Explorations in Economic History 87 (2023) 101473

Contents lists available at ScienceDirect

Explorations in Economic History

journal homepage: www.elsevier.com/locate/eeh

Research Paper

HANA: A handwritten name database for offline handwritten text recognition

Christian M. Dahl[a,*], Torben S.D. Johansen[a], Emil N. Sørensen[b], Simon Wittrock[a]

[a] Department of Economics, University of Southern Denmark, Denmark
[b] School of Economics, University of Bristol, England

ABSTRACT

Methods for linking individuals across historical data sets, typically in combination with AI based transcription models, are developing rapidly. Perhaps the single most important identifier for linking is personal names. However, personal names are prone to enumeration and transcription errors and although modern linking methods are designed to handle such challenges, these sources of errors are critical and should be minimized. For this purpose, improved transcription methods and large-scale databases are crucial components. This paper describes and provides documentation for HANA, a newly constructed large-scale database which consists of more than 3.3 million names. The database contains more than 105 thousand unique names with a total of more than 1.1 million images of personal names, which proves useful for transfer learning to other settings. We provide three examples hereof, obtaining significantly improved transcription accuracy on both Danish and US census data. In addition, we present benchmark results for deep learning models automatically transcribing the personal names from the scanned documents. Through making more challenging large-scale databases publicly available we hope to foster more sophisticated, accurate, and robust models for handwritten text recognition.

1. Introduction

The most important identifier to link individuals across historical datasets is often the personal name of individuals, due to the absence of personal identifiers such as Social Security numbers. However, these names require transcription and, even when transcribed, are prone to enumeration and transcription errors. To improve the linking of individuals across historical data sources it is important to improve handwritten transcription models both with respect to transcription accuracy and the robustness of automatic transcription models. Large-scale databases with a lot of variation are crucial components in creating such models. To this end, we introduce a large HAndwritten NAme (HANA) database, a collection of handwritten names from Denmark in the period from 1890 to 1923. It consists of more than 3.3 million labelled names across 1.1 million images of personal names, making it an enormous database for transcription of handwritten names (and words in general). We find that transfer learning from this database improves transcription accuracy and show three examples hereof and describe how the HANA database can be used for transfer learning to, for example, Danish and US census data.

As part of the global digitization of historical archives, the present and future challenges are to transcribe these efficiently and cost-effectively. We hope that the scale, quality, and structure of the HANA database can offer opportunities for researchers to test the robustness of their handwritten text recognition (HTR) methods and models on more challenging, large-scale, and highly unbalanced databases. The availability of large scale databases for training and testing HTR models is a core prerequisite for constructing high performance models. While several databases based on historical documents are available, only a few have been made available for personal names. For linking, matching, or genealogy, the personal name of individuals is one of the most important pieces of information, and being able to read personal names across historical documents is of great importance for linking individuals. For an overview of matching relying in part on names, see Abramitzky et al. (2012, 2013, 2014, 2016), Massey (2017), Feigenbaum (2018),

* Corresponding author.
E-mail address: cmd@sam.sdu.dk (C.M. Dahl).

https://doi.org/10.1016/j.eeh.2022.101473
Received 17 March 2022; Received in revised form 12 August 2022; Accepted 14 August 2022
Available online 18 August 2022
0014-4983/

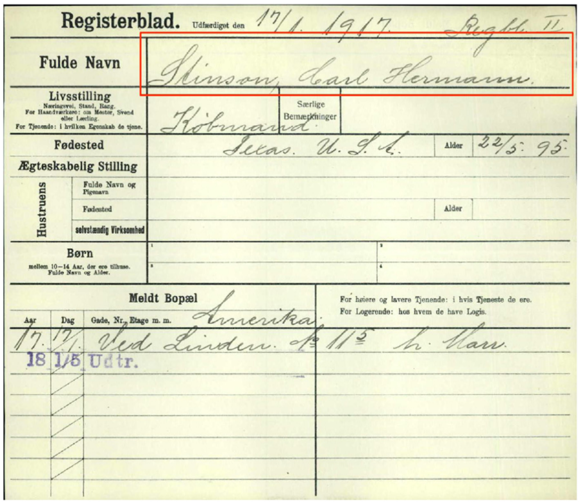
Registerblad. Udfærdiget den 17/1. 1917.
Fulde Navn Skinson, Carl Hermann.
Livsstilling
Næringsvei, Stand, Rang.
For Haandværkere: om Mester, Svend eller Lærling.
For Tjenende: i hvilken Egenskab de tjene.
Købmand
Særlige Bemærkninger
Fødested Texas. U. S. A.
Alder 22/5. 95.
Ægteskabelig Stilling
Hustruens
Fulde Navn og Pigenavn
Fødested
Alder
selvstændig Virksomhed
Børn
mellem 10–14 Aar, der ere tilhuse. Fulde Navn og Alder.
Meldt Bopæl
Aar Dag Gade, Nr., Etage m. m.
Amerika
17. 17/1. Ved Linden. No 11
18 1/5 Udtr.
For høiere og lavere Tjenende: i hvis Tjeneste de ere.
For Logerende: hos hvem de have Logis.

Fig. 1. Example of a Police Register Sheet: The figure shows an example of the raw documents received from Copenhagen Archives. The first line specifies the date the document was filled. The second line contains the full name (which we have outlined by adding a bounding box not present in the original images) while the occupation is written in the smaller region just below the name. The fourth line contains the birth place and the birth date. The sections below contain information on the spouse and children.

Price et al. (2019), Abramitzky et al. (2020), Bailey et al. (2020), and Abramitzky et al. (2021). Here, Bailey et al. (2020) and Abramitzky et al. (2021) both discuss the relatively low matching rates when linking two transcriptions of the same census to each other (removing issues related to mortality and emigration, which otherwise lead to cases where an individual is only present in one of the two censuses). This is in part due to low transcription accuracy of names and in part due to common names (making it impossible to distinguish between which of two individuals to match). Furthermore, both papers are concerned with the lack of representativeness of the linked samples. While issues related to common names is an inherent challenge of historical linking, the errors related to transcription strongly motivate the HANA database, i.e., collecting and sharing more data of higher quality, which allows for better transcription methods. This, in turn, can improve transcription performance and reduce potential biases in record linking.

In total, the HANA database consists of more than 1.1 million personal names written on single-line images with each personal name consisting of an average of three names. All original images are made electronically available by Copenhagen Archives and the processed database described is made freely available. While most of the existing databases contain single isolated characters or isolated words, such as the names available in the Handwriting Recognition database at Kaggle,[1] one of the important features of our database is the resemblance with other challenging historical documents, where source data often contain general image noise, different writing styles, and varying traits across images. Our code is made available by Dahl et al. (2022) at https://github.com/TorbenSDJohansen/HANA.

