HAL open science

Artificial Paleography: Computational Approaches to Identifying Script Types in Medieval Manuscripts

Mike Kestemont, Vincent Christlein, Dominique Stutzmann

To cite this version:

Mike Kestemont, Vincent Christlein, Dominique Stutzmann. Artificial Paleography: Computational Approaches to Identifying Script Types in Medieval Manuscripts. Speculum, 2017, 92 (S1), pp.S86-S109. 10.1086/694112 . hal-01854939

HAL Id: hal-01854939

https://hal.science/hal-01854939v1

Submitted on 7 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Artificial Paleography: Computational Approaches to Identifying Script Types in Medieval Manuscripts

By Mike Kestemont, Vincent Christlein, and Dominique Stutzmann

Introduction

Artificial intelligence (AI) is a vibrant research domain in which systems are developed that can reason and act like humans.[1] In recent years, the endeavor to reproduce human intelligence in software has led to the introduction of many well-known computer applications that are increasingly, and yet sometimes almost unnoticeably, becoming a part of our everyday lives. Representative examples include face recognition on social media, spam filters for email clients, or recommendation systems in online stores. It is to these highly practical applications that AI currently owes its high visibility—as well as its at times controversial status, as exemplified by the ethical debate sparked by the introduction of self-driving vehicles.[2]

Nevertheless, these highly useful, practical applications make it easy to forget that AI also addresses more theoretical issues. Being able to reproduce human intelligence, even if only for specific tasks, can help advance our understanding of the working of the human mind itself—as famous physicist Richard Feynman is credited with saying, "What I cannot create, I do not understand."[3] The humanities, which can be broadly defined as the study of the products of the human mind, in this respect seem a privileged partner for AI.[4] In the field of digital humanities, various forms of AI have played a role of increasing importance for a number of decades; now that computer technologies are maturing at a rapid pace, we expect to see the emergence of many more collaborations between the humanities and AI in the future.

Here we focus on paleography, the scholarly study of historical handwriting, which, apart from being a long-standing discipline in its own right, also remains a crucial auxiliary science in medieval studies for codicologists, literary scholars, and historians alike. Paleography is an interesting case for the application of AI. Whereas most medievalists have at least a superficial reading competency for

[1] For a general introduction consult the classical textbook Stuart J. Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach* (Upper Saddle River, 1995), with definitions on 4–5.

[2] A representative survey is offered in a recent thematic issue of *Science*, including, for example, Julia Hirschberg and Christopher D. Manning, "Advances in Natural Language Processing," *Science* 349, no. 6245 (2015): 261–66.

[3] At the time of his death, this famous sentence was written on a blackboard, of which a photograph is kept in the Caltech Archives: http://archives.caltech.edu/pictures/1.10-29.jpg.

[4] Rens Bod, *A New History of the Humanities: The Search for Principles and Patterns from Antiquity to the Present* (Oxford, 2013), 9.

Speculum 92/S1 (October 2017). © 2017 by the Medieval Academy of America. All rights reserved.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0), which permits non-commercial reuse of the work with attribution.
For commercial use, contact journalpermissions@press.uchicago.edu.
DOI: 10.1086/694112, 0038-7134/2017/92S1-0004$10.00.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

common script forms, experienced paleographers are typically still required to solve more complex tasks, such as dating, localizing, or authenticating specific scripts. Thus, the field of paleography is dominated by expert-based approaches and driven by the opinions of small groups of highly trained individuals.

The problems with the expert-driven nature of paleographic methods have long been acknowledged. Dating and authenticating scribal hands are classic examples of a difficult problem that is typically tackled using methods that have been either justified as corresponding to the subtle nature of human individual and artistic production, or criticized for being too ad hoc, (inter)subjective and difficult to replicate or evaluate.[5] Paleographic skills can often be acquired only through intensive training and prolonged exposure to rare artifacts that can be difficult to access. Much like expert-based methods in the field of art authentication, paleographic knowledge can be difficult to formalize, share, and evaluate. Therefore, paleographers are increasingly interested in digital techniques to support and enhance the traditional practice in the field.[6] Additionally, computer-assisted methodologies for paleographers are now more urgently needed than ever, given the fact that digital libraries such as Gallica, Manuscripta Mediaevalia, BVMM, and (more recently) the Vatican's DigiVatLib are amassing electronic reproductions of medieval manuscripts, sometimes with scarce metadata that are incomplete or out of date.[7] Hiring and training experts to manually enrich or correct the metadata for these collections is expensive and time-consuming. Therefore, there is a strong demand among all stakeholders in the field for automated, computer-assisted techniques to assist scholars in their work.

In this paper we report the results of a recent research initiative that targeted the automated identification of script types in (photographic reproductions of) medieval manuscript folios. In AI it is commonly said that a defining characteristic of human intelligence is the ability to *learn*, that is, that an individual can optimize his

[5] After the seminal work of Léon Gilissen, *L'expertise des écritures médiévales: Recherche d'une méthode avec application à un manuscrit du XIe siècle; le lectionnaire de Lobbes (Codex Bruxellensis 18018)* (Gand, 1973), the debate was formalized on the opposition made by B. Bischoff between the "Kunst des Sehens und Einfühlung" and the "Kunst des Messens" under the title "Commentare Bischoff" in the journal *Scrittura e Civiltà* 19 (1995): 325–48; 20 (1996): 401–7; 22 (1998): 397–418. The contributors were Giorgio Costamagna, Françoise Gasparri, Léon Gilissen, Francisco M. Gimeno Blay, J. Peter Gumbert, Armando Petrucci, and Alessandro Pratesi. The arguments were summarized in Albert Derolez, *The Palaeography of Gothic Manuscript Books from the Twelfth to the Early Sixteenth Century* (Cambridge, UK, 2003), opening the way for a field of research in its own right and reflections on the necessity of cross-disciplinary research.

[6] The developments in different directions are illustrated in Malte Rehbein, Patrick Sahle, and Torsten Schaßan, *Codicology and Palaeography in the Digital Age* (Norderstedt, 2009); Franz Fischer, Christiane Fritze, and Georg Vogeler, *Codicology and Palaeography in the Digital Age 2* (Norderstedt, 2011); Oliver Duntze, Torsten Schaßan, and Georg Vogeler, *Codicology and Palaeography in the Digital Age 3* (Norderstedt, 2015). A recent trend is to evaluate the potential and methodology for an adequate implementation of digital techniques in paleography: see Tal Hassner, Malte Rehbein, Peter A. Stokes, and Lior Wolf, "Computation and Palaeography: Potentials and Limits," *Dagstuhl Manifestos* 2 (2013): 14–35; Tal Hassner, Robert Sablatnig, Dominique Stutzmann, and Ségolène Tarte, "Digital Palaeography: New Machines and Old Texts," *Dagstuhl Reports* 4/7 (2014): 112–34.

[7] This list is representative, but of course incomplete. The online portals for the mentioned libraries can be accessed from the following URLs: Gallica (http://gallica.bnf.fr); Manuscripta Mediaevalia (http://www.manuscripta-mediaevalia.de/); BVMM—Bibliothèque virtuelle des manuscrits médiévaux (http://bvmm.irht.cnrs.fr); and DVL DigiVatLib (http://digi.vatlib.it/mss/).

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

or her behavior on the basis of previous experience, anticipating future reward. This facility is nowadays studied in the domain of machine learning, an important subfield of AI. Here, we aim to verify the challenging hypothesis that it should be possible to teach a software system to identify and classify medieval scripts on the basis of representative examples, much like any freshman student, with no previous experience in paleography, would learn to distinguish a conventional Gothic book letter (*littera textualis formata*) from a more cursive handwriting (*littera cursiva currens*). Apart from a rigorous empirical evaluation of our results, we aim to demonstrate how the interaction between traditional models from paleography and computational ones during this project also raised valuable interpretative issues, as well as conflicts.

The CLaMM Competition

This paper centers around a recently organized competition at the Fifteenth International Conference on Frontiers in Handwriting Recognition. In the field of machine learning, competitions (or "shared tasks") are a common format to attempt to break new ground in a particular area. Typically, the organizers of a competition release a so-called training data set, containing a representative set of digital items (images, texts, sound fragments, etc.) that have been manually annotated with ground-truth "class labels" (for example, the topic of a text or the item depicted in a photograph). Teams can then register for the competition and develop a software system that can learn how the items under scrutiny should be classified. Finally, the teams submit their model to the organizers, who run it on a new data set of previously unseen test items. This test data allows the organizers to evaluate and compare the submissions.

Such competitions are an attractive scientific format because they force different teams to evaluate their software on identical data sets, which are generally also open to the general public. Many participants will also share their solutions under a liberal license in online repositories, which will stimulate further research and facilitate testing improvements to existing solutions. The shared task under scrutiny here was named "CLaMM: Competition on the Classification of Medieval Handwritings in Latin Script."[8] The organizers released a training data set of two thousand grayscale images in an uncompressed image format (TIFF, 300 DPI). Each image featured a photographic reproduction of an approximately 100 × 150 mm part of a (distinct) medieval Latin manuscript. The selection drew heavily on the well-known repertory of *Manuscrits datés*, containing manuscripts that can be dated to the period 500–1600 AD, complemented with other sources.[9] Each training image

[8] The results and organization of this competition are described on the competition's website (https://oriflamms.hypotheses.org/1388) and in Florence Cloppet, Véronique Eglin, Van Kieu, Dominique Stutzmann, and Nicole Vincent, "ICFHR2016 Competition on the Classification of Medieval Handwritings in Latin Script," in *Proceedings of the International Conference in Frontiers on Handwriting Recognition* (Shenzhen, China, 2016), 590–95.

