ICDAR 2017 Competition on the Classification of Medieval Handwritings in Latin Script

Florence Cloppet
Lipade
Paris Descartes University
Paris, France
florence.cloppet@parisdescartes.fr

Véronique Eglin
LIRIS UMR5205
Université de Lyon - CNRS – INSA LYON
Lyon, France
veronique.eglin@insa-lyon.fr

Marlène Helias-Baron
IRHT
CNRS
Paris, France
m.helias@irht.cnrs.fr

Cuong Kieu
Lipade
Paris Descartes University
Paris, France
van-cuong.kieu@parisdescartes.fr

Dominique Stutzmann
IRHT
CNRS
Paris, France
dominique.stutzmann@irht.cnrs.fr

Nicole Vincent
Lipade
Paris Descartes University
Paris, France
nicole.vincent@parisdescartes.fr

***Abstract*—This paper presents the results of the ICDAR2017 Competition on the Classification of Medieval Handwritings in Latin Script (CLaMM), jointly organized by Computer Scientists and Humanists (paleographers). This work follows a competition at ICFHR2016 and aims at providing a rich annotated database of European medieval manuscripts to the community on Handwriting Analysis and Recognition. We proposed four independent classification tasks which attracted 10 registered teams, with 6 submitted classifiers from 4 participants. Those classifiers are trained on a set of 3540 images with their ground truths. In task 1 (Script classification) and task 3 (Date classification), the classifiers have been evaluated by a test set of 2000 greyscale, tiff, 300 dpi images. In task 2 (Script classification) and task 4 (Date classification), the test set consists of 1000 images in different formats, resolutions and color representation. The best scores are respectively 85.2% for task 1, 76.5% for task 2, 59% for task 3, and 49.9% for task 4. An analysis based on the matrix of confusion of each classifier is also given.**

Keywords— Historical documents; Image classification; Quantitative analysis; Feature extraction; Character recognition; Medieval Latin script classification

I. INTRODUCTION

Automated analysis and classification of handwritings by script and date, applied to the written production of the European Middle Ages, is a challenge for Computer Science, Humanities and Cultural Heritage institutions. Digital libraries contain literally ten-thousands of digitized manuscripts of the European Middle Ages. The overwhelming majority of manuscripts in there are written in Latin script, and the digital libraries are growing, with often scarce metadata on script identification and date. Some examples: more than 16,000 fully digitized manuscripts in Gallica (http://gallica.bnf.fr), the digital library of the French National Library; 14,220 in DVL DigiVatLib (http://digi.vatlib.it/mss/), the digital library of the Vatican Library; 13,300 in Manuscripta Mediaevalia (http://www.manuscripta-mediaevalia.de), 8,765 in BVMM (http://bvmm.irht.cnrs.fr/), for the largest ones.

In this context, there is a need for an automated "tagging" or "cataloguing" of the handwriting on the images. Following the first Competition on the Classification of Medieval Handwritings in Latin Script at ICFHR 2016 [1], the present one adds the challenge of dating script to the one of script classification.

Automated classification should not only allow for historical research (when and how which text is written), but is also a pre-requisite for handwritten text recognition (HTR) or automated indexing and data mining. To perform HTR on the digitized manuscripts, one "numerical model" is necessary to recognize the text for each script type and the identification of the script type is the first step. The rest of the paper is organized as follows. Section II provides an overview of the competition, the datasets and the evaluation strategies. Section III presents briefly the participating systems. The evaluated results are presented in section IV and announces the winner. Section V concludes the paper.

II. OVERVIEW OF THE COMPETITION

In this competition, the training dataset and the test dataset for task 1 (Script classification) and 3 (Date classification) encompass script images in only one format. In tasks 2 (Script classification) and 4 (Date classification), the test dataset consists of heterogeneously encoded data.

A. Dataset

The training and test datasets for tasks 1 and 3 consists of grey-level images in TIFF format at 300 dpi (compared to the original pictured manuscripts), picturing a 100 x 150 mm part of a manuscript. The test data-set for tasks 2 and 4 consists of heterogeneously encoded data that is to say, data in different formats (JPEG/TIFF), resolutions (300/400 dpi, but on reproduction at different scales) and color representation (grey-level and color images). The sets for training, tasks 1 and 3,

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.224

and tasks 2 and 4 consist of, respectively, 3540, 2000, and 1000 images. The images are drawn from the photographic material of the French catalogues of dated and datable manuscripts[2], increased with documentation BVMM and Gallica. The images of the training set are tagged according to 12 labels for the Script classification. The classes are based on morphological differences, as defined in standard works on Latin scripts[2], [3].

