Deep Neural Networks for Page Stream Segmentation and Classification

Conference Paper · November 2016

DOI: 10.1109/DICTA.2016.7797031

4 authors, including:

Ignazio Gallo
Università degli Studi dell'Insubria
153 PUBLICATIONS **1,235** CITATIONS
SEE PROFILE

Alessandro Zamberletti
Università degli Studi dell'Insubria
25 PUBLICATIONS **303** CITATIONS
SEE PROFILE

Alessandro Calefati
Università degli Studi dell'Insubria
34 PUBLICATIONS **321** CITATIONS
SEE PROFILE

Deep Neural Networks for Page Stream Segmentation and Classification

Ignazio Gallo, Lucia Noce, Alessandro Zamberletti and Alessandro Calefati
Department of Theoretical and Applied Sciences (DiSTA)
University of Insubria, via Mazzini 5, Varese, Italy
Email: ignazio.gallo@uninsubria.it

***Abstract*—In this manuscript we propose a novel method for jointly page stream segmentation and multi-page document classification. The end goal is to classify a stream of pages as belonging to different classes of documents. We take advantage of the recent state-of-the-art results achieved using deep architectures in related fields such as document image classification, and we adopt similar models to obtain satisfying classification accuracies and a low computational complexity. Our contribution is twofold: first, the extraction of visual features from the processed documents is automatically performed by the chosen Convolutional Neural Network; second, the predictions of the same network are further refined using an additional deep model which processes them in a classic sliding-window manner to help finding and solving classification errors committed by the first network. The proposed pipeline has been evaluated on a publicly available dataset composed of more than half a million multi-page documents collected by an on-line loan comparison company, showing excellent results and high efficiency.**

I. Introduction

Batch scanning of documents' stream is becoming a more and more important requirement in many scenarios. A large number of documents are daily collected and processed from trading companies: forms, claims, invoices and contracts are only few examples of them. Such large stream of documents must be accurately and quickly processed and each document needs to be correctly dispatched in the appropriate recipient.

In this work we propose a method for the automated classification of documents collected from an on-line company that deals with granting quotes for loans or mortgages. A wide range of document typologies are requested and gathered to provide the best available solution. On a daily basis, potential customers send thousands of documents made up of several pages. The stream of documents must be digitized, segmented into consequent pages and then forwarded to the right recipient in order to be processed and then saved into a database.

One of the most important phase of the process consists of the segmentation of the continuous document stream into subsets of pages that represent each single physical document. From the result of this procedure, also known as document separation [1], depends the accuracy of the entire process. Each document is finally classified as belonging to one of the possible classes.

The segmentation and classification of page streams is commonly carried out by human personnel, thus being time-expensive and error-prone. Moreover, documents are often written in different languages and force operators to an even greater effort during the classification phase. An automated mechanism for segmentation and classification of large quantity of documents, can be useful in providing optimization in terms of both segmentation/classification accuracy and processing time.

This task is not trivial because document categories are usually numerous and can easily reach hundreds or even thousands of classes. Additionally, data ambiguity makes this task even harder. A single page may belong to different categories, and often the correct category can be determined only by watching the context in which the page is inserted.

In this work, we propose a method based on deep classification models. Deep models for classification are fast, do not require hand-crafted features extraction and currently represent the state-of-art for document image classification [2]. We propose a supervised approach for page stream segmentation and document image classification using features learned by Convolutional Neural Networks (CNN). In the final step of our approach we correct the CNN predictions using an additional deep model that analyzes the stream of classified image documents.

II. Related works

Document image classification is a widely investigated research area. Several supervised and unsupervised models which aim to classify images representing documents have been proposed in literature [3]. Existing approaches differ from each other on the basis of computed feature types (textual or visual) [4] and/or the type of image analysis performed over the processed document (global or local) [5].

Inspired by the outstanding results reached in the Computer Vision research field, recently presented approaches [2], [6] also employ Deep Convolutional Neural Networks to achieve state-of-the-art results in Document Image Classification [2].

Kang et al. [6] firstly and *Harley et al.* [2] slightly later, explored CNN applied to the document image classification task, demonstrating that such models are able to learn the entire supervised document image classification process, from feature extraction to final classification. Moreover, experimental phases performed on different challenging datasets, show that such approaches outperform all previously explored methodologies.

