Палеографическое датирование берестяных грамот

К. А. Сидоров
sidorovk@cardiff.ac.uk
Cardiff University, Cardiff, UK

В статье предлагается новый метод для автоматического датирования берестяных грамот по виду графем (палеографически). Предлагаемый метод показывает среднюю абсолютную ошибку всего в 18,9 лет, то есть не хуже точности датирования экспертами.

Ключевые слова: *анализ почерка, анализ документов, палеография, вычислительная археология, морфометрика, распознавание образов, берестяные грамоты.*

Paleographic Dating of Birch Bark Manuscripts

K. A. Sidorov
Cardiff University, Cardiff, UK

We propose a novel method for automatically estimating the age of birch bark manuscripts based solely on the appearance of graphemes (paleographic dating). Our method achieves mean absolute accuracy of 18.9 years which is comparable to or surpasses the performance of human experts and of other computational paleography studies.

Keywords: *handwriting analysis, document analysis, paleography, computational archæology, morphometrics, pattern recognition, birch bark manuscripts.*

10 We address the problem of estimating the age of birch bark manuscripts [22] from their appearance (paleographically). These birch bark manuscripts (BBMs) are one of the most valuable corpora of Old Russian texts, and a crucially important source of information about mediæval history and evolution of the language. Accurate dating of BBMs is necessary in order to place them in a correct historical context, before historians and linguists can take advantage of their valuable contents.

1. Background

Paleographic dating is possible due to the fact that the appearance of graphemes does not remain constant over time. For BBMs, a considerable effort has been undertaken by Zaliznyak *et al.* [22] (pp. 134–429) to codify paleographic expertise as *objectively* as possible. They have constructed paleographic tables [22] for each grapheme, relating the occurrences of graphemes' appearance features to dates at which these are known to occur. Recently, using computer vision and machine learning techniques, He *et al.* [7–11] have addressed the problem of direct paleographic dating of mediæval charters, with accuracy in the order of decades. In [7], texture-level features, previously used for writer identification (Hinge, Fraglets), are used to form descriptors for the entire documents, and dating is done with two-level (global and local) support vector regression. The approach in [11] is based on a histogram of orientations of strokes as a descriptor, and a procedure based on 3D self-organising map to discover correlations between features and dates. In [21] a bag-of-words approach on shape context vectors extracted from edge maps in manuscripts is used for dating Swedish mediæval charters.

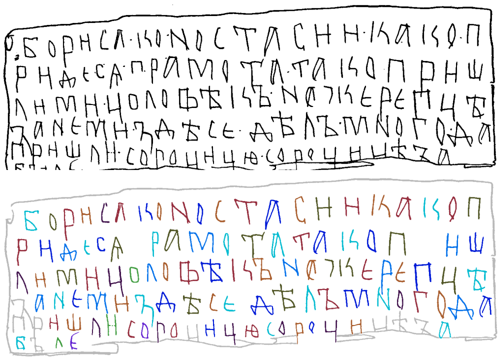


Fig. 1. *Top:* outline drawing of BBM №43. *Bottom:* the result of segmentation into graphemes.

2. The Proposed Approach

We have extracted (see Fig. 1) and annotated 48,870 graphemes from the outline drawings in the BBM corpus (data were predominately taken from the digital archive [1] and from digitised reproductions in the relevant volumes of series [22]). A total of 814 manuscripts, from which at least one grapheme could be reliably extracted, were processed.

Following [22], we reduce the problem of dating a manuscript to the problem of dating individual graphemes in it, and then aggregating the results. We attack the problem on two fronts: first, we use a model of grapheme deformation, based on groupwise nonrigid registration of grapheme images; second, we employ a model based on convolutional neural networks to capture nuanced details in appearance variation.

Deformable Model of Graphemes. As the first angle of attack, we parameterise the large-scale variation in grapheme shapes with a low-dimensional representation. We employ the classic technique of decomposing the appearance into deformable "shape" and the corresponding "texture" and modelling them jointly. This technique has been variously known as Morphable Model or Appearance Model (AM) [5]. This approach allows us to describe as much as possible of the variation in graphemes' appearance as a smooth deformation of their shape, yet describe the residual variation (subtle nuances that cannot be explained by smooth elastic deformation, *e.g.* changes in topology) as variation in "texture". The main challenge involved in building such appearance models is finding correspondences between analogous parts of all deformable exemplars: the problem of groupwise non-rigid registration, for which several approaches have been developed [4, 15]. The standard idea is to iteratively align all exemplars to a common reference model, which itself is iteratively evolved by averaging aligned exemplars. However, we found that intensity-based image registration techniques [3, 4, 15] do not work well with binary grapheme images, as the objective function is not smooth and/or convex enough.

