Election Tally Sheets Processing System

Juan Ignacio Toledo*, Alicia Fornés†, Jordi Cucurull*and Josep Lladós†
*Scytl Secure Electronic Voting
Barcelona, Spain
Email: {JuanIgnacio.Toledo,Jordi.Cucurull}@scytl.com
†Computer Vision Center
Universitat Autònoma de Barcelona
Barcelona,Spain
Email: {afornes,josep}@cvc.uab.cat

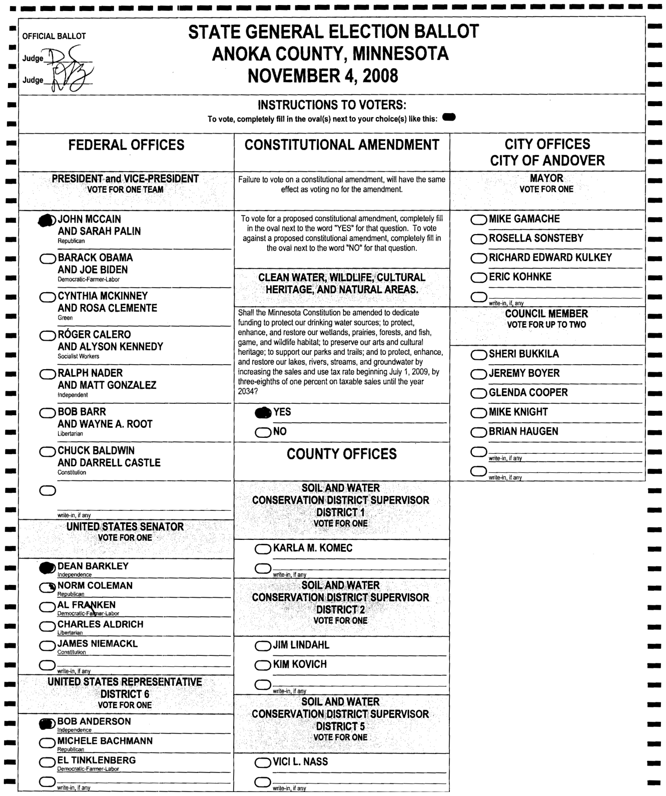
***Abstract*—In paper based elections, manual tallies at polling station level produce myriads of documents. These documents share a common form-like structure and a reduced vocabulary worldwide. On the other hand, each tally sheet is filled by a different writer and on different countries, different scripts are used. We present a complete document analysis system for electoral tally sheet processing combining state of the art techniques with a new handwriting recognition subprocess based on unsupervised feature discovery with Variational Autoencoders and sequence classification with BLSTM neural networks. The whole system is designed to be script independent and allows a fast and reliable results consolidation process with reduced operational cost.**

I. Introduction

Over 300 nationwide elections are held yearly. This sums up to approximately 3600 million registered voters per year, with an average spending of 5 USD per voter, elections constitute a potential market of 18 billion dollars per year. Most of these elections are paper based and they produce a huge amount of documents suitable for automatic processing.

There are mainly two different approaches to electronically produce election results in paper based elections. The first approach is to detect the marks made by voters to each ballot (See Figure 1). Optical Scan machines have been used for that purpose since the 50's. It can be considered a solved problem if we know where to look for the mark and users are required to use a prescribed mark (i.e filling in an oval). That is the case in educational testing, that was the first field to develop Optical Scanners. Nowadays, there is little on this subject, and it is focused on building low cost alternatives to optical scan machines with non-dedicated devices. A threshold on the average gray level of each target area can be enough to decide if it's filled or empty [10].

Since most electoral laws allow different kind of marks besides the prescribed mark (like X or check marks), Optical Mark Recognition becomes more challenging. A first approach to detect marks is performing a difference between previously aligned marked ballots and a blank ballot, usually with some preprocessing like smoothing or mathematical morphology operations [12], [14]. Instead of just detecting a dark patch, we could make the assumption that each voter makes coherent marks in a single ballot and train specific classifiers for X, check marks, or filled ovals in order to increase the accuracy of our mark detector [18]. There has also been some research on how to detect the position of the voting targets. Some authors propose detecting a grid for possible positions of marks by analyzing the geometry of the ballot [13]. Others simply require user collaboration to tag a blank voting target and then locate the rest using pattern matching techniques like Lucas-Kanade; after knowing where voting targets are, they sort them by the number of dark pixels and ask the user to select a boundary [6], [16].