The rest of the paper is organized as follows: In Section 2, we describe the database and the data acquisition procedure in detail. Section 3 presents the benchmark results on the database using a ResNet-50 deep neural network in three different model settings. In addition, this section provides examples validating how the HANA database can be used for transfer learning on Danish and US census data. In Section 4, we discuss the database and the benchmark methods and results, in addition to some considerations for future research. Section 5 concludes.

2. Constructing the HANA database

This section describes the HANA database in detail and the image-processing procedures involved in extracting the handwritten text from the forms. In 1890, Copenhagen introduced a precursor to the Danish National Register. This register was organized and structured by the police in Copenhagen and has been digitized and labelled by hundreds of volunteers at Copenhagen Archives. Fig. 1 shows an example of one of these register sheets, with the name at the top outlined by a bounding box.

The Register Sheets In total, we obtain 1,419,491 scanned police register sheets from Copenhagen Archives. All adults above the age of 10 residing in Copenhagen in the period 1890 to 1923 are registered in these forms. Children between 10 and 13 were registered on their father's register sheet. Once they turned 14, they obtained their own sheet. Married women were recorded on their husband's register sheet, while single women were recorded on their own sheet (Archives, 2022). This is most likely biasing our database to

[1] The Handwriting Recognition database on handwritten names, containing names with somewhat clearly separated characters, is available at https://www.kaggle.com/landlord/handwriting-recognition. Similarly to the transfer learning exercises we present in detail in Section 3, Appendix A shows that our approach also works for this larger dataset, although the performance increase is smaller.

include more men than women, as we focus on the main individual on the register sheets. The final database consists of 1,106,020 personal names with a total of 228,238 spouses registered.

A wealth of information is recorded in the police register sheets, including birth date, occupation, address, name of spouse, and more, all of which is systematically structured across the forms. While this paper focuses on extracting and creating benchmark results for the personal names, the remaining information can be constructed using similar procedures to those presented in this paper and may serve as additional databases for HTR models.

Data Extraction and Segmentation To segment the data, we use point set registration. Point set registration refers to the problem of aligning point spaces across a template image to an input image (Besl and McKay, 1992). To find point spaces that roughly correspond to each other across semi-structured documents, we extract horizontal and vertical lines from the document. We use the intersections as the point space, which we align with the template points. We briefly outline the method below; see Dahl et al. (2021) for more details.

To start the process of extracting the personal names from the forms, we binarize the images. We extract horizontal and vertical lines from the documents by performing several morphological transformations, see, e.g., Szeliski (2010). The intersections are subsequently found using Harris corner detection (Harris and Stephens, 1988). Once we have the point space defined, we use Coherent Point Drift (Myronenko and Song, 2010), which coherently aligns the point space from the input image to the point space on the template image. This yields a transformation function that maps the points found in the input images to the points in the template image. To improve the segmentation performance of the database, we add several restrictions to the transformations such that all extreme transformations are automatically discarded. This reduces the size of the database to just over 1.1 million images with attached labels. Even though this removes more than 20% of the data, we believe the gain from more reliable data outweighs the cost associated with a smaller database. Once we have prepared the images, we clean the labels to fit into a Danish context, which implies that all non-Danish variations of letters are replaced with the Danish equivalent of these. A few of these might be incorrect, e.g., if the individuals are foreigners, but we expect the level of mis-classification arising from this to be smaller than the number of characters labelled incorrectly by the volunteers at Copenhagen Archives. In addition, we restrict the sample to names that only contain alphabetic characters and with a length of at least two characters, yielding a final database of 1,105,904 full names. Fig. 2 shows examples of segmented images with their corresponding labels.

It is possible to increase the number of extracted names for each sheet by considering the spouse and children of an individual. However, this would entail lowering the quality of the data, as the last name is not necessarily present for these individuals and the quality of the segmentation is also lower. Hence, we leave this for future work.

The personal name labels are either categorized as first or last names by Copenhagen Archives. Most commonly, the last name is written as the first word on the image while the subsequent words are the first and middle names (in that order). However, some exceptions occur, and there are other rules that may interfere with the structure of the ordering, such as underlining and numbering. The structure of the database can therefore be challenging for HTR models, as this structuring complication has to be overcome by the models.

Train and Test Splits To evaluate the performance of HTR on the database, we split it into a train and a test split. The test split consists of 5% of the total database and is randomly selected. The training data consists of 1,050,082 documents while the test data consists of 55,822 documents. 2129 surnames are only represented in the test sample, which contains a total of 10,228 unique last names relative to the overall of almost 70,000 unique last names.

The database is highly unbalanced due to vast differences in the commonness of names. Only the 604 most common surnames in the database occur at least 100 times, and only the 3463 most common surnames occur more than 20 times. This covers slightly more than 85% of the data, meaning that almost 15% of the images contain names that occur fewer than 20 times. This naturally leads to challenges for any HTR model, as it needs to learn to recognize names with few or even zero examples in the training data. However, this is also an important and indeed crucial goal to work towards.

Labelling While transcribers at Copenhagen Archives were instructed to make accurate transcriptions of the register sheets, there exist humanly introduced inconsistencies in the labels. The same points made by Deng et al. (2009) can be made here, as there are especially two issues to consider. First, humans make mistakes and not all users follow the instructions carefully. Second, users are not always in agreement with each other, especially for harder to read cases where the characters of an image are “open to interpretation”.

With respect to the first point, we perceive this as part of the challenge for constructing any digital handwriting database based on human transcriptions. For this database, Copenhagen Archives used super users to validate the transcriptions. In addition, it is possible to submit requests for corrections at the website of Copenhagen Archives and thereby change incorrect labels. With respect to the second point, a number of considerations should be taken into account. A common labelling error found in the database is the existence of subtle confusing characters, similar names, or phonetically spelling errors. Characters or names that are often misread are, e.g., *Pedersen* versus *Petersen, Christensen* versus *Christiansen*, and *Olesen* versus *Olsen*. Solutions for these complications are difficult, as it is in many cases a judgement call by the transcriber.

Further Characteristics of the Database Despite there being 69,906 unique surnames and 48,394 unique first and middle names, the total number of unique names amounts to only 105,607, as there is an overlap between the two sub-groups. There are fewer than 50,000 examples of the characters *q, w, x, z, å*, and *æ*. For *q* and *å*, there are fewer than five thousand examples. The vast majority of names contain four to nine characters, with only 6.35% of the names being shorter or longer. Quite frequently reported for Danish last names is the fraction of names ending with *sen*. For this database, 710,117 surnames end with *sen*, which corresponds to 64.21% of all last names in the database. Appendix C provides additional characteristics of the names in the database.

