Date Recognition in Historical Parish Records

Laura Cabello Piqueras[1(✉)], Constanza Fierro[1], Jonas F. Lotz[1,2], Phillip Rust[1], Joen Rommedahl[3], Jeppe Klok Due[3], Christian Igel[1], Desmond Elliott[1], Carsten B. Pedersen[4], Israfel Salazar[5], and Anders Søgaard[1]

[1] University of Copenhagen, Copenhagen, Denmark
{lcp,c.fierro,jonasf.lotz,p.rust,igel,de,soegaard}@di.ku.dk
[2] ROCKWOOL Foundation, Copenhagen, Denmark
[3] The Danish National Archives, Copenhagen, Denmark
{jro,jkd}@sa.dk
[4] Centre for Integrated Register-based Research, Aarhus University, Aarhus, Denmark
cbp@econ.au.dk
[5] Université Paris-Saclay, Gif-sur-Yvette, France
israfel.salazar@ens-paris-saclay.fr

Abstract. In Northern Europe, parish records provide centuries of lineage information, useful not only for settling inheritance disputes, but also for studying hereditary diseases, social mobility, etc. The key information to extract from scans of parish records to obtain lineage information is dates: birth dates (of children and their parents) and dates of baptisms. We present a new dataset of birth dates from Danish parish records and use it to benchmark different approaches to handwritten date recognition, some based on classification and some based on transduction. We evaluate these approaches across several experimental protocols and different segmentation strategies. A state-of-the-art transformer-based transduction model exhibits lower error rates than image classifiers in most scenarios. The image classifiers can nevertheless offer a compelling trade-off in terms of accuracy and computational resource requirements.

Keywords: Handwriting recognition · Parish records · Transfer learning · Robustness

1 Introduction

In 1968, the Danish state put a (digital) Civil Registration System (CRS) [36] into use. This system is a national register containing basic personal details on all individuals residing in Denmark, including ancestry information. To the

Supported by Novo Nordisk Foundation (grant NNF 20SA0066568).
L. C. Piqueras, C. Fierro, J. F. Lotz, and P. Rust—Equal Contribution.

U. Porwal et al. (Eds.): ICFHR 2022, LNCS 13639, pp. 49–64, 2022.
https://doi.org/10.1007/978-3-031-21648-0_4

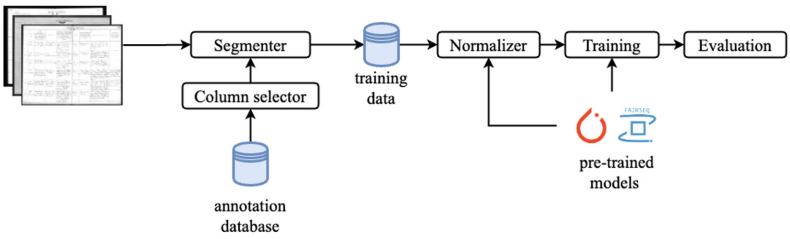


Fig. 1. Our end-to-end pipeline for handwritten date recognition. How the training data is generated depends on the segmentation strategy used in the segmentation module (Sect. 4.2) and the column of interest. Normalization of the input images depends on the chosen pre-trained model (Sect. 4.3). Evaluation (Sect. 4.4) is common across the different model architectures.

extent modern registers include civil registration numbers, the CRS enables us to study hereditary diseases, social mobility, etc., on the population that resides in Denmark after 1968. If we want to go further back in time, we will need to consult the parish records.

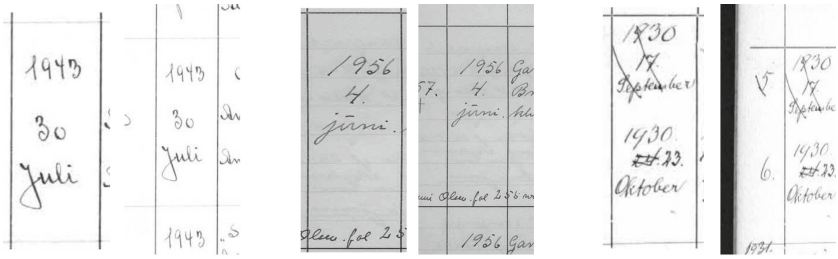
Parish records are registries handwritten in pre-printed books. Even though the layout of parish records is relatively consistent, it may vary across historical periods, regions, and countries that also established this tradition; but typically the books contain much of the same information, and there will be hundreds of thousands of records using the same format. However, since the records were handwritten by clergy, and the books were valuable, the pre-printed layouts were often used somewhat creatively, and many errors were introduced and corrected. The task of digitizing them is therefore both easy and hard at the same time. Examples of Danish parish records can be found on the Danish National Archives' website[1]. We are able to make some of our manual annotations publicly available for research purposes. See Sect. 2 for details.

The rationale for digitizing parish records is straight-forward: the parish records contain our family histories. This information is useful for studying a wide range of scientific topics; family histories can help get precision medicine off the ground [8] and are crucial to understand and prevent hereditary diseases [7]. Many of these diseases are of childhood onset [12], but there are also examples of hereditary diseases that have adult onset, e.g., breast cancer, diabetes, heart disease and blood clots, Alzheimer's disease and dementia, arthritis, depression, high blood pressure and high cholesterol [2].

