Handwritten Word Image Categorization with Convolutional Neural Networks and Spatial Pyramid Pooling

J. Ignacio Toledo[1(✉)], Sebastian Sudholt[3], Alicia Fornés[2], Jordi Cucurull[1], Gernot A. Fink[3], and Josep Lladós[2]

[1] Scytl Secure Electronic Voting, Barcelona, Spain
{JuanIgnacio.Toledo,Jordi.Cucurull}@scytl.com
[2] Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain
{afornes,josep}@cvc.uab.es
[3] Department of Computer Science, TU Dortmund University, Dortmund, Germany
{sebastian.sudholt,gernot.fink}@tu-dortmund.de

Abstract. The extraction of relevant information from historical document collections is one of the key steps in order to make these documents available for access and searches. The usual approach combines transcription and grammars in order to extract semantically meaningful entities. In this paper, we describe a new method to obtain word categories directly from non-preprocessed handwritten word images. The method can be used to directly extract information, being an alternative to the transcription. Thus it can be used as a first step in any kind of syntactical analysis. The approach is based on Convolutional Neural Networks with a Spatial Pyramid Pooling layer to deal with the different shapes of the input images. We performed the experiments on a historical marriage record dataset, obtaining promising results.

Keywords: Document image analysis · Word image categorization · Convolutional neural networks · Named entity detection

1 Introduction

Document Image Analysis and Recognition (DIAR) is the pattern recognition research field devoted to the analysis, recognition and understanding of images of documents. Within this field, one of the most challenging tasks is handwriting recognition [3,6], defined as the task of converting the text contained in a document image into a machine readable format. Indeed, after decades of research, this task is still considered an open problem, specially when dealing with historical manuscripts. The main difficulties are: paper degradation, differences in the handwriting style across centuries, and old vocabulary and syntax.

Generally speaking, handwriting recognition relies on the combination of two models, the optical model and the linguistic model. The former is able to recognize the visual shape of characters or graphemes, and the second interprets

A. Robles-Kelly et al. (Eds.): S+SSPR 2016, LNCS 10029, pp. 543–552, 2016.
DOI: 10.1007/978-3-319-49055-7_48

them in their context based on some structural rules. The linguistic model can range from simple n-grams (probabilities of character or word sequences), to sophisticated syntactic formalisms enriched with semantic information. In this paper we focus in this last concept. Our proposed hipothesis is that in certain conditions where the text can be roughly described by a grammatical structure, the identification of named entities can boost the recognition in a parsing process. Named entity recognition is an information extraction problem consisting in detecting and classifying the text terms into pre-defined categories such as the names of people, streets, organizations, dates, etc. It can also be seen as the semantic annotation of text elements.

However, in many cases, a mere transcription is not the final goal, but more a means to achieve the understanding of the manuscript. Therefore, the aim is to understand the documents and extract the relevant information that these documents contain. For instance, for document collections in archives, museums and libraries, there is a growing interest in making the information available for accessing, searching, browsing, etc. A typical example can be demographic documents containing people's names, birthplaces, occupations, etc. In this application scenario, the extraction of the key contents and its storace in structured databases allows to envision innovative services based in genealogical, social or demographic searches.

A traditional approach to information extraction would be to first transcribe the text, and then use dictionaries, grammars or some other NLP (Natural Language Processing) techniques to detect named entities. Named entity detection [13] has its own caveats dealing with words that have not been seen during training (namely OOV- Out of Vocabulary Words), specially if, like in our case, one wants to detect entities that do not start with a capital letter (e.g. occupations). Moreover in historical handwritten documents, handwriting recognition struggles to produce an accurate transcription further reducing the accuracy of the whole system.

Another option is to transcribe and detect the named entities at the same time. The method described in [16] uses Hidden Markov Models and category n-grams to transcribe and detect categories in demographic documents, obtaining a quite good accuracy. However, the method is following a handwriting recognition architecture, and thus it depends on the performance of the optical model, it needs sufficient training data, and it is unable to detect or recognize OOV words.