Harley et al. [2] reached state-of-the-art results for document image classification and retrieval, asserting that features

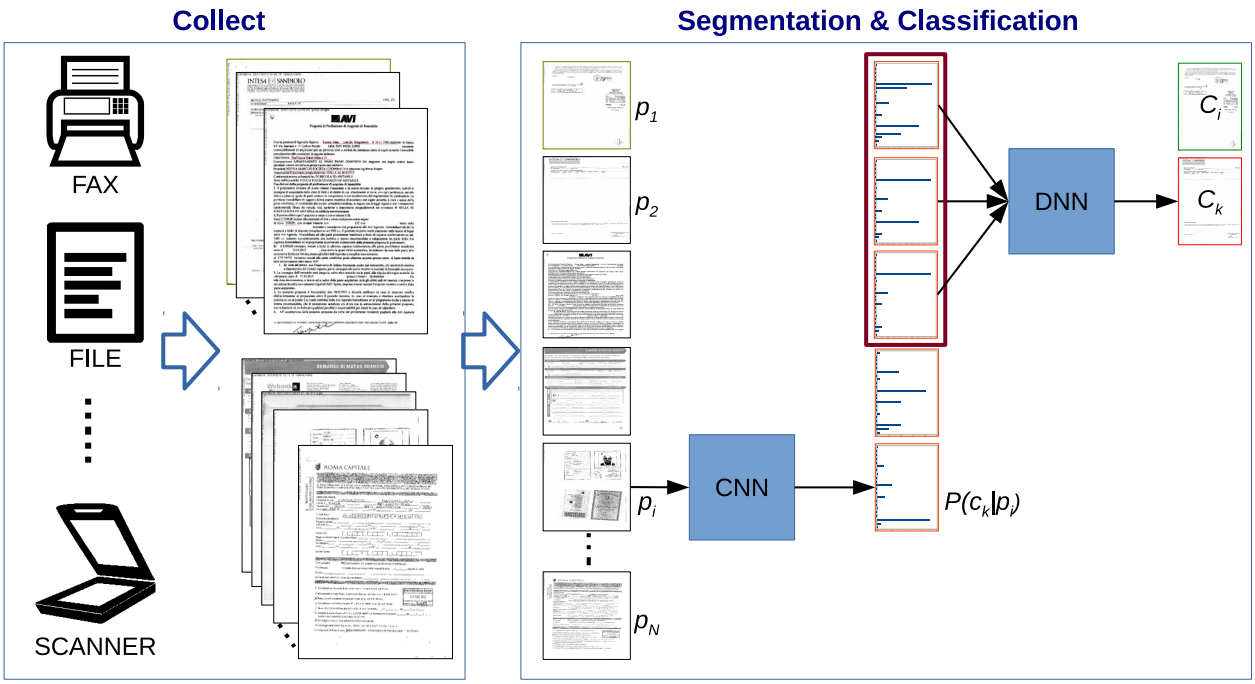

Fig. 1. Work-flow of the proposed approach. On the left: the documents collected from various sources are provided as input to the proposed system. On the right: the stream of pages is firstly segmented with a CNN that analyzes one page at the time; immediately after, a DNN exploits the classes probabilities produced by the CNN on a set of consecutive pages to classify and segment the documents.

extracted from deep CNN exceed the performance of all alternative visual and textual features both on document image classification and retrieval by a large margin.

Among document image classification and retrieval previous works, only few approaches deal with multi-page documents or analyze the document image streams, and none of them exploit the state-of-the-art results reaches with the adoption of Convolutional Neural Networks.

Gordo et al. [1] focused on the segmentation of a continuous page stream into multi-page documents, and the classification of the resulting documents. Authors represent a multi-page document with a single feature vector, as a first step they classified each documents with a one-versus-rest SVM classifiers, and then they used a probabilistic approach to model document validity and to drive the page flow segmentation algorithm. A dataset of approximately 7000 multi-page document images divided in 13 different classes was used during the experimental phase.

