Fast CNN-based document layout analysis

Dário Augusto Borges Oliveira
IBM Research Brazil
Rua Tutóia, 1157, Paraíso, São Paulo, Brazil
dariobo@br.ibm.com

Matheus Palhares Viana
IBM Research Brazil
Rua Tutóia, 1157, Paraíso, São Paulo, Brazil
mpviana@br.ibm.com

Abstract

Automatic document layout analysis is a crucial step in cognitive computing and processes that extract information out of document images, such as specific-domain knowledge database creation, graphs and images understanding, extraction of structured data from tables, and others. Even with the progress observed in this field in the last years, challenges are still open and range from accurately detecting content boxes to classifying them into semantically meaningful classes. With the popularization of mobile devices and cloud-based services, the need for approaches that are both fast and economic in data usage is a reality. In this paper we propose a fast one-dimensional approach for automatic document layout analysis considering text, figures and tables based on convolutional neural networks (CNN). We take advantage of the inherently one-dimensional pattern observed in text and table blocks to reduce the dimension analysis from bi-dimensional documents images to 1D signatures, improving significantly the overall performance: we present considerably faster execution times and more compact data usage with no loss in overall accuracy if compared with a classical bi-dimensional CNN approach.

1. Introduction

Documents are a very important source of information for many different cognitive processes such as knowledge database creation, OCR, graphic understanding, document retrieval, and others. A crucial step to extract information out of documents is the layout analysis, which consists of identifying and categorizing document images regions of interest.

In literature, many methods have been proposed for document image layout analysis, and according to [11] they can be classified into three different groups: (i) region or block based classification methods [21, 17], (ii) pixel based classification methods [14, 13], (iii) connected component classification methods [6, 20, 1]. Methods based on region or block bases classification usually segment a document image page into document zones, and then classify them into meaningful semantic classes. Pixel-based classification methods take each pixel individually into account and use a classifier to generate a labelled image with regions hypotheses. Connected components approaches use local information to create object hypotheses that are further inspected, combined and refined, and finally classified.

When it comes to image classification, convolutional neural networks (CNNs) have been widely adopted in many different fields for a variety of purposes, including document analysis [9, 8]. However, their inherent very intense computational burden usually limits the cost-benefit of using them in document storage and retrieval applications where low memory and fast processing are vital. [3, 4] proposed methods to reduce the computation burden of document analysis using projections for identifying image blocks but do not benefit from the robustness of CNNs using a one-dimensional convolutional architecture. This scenario creates many opportunities for innovative CNN-based methods for document analysis that reduce the computational cost and data usage without decreasing the expected accuracy.

In this paper, we propose a block based classification method that consists of three steps: i) pre-process a document input image and segment it into its blocks of content; ii) use their vertical and horizontal projections to train a CNN model for multi-class classification considering text, image and table classes; iii) detect new documents layout using a pipeline including the trained CNN model.

Our main contribution is the novel one-dimensional CNN approach for fast automatic layout detection of structured image documents. A common bi-dimensional CNN procedure was also implemented for comparing performances and show that our method delivers the same accuracy with a lower computational cost and data usage associated. Our approach can be useful, for example, for applications in mobile devices due to its low computational cost or for services in the cloud that could benefit of sending/receiving compact one-dimensional data.

2. Methodology

The methodology we propose for document image layout analysis implements a pipeline that goes from segmentation of document images into blocks of content to their final classification, as shown in Figure 1. Each step is described in detail in the following subsections.

2.1. Segmenting blocks of content in the document image

The first step executed in our method is segmenting each document image page into its blocks of content, as shown in Figure 2. Single pages are converted into gray-scale images (see Figure 2a), and then processed by the running length algorithm described in [21] to detect regions with high chance of containing information. The algorithm is applied in both horizontal and vertical directions and the resulting binary images are combined using the operator *AND*, as shown in Figure 2b. Next, a 3×3 dilation operation is performed two times over the resulting binary image (see Figure 2c) to create blobs of content.