A third alternative is to directly detect the named entities from the document image, avoiding the transcription step was recently published [1]. They use a traditional handwriting recognition approach, composed of a preprocessing step for binarization and slant normalization, and then extracting handcrafted features that are then fed into a BLSTM [9] (Bi-directional Long Short-Term Memory Blocks) neural network classifier. Afterwards, they use some post-processing heuristics to reduce false positives. For example, discarding short words or words starting by "Wh" or "Th" because they are more likely to be capitalized because they are the first word in a sentence. The performance of the method is quite good, but its goal is only detecting named entities in uppercase and not categorizing these words. Moreover the post-processing heuristics of this method are specific for the English language.

Another interesting recent work in a related area was proposed by Gordo et al. in [5]. In this work, the authors show that it is possible to extract semantic word embeddings directly from artificially generated word images. They show that the network can even learn possible semantic categories of OOV words by reusing information from prefixes or suffixes of known words. However the training in this dataset required datasets of several millions of synthetically generated word images, that is a very different scenario from the typical handwritten dataset where the annotations are scarce.

In this paper we propose a method able to detect and semantically categorize entities from the word image, without requiring any handwriting recognition system. Our approach is based on the recently popularized Convolutional Neural Networks, with a special Spatial Pyramid Pooling layer to deal with the characteristic variability in aspect ratio of word images.

Our approach has several advantages. First, it is able to detect entities no matter if they start with an uppercase or lowercase letters. Secondly, it can categorize these entities semantically. This means that the detected entity is also classified as belonging to a semantic category, such as name, surname, occupation, etc. The information of the semantic category of a word is a useful information in the parsing process. Third, the effort in the creation of training data is lower than the one needed for handwriting recognition (the word is not transcribed, just classified in several categories). Finally, the method does not have any problem with OOV words because it is not based on transcription or dictionaries. Even in scenarios where transcription would later be required, our method can be helpful by allowing us to use category specific models or dictionaries [16]. It can also be used to simply reduce the transcription cost by using the categorization as a way to select only relevant words to transcribe.

This paper is organized as follows: In Sect. 2 we will describe our method and the architecture of the neural network we built, explaining the function of each of the different layers. In Sect. 3 we will explain the technical details of the dataset used, the training of our neural network and also discuss the results of the experiments. Finally we will draw some conclusions and outline possible ideas for future work.

2 A CNN Based Word Image Categorization Method

In order to classify word images into semantic categories, we propose a CNN based method, inspired by [19] (See Fig. 1). The network is divided into three different parts: the convolutional layers, that can be seen as feature extractors; the fully connected layers that act as a classifier and the Spatial Pyramid Pooling layer that serves as a bridge between features and the classifier, by ensuring a fixed size representation. We will describe each of these different parts in this section.