OFFICIAL BALLOT
Judge
Judge

STATE GENERAL ELECTION BALLOT
ANOKA COUNTY, MINNESOTA
NOVEMBER 4, 2008

INSTRUCTIONS TO VOTERS:
To vote, completely fill in the oval(s) next to your choice(s) like this: ●

FEDERAL OFFICES

PRESIDENT and VICE-PRESIDENT
VOTE FOR ONE TEAM

● JOHN MCCAIN
AND SARAH PALIN
Republican

○ BARACK OBAMA
AND JOE BIDEN
Democratic-Farmer-Labor

○ CYNTHIA MCKINNEY
AND ROSA CLEMENTE
Green

○ RÓGER CALERO
AND ALYSON KENNEDY
Socialist Workers

○ RALPH NADER
AND MATT GONZALEZ
Independent

○ BOB BARR
AND WAYNE A. ROOT
Libertarian

○ CHUCK BALDWIN
AND DARRELL CASTLE
Constitution

○ write-in, if any

UNITED STATES SENATOR
VOTE FOR ONE

● DEAN BARKLEY
Independence

○ NORM COLEMAN
Republican

○ AL FRANKEN
Democratic-Farmer-Labor

○ CHARLES ALDRICH
Libertarian

○ JAMES NIEMACKL
Constitution

○ write-in, if any

UNITED STATES REPRESENTATIVE
DISTRICT 6
VOTE FOR ONE

● BOB ANDERSON
Independence

○ MICHELE BACHMANN
Republican

○ EL TINKLENBERG
Democratic-Farmer-Labor

○ write-in, if any

CONSTITUTIONAL AMENDMENT

Failure to vote on a constitutional amendment, will have the same effect as voting no for the amendment.

To vote for a proposed constitutional amendment, completely fill in the oval next to the word "YES" for that question. To vote against a proposed constitutional amendment, completely fill in the oval next to the word "NO" for that question.

CLEAN WATER, WILDLIFE, CULTURAL HERITAGE, AND NATURAL AREAS.

Shall the Minnesota Constitution be amended to dedicate funding to protect our drinking water sources; to protect, enhance, and restore our wetlands, prairies, forests, and fish, game, and wildlife habitat; to preserve our arts and cultural heritage; to support our parks and trails; and to protect, enhance, and restore our lakes, rivers, streams, and groundwater by increasing the sales and use tax rate beginning July 1, 2009, by three-eighths of one percent on taxable sales until the year 2034?

● YES
○ NO

COUNTY OFFICES

SOIL AND WATER
CONSERVATION DISTRICT SUPERVISOR
DISTRICT 1
VOTE FOR ONE

○ KARLA M. KOMEC
○ write-in, if any

SOIL AND WATER
CONSERVATION DISTRICT SUPERVISOR
DISTRICT 2
VOTE FOR ONE

○ JIM LINDAHL
○ KIM KOVICH
○ write-in, if any

SOIL AND WATER
CONSERVATION DISTRICT SUPERVISOR
DISTRICT 5
VOTE FOR ONE

○ VICI L. NASS
○ write-in, if any

CITY OFFICES
CITY OF ANDOVER

MAYOR
VOTE FOR ONE

○ MIKE GAMACHE
○ ROSELLA SONSTEBY
○ RICHARD EDWARD KULKEY
○ ERIC KOHNKE
○ write-in, if any

COUNCIL MEMBER
VOTE FOR UP TO TWO

○ SHERI BUKKILA
○ JEREMY BOYER
○ GLENDA COOPER
○ MIKE KNIGHT
○ BRIAN HAUGEN
○ write-in, if any
○ write-in, if any

Fig. 1. An example of a ballot to be processed with OMR technology.

Another very extended way of conducting paper based elections consists in a manual tally at polling station level. After performing the tally, election officers at each polling station must fill in and sign a tally sheet document that will be the base for the results consolidation process. A tally sheet is a form-like document combining printed information such as text, barcodes or ROI marks with handwritten text or digits (See Figure 2). Several electoral commissions from different countries have shown interest in a system that can reduce the time required to process all the tally sheets that can seamlessly integrated into their traditional election processes. To our knowledge, a document analysis system specially designed to process handwritten tally sheets has not been described in the literature, so we decided to design one. Such a system should be able to deal with an extreme multi-writer scenario, given that each tally sheet will be written by a different writer we might have to deal with tens of thousands of different writers, on a country-wide election. In addition, the system should also be able to work with different scripts.

