Transfer Learning for Structures Spotting in Unlabeled Handwritten Documents using Randomly Generated Documents

Geoffrey Roman-Jimenez, Christian Viard-Gaudin, Adeline Granet and Harold Mouchère
University of Nantes, LS2N, UMR 6004, F-44100, France

Keywords: Handwritting Recognition, Image Generation, Digit Detection, Deep Neural Networks, Knowledge Transfer.

Abstract: Despite recent achievements in handwritten text recognition due to major advances in deep neural networks, historical handwritten documents analysis is still a challenging problem because of the requirement of large annotated training database. In this context, knowledge transfer of neural networks pre-trained on already available labeled data could allow us to process new collections of documents. In this study, we focus on localization of structures at the word-level, distinguishing words from numbers, in unlabeled handwritten documents. We based our approach on a transductive transfer learning paradigm using a deep convolutional neural network pre-trained on artificial labeled images randomly generated with strokes, word and number patches. We designed our model to predict a mask of the structures positions at the pixel-level, directly from the pixel values. The model has been trained using 100,000 generated images. The classification performances of our model were assessed by using randomly generated images coming from a different set of images of words and digits. At the pixel level, the averaged accuracy of the proposed structures detection system reach 96.1%. We evaluated the transfer capability of our model on two datasets of real handwritten documents unseen during the training. Results show that our model is able to distinguish most ”digits” structures from ”word” structures while avoiding other various structures present in the documents, showing the good transferability of the system to real documents.

1 INTRODUCTION

This work takes part of the CIRESFI project [1]. The CIRESFI project aims to improve our knowledge about the Italian theater during the 18th century in France. The researchers in human and social sciences intend to reveal the cultural adaptation for the Italian actors in France. Whereas the political situation was against them, they succeeded in establishing themselves as an official and reputed institution. So, this projects aims at providing efficient tools to access to different kinds of information which are included in a set of financial records of the Italian comedy dating from the XVIII century with more 28 000 pages (Luca, 2011)(Cethefi, 2016). Information retrieval within a large collection of handwritten historical documents is a very complicated task, specifically when no ground truth training dataset is available. As a first functionality, we focus on the detection of the structure of these financial documents. In this paper, we are concentrating on localization of all the number areas, distinguished from word or strokes structures, which are one the main constituents of these financial records of the Italian Comedy (Cethefi, 2016). Later works could extract meaningful area and columns based on these typed localizations.

Neural network-based deep learning have recently achieved great advances in the pattern recognition area including classification, regression and clustering (LeCun et al., 2015)(Schmidhuber, 2015). In text recognition, recent works have shown that deep neural network (DNN) models can tackle the automatic detection of specific objects in handwritten documents (Moysset et al., 2016)(Butt et al., 2016). However, since DNN models based their learning on the data, most of the applications need large amount of labeled data to train the models. In text recognition of handwritten document, training data and test data are generally taken from the same corpus of documents, respecting the same text arrangement, same handwriting style and/or similar paper type. Unfortunately,

[1] Project ANR-14-CE31-0017 ”Contrainte et Intégration : pour une Réévaluation des Spectacles Forains et italiens sous l’Ancien Régime”

Roman-Jimenez, G., Viard-Gaudin, C., Granet, A. and Mouchére, H.
Transfer Learning for Structures Spotting in Unlabeled Handwritten Documents using Randomly Generated Documents.
DOI: 10.5220/0006598204170425
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 417-425
ISBN: 978-989-758-276-9

ground-truth in handwritten documents are rarely labeled and manual annotations are very time expensive. When the labels of the target data are not available, a *transductive transfer learning* strategy (Pan and Yang, 2010) can be considered to learn the detection task by training a model with annotated data from another database. Of course, the training dataset should present enough structural variability and representativeness with respect to the target dataset to both learn the detection task and ensure its transferability. Thus, the challenge here is to learn a generic detection task without learning the specific characteristics and patterns present in the training dataset. Once trained on this train dataset, the system should be able to retrieve in the target dataset the same kind of patterns.

In this paper, we propose to bridge the gap between a non-labeled dataset and a labeled dataset by generating randomly artificial images containing known patterns in known locations but with variable layouts and backgrounds. The strategy is based on a random placement of labeled patches of word and number structures on a background image from the target database. Clearly, the main advantage of such strategy is that it allows to generate as many images as necessary, can be adjusted to the target task, and ensure structural variability(Delalandre et al., 2010). Note that this generation strategy is quite close to (Kieu et al., 2013) where authors create full pages of realistic documents. Furthermore, we show that a fully convolutional network can be trained to solve a digit/word detection task.

The paper is structured as follows: Section 2 introduces the notations, definitions and objectives of this work; Section 3 presents the images databases used for the generation of artificial documents and the real handwritten documents for which we want to detect number/word structures; Section 4 presents the architecture of the neural network trained for our pixel-wise digit spotting problem; Section 5 presents how we evaluated the classification performance of our model; Section 6 presents the results obtained with our model classification considering the artificial generated images and a qualitative evaluation on real handwritten mails and historical documents that were not used for training.