2 Data

The original parish records are handwritten on pages with pre-printed layouts. The pre-printed books used and released as part of this project were digitized

[1] An example page can be found at: https://www.sa.dk/ao-soegesider/da/billedviser?epid=17125564\#167405,28108453.



(a) Easy (b) Average (c) Difficult

Fig. 2. Examples of birth date cells in the different test splits. Each test split displays the same cell segmented with our two different approaches: U-Net (left) and fix-size (right) approaches. See Sect. 4.2 for further details.

and transcribed by paid Danish transcribers. These records comprise over 8100 files, including 26000 births and 7000 deaths approximately, from which nearly 7700 files are fully annotated. They cover information gathered from 10 parishes from 1920 to 1960. In total, the Danish National Archives (Rigsarkivet) stores 20.000 parish records from 1900–1980, comprising a total of approximately 12 million entries.

The books used for registering births and deaths use a pre-printed tabular layout, although with important variations in the information reported. Common aspects between both type of books are: (a) a sequential page number at the top-right corner of each page; (b) the title, indicating the type of registry (births [Fødte] or deceased [Døde]) and the gender of the persons on that page (men [Mandkøn] or women [Kvindekøn]); (c) the table header explaining the content of each column; and (d) the subsequent handwritten registries. Among the columns, the left-most one contains the row ID in both registries that, together with the page number, uniquely identifies each row. The IDs are sequential numbers with the exception of non-integers, indicating that the entry is duplicated because the actual birth took place in another parish. Birth dates are in the second column in the new-born registries and in the sixth column in the deceased registries (the second column contains date and place of death). The remaining columns are not of interest for the task of birth date recognition[2].

The data used is made publicly available[3], to enable further research on handwritten date recognition systems. Note that the dataset released includes only the columns containing the row ID and birth date.

3 Date Recognition

We distinguish between handwritten text recognition (HTR) and optical character recognition (OCR) as closely aligned but separate tasks: OCR is for machine

[2] Future work includes the integration of other columns containing dates in our training set.

[3] github.com/coastalcph/mgr-birthdates.

Table 1. Size of the different data splits depending on the segmentation strategy. #pages is the number of different images (book pages) in each split. #cells is the total number of birth date cells.

	Fix-size		**U-Net**	
	#Pages	#Cells	#Pages	#Cells
Train	6757	28308	5383	22978
Validation	746	3159	593	2565
Test easy	53	225	44	188
Test average	58	231	47	186
Test difficult	63	217	52	181
Total	7677	32140	6119	26098

printed text where the variation in style originates from the use of different fonts. Layout analysis is often integrated into the OCR engine [30,33,49] and the characters from a given document are processed individually. HTR, on the other hand, typically attempts to decode sequences from the data at word or sentence level and segmentation is a separate step in the HTR workflow.

Our approaches to segmentation (elaborated on in Sect. 4.2) attempt to identify the row and column structure of the parish records. Arriving at a grid structure is sufficient for transcription of birth dates and serves as a good starting point for the more advanced columns (future work) where text line detection can still be applied within the region of interest. With this approach, we do not have to address an additional problem of resolving what text lines refer to what row/column. After segmentation, the cells containing birth dates are passed, together with their row ID, to the models described in Sect. 4.3 which predict the content. Figure 1 depicts the end-to-end solution adopted for handwritten date recognition.

4 Experiments

4.1 Data Splits

The scanned images are split into separate sets for training, validation, and testing. The test set is further split into categories of easy, average, and difficult based on an assessment of the level of noise in the images. The easy and difficult test splits are constructed by manually inspecting individual rows across all parishes, selecting some that are clean and high quality for the former, and some that contain corrections or are in some other way hard to read for the latter (see examples in Fig. 2). For the average test split, we randomly select 58 pages. The remaining data are shuffled and 10% are kept for validation. Note that for constructing the easy and difficult test splits, we select individual *rows* rather than entire pages because the difficulty of cells can vary within pages. When selecting only a subset of the rows in a given page for testing, we

add the remaining to the training set. Table 1 shows a summary of the final splits which, at the same time, depends on the deployed segmentation strategy (detailed below).

4.2 Segmentation

The first step in our pipeline is to segment every image into rows of the different individuals and columns with the different types of information. We utilize the tabular layout of the parish records and aim for a segmentation strategy that will return a grid of cells: each cell containing all the text lines from the corresponding record entry. The purpose of the segmentation is twofold: to identify the number of individuals registered on the given page and to locate the columns of interest. We implement two methods: a heuristic fix-size segmentation and a semantic segmentation based on a U-Net convolutional neural network architecture [45]. Figure 2 displays examples of birth date cells extracted with each of these methods. After obtaining the segmented columns for every individual, the cells containing the birth dates are fed to the text recognition models (described in Sect. 4.3) together with the annotated date as label.

Fix-Size Segmentation. The fix-size implementation takes advantage of the relatively consistent layout of the parish records. After reshaping the images to 1920×1080 pixels to align the sizes, the columns are detected based on a deterministic estimate of their horizontal offset. The horizontal image slices that split the rows are found using the number of annotations we have for each page and cropping the image into homogeneous segments. A small margin is added in each direction to include content that might be misaligned due to variation of the page position or writings close to the cell edges. See Fig. 2 (right side images) for an example.