The approach proposed by *Daher et al.* [7] employs a framework composed of two modules (segmentation and validation) for the segmentation and classification of document streams. An incremental classifier is adopted during both the two phases, in the segmentation module, the relationship between consecutive pages is classified as either continuity or rupture, while the verification module predicts a confidential score of the belonging of a specific document to a class. A manually performed correction module allows the incremental learning of the system. Experiments were performed on three different datasets which reach 164 classes in total.

Rusiñol et al. [8] addressed the retrieval problem rather than the classification task, testing their approach on a dataset of multi-page document images coming from a real banking work-flow. Authors adopted a bag-of-words framework to represent single pages using textual or visual information and performed a multi-page document retrieval system based on single-page similarities. The classification task was addressed by some of the same authors in a newer work [9], that extended the previously described by adding a final step that, using an n-gram model, provides pages classification.

We exploit the state-of-the-art results achieved for the document image classification task by adopting Convolutional Neural Network in our pipeline. This allows us to bypass the feature extraction step. The employed model exploits the existing visual similarity between documents belonging to the same class, saving time, limiting the risk of do not consider relevant features, and avoiding to built ad-hoc and document-dependent features.

Differently from other approaches we use the output of the CNN, not as the label of a single class, but as the probability distribution of each class which will be used in the segmentation phase. The model was tested over a challenging dataset made up of 224 different classes, which is a much higher number of classes if compared to the reported related works.

III. Proposed Approach

The main task at the basis of our approach is the automated classification of documents containing textual elements, exploiting only the graphical format of text rather than applying

OCR algorithms to read the content. For our purpose, a document is a set of pages in image format used as input to a CNN classifier. Instead of using the traditional "winner takes all" approach in order to obtain the class for each page, we consider the top-k classes with the highest probability. In several cases the CNN is not able to classify a single page belonging to a specific class, giving an almost constant probability distribution. The reasons are mainly two: the high number of classes and the similarity between different documents. Usually the aim of neural networks, or any classifier in general is to maximize the similarity intra-class and the difference inter-classes. This is not a trivial task because sometimes, also for a human, could be difficult to assign a document to a class or another. To overcome this problem, the membership of a page to a class is computed considering also pages that are in the neighborhood of the central page. In this way, many ambiguities have been solved.

Another key aspect in the document stream context is the processing speed of a single page. Using OCR algorithms for text extraction raises the amount of time required for each page, making the proposed pipeline unusable in near real-time contexts. The proposal uses the actual best algorithms for image classification. In particular we use a Convolutional Neural Network for the stream segmentation phase and a Deep Neural Network for the multi-page document classification (see Figure 1). These models can be considered the state-of-the-art in image classification and are very fast when executed using a Graphic Processing Unit.

More formally, let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ be a stream of N pages that we want transform into a set $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ of M sequential multi-page documents, where each $d_k = \{p_i, p_{i+1}, \ldots, p_j\}$ for $i \leq j$. In literature this process is usually called *page stream segmentation* [1], [7], [10]. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ represent the set of possible document classes, then for each d_i exists a class c_k such that $d_i \in c_k$ In general, different documents d_k will contain a different number of pages, even if they belong to the same class. The proposed general framework consists of two modules in cascade that have as main objective the classification of a pages' stream $\mathcal{P}$ in order to separate them in a set of documents $\mathcal{D}$.

A. Page Stream Segmentation

The first module is mainly involved in documents separation from a stream $\mathcal{P}$ of N pages, assigning to each page p_i a probability $P(c_k \mid p_i)$ of belonging to each class $c_k \in \mathcal{C}$.

To obtain all the probabilities for each page we use the CNN proposed by Krizhevsky *et al.* [11] and configured with a softmax [12] in the output layer. The softmax function is the gradient-log-normalizer of the categorical probability distribution. For this reason, the softmax function is used in various probabilistic multi-class classification methods including artificial neural networks [13]. Similar to [6], we propose to use a CNN for document image classification with the main objective to learn a hierarchy of feature detectors and train a nonlinear classifier to identify complex document layouts. Given a document image, we first perform down-sampling and pixel value normalization, then feed the normalized image to the CNN to obtain the classes probability in output.