3. Benchmark results

This section describes the benchmark results published together with the HANA database and the value of transfer learning is illustrated. We use a variant of a ResNet-50 network for estimating the benchmark results. We transcribe the surnames in a character-by-character classification fashion. The predictions are subsequently matched to the closest existing name. One could also consider the surnames as an entity and classify each word in a holistic sense. We believe this is problematic due to the unbalanced nature of this database, meaning that the training sample does not contain all unique names. We train three neural networks, one to predict the last name, one to predict the first and last name, and one to predict the entire name, i.e. first, middle, and last names. We start by describing the architecture, optimization, and other details of the neural networks used in the paper. Scripts for the implementations are all in Python (Van Rossum and Drake, 2009) using PyTorch (Paszke et al., 2019). We then present benchmark results on the HANA database as well as the examples showing the value of transfer learning from the HANA database.

Network Architecture Each neural network uses a ResNet-50 with bottleneck building blocks (He et al., 2016) as its feature extractor; the weights of the PyTorch version of ResNet-50 pretrained on ImageNet (Deng et al., 2009) are used as the initial weights. The neural networks differ only insofar as their classification heads differ. Here, a method similar to the one described in Goodfellow et al. (2013) is used, with the exception that the sequence length is never estimated. The weights (and biases) of the classification heads are randomly initialized. For the last name network, 18 output layers are used (names are at most 18 letters long), each with 30 output nodes (letters a-å as well as a “no character” option). For the first and last name network, 36 output layers are used (2 names of at most 18 letters), each with 30 output nodes. For the full name network, 180 output layers are used (up to 10 names of at most 18 letters), each with 30 output nodes.

Optimization All neural networks are optimized using stochastic gradient descent with momentum of 0.9, weight decay of 0.0005, and Nesterov acceleration based on the formula in Sutskever et al. (2013). The batch size used is 256 and the learning rate is 0.05. The networks are trained for 100 epochs and the learning rate is divided by ten every 30 epochs. The loss of each classification head is the negative log likelihood loss of the head, and the total loss is the average of the negative log likelihood loss of each head.

Image Preprocessing Images are resized to half width and height for computational reasons (resulting in images of width 522 and height 80). The images are normalized using the ImageNet means and standard deviations (to normalize similarly to the pretrained ResNet-50 feature extractor). During training, image augmentation in the form of RandAugment with $N = 3$ and $M = 5$ is used (Cubuk et al., 2020); the implementation is based on Kim (2020).

Prediction of Networks Some post processing of predictions is performed. Each layer is mapped to its corresponding character (the 29 letters and the “no character” option). Then, for each name (i.e. sequence of 18 output layers), the “no character” predictions are removed and the remaining letters form the prediction. Letting, θ_i denote the “no character” option for character i, this means that both [h, a, n, θ_4, s, $\theta_6, \dots, \theta_{18}$] and [h, a, n, s, $\theta_5, \dots, \theta_{18}$] will be transformed to *hans*.

Matching As an additional step, we also test the performance if we refine the predictions of the networks by using matching. In some cases, a list of possible names (i.e., a lexicon of valid outcomes) may be present, in which case this can be used to match predictions that are not valid to the nearest valid name.[2] Specifically, we use the procedure in the `difflib` Python module to perform this matching.

For the last name network, the predictions that do not fall within the list of valid last names are assigned to the nearest last name. For the first and last name network, a similar procedure is used separately for the first name and the last name. For the full name network, a similar procedure is used separately for the first name, the up to eight middle names, and the last name.

Performance Measures To measure the performance of our networks, we focus on the character error rate (CER) and the word accuracy (WACC) of our models. Here, we define the WACC such that a name prediction is considered to be incorrect if a single character or more of the name is transcribed incorrectly, and thus the character error rates are significantly lower than the word error rates implicitly reported.[3] We consider performance both with and without using matching to a lexicon as a post processing step. Further, we report performance at different levels of *data coverage*. As our networks report a measure of their confidence for each prediction, we can rank all predictions by this measure. Then, we can calculate the CER and the WACC at, e.g., 90% data coverage by removing the 10% of predictions where the network is the least certain. We believe this metric is interesting, as it might be used to, e.g., (1) select the predictions where a sufficient WACC is reached or (2) let humans assist in transcribing images that the network is particularly uncertain about.

Results Table 1 shows the CER and the WACC of our three different models on the HANA database. Further, the WACC is also reported when matching the name predictions to a lexicon of valid names. The table reports the performance at both 100% and 90% data coverage, showing the CER and WACC if we use all transcriptions in the test set compared to only testing on the 90% of the test data on which the model is the most certain. As can be seen, there is a trade-off between data coverage and accuracy, which is the motivation for also showing the results using a threshold at the 90th quantile. The table represents three different models for character-by-character recognition. The first model predicts only the characters in the last name, the second model predicts the first name and the last name, and the third model predicts the full name sequence. All of them are trained on the full database. For the

[2] When using matching, we define our lexicons by assuming that the sample of names available in our data represents all possible names, and thus our lexicons might be considered too perfect, which could lead to an upward bias of the performance we report when using matching. We have tested the robustness of our matching procedure in the last name setting to including a much larger set of names, expanding it from around 70,000 to around 200,000 last names by adding around 130,000 “new” US last names taken from https://fivethirtyeight.datasettes.com/fivethirtyeight/most-common-name~2Fsurnames, which barely changes the performance (no more than a 0.21% decrease in the word accuracy).

[3] WACCs are reported and the word error rates are given as 1 minus WACCs.

Table 1
CER and WACC on the HANA database.

Names	Data coverage	CER	WACC	
			Raw	Matched
Last	100%	1.48%	94.33%	95.68%
First and last	100%	1.66%	93.52%	94.79%
Full	100%	11.82%	67.44%	68.81%
Last	90%	0.36%	98.36%	98.41%
First and last	90%	0.55%	97.29%	97.46%
Full	90%	8.71%	72.78%	74.10%

The table shows the test performance of the HTR models as measured by CER and WACC. The data coverage is defined as the fraction of the test database the model is tested on (keeping predictions where the network is most confident). For the models with 90% data coverage, we remove the 10% of the test sample where the model is most uncertain. All models are trained on the full train database allowing the networks to learn primitives and characters from uncommon names.