It is important to note that this rather naïve method serves primarily as a baseline and that it does not account for any source of variation within the images. In addition, the provided annotations are not perfect and are an additional source of noise when deciding on the number of horizontal crops. To use this strategy in a real world application, we would need to train a separate model that predicts the number of rows used for cropping unlabeled images.

U-Net. We train a U-Net model to predict the boundaries of regions of interest, allowing for a more flexible segmentation that does not rely on the assumption that entries are uniformly distributed across the page. We manually annotate the cell boundaries for every row and column for 282 pages that are chosen at random from the training split. The implementation follows [45] but with ELU [9] activations instead of ReLU [31] and replacing the up-convolution with nearest neighbor up-sampling, as suggested in [37,38]. The model is trained from scratch for 1000 epochs with a batch size of 8 on 80% of the annotated pages. The remaining 20% are used for validation. During inference, we rely on the Harris corner detector [16] to obtain pixel coordinates of where the predicted row

separators and column separators intersect. As a result, the segments retrieved are precisely aligned with the cell's margin, as we can see in the images (left side) in Fig. 2.

As seen in Table 1, this segmentation approach yields less data for the subsequent steps of our HTR pipeline than the fix-size approach. This is caused by a sanity check that discards the given page when the U-Net model and the number of annotations for that page do not agree. Out of the 1558 pages that are discarded, 1176 of them (75%) contain at least one segmentation error. For 208 of the discarded pages, the model incorrectly includes cells that only contained noise, e.g. squiggles or smudges, and should have been ignored. And in 174 of the cases, the U-Net gets the layout right but the provided annotations are wrong, typically because of too few registrations.

4.3 Models

We approach the problem of handwritten date recognition from the perspective of a multi-label image classification task and as a sequence-to-sequence text generation task. We benchmark results for two image classification models, one based on ResNet [17], the other on EfficientNet [48], and for a sequence-to-sequence model based on a pre-trained TrOCR model [27], which leverages the Transformer architecture [52].

Classification Models. We experiment with ResNet-18, the shallowest ResNet variant (11M trainable parameters) presented in [17] which is able to achieve competitive accuracy in our date recognition task. We also experiment with a larger classification model, EfficientNet-B4, which among the various models presented in [48] provides a good trade-off between accuracy and FLOPs (19M trainable parameters). We download the pre-trained ResNet-18 and EfficientNet-B4 models from Torchvision[4] and fine-tune them on our training datasets—obtained with either fix-size or U-Net segmentation—for 40 epochs, validating the performance on the validation split at the end of every epoch. We train with full precision in batches of 256 samples for the ResNet-18, and 128 samples for the EfficientNet-B4. Input images are normalized to 224×224 pixels. For both architectures, we perform a grid search over a set of specified learning rates, $\{5e-4, 1e-3, 3e-3, 5e-3, 1e-2\}$, and 5 different initializations, resulting in 5×5 runs over each training set. Model checkpoints are saved every 100 steps. After training, we select the checkpoint that achieved the highest validation accuracy to compare results. Evaluation on the test data is performed with the 5 versions of the model that reported the highest validation accuracy, i.e., we test the best-performing model along with the remaining 4 random initializations trained with the same learning rate.

In addition, we explore 2 different label encodings for computing the loss: (1) *datetime encoding* where every day since January 1 1800 up until December 31 1999 is encoded as a unique class; and (2) *digit encoding* where the year is

[4] https://pytorch.org/vision/stable/models.html.

encoded as digits from 0 to 9, the month as 1 digit from 1 to 12, and the date as 2 digits from 0 to 9.

We also investigated the effect of tailoring the input dimensions and different data augmentation techniques using the ResNet-18 architecture. We tested several standard augmentation transformations as well as the same augmentations used when training the TrOCR models as described below. Neither changing the input image size (including using the same size as used in the TrOCR experiments) nor the augmentations significantly altered the ResNet-18 results.

TrOCR Models. We initialize TrOCR with a BEiT-base [3] encoder and a RoBERTa-large [28] decoder from the pre-trained `trocr-base-handwritten` checkpoint provided by [27][5]. The full TrOCR model has 334M trainable parameters, making it significantly larger than our ResNet and EfficientNet models. We preprocess training examples by converting the labels' date format from *yyyymmdd* to one that matches the Danish handwriting, e.g., "*19500331*" to "*31 marts 1950*". We then fine-tune for the text recognition task outlined in [27] on the training datasets obtained via fix-size or U-Net segmentation. Following [27], we perform data augmentation by selecting from a list of possible transformations at random with uniform probabilities: random rotation (−10 to 10°C), Gaussian blur, image dilation, image erosion (all with kernel size 3), downscaling by a factor of 3, underlining with black pixels, and keeping the original. We train for up to 300 epochs and validate model performance based on the validation loss after every epoch. To reduce computational overhead, we use early stopping [40] with a patience of 20 validation steps, i.e., if the model does not improve its validation loss within 20 epochs, training is terminated preemptively. In practice, training was typically terminated after 30–80 epochs, depending on the learning rate. We train in batches of 256 input images of size 384×384 pixels, resulting in a sequence length of 576. We use the Adam optimizer [23] and perform a grid search over peak learning rates $\{6e-6, 9e-6, 2e-5, 5e-5, 8e-5\}$. We warm up to these peak learning rates linearly from $1e-8$ over the first 500 training steps and then decay with an inverse square root schedule. Weight decay is set to $1e-4$. For each learning rate, we run 5 random initializations to account for randomness. We use automatic mixed precision (fp16) training with Nvidia Apex[6].