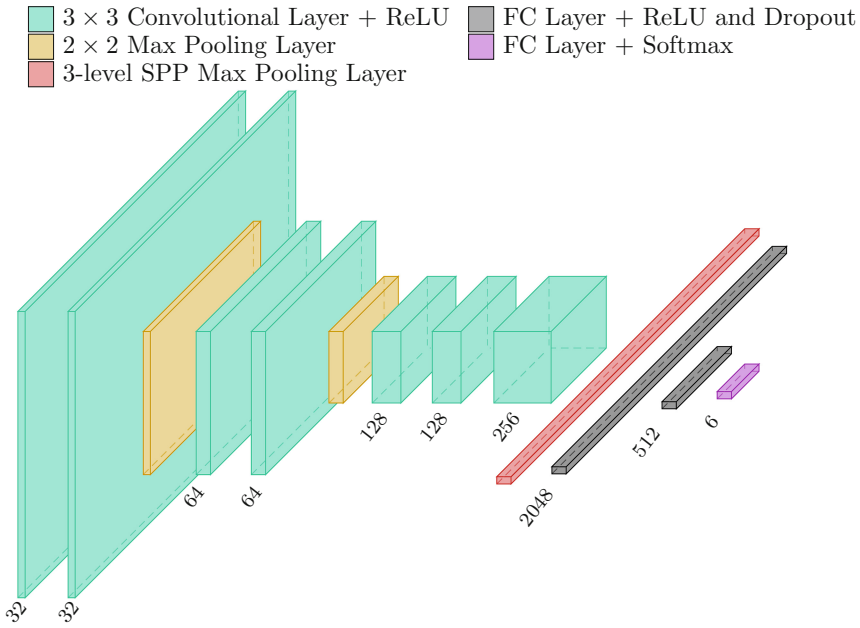

Fig. 1. Outline of our CNN architecture

2.1 Convolutional Neural Networks

Although Convolutional Neural Networks (CNN) were known since at least the early 1990's [12], it has only been recently that they gained major attention due to their high performance in virtually all fields of computer vision. The main building block of these artificial neural networks are the convolutional layers. These layers can be seen as a certain amount of filters. The output of a convolutional layer is generated by a discrete convolution of these filters with the input to the layer. Furthermore, an activation function is applied to the result of the convolution in order to make the layer able to learn non-linear functions. Compared to a standard perceptron, the filters allow sharing weights for different spatial locations thus considerably reducing the number of parameters in general [12].

Convolutional layers serve as feature detectors where each individual filter learns to detect certain features of the input image. In order to introduce a certain amount of translation invariance with respect to these detected features, CNNs usually make use of so called Pooling Layers. In these layers, activations across a certain receptive field are pooled and a single activation is forwarded to the next layer. In most cases this pooling is performed by taking the maximum value seen in the receptive field [10,17].

When stacking layers of convolutional and pooling layers, the filters in the individual convolutional layers learn edge features in the lower layers and more abstract features such as textures and object parts in the higher layers [20].

However, stacking a large amount of layers results in the so called Vanishing Gradient Problem [14] when using traditional activations such as sigmoid or hyperbolic tangent functions. Thus up until the early 2010's, neural network architectures where still fairly shallow [11]. The Vanishing Gradient Problem could first be tackled with the advent of using Rectified Linear Units (ReLU) as activation function [4]. This function is defined as the truncated linear function $f(x) = \max(0, x)$. Using the ReLU, deep CNN architectures are effectively trainable which was first successfully demonstrated in [10].

All of the convolutional layers in our architecture are a set of 3×3 Rectified Linear Units. The size of the filter was chosen to be 3×3 because they have shown to achieve better results compared to those with a bigger receptive field as they impose a regularization on the filter kernels [17]. Similar to the design presented in [17,19], we select a low number of filters in the lower layers and an increasing number in the higher layers. This leads to the neural network learning fewer low-level features for smaller receptive fields that gradually combine into more diverse high-level abstract features.

2.2 Fully Connected Layers

The general layout of CNNs can be split up in a convolutional and a fully connected part. While the convolutional and max pooling layers constitute the former, the latter is a standard Multilayer Perceptron (MLP). Thus, the convolutional part can be seen as a feature extractor while the MLP serves as a classifier. The layers of the MLP are often referred to as Fully Connected Layers (FC) in this context. Just as convolutional layers, the use of ReLU as activation function has shown itself to be effective across various architectures [10,17].

The large amount of free parameters in fully connected layers leads to the problem of the MLP learning the training set "by heart" if the amount of training samples is low. But even for larger training sets, co-adaptation is a common problem in the fully connected layers [8].

In order to counter this, various regularization measures have been proposed with Dropout [18] being one of the most prominent. Here, the output of a neuron has a probability (usually 0.5) to be set to 0 during training. A neuron in the following layer can now no longer rely on a specific neuron in the preceding layer to always be active for the same input image. Thus, the CNN has to learn multiple paths through the neural network for a single input image. This leads to more robust representations and can be seen as an ensemble within the CNN model.

The size of the different layers is a hyperparameter to tune experimentally, except for the final layer whose size has to match the number of classes we want to classify. This final layer usually uses a "softmax" activation function that outputs a probability distribution over the possible semantic categories in our experiment for each input image.

2.3 Spatial Pyramid Pooling

In general, the input to a CNN has to be of a fixed size (defined before training the network). For input images bigger or smaller than this defined size, the usual approach is to perform a (potentially anisotropic) rescale or crop from the image. For word images, with an important degree of variability in size and aspect ratio, cropping is of course not an option and resizing might introduce too strong artificial distortions in character shapes and stroke width. Thus it is important in our case that we allow our CNN to accept differently sized input images.

The key observation is that, while convolutional layers can deal with inputs of arbitrary shape and produce an output of variable shape, the fully connected layers demand a fixed size representation. Thus, the critical part is the connection between the convolutional and the fully connected part. In order to alleviate this problem, the authors in [7] propose a pooling strategy reminiscent of the spatial pyramid paradigm.