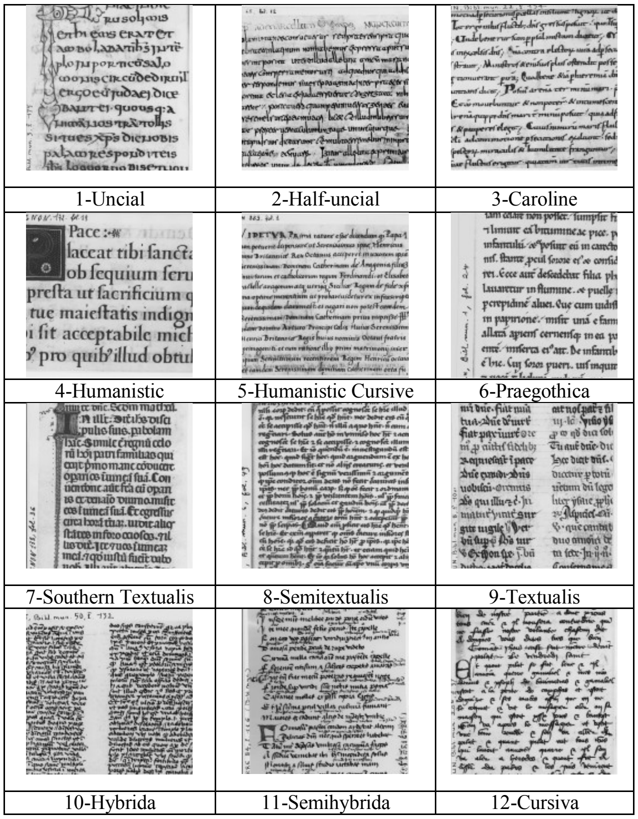


Fig. 1. Examples of the twelve concerned medieval script types

The images of the training set are tagged according to 15 labels for the Manuscript dating.

TABLE I. CLASSES FOR MANUSCRIPT DATING

1	before the year 1000 C.E.	9	between 1426 & 1450 C.E.
2	between 1001 & 1100 C.E.	10	between 1451 & 1475 C.E.
3	between 1101 & 1200 C.E.	11	between 1476 & 1500 C.E.
4	between 1201 & 1250 C.E.	12	between 1501 & 1525 C.E.
5	between 1251 & 1300 C.E.	13	between 1526 & 1550 C.E.
6	between 1301 & 1350 C.E.	14	between 1551 & 1575 C.E.
7	between 1351 & 1400 C.E.	15	between 1576 & 1600 C.E.
8	between 1401 & 1425 C.E.		

TABLE II. DISTRIBUTION OF TRAINING DATA SET IMAGES ACCORDING TO THEIR SCRIPT AND DATE

	Date class															
Script class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Total**
1	211															**356**
2	200															**395**
3	258	59	39													**200**
4								3	18	102	29	26	16	6	7	**207**
5								1	6	56	26	61	32	20	15	**217**
6	2	19	373	15		2										**341**
7				2	23	64	39	15	4	25	22	6	1		1	**411**
8			2	19	74	66	24	10	28	52	10	5	1		2	**265**
9			2	88	86	77	42	35	19	37	21	13	10	7	5	**293**
10			1	1			7	11	66	181	38	25	8	2	1	**202**
11					4	9	3	6	27	131	65	9	9	2		**442**
12					5	15	69	88	58	91	51	13	3	2		**211**
Total	**671**	**78**	**417**	**125**	**192**	**233**	**184**	**169**	**226**	**675**	**262**	**158**	**80**	**39**	**31**	**3540**