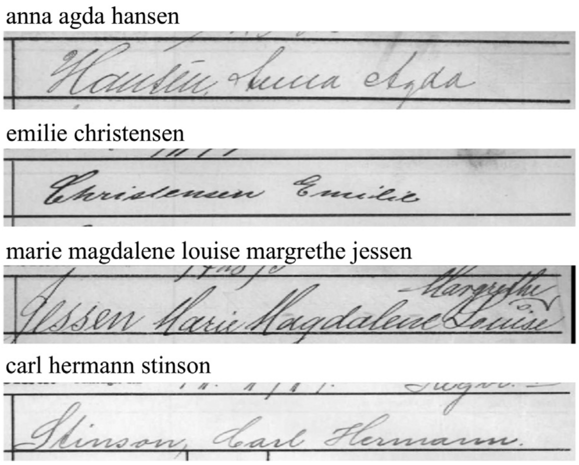

Fig. 2. Examples from the HANA Database: The figure shows examples from the HANA database with the corresponding labels written above. The last name is typically written as the first word followed by the first and middle names, which is the case for all images above.

full name model, the number of names present in a person's predicted name is equal to the number of names in the corresponding label in 96.85% of the cases.

At full data coverage, the WACC (without matching) for the last name model is 94.33%, 93.52% for the first and last name model, and 67.44% for the full name model.[4] These WACCs increase by just over 1 percentage point when we use matching for all three models. In some downstream analyses, such as matching, the CER rather than the WACC might be more relevant, since matching methods are often able to overcome single characters being incorrect. Here we calculate the CER using the Levenshtein distance of the predictions (without matching), documenting very low CERs of 1.48% for the last name network, 1.66% for the first and last name network, and 11.82% for the full name network. These error rates are calculated using the full predicted names (including separation) since we perceive a predicted lack of separation as a mistake as well.

When turning to the performance achieved by our models at less than complete data coverage, the results are even more impressive: At 90% data coverage, the CER is significantly lower and the WACC significantly higher for all models (see Table 1). As an example, the last name model now achieves 98.36% WACC and a CER of just 0.36%. This means that that the model is able to transcribe 90% of the names with this impressive performance, and direct attention to the remaining 10% of the names, which may be either discarded or manually reviewed, depending on the goal of the downstream task. However, the additional benefit provided by matching is now smaller, increasing the WACC by a negligible amount. This is likely due to the network sorting away predictions that are incorrect, but which may be fixed through matching, on top of the fact that additional increases to performance is very limited, given the already near perfect performance. Turning our attention to the other two models, we see quantitatively the same: The CERs are significantly lower and the WACCs significantly higher. For the first and last name model, matching no longer improves performance by more than a negligible amount, but for the full name model there is still a tangible benefit provided by matching.

To study in more detail the performance achieved at different levels of data coverage, Figs. 3 and 4 show the WACC at different levels of data coverage for the last name and the first and last name model, respectively. Fig. 3 shows the WACC achieved by the

[4] The full name model is evaluated on the full names of the individuals and has to take into account the correct ordering of the all names in order to transcribe each name correctly, which we believe explains the performance deterioration.

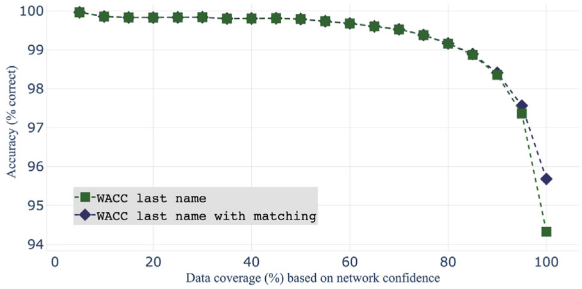

Fig. 3. Performance on the HANA Database: Last Name: The figure shows the WACC of the last name model on the test set of the HANA database at different levels of data coverage, both with and without matching. The leftmost point corresponds to 5% data coverage and the rightmost point to complete data coverage.

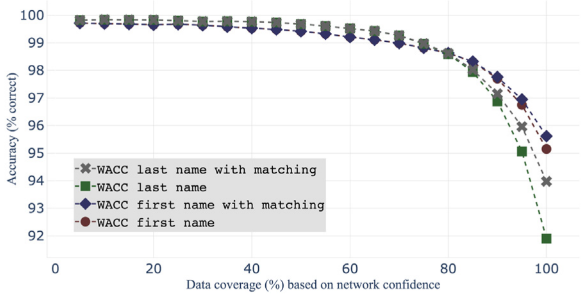

Fig. 4. Performance on the HANA Database: First and Last Name: The figure shows the WACC of the first and last name model on the test set of the HANA database at different levels of data coverage, both with and without matching and separately for the first and last names. The leftmost point corresponds to 5% data coverage and the rightmost point to complete data coverage.

last name model both with and without matching at between 5% and 100% data coverage. While both the model with and without matching achieve higher WACCs on lower data coverage, the difference in performance between the two models only appear at high levels of data coverage (a finding echoed by their WACCs at 90% compared to 100% data coverage, see Table 1); in fact, their performance is very close even at 95% data coverage.

Fig. 4 shows the WACC achieved by the first and last name model both with and without matching at between 5% and 100% data coverage, separately for the first and the last name. While a similar pattern emerges in regards to the performance with and without matching, i.e., matching makes a difference only at high levels of data coverage, a new, interesting observation is that the difference in performance between the first and the last name: At high levels of data coverage, the model achieves a higher WACC on the first name relative to the last name, while at lower levels of data coverage this reverses (at around 80% data coverage). Finally, note that the performance achieved by this model on the last name is lower than the model trained only to transcribe last names, suggesting that it sacrifices some performance in this regard by also having to transcribe first names.

As we discussed in Section 2, the commonness of names varies significantly, with some names represented very few times (including 2143 cases where a name is only present in the test split of the HANA database). This, in turn, leads to large differences in the number of training examples available by name, which leads to differences in the performance by name commonness. To shed light on the performance difference by name commonness, Fig. 5 shows the WACC for the last name model on subsets of the test data, determined by the commonness of the names in the training data. Specifically, we evaluate the model under two conditions: First, we gradually *remove* the rarest names (i.e., the names that occur fewest times in the training data) and evaluate it on the remaining names (Panel 5a with WACC on the y-axis and the x-axis indicating removing names that occur no more than the specified number of times in the training sample). This shows how the WACC of the model improves as we remove the rarest names. For example, when removing names that occur no more than 50 times in the training data, the model reaches almost 99% WACC. Second, we gradually *add* the rarest names and evaluate it on the names that are now included (Panel 5b, with the x-axis now indicating adding names that occur at most the specified number of times in the training data). This shows that the model achieves 60.01% WACC even on the names that occur zero times in the training data, and that this performance rapidly improves as we add names that occur more times, with around 85% WACC achieved on the subset of names that occur 200 or fewer times in the training data, until finally the WACC on the full sample of 94.33% is reached when including all names.