Images must be resized before the classification because they are too large to be given as input to the CNN and to avoid overfitting problem. Moreover training time becomes longer if images used are too big. Considering that our approach doesn't need to read the content of a page, it is possible to reduce the size as long as the layout informations are still identifiable. Initially supplied images have different dimensions, so we convert all images to 256×256 pixels with bilinear interpolation. Using this resolution, most of the characters are no longer recognizable, but the overall layout is preserved and the position of title, text or tables can be easily determined watching it. An approach with higher resolution, not necessarily using the entire image as a single input to the network, is not necessary. In fact, as shown in [2], given sufficient training data, a single CNN trained on entire images performed approximately as well as an ensemble of CNNs trained on specific subregions of document images. In Fig. 1 and Fig. 2 is possible to see some example of resampled pages of different classes.

The CNN used here is composed of five convolutional layers some of which are followed by max-pooling layers, three fully-connected layers with a final softmax. The full architecture can be written as $227 \times 227 - 11 \times 11 \times 96 - 5 \times 5 \times 256 - 3 \times 3 \times 384 - 3 \times 3 \times 384 - 3 \times 3 \times 256 - 4096 - 4096 - |\mathcal{C}|$. The input images size is 227×227, automatically cropped from the 256×256 input images. The first convolutional layer filters the input image with 96 kernels of size 11×11. The second convolutional layer filters the output of the first convolutional layer with 256 kernels of size 5×5. The third and fourth convolutional layers have 384 kernels of size 3×3, and the fifth convolutional layer has 256 kernels of size 3×3. The fully-connected hidden layers have 4096 neurons each. The last layer is a logistic regression with softmax that outputs the probability on each class, as defined in the following equation

$$P(c_i) = \frac{e^{W_i x + b_i}}{\sum_{j=1}^{|\mathcal{C}|} e^{W_j x + b_j}} \tag{1}$$

where x is the output of the last fully connected hidden layer, W_i and b_i are the weights and biases of i^{th} neuron in the output layer.

Class probability values associated with each page are stored for later use in the next module, while for the page stream segmentation, only the highest probability value is taken into account: every time that the model assigns a class c_k to a page p_{i+1} applying the *winner-takes-all* principle to the output layer, that is different from the label of previous page p_i, then we complete a document d_j and start with a new document d_{j+1}.

Although the process described heretofore allows to segment a stream of pages $\mathcal{P}$ into a sequence of documents $\mathcal{D}$ and to assign a class label to each document, many misclassification errors that split documents improperly, may