Finally, we iteratively detect the largest connected component in the binary image and denote it as a block of content. The detection process continues until no more connected components are found in the image. An example of the final result achieved is shown in Figure 2d.

2.2. Classifying blocks of content in the document image

Once the document image is segmented into its blocks of content, we use a CNN model to classify them into three different classes: text, tables and images. In this paper, we actually implemented two different CNN architectures: a bi-dimensional approach commonly used in different computer vision problems, used as a baseline; and the herein proposed fast one-dimensional architecture that uses one-dimension projections to deliver very similar results with much less data usage and processing time, as shown in the results section.

The CNN architectures used in this paper were inspired in the VGG architecture [18] and consists of a number of convolutional layers responsible for computing convolutional features, followed by a number of fully connected layers, which classify the features and generate probabilities for each class of interest.

2.2.1 Using a baseline bi-dimensional CNN based classification

For the bi-dimensional baseline we implemented an architecture that receives as input a bi-dimensional image tile and processes it using a sequence of three 2D convolutional layers with 50 filters and ReLu [15] activation; then evaluate the convolutional features using a fully connected layer with 50 nodes connected to a fully connected layer with 3 nodes and softmax activation (for three classes categorical classification). In this model, each bi-dimensional convolutional layer is followed by a MaxPooling layer [7] with 2 pixels kernel and a 0.1 dropout [19] for regularization. A 0.3 dropout is also present between the two fully connected layers for better model generalization. The convolutional kernel size used trough all the experiments was 3x3 pixels, and a schema for this architecture is depicted in figure 4.

2.2.2 Our approach: a fast 1D CNN based classification

In our approach we propose an one-dimensional CNN architecture that uses image tiles vertical and horizontal projections to classify blocks of content. Text, table and image tiles have very different and highly discriminative signatures for such projections, as depicted in figure 3: text tiles usually have a squared-signal-like shape in the vertical projection, due to text lines, and a roughly constant-signal-like shape in the horizontal projection; table tiles have also a squared-signal-like shape in the vertical projection, but a roughly squared-signal-like shape also in the horizontal projection, due to columns organization; and figures do not hold any special pattern in both horizontal and vertical projections.

The one-dimensional CNN architecture receives as input two one-dimensional arrays corresponding to the horizontal and vertical projections of a given image tile. Each projection is processed by an independent convolutional track composed by a sequence of three 1D convolutional layers with 50 filters each and ReLu [15] activation. The outputs of each track are concatenated and submitted to a fully connected layer with 50 nodes connected to a 3 nodes fully connected layer with softmax activation for three classes categorical classification. In this model, each one-dimensional convolutional layer is followed by a MaxPooling layer [7] with a kernel size of 2 pixels and a 0.1 dropout [19] for regularization. Similar to the bi-dimensional model, there is a 0.3 dropout between the two fully connected layers for better generalization. The convolutional kernel size used trough all the experiments was 3x1 pixels, and the architecture is depicted in figure 4.

It is to be noticed that we used the same amount of layers, filters and kernel sizes in both baseline and one-dimensional architectures, so we could evaluate the impact of reducing the dimension for the analysis and not the parametric differences in the models.