2 NOTATIONS, DEFINITIONS AND OBJECTIVES

In this section we introduce some notations and definitions used in this paper.

Let $X = \{x_1, x_2, ..., x_N\}$ with $x_i \in \mathbb{R}$, be the image of N pixels of a given handwritten document. $S = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\}$, with $s_i = (s_i^0\ s_i^1\ s_i^2)^\top$, $s_i^k \in \{0,1\}$ corresponds to the pixel-wise classification map of one-hot vectors indicating in which class belongs each pixel of X. In our context, $k = 0$ corresponds to the class "background", $k = 1$ is the class "number" and $k = 2$ is the class "word". The goal of our model is to build the map $Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, with $\mathbf{y}_i = (y_i^0\ y_i^1\ y_i^2)^\top$, $\mathbf{y}_i^k \in [0,1]$ as the closest estimation of S, from the image X.

Using a set of M images $\mathcal{X} = \{X_1, X_2, ..., X_M\}$, the corresponding set of structures maps S and a neural network model with the parameters Θ, we aim to learn a transformation function $\mathcal{T}(X,\Theta) = Y$ by finding the parameters $\Theta^\star$ that minimize the cost function C,

$$\Theta^\star = \underset{\Theta}{\operatorname{argmin}}\, C(\mathcal{T}(X,\Theta), S). \qquad (1)$$

As the targeted $Y = \mathcal{T}(X,\Theta)$ should be the closest estimation of S, we define the cost function C as the weighted cross entropy between Y and S:

$$C(Y,S) = -\sum_{i}^{N}\sum_{k=0}^{2}\frac{1}{p_i^k}(s_i^k.log(y_i^k)), \qquad (2)$$

where the weighting coefficient p_i^k is the probability of the pixel x_i to belong to the class k. This is expected to readjust the weight of error depending on the proportion of each class within each image X. Given an image X, we computed p_i^k as the proportion of the pixels belonging to the class k within the image X.

Considering a set of unlabeled image $\mathcal{X}^u$, the set of maps of one-hot vectors $\mathcal{S}^u$ is not available to learn the transformation $\mathcal{T}^u(\mathcal{X}^u,\Theta) = \mathcal{Y}^u$. The objective of transfer learning is thus to learn a transformation $\mathcal{T}^l(\mathcal{X}^l,\Theta) = \mathcal{Y}^l$ using a set of labeled images $\mathcal{X}^l$ that is similar enough to $\mathcal{X}^u$ to ensure that $\mathcal{T}^l(\mathcal{X}^u,\Theta) \approx \mathcal{Y}^u$.

3 DATABASES

3.1 Real Images

Three databases were considered in this study: the IReste Online Offline database (IROnOff) (Viard-Gaudin et al., 1999), the handwritten mail of the RIMES database (Augustin et al., 2006) and the unlabeled registers accounts of Italian Comedy (RECITAL) (Cethefi, 2016). RECITAL images were scanned by the BNF (National French Library)[2] at 400dpi.

[2] As an example, the register numbered 41 is available here: http://catalogue.bnf.fr/ark:/12148/cb42447323f

We used the IROnOff database to construct $\mathcal{X}^l$ and $\mathcal{S}^l$ to learn the transformation $\mathcal{T}^l$. The RECITAL database corresponds to the set of images $\mathcal{X}^u$ for which we want to process automatic structures-spotting.

The IROnOff database contains a total of 61,291 images (300dpi) of isolated handwritten characters, digits, words, with their corresponding transcriptions. We randomly separated the IROnOff database in two groups; the training set D_{train} corresponding to 67% of the total dataset (40,861 images) and the test set D_{test} corresponding to 33% of the dataset (20,432 images). To evaluate the model during the learning, we picked 20% of D_{train} as the validation set D_{valid} (8,172 images).

Besides, we used the set of patches from the RIMES database to quantitatively evaluate the retrieval capability of our classifier. The RIMES database is a set of 1,500 images of labeled paragraphs from handwritten mails (Grosicki and El-Abed, 2011) with a total of 66,979 patches of words and numbers present in the documents. In RIMES, patches with numbers correspond to isolated numbers and identification strings using digits (as license plates). A total of 918 patches contains digits.

3.2 Random Image Generation

We used images of D_{train}, D_{valid} and D_{test} as patches to generate 1536x1536 images and created the set of artificial images $\mathcal{X}^l_{train}$, $\mathcal{X}^l_{valid}$ and $\mathcal{X}^l_{test}$ with their associated pixel-wise one-hot vector maps $\mathcal{S}^l_{train}$, $\mathcal{S}^l_{valid}$ and $\mathcal{S}^l_{test}$.

In order to generate images that are comparable with historical documents, we selected a total of 97 images from the RECITAL dataset without any handwriting, and used them as background. We also manually segmented 50 handwritten strokes (accolades, separators, ...) from the RECITAL dataset for adding various structures, distinct from the word/number patches.