The pooling strategy performed over the last layer in the convolutional part is a pyramidal pooling over the entire receptive field. This way, the output of this Spatial Pyramid Pooling layer (SPP) is a representation with fixed dimension which can then serve as input for the ensuing MLP. It was also shown by the authors that this pooling strategy not only enables the CNN to accept differently sized input images, but it also increases the overall performance. In our method, we use a 3-level Spatial Pyramid max pooling with 4×4, 2×2 and 1×1 bin sizes. This allows us to capture meaningful features at different locations and scales whithin the word image.

3 Experimental Validation

In this section we will describe the experimental validation of our proposal. We will first explain iin detail the dataset used as well as some practical details relative to our training. We will then show the results achieved and discuss them.

3.1 Esposalles Dataset

For our experiments we used the Esposalles dataset [2,15]. This dataset consists of historical handwritten marriages records stored in the archives of Barcelona cathedral. The data we used corresponds to the volume 69, which contains 174 handwritten pages. This book was written between 1617 and 1619 by a single writer in old Catalan.

For our purpose the datasets consist of 55632 word images tagged with six different categories: *"male name"*, *"female name"*, *"surname"*, *"location"*, *"occupation"* and *"other"*. From this total we reserve 300 images of each class for testing, up to a total of 1800 images. After discarding word images smaller than 30×30pixels we end up with 53568 training examples for training and 1791 for test. In the training dataset there is a big class imbalance, with 31077 examples

Fig. 2. Several examples of word images in the Esposalles Dataset. The big degree of variability both in size and aspect ratio of the images makes impractical the common approach of resizing images to a common size.

Table 1. Comparative with other methods.

	Precision	Recall	F_1-Measure	Classification accuracy
Adak et al. [1]*	68.42	**92.66**	78.61	-
Romero et al. [16]**	69.1	69.2	69.15	-
Our approach	**84.23**	75.48	**79.61**	78.11

of the class *"other"*, 3636 *"female name"*, 4565 *"male name"*, 2854 *"surname"*, 6581 *"location"* and 4855 *"occupation"*. No normalization or preprocessing was done to word images besides remapping them to grayscale in the interval [0–1] (0: background, 1 foreground). It is worth noting that there are words with the same transcription that could potentially belong to different classes. This is specially significant for the *"surname"* class, since it is quite common for surnames to be related to a location (i.e. a city name), an occupation or even a male name. We can see several examples of word images in Fig. 2. The dataset is available upon request to the authors.

3.2 Experiments and Results

The network was trained using standard backpropagation with stochastic gradient descent with a learning rate of 10^{-4} Nesterov momentum 0.9 and a decay rate of 10^{-6} for 100 epochs, which proved enough to obtain a training accuracy of over 99 % in all the experiments. Since we are working with images of different size, each example had to be processed individually (batch size 1), that is the reason for the low value for the decay rate.

We used the standard categorical cross entropy as loss function. In order to deal with the class imbalance problem, we introduced a "class weight" parameter in the loss function to relatively increase the impact of misclassifying the classes

Table 2. Confusion Matrix for our CNN architecture, with a global accuracy 78.11 %.

	True class					
Predicted class	Other	Surname	Female name	Male name	Location	Occupation
Other	272	52	21	9	61	38
Surname	5	153	19	2	21	3
Female name	2	34	247	9	4	4
Male name	2	14	5	274	7	1
Location	14	33	6	4	203	4
Occupation	3	10	1	0	4	250

with less examples. The weight for each class is calculated by dividing the number of samples of the most populated class by the number of samples of each class.

$$w_i = \frac{max(n_i)}{n_i}$$

We performed several experiments with slightly different network architectures, to empirically calibrate the hyperparameters. After having fixed the learning and decay rates and the number of epochs we noticed that even drastically changing the number of parameters in our architecture the results were similar.

The proposed network architecture, as depicted in Fig. 1 achieved an accuracy of 78.11 % in our dataset. An alternative network with the similar architecture but, halving the number of parameters of all the layers of the network (half of the channels in each of the convolutional layers and half of the neurons in each of the fully connected layers, and keeping the same 3-level pyramid pooling) produced an accuracy of 77.33 %.