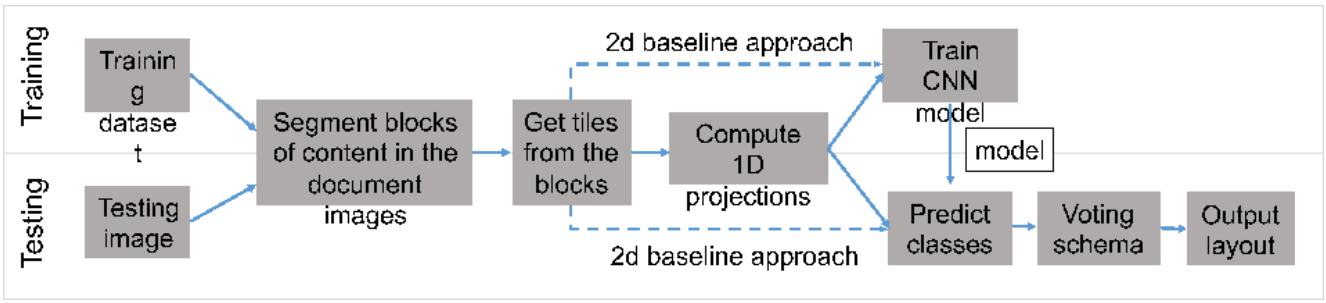


Figure 1. Block-diagram of out proposed methodology for document image layout analysis.

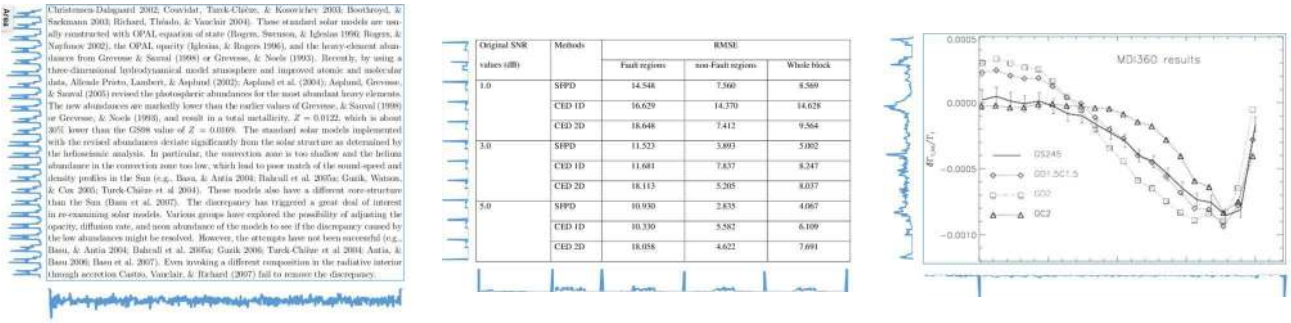
Figure 2. *Pipeline used to break pages into blocks of content.* a) Input grayscale page. b) Binary image resulting from the running length algorithm [21]. c) Binary image resulting from two times dilation by a mask 3×3. d) Resulting blocks of content.

3. Experimental Design

3.1. Data

For running our experiments we built a database from the ArXiv papers filtering the last 300 documents with the word *seismic* in the abstract. The use of seismic keyword is related to reasons beyond the scope in this paper, and we believe our results would be similar for papers in different areas with no loss in generalization.

We segmented the documents into their blocks of content as described in section 2.1, and manually classified them as *text*, *table* or *figure* according to their content. This process derived an annotated database composed of 99 table blocks, 2995 image blocks and 4533 text blocks.



(a) (b) (c)

Figure 3. Examples of text, table and image blocks of content and their corresponding vertical and horizontal projection (in blue): (a) text; (b) image; and (c) table.