Algorithm 1 Register grapheme images to build AM

```
Require: Images $I^i$ ($i \in \{1 \ldots n\}$); tolerance $\varepsilon_{\mathrm{tol}}$; max epochs
    $t_{\mathrm{max}}$; buffer dimensions $w$, $h$; point cloud size $p$.
 1 function RegisterGraphemes($I$, $\varepsilon_{\mathrm{tol}}$, $t_{\mathrm{max}}$, $w$, $h$, $p$)
 2 initialisation:
 3 $a_{\mathrm{orig}} \leftarrow 0$                                         ▷ total area
 4 for $i \leftarrow 1$ to $n$
 5     triangulate images:
 6     $\{V^i_{2\times v_i}, T^i_{3\times t_i}\} \leftarrow$ Triangulate($I_i$)
 7     $D^i \leftarrow \mathbf{0}_{2\times v_i}$                          ▷ initialise deformations
 8     $a_{\mathrm{orig}} \leftarrow a_{\mathrm{total}}$ + Area($V^i$, $T^i$)
 9 iterative alignment:
10 for $t \leftarrow 1$ to $t_{\mathrm{max}}$
11     $R \leftarrow \frac{1}{n}\sum_{i=1}^{n}$Rasterise($V^i + D^i$, $T^i$)           ▷ reference
12     if Error($R$, $V$, $T$, $D$) $< \varepsilon_{\mathrm{tol}}$ then
13         break                                        ▷ tolerance reached
14     $a \leftarrow 0$
15     for $i \leftarrow 1$ to $n$                       ▷ align each grapheme to $R$
16         remove $i$-th image from reference:
17         $\hat{R} \leftarrow (nR-$Rasterise$(V^i + D^i, T^i))/(n-1)$
18         $P_{2\times p} \leftarrow$ Sample($\hat{R}/\sum\hat{R}$, $p$)              ▷ sample points
19         $D^i \leftarrow$ CPD($P$, $V^i + D^i$)                      ▷ align with [16]
20         $a \leftarrow a+$Area($V^i + D^i$, $T^i$)
21     ensure area is preserved:
22     $\mathbf{c} \leftarrow \left(\sum_{i=1}^{n}\sum_{j=1}^{v_i}(V^i(:,j) + D^i(:,j))\right) / \sum_{i=1}^{n} v_i$
23     for $i \leftarrow 1$ to $n$
24         $D^i \leftarrow \mathbf{c} - V^i + (V^i + D^i - \mathbf{c}) \cdot \sqrt{a_{\mathrm{orig}}/a}$
25 compute shape and texture for appearance model:
26 $X_{2\times s} \leftarrow \forall(x, y) \mid R(y, x) > 0$
27 for $i \leftarrow 1$ to $n$                          ▷ align $R$ to each grapheme
28     $P_{2\times p} \leftarrow$ Sample($R/\sum R$, $p$)                    ▷ sample points
29     $D^i_{\mathrm{ref}} \leftarrow$ CPD($V^i$, $P$)                          ▷ align with [16]
30     $S^i_{2\times s} \leftarrow$ Extrapolate($D^i_{\mathrm{ref}}$ on $X$)                   ▷ shape
31     $T^i_{1\times s} \leftarrow I^i(S^i)$                                   ▷ texture
32 return $S$, $T$
33 function Error($R$, $V$, $T$, $D$)
34 $\varepsilon \leftarrow 0$
35 for $i \leftarrow 1$ to $n$
36     $\hat{I} \leftarrow$ Rasterise($V^i + D^i$, $T^i$, $w$, $h$)
37     $\varepsilon \leftarrow \varepsilon + \sum_{x=1}^{w}\sum_{y=1}^{h}|R(y, x) - \hat{I}(y, x)|$
38 return $\varepsilon/n$
```

We, therefore, employ a hybrid approach: we evolve the reference model as an image, yet use point cloud registration to align graphemes to it. The procedure is summarised in Algorithm 1. The input grapheme images are first densely triangulated into meshes (line 6, see Fig. 2 (a)) and the total area of all meshes in the ensemble is remembered (lines 3, 8). The algorithm proceeds in epochs (lines 10–24). In each epoch, the reference R is first computed (line 11) by averaging the rasterised warped meshes. The algorithm proceeds to align each grapheme to the current reference (lines 15–24). (To avoid problems with local minima, the grapheme being aligned is excluded (line 17) from the reference.) To do this alignment, the reference R is treated as a probability mass function of some distribution, and p points $P_{2\times p}$ are randomly drawn from it (line 18) (importantly, different points are drawn at each iteration). Figure 2 (b) shows the reference R at some iteration, with sampled points P overlaid (in red). The vertices V^i of the grapheme's mesh are aligned to these points P (line 19), yielding vertex displacements D^i. To align the two point clouds V^i and P we use the Coherent Point Drift (CPD) algorithm described in [16]. We use the following parameters for CPD: Gaussian width $\beta = 1.4$, regularisation weight $\lambda = 8$, outliers ratio 0.7, and we run it for maximum of 100 iterations (for more detail on these parameters see [16]). Figure 2 shows a grapheme $\{V^i, T^i\}$ (a) being aligned to reference points P (b) yielding a warped mesh $\{V^i + D^i, T^i\}$ (c) rasterised as (d). To avoid a run-away effect to the trivial solution, whereby all meshes may become compressed into a point, after each epoch we re-scale the meshes to preserve the total original area (lines 21–24). Figure 4 shows the reference R as the registration proceeds. Note how as the alignment becomes better, the average grapheme appearance R becomes progressively crisper, indicating good alignment. The registration terminates either when the maximum number of epochs t_{max} has