In this paper we present a system that is able to automatically process this kind of document, reducing the operational costs and greatly speeding up the results consolidation process. Section 2 is an overview of the whole system, from the preprocessing steps performed on the original scanned image to the digit and handwriting recognition. In Section 3 we discuss the main contribution of the paper, a combination of unsupervised feature learning with a Bidirectional Long Short Term Memory recurrent neural network for handwriting recognition. In Section 4 we show the experiments performed and the results obtained. Finally we present some conclusions and future work on Section 5.

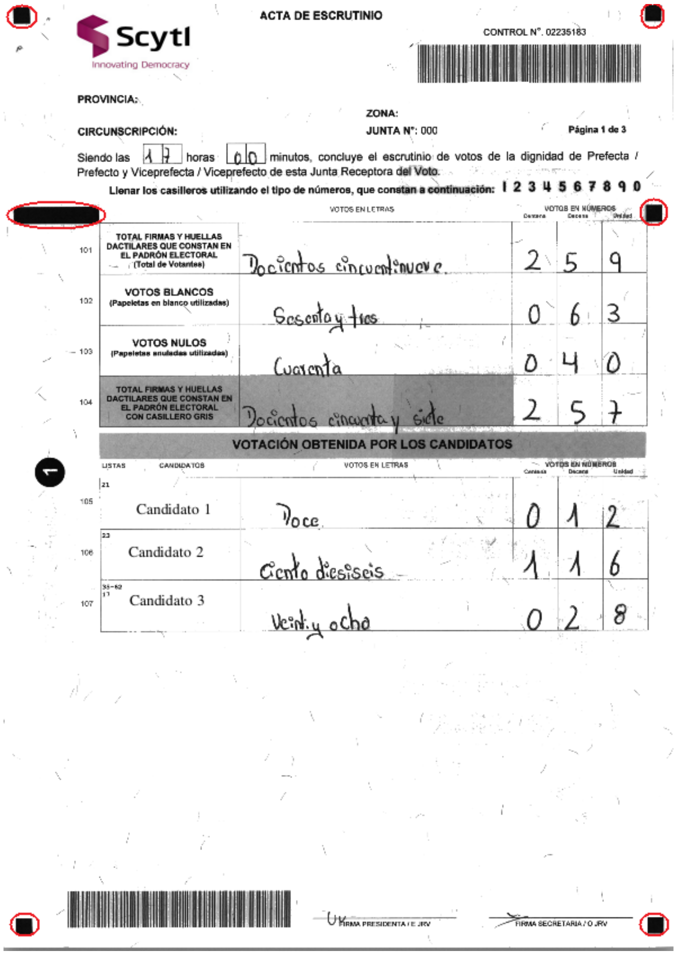
II. System Description

In this section, we present a system for tally sheet processing. Each tally sheet page is uniquely identified by a bar code, this allows us to retrieve from a database the candidates corresponding to each line of the tally sheet. From a document analysis perspective, it is only needed to extract all the lines in each tally sheet, since we already have the mapping to the candidates from the database. Finally, for each line, the goal is to recognize both the handwritten text and the digits.

A. Preprocessing

The preprocessing process consists of skew removal and the extraction of the different lines corresponding to each one of the different candidates. In order to perform the skew correction we will look at the different fiducial marks on the document (See Figure 2. The biggest fiducial mark allows us to detect if the page is upside down, while the smaller marks are used to correct rotations.

1) Orientation and skew removal: The first thing to check is the size of the image. Since we are working with vertical tally sheets, we expect the height of our image to be larger than its width. If this is not the case (possibly due to wrong scanner configuration), the image has to be rotated by 90 degrees. Once we have a vertical image we perform a template matching with a rectangle of size 244x64 pixels in order to detect the biggest fiducial mark. If it is found on the first quadrant of the image we already have the correct orientation while if it is found on the fourth quadrant, we must rotate our image by 180 degrees. This step is required because images scanned upside-down are a fairly common.