4.4 Evaluation Metrics

The different approaches are evaluated using accuracy. We compute both top-1 and top-5 accuracy for the full date, i.e., a classification is correct provided that day, month, and year are correct. Top-1 results are further decomposed, reporting values for the day, month and year individually. On each of the 3 test sets, we report the mean and standard deviation of the 5 random training initializations

[5] https://github.com/microsoft/unilm/tree/master/trocr#fine-tuning-and-evaluation.

[6] https://github.com/NVIDIA/apex.

Table 2. Top-1 validation set error rate of the best-performing models. TrOCR has the lowest validation error rate on the U-Net data whereas the EfficientNet models have the lowest on the (more noisy) data obtained via fix-size segmentation.

Model	**Fix-size**	**U-Net**
	Error (%)	Error (%)
ResNet-18 (1)	13.6	8.0
ResNet-18 (2)	13.8	8.0
EfficientNet (1)	10.1	7.8
EfficientNet (2)	10.2	6.2
TrOCR	12.7	6.0

that performed best on the validation set. Note that in this task, the Word Error Rate (WER) metric, commonly found in the OCR literature, would be complementary to the *full date* accuracy where any mismatch is counted as an error.

5 Results and Analysis

Looking at results from Table 3 and Table 4, we find that TrOCR achieves the highest *full date* recognition accuracy, outperforming the ResNet and EfficientNet models across all three test splits. ResNet-18 has the lowest performance. The performance gap between TrOCR and ResNet-18 is around 5–7% of accuracy. EffienctNet-B4 is approximately 1–3% worse than TrOCR, depending on the test split and label encoding. This result is in line with [47] who have shown that TrOCR outperforms Transkribus [20] on historic handwritten records in Latin. The result is also expected considering the differences in model capacity: TrOCR has 334M trainable parameters whereas EfficientNet-B4 has 19M and ResNet-18 has 11M. With these differences in mind, the EfficientNet model may provide the most satisfactory trade-off in terms of computational requirements and performance.

For the classification models, there seems to be no clear winner among the two label encodings. The difference in mean top-1 accuracy between the two encodings is largest for our ResNet-18 models on the difficult test split (3.4% accuracy using fix-size segmentation and 1.4% with U-Net segmentation)—here in favor of the *datetime* encoding—although even in this case, both encodings are within one standard deviation for the U-Net data, and the top-5 accuracy in fact favors the *digit* encoding. The EfficientNet-B4 models are less sensitive to the choice of label encoding.

We now perform an error analysis of the best-performing models, i.e., models with the highest full date validation accuracy after hyperparameter tuning on the validation data. See Table 2 for an overview of the errors. We analyse errors across two dimensions: datasets and models (which can be further split into encodings for the two image classification models).

Table 3. Mean accuracy and standard deviation of 5 random initializations each using the crops generated by the fix-size segmentation heuristic. The numbers in parentheses refer to the label encoding used in each experiment ((1): datetime encoding, (2): digit encoding), as explained in Sect. 4.3. Highest mean accuracy is highlighted in **bold**.

Model	Test	Day	Month	Year	Full date	Top5
ResNet-18 (1)	Easy	$0.934 \pm 8e-3$	$0.977 \pm 5e-3$	$0.966 \pm 6e-3$	$0.896 \pm 6e-3$	$0.955 \pm 8e-3$
ResNet-18 (2)		$0.932 \pm 1e-2$	$0.968 \pm 6e-3$	$0.972 \pm 1e-2$	$0.892 \pm 1e-2$	$0.966 \pm 6e-3$
EfficientNet-B4 (1)		$0.948 \pm 2e-3$	$0.978 \pm 2e-3$	$\mathbf{0.988} \pm 4e-3$	$0.933 \pm 2e-3$	$0.964 \pm 4e-3$
EfficientNet-B4 (2)		$0.948 \pm 1e-3$	$0.978 \pm 5e-3$	$0.984 \pm 5e-3$	$0.928 \pm 1e-2$	$0.971 \pm 3e-3$
TrOCR		$\mathbf{0.977} \pm 9e-3$	$\mathbf{0.994} \pm 2e-3$	$0.974 \pm 5e-3$	$\mathbf{0.947} \pm 7e-3$	$\mathbf{0.996} \pm 3e-3$
ResNet-18 (1)	Average	$0.903 \pm 1e-2$	$0.970 \pm 1e-3$	$0.956 \pm 4e-3$	$0.867 \pm 1e-2$	$0.938 \pm 1e-3$
ResNet-18 (2)		$0.900 \pm 9e-3$	$0.968 \pm 5e-3$	$0.954 \pm 8e-3$	$0.859 \pm 5e-3$	$0.961 \pm 6e-3$
EfficientNet-B4 (1)		$0.929 \pm 2e-3$	$0.970 \pm 2e-3$	$0.967 \pm 7e-3$	$0.897 \pm 5e-3$	$0.948 \pm 7e-3$
EfficientNet-B4 (2)		$0.925 \pm 6e-3$	$0.968 \pm 3e-3$	$0.965 \pm 6e-3$	$0.892 \pm 3e-3$	$0.965 \pm 5e-3$
TrOCR		$\mathbf{0.957} \pm 9e-3$	$\mathbf{0.990} \pm 5e-3$	$\mathbf{0.971} \pm 6e-3$	$\mathbf{0.923} \pm 9e-3$	$\mathbf{0.976} \pm 3e-3$
ResNet-18 (1)	Difficult	$0.811 \pm 1e-2$	$0.873 \pm 1e-2$	$0.892 \pm 1e-2$	$0.716 \pm 4e-3$	$0.865 \pm 1e-2$
ResNet-18 (2)		$0.785 \pm 1e-2$	$0.866 \pm 1e-2$	$0.882 \pm 7e-3$	$0.682 \pm 8e-3$	$0.905 \pm 6e-3$
EfficientNet-B4 (1)		$0.841 \pm 6e-3$	$0.896 \pm 1e-2$	$\mathbf{0.925} \pm 5e-3$	$0.760 \pm 1e-2$	$0.891 \pm 1e-2$
EfficientNet-B4 (2)		$0.849 \pm 8e-3$	$0.899 \pm 1e-2$	$0.919 \pm 4e-3$	$0.759 \pm 1e-2$	$\mathbf{0.924} \pm 1e-2$
TrOCR		$\mathbf{0.910} \pm 4e-3$	$\mathbf{0.904} \pm 5e-3$	$0.863 \pm 8e-3$	$\mathbf{0.781} \pm 8e-3$	$0.872 \pm 1e-2$