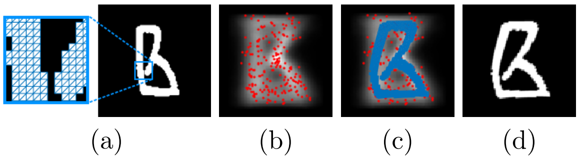


Fig. 2. Registration of graphemes to the reference.

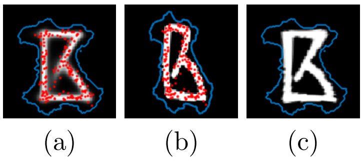


Fig. 3. Registration of the reference to graphemes and sampling of the texture.

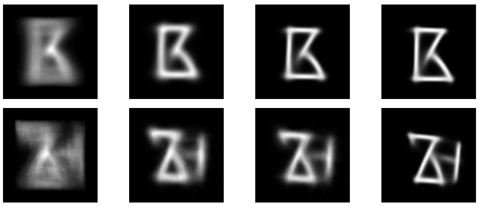


Fig. 4. The average shape and texture of the ensemble as the registration progresses.

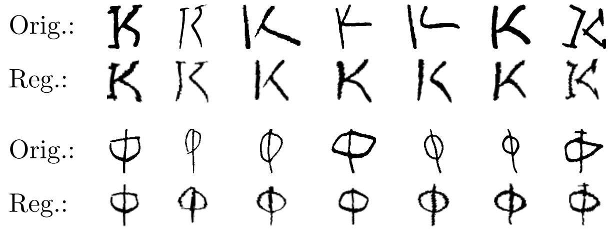


Fig. 5. Examples of graphemes before and after registration (shape normalised).

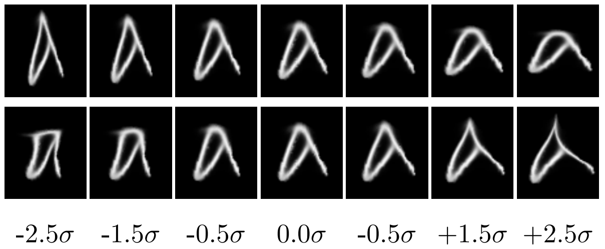


Fig. 6. Most significant of appearance variation for some of the graphemes.

exceeded or when the registration error is below acceptable ε_{thr} (lines 12–13, 33–38).

Finally, we compute the shape and texture describing each grapheme, by finding an optimal alignment of the reference to the graphemes. To do so, we find (line 26) the union X of all aligned graphemes (its boundary is shown in blue in Fig. 3). For each grapheme, as above, points are sampled from the reference, but this time the reference points are aligned to the grapheme mesh (line 29). The thus computed deformation is extrapolated to the entire domain X using radial basis functions (line 30) to find the shape deformation S^i that best explains a grapheme in terms of the warped reference. Given the extrapolated deformations, the texture T^i is found by appropriately sampling the original grapheme image (line 31). Figure 3 shows the reference shape (a) warped to align with a grapheme (b), and the texture sampled from the grapheme into reference shape (c). Figure 5 shows the results of the groupwise registration of graphemes.

Having obtained samples of the deformed shapes S^i and the corresponding textures T^i, we seek to obtain their low-dimensional representation. In this work we employ basic linear dimensionality reduction with Principal Component Analysis (PCA), as done in [5], yielding a twice-linear model

$$E\boldsymbol{a} = \begin{pmatrix} E_{cs} \\ E_{ct} \end{pmatrix} \boldsymbol{a} = \begin{pmatrix} \boldsymbol{c}_s \\ W_t \boldsymbol{c}_t \end{pmatrix} = \begin{pmatrix} E_s^T(\boldsymbol{s} - \boldsymbol{\mu}_s) \\ W_t E_t^T(\boldsymbol{t} - \boldsymbol{\mu}_t) \end{pmatrix},$$

with the shape model $\boldsymbol{s} = E_s\boldsymbol{c}_s + \boldsymbol{\mu}_s = E_s E_{cs}\boldsymbol{a} + \boldsymbol{\mu}_s$ and the texture model $\boldsymbol{t} = E_t\boldsymbol{c}_t + \boldsymbol{\mu}_t = E_t W_t E_{ct}\boldsymbol{a} + + \boldsymbol{\mu}_t$, where $\boldsymbol{a}$, $\boldsymbol{s}$, and $\boldsymbol{t}$ are the appearance, shape, and texture parameters (feature vectors) respectively; E, E_s, E_t are the corresponding eigenvectors, and W_t are normalisation weights [5]. We preserve 128 dimensions in appearance feature vectors.