[9] Charles Samaran and Robert Marichal, *Catalogue des manuscrits en écriture latine portant des indications de date, de lieu ou de copiste*, 14 vols. (Paris, 1959–84); Denis Muzerelle, *Manuscrits datés des bibliothèques publiques de France*, 2 vols. (Paris, 2000–2013).

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

was classified into one of twelve common script types, ranging from early medieval uncial and Carolingian script types to late medieval Gothic book letters and humanistic scripts (see Fig. 1 below). A consensus about defining any number of different classes is currently beyond reach within the paleographic community and represents an ill-posed problem, so that, in regard to artificial intelligence, we first have to test, extensively and systematically, one coherent classification, based on formal criteria only.[10] For this competition, classes were characterized using standard definitions for uncial, semiuncial, Caroline, humanistic and humanistic cursive,[11] and the main script types of Derolez' classification for Gothic scripts (Praegothica, Textualis, Semitextualis, Southern Textualis, Hybrida, Cursiva, Semihybrida).[12] On the basis of the two thousand training images, participants had to train a classification system that was able to provide predictions for new, previously unseen images.

The CLaMM competition can be situated in the domain of computer vision, a popular branch in present-day AI and machine learning.[13] In this multidisciplinary field, algorithms are developed that mimic the perceptual abilities of humans and their capacity to construct high-level interpretations from raw visual stimuli. Face identification on social media or autonomous driving are probably its best-known applications nowadays. In the digital humanities (DH) it is a well-known fact that most of the seminal research, beginning with Busa's acclaimed *Index Thomisticus*, has been heavily text oriented.[14] At a lower level (for example, simple search), text is generally easier to process than images, especially because plain text corpora typically come with much more limited memory requirements than high-resolution image collections. In recent work in DH, image analysis has started to attract more attention. Optical character recognition (OCR), the process of extracting machine-readable text from scans of printed works, has arguably been one of the most prominent applications.

While OCR is today sometimes (mistakenly) considered a solved problem in computer vision, *handwritten* text recognition (HTR) still presents an open challenge for many languages and document types.[15] Conventional OCR applications still have huge difficulties in processing continuous script forms and their ligatures. Even simple layout analysis (for example, recognizing columns and text-line detection) presents major impediments.[16] This is especially true for historical samples of

[10] Dominique Stutzmann, "Clustering of Medieval Scripts through Computer Image Analysis: Towards an Evaluation Protocol," *Digital Medievalist* 10 (2015), https://journal.digitalmedievalist.org/articles/10.16995/dm.61/.

[11] Bernhard Bischoff, *Paläographie des römischen Altertums und des abendländischen Mittelalters* (Berlin, 1986).

[12] Albert Derolez, *The Palaeography of Gothic Manuscript Books*.

[13] An often-cited general-purpose introduction to the field is Richard Szeliski, *Computer Vision: Algorithms and Applications* (New York, 2010).

[14] See, e.g., Susan Hockey, "The History of Humanities Computing," in *A Companion to Digital Humanities*, ed. Susan Schreibman, Ray Siemens, and John Unsworth (Oxford, 2004), http://www.digitalhumanities.org/companion/. Consult the introduction to the supplement on the *Index Thomisticus*.

[15] Alex Graves, *Supervised Sequence Labelling with Recurrent Neural Networks* (New York, 2012).

[16] Laurence Likforman-Sulem, Abderrazak Zahour, and Bruno Taconet, "Text Line Segmentation of Historical Documents: A Survey," *International Journal on Document Analysis and Recognition* 9 (2007): 123–38. Also see F. Simistira et al., "DIVA-HisDB: A Precisely Annotated Large Dataset of Challenging Medieval Manuscripts," in *International Conference on Frontiers in Handwriting Recognition* (Shenzhen, 2016), 471–76.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

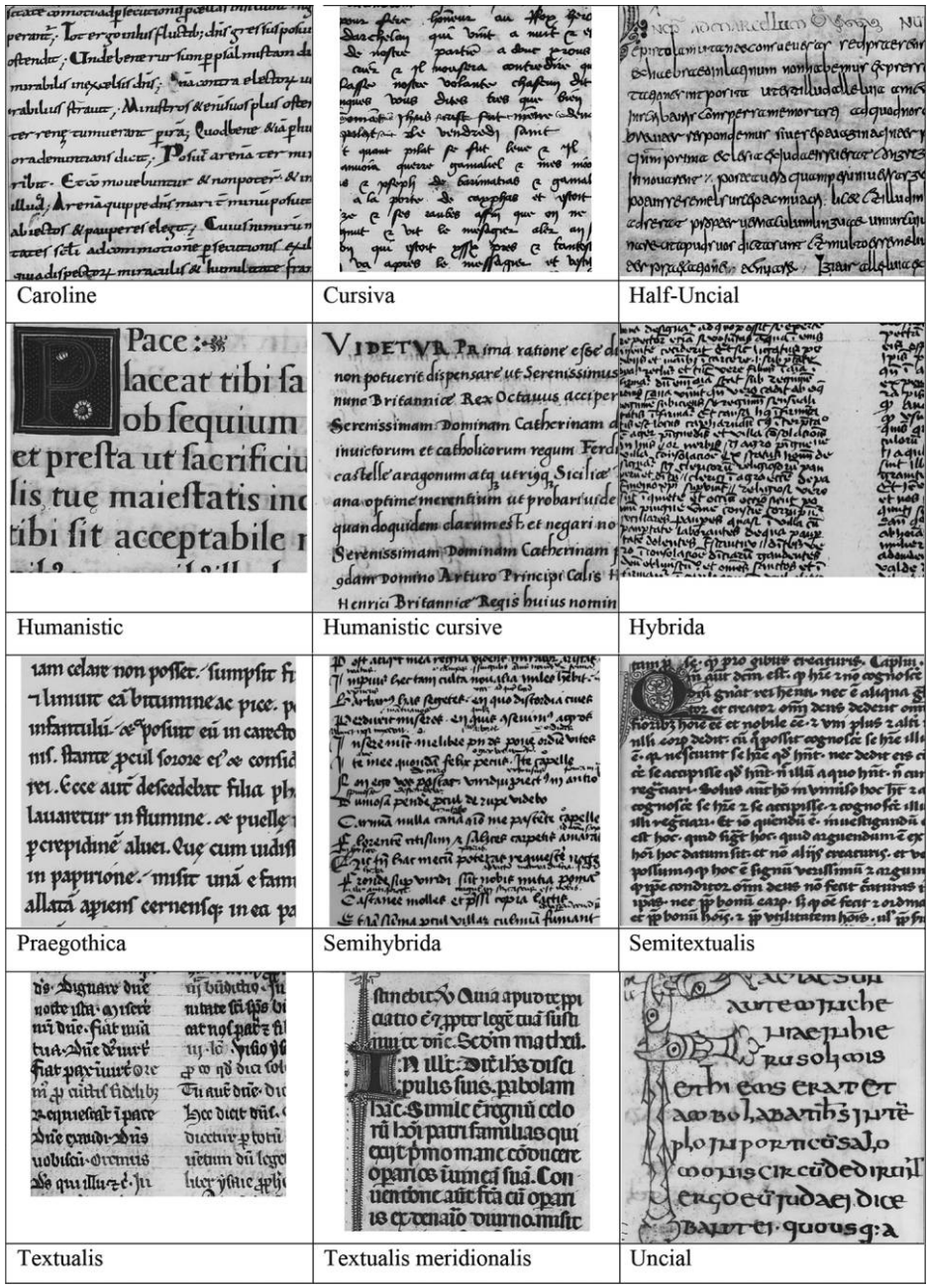


Fig. 1. Examples of the twelve script classes contained in the data set. Caroline (Autun, Bibliothèque municipale, MS 22, fol. 154r); Cursiva (Autun, Bibliothèque municipale, MS 206, fol. 37r); Half-Uncial (Epinal, Bibliothèque municipale, MS 68 fol. 12r); Humanistic (Avignon, Bibliothèque municipale, MS 172, fol. 19r); Humanistic cursive (Besançon, Bibliothèque municipale, MS 389, fol. 1r); Hybrida (Autun, Bibliothèque municipale, MS 50, fol. 132r); Praegothica (Auch, Bibliothèque municipale, MS 1, fol. 24r); semihybrida (Auxerre, Bibliothèque municipale, MS 84, fol. 116r); semitextualis (Auch, Bibliothèque municipale, MS 6, fol. 49r); textualis (Autun, Bibliothèque municipale, MS 8, fol. 10v); textualis meridionalis (Avignon, Bibliothèque municipale, MS 138, fol. 36r); uncial (Autun, Bibliothèque municipale, MS 3, fol. 175r).