The procedure to generate an image is defined by Algorithm 1. The first part of the procedure randomly create a grid G in a 1536x1536 area using a randomly selected background image. Secondly, for each cell of the created grid, a type of patch is selected among $\{empty, word, number\}$. If the type is not *empty*, a patch of the corresponding type is then selected, randomly scaled, and placed in the current cell. Note that the random scaling is constrained to keep the patch in the size of the cell. The corresponding ground-truth map S is built at this stage using the type of cell and its position and size. Then, a random number of strokes are placed on the created document, after random scale and rotation. Finally, a Gaussian noise with a random signal-to-ratio (from 10 to 100) is added to the artificial document.

Algorithm 1: Artificial document generation algorithm.

```
procedure ARTIDOC
#Inputs
    $D$: dataset of word/number patches
    $B$: dataset of background images
    $S$: dataset of strokes patches
    $MinH$: minimum height of patches
    $MinW$: minimum width of patches
    $(SizeH, SizeW)$: size of generated image

#Random statement
    $b \leftarrow$ random image in $B$
    $A \leftarrow$ random area $(SizeH, SizeW)$ in $b$
    $W \leftarrow \mathcal{U}(1, SizeW/minW)$
    $H \leftarrow \mathcal{U}(1, SizeH/minH)$
    $C \leftarrow$ grid $(W \times H)$ of $A$

#Random patches placement
    for each cell $c \in C$ do
        $t \leftarrow$ random type of patch $\in \{\emptyset, word, number\}$
        $d \leftarrow$ random patch $\in D$ of type $t$
        $d' \leftarrow$ random scaling $d$ (keeping it in $c$)
        $A(c) \leftarrow$ random placement of $d'$ in $c$

#Random strokes placement
    $nStroke \leftarrow \mathcal{U}(0, W \times H)$
    for $n \leftarrow 1, 2, ..., nStroke$ do
        $s \leftarrow$ random stroke $\in S$
        $s \leftarrow$ random scaling and rotation of $s$
        $A \leftarrow$ random placement of $s$

#Add noise
    $SNR \leftarrow \mathcal{U}(10, 100)$
    $\sigma^2_{noise} \leftarrow \sigma^2_{signal} \cdot 10^{-\frac{SNR}{10}}$
    $Artidoc \leftarrow \mathcal{N}(0, \sigma^2_{noise})$

#Output
    return $Artidoc$
```

Note:
$\mathcal{U}(\alpha, \beta)$: Discrete Uniform Distribution (from α to β)
$\mathcal{N}(\mu, \sigma^2)$: Normal Distribution with mean μ and variance σ^2

Note that the number of generated cells c in the grid is chosen to generate sizes of numbers/words in the same range than in the RECITAL images (this is an approximation as numbers and words of different sizes are presents in each original page). Fig. 1 shows three examples of artificially generated documents. Source code for the artificial document generator is available at https://github.com/GeoTrouvetout/CIRESFI.

A total of 100,000 images were generated on-the-fly for $\mathcal{X}^l_{train}$ to train our model, 100,000 images for $\mathcal{X}^l_{valid}$ to validate the model during training and 10,000 images for $\mathcal{X}^l_{test}$ to test the model. It means that each artificial image is used only once.

4 NEURAL NETWORK MODELING AND TRAINING

The model is based on a fully-convolutional neural network (FCNN) that produces a pixel-wise classification of every pixels of the input image in three classes; background, number and word. A graphical representation of the architecture of the trained FCNN if presented on Fig. 2 and detailed in TABLE 1. Since it is fully convolutional, this model is adaptable to all input image sizes so that it can be applied on the mails from the RIMES database and on the unlabeled documents from the RECITAL database $\mathcal{X}^u$. A 5-level pyramid representation of the input image was used as input to enlarge the reception field of each feature maps and ensure that the network can handle recognition of various sizes of structures. Roughly, the network is composed of two parts: Features extraction and Structures map construction. The features extraction part composed of 5 layers of convolution (kernel size of 5x5 with zero-padding) linked with 2x2 max-pooling leading to a middle layer shape of 48x48x256. The digit mask construction part is composed of 6 layers of transposed convolution (Dumoulin and Visin, 2016) (kernel size of 5x5 and padding with half of the filter size on both sides) and two 2x2 upscale layers (upscaling by repetition) reconstructing a 384x384x3 tensor finally upscaled to 1536x1536x3. The rectify linear unit (ReLU) function was chosen as the output nonlinearity of each layer, with the exception of the output layer with a softmax function. Originally proposed by Nair and Hinton in (Nair and Hinton, 2010), ReLU function is define as $ReLU(x) = max(0,x)$. Besides it does not require input normalization, the ReLU function has been shown to help the training speed of FCNN models (Nair and Hinton, 2010)(Krizhevsky et al., 2012). The softmax function performed a exponential normalization of each one-hot vector of the produced map, thus $\forall i, y_i^0 + y_i^1 + y_i^2 = 1$, defined by

$$\text{Softmax}(x)_j = \frac{e^x_j}{\sum_k e^x_k} \text{ with } k = 0,1,2.$$

The model was built and trained using the Python library *Theano* (Bergstra et al., 2010) and the deep learning wrapper *Lasagne* (Dieleman et al., 2015).

We trained the model by minimizing the objective function C (equation 2) using the *Adam* stochastic gradient-based algorithm described in (Kingma and Ba, 2014).