№	Type	Output	№	Type	Output
0	Input	64×64×1	4	Conv 3×3@128	4×4×128
1	Conv 3×3@16	32×32×16	5	Conv 3×3@256	2×2×256
2	Conv 3×3@32	16×16×32	6	Conv 1×1@512	1×1×512
3	Conv 3×3@64	8×8×64	7	Conv 1×1@43	1×1×43
			8	Softmax Loss	

Table 1. Structure of our CNNs.

Analysis with CNNs. Deep convolutional neural networks (CNNs) have recently demonstrated excellent performance on diverse tasks ranging from image classification, to speech recognition, to natural language processing [6,13]. The main challenge in our case is that the number of training samples available per each of the 43 alphabetic classes divided between 19 temporal classes is orders of magnitude too scarce for naïvely training CNNs. We employ two strategies to tackle sample deficit. First, we train a CNN on the task of classifying graphemes into alphabetic classes (OCR), without subdivision into temporal classes. (This OCR CNN was also used in annotation of the dataset.) We discovered that it is possible to achieve this with a very small CNN that does not over-fit despite sample deficit. Having trained the OCR CNN, we reuse the low- and mid-level representations learned by it for the task of dating. This technique is known as transfer learning [2, 18]. Second, instead of pursuing a classifier with 19 temporal classes, we reduce the problem to an ensemble of binary classifiers, as will be discussed below. In this regime there are substantially more samples available per class.

The structure of the network is shown if Table 1. It consists of 9 convolutional blocks. Each block consists of a convolutional layer (see Table 1 for filter dimensions and numbers; we use zero-padding throughout), followed by a dropout layer [17] with 0.2 rate, followed by a ReLU non-linearity, followed by 2×2 max-pooling (except in the last two blocks). The total number of parameters is 1.6×10^6.

The network takes as an input 64×64 images. All grapheme images were resized to 48×48, preserving the aspect ratio, and leaving 8 pixel border on each side (necessary for augmentation, see below). The size of the images was choosen by hyper-optimisation. We observed that much smaller images result in poor discrimination, while much larger sizes do not yield any appreciable increase in performance. Training was done for 1500 epochs using the standard stochastic gradient descent optimiser, with an initial learning rate of 0.001, descending to 0.0001 after 750 epochs. The batch size of 100 was chosen. We did not observe any increase in performance on the development set with longer training. To partially alleviate the imbalance between classes, batches are formed in a stratifying manner, sampling the more rare classes proportionately more frequently.

In order to decrease the sensitivity of the CNN to variations that may occur in the data yet do not have paleographic significance, the images in each batch were subject to augmentation by random affine transforms, and random alteration of the stroke widths by morphological erosion and dilation, which we perform on-the-fly. This augmentation serves as regularisation and helps alleviate over-fitting. For all experiments with CNNs we used the MatConvNet toolbox [20]. The error rate of the CNN on the OCR task was 2.04% on the testing set (1/8 of all letters) and 0.98% on the training set: over-fitting is very insignificant. Investigation of the misclassified samples shows that most of them are either highly distorted or mis-written letters.

We take the 512-dimensional output of the penultimate fully connected layer (block 7) as the high-level feature representation of the graphemes. As will be discussed below, multiple networks we re-trained (fine-tuned) as binary classifiers, to further improve their dating performance. The tuning proceeded as above (with block 8 replaced for the case of two labels), for 100 epochs and with a very small learning rate of 1×10^{-5}.

Ensembles of Classifiers. As in [22], we discretise the entire time line into $N_t = 19$ bins (temporal classes): between the years 1100–1400 each bin is 20 years long, in the 15th century the bins are 1400–1410, 1410–1420, 1420–1450, and in the 11th century, there is only one bin 1025–1100.

We proceed by reducing the problem of multiclass classification to training an ensemble of binary classifiers and combining their results. Two most common approaches are: one-vs-all (N classifiers are trained, to differentiate each class from all other classes) and one-vs-one ($N(N-1)/2$ classifiers to differentiate between all possible pairs of classes) [12]. The relevant more general theory here is the method of classifier ensembles based on error-correcting codes (ECOC) [12], which considers arbitrary dichotomies of a multi-class set into two-class sets. Using the ECOC terminology [12], we represent a dichotomy of an N_t-class set as a binary vector $\boldsymbol{d} \in \{0,1\}^{N_t}$, whose elements represent to which of the two new classes the original classes correspond. Similarly, ensemble of N_c classifiers, each with its own dichotomy, can be represented by a binary *coding matrix* $D_{N_t \times N_c}$. Standard ECOC coding matrices include [12] the above one-vs-all and one-vs-one schemes (as a generalisation), random codes, and even exhaustive codes (prohibitive in our case).