TABLE III. DISTRIBUTION OF TEST DATA SET IMAGES ACCORDING TO THEIR SCRIPT AND DATE FOR TASKS 1 AND 3

	Date class															
Script class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Total**
1	40															**40**
2	75	4														**79**
3	111	184	6													**301**
4									3	29	1	26	11	3		**73**
5								1	2	16	8	2	13	15	2	**59**
6			241	9												**250**
7				1		10	15	1	1	5		7	2			**42**
8				3	13	15	3	1	5	12	5					**57**
9			4	45	64	92	41	18	23	37	15	13	5	1		**358**
10				1		2	2	12	42	101	40	23	7	1		**231**
11		1				2		5	17	64	47	16	6	2	1	**161**
12					2	7	51	67	58	88	42	19	11	1	3	**349**
Total	**226**	**189**	**251**	**59**	**79**	**128**	**112**	**105**	**151**	**352**	**158**	**106**	**55**	**23**	**6**	**2000**

TABLE IV. DISTRIBUTION OF TEST DATA SET IMAGES ACCORDING TO THEIR SCRIPT AND DATE FOR TASKS 2 AND 4

	Date class															
Script class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Total**
1	13															**13**
2	30															**30**
3	61	90														**151**
4									6	1	1	1				**9**
5									2	7	5	3	1	2	1	**21**
6			116	14												**130**
7					1	3	7	3			3					**17**
8				6	23	15	3	1	1	3	2					**54**
9				80	65	76	48	24	15	15	9	3	6	1		**342**
10					4	2	1	6	9	22	7	3	3		1	**58**
11							2	3	4	15	5	1	2			**32**
12					1	8	34	30	21	23	9	8	6	1	2	**143**
Total	**104**	**90**	**116**	**100**	**94**	**104**	**95**	**67**	**58**	**86**	**41**	**19**	**18**	**4**	**4**	**1000**

B. *Evaluation Strategies*

The participants had to provide for each task (1) a “distance matrix” of the images of the dataset; (2) a “belonging matrix” of the images of the dataset describing the degree to which each image I$_i$, $i \in \{1..N\}$, is associated with a particular label with each of the labels.

The ranking is based on the average accuracy (see equation (1)) evaluated on the complete test dataset: the label with the highest weight denoted DL_i (*Decision Label)* for the image I$_i$ is compared to its unique ground-truth label denoted GTL_i. The average accuracy is then measured as follows with the Kronecker delta δ (equal to 1 if $DL_i = GTL_i$):

$$A \quad = \frac{\sum_{i=1}^{N} \delta_{(DL_i)(GTL_i)}}{\mathrm{N}} \qquad (1)$$

To further analyze the behavior of the systems, figure 2 displays the confusion matrices, whose rows are associated with ground truth classes, and columns with system decisions. The color of the patches expresses the percentage of the GT writings well classified by the systems. Ideally the matrix should be white with black elements on the diagonal. The gray values that appear in other places express the ambiguities that may appear between the classes.

III. SUBMITTED SYSTEMS

The competition has been organized as follows. There were an overall of 6 registered systems (provided by individuals or two-person teams): one system only handles with task 1 (and none of the three others), two with all four proposed tasks, and one system only designed for tasks 1 and 3. All teams or individuals provided a “distance matrix” and a “belonging matrix” as described in section II for each respective performed task. The competition did not require submission of executable or source code. Here we briefly present the six named classification propositions: S2-CCMV (for task 1 only), P-CNN (for tasks 1 and 3), TDeepCNN, CK1, CK2, CK3 (for the four tasks), including their basic methodological fundaments. In the competition, three propositions lie on Deep Convolutional Neural Networks architectures (CK1, CK3, T-DeepCNN).

A. T-DeepCNN System

The ***T-DeepCNN*** System lies on Deep CNNs that employ residual learning [4] and batch normalization [5]. The architecture of each network is that of the 50-layer network in [4], but with fewer convolution filters employed at each layer. CNNs are trained to classify 227x227 subwindows into one of 12 *Script* classes or into one of 15 *Date* classes. The CNNs are trained for each task independently to the others. To construct a training set, first all images are downsampled by a factor of 2. Then, 256x256 subwindows are densely extracted from each of the downsampled images at a stride of 42x42 pixels. They are randomly resized (preserving square dimensions) to have a side length in [227,285]. Let's notice that the random 227x227 crop is applied as on-the-fly preprocessing of the 256x256 input sub-windows that are linearly scaled to the range [-0.5,0.5]. To improve the system performance (from 1 to 2%), stochastic transformations are also applied to each subwindow during training phase only. They consist in a modification of the intensity of foreground or background pixels by adding a value drawn from a Normal distribution, and to the application of either a horizontal or vertical shearing with a shear angle uniformly chosen from [0,20]. The label for each subwindow is the script class of the image from which it was extracted. The standard training protocol is based on stochastic Gradient Descent Optimization for 350,000 data mini-batches of 40 randomly drawn instances, an initial learning rate (initial rate = 0.01) divided by 10 every 80,000 mini-batches, a L2 Weight Decay regularization of 0.0005, a Momentum of 0.9.