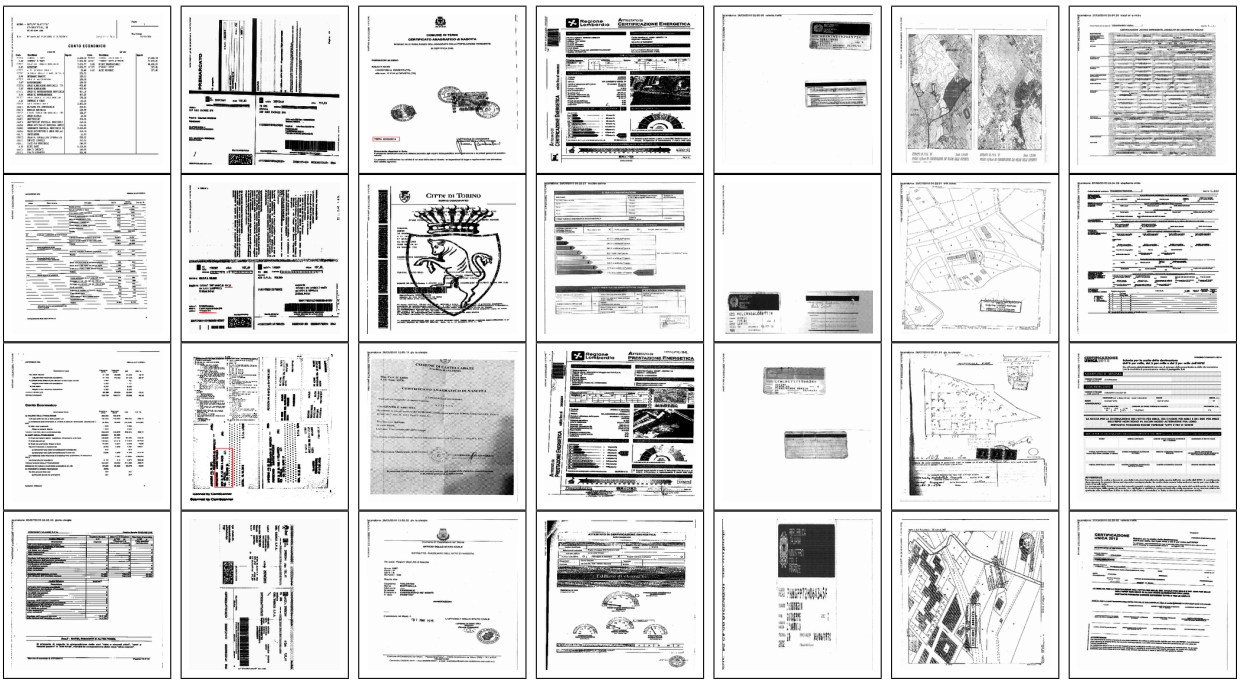

Fig. 2. Documents belonging to seven different classes from the *document page samples* dataset. Each column shows four different pages of a class. From the leftmost column to the rightmost, class names are: *financial-statements*, *utility-bills*, *birth-certificate*, *energy-certification*, *tax-code*, *cadastral-map* and *CUD-model*.

occur. For this reason a further step of classification and errors correction is mandatory, as described in the following section.

B. Refinement and Documents Classification

Because the class of a page p_i is influenced by the surrounding pages $\{p_{i-W}, \ldots, p_i, \ldots, p_{i+W}\}$, it is necessary to observe the page in a wider context. We use a sliding window of size $S = 2W + 1$ centered on the page p_i that we want to classify, and the rest of the pages in the window are the *neighbors of the page* to classify. For each page to classify we give in input to our model, the probabilities $P(c_k \mid p_i)$ to belong to all classes $c_k \in \mathcal{C}$, for all the pages in the sliding window $\{p_{i-W}, \ldots, p_i, \ldots, p_{i+W}\}$. The probabilities for each page are derived from the CNN described in the previous section III-A. As showed in the right part of the Fig. 1, the DNN module classifies pages by analyzing the probability of the pages contained in the sliding window. Assuming an S size sliding window, the pattern to give as input to the neural model used in this phase will be a vector having the following $K * S$ probability values $[P(c_1 \mid p_{i-W}), \ldots, P(c_K \mid p_{i-W}), \ldots, P(c_1 \mid p_{i+W}), \ldots, P(c_K \mid p_{i+W})]$.

In order to classify the first and the last W pages of each stream, we use a *zero-padding* strategy. All pages that would fall outside of the stream are taken as a sequence of K probabilities values equal to zero. By doing this we can apply the sliding window to every page of our input stream, and get an equally sized output.

Below is reported an outline of the training steps:

- We manually label the dataset considering a similar number of pages for each class.
- We extract classes probability from each page of the input stream using a CNN model.
- We train a DNN using a sliding window of size S centered in each page of the training set.

The DNN model used here is a full-connected Multilayer Perceptron (MLP) with Rectified Linear Units or ReLU [14] activation functions for each neuron. This type of activation allows the use of a MLP as a deep neural model, and then with all the advantages deriving from there.

IV. Experiments

This section has a twofold purpose: to demonstrate that CNNs are an excellent model to get the probability of belonging to a class, and to show that the proposed algorithm represents an excellent solution for documents segmentation and classification from the stream.

A. Datasets

To the best of our knowledge we are not aware of any public, multi-page document dataset on which we can evaluate our classification and segmentation methods. For this reason, all the experiments were performed on two datasets that we built using real documents.