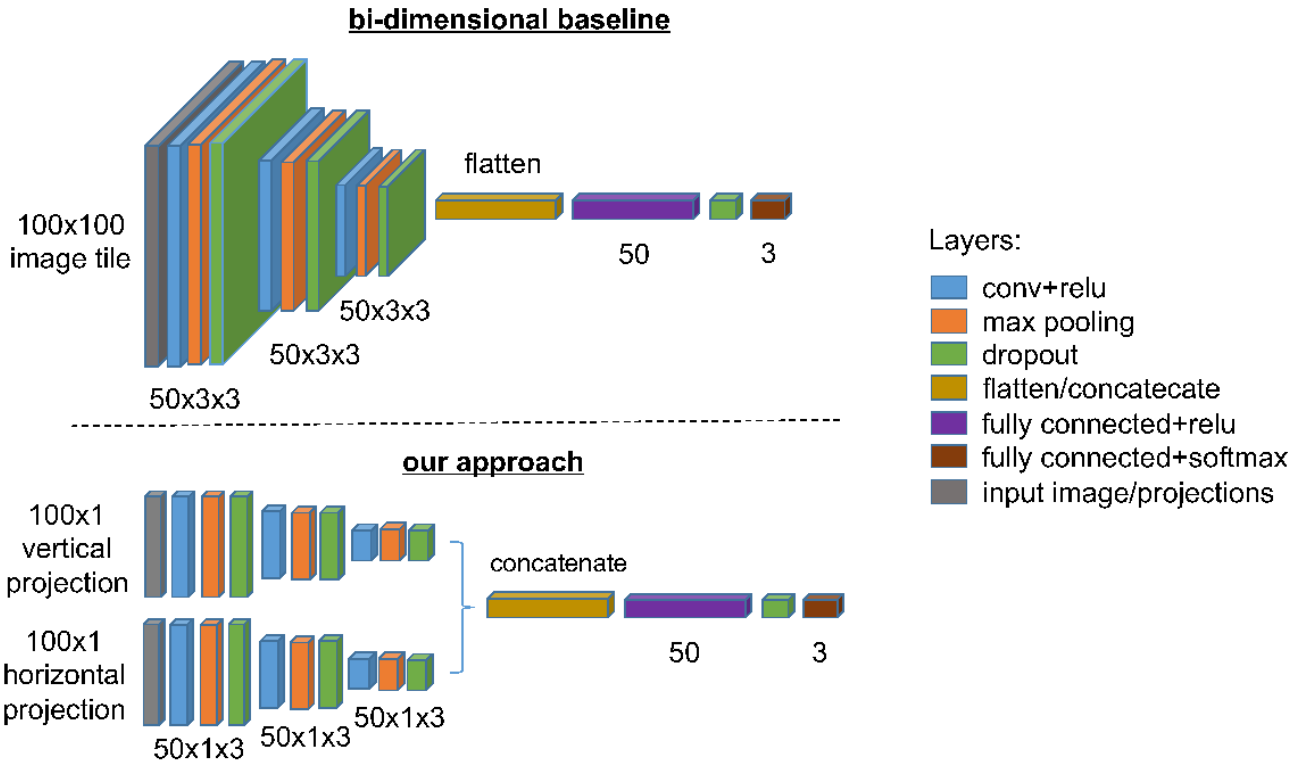


Figure 4. CNN models architectures: bi-dimensional baseline and the proposed one-dimensional approach.

3.2. Training and testing procedures

For dealing with this very unbalanced database and to stress the CNN models, we decided to sub-sample and balance the database for training the CNN models, creating a database composed of 90 randomly selected blocks per class. We performed data augmentation using a 100 pixels per 100 pixels sliding window sampling, with stride of 30. This process derived a training database with roughly 6092 tiles of 100x100 pixels per class, performing a total of 18278 samples. This dataset was further divided in 80% for training (of which we used 20% for validation) and 20% for testing the models classification accuracy. We show these results in section 4.

Once the models were trained, we evaluated them against the whole unbalanced database. In this test, we evaluated not the 100x100 pixel tiles, but the full blocks of contents classification, and for that we used a simple voting schema. For each block of content, we extracted several tiles using the same sliding window schema above, and evaluated them

	2D baseline			Our approach		
	Text	Image	Table	Text	Image	Table
Text	1172	17	12	1186	14	3
Image	32	1144	12	46	1144	22
Table	24	18	1225	12	25	1204

Table 1. Confusion matrix showing the tiles performance of both the bi-dimensional baseline and our one-dimensional approach.

	2D baseline			Our approach		
	Text	Image	Table	Text	Image	Table
Text	4345	49	139	4360	61	112
Image	8	2969	18	15	2920	60
Table	0	0	99	0	0	99

Table 2. Confusion matrix showing the full blocks performance of both the bi-dimensional baseline and our one-dimensional approach.