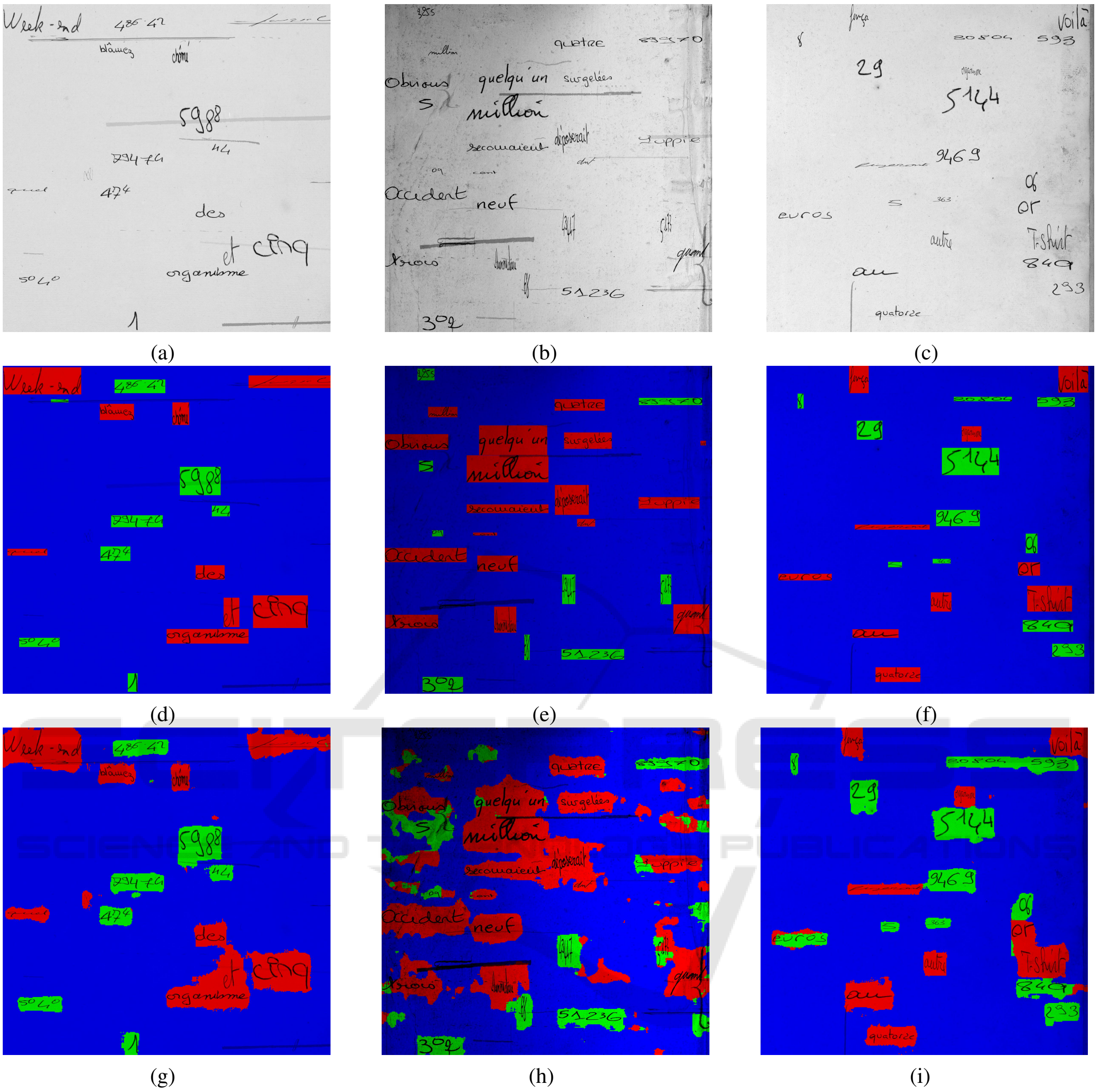
Symmetrically to the set of inputs images $\mathcal{X}^l_{train}$, $\mathcal{X}^l_{valid}$ and $\mathcal{X}^l_{test}$, the set of output maps of our network are denoted $\mathcal{Y}^l_{train}$, $\mathcal{Y}^l_{valid}$ and $\mathcal{Y}^l_{test}$. Note that the direct output of the network is $Y = \{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_N}\}$ with $y_i = (y_i^0 \; y_i^1 \; y_i^2)^\intercal$ and $y_i^k \in [0,1]$. To evaluate the classification performance of our model with binary outputs, we computed the resulting classification map $\hat{S} = \{\hat{\mathbf{s}}_\mathbf{1}, \hat{\mathbf{s}}_\mathbf{2}, ..., \hat{\mathbf{s}}_\mathbf{N}\}$ with

$$\hat{\mathbf{s}}_\mathbf{i} = \begin{pmatrix} \hat{s}_i^0 \\ \hat{s}_i^1 \\ \hat{s}_i^2 \end{pmatrix}, \hat{s}_i^k = \begin{cases} 1 & \text{if } max(\mathbf{y_i}) = y_i^k \\ 0 & \text{else} \end{cases} .$$

Table 1: Architecture of our model based on convolutional neural network.