Transfer Learning By publishing the database we aim to establish a foundation for transfer learning to handwritten names from other data sources. This in turn can help others transcribe handwritten names more accurately – while also reducing costs, as less manual labelling will be needed. To motivate the usefulness of the HANA database for transfer learning, we present results for three separate transfer learning examples: Transcription of handwritten surnames from Danish and US census data (see Fig. 6 for some examples of these images), and transcription of the handwritten names from the Handwriting Recognition database from Kaggle

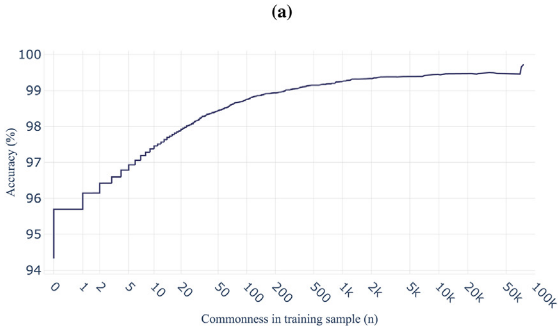

Fig. 5. Performance on the HANA Database by Name Commonness: Last Name: The figure shows the WACC of the last name model on subsets of the test data, determined by the inclusion/exclusion of uncommon names. Panel 5a shows the WACC achieved by the model when *excluding* names that occur fewer times than specified on the x-axis, such that the leftmost point corresponds to the WACC on the full test sample and the rightmost point to the WACC calculated only on names that occur more than 75k times. Panel 5b shows the WACC achieved by the model when *including* names that that occur fewer times than specified on the x-axis, such that the leftmost point corresponds to the WACC on the test subset of names that occur zero times in the training sample and the rightmost point to the WACC on the full test sample.

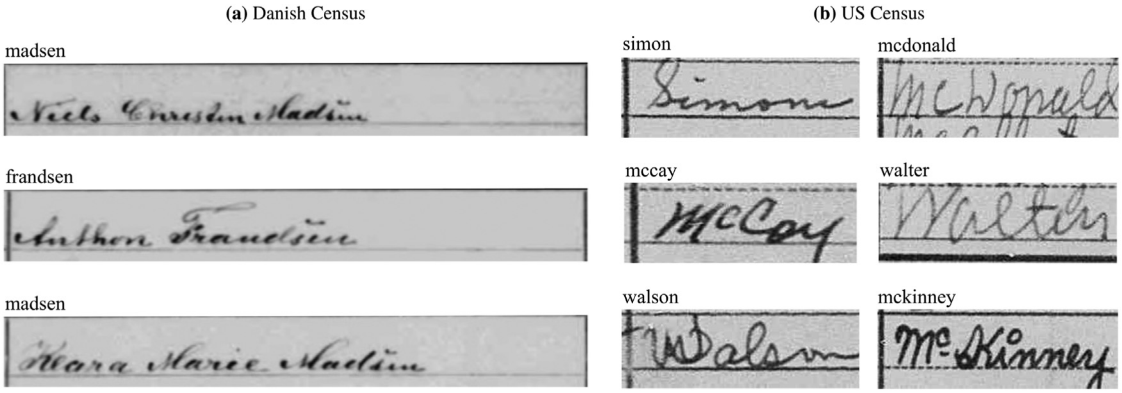

Fig. 6. Examples from the Danish and US Censuses: The figure shows examples of the surnames from the (Panel 6a) Danish and (Panel 6b) US census with the last names included in the text box above each minipic. As seen from these examples, the Danish census images mimics to a greater extent the images in Fig. 2 while the US census minipics include only the surnames of the individuals. However, it seems that in the Danish census the surname tend to be located to the right on the images while in the HANA database the surnames are usually located to the left, see Fig. 2. The width to height ratio of the HANA database is 6.5 which is similar to the Danish census with a ratio of 7.2 while the US census ratio is 3.7.

which contains transcriptions of 410,000 handwritten names. In this section, we provide details from our experiments for the two census datasets; Appendix A provides details for our third example. In all cases, our results demonstrate that adopting a transfer learning strategy based on the HANA database can increase transcription accuracy, even when large amounts of training data is available.

For both the Danish and US census data, we present two sub-cases. We analyze the performance when a relatively small number of training images are available (approximately 10,000) and when a larger number of training images are available (approximately 50,000). By training networks both with and without transfer learning on these datasets, we can infer the magnitude of the performance boost achieved by using the HANA database for transfer learning.

In total, we train eight new networks. Due to the difference in the number of labelled images for each dataset and the use of transfer learning from the HANA database last name network, we expect that the optimal learning rate for each network might differ substantially. For this reason, we perform a grid search on a validation set consisting of five percent of the training data for each network to tune the learning rate. All other training settings are similar to those we used to train models on the HANA database. Thus, all new models are similar to the last name model on the HANA database, and training only differs with respect to the learning rate used and the starting weights.[5]

Fig. 7 shows our main findings on transfer learning. The performance based on the Danish census is illustrated in Panel 7a, while the performance based on the US census is illustrated in Panel 7b. The data coverage is gradually increasing along the first axis, and

[5] The image sizes of the source files also differ, and are around 465 by 65 for the Danish census and 350 by 95 for the US census, which are the resolutions we train these networks at. To provide the most fair comparison, we use weights from pretrained models on ImageNet as the initial weights for the models where we do not transfer learn from the HANA database, similarly to our training of the models on the HANA database.

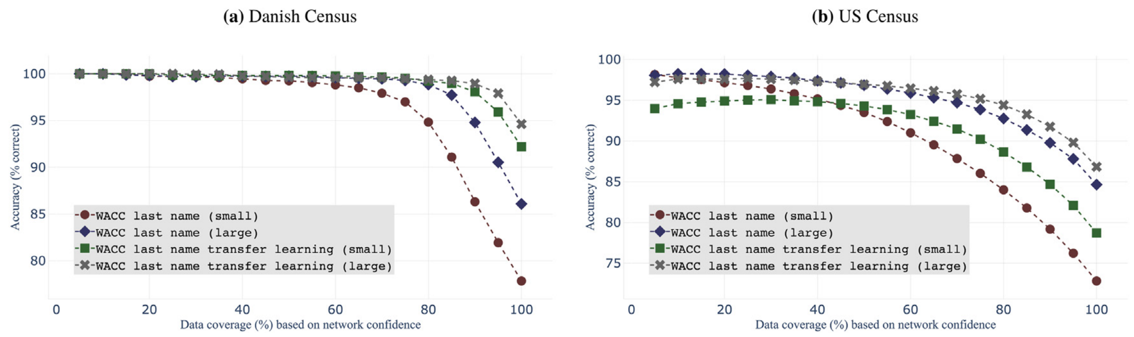

Fig. 7. Transfer Learning Performance on Danish and US Census: The figure shows the performance gain from adopting a transfer learning strategy based on the HANA database. Panel 7a shows the performance on the Danish census data and Panel 7b on the US census data.