	Proc. time per document image page (secs)
2D baseline	6.1 ± 0.223
Our approach	0.783 ± 0.078

Table 3. Processing time for classifying blocks of content per page using both the bi-dimensional baseline and our one-dimensional approach.

all against the trained models. The final classification for a given block of content was obtained by computing the highest mean class probability using all tiles in the block, which mimics a probability weighted voting schema. Then, we compared the annotated reference with our outcome and compiled the results in a confusion matrix, shown in section 4.

4. Results and discussion

We organized our results in three different analysis: (i) we compared our model against the bi-dimensional baseline we implemented, using the dataset we created; (ii) then, we compared ourselves with the state-of-art published results; (iii) and finally, we analyzed the errors we observed in our experiments. They are presented in detail in the following subsections.

4.1. Comparing our method against a bi-dimensional baseline

For a fair comparison of our method and the implemented bi-dimensional baseline, we used exactly the same training parameters: 30 training epochs (enough for training convergence in our experiments), a mini-batch of 50 samples and saving only the best trained model. Here we report our first outcome: the training process took 823.82 seconds in the bi-dimensional baseline approach, but only 126.92 seconds in our one-dimensional approach, **6.5 times faster**. The training of both models was performed using a NVidia Tesla K80 GPU.

With the trained models, we firstly assessed how they performed in the 100x100 tiles classification. Recalling section 3.2, we separated 20% of tiles for testing from a total of 18278 samples, corresponding to 3656 samples or roughly 1218 samples per class. The evaluation results are shown in table 1 in the form of a confusion matrix. It is noticeable that both models have good and similar performance: the bi-dimensional baseline achieves an overall accuracy of 96.8%, while our approach achieves 96.6%. Reducing the dimension from bi-dimensional patches to one-dimensional projections have not impacted the overall accuracy in our experiments.

We also evaluated how the two models performed for full blocks of content classification. In this experiment we used the whole unbalanced dataset, comprising 99 table blocks, 2995 image blocks and 4533 text blocks. As stated in section 3.2, we computed the probabilities for each class for all tiles in that block, and chose the class corresponding to the highest mean probability, similarly to a probability weighted voting schema. Our results are presented in table 2 using again a confusion matrix. It is possible to notice that both models perform well: the bi-dimensional baseline achieves an overall accuracy of 97.19%, while our approach achieves 96.75%. Reducing the dimension from 2D patches to 1D projections seems to have not a very significant impact on overall accuracy in our experiments.

To highlight the advantages of using the proposed one-dimensional approach, we also evaluated the processing time for evaluating a new document image. To do so, we computed the processing time spent in classifying the blocks of content for a giving page, and show the results in table 3, where our approach is **7.8 times faster** than the baseline bi-dimensional approach we tested, with virtually no loss in accuracy.

Finally, for completeness, we also show in figure 5 layout results for full-pages classification using our approach. It is possible to see the text blocks in gray, the images in yellow and the tables in blue, and the results are consistent with visual inspection.

4.2. Our method and the state-of-art

We also compared our results with the state-of-art. Here, it is worth mentioning that even though we are using public documents for evaluating our method, and have implemented a baseline for fair comparison, the data used in all methods is not the same, so one should look at it parsimoniously. The comparison of document image analysis methods using the same datasets is not simple because some are paid (UW-III), some are unavailable in their home web-