Layer type	Filter size	Output layer shape	Activation function
Input image	///	1536x1536	///
$Py(X)$	///	1536x1536x5	///
Convolution + maxPool (2x2)	5x5x32	768x768x32	ReLU
Convolution + maxPool (2x2)	5x5x32	384x384x32	ReLU
Convolution + maxPool (2x2)	5x5x64	192x192x64	ReLU
Convolution + maxPool (2x2)	5x5x128	96x96x128	ReLU
Convolution. + maxPool (2x2)	5x5x256	48x48x256	ReLU
Convolution. + maxPool (2x2)	5x5x128	96x96x128	ReLU
Trans. Conv. + upscale (2x2)	5x5x64	192x192x64	ReLU
Trans. Conv. + upscale (2x2)	5x5x32	384x384x32	ReLU
Trans. Conv.	5x5x16	384x384x16	ReLU
Trans. Conv.	5x5x8	384x384x8	ReLU
Conv. + upscale (4x4)	5x5x3	1536x1536x3	Softmax
Output map	///	1536x1536x3	///

Note: ReLU corresponds to the Rectified Linear Unit function defined by $ReLU(x) = max(0,x)$. Softmax corresponds to the normalized exponential function defined as $Softmax(x)_j = \frac{e^x_j}{\sum_k e^x_k}$ with $k = 0,1,2$. Transposed Convolution performs the backward pass of a normal convolution as described in (Dumoulin and Visin, 2016).


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 1: Three examples of generated images with randomly selected background, word and digit patches from the IROnOff database. (a),(b) and (c) present three examples of 1536x1536 artificial document generated using the Algorithm 1. (d), (e) and (f) present the target classification map (bounding boxes of the numbers and words) superposed on the corresponding artificial document. (g), (h) and (i) are the corresponding outputs of the detection system.

5 EVALUATION

We evaluated the performance of the structures-spotting pixel-wise localization carried out by our model in three phases: firstly, by considering the set of artificial generated images $\mathcal{X}^l_{test}$ and the corresponding set of target classification map $\mathcal{S}^l_{test}$; secondly, by evaluating the detection of numbers among the N_R patches of the RIMES database, $\mathcal{R} = \{R_1, R_2, ..., R_{N_R}\}$, and their associated states $\mathcal{B} = \{b_1, b_2, ..., b_{N_R}\}$, $b_i \in \{0, 1\}$, indicating the presence or not of a number.

Given an output $\hat{S}$ map (estimated segmentation) and a ground-truth structures map S, we computed the precision, recall and accuracy of the structures classification at the pixel level. Because the images contains significantly more background than numbers or words, we also computed the Matthew Correlation

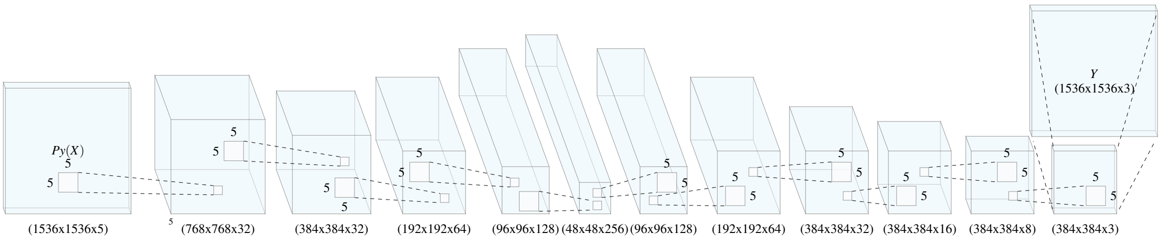


Figure 2: Graphical representation of the architecture of the fully-convolutional neural network. $Py(X)$ corresponds to the pyramid representation of the input image X with 5 levels of resolutions. A filter size of 5x5 were used for both convolution and transposed convolution layers. Each convolution layer is associated with a max-pooling layer of 2x2. The two first transposed convolution layers are associated with a nearest-neighbor upscale layer of 2x2.

Coefficient (Matthews, 1975) extended for multiclass classification by J. Gorodkin in (Gorodkin, 2004), which have the advantage of taking into account the balance between the number of pixels belonging to classes. Note that the Matthew correlation coefficient is ranging from -1 to +1 with a value of 1 representing a perfect classification, 0 corresponds to random classification and -1 indicates total disagreement between prediction and the ground-truth.

We describe below how we computed these metrics of classification performance namely precision (PRE), recall (REC), accuracy (ACC) and Matthew correlation coefficient (MCC):

$$PRE_k = \frac{C_{kk}}{\sum\limits_l C_{lk}}, \quad REC_k = \frac{C_{kk}}{\sum\limits_l C_{kl}}, \ ACC = \frac{\sum\limits_k C_{kk}}{\sum\limits_{k,l} C_{kl}},$$

$$mPRE = \frac{1}{K}\sum_k PRE_k, \qquad mREC = \frac{1}{K}\sum_k REC_k,$$

$$MCC = \frac{\sum\limits_{k,l,m} C_{kk}.C_{lm} - C_{kl}C_{mk}}{\sqrt{\sum\limits_k (\sum\limits_l C_{kl})(\sum\limits_{\substack{k',l' \\ k'\neq k}} C_{k'l'})}.\sqrt{\sum\limits_k (\sum\limits_l C_{lk})(\sum\limits_{\substack{k',l' \\ k'\neq k}} C_{l'k'})}}, \tag{3}$$

where $K = 3$, k, l and $m \in \{0,1,2\}$ and C being the 3x3 multiclass confusion matrix for one image. Note that $mPRE$ and $mREC$ correspond to averaged precision and recall computed for each class versus the others.