The first dataset that we named *document page samples*, consists of 576053 pages in image format of incoming documents sampled from a stream of requests for loan quotes of the real world. The images were saved using the RGB color model, then have been resampled and resized to 256×256. The dataset is divided into 224 different classes of documents (identity documents, payroll, contracts, etc.) that have been

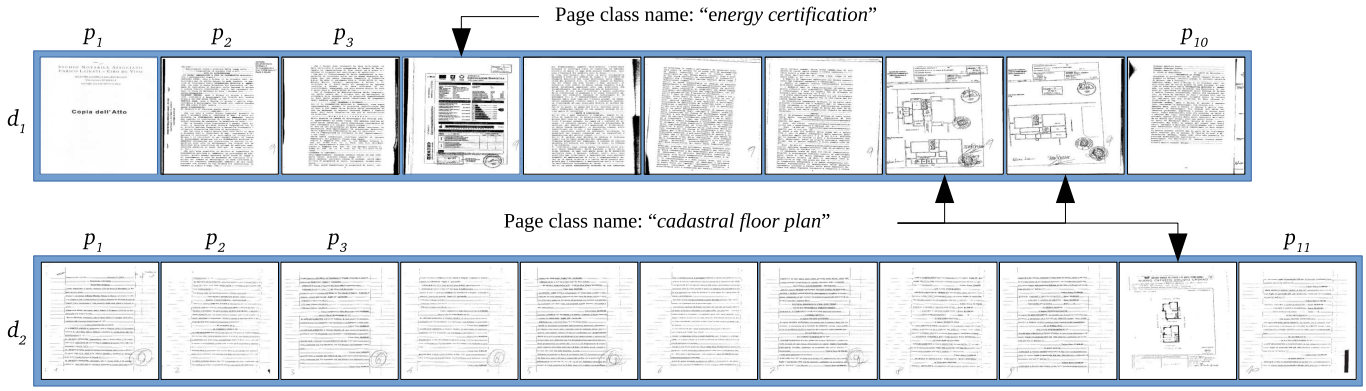

Fig. 3. In this visual example, both documents d_1 and d_2 belong to the same class *deed of sale of the building* $\in \mathcal{C}$. Since the CNN processes individual pages and has no information about the context in which those pages reside, some pages p_i are misclassified as belonging to classes *energy certification* and *cadastral floor plan* $\in \mathcal{C}$. If we also exploit information about the local context in which pages appear within the processed stream, it becomes much easier to understand that all of the pages belong to the same document class since they are surrounded by pages previously classified by the CNN as belonging to class *deed of sale of the building* $\in \mathcal{C}$.

manually labeled. We use this dataset to train and evaluate the CNN that produces the class probabilities for each page. In order to avoid having a dataset unbalanced on some classes, we selected up to 3000 images per class, thus obtaining a dataset with 260059 training pages and 86664 (25%) test pages. Fig. 2 shows examples of multi-page documents of our dataset. The *document page samples* dataset will be made available on the Arte-Lab Repository[1] because we believe that dissemination of anonymous data is essential to the future health of our field.

The second dataset that we named *document page streams*, consists of 1413 streams of documents in image format that represents a stream of requests for real world mortgages. The entire dataset contains 56249 individual pages and 12882 different documents. These streams of documents do not have any intersection with the first dataset and represent real cases processed in a second time. Each stream of documents arriving from different potential customer, is made up of a variable number of documents and each document contains a variable number of pages. Note that the class label c_k of each document d_i is available at document level, but not at page level, thereby many documents may also contain pages belonging to other categories $c_m \in \mathcal{C}$, implying that the dataset *document page streams* may contain pages apparently misclassified as ground-truth (see example in Fig. 3). This dataset is used to evaluate the performances during the last step of segmentation refinement and document classification. The dataset is divided into 224 classes of documents (the same number of classes of the data set described above) which were labeled always manually at the document level. All the document streams were randomly divided into training and test (25%).

B. Classes probability for a page

In our first set of experiments we explore the performances of the CNN described in section III-A, by interpreting its output as classification or as class likelihood.

[1]http://artelab.dista.uninsubria.it/downloads.html

TABLE I
SINGLE PAGE CLASSIFICATION ACCURACY OF THREE DIFFERENT DNN CONFIGURATIONS. THE INPUT SIZE DEPENDS ON THE SIZE OF THE SLIDING WINDOW.