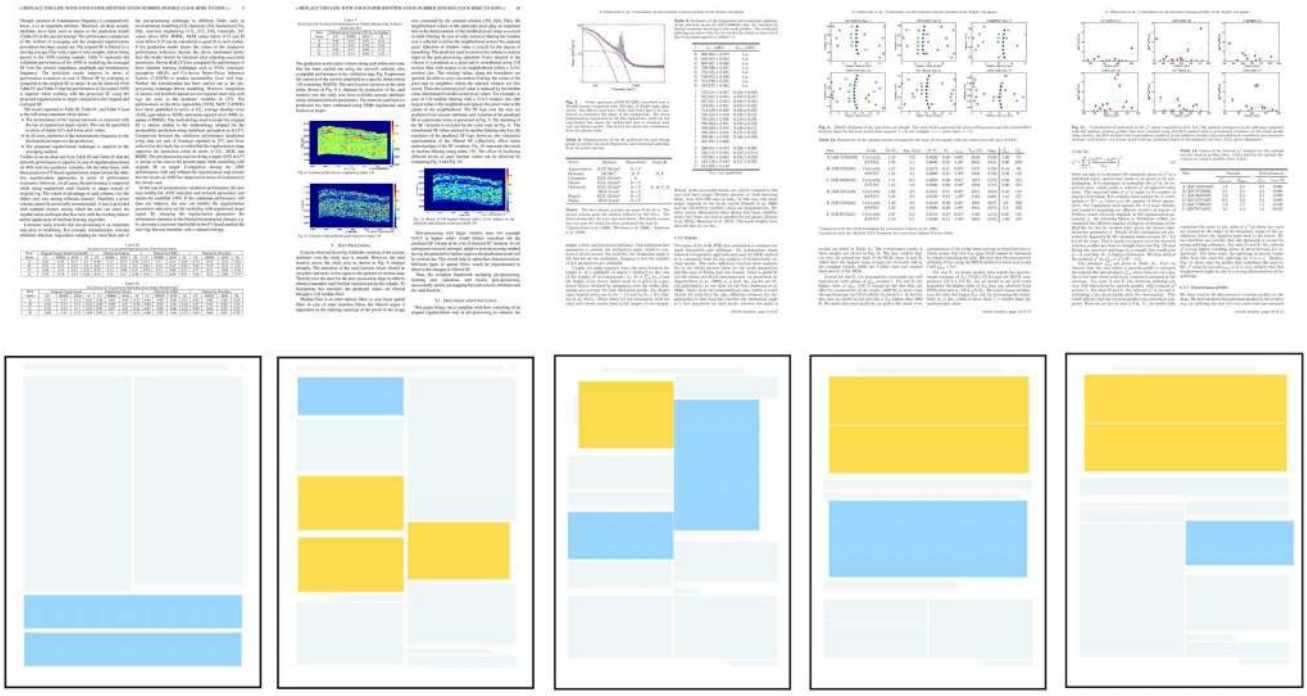


Figure 5. The results achieved using our approach are consistent with visual inspection: in the five pages blocks are correctly segmented and classified as text, image and tables.

		Acc error rate (%)	
Database	Reference	Text	Image
3*MediaTeam	[12]	8.92	12.5
	[3]	21.8	15.3
	[5]	12.9	8.8
3*UW III	[2]	0.49	0.81
	[10]	0.51	0.65
	[16]	1.10	17.7
3*ICDAR 2009	[2]	0.58	1.59
	[10]	7.63	1.10
	[16]	2.80	6.80
Seismic/ArXiv	Our approach	3.81	2.5

Table 4. Comparison with other methods: our method delivers good results in the classes usually handled in the literature.

sites (ICDAR-2009), or do not have the same kind of documents (academic papers) we built in out database (MediaTeam). Still, since the challenge of classifying text and image blocks share many premises, we see value in comparing works in such conditions, and compile them in table 4, where our results are comparable with the state-of-art methods here presented. It is also worth mentioning that results for tables blocks were not included, since this class is not handled by the other compared methods.