At test time, color images are converted to grayscale and the large manuscript image is cut up into overlapping 227x227 subwindows at a stride of 100x100 pixels and each is classified. The resulting classification is then the average prediction of each subwindow. For improved accuracy, predictions are also averaged over an ensemble of CNNs. The deep learning libary Caffe was used to train the CNNs.

For ensemble predictions, each CNN computes a prediction (averaged over the subwindows extracted at the appropriated scale) and these predictions are uniformly averaged over all CNNs in the ensemble. A number of 5 networks in the ensemble was chosen.

The distance matrix and belonging matrix are derived from the average prediction of the CNN ensemble, which is a probability distribution over the types (either script or dates). The belonging matrix is precisely these predictions and the distance matrix is computed as the pairwise Euclidean distance between the prediction vectors. Predictions are differently performed on JPG images and on TIFF images. JPG images are cropped to be 70% of the original dimensions. JPG images are then evaluated on a separate ensemble of networks that are identically trained.

B. S2-CCMV System

The ***S2-CCMV*** System is a connected component based system. Authors focus on the most current ductus properties excluding material properties, like in [6]. Their proposition cannot be seen as a black box system which only provides answers but must also be interpretable. As a consequence, their system is based on color, luminance and geometrical simple features. In a first step, connected components (CCs) are extracted from a binary version of the document image to be classified. Authors use a global threshold on Difference-of-Gaussian (DOG)-enhanced document image to binarize the images [7]. The settings of this method partially depend on the size of the content. After the first thresholding step, they extract the CCs, compute their median height m_H and then scale the image by a factor of 64/ m_H, so as to consider the median character height to be approximately of 64 pixels. Then they filter the characters CCs by rejecting the ones which differ too much from the median CCs height. The patterns are created by applying one k-means clustering on the CCs belonging to each script class providing one set of centroids per class that are used as model patterns describing the base-script of each class. As distance measurement, they use the simple Euclidean distance on the pixels to compute the class probability of each pattern. This is done by going through all CCs of the training data and attributing them to their closest model pattern (regardless of the base classes). For each model pattern, they count how many CCs are attributed to it, and how many from each script class, and then can estimate a probability of belonging to each script class. The combination of a model pattern and a set of class probabilities is what they call model. The CCs are then filtered using three kinds of criterion. First, if the model has been matched less than N times (N=50) in the training data, then they consider it as unreliable and the CC is rejected. Second, if the most probable class of the model does not correspond to the base class of the model, then they consider that the model pattern does not well represent the base class and the CC is rejected. Third, based on the confusion matrix obtained on 300 test pages selected from the training set, they select some models whose matching CC should be ignored. Finally, a majority voting on the CC classification results is used to attribute a class label to the whole page. They took out of their training data random 300 pages and used them to have an evaluation of their method on unseen data. For the competition need, the test pages will refer to the 2000 provided test pages.

C. The three CK Systems

CK1. This method relies on a convolutional neural network architecture for image classification. In particular, authors use a variation of the well-known Residual Network architecture [4], but of a more modest depth (and with less max pooling in the early layers). In the previous edition of the competition in 2016 [1], it became clear that the manual annotators took classification decisions at a very local level (i.e. parts of specific letter) whereas the proposed network seemed much more sensitive to more global layout features. As a consequence, authors hypothesize that the use of residual connections will allow a more efficient information flow from low-level, local characteristics to the higher, global layers in the network. Additionally, to stay away from layout information, a global averaging layer [10] is applied to the output of the convolutional stack of the network (instead of e.g. an affine layer, which also needs much more parameters). Finally, two label sets for each training image are accessed, i.e. script type and date. Instead of training a different model for both prediction tasks, a single multi-headed architecture is used, where two independent dense layers, one for each prediction task, are laid on top of the global average layer (followed by a regular softmax). The training objective is to minimize the sum of the categorical cross-entropy for both tasks simultaneously, hoping to leverage useful information from both tasks when optimizing the convolutional filters.