To focus the performance measures on handwriting, and evaluate more precisely the structures retrieval, we also computed these measures by weighting each classification with the ink amount of the pixels. Considering a pixel x_i of an image X, its ink amount τ_i is computed as the normalized pixel intensity $\tau_i = \frac{x_i - min(X)}{max(X) - min(X)}$. With this definition, $\tau \in [0,1]$, $\tau = 0$ corresponds to a white pixel, and $\tau = 1$ corresponds to a black pixel.

Besides, we also computed precision, recall and accuracy to evaluate the digits detection capacity of our classifier, at a word-level, using the RIMES word patches $\mathcal{R}$. The presence or not of a digit in a patch (set $\mathcal{B}$) was stated when the area formed by the pixels classified in the class ”*number*” was equal or greater than 25 pixels.

6 RESULTS

6.1 FCNN Training

Fig. 3 shows the evolution of losses $C(X^l_{train}, Y^l_{train})$ and $C(X^l_{valid}, Y^l_{valid})$ during the training of our model. The concomitant diminution of losses on both D_{train} and D_{valid} shows that the variability of random artificial document preventing our model from overfitting. Because of the trending cost reduction, we choose to keep the last computed weights. Fig. 1(g), Fig. 1(h) and Fig. 1(i) present three output examples of the resulting system.

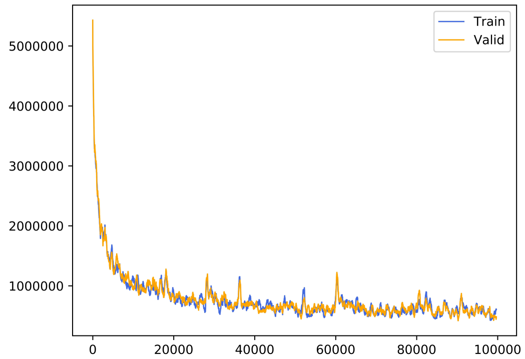


Figure 3: Evolution of losses $C(S_{train}, Y_{train})$ and $C(S_{valid}, Y_{valid})$ obtained during the training of the FCNN model.

6.2 Structure-spotting Evaluation on Artificial Images

TABLE 2 presents the confusion matrix obtained on the set of artificial images $\hat{S}_{test}$. Note that the number of occurrences in each cell of the confusion matrix are presented in averaged percentage of pixels among images of $\hat{S}_{test}$. This shows the unbalance between the number of occurrences for each class. Globally, we can see that classes 1 (number) and 2 (word) are over estimated (e.g 2.72% of pixels are classified as numbers but only 1.5% of pixels are really numbers). This effect is due to the balance cost function (eq. 2) which prevents the background class from over-estimation. TABLE 3 resumes the averaged classification results computed on $\hat{S}_{test}$. The averaged recall, precision and accuracy reach the percentages of 97.4%, 69.9% and 96.6% respectively. The averaged Matthew correlation coefficient was 0.738, showing a robust global performance of the system. The standard deviations are computed over the different images from the test set $\mathcal{Y}^{l}_{test}$. Considering each class versus the others, we can observe poor precision despite good recall measures for word and number structures. This can be explained by the fact that among pixels correctly detected as word/number structures, numerous neighboring background pixels are included in these classes as well. The measurements using the signal-weighted classification allow to increase the precision for the 3 classes. It means that the low precision is mostly due to white pixels which are detected. This can be observed in Fig. 1g), Fig. 1(h) and Fig. 1(i).

Table 2: Averaged matrix confusion obtained on artificial images of the test set. The numbers of occurrence are presented in averaged percentage of pixels over the images of $\hat{S}_{test}$.

		Predicted class			
		0	1	2	Total
Actual class	0	92.82%	1.2%	1.76%	95.78%
	1	0.01%	1.47%	0.02%	1.50%
	2	0.04%	0.05%	2.64%	2.73%
	Total	92.87%	2.72%	4.42%	

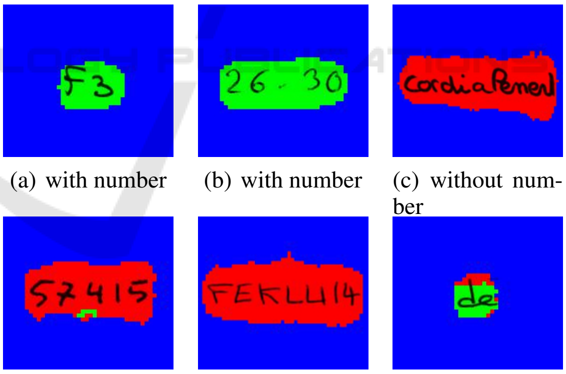
6.3 Transfer Learning Evaluation on Real Handwritten Documents