ACTA DE ESCRUTINIO

Scytl
Innovating Democracy

CONTROL N°. 02235183

PROVINCIA:

ZONA:

CIRCUNSCRIPCIÓN: JUNTA N°: 000 Página 1 de 3

Siendo las 17 horas 00 minutos, concluye el escrutinio de votos de la dignidad de Prefecta / Prefecto y Viceprefecta / Viceprefecto de esta Junta Receptora del Voto.

Llenar los casilleros utilizando el tipo de números, que constan a continuación: 1 2 3 4 5 6 7 8 9 0

		VOTOS EN LETRAS	VOTOS EN NÚMEROS Centena	Decena	Unidad
101	TOTAL FIRMAS Y HUELLAS DACTILARES QUE CONSTAN EN EL PADRÓN ELECTORAL (Total de Votantes)	Docientos cincuentinueve.	2	5	9
102	VOTOS BLANCOS (Papeletas en blanco utilizadas)	Sesenta y tres	0	6	3
103	VOTOS NULOS (Papeletas anuladas utilizadas)	Cuarenta	0	4	0
104	TOTAL FIRMAS Y HUELLAS DACTILARES QUE CONSTAN EN EL PADRÓN ELECTORAL CON CASILLERO GRIS	Docientos cincuenta y siete	2	5	7

VOTACIÓN OBTENIDA POR LOS CANDIDATOS

	LISTAS	CANDIDATOS	VOTOS EN LETRAS	VOTOS EN NÚMEROS Centena	Decena	Unidad
105	21	Candidato 1	Doce	0	1	2
106	23	Candidato 2	Ciento diesiseis	1	1	6
107	35-62 17	Candidato 3	Veinti y ocho	0	2	8

1

FIRMA PRESIDENTA / E JRV

FIRMA SECRETARIA / O JRV

Fig. 2. The fiducial marks in a tally sheet (highlighted in red) used to detect the orientation and skew also allow us to segment the Region of Interest for later handwriting and digit recognition steps.

The next step is finding the smaller marks in order to fix the skew. Using a convolution of the negated image and models of the different fiducial marks we can detect them. If we can not find all of the fiducial marks (usually due to the paper being misplaced on the scanner or partially folded) an error is raised. Using the Hough transform on the image containing only the detected fiducial marks we can find the skew and correct it if necessary.

2) Region of interest extraction and noise removal: Once we have corrected the orientation and the skew of the image, we can select the area delimited by the big fiducial mark and the ones on the lower part of the sheet. This area will be then divided in several lines using a fixed parameter. And each line will be again divided into different fields. Noise removal based on morphological operations is then applied on each one of the negated images of these regions. First we perform an opening with a 3x3 square structuring element, in order to remove noise from the image. On the result we apply a dilation with a rectangular structuring element to obtain the minimum number of connected components of 23 pixels height and width depending on the region. Finally we perform an opening with a 31x31 structuring element to remove regions that can not be considered digits or characters. Finally we will keep the regions with height bigger than a threshold t1=40 pixels and a width bigger than one fifth of the region. These parameters have been determined empirically.

B. Intelligent Character Recognition

The Intelligent Character Recognition subsystem receives an individual character image from the cropping module and computes a description of the image based on Histogram of Oriented Gradients (HOG) features [5]. The feature description is then fed into a support vector machine based classifier that predicts the most likely with a confidence measure on that prediction.

C. Handwriting Recognition

The most challenging scenario is handwriting recognition due the huge variation in writing style. In order to be able to deal with the variation from thousands of writers and different scripts we propose a handwriting recognition subprocess based on unsupervised feature discovery with variational autoencoders. The sequence alignment and recognition is performed with BLSTM (Bidirectional Long Short Term Memory) recurrent neural network with CTC (Connectionist Temporal Classification) [3] and the output of the recognizer is mapped to the word of the dictionary with smallest string edit distance. Handcrafted features are usually designed for a specific alphabet and they lack a clear justification.

A similar approach using unsupervised feature learning has been recently published for alphabet independent OCR [11] with promising results. In their case they use Restricted Boltzman Machines, while we use Variational Autoencoders. Both perform similar tasks but following quite different approaches. The approach of Variational Autoencoders allows a faster and simpler training, with traditional backpropagation and has also shown to achieve lower reconstruction error. Figure 3 shows a schematic view of the handwriting recognition process, which is described in more detail in the next section.