All images (both at training and prediction time) are first turned into greyscale images and their rather generous resolution was downscaled by a factor of two. The network is fed randomly with sampled crops or patches from the original images (150x150 pixels). During training, the data are augmented, through the introduction of small variations in the zoom factor, shearing, rotation and translation of the patches. At the beginning of each epoch, ten augmented patches are randomly sampled from each image. At prediction time, fifty patches are selected from each test image, which are not augmented, since this does not yield a clear added value. The final classification and probabilities are averaged over the patches for each to obtain the final probability vector. Training accuracy/loss are then calculated at the individual patch level and validation accuracy is monitored at both the image and patch level. The model settings which yield the highest average patch accuracy on the validation set (averaged for the script and date classification) are finally used.

The network is trained using standard mini-batch stochastic gradient descent, with a batch size of 30 (larger batches were much slower to converge) and an initial learning rate of 0.01 (which was decimated every 20 epochs) and Nesterov momentum. For development, authors make use of a 90%-10% train-test-split at the image level; this split was stratified at the level of script/date label combinations (since they trained for both tasks at the same time). Most models are trained for around 45 epochs. The use of batch normalization seems especially promising to deal with the 'heterogenous' data sets in both tasks, since it should take care of some of the differences in preprocessing. Compared to the previous DeepScript edition of the script [1], the current architecture includes several improvements: residual connections, batch normalization, global average pooling, an increased depth and the use of a multi-headed network.

CK2. This approach is based on successful writer identification methods [8, 9]. It represents a traditional bag-of-(visual)-words approach, based on local descriptors that are encoded via a background model to form a global descriptor. RootSIFT descriptors, a variant of SIFT in which the SIFT descriptors are normalized using the Hellinger kernel, are used as local descriptors. Since script is typically not rotated, they ignore the rotational-invariance property of SIFT and set the rotation angle to 0. The RootSIFT descriptors are PCA-whitened and dimensionality reduced to 64 components. They use Vectors of Locally Aggregated Descriptors (VLAD) as encoding method. It relies on the first order statistics computed from a background model. For the background model, they rely on a mini-batch version of k-means. The VLAD encoding is subsequently normalized using intra-normalization, i.e., a component-wise L2-normalization, followed by a global L2-normalization. To further decorrelate the global descriptors, they use multiple background models by using different subsets of the trainings-data for k-means and compute for each background model a VLAD descriptor. Finally, all VLAD descriptors are jointly decorrelated using PCA-whitening. This vector is then classified using linear SVMs with a squared-hinge loss function. Note that for the second test set, they use only an inner crop of 2000x2000 pixels to avoid the most of the background and to limit the number of keypoints.

CK3. The last method is variation of CK1 where the 1024 dimensional CNN activation features of the penultimate layer are simply aggregated and then L2- normalized, these features are power normalized and classified with a linear SVM.

D. The P-CNN System

The *P-CNN* system is based on Convolutional Neural Networks (CNNs). Initially, it relies on a ResNet152 network with weights that are trained on the ImageNet dataset for the ImageNet Large Scale Visual Recognition Competition (ILSVRC). The final classification layer of this network is discarded, and replaced with a new fully connected layer with 12 outputs (tasks 1 and 3). This modified network is then trained end-to-end on the script classification task. For training, the complete dataset is partitioned into train and validation splits randomly. Then the modified ResNet152 is trained on the train split of the dataset. During training, a random crop of 256x256 is extracted from each input image after resizing the input image such that smaller edge corresponds to 768 pixels, and the other edge is resized such that the aspect ratio is maintained. The resizing process is performed using bilinear interpolation. This training process is repeated for 90 epochs, using the Adam optimizer with a learning rate of 0.002.

During evaluation time, each test image is resized in a similar manner as to the training images, and 10 random crops (of size 256x256) are obtained from each test image. The network then classifies each crop, and the average of all classification over all crops is used as the classification for the test image. For computing the distance between all inputs, the activations of the second-last layer are extracted. This produces a 2048 dimensional vector for each input image. The pairwise cosine distances are then computed between all such vectors. The training and evaluation procedure is identical for both Task 1 and Task 3, with the difference only in the final fully connected layer of the CNN.