at a data coverage of 100% it is clear that the worst performing model for each census is the network trained on the small database without transfer learning while the best performing model is the network trained on the large database with transfer learning. Quite interestingly, there is a difference between which model is the second best between the Danish and US census. For the Danish census, the model trained on the small database with transfer learning is better than the model trained on the large database without transfer learning, while this is reversed for the US census. This is likely due to the larger similarity between the HANA database and the Danish census compared to the similarity between the HANA database and the US census (see Figs. 6 and 2). However, we also find large performance gains for the US census, particularly for the small database, which seems intuitive as smaller datasets have less information to learn from and thus would benefit more from transfer learning. The US census images differ from those of the HANA database and the Danish census in that they contain only the surname. This might contribute to the smaller performance gain we see when applying transfer learning, compared to the Danish census. Further, the performance both with and without transfer learning is worse on the US census. We use a large test sample from the US census to validate the performance with more than 60,000 test examples, while for the Danish census data we have approximately 6000 test examples. In general, it seems that the US census data is more difficult to transcribe, making the reduction in error rates from transfer learning even more promising. At full data coverage on the Danish census, the WACC increases from 77.8% to 92.2% for the small training set and 86.1% to 94.6% for the large training set. On the US census, the WACC increases from 72.8% to 78.7% for the small training set and from 84.7% to 86.8% for the large training set.

To shed more light on the lower performance gain on the US compared to the Danish census from adopting our transfer learning strategy, we review the performance of the four US models on a subset of the US census data, selected by keeping only names that also occur in the HANA database. In total, 24,689 test examples from the US census data contain names that are also present in the HANA database. On this subset, we expect the gain from adopting our transfer learning strategy to be higher. Across the four models, slightly lower WACCs are achieved on this subset (around 1 percentage point), indicating that this subset is, for some reason, slightly more difficult than the overall US census. However, the performance boost from using transfer learning is roughly the same as for the other US census models. Overall, it appears that it is not the names that are the main drivers of the lower performance gain on the US census, but rather other features such as the handwriting style. Appendix B provides further details.

We believe that transfer learning from the HANA database can provide large gains when transcribing handwritten names from other data sources. These gains are particularly large when transferred to a domain that is close to the HANA database and when relatively few labelled images are available. However, the gains can also be substantial when transferred to a domain that is further away and when relatively many labelled images are available. Using transfer learning with more than 50,000 training sample points, we achieve an error rate reduction of 61.4% for the Danish census and 14.2% for the US census data. This equates to 21,772 corrections of falsely transcribed images when transcribing one million handwritten US names. For the Danish names, and for the US names when only 10,000 labelled images are available, the increase in transcription accuracy is much larger. Thus, while transfer learning leads to smaller gains when more labelled images are available, the benefits are still tangible. Further, we find that most currently available datasets with handwritten names contain fewer than 50,000 labelled handwritten names, and labelling thousands of images is both time-consuming and expensive. This means that transfer learning from the HANA database not only help improves transcription accuracy, it also reduces costs as fewer labelled images are needed.

4. Discussion

Due to computational constraints, we tested the performance of relatively few models on the HANA database. Yet, our models still achieved impressive performance, being able to transcribe names with high accuracy. As these models are the first results on this database, there are currently no available comparable results, and we hope that other researchers can use these results as a benchmark and transfer learn from this database.[6] To show the validity of such a strategy, we presented two transfer learning exercises in detail (with a third one discussed in Appendix A), showing that the use of the HANA database can significantly increase the transcription accuracy of names from both Danish and US censuses.

We believe that large-scale databases are a necessary prerequisite for achieving high accuracy when transcribing handwritten text. This database proves to be sufficiently large for models to read handwritten names with high accuracy. The high performance is achieved despite several stated complications. The most common complications with the labels and the corresponding images are the structure of the personal names on the images relative to the labels, confusion of certain characters, and general typos. We emphasize that the labels are not perfect, and we find that this is especially true for harder to read cases where certain characters are "open to interpretation".

Names that are nearly identical, e.g., *Pedersen* versus *Petersen* or *Poulsen* versus *Paulsen*, are often transcribed incorrectly and these account for a relatively large share of the incorrect transcriptions. As many of these disparities are debatable and the "ground truth" can be difficult to infer, we try allowing for one character to be incorrect in the transcription (as the label might mistakenly be *Petersen* instead of *Pedersen*) and find that 97.97% of the transcribed (last) names have *at most* one character incorrect and 98.96% of the predictions have *at most* two characters incorrect, which is in line with the low CERs reported in Table 1.

5. Conclusion

This paper introduces the HANA database, which is the largest publicly available handwritten personal name database. The large-scale HANA database is based on Danish police register sheets, which have been made freely available by Copenhagen Archives. The final processed database contains a total of 3,355,033 names distributed across 1,105,904 images. Benchmark results for transcription based deep learning models are provided for the database on the last name, first and last name, and full name.

Our goal is to create and promote a more challenging database that in many ways is more comparable to other historical documents. Specifically, historical documents are often tabulated and can therefore be cropped into single-line fragments, which should make it easier to train HTR models and to make more efficient transcriptions. Second, the naturalism of the police register sheets are in our opinion quite comparable to a lot of widely used historical documents such as census lists, parish registers, and funeral records. This makes any performance based on these documents more representative of the performance that would be obtained in custom applications. To validate this point, we showed examples of models transfer learning from the HANA database on Danish and US census handwritten names. We find that transfer learning increases the word accuracy from 77.8% to 92.2% (86.1% to 94.6%) for the Danish census and from 72.8% to 78.7% (84.7% to 86.8%) for the US census when 10.000 (50.000) training examples are available.

We want to highlight two important features of our database. First, despite the challenges associated with labelling errors and unstructured images, the size of the database appears to compensate, making possible high performance models for automatically transcribing handwritten names. Second, related to the prior point, despite the commonness of names being far from evenly distributed, resulting in a highly unbalanced sample of the represented names, with 65,020 names singularly represented out of a total of 105,607 different names, the models still generalize well. We view this as very encouraging, suggesting that high performance automatic transcription is possible even in difficult and realistic scenarios.

We have performed image-processing procedures to make the database useful for training single-line learning systems. Further, the code for replicating our results and transfer learning from our models is made freely available. We strongly encourage other researchers to use the HANA database and to make improvements to our procedures in order to continuously increase the size and quality of the database. Ultimately, we believe this can help making automatic transcriptions of personal names and other handwritten entities much more precise and cost efficient in addition to making the transcriptions fully end-to-end reproducible. By adding improvements to existing linking methods, due to fewer transcription error rates, this could further incentivize the usage and construction of reliable long historical databases across multiple generations.