Win size	Input	Hidden layers	Accuracy	Kappa
3	672	2000,1000,1000	81.72%	0.81
7	1568	2000,1000,1000	94.16%	0.95
11	2464	2000,1000,1000	97.45%	0.98

TABLE II
COMPARISON BETWEEN THE COMPLETE SEGMENTATION PROCESS, WHICH USES A SLIDING WINDOW EQUAL TO 11, WITH THE SAME PROCESS THAT DOES NOT USE THE DNN

Accuracy	CNN	CNN+DNN
Pages	58.48%	97.45%
Documents	54.49%	88.16%

We trained the CNN for 60 epochs on a GeForce GTX 980 GPU using the dataset *document page samples*. During the training, the learning rate decreases exponentially from 0.01 up to 0.00001. Using the mini-batch size as 100 and the Caffe library [15], the training time reported was equal to 16 hours and 24 minutes. In Fig. 4 it is interesting to note that most of the filters automatically found by CNN for extracting the lower-level features, are specialized in the extraction of sequences of horizontal and vertical lines.

The model we obtained, achieves a top-1 test accuracy equals to 85.2% and a top-5 test accuracy equals to 96.22%. This result shows that by interpreting the output of CNN as probabilities we are sure that, to 96.22%, the correct class of each page is among the five classes with higher probability. Hence the idea of using these probability values to train a new deep model capable of correcting some segmentation and classification errors caused by CNN.

C. Segmentation and Classification

In the second set of experiments we evaluate the capacity of the entire segmentation and classification process. Here, we use the Weka [16] plug-in in order to train and use the fully-connected deep neural network with dropout regularization and Rectified Linear Units [14].

As a first step we classify all the pages of the *document page streams* test set using only the CNN trained in the experiment described in section IV-B. We first calculated the classification accuracy of individual pages, obtaining an accuracy equals to 58.48%. After that, we have also calculated the segmentation accuracy of all the documents contained in each stream. A document is segmented correctly when the first page, the last page and all intermediate ones are classified in the same category declared in the ground truth. The measured segmentation accuracy is equal to 54.49%. The segmentation and classification accuracy is quite low because the *document page streams* dataset contains many types of documents never seen before and which for that reason have been traced to the category "not a class" of the same dataset.

As a second experiment we tried to figure out what was the best setup for DNN. We fixed the fraction of units to dropout during training: 0.2 in the input layer and 0.5 for all the hidden layers. Learning rate, batch size and stopping criteria are found automatically by the used tool. The first important variable is the size of the sliding window in input to the DNN, while the second is the topology of the DNN in terms of number of hidden layers and number of neurons per layer. In the present work we tried with three different dimensions S of the sliding window, obtaining in this way three different configurations of the input layer of the DNN. The Table I shows all the values used. The number of hidden layers of the network and the number of neurons for each layer was found in an experimental way, starting from a network with only one hidden layer and having only 100 neurons, we increased the neuron and layers in a progressive way. The best configuration that has been found is that reported in the Table I, achieved by increasing the size of the sliding window. In our case, using a window that sees 11 input consecutive pages of the stream, we have obtained a page classification accuracy equal to 97.45% on the test streams. This result is very important because it shows how we can correct many misclassification errors on the individual pages of a stream, simply by using classification probabilities of the CNN.

As a final experiment, we wanted to assess the contribution of the DNN for segmentation and classification. To do this, we compared the complete segmentation process, which uses sliding window equal to 11, with the same process that does not use the DNN (see Table II). In the same table the classification accuracy values of individual pages, in addition to the classification accuracy of segmented documents, are shown. By analyzing the results shown in the two Tables I and II, it is quite clear that the contribution of DNN is crucial for the success of the entire segmentation and classification process. Fig. 5 shows a representative example of classification and segmentation of a single stream of documents.

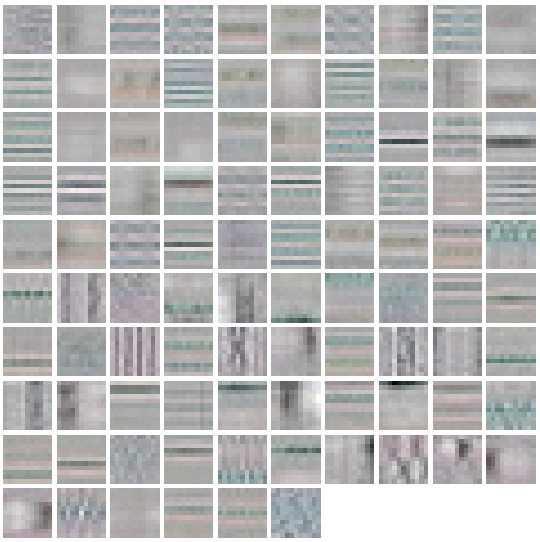

Fig. 4. Graphical representation of the 96 kernels 11×11 of the first convolutional layer of the CNN, after the training phase.