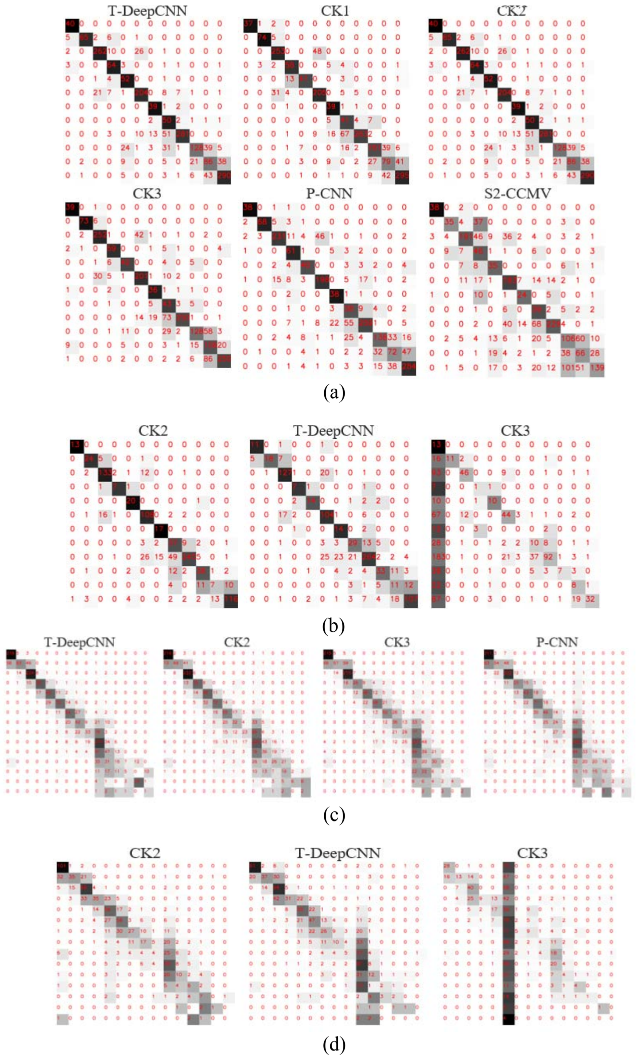
IV. Experiments And Evaluation

Six systems (one, corresponding to CK1, after the deadline) were submitted for task 1, which is equivalent to ICFHR2016 CLAMM competition task 1 [1]. The winner for task1 is T-DeepCNN system that yields an accuracy of 85,2%, which is superior to the winning 83,9% of [1]. Four systems range from 56.55 to 78.05%. Accuracy on task 3, performed on the same data set, is much lower, with a maximum of 59%. The automated classification on scripts and dates can be successfully applied for manuscript cataloguing as a first step for HTR; dating is, however, still a more challenging task.

For tasks 2 and 4, three systems were run. These tasks evaluate the robustness of the systems regarding formats and color. The winner is CK2, with an accuracy of 76,5% (task 2) and 49.9% (task 4). The decrease in accuracy between task 1 and 2, resp. 3 and 4, ranges from 1.5 % to 46.7%, resp. 5% to 30%.

TABLE V. GLOBAL ACCURACY (GA) IN PERCENTAGE AND RANKING (R) FOR TASKS 1 TO 4 (T1, T2, T3, T4)

	Results							
	GA T1	R T1	GA T2	R T2	GA T3	R T3	GA T4	R T4
S2-CCMV	56.55	5						
T-DeepCNN	**85.20**	1	73.90	2	59	1	48	2
P-CNN	73.25	4			49.80	4		
CK1	*77.30*		*29.20*		*54.15*		*21.30*	
CK2	78.05	2	**76.50**	1	54.85	2	**49.90**	1
CK3	74.05	3	27.40	3	52.55	3	22.60	3



(a)

(b)

(c)

(d)

Fig. 2. Confusion Matrices for tasks 1 (a), 2 (b), 3 (c) and 4 (d)