Zaliznyak [22] notes that, instead of gradually evolving over time, writing style features first *abruptly* appear (get invented), then remain in use for a number of decades (co-existing and competing with other styles for their relative frequency), then finally die down [22]. We take advantage of this fact and train an ensemble where each classifier distinguishes between a particular *continuous* segment in time and all other dates, with one classifier for each possible segment: $[1\,..\,1]$, $[1\,..\,2]$, $\ldots$, $[1\,..\,(N_t-1)]$, $[2\,..\,2]$, $[2\,..\,3]$, $\ldots$, $[2\,..\,(N_t-1)]$, $\ldots$, $[(N_t-2)\,..\,(N_t-1)]$, $[(N_t-1)\,..\,(N_t-1)]$. It is easy to see that there are $N_t(N_t+1)/2 - N_t/2 = N_t(N_t - - 1)/2 = 171$ such classifiers. With this arrangement, we expect that for some of the classifiers in the ensemble, their corresponding segments will coincide *exactly* (modulo discretisation) with the lifetime of some paleographically significant features, hence making (at least some of) the classifiers much more sensitive that in any other regime.

We have trained, for each grapheme in the alphabet, a full set of 171 classifiers using a basic linear SVM using appearance model features, and a further ensemble using CNN features. For each classifier, we estimated its performance by computing the confusion matrices $C_{2\times2}$ on a small validation subset of the training set. ($C(i,j)$ shows the number of samples known to be from class i classified as belonging to class j.) We have also computed the unweighted average recall for each classifier:

$$\mathrm{AR}(C_{K\times K}) = \frac{1}{K}\sum_{i=1}^{K}\frac{C(1,1)}{\sum_{j=1}^{K} C(i,j)}. \quad \text{(here, } K=2\text{)} \qquad (1)$$

We then proceeded to fine-tune an ensemble of CNNs as binary classifiers, for those dichotomies for which corresponding SVM classifiers yielded a “promising” performance (AR $\geqslant 60\%$). We then re-trained the ensemble of SVM classifiers from the thus updated CNN features. The rationale for tuning is as follows: the generic OCR network serves as a good initialisation (and its features, as we noted, are already useful for dating), and tuning the ensemble of networks, each to its assigned time segment, further improves the sensitivity by improving the learned features (at all levels) that may be specific to a particular time segment.

Aggregating Evidence. We adopt Bayesian inference as the most natural framework in which to combine the evidence from multiple graphemes, and for each grapheme — the predictions of multiple classifiers. Given an unseen manuscript to be dated, we extract all the graphemes and evaluate each of them with the corresponding classifier ensemble. To fuse the evidence from all classifiers, we maintain a discrete probability mass function $P_{1\times N_t}$, $P(a) = P(a|L_1, L_2, \ldots)$, which depends on observing graphemes $L_1, L_2, \ldots$, over the discrete set of $N_t = 19$ temporal classes $a \in \{1 \ldots N_t\}$.

Out of necessity, we make the assumption that observing each next grapheme L_i in the manuscript is an event independent from observing other graphemes in the same manuscript.

Algorithm 2 Bayesian aggregation step

```
Require: Prior distribution $P_{N_t\times1}$; classifier's dichotomy $\boldsymbol{d}_{N_t\times1} \in \{0,1\}^{N_t}$, vote $e \in \{0,1\}$, normalised confusion matrix $C_{2\times2}$, smoothing parameter $\alpha$, threshold $\mathrm{AR}_{\mathrm{thr}}$.
function UpdateBelief($P$, $\boldsymbol{d}$, $v$, $C$)
if $\sum_{i=1}^{2} C(1,i) = 0$ or $\sum_{i=1}^{2} C(2,i) = 0$ then
    return $P$                                  ▷ uninformative classifier, abort
if $\frac{1}{2}(C(1,1)+C(2,2)) <\mathrm{AR}_{\mathrm{thr}}$ then
    return $P$                                  ▷ below AR threshold, abort
$P_{\mathrm{new}}(i) \leftarrow \mathbf{0}_{N_t\times1}$
for $i \leftarrow 1$ to $N_t$                   ▷ update belief
    $P_{\mathrm{new}}(i) \leftarrow P(i)\cdot\hat{C}(\boldsymbol{d}(i)+1, e+1)$
$P_{\mathrm{new}} \leftarrow P_{\mathrm{new}}/\sum_i P_{\mathrm{new}}(i)$     ▷ normalise
return $(1-\alpha)P_{\mathrm{new}} + \alpha P$   ▷ apply smoothing
```

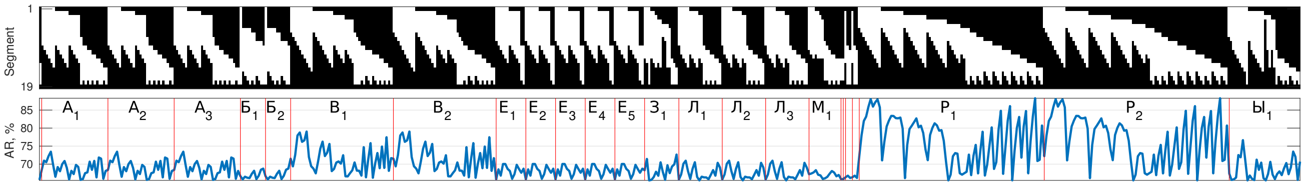


Fig. 7. Example of dating with a classifier ensemble (BBM 436). *Top:* map of classifier segments. *Bottom:* unweighted average recall of the classifiers. Red lines delineate individual letters (25 letters survived thresholding in this example).

Using the independence assumption, we thus perform Bayesian update of P for each evidence $e \in \{0, 1\}$ from each classifier from each grapheme as follows:

$$P_t(a|e_t) \leftarrow \frac{P(e_t|a)P_{t-1}(a)}{P(e_t)} \qquad (\forall a \in \{1 \dots N_t\}). \qquad (2)$$

We can estimate $P(e_i|a)$ and $P(e_i)$ from the classifiers' confusion matrices. Let C be the row-normalised confusion matrix for some classifier with coding vector $\boldsymbol{d} \in \{0, 1\}^{N_t}$, and assume the classifier outputted evidence $e \in \{0, 1\}$, then Eq. (2) becomes:

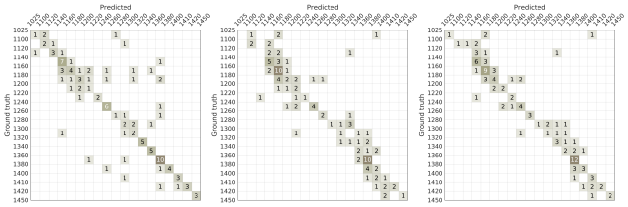
$$P_t(a|e_t) \leftarrow \frac{C(\boldsymbol{d}(a) + 1, e_t + 1)P_{t-1}(a)}{\sum_{i=1}^{N_t} C(\boldsymbol{d}(i) + 1, e_t + 1)P_{t-1}(i)}. \qquad (3)$$

The Bayesian update step is summarised in Algorithm 2. As an additional measure to improve the quality of evidence, we reject all classifiers for which the average recall is below a threshold $\mathrm{AR}_{\mathrm{thr}}$ (lines 4–5), as well as discarding completely uninformative classifiers (lines 2–3). To partially compensate for the violation of the independence assumption, we introduce a smoothing parameter α which reduces the amount of information each next classifier contributes (line 10). (We optimised the smoothing parameter α and the rejection threshold $\mathrm{AR}_{\mathrm{thr}}$ on a small (validation) subset of the training set.) Algorithm 2 is called to update P for each grapheme in a manuscript, for each classifier in the grapheme's ensemble, yielding the estimated date distribution for the manuscript.

Figure 7 illustrates the ensemble dating (Bayesian aggregation) on an example of a particular manuscript (BBM №436). It can be seen, in this example, that graphemes "р", "в", and "л" were among the most significant for dating.

3. Results and Evaluation

We applied our method to the task of dating the BBM corpus. We evaluate the performance of our method using eight-fold cross-validation (7/8th of the data in the training set, 1/8th in testing). Special care has been taken to ensure that documents (or parts of a document) written by the same hand are always in the same set. To qualitatively measure the dating performance of our system, we compute the mean absolute error (MAE), and the cumulative score (CS_α) which measures the fraction of test documents for which the date estimation error is no greater than α, as done in [7–11]. We additionally compute the Kullback–Leibler divergence (KDL) and the earth mover's distance (EMD) between the ground truth and the predicted distributions.



MAE=29.01 MAE=23.78 MAE=**18.93**

AM CNN AM+CNN

Fig. 8. Confusion matrices for the different combination of features.