III. Handwriting Recognition Subprocess

In this section we discuss one the most important contributions of the paper, an Handwriting Recognition module that performs an unsupervised feature discovery and sequence classification.

A. Unsupervised Feature Learning

Autoencoders are neural networks trained to reproduce its inputs at the output layer. In their most basic implementation, they consist of two layers, the encoder that takes us from image space into an internal representation and the decoder that does the opposite. In the most common scenario we want to learn an internal representation that is a lower dimensional representation of our data. This process can be seen as a feature extraction process [15] and has also been used in deep learning as an ”unsupervised pretraining”. Different architechtures for autoencoders have been proposed recently, denoising autoencoders, sparse autoencoders, convolutional autoencoders, etc. One of the most promising at the moment is the Variational Autoencoder [7], [9] which has yielded impressively low reconstruction error with a really fast training times.

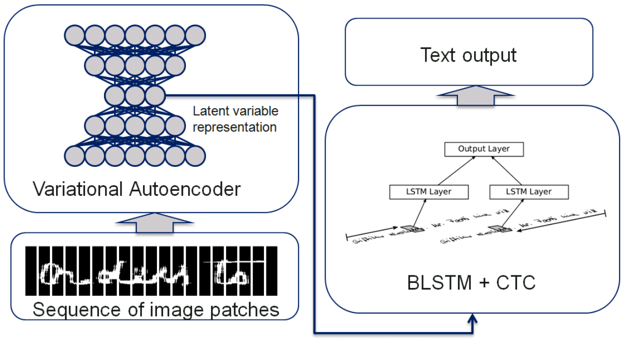


Fig. 3. A sequence of image patches is passed through the encoder of a Variational Autoencoder to get a latent variable representation. This representation is then fed into a bidirectional long short term memory neural network to perform the final recognition.

Variational Autoencoders: In order to learn about the structure of our data $\boldsymbol{x}$, in Variational Autoencoders we assume that it was generated by an unobserved random variable $\boldsymbol{z}$. Since the marginal likelihood $p(\boldsymbol{x}) = \int p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})dz$ is generally intractable, we can use variational inference in order to learn an approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ of the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$.

The log-likelihood of each datapoint can then be expressed as

$$\log p_\phi(\boldsymbol{x}) = \mathrm{KL}(q_{\boldsymbol{z}} \| p_{\boldsymbol{z}|\boldsymbol{x}}) + \mathcal{L}(\theta, \phi; \boldsymbol{x}),$$

where

$$\begin{aligned} \mathcal{L}(\theta, \phi; \boldsymbol{x}) &= \int q_\phi(\boldsymbol{z})(\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z}))d\boldsymbol{z} \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x}))\right] \end{aligned}$$

That is, a sum of the KL divergence term between the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ and our approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, which is always positive, and $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ a lower bound of the log likelihood of our data. Thus our goal will be maximizing this $\mathcal{L}(\theta, \phi; \boldsymbol{x})$. We can do this with standard gradient ascent algorithm using backpropagation thanks to the “reparametrization trick” proposed by the author in the orignal paper [7].

This “reparametrization trick” consists in modeling $q(z|x) \sim \mathcal{N}(\mu(x), \sigma(x)^2)$, and generating random perturbation $\varepsilon \sim \mathcal{N}(0, I)$. By doing so, we are to sample from $z = \mu(x) + \sigma(x)\varepsilon$ in a way that is efficient and appropiate for differentiation with respect to our parameters.

In other words, we use Variational Autoencoders to learn a generative model of character parts or pseudo strokes. And we use the representation of those character parts in the latent variable space $\boldsymbol{z}$ as our features.

We perform a height normalization on all the text lines of our training dataset, and we use a sliding window approach with a step to extract 20 pixel width, 120 pixel high, image patches from our dataset ignoring all label information. We then feed them to a Variational Autoencoder in order to find a lower dimensional latent representation. Once the autoencoder is trained, we can use the encoder weights to move from image space to this generative latent space that will constitute our features. It is worth noting that in this step, each image patch is treated as an independent example of a handwriting pseudo stroke.

B. Sequence Alignment and Recognition

After the unsupervised training has finished we use the same sliding window approach that we used to train the autoencoder to get a sequence of image patches that represents each text line. Each of the image patches is then fed to our encoder to perform forward propagation in order to get a sequence of observations in latent space. Each sequence, along with its transcription is now fed to a Bidirectional Long Short Term Memory (BLSTM) network with a Connectionist Temporal Classification (CTC) output layer [3] in order to learn the transcription.