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

handwriting, where algorithms must cope with much higher levels of individual variation among writers than in the case of typeset fonts. Script classification is an extremely relevant preprocessing step in this respect: in order to be able to machine-read a medieval manuscript, it goes without saying that an indication of the script type used in it provides crucial information for selecting the best HTR engine. Script-type classification is also related to other historical applications of computer vision. Writer identification, for instance, is a topic that has been explored with encouraging results for medieval authors such as Chaucer and many other historical data sets.[17]

Methods

In this paper, we introduce two complementary methods that have been submitted to the CLaMM competition, each of which ranked first in one of the competition's tasks. One, the DeepScript approach, relies on the use of deep convolutional neural networks, which recently attracted much interest in the computer vision community; and the other one, the FAU submission, uses a more established computer-vision approach, which is known as "Bag of (visual) Words." In this section, we introduce both methods in nontechnical language that should be accessible to the broad readership of the journal.

Bag of Words Model

The Bag of Words model (BoW) is a representation strategy that was originally borrowed from parallel research into automated text classification. Modern spam filters in e-mail clients are a textbook example of applications in machine learning that rely on BoW models.[18] To determine whether an incoming email should be moved to the junk folder, algorithms are trained on large sets of example messages, which have been flagged by moderators as "spam" or "not spam." These methods typically assume that the document-level frequencies of sensitive words, such as "lottery," suffice to solve this classification task. The exact order or position of the words in an e-mail is largely considered irrelevant in many spam filters. Thus the algorithms consider e-mails as randomly jumbled "bags of words" in which only the frequencies of items matter, and not their order or position. In computer vision, three steps are involved in constructing a similar BoW strategy for images: first, we need to extract the local feature descriptors (that is, the visual "words") from the image. Second, these local descriptors have to be combined, or encoded; that is, the local feature descriptors need to be aggregated to form a global feature descriptor, or "supervector." Third, this global supervector has to be classified into one of the script-type classes.

[17] Marius Bulacu and Lambert Schomaker, "Automatic Handwriting Identification on Medieval Documents," in *Proceedings of 14th International Conference on Image Analysis and Processing* (Modena, 2007), 279–84. Also see Vincent Christlein et al., "Automatic Writer Identification in Historical Data: A Case Study," *Zeitschrift für digitale Geisteswissenschaften* (2016), doi:10.17175/2016_002.

[18] Fabrizio Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computer Surveys* 34 (2002): 1–47.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

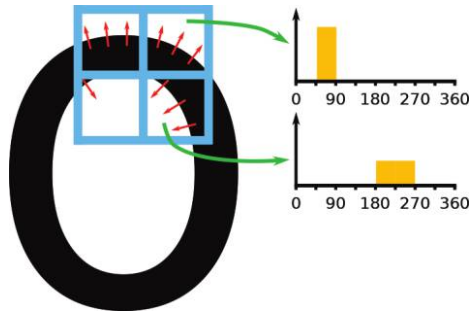


Fig. 2. SIFT computes the gradients at each pixel in a grid of N areas (here N = 2 × 2) around a keypoint (here the midpoint of the blue rectangle) and creates N histograms of the gradients' orientations around that keypoint. Two example histograms are depicted here. The orientation angles are divided into 8 bins (e.g., the top right area has one large bin for orientations between 45° and 90°).

In the FAU approach, scale-invariant feature transformation (SIFT) is used for the identification of local features.[19] This is a well-known approach in computer vision due to its robustness to image transformations, such as changes in the brightness and contrast, scale, or rotation of an image. SIFT depicts the orientation of what is called the gradient information or the directional change of the colors in a small region of the image (see Fig. 2). In homogeneous regions, with few changes, the gradients will be zero; otherwise the gradients capture the boundary between script and nonscript areas. These gradients are calculated around a "keypoint" so that the algorithm computes the distribution of the orientations of the gradients in a small image patch, that is, the directions in which the gradient points are collected. SIFT keypoints are points in the image that have stable gradients across several scales. Using histograms representing the gradient information around the keypoint, we can then compute the main orientation. This enables the descriptor to become rotationally invariant, meaning that the same descriptor would be computed also for rotated versions of the script. S. Fiel and R. Sablatnig, however, have demonstrated that disabling rotational invariance enhances the results in writer identification,[20] probably because the Latin script uses rotated or mirrored signs with different significations and stylistic features (as for *d*, *b*, *p*, *q* in their modern forms). Thus, this property was intentionally removed in this approach for a corpus without rotated scripts or vertical lines. Examples for SIFT keypoints are visualized in Fig. 3: these keypoints indicate areas in the images that seem of particular relevance to the model and function as the salient "words" in the BoW model. Note, for instance, how the flourishing of decorated initials invites the detection of many more keypoints than do the page's margins.

[19] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision* 60 (2004): 91–110.

[20] Stefan Fiel and Robert Sablatnig, "Writer Identification and Writer Retrieval Using the Fisher Vector on Visual Vocabularies," in *12th International Conference on Document Analysis and Recognition: ICDAR 2013* (Washington, DC, 2013), 545–49.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

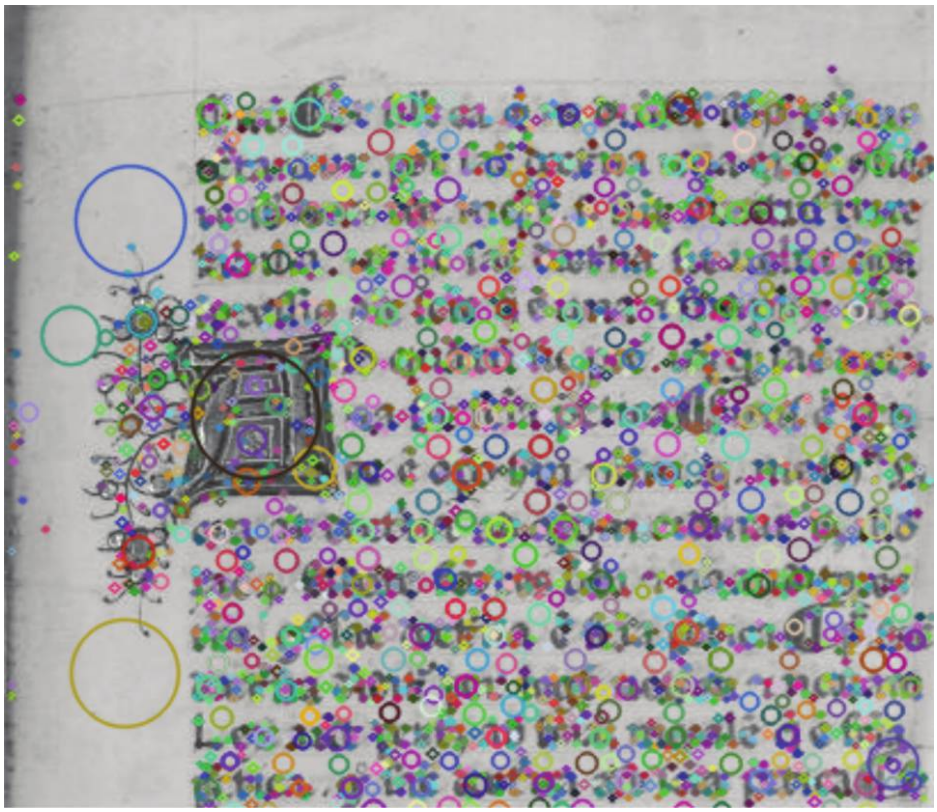


Fig. 3. Visualization of SIFT keypoints (Toulouse, Bibliothèque municipale, MS 214). The (randomly colored) circles visualize the local scope of the features. Note also that SIFT keypoints that lie between two lines may contain information about, for example, the typical line height and ascenders or descenders.

On the basis of these "visual words," we now have to create a global descriptor for the entire image. The simplest approach would be to take the average of all local descriptors, but more sophisticated methods have shown better performance in the past.[21] These encoding methods typically rely on a background model that needs to be computed from the training data in advance. When presenting an incoming image to the system, a global image descriptor is determined by aggregating statistics drawn from the background model of the local descriptors of this image. The background model is created by clustering a subset of the local descriptors of the training set, typically using established clustering techniques (see Fig. 4). One of the simpler encoding methods would be vector quantization: for each cluster center of the background model, the number of nearest descriptors is counted to create the global supervector (see Fig. 5).