Acknowledgments

We are grateful to the BYU Record Linking Lab for providing the US census data and the Copenhagen Archives who have supplied large amounts of scanned source material. The authors also gratefully acknowledge valuable comments from Philipp Ager, Anthony Wray, Paul Sharp, the editor and an anonymous referee. Torben gratefully acknowledges financial support from the Independent Research Fund Denmark, grant 8106-00003B. Emil gratefully acknowledges financial support from the European Research Council (Starting Grant Reference 851725). The HANA database is available at https://www.kaggle.com/sdusimonwittrock/hana-database.

Appendix A. Additional transfer learning illustration

As an additional illustration, we transfer learn from our last name model to the Handwriting Recognition database available from Kaggle containing roughly 410,000 images. To do so, we make a few changes to make the data fit into our current framework. First, we split names including hyphens and only include the last part of the names. This means that 806 labels in the test set are altered

[6] Comparing the performance we achieve with the performances of other methods for handwriting text recognition or scene text recognition is difficult, as the inherent challenge of tasks differ significantly between datasets, as evident by large differences in transcription accuracy of the same method applied to different datasets. However, we note that we achieve high levels of WACC both in absolute terms and in comparison with WACCs achieved by state of the art methods on many other text recognition challenges.

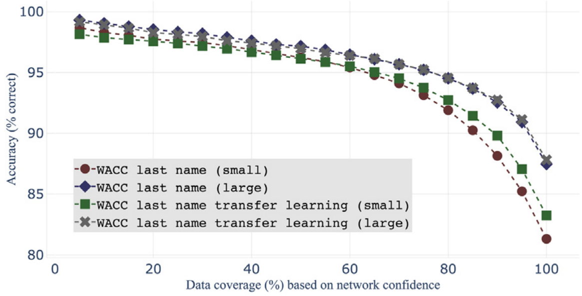

Fig. A1. Transfer Learning Performance on Kaggle HTR: The figure shows the performance gain from adopting a transfer learning strategy based on the HANA database. We find that the performance gain is larger for the smaller training sets, but still present even when nearly 400,000 observations for training are available.

in this manner, potentially upward biasing our models. In addition, we remove empty name labels and names including special characters, which reduces the test set from 41,370 to 41,264 images. Finally, we remove three images from the training set and one image from the validation set, due to above 18 characters in the corresponding names. In total, 329,982 training images, 41,252 validation images, and 41,264 test images remain after these corrections.

The purpose of showing the performance of transfer learning from the HANA database to this dataset is to show the performance gain from transfer learning in a setting with a very large training sample consisting of hundreds of thousands of sample points. We create two training sets: A "small" set, where only the validation images are used for training (41,252 images) and a "large" set, where both the training *and* the validation images are used for training (371,234 images). In both cases, we use the test images to evaluate our models. We train two models for each set: One where we transfer learn from the HANA database last name model and one where we do not use transfer learning.

We proceed similarly to the transfer learning examples discussed in Section 3. The only differences are: (1) the image size is smaller, here around 388 by 40, which is the resolution we train at, and; (2) for computational reasons, we conduct only a search for the learning rate for the two models trained on the small set, and then use the learning rates found here also for the two models trained on the large set.

As we expect, the performance increase of using transfer learning is highest for the small training set, where accuracy increases from 81.33% to 83.24%. For the large training set, accuracy increases from 87.48% to 87.83%. While these increases are lower compared to our other transfer learning exercises, it is important to note that the size of the training sets are much larger: The "small" training set we use for this example is almost as large as the large training set we use in our other examples, and the "large" training set we use for this exercise contains 371,234 samples, a scale often not realistically obtained.

We also study the performance of these models at different levels of data coverage. Fig. A.1 shows the performance of the four models at different levels of data coverage. Much in line with our earlier results, the models trained on the small set perform worse than the models trained on the large set, but in both cases the model transfer learning from the HANA database outperforms the model not utilizing the HANA database for transfer learning.

Appendix B. US census "Danish" subset

In this appendix, we show details regarding the performance of our models on the US census data, when restricting this dataset to only names that are also present in the HANA database. This corresponds to 24,689 test examples from the US census as opposed to the original 60,031 test images. On this smaller subset, we obtain WACCs of 78.4% and 71.6% for the models trained on the small

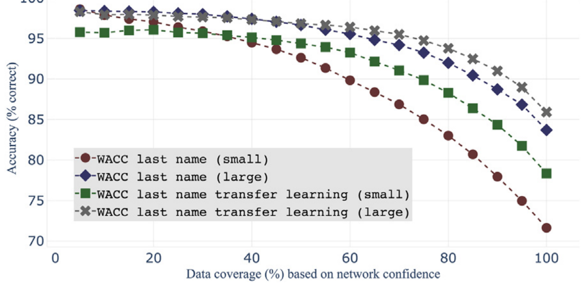

Fig. B1. Performance on US Census: "Danish" Subset: The figure shows the WACC of the four models trained on the US census data, evaluated on the subset of the test split that contains names that are present in the HANA database, at different levels of data coverage. The leftmost point corresponds to 5% data coverage and the rightmost point to complete data coverage.