6.3.1 Word Level Evaluation

We measured the number detection performance on the set of patches of the RIMES database $\mathcal{R}$. With the 66,797 patches of $\mathcal{R}$, the Matthew correlation coefficient was 0.634 and the recall, precision and accuracy were 80%, 75.6% and 82.5% respectively. It means that 80% of patches including digits are retrieved but only 75.6% of detected images really contain digits. Noting that no pre-processing (binarization, denoising, reshape, etc.) were applied on the patches, these results show that the system can handle all types of data and distinguish number structures among handwritten documents. The Fig. 4 shows some examples of some well-classified and miss-classified images.

Table 3: Averaged classification performances on artificial images (test set $\hat{S}_{test}$)

Measures	pixel classification	signal-weighted pixel classification
mREC	0.974 ± 0.034	0.975 ± 0.034
mPRE	0.699 ± 0.055	0.785 ± 0.061
ACC	0.969 ± 0.020	0.969 ± 0.019
MCC	0.738 ± 0.057	0.816 ± 0.058
REC_0	0.969 ± 0.021	0.966 ± 0.022
REC_1	0.985 ± 0.058	0.981 ± 0.056
REC_2	0.969 ± 0.083	0.957 ± 0.082
PRE_0	0.999 ± 0.002	0.999 ± 0.002
PRE_1	0.518 ± 0.114	0.661 ± 0.127
PRE_2	0.579 ± 0.107	0.693 ± 0.103



(a) with number (b) with number (c) without number

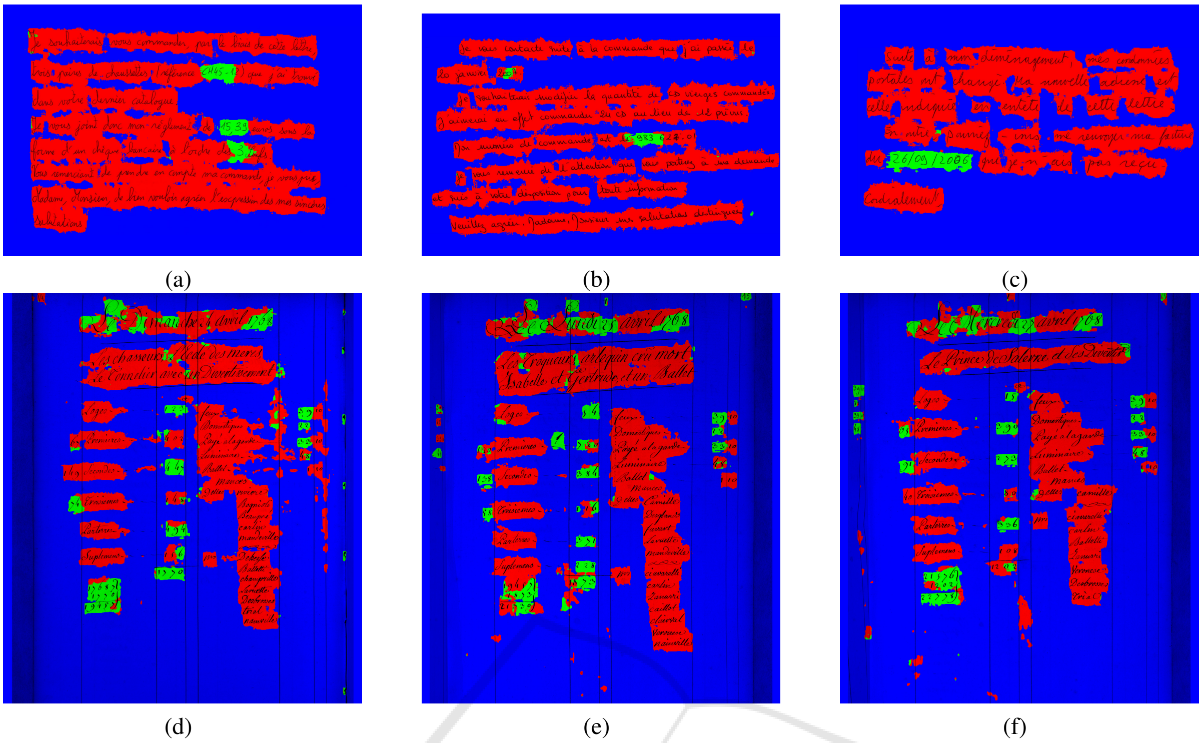
(d) with number (e) with number (f) without number

Figure 4: Examples of word classified images from Rimes. Images (a), (b) and (c) are well-classified and (d), (e), (f) are miss-classified. Ground-truth is specified below each image.

6.3.2 Page Level Qualitative Evaluation

We qualitatively analyze the transferability of our model by considering RIMES paragraphs and RECITAL complete pages. Note that none of these documents were seen during the training.