4.3. Detailing errors in our experiments

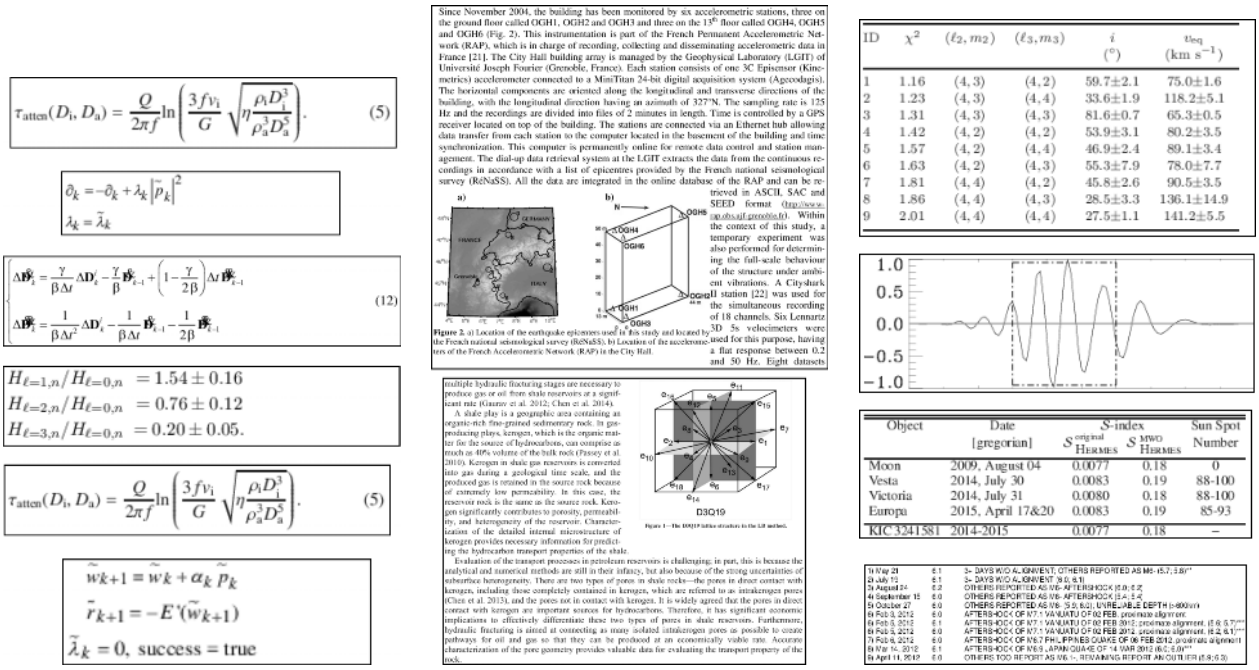
During the evaluation of our results, we found some interesting patterns in the errors. As depicted in figure 6, the mistakes were usually related to either the presence of formulas, annotated as text but consistently classified as images; problems in blocks of content segmentation; or mistakes in the manual annotation (especially regarding tables annotated as text).

Errors related to formulas (see figure 6a) could be theoretically fixed by adding a new formulas class to our models, but this would involve re-annotating the data, and we have not done it in this work. Concerning problems with blocks segmentation, we noticed that sometimes the algorithm we used to segment blocks merges different blocks together, as seen in figure 6b, which leads to mistakes in the blocks classification.

An interesting phenomena is related to errors concerning the manual database annotation, as shown in figure 6c. In this cases, the models actually classified the blocks right, and could be used to refine the database, which is a positive aspect.

5. Conclusions

This paper presents a fast one-dimensional CNN-based approach for document image layout analysis. In our methodology, we first segment blocks of content in doc-



(a) (b) (c)

Figure 6. The most common errors found in our methodology are related to: (a) formulas annotated as text and classified as image; (b) problems in blocks segmentation annotated as text and classified as image; and (c) mistakes in manual annotation: the graph as annotated as text and classified as image, the tables were annotated as text and classified as tables.

ument images, then we divide them into several tiles and compute their horizontal and vertical projections, which were jointly used by a one-dimensional CNN model that classifies them into text, tables and images. Then, the tiles classification outcome in combined a simple voting schema that defines the final class for each block of content in the document image.