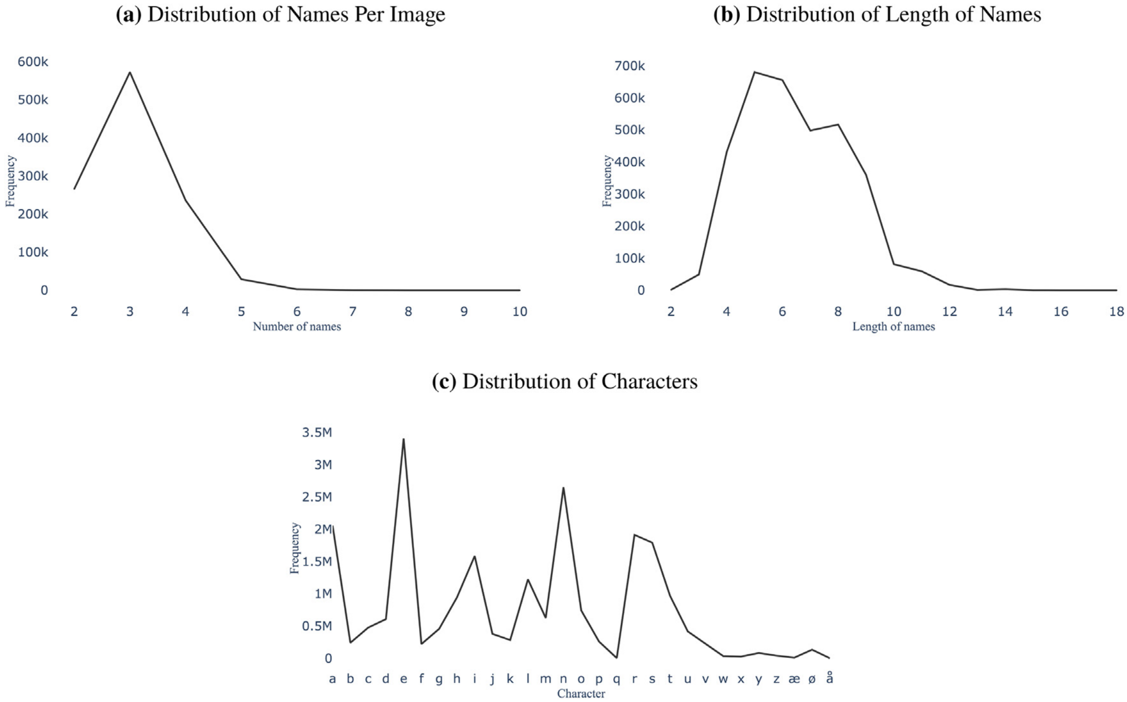

Fig. C1. Further Database Characteristics: Panel C.1a shows the distribution of names per image file. As shown, the majority of images contain two to four names; the longest full name consists of 10 separate names. Panel C.1b shows the length of names per image. The name is in this context defined as each word in a full name, i.e. either first, middle, or last name. The longest name consists of 18 characters with most names being somewhere between 3 and 10 characters long. Panel C.1c shows the distribution of characters in the names. As seen, the most frequent character used in the names is *e*, which appears approximately 3.4 million times, while both *q* and appear fewer than 5000 times. All panels aggregates across all individuals present in either the train or test data.

training set with and without transfer learning, respectively. For the models trained on the large training set, we obtain WACCs of 85.9% and 83.7% with and without transfer learning, respectively. Hence, the performance is approximately one percentage point lower than the WACC we obtained using the full test set (except for the small without transfer learning where the difference is only 0.35%). It therefore seems as though these names are more difficult on average than the full US census test set. If we only look at the increase in performance from transfer learning, we find reductions in the error rates of 23.7% for the small model and 13.6% for the larger model. For the original models, the error rate reductions are 21.7% and 14.2%. We conclude from this that it does not seem to be important whether the names in the transfer learning task contains the same names as the HANA database. Rather, it seems that transfer learning is beneficial either way, and the lower performance gain obtained compared to on the Danish census data is more likely due to factors such as larger differences in handwriting style. For completeness, Fig. B.1 shows the plots corresponding to Panel 7b of Fig. 7, but restricted to the test set of the US census that consists of names also present in the HANA database.

Appendix C. Further characteristics of the database

In this appendix, we present additional characteristics of the HANA database. Fig. C.1 shows the distribution of individual names for each full name, the length of the names, and the distribution of characters. Panel C.1a shows the number of names per image. While most individuals have either just a first and last name, potentially in combination with a single middle name, a significant number have more than one middle name. Panel C.1b shows the length of each name. This should be interpreted as being the length of either the first, middle, or last name. The longest single name in our database is 18 characters. It is important to note that this figure does not represent the full name sequence length, as this could potentially be 10x18 characters long. The final distribution plot in Panel C.1c shows the character distribution aggregating across all names.

References

Abramitzky, R., Boustan, L.P., Eriksson, K., 2012. Europe's tired, poor, huddled masses: self-selection and economic outcomes in the age of mass migration. Am. Econ. Rev. 102 (5), 1832–1856.
Abramitzky, R., Boustan, L.P., Eriksson, K., 2013. Have the poor always been less likely to migrate? Evidence from inheritance practices during the age of mass migration. J. Dev. Econ. 102, 2–14.
Abramitzky, R., Boustan, L.P., Eriksson, K., 2014. A nation of immigrants: assimilation and economic outcomes in the age of mass migration. J. Polit. Economy 122 (3), 467–506.

Abramitzky, R., Boustan, L.P., Eriksson, K., 2016. Cultural Assimilation During the Age of Mass Migration. Working Paper 22381. National Bureau of Economic Research doi:10.3386/w22381. http://www.nber.org/papers/w22381

Abramitzky, R., Boustan, L.P., Eriksson, K., Feigenbaum, J., Pérez, S., 2021. Automated linking of historical data. J. Econ. Lit. 59 (3), 865–918. https://siepr.stanford.edu/research/publications/automated-linking-historical-data

Abramitzky, R., Mill, R., Pérez, S., 2020. Linking individuals across historical sources: a fully automated approach. Hist. Methods J. Quant.Interdiscip. Hist. 53 (2), 94–111. https://www.tandfonline.com/doi/full/10.1080/01615440.2018.1543034

Archives, C., 2022. Politiets mandtaller. https://kbharkiv.dk/brug-samlingerne/kilder-paa-nettet/politiets-mandtaller. Accessed: 2022-08-02.

Bailey, M., Cole, C., Henderson, M., Massey, C., 2020. How well do automated linking methods perform? Lessons from U.S. historical data. J. Econ. Lit. 58 (4), 997–1044.

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-D shapes. In: Sensor fusion IV: Control Paradigms and Data Structures, Vol. 1611, pp. 586–606.

Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V., 2020. RandAugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702–703.

Dahl, C. M., Johansen, T. S. D., Sørensen, E. N., Westermann, C. E., Wittrock, S., 2021. Applications of machine learning in document digitisation. arXiv preprint arXiv:2102.03239.

Dahl, C. M., Johansen, T. S. D., Sørensen, E. N., Wittrock, S., 2022. HANA. https://github.com/TorbenSDJohansen/HANA. Accessed: 2022-08-08.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.

Feigenbaum, J.J., 2018. Multiple measures of historical intergenerational mobility: iowa 1915 to 1940. Econ. J. 128 (612), F446–F481. doi:10.1111/ecoj.12525.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V., 2013. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082.

Harris, C.G., Stephens, M., 1988. A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference, pp. 147–151.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Kim, I., 2020. PyTorch-RandAugment. https://github.com/ildoonet/pytorch-randaugment. Accessed: 2022-08-02.

Massey, C.G., 2017. Playing with matches: an assessment of accuracy in linked historical data. Hist. Methods J. Quant.Interdiscip. Hist. 50 (3), 129–143.

Myronenko, A., Song, X., 2010. Point set registration: coherent point drift. In: 2010 IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE, pp. 2262–2275.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037.

Price, J., Buckles, K., Van Leeuwen, J., Riley, I., 2019. Combining Family History and Machine Learning to Link Historical Records. Working Paper 26227. National Bureau of Economic Research doi:10.3386/w26227. http://www.nber.org/papers/w26227

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning, pp. 1139–1147.

Szeliski, R., 2010. Computer Vision: Algorithms and Applications. Springer Science & Business Media.

Van Rossum, G., Drake, F.L., 2009. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA.