Handwritten information extraction from historical census documents

Thibauld Nion, Farès Menasri, Jérôme Louradour
Cédric Sibade, Thomas Retornaz, Pierre-Yves Métaireau, Christopher Kermorvant
A2iA
39 rue de la Bienfaisance, Paris, France

***Abstract*—This paper describes a complete system for handwritten information extraction in historical documents. The system was evaluated in real conditions and at a large scale (8 millions of snippets) on the tables of the 1930 US Census. The location of the table position was based on a registration algorithm using printed word anchors. The rows and columns were extracted for nine different fields. For each field, a recognizer based either on convolutional neural networks for small lexicon fields or recurrent neural networks for large lexicon fields were trained. This system yields very high results for data extraction, allowing to achieve more than 70% of automation rate at a error rate similar to human keyers for a complete identity field.**

***Keywords*-Historical document processing, Document layout analysis, Handwriting recognition, convolutional and recurrent neural networks.**

I. Introduction

This paper reports the results of the large scale evaluation (8 millions of snippets) of a complete system for handwritten data extraction from historical US census documents as shown of Figure 1. This evaluation was proposed by FamilySearch, a genealogical organisation, in order to investigate how an automatic transcription system can be used to index their image database [1].

Recognition of tables in historical documents has been extensively studied, since a large amount of information is available in this form in historical documents (indexes, census tables, inventories) and the access to this information is easier than full text [2].

Different kinds of recognition techniques have been proposed to recognize tables in historical documents: techniques based on a formal language of document description [3], techniques based on statistical machine learning algorithms [4], or bottom-up techniques, starting from low level features such as table lines to extract the structure [5].

Regarding the recognition of the information found in tables, the approaches based on word spotting [6] cannot be used when all the data in the table must be recognized and indexed.

The system developed for this evaluation was based on a system previously used for information extraction in old French census documents [7]. The document layout analysis system was improved with a table registration algorithm and extended to extract information in ten different data fields. Regarding the handwriting recognition system, the main improvement was due to the use of recurrent and convolution neural networks trained on a large amount of data.

Thibauld Nion is now with DxO Labs, Paris. Farès Menasri is now with Criteo, Paris.

II. Document Layout analysis

The analysis of the document layout was composed of two steps: first the location of the table was found on the page using a registration algorithm, then the table was decomposed into columns, lines and cells with the algorithm described in [7].

The registration algorithm was based on the definition of printed text anchors and the position of the table relatively to the anchors. A list of possible anchors was produced by the OCR transcription of the page. The anchors were selected as stable and discriminant words or group of words at several locations on the page. The position of the table was then registered relatively to the anchors. During the recognition phase, the anchors are found on the documents and the table is detected relatively to the detected anchors. Since several text anchors are used, the registration process is robust to noise and deformation on the documents, such as translation and global skew.

The cell extraction algorithm was previously described in [7] and is summarized in Figure 2. Note that at each stage, the process can be stopped if the information cannot be extracted or is unreliable. In this case, the document, line or cell can be manually processed.

The table registration and the cell extraction were also used to produce annotated images for the training of the neural networks on which were based the handwriting recognizers.

III. Handwriting recognition with neural networks

In this evaluation, the handwritten fields to be automatically recognized in census tables can be grouped in two types:

1) fields with target symbols within a small lexicon (*e.g.* check boxes, codes), and

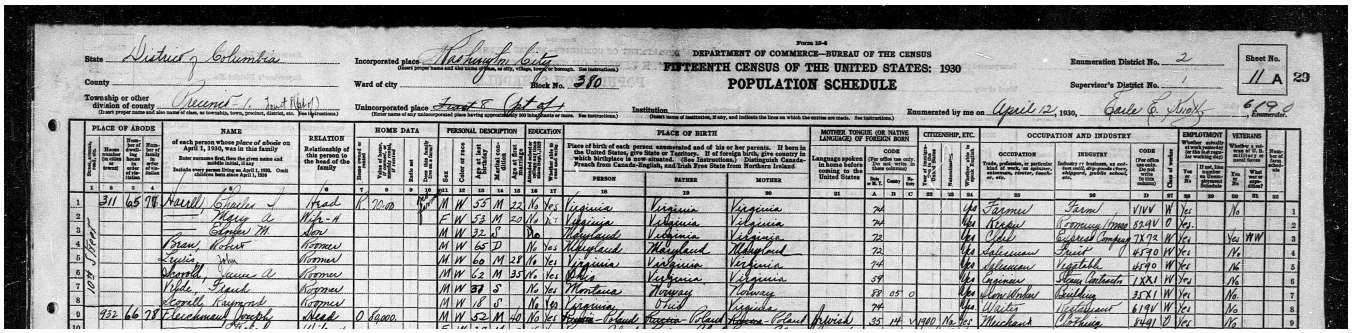


Figure 1. The upper part of a 1930 US Census document.

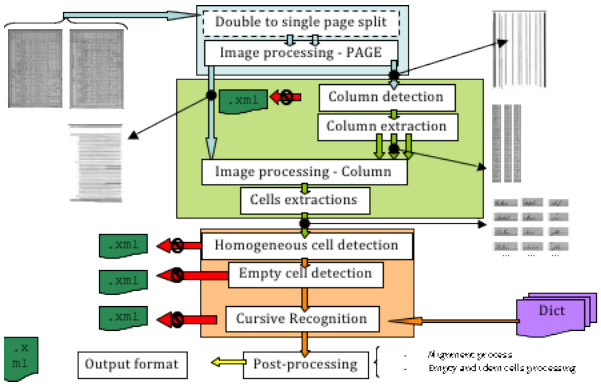


Figure 2. Page layout analysis

2) fields with target words in a very large vocabulary (*e.g.* surnames, year).

Sub-section III-A presents the classification model used to identify the first ones. III-B presents the handwritten word recognition system used to deal with large vocabularies.

A. Classification of 2D shapes (small lexicon)

1) Convolutional Neural Networks (ConvNN): Convolutional Neural Networks [8] are very appealing models for classification of 2-Dimensional shapes, for at least two reasons. Firstly, because they permit to exploit thousands or millions of labelled data thanks to well-mastered optimization technique. Namely, Stochastic Gradient Descent (SGD) is a powerful technique to effectively find minima of non-convex costs, and has a convergence rate that is roughly independent of the size of the database. Secondly, ConvNN are more likely to generalize well on unseen data, in comparison with other neural networks involving full connections between layers. Their highly competitive generalization performance in Vision challenges is due to the presence of 2D convolutions and sub-sampling layers. Alternating these two layers is a good mean to make use of the prior knowledge that inputs are images and to keep the 2D structure in the intermediate layers. Convolutions can be viewed as shape detectors applied on fixed-size 2D windows that browse the image with overlapping. All these hidden non-linear features are learned to work all together in combination. By averaging the features on small zones, sub-sampling layers provide invariability to small translations *i.e.* tolerance to localization imprecision. When the task is to classify fixed-size shapes, the output layer of a ConvNN is a non-linear normalization of unbounded activations so that the network's outputs can be used as probabilities. SGD is used to minimize the Negative Log-Likelihood (NLL) considering these outputs as class posterior probabilities.

2) Network topology: Choosing the topology of the ConvNN plays a crucial part in obtaining good generalization performance along with effectiveness. Regarding the number of convolutional filters in each hidden layer, there must be a balance between having enough hidden features to be able to solve the optimization problem (on training data) and being able to generalize efficiently and achieve fast classification at the same time. Choosing the sizes (height and width) of the convolutional and sub-sampling windows is more a matter of fitting the size and the resolution of the input images. Special care must be taken so as to avoid throwing away any relevant information: the biggest symbol in the database should be encompassed in the ConvNN input window, as well as additional borders around so that any stroke end-point and corner appears in the center of the receptive field of the highest-level feature detectors as recommended by [8].

To recognize fields such as characters, check boxes and words within a small lexicon, we use a ConvNN that is similar to LeCun's LeNet 5. The only difference was the convolutional and sub-sampling window sizes that are chosen to fit the resolution and width/height ratio of each cell to be recognized.

3) Splitting the database: In order to validate some parameters related to the optimization procedure (learning rate of the SGD, early-stopping w.r.t the number of udpates, . . .) and to assess the performance of our field recognizers, we need to split all the available labelled data into at least three disjunct subsets. We observed that splitting the database without taking into account meta-information about

how each sample was produced could lead to an important bias in the evaluation of the modelling accuracy. Namely, tables filled by a same scribe should not appear both in the training dataset and in the evaluation dataset. For the sake of simplicity we grouped the scribes per US state, and split the database into $K+1$ sub-databases, each one corresponding to a set of a few states (not represented in the other sub-databases). We use the sub-databases in a K-fold cross validation scheme whose interest is twofold: (1) making accurate performance assessment (no bias because of the arbitrary selection of states), and (2) enabling to combined the K trained neural networks in order to improve the overall performance. Note that the gain provided by the model combination must be assessed on a $(K+1)^{th}$ sub-database with some isolated states.

B. Cursive words recognition (large vocabulary)

1) Recurrent Neural Networks (RNN): Although ConvNN are fast and achieve the best performance to classify fixed-size images, they are not designed to recognize variable-length sequences of symbols within a variable-size images, when the alignment between the target sequences and the regions of input images are unknown. This is the case when the goal is to recognize a word or a number (harder than recognizing a character or a digit in a localized cell). [9] gives an effective and robust Connectionnist Temporal Classification (CTC) training procedure to optimize word recognition using space-displacement neural networks. It consists in computing the gradient of the NLL on the target sequence by summing the contribution of all possible alignments with the neural networks output activations.

RNN with layers of 2-Dimensional Long-Short Term Memory units (LSTM) [10], [11] are the state-of-the-art neural networks to solve handwritten recognition. These special recurrent neurons enable to learn correlations between locations in the image that can be close (within a stroke or a grapheme) or more distant (one or two characters). LSTM layers are used in alternation with parametrized convolutional layers [10].

2) Learning RNN on degraded documents: As the ConvNN, the RNN are trained using SGD and several RNN are combined with the methodology described in III-A3. However, optimizing RNN is much harder than optimizing ConvNN, because of the long chain of non-linear functions involved in the recurrences. We observed that SGD succeeds to train RNN on databases of documents with a certain quality, but that it is hopeless to run SGD on a on a database with only degraded documents, starting from a random initialization of RNN free parameters. A good solution is to adapt a RNN that has already been trained to learn word recognition with good quality images. This idea of starting to learn with simple and carefully selected samples before switching to harder samples of the real world has already been recommended for SGD under the name "curriculum" [12].

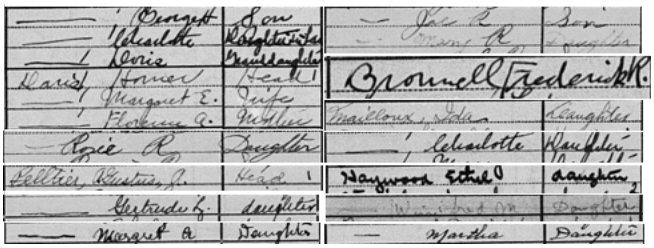


Figure 4. Examples of writing variability.

3) Process of RNN outputs with a large vocabulary: A specific model had to be designed for the recognition of the identity field. This field was composed of the surname, the first name and the initials of the person, in one cell written without separation. The recognizer had to be able to recognize and to type each element using a very large dictionary.

We used a weighted finite state transducer as a syntactic constraint for the identity field. For each US state, a weighted list of surnames and first names was used to define the transducer: the transitions are labelled by the surname or first name and the weights are computed using the weight of the list ($-log(probability)$). The initials are composed of the 26 letters of the alphabet, with equal weight.

The different recognizers were combined after the application of the identity field transducer with a ROVER algorithm [13].

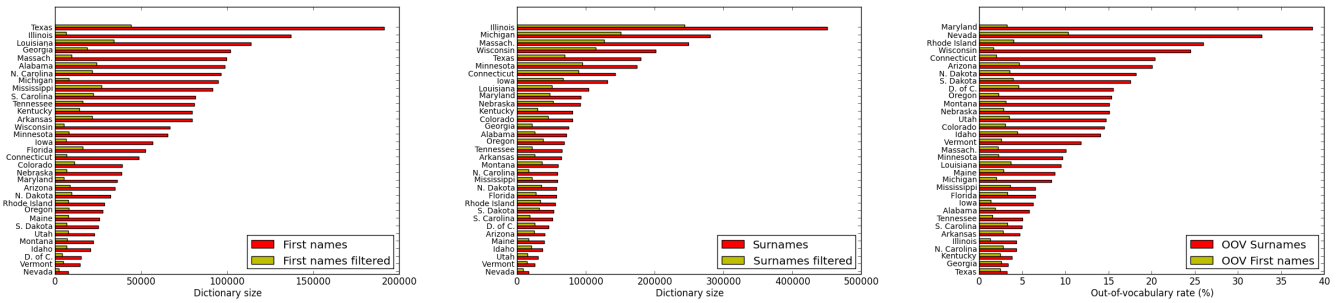
IV. Experiments

A. 1930 US Census database

The scope of this evaluation was the transcription of documents from the 1930 US Census collection. As shown on Figure 1, a 1930 US census sheet consisted in a printed table in landscape format with a 32 columns per 50 lines table and a header. The data to be extracted was handwritten, cursive style or hand printed for numerical and single letter fields. The 1930 US census training data set included 14639 images,with an average size of 4100*2800 pixels, scanned at 200 DPI.

The cells which correspond to handwritten fields to be recognized are given in the following list, sorted by their column index in the original tables (see Figure 1):

5. Identity: surname, given name, and sometimes middle initial and/or titles (“Sr”, “Jr”, . . .).
6. (*) Relationship of the person to the head of the family (with a few possible short words: “Wife”, “Son”, . . .).
11. (*) Sex code: ‘M’(ale) or ‘F’(emale).
12. (*) Skin color code: ‘W’(hite), ‘B’(lack), . . .
13. (*) Age: *e.g* “3/12”, “1 6/12”, “7”, “46”, . . .
14. (*) Marital status code: ‘M’(arried) or ‘S’(ingle).
18. Birth place: a country name ($''$).



(a) (b) (c)

Figure 3. For each US state, dictionary sizes for first names and surnames before and after filtering and coverage (out-of-vocabulary rate) of the filtered dictionary.

19. Father's birth place (″).
20. Mother's birth place (″).
22. Immigration year (empty when the birth place is US).

All fields marked with (*) are related to a *small lexicon* and are processed using ConvNN as described in III-A. All the others involve *large vocabularies* and are processed using RNN as described in III-B. The (″) indicate three cells for which we share the same recognizer dedicated to country names.

1) Images: The main challenges of this task were the different writing styles due to the wide diversity of census takers as shown on Figure 4: the tautness of the writing in the cells, the type of pens which can be thick or not, dark or light , the different quality of ink sometimes on the same sheet (or different color), the presence of strokes, either on the whole page or for some lines only.

2) Dictionaries: The two main sources for the name and first name dictionaries were the US Census from 1920 and 1930. Using the data from 1920 US Census allowed us to evaluate a realistic situation in which the coverage of the name dictionaries is not complete. Moreover, we have filtered the 1920 dictionaries to remove very rare forms which may correspond to transcription errors. The size of the dictionaries and their coverage for each state are given on Figure 3.

Each time it was possible and relevant, we created dictionaries for each state rather than dictionaries for the whole US territory so that the specificities of each state could be taken into account. Furthermore this made it possible to take advantage of the different occurrences distribution. A simple dictionary with entry from A to Z was built for the middle initial recognition.

3) Metrics: The goal of the system presented in this paper was to replace a proportion of the manual data entry for the 1930 US Census indexation. Since the recognition results are associated with a confidence level, the documents or cells for which this confidence level is too low are sent to manual keying. The threshold on the confidence score defines a compromise between the automation rate and the error rate, called a working point. In order to define this working point, a automation/error curve is plotted: for each value of the threshold (between 0 and 1000 for example) the automation rate and the error rate are computed. All the recognition results in this evaluation are given with the automation/error curves.

B. Results

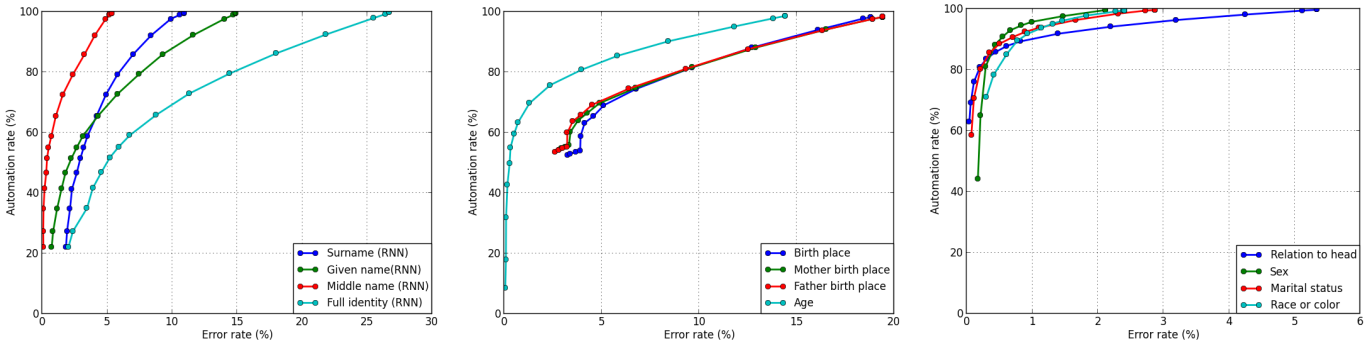
The recognition results are presented for each field in Figure 5.

The identity field is the most important field since it is used to index the documents. As shown in Figure 5a, at full automation rate, the error rate of the surname and first name field are at 10% and 15% respectively, which is quite low considering the size of the lexicon and the variability and the quality of the images. The full identity field, composed of surname, first name and middle name shows an error rate at 26% at full automation rate. For an automation rate of 70%, the error rate decreases at 10%, which is roughly the human error rate on this task [1]. This result proves that our system can reduce significantly the manual keying in the indexation of this kind of document and can fully replace one keyer in a double keying setting.

The age and birth place field present a relatively high error rate at full automation rate, considering that the possible values are limited (see Figure 5b). But the error rate drops rapidly as the automation rate decreases. Finally, as shown in Figure 5c, the error rate for the small lexicon fields, recognized with convolutional neural networks, is low (between 2% adn 6% error rate at full automation rate) and less than 1% error rate can be reached at 90% automation rate, which is very high.

V. Conclusion

In this paper, a complete system for the automatic recognition of handwritten fields in the 1930 US Census documents is described. This system combines a table registration based



(a) (b) (c)

Figure 5. Automation/error curves for the different fields for the two recognizer 1) Recurrent Neural Networks: identity field, composed of surname, first name and middle name or initials, birth place of the person and both his/her mother and father 2) Convolutional Neural Networks: age, marital status, relation to head, sex, race or color.

on printed keywords, the location of each cells with column and line extraction in the table and the automatic recognition of 9 handwritten fields with recurrent and convolutional neural networks. For the main field of the table, the identity field, the performance of this system allows to reach 70% of automation rate for an error rate comparable to a human keyer. The performance of the systems on the other fields of interest is also very high, since less than 1% error rate can be reached at 90% automation rate. In a double keying setting used for high accuracy indexing, this level of performance could reasonably serve as a replacement of one of the keyers. Moreover, the bounding boxes could be useful for producing the snippets needed for the manuel keying.

VI. Acknowledgment

The authors thank FamilySearch for organizing and funding the IRIS evaluation campaign for which the system presented in this paper was developed.

References

[1] P. Schone, H. Nielson, and M. Ward, "Evaluation of Handwriting Recognition Systems for Application to Historical Records," in *Annual Family History Technology Workshop*, 2013.

[2] M. Bulacu, R. Van Koert, L. Schomaker, and T. Van Der Zant, "Layout Analysis of Handwritten Historical Documents for Searching the Archive of the Cabinet of the Dutch Queen," in *International Conference of Document Analysis and Recognition*, vol. 1, 2007, pp. 357–361.

[3] B. Coüasnon, J. Camillerapp, and I. Leplumey, "Access by content to handwritten archive documents: generic document recognition method and platform for annotations," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 223–242, Mar. 2007.

[4] K. Laven, S. Leishman, and S. Roweis, "A statistical learning approach to document image analysis," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2005, pp. 357–361.

[5] H. Nielson and W. Barrett, "Consensus-based table form recognition of low-quality historical documents," *International Journal on Document Analysis and Recognition*, vol. 8, no. 2-3, pp. 183–200, Feb. 2006.

[6] R. Manmatha and T. M. Rath, "Indexing of Handwritten Historical Documents - Recent Progress," in *Proc. of the Symposium on Document Image Understanding Technology*, 2003, pp. 77–85.

[7] C. Sibade, T. Retornaz, T. Nion, R. Lerallut, and C. Kermorvant, "Automatic indexing of French handwritten census registers for probate genealogy," in *First International Workshop on Historical Document Imaging and Processing*. ACM, 2011, pp. 51–58.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine learning*. ACM, 2006, pp. 369–376.

[10] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural Information Processing Systems*. MIT Press, 2008, pp. 545–552.

[11] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.

[12] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *International Conference on Machine Learning*, no. 1330. ACM Press, 2009, pp. 1–8.

[13] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997, pp. 347–354.