BLSTM+CTC: Long Short Term Memory networks [4] are a type of recurrent neural networks designed to deal with the vanishing gradient problem by incorporating multiplicative input, output and forget gates, that allow the cells to ignore unimportant inputs keeping their internal state unchanged, making them specially suited for learning over long sequences. Connectionist Temporal Classification [3] is an algorithm that allows the network to perform sequence alignment with differentiable errors. A neural network using LSTM and CTC can be trained using standard backpropagation algorithm. In Offline Handwriting Recognition, for more robustness, two LSMT+CTC neural networks are trained, processing the sequence forwards and backwards.

IV. Experiments

In this section we will show the experiments performed to evaluate the accuracy of our tally sheet processing system.

A. Intelligent Character Recognition

We performed several experiments with support vector machine on histogram of gradients based digit recognition.First we trained with the MNIST dataset [8] using the proposed division of 50.000 digits for training and 10.000 for testing. We got an accuracy of 99% on the MNIST test set. We tested it on our internal dataset from electoral tally sheets, using the process described earlier on the paper to segment the 64596 individual digits from over 1000 different writers. On this dataset we got a 66.55% accuracy, with high percentage of the errors being on the number 7.

A second experiment was performed using both the MNIST and the CVL [1] datasets for training, getting a 98.08% accuracy on MNIST. On the CVL dataset, there are examples of the number seven with a line crossing its long line. When we tested this model on the dataset of real electoral tally sheets the accuracy went up to 89.32%. The results are summarized on Table I. The difference in accuracy is probably due to the fact that the area assigned to digits in the tally sheet is usually much bigger than the digit itself, generating a variation in size and position that was not present in the original training datasets where the segmentation of the digits completely fill the image size.

TABLE I
Average digit error rate on MNIST and TallySheets datasets.

Training Data	MNIST	Tally Sheets
MNIST	1.00%	33.45%
MNIST+ CVL	1.92%	10.38%

B. Handwriting Recognition Experiments

We performed experiments with our Handwriting Recognition process on the George Washington database [17] composed of binarized and normalized text line images written in 18th century English language with two different writers splitting the dataset into train, validation and test. We decided to use the George Washington dataset because it is a standard database that allows us an easy comparison of the results with other state of the art works. The text lines were already normalized to a height of 120 pixels, we extracted individual patches of 20 pixels width with a step size of 4 pixels. Theses patches were used to train a Variational Autoencoder with an internal latent representation of 40 and 80 dimensions for a fixed amount of 100 iterations, which empirically showed to provide a good reconstruction error. The same patches of 120 pixels height and 20 pixels width were presented as a sequence of observations, with their labels to a standard BLSTM network with 100 cells that was trained until no improvement was observed on the validation set for 20 iterations. The experiments were repeated five times in order to reduce the impact of the random initializations of the neural networks. The parameters were selected to match those used by Fischer [2] obtainning very similar results.

The results shown in Table II were similar to the state of the art approach with Marti features using a descriptor of 40 dimensions and slightly better when using an 80 dimensions descriptor. In both cases the uncertainty due to random initializations was greatly reduced. The convergence time improved dramatically both in number of iterations and duration of the iterations, as shown in Table III. The faster convergence is due to the reduction of the length of the sequences, by using one observation every 4 columns instead of each column. The impact of the dimensionality of the features is negligible when compared to the length of the sequence. We would have liked to validate the results with our dataset of real tally sheets, but we lacked the ground-truth associated with the handwritten text field.

TABLE II

AVERAGE CHARACTER ERROR RATE AND STANDARD DEVIATION OVER FIVE DIFFERENT EXPERIMENTS FOR EACH SET OF HYPERPARAMETERS.

Features	Avg CER	std
Marti Features	26.45%	2.12
VAE (40 dim)	26.66%	0.50
VAE (80 dim)	25.58%	0.97

TABLE III

AVERAGE NUMBER OF ITERATIONS REQUIRED FOR CONVERGENCE AND STANDARD DEVIATION OVER FIVE DIFFERENT EXPERIMENTS FOR EACH SET OF HYPERPARAMETERS.