[21] Ken Chatfield et al., "The Devil Is in the Details: An Evaluation of Recent Feature Encoding Methods," in *British Machine Vision Conference (BMVC)*, ed. Jesse Hoey, Stephen McKenna, and Emanuele Trucco (Dundee, UK, 2011), 2:8, no. 4.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

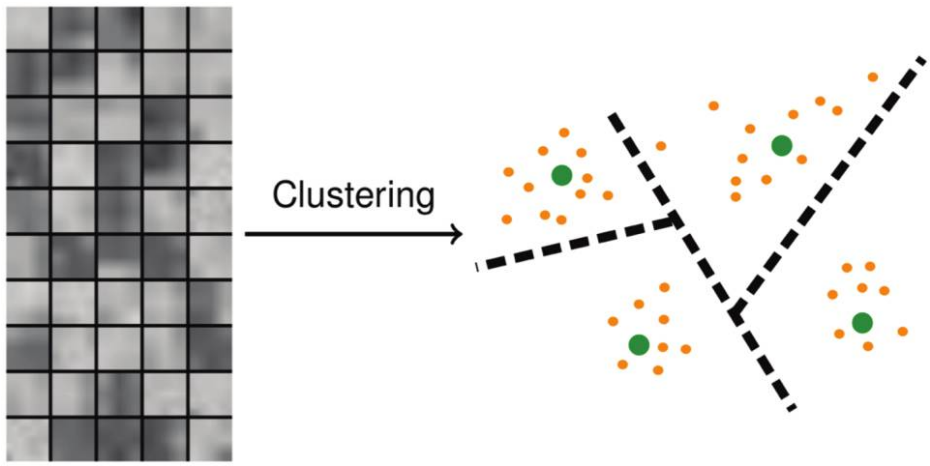


Fig. 4. Creation of the background model using clustering of local SIFT descriptors (the green dots represent the cluster centers).

We rely on a well-known technique for speaker verification and spoken language classification involving the use of what are known as i-vectors.[22] In order to overcome the variability within each script type and to allocate different documents written by different hands and in somewhat different styles into the same class, we used "within-class covariance normalization" (WCCN).[23] WCCN assigns more importance to the dimensions with higher between-classes variance, which means that the visual aspects that separate the script type are emphasized.

The last step of the approach lies in the classification of the global image descriptor. To this end, linear support-vector machines (SVM) are employed, a highly popular binary classifier in the field of machine learning.[24] Given training examples of two script categories, an SVM model is trained such that a decision boundary between the classes is fit, having a margin between the categories that is as wide as possible. For each script type, a separate SVM is trained using all the supervectors of this script type as (positive) training samples of one class and all others as (negative) training samples of the other classes. During evaluation, all SVMs are queried after a test image has been encoded and their output scores are ranked: eventually, the class that invited the highest score by one of the classifiers gets assigned to the input image.[25]

[22] Najim Dehak et al., "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing* 19/4 (2011): 788–98.

[23] Andrew O. Hatch, Sachin S. Kajarekar, and Andreas Stolcke, "Within-Class Covariance Normalization for SVM-Based Speaker Recognition," in *International Conference on Spoken Language Processing, Pittsburgh, Pennsylvania, USA, September 2006* (Pittsburgh, 2006).

[24] Corinna Cortes and Vladimir Vapnik, "Support-Vector Networks," *Machine Learning* 20 (1995): 273–97.

[25] For Task 2 of the competition, linear discriminant analysis (LDA) was used instead of SVM.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

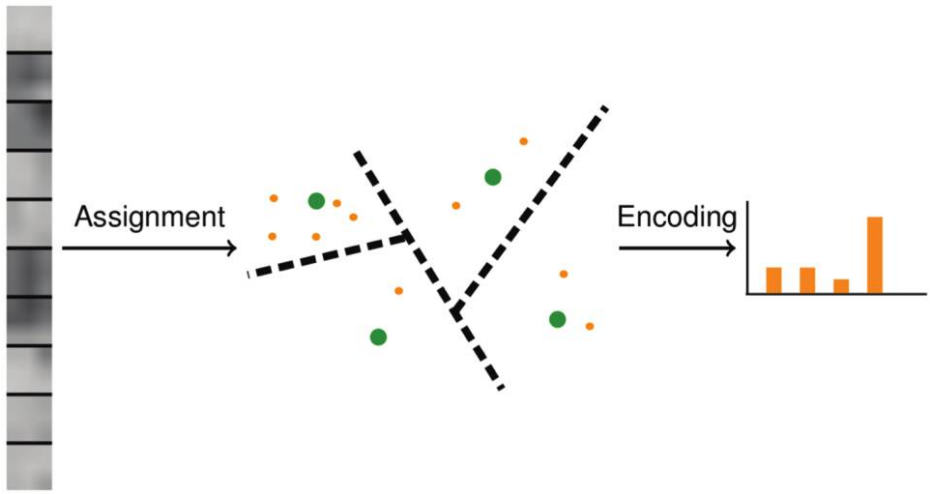


Fig. 5. Encoding of local descriptors to compute a global image descriptor by counting the nearest neighbors for each cluster center.

Deep-Learning-Based Classification

A major (re)innovation in artificial intelligence is so-called deep-representation learning.[26] In fact, many applications, such as speech recognition in mobile phones, autonomous driving, or handwritten text recognition, are already based on deep-learning techniques.[27] Deep learning typically relies on neural networks, an information processing model that consists of "neurons," or small information units, that are linked by weight connections.[28] The neurons in such networks are typically organized in layers that are stacked on top of one another. As Fig. 6 shows, neural networks typically have an input layer, which processes the raw information that goes into a model (e.g., a raster of pixel values that represent an image). The original information is constantly being processed and transformed as it is fed forward through the stack of layers in the network, until it reaches the output layer, where the final classification decision is made. The output layer in the DeepScript network consists of twelve neurons, one for each script class involved in the CLaMM competition. Images are categorized into one of the script types involved according to which output neuron receives the highest activation.

Their layered nature sets neural networks apart from other learning techniques that, conventionally, do not have all these intermediary stages between input and output. It has been noted that in these layers different levels of abstraction are captured. In the task of face recognition, for instance, where the system's task is to identify a specific individual, we see that very primitive features are being detected

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature* 521 (2015): 436–44.

[27] The code for applying the FAU system is available online, https://github.com/vchristlein/clamm-icfhr16.

[28] Yoshua Bengio, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013): 1798–828.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

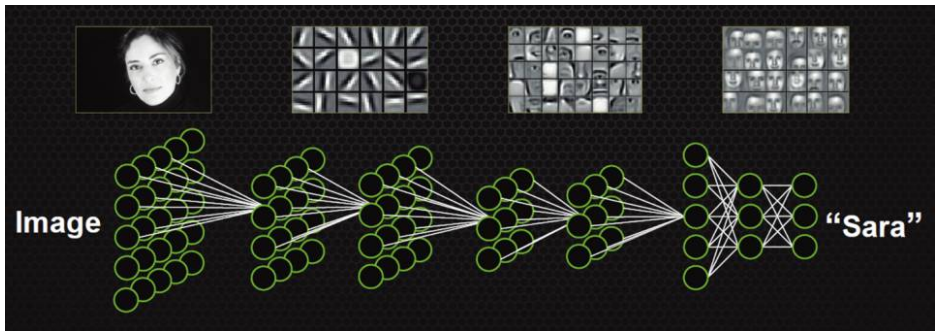


Fig. 6. Visualization of a neural network for face identification: the network consists of interconnected neurons that are organized in layers that are stacked on top of one another and that transform the raw, original input signal (e.g., an image) from the input layer (left) to the output layer (right), where the person in the image gets recognized as a specific individual ("Sara"). The information flows through a sequence of intermediate, "hidden" layers, which are sensitive to increasingly complex patterns or features.[29]

in the first layers, such as "edges" or stark local contrasts. Gradually, these primitive shapes get combined into more complex features at higher layers in the networks, which detect more abstract face parts, such as noses or ears. It is only in the highest layers that the network becomes sensitive to entire faces and is able to recognize specific individuals. This sort of machine learning is therefore often called representation learning or deep learning: apart from learning to solve a specific problem, the model also learns to extract features of an increasing complexity from images. At subsequent levels, deeper or more abstract features are being detected.

In image classification, deep "convolutional" neural networks (CNNs) have become a state-of-the-art tool for large-scale image classification. "Convolutional" means that such a network typically starts by sliding a series of low-level feature detectors over the entire image.[30] These detectors are first applied to small areas in the original image (for example, square patches of 3×3 pixels). The features detected by these low-level "filters" are subsequently fed into higher-level neurons, which thus have a larger receptive field (e.g., 27×27 pixels) in the sense that they "see" a larger part of the original image.

In comparison with the FAU-BoW model, one clear drawback of neural networks is that they are meant to work with large amounts of training data, typi-

[29] This image was taken from a blog by NVIDIA (https://devblogs.nvidia.com/parallelforall/accelerate-machine-learning-cudnn-deep-neural-network-library/), a computer-chip manufacturer that is actively involved in neural network research and applications. The weight visualizations in this image are borrowed from the work in Honglak Lee et al., "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, 2009), 609–16.

[30] A selection of seminal papers in this area includes Yann LeCun et al., "Handwritten Digit Recognition with a Back-Propagation Network," in *Proceedings of Advances in Neural Information Processing Systems* (San Francisco, 1990), 396–404; LeCun et al., "Gradient-Based Learning Applied to Document Recognition," in *Proceedings of the IEEE* 86 (1998): 2278–324; Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, "ImageNet Classification with Deep Convolutional Networks," *Advances in Neural Information Processing Systems* 25 (2012): 1090–98.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

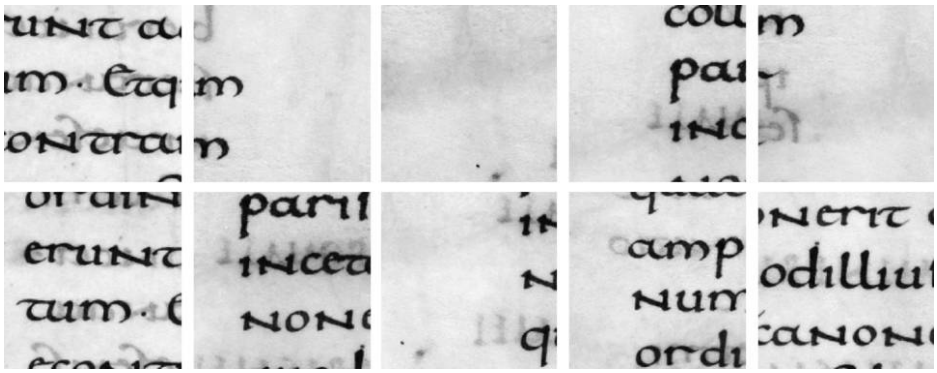


Fig. 7. An example series of random crops (150 × 150 pixels) from the original manuscript image from Paris, Bibliothèque nationale de France, MS lat. 266 (after downscaling the original resolution by a factor of two).

cally in the range of hundreds of thousands of images. The data set of the CLaMM competition, which was already difficult to create in the first place, is rather small from this perspective. Moreover, because of their powerful modeling capacities, the danger exists that networks naively start "memorizing" the training examples: this will result in the undesirable situation that the network produces perfect predictions for the training data—simply because it has learned to memorize the example images—in a manner that does not generalize or scale well to new images that have to be classified. This learning artifact is commonly known as "overfitting."

To combat overfitting, the DeepScript approach proceeded as follows.[31] First, the resolution of the original training images was downsized by a factor of two. During training, we would iteratively extract a random series of rectangular crops or patches from these images, measuring 150 × 150 pixels (Fig. 6). We would train our system on such smaller 150 × 150 patches instead of on the full images—the size of which was generally too generous to be processed by standard neural networks anyway. For new incoming images, we would select thirty such random patches from the image and average our predictions for these individual patches to obtain an aggregated prediction for the entire image. Interestingly, these crops were selected from the images in a fully random fashion and no explicit attempts were undertaken to identify more specific "regions of interest" in an image, such as columns, text lines, or words. Although such a random harvesting procedure would frequently yield useless patches (for example, taken from a folio's margins: see Fig. 7), the idea is that we would still be able to collect enough relevant information from a manuscript page, provided enough sample patches were drawn from it.

[31] Our implementation (and a trained model) are publicly available: see https://github.com/mikekestemont/DeepScript. Our code depends on the popular Keras library (https://keras.io/), which serves as an interface to Theano: see Rami Al-Rfou et al. (the Theano Development Team), "Theano: A Python Framework for Fast Computation of Mathematical Expressions," *ArXiv* 2016, http://arxiv.org/abs/1605.02688. We gratefully acknowledge the support of NVIDIA Corporation for the donation of the TITAN X used for this research. A special word of thanks goes out to Dr. Sander Dieleman (Google DeepMind, London) for his valuable feedback and help in the development process.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

Fig. 8. A set of examples of randomly augmented patches for a single training image (Autun, Bibliothèque municipale, MS 124): noise was injected in the training patches through the introduction of small, random changes in the rotation, zoom level, and shearing of the original crops.

Additionally, to discourage the network from simply memorizing such patches, we used an "augmentation" procedure. Before feeding a patch to the network, we would randomly distort the image through the introduction of small changes in the rotation, zooming level, and shearing of the patches. An example of such random perturbations for a single folio is offered in Fig. 8. The underlying hypothesis is that the introduction of such artificial noise is too small to be completely detrimental to the classification model, but large enough to make it more difficult for the network to memorize the training instances. When classifying new patches, no augmentation would be applied to them. In the past, augmentation approaches have yielded impressive results in other competitions in computer vision where only limited training data was available.[32] Note that we did not mirror flip the input image, as is commonly done in other vision tasks, because Latin scripts mostly run from left to right.

Comparison

It is interesting to compare the FAU and DeepScript systems. Both systems share the characteristic that they start from local-feature descriptions in the image, which are subsequently aggregated in a more complete representation of the full image. This attention to low-level information reflects the fact that the manuscripts have been categorized by the annotators based on local, morphological features, such as Derolez', which are exclusively situated at the level of individual characters, instead of, for example, page-level layout information. The exact manner in which

[32] Our augmentation derives from that of Sander Dieleman, described here: http://benanne.github.io/2015/03/17/plankton.html.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

visual features are detected is nevertheless clearly different. FAU extracts SIFT keypoints using an established generic feature detector, which is known to work well across many problems in computer vision but which cannot be fine-tuned in the light of a specific data set. In other words, the keypoint detection algorithm is fixed and does not get adapted to the particularities of the script-type classification task. DeepScript's neural network approach is, in principle, able to learn more task-specific filters, but here the limited size of the training data might pose problems for the feasibility of this approach.

Note that both approaches explicitly try to reach scaling invariance, which is an important quality of a computer vision system: the detection of a given script type should not break down for scribes who wrote relatively larger or smaller letters, or for manuscripts that were photographed at a different zoom level. Whereas the visual recognition of objects across different scales is typically easy for humans, this is difficult for computers. Most other submissions to the CLaMM competition can be likened to FAU or DeepScript—two other submissions, for instance, also used variations of convolutional neural networks. Thus, while competing methods exist, the two approaches discussed in this paper give a representative idea of the sorts of approaches that are used in the field.

Results

Evaluation

The competition had two separate evaluation tracks, and teams could sign up for both or just one. For Task 1 ("Crisp classification"), the evaluation procedure involved a data set of one thousand images that had been classified into one of the twelve script classes involved. The participants had to provide (1) a "hard" classification for each image (that is, the most likely script type according to their algorithm) and (2) a square matrix of distances containing scores (between 0 and 1) that indicated how dissimilar each test image was from each other test image. With respect to (1), the submitted systems were simply ranked according to their average prediction accuracy; for (2), the systems were ranked according to a metric called "average intraclass distances" (AID). Naturally, the latter metric was intended to verify the hypothesis that a strong classification system would assign relatively lower distance scores to image pairs that belonged to the same script type.

Task 2 ("Fuzzy classification") was a more complex evaluation track, which tried to account for the historical reality that many medieval manuscripts contain a mix of multiple script types, with titles and rubrics, for instance, belonging to a clearly different script type than the main text. For Task 2, the submissions were therefore evaluated against a test data set of two thousand images in which two script types could be discerned (see the example in Fig. 9). Consequently, the submitted systems had to output the two most likely classification labels for the test images in this track. To evaluate the results, the organizers adopted an ad hoc scoring mechanism. Systems got +4 points if both predicted labels matched the ground truth (maximal score), +2 in case only the first label matched one of ground truth labels, +1 if only the second label matched a ground truth label, and –2 if none of the labels matched (minimal score). To rank the submissions, the average score (theoretically in the range –2 to +4) over all test images was used.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