Table 4. Mean accuracy and standard deviation of 5 random initializations each using the crops generated by the U-Net model. The numbers in parentheses refer to the label encoding used in each experiment ((1): datetime encoding, (2): digit encoding), as explained in Sect. 4.3. Highest mean accuracy is highlighted in **bold**.

Model	Test	Day	Month	Year	Full date	Top5
ResNet-18 (1)	Easy	$0.950 \pm 1e-2$	$0.967 \pm 4e-3$	$0.985 \pm 1e-2$	$0.926 \pm 1e-2$	$0.971 \pm 2e-3$
ResNet-18 (2)		$0.951 \pm 4e-3$	$0.964 \pm 6e-3$	$0.994 \pm 7e-3$	$0.935 \pm 1e-2$	$0.972 \pm 4e-3$
EfficientNet-B4 (1)		$0.961 \pm 6e-3$	$0.971 \pm 6e-3$	$0.991 \pm 2e-3$	$0.950 \pm 2e-3$	0.973 ± 0
EfficientNet-B4 (2)		$0.960 \pm 7e-3$	$0.976 \pm 2e-3$	$\mathbf{0.996} \pm 5e-3$	$0.954 \pm 4e-3$	$0.973 \pm 3e-3$
TrOCR		$\mathbf{0.994} \pm 6e-3$	$\mathbf{0.998} \pm 3e-3$	$0.990 \pm 4e-3$	$\mathbf{0.982} \pm 7e-3$	$\mathbf{0.999} \pm 2e-3$
ResNet-18 (1)	Average	$0.933 \pm 8e-3$	$0.961 \pm 7e-3$	$0.980 \pm 4e-3$	$0.910 \pm 1e-2$	$0.952 \pm 5e-3$
ResNet-18 (2)		$0.933 \pm 1e-2$	$0.961 \pm 5e-3$	$0.967 \pm 6e-3$	$0.902 \pm 8e-3$	$0.969 \pm 6e-3$
EfficientNet-B4 (1)		$0.952 \pm 9e-3$	$0.967 \pm 2e-3$	$0.976 \pm 4e-3$	$0.931 \pm 8e-3$	$0.954 \pm 2e-3$
EfficientNet-B4 (2)		$0.945 \pm 2e-3$	$0.963 \pm 4e-3$	$0.976 \pm 4e-3$	$0.923 \pm 4e-3$	$0.969 \pm 4e-3$
TrOCR		$\mathbf{0.982} \pm 3e-3$	$\mathbf{0.995} \pm 5e-3$	$\mathbf{0.981} \pm 4e-3$	$\mathbf{0.960} \pm 4e-3$	$\mathbf{0.985} \pm 5e-3$
ResNet-18 (1)	Difficult	$0.833 \pm 1e-2$	$0.898 \pm 1e-2$	$0.906 \pm 9e-3$	$0.743 \pm 2e-2$	$0.860 \pm 1e-2$
ResNet-18 (2)		$0.804 \pm 1e-2$	$0.901 \pm 7e-3$	$0.901 \pm 1e-2$	$0.729 \pm 1e-2$	$\mathbf{0.901} \pm 7e-3$
EfficientNet-B4 (1)		$0.836 \pm 7e-3$	$0.916 \pm 5e-3$	$0.913 \pm 9e-3$	$0.769 \pm 1e-2$	$0.864 \pm 1e-2$
EfficientNet-B4 (2)		$0.823 \pm 1e-2$	$\mathbf{0.917} \pm 7e-3$	$\mathbf{0.927} \pm 4e-3$	$0.770 \pm 9e-3$	$0.898 \pm 1e-2$
TrOCR		$\mathbf{0.928} \pm 8e-3$	$0.905 \pm 1e-2$	$0.863 \pm 4e-3$	$\mathbf{0.797} \pm 1e-2$	$0.842 \pm 7e-3$

Dataset Dimension. We observe that models trained on the U-Net segmentations make considerably fewer errors. This finding is expected considering that the fix-size segmenter makes a naïve assumption that all rows on a given page have the same height. Consequently, the examples obtained via fix-size segmentation are a lot noisier, in particular around the edges of a cell (see Fig. 2). This noise appears to not only cause more mispredictions overall, but also elicits a

(a)	(b)	(c)	(d)	(e)
19240512	**19501225**	**19521029**	**18720515**	**19210524**
19120512	19511220	19521028	18620512	19260524
19200512	19501220	19521027	18720506	19240524
19280512	19501223	19521027	18720516	19240524

(f)	(g)	(h)	(i)	(j)
19420329	**18910909**	**19220504**	**18370126**	**19220107**
19420322	18710907	19240504	18570126	19220127
19420322	18710907	19240504	18371026	19220127
19420324	18710907	19330504	18371126	19220127

Fig. 3. A subset of errors shared among all models trained with data from U-Net segmentation. Beneath each image are displayed first the label in **bold**, then the predicted date from ResNet-18, the predicted date from EfficientNet-B4, and lastly the output from TrOCR. Both ResNet-18 and EfficientNet-B4 use the *datetime* encoding. Date format is "yyyyMMdd". The exact errors are highlighted in red. Common sources of errors that all models make are ambiguous handwriting, noisy cells with corrections or text belonging to other columns, and missing information.

more *consistent* misprediction behavior. In other words, the noise introduced in the fix-size data confounds all models in a similar manner; our best-performing ResNet and EfficientNet models share 58.1% of their prediction errors for the fix-size validation split, compared to only 49.8% using the U-Net validation split. Likewise, the benefit of employing the U-Net segmentation is evident across all of the *test* sets and models, in particular for TrOCR.

Model Dimension. Here, we only consider models trained on the U-Net data for which all models exhibited better performance. We find that approximately half of the *full date* errors (49.8% with *datetime* and 51.5% with digit encoding)

made by the ResNet-18 and EfficientNet-B4 overlap. Overall, a total of 94 validation set examples were misclassified by all three models (ResNet, EfficientNet, and TrOCR). We visualize a subset of these in Fig. 3. Out of the 94 misclassified examples, we find that 37 were labeled incorrectly. Note that any incorrect labels were manually corrected in the dataset's test splits. Likewise, 38 of these examples were missing crucial information, usually the year component (see for instance example (h) in Fig. 3). Other common sources of mispredictions were unclear or ambiguous handwriting (for instance examples (c) and (g) in Fig. 3) and noisy cells containing corrections (examples (e), (f), (j)) or text belonging to adjacent columns (examples (d), (i)). We do not find clear error patterns related to the inherent visual ambiguity of some handwritten digits, e.g. between the visually similar digits 2 and 7 or 4 and 9 [25]. We believe that errors caused by ambiguous or unclear handwriting and incorrect labels most likely cannot be prevented by leveraging a different model architecture, segmentation strategy, or training on a larger dataset. They are, ultimately, results of the data creation and annotation processes that cannot easily be resolved post-hoc. However, the error patterns that can likely be attenuated are mispredictions caused by corrections in the handwriting and information leaking in from adjacent columns, in which case more robust training that learns to ignore irrelevant information could help. We will address these in future work.

Analyzing individual mispredictions, we observe a trend in the validation split that is also observable for the test splits (Table 3, Table 4): whereas the ResNet-18 and EfficientNet-B4 models consistently perform best on the year component (compared to days or months), the year component misleads TrOCR the most. The main source of errors within the year component is missing year information, as mentioned above. In such cases, the year is typically not written within each cell but once on top of the date of birth column, making the year effectively impossible to predict based on the segmented image alone. For TrOCR with U-Net segmentation, 51 out of the 79 incorrectly classified years fall into this category, which may also partly explain why TrOCR's performance is lowest on the year component. Note, however, that no examples without a year column are present in the manually curated test data. This source of error is merely an artifact of the data segmentation process but can be addressed with a post-processing step that could e.g. involve another model identifying and classifying years written at the top of a column. We will devise such a post-processing procedure in future work, which will improve overall performance even further.

6 Related Work

The work presented in [35] is similar to ours, using machine learning to transcribe 2.3 million handwritten occupation codes from the Norwegian 1950 population census. They combined convolutional and recurrent neural network (CNN-RNN) architectures to achieve an accuracy of 97%. [32] extracted handwritten data from the 1930 US census documents, targeting 10 different columns including names, gender, and age, also using a CNN-RNN model. [46] focused on

developing a system for automatically extracting names and digits from a historical French population. [26] experimented with pre-trained OCR software Transkribus [30], Tesseract 4 [49], and OCRopy [34] when digitizing historical weather data in a tabular format from the 18th century. They showed that the OCR engines perform sub-optimally when applied out-of-the-box to detect and transcribe handwritten text in one step and that tuning the models to the specific data was required to obtain even modest results.

[21] showed that a HTR model based on the Transformer architecture with self-attention [53] can achieve competitive results using less training data while recognizing sentences at character level rather than word level. Integrating language models into digitization frameworks as a separate step have shown to boost overall accuracy when working with more complex manuscripts, guiding the search space of the optical recognition step to settle uncertainties about visually similar words or characters [22,44,51].