On graphs of figures 3 and 4, the accuracy per class is displayed with, on the top, the results based on images of the same format, and, on the bottom, based on heterogeneously encoded images. The doted lines indicate the general accuracy (GA). First, it can be noticed, that nearly all methods have difficulties on the same classes for different tasks. The graph relative to tasks 2 and 4 (on the bottom of the figures 3 and 4) illustrates the generalization capability of the methods. The observed variations between results of tasks 1 and 2, resp. tasks 3 and 4, can be explained by the training strategy (learning by heart and overfitting) or by the used characteristics that are probably not well adapted to all the variability that is present in the images of task 2. The peaks of class 10 and class 11 in figure 4 correspond to the tendency of all systems to ascribe too many images to these classes, including the expected ones. Class 15 is scarcely represented in the test sets, which may explain the corresponding lesser results.

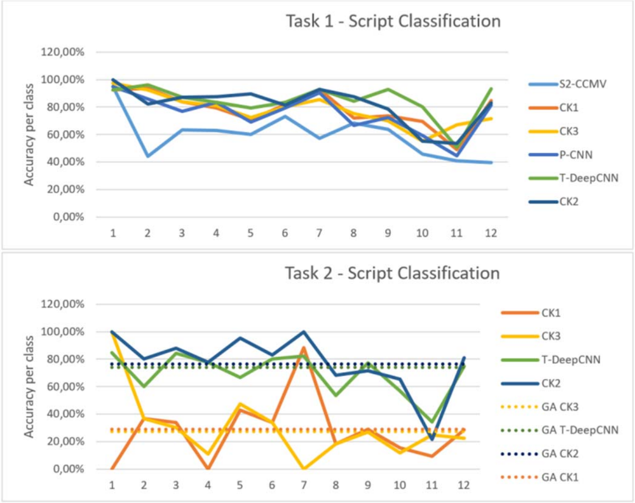


Fig. 3. Accuracy per class for tasks 1 and 2

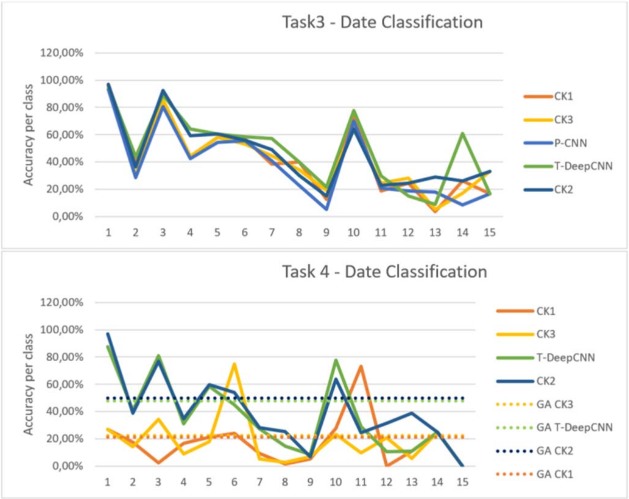


Fig. 4. Accuracy per class for tasks 3 and 4

The results given by the best performing systems can be evaluated not only from an image analysis point of view, but also towards a historical interpretation (according to cultural and chronological proximity of script classes). As for script classes (tasks 1 and 3), the systems integrate *de facto* some pre-assumptions, methods, features, and historical conclusions of paleographers, which are embedded within the training dataset, but not all beyond debate [2]. The outputs cannot aim at revising the extant classification, but they can be evaluated along their heuristic efficiency.

As for dates, the historical question is to determine which of script type and date is the more distinctive feature. The lower accuracy probably relates to the extended period of some scripts used during several centuries as well as to the contemporaneous use of several script types in some periods. Systems were trained on date and script type either independently or together. It does not seem to affect the accuracy significantly.

Confirming the results of [1], the confusion matrices (figure 2) appear all but random from a historical perspective. Unlike in [1], the historically related script types Uncial, Semi-Uncial and Caroline scripts are clearly separate, and even the confusions occur mainly on later Semi-Uncial scripts used in a Caroline context. Like in [1], Humanistic and Humanistic cursive are sparsely intermingled by all systems, but are very clearly separated from all other classes, however surprising.

Much more than in [1], Caroline and Praegothica are intermingled. Praegothica emerged from Caroline and these scripts are very close at the beginning of the 12th century and their features can be confused. This greater confusion may be linked to the training data set, in which many more samples have been provided, illustrating the full diversity of these script types.

All systems confirm the difficulty in distinguishing Praegothica, Southern Textualis, Semitextualis, and Textualis, as in [1], but a new phenomenon can be seen in T-DeepCNN: those scripts are rather attributed to Textualis than the reverse. It may be a hint at the great variability of Textualis, due to its long life from the 13th to the 16th century, which can perhaps easier accommodate samples of other script types than other, more homogeneous classes.

The clear distinction between Textualis and Southern Textualis confirms the conclusion of [1] on the more direct phylogenetic link from Praegothica to Textualis and a derivation from Textualis to Southern Textualis. Semitextualis also overlaps towards Hybrida, which reflects the inclusion in the training and test sets of more samples proving the existence of script types mixing the *tradizione corsiva* and the *tradizione libraria* [11]. The gathering of Hybrida, Semihybrida, and Cursiva, and additional connections from Hybrida towards Semitextualis is confirmed as in [1], with the same directions in the confusion. For the script dating (tasks 3 and 4), the lower accuracy reflects the historical evolution and continuum in which discretizing best performing system (T-DeepCNN) tends to predate more than postdate. The confusions are more and more important for the date-classes 10-15 (from 1451 to 1600). Several factors probably explain that scripts from date classes 11-15 are largely labelled in class 10, esp. less training and test data; classes corresponding to shorter time periods. Moreover, after 1450, script types in use are more diverse (several Gothic and Humanistic script types), with Hybrida and Cursiva competing as formal scripts [12].

V. Conclusion

Given an image of a medieval manuscript page, several systems are able to classify the script type with a reasonable accuracy (more than 80% in task 1, ca. 60% in task 3). The aims of cataloguing and classifying for HTR are achievable. As for dating, the accuracy still needs to be improved (less than 80% in task 2, less than 50% in task 4).

The confusion matrices allow to analyze the errors or dispersion of the results, which are largely to be explained by the overall historical and graphical connection between script types. In this regard, the methods reflect accurately the paleographical classifications and the historical evolution. Further improvements and a cross-correlation of the results given for script types and dates appear to be potentially relevant also for historical analysis.

References

[1] F. Cloppet, V. Eglin, V. C. Kieu, D. Stutzmann, N. Vincent, “ICFHR2016 Competition on the Classification of Medieval Handwritings in Latin Script”, Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016.

[2] D. Stutzmann, “Clustering of medieval scripts through computer image analysis: towards an evaluation protocol”, Digital Medievalist Journal, vol. 10, 2015.

[3] B. Bischoff, Paläographie des römischen Altertums und des abendländischen Mittelalters. Berlin: Erich Schmidt, 1986.

[4] K. He, X. Zhang, S. Ren, et J. Sun, “Deep Residual Learning for Image Recognition”, ArXiv151203385 Cs, déc. 2015.

[5] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ArXiv150203167 Cs, févr. 2015.

[6] S. He, P. Samara, J. Burgers, and L. Schomaker, “Historical manuscript dating based on temporal pattern codebook,” Computer Vision and Image Understanding, vol. 152, pp. 167–175, 2016.

[7] A. Fischer, E. Indermuehle, H. Bunke, G. Viehhauser, and M. Stolz,“Ground Truth Creation for Handwriting Recognition in Historical Documents,” in Int. Ws. on Document Analysis Systems, pp. 3–10, 2010.

[8] V. Christlein, D. Bernecker, and E. Angelopoulou. Writer identification using VLAD encoded contour-Zernike moments. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, Nancy. Aug 2015, pp. 906–910.

[9] V. Christlein, D. Bernecker, F. Hönig, A. Maier, and E. Angelopoulou. Writer Identification Using GMM Supervectors and Exemplar-SVMs. Pattern Recognition, vol. 63, pp. 258–267, 2017.

[10] B. Zhou et al. (2016). Learning Deep Features for Discriminative Localization. CVPR’16 .

[11] E. Casamassima. Tradizione corsiva e tradizione libraria nella scrittura latina del Medioevo. Roma: Gela, 1988.

[12] D. Stutzmann, “Les écritures des livres d’heures dans l’espace français (1290-1550),” in ‘Change’ in medieval and Renaissance scripts and manuscripts, E. Overgaauw and M. Schubert, Eds. Turnhout: Brepols, 2017.