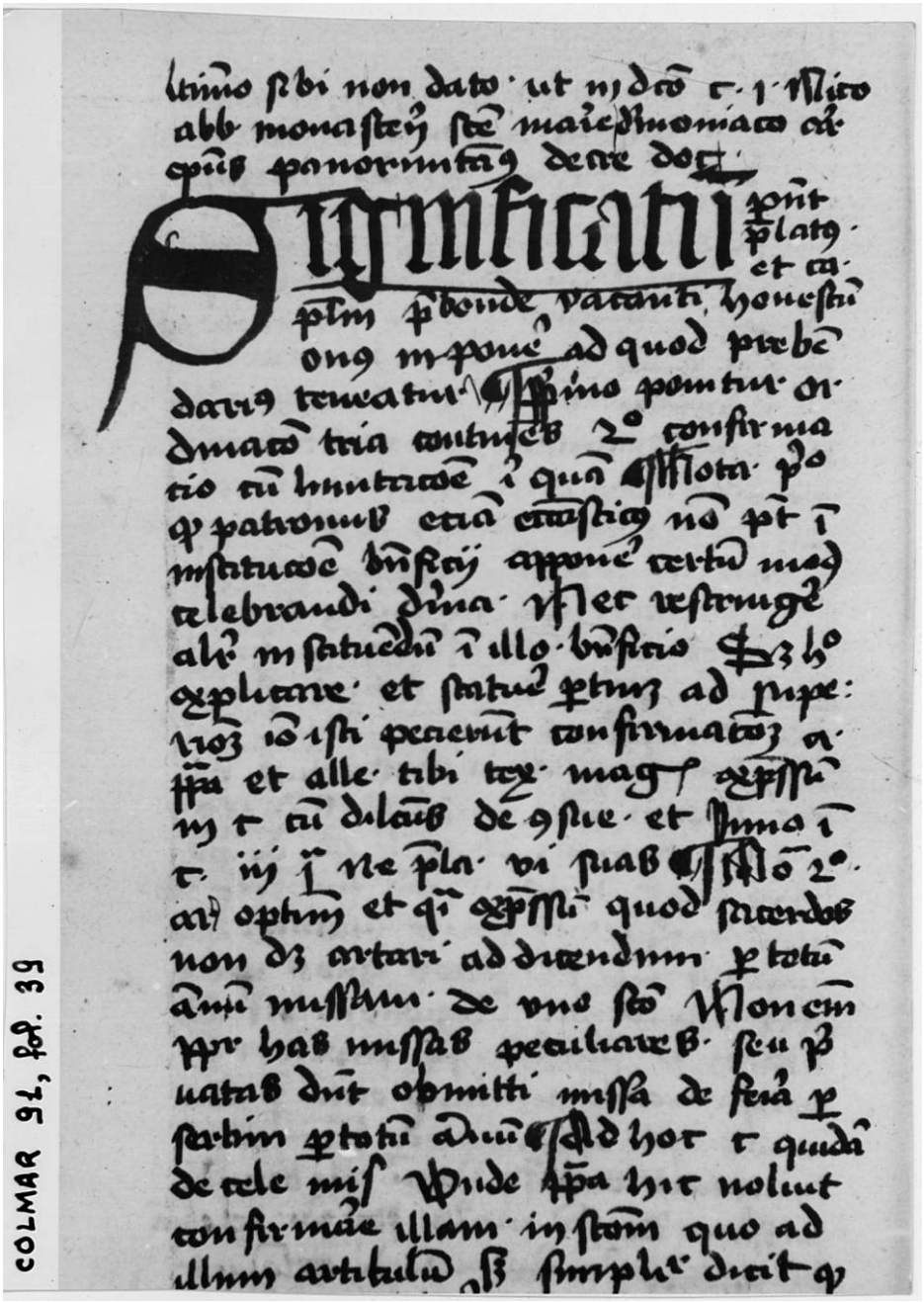


Fig. 9. Manuscript with two different script types (Task 2): Textualis for the first word of the chapter and Hybrida for the text (Colmar, Bibliothèque municipale, MS 91, fol. 39r).

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

Table 1
Competition Results for Task 1: "Crisp" Classification of 1,000 Single-Script Type Manuscripts

System	Accuracy (%)	Ranking according to accuracy	Average Intraclass Distance	Ranking according to AID
DeepScript	76.49	4	0.039	3
FAU	83.90	1	0.068	4
FRDC-OCR	79.80	3	0.018	1
NNML	83.80	2	0.026	2
TAU-1	49.90	7	0.421	7
TAU-2	50.10	6	0.417	6
TAU-3	52.80	5	0.393	5

Ranking

In the tables below (Tables 1–2), we present the results obtained for all the submitted systems after evaluating them on the competition test data.[33]

An insightful visualization of the systems' test predictions can be offered through so-called confusion matrices (Fig. 10). In such matrices, the horizontal axis plots the ground-truth labels for the test images (that is, the "correct" script types), whereas the vertical axis shows the labels that were predicted by a system. As such, these tables illustrate which script types were often misclassified in the test set and with which other script types they were typically confused. Good classification systems will be characterized by a dark diagonal, running from the top left to the bottom right of the plot, because the cells on this diagonal correspond to correct predictions.

When comparing the confusion matrices for both the FAU and DeepScript, we find it interesting that a number of similar patterns can be gleaned from them. Of particular relevance is the observation that the majority of confusions that arose in both systems make sense from a historical perspective and correspond to paleographical expert knowledge. The various subtypes of textualis script types, for instance, are regularly confused (textualis, Southern textualis, and semitextualis), including at the upper chronological border some examples of Praegothica. The most obvious misclassifications, however, occur between the various forms of Gothic cursive handwritings, such as the (semi)hybrida and cursiva, including the category of the semitextualis script type. The humanistic cursive letter is rarely misclassified, a result showing that what causes the confusion in Gothic scripts is not solely the cursive appearance, but may be some other feature or features specific to a given script family. A large number of prediction errors, for instance, arise because systems commonly confuse a hybrida with a semihybrida or a cursiva with a semihybrida. For textualis scripts as well as for cursiva ones, a very valuable result is that the confusion is not reciprocal but has a direction: in both systems, there are significantly more textualis examples that are misclassified as semitextualis than the reverse. Likewise, hybrida scripts are mapped onto semihybrida and (marginally) cursiva, while, on the one hand, semihybrida scripts are heavily confused with hy-

[33] The data have been taken from the competition's overview paper.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

Table 2
Competition Results for Task 2: "Fuzzy" Classification of 2,000 Dual-Script Type Manuscripts

System	Average score (%)	Ranking according to score	Average Intraclass Distance	Ranking according to AID
DeepScript	2.967	1	0.146	3
FAU	2.784	2	0.174	4
FRDC-OCR	2.631	4	0.120	1
NNML	2.771	3	0.134	2
TAU-1	0.615	6	0.260	6
TAU-2	0.590	7	0.259	5
TAU-3	1.226	5	0.356	7

brida and cursiva scripts. On the other hand, cursiva scripts are mostly correctly identified. This corresponds not only to the definition of these script classes (semihybrida is an intermediary stage between hybrida and cursiva) but also to historical developments: the semihybrida is a later, less formal creation, derived from hybrida and reintegrating some features (for example, looped ascenders) of cursiva, from which hybrida was deliberately distinguished in the first place.

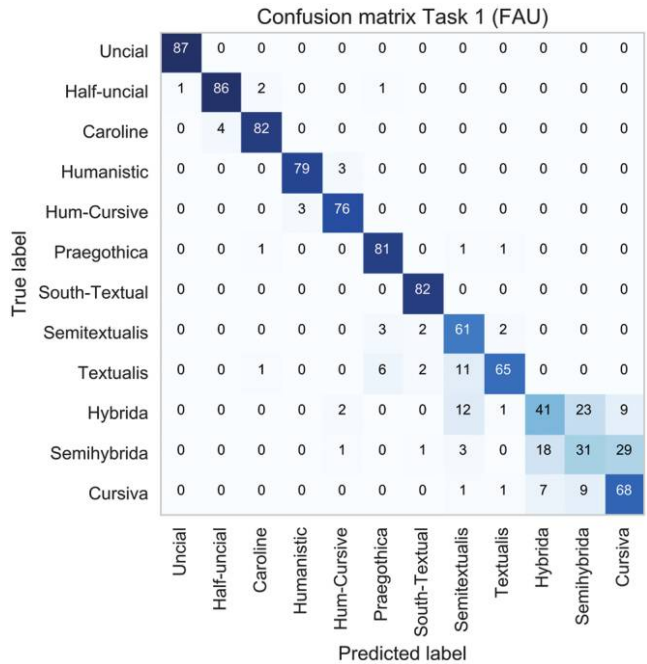
To what extent should such misclassifications be considered "errors"? Follow-up discussions about the competition's results among paleographers showed that the ground-truth annotations provided by the organizers, while certainly defendable, were not always free of controversy.[34] The categorization system used, while trying to discard the geographic component (except for northern and southern textualis), contradicts the classification systems developed by different paleographers in different parts of the world. It merges graphic phenomena that may be disconnected (loopless mercantesca, loopless Dutch hybrida) and separates script types that have a clear historical connection (cancelleresca as cursiva or as semihybrida).[35] Conversely, some geographically defined classifications lack some of the script categories in the CLaMM data altogether and cannot be applied consistently. Arguably, the competition therefore yielded the valuable insight that our computer systems, although trained on a consistently annotated data set, nevertheless experienced great difficulties in attempting to project this classification model onto new, unseen images. Epistemologically, this sheds an interesting light on the concept of "ground truth" in humanities research. The classification systems would have reached much higher accuracy scores if specific class pairs (such as the southern textualis and the northern textualis for DeepScript, or semihybrida and cursiva for both systems), would have been merged in the ground truth.

Therefore the computational modeling of script types raises interesting questions as to the feasibility—or even desirability—of distinguishing between specific

[34] These discussions took place at the center of an international seminar "Paléographie numérique: Du défi technique au défi épistémologique / Digital Paleography: From Technical to Epistemological Challenge" at the Fondation des Treilles: see http://www.les-treilles.com/paleographie-numerique-digital-palaeography/.

[35] Derolez, *The Palaeography of Gothic Manuscript Books*, 156 and 171.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).



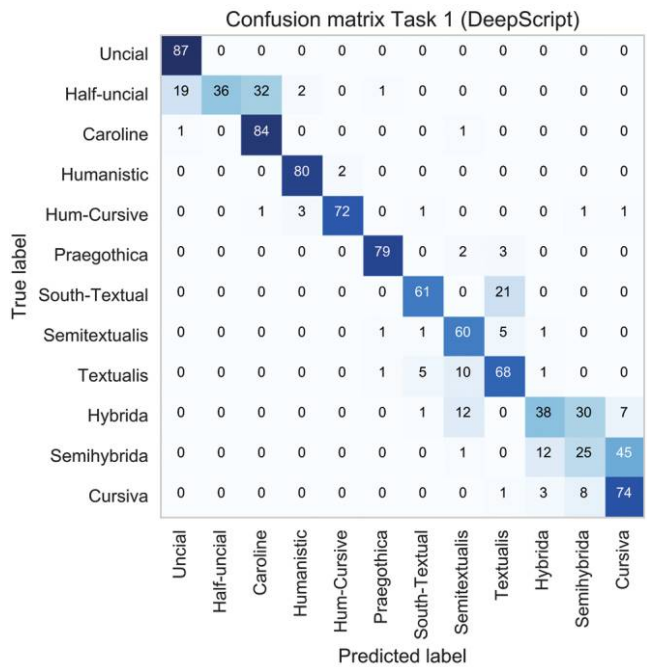


Fig. 10. Confusion matrices for the crisp problem classification in Task 1 (FAU and DeepScript, respectively).