The problem of identifying the tabular structure in registry or census books, often referred to as document or layout analysis, is a common research problem [1,5,11,13,39]. Popular methods include relying on models from object detection such as Faster R-CNN [43] or YOLO [4,42], detection of horizontal and vertical ruling lines using Hough transform [19], and detection of individual text lines [6,15,24] and from there turning the problem into one of graph labeling to resolve which text lines comprise a cell [41]. [29] showed that in their setup errors during layout analysis have a greater negative impact on performance than errors during the recognition step.

The value of digitization stretches beyond the use cases of historical certificates and records [10,50] and can also be seen as a way of preserving culture [14].

7 Future Work

Reliable, automatic transcription of dates in the parish registries is the first step towards our visionary goal of creating a Multi Generation Registry (MGR) with familial relations for all Danes. The idea is to extend the Danish Civil Registration System (CRS) [36] backwards, going as far back as birth cohort 1920 by digitizing and linking historical Parish Registries with the CRS.

By effectively back-filling the CRS, which only contains individuals who were alive in 1968 or later, the final MGR will include all Danes from 1920 onwards. To accomplish this, we first need to have reliable, automatic transcription of names and birth dates from pairs of child and parents. Afterwards, the data transcribed from the parish registries needs to be linked with data in the CRS. Note that this will allow us to identify individuals who are present in *both* data sets. Experience can be drawn from [18] that describe the algorithms used to link the Norwegian Historical Population Register back to 1801. The MGR containing family relations for the majority of the Danish population will make it possible to analyze diseases and traits over a long period of time across several generations, as well as laterally between cousins in families with particular health histories.

8 Conclusion

We presented a new dataset of handwritten birth dates from Danish parish records which will help develop better handwriting recognition systems for use in future digitization efforts. We used the dataset to benchmark different models on handwriting recognition, including convolutional image classifiers and a sequence-to-sequence transformer architecture. We evaluated these approaches across three data splits of varying difficulty and two different strategies to segment scanned pages of the parish records. We found that the larger, transformer-based model, which was pre-trained for character recognition, performed better in most cases compared to the more resource-efficient standard image classifiers in our study.

Ethics Statement. The data published in this paper is in accordance with the applicable national law. Scanned books with personal information used in this project meet the European data protection laws. The original books are freely available to the public on the Danish National Archives' (Rigsarkivet) website. We release a curated subset of said data where individuals cannot be uniquely identified as only birth dates are provided without context, to favor academic research. We do not foresee any conflict of interest.

References

1. Andrés, J., Prieto, J.R., Granell, E., Romero, V., Sánchez, J.A., Vidal, E.: Information extraction from handwritten tables in historical documents. In: Uchida, S., Barney, E., Eglin, V. (eds) International Workshop on Document Analysis Systems, DAS 2022. LNCS, pp. 184–198. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_13
2. Bancroft, E.K.: Genetic testing for cancer predisposition and implications for nursing practice: narrative review. J. Adv. Nurs. **66**(4), 710–737 (2010). https://doi.org/10.1111/j.1365-2648.2010.05286.x
3. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=p-BhZSz59o4
4. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint. arXiv:2004.10934 (2020)
5. Boillet, M., Kermorvant, C., Paquet, T.: Multiple document datasets pre-training improves text line detection with deep neural networks. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 2134–2141. IEEE (2021)
6. Boillet, M., Kermorvant, C., Paquet, T.: Robust text line detection in historical documents: learning and evaluation methods. Int. J. Doc. Anal. Recogn. (IJDAR) **95**, 1–20 (2022). https://doi.org/10.1007/s10032-022-00395-7
7. Boone, P.M.: Adolescents, family history, and inherited disease risk: an opportunity. Pediatrics **138**(2), e20160579 (2016). https://doi.org/10.1542/peds.2016-0579
8. Bylstra, Y.: Family history assessment significantly enhances delivery of precision medicine in the genomics era. bioRxiv (2020). https://doi.org/10.1101/2020.01.29.926139, www.biorxiv.org/content/early/2020/01/30/2020.01.29.926139

9. Clevert, D., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016, Conference Track Proceedings (2016). http://arxiv.org/abs/1511.07289
10. Dahl, C.M., Johansen, T.S., Sørensen, E.N., Westermann, C.E., Wittrock, S.F.: Applications of machine learning in document digitisation. arXiv preprint. arXiv:2102.03239 (2021)
11. Déjean, H., Meunier, J.L.: Table rows segmentation. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 461–466. IEEE (2019)
12. Ross, L.F., Saal, H.M., David, K.L., Anderson, R.R.: Technical report: ethical and policy issues in genetic testing and screening of children. Genet. Med. **15**(3), 234–245 (2013). https://doi.org/10.1038/gim.2012.176
13. Gao, L., et al.: ICDAR 2019 competition on table detection and recognition (cTDaR). In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1510–1515. IEEE (2019)
14. Granell, E., Chammas, E., Likforman-Sulem, L., Martínez-Hinarejos, C.D., Mokbel, C., Cîrstea, B.I.: Transcription of spanish historical handwritten documents with deep neural networks. J. Imaging **4**(1), 15 (2018)
15. Grüning, T., Leifert, G., Strauß, T., Michael, J., Labahn, R.: A two-stage method for text line detection in historical documents. Int. J. Doc. Anal. Recogn. (IJDAR) **22**(3), 285–302 (2019). https://doi.org/10.1007/s10032-019-00332-1
16. Harris, C., Stephens, M., et al.: A combined corner and edge detector. In: Alvey vision conference, vol. 15, pp. 10–5244. Citeseer (1988)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
18. Holden, L., Boudko, S., Thorvaldsen, G.: Lenking og kobling i historisk befolkningsregister. Heimen **57**(3), 216–229 (2020)
19. Hough, P.V.: Method and means for recognizing complex patterns (1962). US Patent 3,069,654
20. Kahle, P., Colutto, S., Hackl, G., Mühlberger, G.: Transkribus-a service platform for transcription, recognition and retrieval of historical documents. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 4, pp. 19–24. IEEE (2017)
21. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: non-recurrent handwritten text-line recognition. Pattern Recogn. **129**, 108766 (2022)
22. Kang, L., Riba, P., Villegas, M., Fornés, A., Rusiñol, M.: Candidate fusion: integrating language modelling into a sequence-to-sequence handwritten word recognition architecture. Pattern Recogn. **112**, 107790 (2021)
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA (2015). http://arxiv.org/abs/1412.6980
24. Kodym, O., Hradiš, M.: Page layout analysis system for unconstrained historic documents. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12822, pp. 492–506. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86331-9_32
25. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