Fig. 5 shows six examples of structures spotting



(a) (b) (c)

(d) (e) (f)

Figure 5: Examples of the structures spotting performed by our model on real handwritten documents. (a), (b) and (c) unlabeled handwritten mails of the RIMES database superposed with their corresponding classification maps. (d), (e) and (f) historical handwritten documents of the RECITAL database superposed with their corresponding classification maps. (blue), (green) and (red) pixels correspond to classes *background*, *number* and *word*, respectively.

on real handwritten documents; three mails of the RIMES database and three RECITAL documents.

Concerning the RIMES mails, we can observe that the totality of the word structures were well retrieved. However, in Fig. 5(b), we can note that some numbers structures (especially the ”20 and ”2007” within the first sentence) were partially missed by the network. This could be due to the cursively-writing style of these structures, rarely present within the patches containing number used during training.

The RECITAL documents are more challenging because of the natural noise included in the background, the different sizes of characters within the same page, and mainly the writing style which is completely different from modern IROnOff dataset. Despite that most of the numbers were retrieved, we can observe that calligraphic letters, present at the top of the pages, are often detected as digits. This can be explained by the fact that the IROnOff database does not contains calligraphic writing. Thus, our model did not learn to distinguish such large structures from digits. Also, we can observe that, at the pixel level, some number structures starting with ”10” are missed by the network and classified as words. We think that the model cannot differentiate ”10” structures from ”lo”, ”la” and ”le” structures, often present in IROnOff word patches.

7 CONCLUSION

In this paper, we proposed a method for word/number structures spotting in handwritten documents based on a fully-convolutional neural network trained on artificial documents.

Since the targeted database of documents were not annotated, we tackled this as a transfer learning problem. We proposed to train the model on randomly generated labeled images, built with number/word patches of the IROnOff database and page backgrounds from the RECITAL database.

The performance of our model was assessed as a pixel-wise classification of the structures within a set of artificial documents. On artificial data, the system was able to correctly classify 96.9% of the pixels. This shows that our model performed robust pixel-wise classification on artificial data.

We also evaluated the model as a digit detector by using the word/number patches of the RIMES database. The model detected 80% of the patches containing digits. Thus, the learned model is showed to be transferable to real handwritten documents, without any preprocessing.

On unlabeled historical RECITAL documents, our model detected most of the word/number structures. However, we showed that the model hardly distinguish word from number structures in calligraphic structures. Also, the model seems to miss few number structures, mainly due to the confusion between ”1” and ”l” structures.

Despite these limitations, considering the high variability of the RECITAL documents, in terms of shape, structure and handwriting style, we observed a good transferability of our model.

To improve the classification performance of our model on unlabeled data, future work will focus on adding more variability of handwriting styles and structures in the generation of artificial documents. Then, this classification map will be embedded in a larger document analysis system.

Source codes for the artificial document generator and for the structure detection system are available at https://github.com/GeoTrouvetout/CIRESFI.

REFERENCES

Augustin, E., Brodin, J.-m., Carré, M., Geoffrois, E., Grosicki, E., and Prêteux, F. (2006). RIMES evaluation campaign for handwritten mail processing. In *Proc. of the Workshop on Frontiers in Handwriting Recognition*, number 1.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7.

Butt, U. M., Ahmad, S., Shafait, F., Nansen, C., Mian, A. S., and Malik, M. I. (2016). Automatic signature segmentation using hyper-spectral imaging. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 19–24. IEEE.

Cethefi, T. (2016). Project anr-14-ce31-0017 ”contrainte et intégration : pour une réévaluation des spectacles forains et italiens sous l’ancien régime”.

Delalandre, M., Valveny, E., Pridmore, T., and Karatzas, D. (2010). Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems. *International journal on document analysis and recognition*, 13(3):187–207.

Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S. K., Nouri, D., Maturana, D., Thoma, M., Battenberg, E., Kelly, J., Fauw, J. D., Heilman, M., de Almeida, D. M., McFee, B., Weideman, H., Takács, G., de Rivaz, P., Crall, J., Sanders, G., Rasul, K., Liu, C., French, G., and Degrave, J. (2015). Lasagne: First release.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Gorodkin, J. (2004). Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5):367–374.

Grosicki, E. and El-Abed, H. (2011). ICDAR 2011: French handwriting recognition competition. In *Proc. of ICDAR*, pages 1459–1463.

Kieu, V. C., Journet, N., Visani, M., Mullot, R., and Domenger, J.-P. (2013). Semi-synthetic Document Image Generation Using Texture Mapping on Scanned 3D Document Shapes. In *The Twelfth International Conference on Document Analysis and Recognition*, United States.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Luca, E. D. (2011). *Le Répertoire de la Comédie-Italienne (1716-1762)*.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Moysset, B., Louradour, J., Kermorvant, C., and Wolf, C. (2016). Learning text-line localization with shared and local regression neural networks. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 1–6. IEEE.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Viard-Gaudin, C., Lallican, P. M., Knerr, S., and Binter, P. (1999). The ireste on/off (ironoff) dual handwriting database. In *Document Analysis and Recognition, 1999. ICDAR’99. Proceedings of the Fifth International Conference on*, pages 455–458. IEEE.