In Table 2 we see that the errors are not evenly distributed. Despite the introduced class weights, the network is still more likely to mistakenly assign the class *"other"*. We can also see that examples corresponding to the *"surname"* class are harder to classify. This may be due to several reasons since, as discussed earlier, surnames are usually derived from names, places or occupations. It is also worth noting that the *"surname"* class is the one with fewer samples, thus having seen less examples the model is more likely to overfit this particular class.

The comparison of our method with similar methods in the literature is not an easy task, because this is a relatively recent area of research and there are few publications addressing similar issues. Even the most similar methods have big differences, for instance none of the methods provides a classification accuracy metric. In [16] they address the classification as an aid to transcription, and they work with the Esposalles dataset but with a different labeling with a different number of classes. In the case of [1] the aim is a named entity detection, with a binary output. They provide results with different datasets, so we selected the best result they achieved, usingn the IAM dataset.

We calculated our precision/recall metrics following the approach described in [16], and defined as: *"Let R be the number of relevant words contained in the*

document, let D be the number of relevant words that the system has detected, and let C be the number of the relevant words correctly detected by the system. Precision (π) *and recall* (ρ) *are computed as":*

$$\pi = \frac{C}{D} \qquad \rho = \frac{C}{R}$$

In order to compare our results with the state of the art we can see the class "*Other*" as non-relevant words. Then, for "relevant words" we understand words with a "True Class" other than "*Other*" and for "relevant detected word" we understand word-images assigned with a label other than "*Other*". Finally for "correctly detected" we understand examples where the system correctly assigned a label other than "*Other*". That means we do not consider a word as "correctly detected" unless it is also assigned to the correct category (Table 1).

4 Conclusions

We have presented a simple approach to word categorization using convolutional neural networks. The spatial pyramid pooling layer allows us to deal with the important variability in aspect ratio of word images without artificially distorting our image. We believe that the results are specially promising given that we are classifying just isolated words images with no transcription, context information or language model of any kind. Thus, the addition of context information or simple language models should significantly boost the performance, specially in the mentioned case of surnames, and is probably the next step in this research. It would also be interesting to perform more experiments in order to determine if the network learns heuristic similar to what a human would use. For instance, names, surnames and locations usually start with a capital letter whereas occupations and other words usually do not and some word endings have a much higher likelihood on a particular class.

Finally, we also believe that this model can improve the performance of word spotting methods by reducing the search space or making a more semantic search (for example, one might specifically search for the surname Shepherd or the occupation shepherd).

Acknowledgements. This work has been partially supported by the Spanish project TIN2015-70924-C2-2-R, 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. Adak, C., Chaudhuri, B.B., Blumenstein, M.: Named entity recognition from unstructured handwritten document images. In: Doermann, D.S., Govindaraju, V., Lopresti, D.P., Natarajan, P. (eds.) Document Analysis Systems, pp. 375–380 (2016)

2. Fernández-Mota, D., Almazán, J., Cirera, N., Fornés, A., Lladós, J.: BH2M: the Barcelona historical, handwritten marriages database. In: 22nd International Conference on Pattern Recognition (ICPR), pp. 256–261. IEEE (2014)
3. Frinken, V., Bunke, H.: Continuous handwritten script recognition. In: Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition, pp. 391–425. Springer, London (2014)
4. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
5. Gordo, A., Almazan, J., Murray, N., Perronin, F.: LEWIS: latent embeddings for word images and their semantics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1242–1250 (2015)
6. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **31**(5), 855–868 (2009)
7. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **37**(9), 1904–1916 (2015)
8. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
12. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: NIPS, pp. 396–404 (1990)
13. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
14. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, no. 2, pp. 1310–1318 (2013)
15. Romero, V., Fornés, A., Serrano, N., SáNchez, J.A., Toselli, A.H., Frinken, V., Vidal, E., Lladós, J.: The esposalles database: an ancient marriage license corpus for off-line handwriting recognition. Pattern Recogn. **46**(6), 1658–1669 (2013)
16. Romero, V., Sánchez, J.A.: Category-based language models for handwriting recognition of marriage license books. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 788–792. IEEE (2013)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
19. Sudholt, S., Fink, G.A.: PHOCNet: a deep convolutional neural network for word spotting in handwritten documents. arXiv preprint arXiv:1604.00187 (2016)
20. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_53