26. Lehenmeier, C., Burghardt, M., Mischka, B.: Layout detection and table recognition – recent challenges in digitizing historical documents and handwritten tabular data. In: Hall, M., Merčun, T., Risse, T., Duchateau, F. (eds.) TPDL 2020. LNCS, vol. 12246, pp. 229–242. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54956-5_17
27. Li, M., et al.: TrOCR: transformer-based optical character recognition with pre-trained models (2021). www.microsoft.com/en-us/research/publication/trocr-transformer-based-optical-character-recognition-with-pre-trained-models/
28. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint (2019)
29. Monroc, C.B., Miret, B., Bonhomme, M.L., Kermorvant, C.: A comprehensive study of open-source libraries for named entity recognition on handwritten historical documents. In: Uchida, S., Barney, E., Eglin, V. (eds.) DAS 2022. LNCS, vol. 13237, pp. 429–444. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_29
30. Muehlberger, G., et al.: Transforming scholarship in the archives through handwritten text recognition: transkribus as a case study. J. Doc. (2019)
31. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning (ICML-10), 21–24 June 2010, Haifa, Israel, pp. 807–814. Omnipress (2010). https://icml.cc/Conferences/2010/papers/432.pdf
32. Nion, T., et al.: Handwritten information extraction from historical census documents. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 822–826. IEEE (2013)
33. OCR, G.C.: https://cloud.google.com/vision/docs/ocr. Accessed 01 June 2022
34. OCRopy: https://github.com/ocropus/ocropy. Accessed 01 June 2022
35. Pedersen, B.R., Holsbø, E., Andersen, T., Shvetsov, N., Ravn, J., Sommerseth, H.L., Bongo, L.A.: Lessons learned developing and using a machine learning model to automatically transcribe 2.3 million handwritten occupation codes (2022)
36. Pedersen, C.B., Gøtzsche, H., Møller, J.O., Mortensen, P.B.: The danish civil registration system. a cohort of eight million persons. Dan. Med. Bull. **53**, 441–449 (2006)
37. Perslev, M., Dam, E.B., Pai, A., Igel, C.: One network to segment them all: a general, lightweight system for accurate 3d medical image segmentation. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.-T., Khan, A. (eds.) MICCAI 2019. LNCS, vol. 11765, pp. 30–38. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32245-8_4
38. Perslev, M., Darkner, S., Kempfner, L., Nikolic, M., Jennum, P.J., Igel, C.: U-sleep: resilient high-frequency sleep staging. NPJ Digit. Med. **4**(1), 1–12 (2021)
39. Prasad, A., Déjean, H., Meunier, J.L.: Versatile layout understanding via conjugate graph. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 287–294. IEEE (2019)
40. Prechelt, L.: Early stopping — but when? In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 53–67. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_5
41. Prieto, J.R., Vidal, E.: Improved graph methods for table layout understanding. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12822, pp. 507–522. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86331-9_33
42. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint. arXiv:1804.02767 (2018)

43. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
44. Romero, V., Fornés, A., Granell, E., Vidal, E., Sánchez, J.A.: Information extraction in handwritten marriage licenses books. In: Proceedings of the 5th International Workshop on Historical Document Imaging and Processing, pp. 66–71 (2019)
45. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
46. Sibade, C., Retornaz, T., Nion, T., Lerallut, R., Kermorvant, C.: Automatic indexing of french handwritten census registers for probate geneaology. In: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, pp. 51–58 (2011)
47. Ströbel, P.B., Clematide, S., Volk, M., Hodel, T.: Transformer-based HTR for historical documents. arXiv preprint. arXiv:2203.11008 (2022)
48. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (2019). https://proceedings.mlr.press/v97/tan19a.html
49. Tesseract: https://github.com/tesseract-ocr/tesseract. Accessed 01 June 2022
50. Thorvaldsen, G.L., Sommerseth, H., Holden, L.: Anvendelser av norges historiske befolkningsregister. Heimen **57**(3), 230–243 (2020)
51. Toledo, J.I., Carbonell, M., Fornés, A., Lladós, J.: Information extraction from historical handwritten document images with a context-aware neural model. Pattern Recogn. **86**, 27–36 (2019)
52. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
53. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)