We have evaluated our approach with both types of features, AM and CNN, and their combinations: at the feature level (AM∪CNN) and at the decision level (concatenating classifier ensembles), AM+CNN. We have done this for both simple majority voting (as a baseline) and Bayesian aggregation, with flat and empirical priors. As a baseline, we also performed aggregation by simple majority voting. The results are summarised in Table 2, the corresponding cumulative scores in Fig. 9 (left), and the resulting confusion matrices in Fig. 8. The best MAE achieved was **18.93** years with AM+CNN features and empirical prior. (Overall, using empirical prior gave minor improvement over flat prior, except for AM∪CNN features.) Bayesian aggregation overall yielded much better results than majority voting. Individually, CNN features performed better than AM. Combination at the decision level produced better result than at the feature level. We compare of our method (Table 3) with other relevant computer paleography studies [7–9, 11, 14], however exact comparison is difficult due to the different nature and amount of material

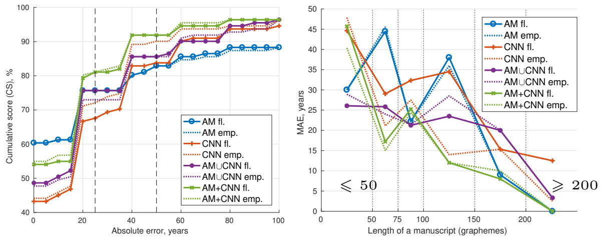


Fig. 9. *Left:* Cumulative scores (CS) for the different features. *Right:* Dating error as a function of the manuscript length.

Feat.	AM				CNN				AM∪CNN				AM+CNN			
Agg.	voting		Bayes		voting		Bayes		voting		Bayes		voting		Bayes	
Prior	fl.	emp.	fl.	emp.	fl.	emp.	fl.	emp.	fl.	emp.	fl.	emp.	fl.	emp.	fl.	emp.
MAE	96.71	77.39	28.83	29.01	41.19	47.84	24.10	23.78	46.80	51.80	22.41	22.68	50.23	42.43	**20.34**	**<u>18.93</u>**
EMD	121.15	116.47	31.08	30.83	110.53	104.13	25.15	24.25	118.54	113.54	23.99	23.22	108.16	95.26	**20.30**	**<u>18.91</u>**
KLD	9.11	9.07	**5.36**	**<u>5.33</u>**	9.01	8.92	7.88	7.75	9.07	9.05	7.14	7.04	8.99	9.07	6.47	6.28
CS_{25}	37.84	36.94	75.68	75.68	60.36	47.75	67.57	72.07	56.76	45.05	75.68	72.97	54.95	47.75	**81.08**	**81.08**
CS_{50}	49.55	54.05	82.88	82.88	74.77	65.77	83.78	90.09	70.27	63.96	85.59	85.59	69.37	68.47	**91.89**	**91.89**

Table 2. Summary of the results. The dating performance is shown for the appearance (*AM*) and neural network (*CNN*) features, and their combination at feature level (*AN∪CNN*) and decision level (*AM+CNN*). The results are shown for both voting and Bayesian aggregation, with flat (*fl.*) and empirical (*emp.*) priors.

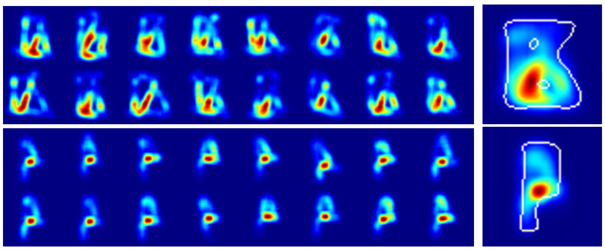


Fig. 10. Importance map of grapheme parts as measured by the dating CNN response. *Left:* individual importance maps for 16 graphemes that cause the strongest response. *Right:* average map with grapheme outline superimposed. *Odd rows:* response to ⩽1300, *even rows:* >1300.

Study	MAE	CS_{25}	CS_{50}
[7]	35.4	63.5%	(≈ 85%)*
[9]	20.9	77.5	88.5%
[14]	20.5	—	—
[8]	14.1/41.0‡	74.3%/53.3%	(≈ 80%/70%)
[11]	15.9	85.4%	(≈ 90%)
Our	18.9	81.1%	91.9%

Table 3. Comparison with other studies. $^{(\star)}$Numbers in brackets are not given in the paper, but estimated from the plots.

(our BBMs are orders of magnitude shorter) and the differences in time scales. Our method performs better than [7, 9, 14] both in terms of MAE and CS. The MAE in [8] (14.1) and [11] (15.9) are lower than our best result (18.9), but our CS scores are substantially better that [8], indicating higher reliability. Further, results in [8] drastically drop to MAE=41.0 (‡) when they apply dating across cities. Only the study [11] surpasses our result in MAE and CS_{25}, but we note that their range of dates is narrower and random guess in [8, 11] yields MAE=85.3 (in our case random guess MAE=124.2 years), therefore it is more meaningful to compare the results as percentages of the random MAE: scaled to our range, the results of [8, 11] would yield 20.5 and 23.2 years respectively — thus the relative accuracy with our method is still better.

