier, A.; Angelopoulou, E.: *Writer Identification Using GMM Supervectors and Exemplar-SVMs*. *Pattern Recognition* 63/, pp. 258–267, 2017.
- [Ch17b] Christlein, V.; Gropp, M.; Fiel, S.; Maier, A.: *Unsupervised feature learning for writer identification and writer retrieval*. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1, IEEE, Kyoto, pp. 991–997, Nov. 2017.
- [Ch18] Christlein, V.: *Handwriting Analysis with Focus on Writer Identification and Writer Retrieval*, PhD thesis, FAU Erlangen-Nürnberg, 2018.
- [Ch19a] Christlein, V.; Nicolaou, A.; Seuret, M.; Stutzmann, D.; Maier, A.: *ICDAR 2019 Competition on Image Retrieval for Historical Handwritten Documents*. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. Pp. 1505–1509, 2019.
- [Ch19b] Christlein, V.; Spranger, L.; Seuret, M.; Nicolaou, A.; Král, P.; Maier, A.: *Deep Generalized Max Pooling*. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. Pp. 1090–1096, Sept. 2019.

- [Cl16] Cloppet, F.; Eglin, V.; Kieu, V. C.; Stutzmann, D.; Vincent, N.: ICFHR2016 Competition on the Classification of Medieval Handwritings in Latin Script. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). Pp. 590–595, 2016.
- [Cl17] Cloppet, F.; Eglin, V.; Helias-Baron, M.; Kieu, C.; Vincent, N.; Stutzmann, D.: ICDAR2017 Competition on the Classification of Medieval Handwritings in Latin Script. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Kyoto, pp. 1371–1376, Nov. 2017.
- [CM18] Christlein, V.; Maier, A.: Encoding CNN Activations for Writer Recognition. In: 13th IAPR International Workshop on Document Analysis Systems. Vienna, pp. 169–174, Apr. 2018.
- [Fi17] Fiel, S.; Kleber, F.; Diem, M.; Christlein, V.; Louloudis, G.; Stamatopoulos, N.; Gatos, B.: ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 01, Kyoto, pp. 1377–1382, Nov. 2017.
- [Ge5–] für den Gesamtkatalog der Wiegendrucke, K., ed.: Gesamtkatalog der Wiegendrucke. Hiersemann [u.a.], Leipzig; [later:] Stuttgart, 1925–.
- [Ha05] Haebler, K.: Einführung. In (Haebler, K., ed.): Typenrepertorium der Wiegendrucke. Abt. I. Deutschland und seine Nachbarländer. Haupt, pp. IX–XXVIII, 1905.
- [Hu17] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Pp. 4700–4708, 2017.
- [Jé12] Jégou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Pérez, P.; Schmid, C.: Aggregating Local Image Descriptors into Compact Codes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34/9, pp. 1704–1716, Sept. 2012.
- [Lo04] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60/2, pp. 91–110, Nov. 2004.
- [LZJ20] Lai, S.; Zhu, Y.; Jin, L.: Encoding Pathlet and SIFT Features With Bagged VLAD for Historical Writer Identification. IEEE Transactions on Information Forensics and Security 15/, pp. 3553–3566, 2020.
- [Mu16] Murray, N.; Jegou, H.; Perronnin, F.; Zisserman, A.: Interferences in Match Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 39/9, pp. 1797–1810, Oct. 2016, arXiv: 1611.08194.
- [Sc18] Schmitz, W.: Grundriss der Inkunabelkunde: das gedruckte Buch im Zeitalter des Medienwechsels. Hiersemann, Stuttgart, 2018, ISBN: 9783777218007.

- [Se19] Seuret, M.; Limbach, S.; Weichselbaumer, N.; Maier, A.; Christlein, V.: Dataset of Pages from Early Printed Books with Multiple Font Groups. In: Proceedings of the 5th International Workshop on Historical Document Imaging and Processing. Association for Computing Machinery, Sydney, pp. 1–6, 2019.
- [Wa15] Wang, Z.; Yang, J.; Jin, H.; Shechtman, E.; Agarwala, A.; Brandt, J.; Huang, T. S.: DeepFont: Identify Your Font from An Image. In: Proceedings of the 23rd ACM International Conference on Multimedia. MM '15, Association for Computing Machinery, Brisbane, Australia, pp. 451–459, 2015, ISBN: 9781450334594.
- [We20] Weichselbaumer, N.; Seuret, M.; Limbach, S.; Hinrichsen, L.; Christlein, V.: The rapid rise of Fraktur. In (Schöch, C., ed.): DHd 2020, Spielräume, Paderborn Mar. 2, 2020–Mar. 6, 2019. Zenodo, pp. 229–232, 2020.
- [WMB15] Wahlberg, F.; Mårtensson, L.; Brun, A.: Large Scale Style Based Dating of Medieval Manuscripts. In: 3rd International Workshop on Historical Document Imaging and Processing (HIP'15). ACM, Nancy, pp. 107–114, Aug. 2015.