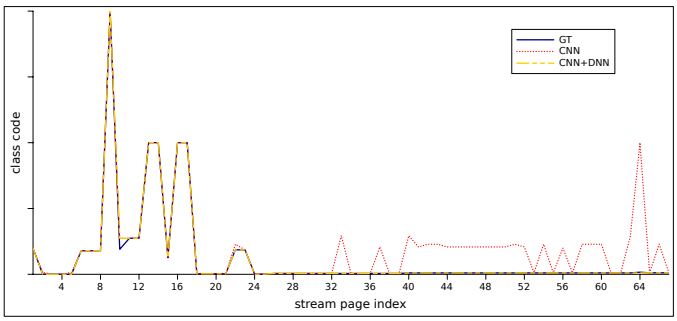

Fig. 5. Classification of a single stream with 16 documents and 67 pages. The ground truth (GT) is compared with the classification of the CNN and with the classification of the entire process proposed (CNN + DNN). When the curves are superimposed on the GT, then the classification is correct. Consecutive pages with the same class label form a document. In this stream, classification accuracy of single pages increased from 53.73% using only CNN, to 97.01% using CNN + DNN.

V. Conclusion

The proposed method introduces a novel way of refining the predictions generated by a Convolutional Neural Network by exploiting additional contextual information implicitly provided by the order in which different documents appear in a multi-page stream. In our experimental evaluation, we have proved that by refining the document classification predictions generated by a Convolutional Neural Network using another Deep model in a classic sliding-window manner, we are able to substantially improve document classification accuracy without increasing computational complexity. Since no hand-crafted features are used in the proposed pipeline, our method is highly generic and can potentially achieve good classification results in many real-world multi-page document classification tasks.

ACKNOWLEDGMENT

The authors thank Gruppo MutuiOnline S.p.A. for posing the problem and providing anonymous data with helpful discussions. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeForce GTX 980 GPU used for this research.

REFERENCES

[1] A. Gordo, M. Rusiñol, D. Karatzas, and A. D. Bagdanov, "Document classification and page stream segmentation for digital mailroom applications," in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, ser. ICDAR '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 621–625.

[2] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval," in *International Conference on Document Analysis and Recognition (ICDAR)*.

[3] N. Chen and D. Blostein, "A survey of document image classification: Problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition*, 2007.

[4] Jayant Kumar and David Doermann, "Unsupervised Classification of Structurally Similar Document Images," in *Intl. Conf. on Document Analysis and Recognition (ICDAR 13)*, 2013.

[5] P. Sarkar, "Learning image anchor templates for document classification and data extraction," in *20th International Conference on Pattern Recognition (ICPR)*, 2010.

[6] Le Kang, Jayant Kumar, Peng Ye, Yi Li, and David Doermann, "Convolutional Neural Networks for Document Image Classification," in *International Conference on Pattern Recognition (ICPR 2014)*, August 2014, pp. 3168–3172.

[7] H. Daher, M. Bouguelia, A. Belaïd, and V. P. D'Andecy, "Multipage administrative document stream segmentation," in *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, 2014, pp. 966–971.

[8] M. Rusiñol, D. Karatzas, A. D. Bagdanov, and J. Lladós, "Multipage document retrieval by textual and visual representations," in *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, 2012.

[9] M. "Rusiñol, V. Frinken, D. Karatzas, A. D. Bagdanov, and J. Lladós, "Multimodal page classification in administrative document image streams," *International Journal on Document Analysis and Recognition (IJDAR)*, 2014.

[10] M. Rusiñol, V. Frinken, D. Karatzas, A. D. Bagdanov, and J. Lladós, "Multimodal page classification in administrative document image streams," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 331–341, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems (NIPS) 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," in *Neurocomputing: Algorithms, Architectures and Applications*, ser. NATO ASI Series, Fogelman-Soulie and Herault, Eds. Springer, 1990, pp. 227–236.

[14] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM '14, 2014, pp. 675–678.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, nov 2009.