Given that the success in estimating age depends on the amount of available evidence (graphemes), we plot the MAE as a function of manuscript length in Figure 9 (right). It is also prudent to visualise the separability of classes by plotting the embedding of feature vectors into a low-dimensional space. We accomplish this with t-SNE algorithm [19]. Figure 11 shows the embeddings for AM and CNN features (grapheme "в"). At least in this example, CNN features appear more clearly separable. Importantly, the different topologies of the embeddings suggest that the AM and CNN features capture different aspects of graphemes' appearance and are complementary. Further, having fine-tuned the CNNs on binary dating task, we can explore what features in the graphemes contribute to the network's response, thus potentially revealing human-interpretable features. We have done so by systematically occluding parts of a grapheme image with a 8 × 8 black box, and measuring the change in activations in the final classification layer [23]. We show the results in Fig. 10 (for the 16 samples that produce the strongest responses *(left)*, and the average *(right)*). Remarkably, the most significant areas approximately correspond to to the location of some of the features in paleographic tables [22], *e.g.* a triangle-shaped loop in "ρ" is characteristic of the later period, and for "в" the connection of the diagonal stroke to the vertical stem.

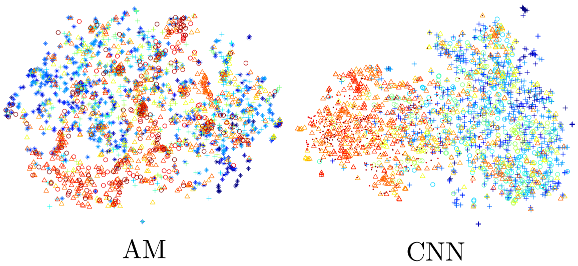


Fig. 11. Embeddings of the CNN and AM features in $\mathbb{R}^2$ using t-SNE [19] (grapheme "в"). Colour indicates age.

Conclusion We have investigated paleographic dating using an elastic model of graphemes' deformation, features learned by CNNs, and Bayesian aggregation of results from an ensemble of specially tuned classifiers. The experimental results clearly demonstrate the efficiency of our approach (MAE=18.93 years). Our method may corroborate and refine the existing paleographic analysis by human experts.

4. Bibliography

[1] "Birchbark literacy from Medieval Rus: Contents and contexts (digital archive)," http://gramoty.ru.

[2] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *ICCV*, Nov 2011, pp. 2252–2259.

[3] T. F. Cootes *et al.*, "Groupwise construction of appearance models using piece-wise affine deformations," in *BMVC*, 2005, pp. 879–888.

[4] T. F. Cootes *et al.*, "Computing accurate correspondences across groups of images," *IEEE PAMI*, vol. 32, no. 11, pp. 1994–2005, Nov 2010.

[5] T. F. Cootes *et al.*, "Active appearance models," *IEEE PAMI*, vol. 23, no. 6, pp. 681–685, 2001.

[6] J. Gu *et al.*, "Recent advances in convolutional neural networks," *CoRR*, vol. abs/1512.07108, 2015.

[7] S. He *et al.*, "Towards style-based dating of historical documents," in *ICFHR*, Sept 2014, pp. 265–270.

[8] S. He *et al.*, "Discovering visual element evolutions for historical document dating," in *ICFHR*, Oct 2016, pp. 7–12.

[9] S. He and L. Schomaker, "A polar stroke descriptor for classification of historical documents," in *ICDAR*, Aug 2015, pp. 6–10.

[10] S. He *et al.*, "Historical document dating using unsupervised attribute learning," in *DAS*. IEEE Computer Society, 2016, pp. 36–41.

[11] S. He *et al.*, "A multiple-label guided clustering algorithm for historical document dating and localization," *IEEE Trans. Image Processing*, vol. 25, no. 11, pp. 5252–5265, 2016.

[12] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

[13] Y. LeCun *et al.*, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] Y. Li *et al.*, "Publication date estimation for printed historical documents using convolutional neural networks," in *Proc. 3rd Int. Workshop on Historical Document Imaging and Processing*, ser. HIP '15, 2015, pp. 99–106.

[15] K. Sidorov *et al.*, "An efficient stochastic approach to groupwise non-rigid image registration," in *CVPR*, 2009, pp. 2208–2213.

[16] X. Song and A. Myronenko, "Point set registration: Coherent point drift," *IEEE PAMI*, vol. 32, pp. 2262–2275, 2010.

[17] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[18] T. Tommasi *et al.*, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *CVPR*, 2010, pp. 3081–3088.

[19] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[20] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *ACM Int. Conf. on Multimedia*, 2015.

[21] F. Wahlberg *et al.*, "Large scale style based dating of medieval manuscripts," in *IWHDIP*, 2015, pp. 107–114.

[22] V. L. Yanin and A. A. Zaliznyal, *Novorodskiye gramoty na bereste (iz raskopok 1990–1996 godov).* Moscow: Nauka, 2000.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.