Features	Epochs	std	Epoch time
Marti Features	123.40	15.09	15 min
VAE (40 dim)	59.60	6.41	4 min
VAE (80 dim)	64.40	8.40	4.5min

V. CONCLUSIONS AND FURTHER WORK

We have described an electoral tally sheet document analysis system that covers all the pipeline from the scanned image to the number of votes each candidate receives. We also show a promising handwriting recognition process with unsupervised feature learning using variational autoencoders, and show that it can slightly improve the state of the art Marti Features in the character error rate, while also greatly reducing both the number of epochs needed to convergence and their duration. The uncertainty due to the random initialization is also greatly reduced. As a future work, we would like to use our recently labeled dataset of electoral tally sheet images in order to test our handwriting recognition process with the real data it was designed to work with. We think that there is still room for improvement in the use of autoencoders for handwriting recognition, we would like to explore different autoencoder architectures, trying to find a way to get more discriminative features. We would also like explore more intelligent ways of combining the results of the handwriting recognition and the digit recognition. With some minor adjustments to our digit recognizer it could output several predictions each of them with their associated confidence. We could try to match the result of the handwriting recognition with each of these possible predictions of the digit recognition taking into account their confidence.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish project TIN2012-37475-C02-02, the European project ERC-2010-AdG-20100407-269796, the grant 2013-DI-067 from the Secretaria d'Universitats i Recerca del Departament d'Economia i Coneixement de la Generalitat de Catalunya and the Ramon y Cajal Fellowship RYC-2014-16831.

REFERENCES

[1] Markus Diem, Stefan Fiel, Angelika Garz, Manuel Keglevic, Florian Kleber, and Robert Sablatnig. Icdar 2013 competition on handwritten digit recognition (hdrc 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1422–1427. IEEE, 2013.

[2] Andreas Fischer. *Handwriting recognition in historical documents*. PhD thesis, 2012.

[3] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31(5):855–868, 2009.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Daniel Keysers, Christian Gollan, and Hermann Ney. Local context in non-linear deformation models for handwritten character recognition. In *17th International Conference on Pattern Recognition (ICPR), 2004.*, volume 4, pages 511–514. IEEE, 2004.

[6] Eric Kim, Nicholas Carlini, Andrew Chang, George Yiu, Kai Wang, and David Wagner. Improved support for machine-assisted ballot-level audits. In *Presented as part of the 2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, Berkeley, CA, 2013. USENIX.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR), 2014*, page 1, 2014.

[8] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits.

[9] Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML), 2014)*, pages 1278–1286, 2014.

[10] Rakesh S., Kailash Atal, and Ashish Arora. Cost effective optical mark reader. *International Journal of Computer Science and Artificial Intelligence(IJCSAI)*, 3(2):44–49, Jun 2013.

[11] Devendra Sahu and Jawahar C. V. Unsupervised feature learning for optical character recognition. In *13th International Conference on Document Analysis and Recognition (ICDAR), 2015*, pages 1041–1045. IEEE, 2015.

[12] Elisa H. Barney Smith, Shatakshi Goyal, Robbie Scott, and Daniel P. Lopresti. Evaluation of voting with form dropout techniques for ballot vote counting. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 473–477. IEEE, 2011.

[13] Elisa H. Barney Smith, Daniel P. Lopresti, George Nagy, and Ziyan Wu. Towards improved paper-based election technology. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1255–1259. IEEE, 2011.

[14] Elisa H. Barney Smith, George Nagy, and Daniel P. Lopresti. Mark detection from scanned ballots. In Kathrin Berkner and Laurence Likforman-Sulem, editors, *DRR*, volume 7247 of *SPIE Proceedings*, pages 1–10. SPIE, 2009.

[15] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[16] Kai Wang, Eric Kim, Nicholas Carlini, Ivan Motyashov, Daniel Nguyen, and David Wagner. Operator-assisted tabulation of optical scan ballots. In *Presented as part of the 2012 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, Berkeley, CA, 2012. USENIX.

[17] Safwan Wshah, Girish Kumar, and Vengatesan Govindaraju. Script independent word spotting in offline handwritten documents based on hidden markov models. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 14–19. IEEE, 2012.

[18] Pingping Xiu, Daniel P. Lopresti, Henry S. Baird, George Nagy, and Elisa H. Barney Smith. Style-based ballot mark recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 216–220. IEEE, 2009.