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

script types, questions that can be expected to fuel further debate in the paleographical community. Indeed, while Northern textualis and Southern textualis are established, uncontroversial script types (except in regard to the names that are conventionally used), computational modeling helps us to understand the shortcomings of some class definitions that are based on formal criteria that may be seen as "symptoms" of a script type rather than its core definition. In this regard, it is absolutely crucial to gain insight into the decision mechanisms that underlie the automated classification (What features were considered? How does the machine distinguish between class A and class B?) and the paleographical characteristics on which the classification is based. For example, if loops are taken into account, then the mercantesca may rightly be divided into hybrida and semihybrida, but one can choose other criteria for analysis. From the point of view of machine learning, it might also be worth merging script types that are difficult to distinguish in order to generate fewer models. Interestingly, these results also suggest that the differences between, for instance, a hybrida and semihybrida should probably by viewed in a gradual continuum in which hybrid manuscripts might resist a binary, single-label classification.

Model Inspection and Model Criticism

Unsupervised Learning

As mentioned above, the FAU system depends on the extraction of SIFT keypoints from images, which are subsequently aggregated and used to train a classification system. The SIFT algorithm has the disadvantage that these generic features are extracted using a fixed algorithm that is hard to optimize in the light of a specific task, such as script-type identification. However, because these features can automatically be extracted from manuscript images, even if we have no other metadata concerning these manuscripts' provenance this result opens up interesting possibilities for "unsupervised" methods that do not depend on the availability of labeled data. An interesting question is therefore how a computational model might categorize the available manuscript images if the model was unaware of the preexisting classification available in the ground truth.

One visualization method is t-*Distributed Stochastic Neighbor Embedding* (t-SNE).[36] It enables a visualization of high-dimensional data in, for instance, a two-dimensional map that preserves the local structure of the data as much as possible—that is, images close to each other in the high-dimensional space will also lie close to each other in the result. Fig. 11 visualizes the global descriptors of the FAU system. The colors are arbitrarily chosen, but the dimensionality reduction preserves the clusters and their relative positions.

With very few outliers, all script types form clearly separated groups, except for the Gothic cursive family (hybrida, cursiva, and semihybrida). While these results might have been expected on the basis of the confusion matrices, such visualization provides new insights. As was done with the projection of such classes in the

[36] Laurens J. P. van der Maaten and Geoffrey E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *Journal of Machine Learning Research* 9 (2008): 2579–605.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

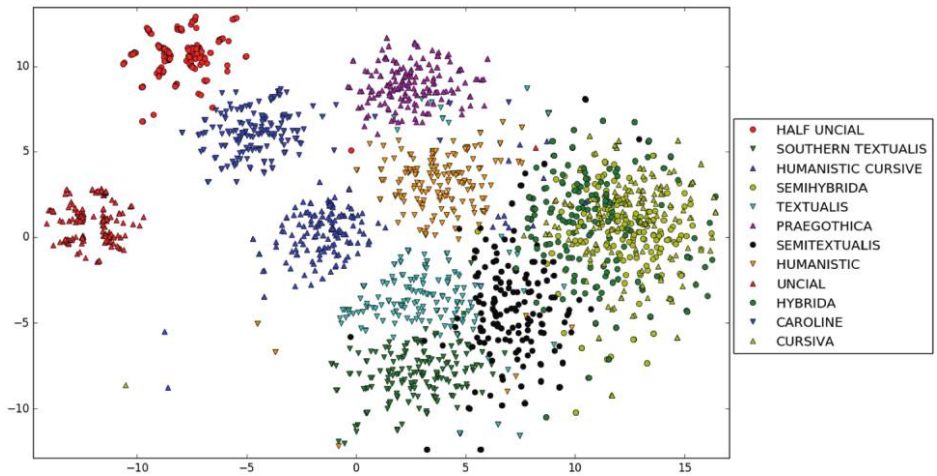


Fig. 11. Visualization of the FAU descriptors using t-SNE.

Graphem project,[37] we can try to analyze the repartition in evolutional terms. Uncial, Half Uncial, Caroline, and Praegothica are located in the upper left part. Unsurprisingly, Humanistic script lies close to the Praegothica from which it has been imitated, whereas Humanistic cursive seems slightly closer to pure Caroline script. Both humanistic script types can be found, surprisingly, in the middle of the graph. The Gothic evolutions may be read in the trajectory from Praegothica to Textualis and Southern Textualis and a sharp bend towards the Cursiva and (Semi)hybrida types. Semitextualis is in the middle, corresponding to the main threads of the Semitextualis types (the Southern-specific scripts, the simplified scripts used in universities, and the simplified formal Textualis used in late manuscripts in the Low Countries and Germany).[38] Within the Cursiva and Semi(hybrida) types, it is also natural to find the Hybrida positioned closer to the Textualis and Praegothica types. The relative positions in the t-SNE visualization are no proof of historical dynamics but are a very convenient way to highlight the similarities and connections between scripts.

Filter Visualization

In the past, neural network approaches such as the one used in DeepScript have often been considered "black boxes" because it proved difficult to explain on which sort of input features a trained model based itself when classifying new images. Luckily, many advances have been made in recent years with respect to the ex-

[37] Dominique Stutzmann, "Conjuguer diplomatique, paléographie et édition électronique: Les mutations du XIIe siècle et la datation des écritures par le profil scribal collectif," in *Digital Diplomatics: The Computer as a Tool for the Diplomatist*, ed. Antonella Ambrosio, Sébastien Barret, and Georg Vogeler (Vienna, 2014), 271–90, at 273 and plate 16 on p. 333.

[38] Derolez, *The Palaeography of Gothic Manuscript Books*, 118–23.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

planatory power of neural networks. The following technique, for instance, is commonly applied. As described above, a neural network consists of an interconnected structure of neurons: these small information units process the information streams that they obtain from lower layers in the network. Depending on what sort of incoming information an individual neuron is sensitive to, the neuron will be assigned an activation score that indicates how strongly the neuron "fires."

One common technique to explore the inner working of a model is, for instance, to inspect the twelve neurons in the output layer that control the ultimate classification of an image into one of the script types included. Next, we feed a series of image patches through the network and we keep track of which of these image patches maximally activate the output neuron associated with a particular script type. As a result, we can obtain a list of image patches per script type which, according to the system, present the most typical examples of a class. In Fig. 12, we show the highest-scoring patches for a number of script types, which were obtained following this procedure. As can be gleaned from this figure, the patches indeed offer clean, textbook examples of these script types.

Another widespread visualization technique also builds on the idea that the activation scores of individual neurons in a network can be manipulated for explan-

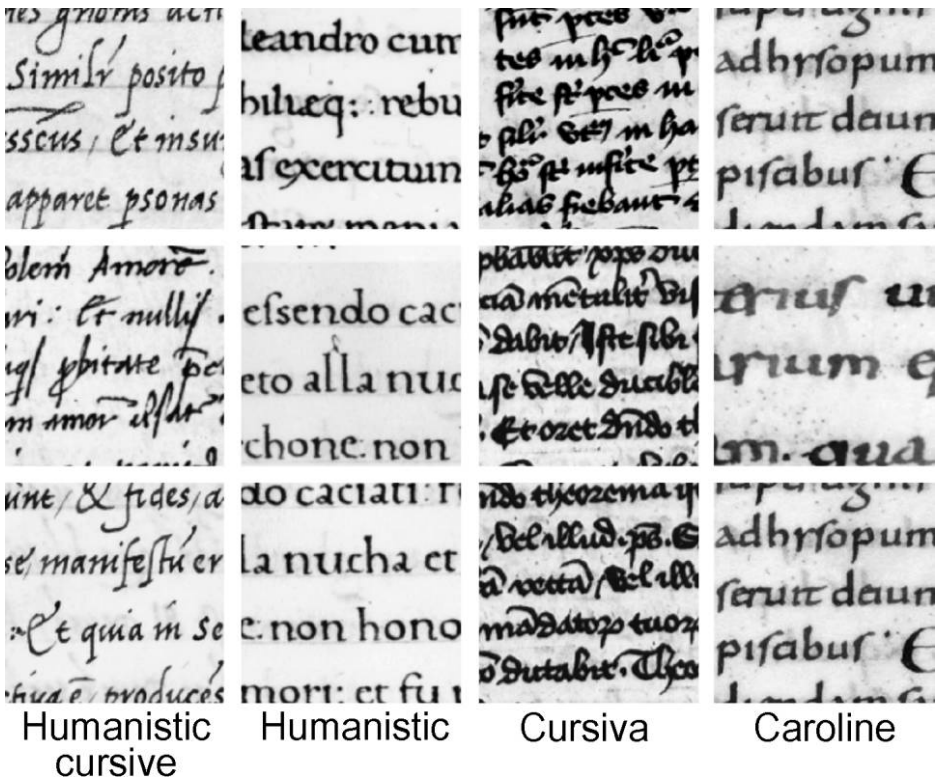


Fig. 12. For a number of representative script types, we show the patches that maximally activated the corresponding output neuron in the DeepScript network.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

atory purposes.[39] With this method, we first select a high-level layer in the network (for example, the last convolutional layer in the network) and iteratively visit each of the neurons in this specific layer. For each neuron, we randomly generate a gray-scale image of the same size as the input patches, which will initially be reminiscent of the "snowy screen" on a broken television. Subsequently, we start a loop in which we feed this image through the network, and each time we inspect how strongly it activates the neuron we are currently analyzing. In each iteration, we use a mathematical principle called "gradient ascent" to apply small changes in the pixel values in the randomly generated image, so that the artificial image gradually comes to maximize the activation of this specific neuron. In the literature, it has often been demonstrated that this procedure reveals interesting patterns that the neural network is particularly sensitive to in images.