For evaluating our model we built a database using publicly available documents from the ArXiv website, where we filtered the last 300 documents with the word *seismic* in the abstract. We segmented the documents pages as described in section 2.1, and manually classified individual blocks of content as *text*, *table* or *figure*. This process derived an annotated database composed of 99 table blocks, 2995 image blocks and 4533 text blocks, which were used through our experiments.

Our model was evaluated against a baseline bi-dimensional CNN-based approach we implemented with a very similar architecture to our approach, which allowed us to assess the impact of reducing the dimension in the models performance. Our results show that our method achieves the same accuracy observed in the bi-dimensional baseline, but with considerably lower corresponding data usage (from 100x100=10M pixels to 2x100=200 pixels, **50 times less**) and processing time (roughly **7.8 times faster** for document page evaluation). These results are potentially useful for mobile applications or cloud-based services.

Further research would involve including other semantic classes for blocks of content classification and tackling different kinds of documents, such historical ones.

References

[1] S. S. Bukhari, M. I. A. Al Azawi, F. Shafait, and T. M. Breuel. Document image segmentation using discriminative learning over connected components. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 183–190, New York, NY, USA, 2010. ACM.

[2] S. S. Bukhari, F. Shafait, and T. M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In *IS&T/SPIE Electronic Imaging*, pages 78740D–78740D. International Society for Optics and Photonics, 2011.

[3] M. S. Erkilinc, M. Jaber, E. Saber, P. Bauer, and D. Depalov. Page layout analysis and classification for complex scanned documents, 2011.

[4] M. S. Erkilinc, M. Jaber, E. Saber, P. Bauer, and D. Depalov. Text, photo, and line extraction in scanned documents. *Journal of Electronic Imaging*, 21(3):033006, July 2012.

[5] M. S. Erkilinc, M. Jaber, E. Saber, P. Bauer, and D. Depalov. Text, photo, and line extraction in scanned documents. *Journal of Electronic Imaging*, 21(3):033006–1, 2012.

[6] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):910–918, Nov. 1988.

[7] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *ICIP*, page in press, 2013.

[8] A. W. Harley, A. Ufkes, and K. G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, ICDAR '15, pages 991–995, Washington, DC, USA, 2015. IEEE Computer Society.

[9] L. Kang, J. Kumar, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for document image classification. In *2014 22nd International Conference on Pattern Recognition*, pages 3168–3172, Aug 2014.

[10] V. P. Le, N. Nayef, M. Visani, J.-M. Ogier, and C. De Tran. Text and non-text segmentation based on connected component features. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1096–1100. IEEE, 2015.

[11] V. P. Le, N. Nayef, M. Visani, J. M. Ogier, and C. D. Tran. Text and non-text segmentation based on connected component features. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1096–1100, Aug 2015.

[12] M.-W. Lin, J.-R. Tapamo, and B. Ndovie. A texture-based method for document segmentation and classification. *South African Computer Journal*, 2006(36):49–56, 2006.

[13] M. A. Moll and H. S. Baird. Segmentation-based retrieval of document images from diverse collections. In *Electronic Imaging 2008*, pages 68150L–68150L. International Society for Optics and Photonics, 2008.

[14] M. A. Moll, H. S. Baird, and C. An. Truthing for pixel-accurate segmentation. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 379–385, Sept 2008.

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

[16] N. Nayef and J.-M. Ogier. Text zone classification using unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 776–780. IEEE, 2015.

[17] O. Okun, D. Doermann, and M. Pietikainen. Page segmentation and zone classification: The state of the art. LAMP-TR-036,CAR-TR-927,CS-TR-4079, 11 1999.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[20] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. Text/graphics separation revisited. In *Proceedings of the 5th International Workshop on Document Analysis Systems V*, DAS '02, pages 200–211, London, UK, UK, 2002. Springer-Verlag.

[21] K. Y. Wong, R. G. Casey, and F. M. Wahl. Document analysis system. *IBM journal of research and development*, 26(6):647–656, 1982.