In Fig. 13, we show a raster of patches generated for the twenty-five neurons from the last convolutional layer in our network for which we were able to obtain the highest activation scores (after three thousand steps of gradient ascent). A straightforward example of how these visualizations support interpretation are the filters plotted in the second and third positions (in the top row): here, the procedure has automatically generated a cloud of highly similar letter-like forms, which have in common that they have an ascender that contains a loop. Such filters offer a compelling demonstration that the neural network has indeed automatically learned to detect one of the primary morphological features—that is, the presence or absence of loops in ascenders—that are used to distinguish textualis script types from cursiva ones. Of course, the classification of script types in the training data will indirectly have guided the network to detecting such features, although it is an interesting added value of neural networks that they automatically learn to develop filters that are sensitive to such complex features: in other words, we never steered the algorithm towards specific regions of interest such as words or characters.

The neurons with ascenders in Fig. 13 are clear-cut examples of filters that are highly sensitive to very specific, local features in script: we see that during the gradient-ascent procedure, a series of isolated, roughly identical character shapes are created and displayed at seemingly random locations in the generated patches. However, it is clear that the majority of filters in Fig. 12 do not yield such a straightforward repetition of local morphological features. Rather, these filters seem to be sensitive to higher-level patterns in a given script type. How do these filters relate to the original data? A first interpretation, for instance, for the second filter in the second row was that it perhaps specifically picked up on the texture of paper (as opposed to vellum) in the manuscripts' margins, which would be a useful, albeit indirect, feature to help separate humanistic from early medieval writing. This possibility, however, was quickly rejected in a discussion with paleographers, because such patterns were generally too faint to be discerned in the actual input images. To Marc Smith (École nationale des chartes, Paris), we owe the valuable suggestion that the sensitivity to higher-level patterns might rather relate to the formality, regularity, or "rhythm"

[39] Matthew D. Zeiler and Rob Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings*, part 1, ed. David Fleet et al. (Zurich, 2014), 818–33.

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

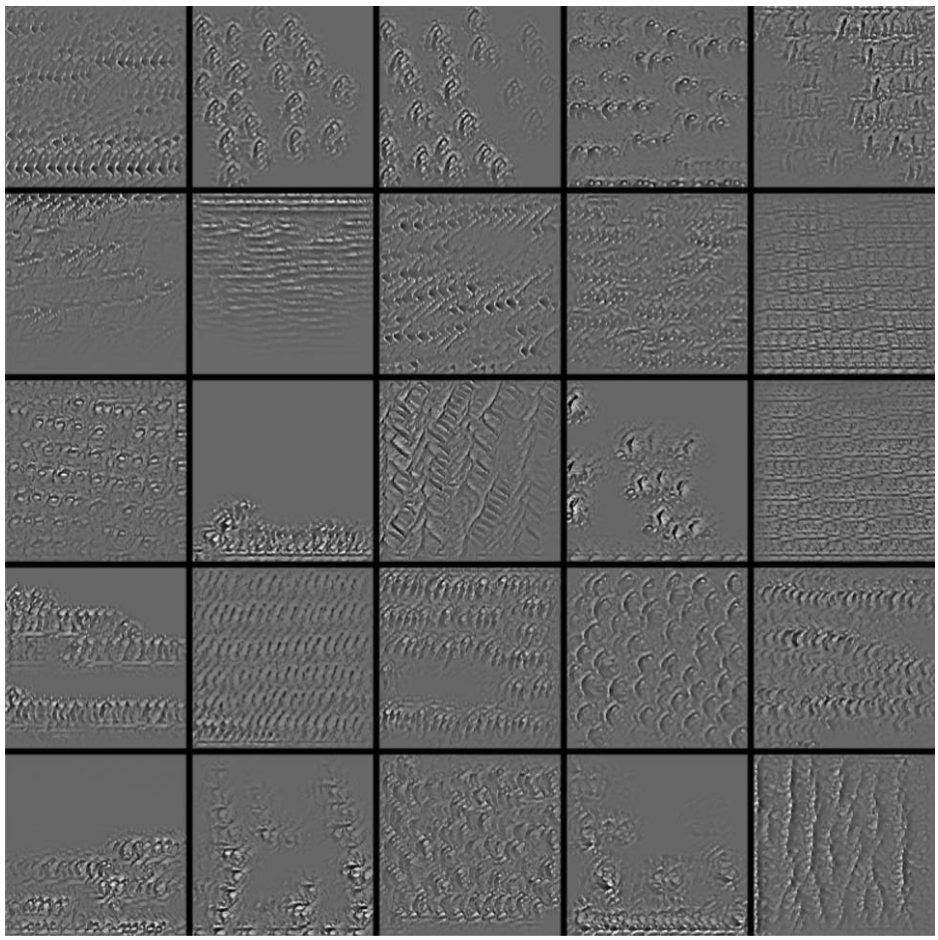


Fig. 13. Explanatory filter visualization for some of the neurons in a higher-level layer in the DeepScript network. These patches have been artificially generated in order to maximally excite a particular neuron.

of a script, a characteristic that Peter Stokes elsewhere in this volume defines as "the regularity of the strokes and the spatial relationship between them." Indeed, some of the filters seem to capture the regularity with which certain letter shapes appear, such as the pattern (of cursive minims?) that seems to be captured by the second filter in the penultimate row in Fig. 13.

Conclusion

In paleography, as is much the case in any other field in the humanities that involves a considerable hermeneutical effort, meaning is under constant negotiation: an impression of "truth" can only emerge when the opinions of the various stakeholders in a field temporarily align and reach an intersubjective equilibrium. In this paper, we discussed a competition in which intelligent machines attempted to repro-

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).

duce such a "truth" in the form of a categorization of images of medieval manuscripts in twelve script types. To this end, these machines had access to a reference data set of hand-labeled images as a so-called ground truth. Depending on the systems used, we see that computer algorithms can solve such questions with reasonable accuracy, but remain far from "flawless" in their ability to predict. The obvious question now remains: have we learned anything in this attempt?

The first thing we can deduce from this exercise is that the concept of "ground truth" is a treacherous one. The categorization of specimens of medieval manuscript writing into twelve mutually exclusive script types essentially is an act of modeling that deliberately aims to simplify a complex reality.[40] Although it goes without saying that the competition's organizers have provided these annotations to the best of their abilities, they had to make difficult, ad hoc choices that might be difficult for machines to reproduce. Moreover, framing the problem of script identification as a classification task, in which only one from a series of possible labels must be chosen, hides the fact that many instances of medieval script might resist a "hard" single-label categorization. We should always remember that the difference between semihybrida and hybrida scripts, for instance, overall remains a gradual one, which might perhaps best be explained in probabilistic terms, where the presence of elements from one script type need not necessarily imply the full absence of characteristics from others, as the very names of those scripts indicate. The modeling efforts involved in this competition make us acutely aware of the need to address hybridity through more than one feature.

The modeling goal of this competition, as is typical of so much work in the digital humanities, forces scholars to rethink and formalize, in a fully explicit manner, the set of silent assumptions that they subconsciously rely on when describing a particular script as a "hybrid" script. The usefulness of computer simulations therefore lies primarily not in the fact that they may ultimately be able to solve certain problems for us, but in the ways in which they help us to challenge our own assumptions and sharpen our formulation of the problems.

[40] Willard McCarthy, "Modeling: A Study in Words and Meanings," in Schreibman, Siemens, and Unsworth, *Companion to Digital Humanities*, 254–70.

Mike Kestemont, University of Antwerp (mike.kestemont@uantwerpen.be)
Vincent Christlein, Friedrich-Alexander University Erlangen-Nürnberg (vincent.christlein@fau.de)
Dominique Stutzmann, Institut de Recherche et d'Histoire des Textes–Centre National de la Recherche Scientifique (IRHT-CNRS) (dominique.stutzmann@irht.cnrs.fr)

This content downloaded from 194.080.232.075 on July 02, 2018 16:21:24 PM
All use subject to University of Chicago Press Terms and Conditions (http://www.journals.uchicago.edu/t-and-c).