

# A deep learning-based compatibility score for reconstruction of strip-shredded text documents

Thiago M. Paixão<sup>\*†</sup>, Rodrigo F. Berriel<sup>\*</sup>, Maria C. S. Boeres<sup>\*</sup>, Claudine Badue<sup>\*</sup>,  
Alberto F. De Souza<sup>\*</sup> and Thiago Oliveira-Santos<sup>\*</sup>

<sup>\*</sup>Universidade Federal do Espírito Santo (UFES)

<sup>†</sup>Instituto Federal do Espírito Santo (IFES)

<sup>†</sup>Email: paixao@gmail.com

**Abstract**—The use of paper-shredder machines (mechanical shredding) to destroy documents can be illicitly motivated when the purpose is hiding evidence of fraud and other sorts of crimes. Therefore, reconstructing such documents is of great value for forensic investigation, but it is admittedly a stressful and time-consuming task for humans. To address this challenge, several computational techniques have been proposed in literature, particularly for documents with text-based content. In this context, a critical challenge for automated reconstruction is to measure properly the fitting (compatibility) between paper shreds (strips), which has been observed to be the main limitation of literature on this topic. The main contribution of this paper is a deep learning-based compatibility score to be applied in the reconstruction of strip-shredded text documents. Since there is no abundance of real-shredded data, we propose a training scheme based on digital simulated-shredding of documents from a well-known OCR database. The proposed score was coupled to a black-box optimization tool, and the resulting system achieved an average accuracy of 94.58% in the reconstruction of mechanically-shredded documents.

## I. INTRODUCTION

The paper shredder machine was invented in the early 1900's [1] with the purpose of mechanically destroying waste paper to make their content unintelligible. This kind of device is still commonly seen today in different organizational environments, where huge amounts of documents must be constantly disposed preventing to disclose the information they contain. Besides, paper shredders have become popular for personal use due to the decreasing in prices, which allows people to manage personal files more safely.

However, document destruction is also historically associated to espionage cases, as in the 1979 Iran hostage crisis [2], and in the documents left behind by the official state security service of the former East Germany (*Stasi*) after the fall of the Berlin Wall [3]. Additionally, shredding may be illicitly motivated when the objective is to destroy evidence of fraud and other sorts of crimes. In this context, revealing the original content of shredded papers is of great relevance for the forensic investigation, which can be achieved by first joining coherently the paper strips as in a jigsaw puzzle. This procedure is portrayed in the fictional Netflix production “The Mechanism”<sup>1</sup>, where criminal evidence was exposed after a

<sup>1</sup>A fictional series inspired in the “Carwash”, the largest anti-corruption operation ongoing in Brazil. Official trailer (accessed on 2017-06-15): <https://www.youtube.com/watch?v=130TvUxOeUU>.

police chief has manually spliced dozens of paper strips.

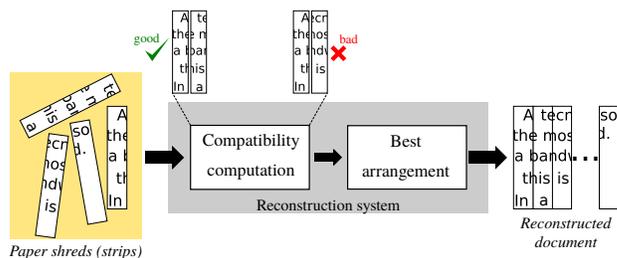


Fig. 1. Overview of a typical reconstruction framework.

Despite its relevance, the manual reconstruction task can be tiresome and time-consuming even for a few paper fragments. Besides, touching the documents should be restricted as much as possible to avoid damage on paper and fingerprint contamination. To deal with these issues, it is highly desirable to automate the reconstruction process, which now would be applied over digitized strips of shredded documents. Typically, the digital reconstruction problem can be subdivided into two main subproblems [4], as depicted in Figure 1: local strips compatibility evaluation, and the determination of the best strips arrangement. The first subproblem, which is the focus of this paper, consists in using image content to infer whether (or how much) any two strips fit each other. The second subproblem can be viewed as a combinatorial optimization search that aims to output the maximum compatibility (or minimum cost) arrangement of strips based on the previously computed local compatibilities. Such arrangement can be 1-D for strip-shredded documents (Figure 2a), or 2-D for cross-cut documents (Figure 2b).

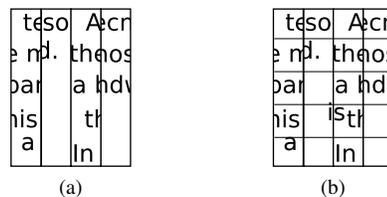


Fig. 2. Shredding type: (a) Strip-cut; (b) Cross-cut.

Several approaches have been proposed along the years in order to perform (semi-)automatic reconstruction. The

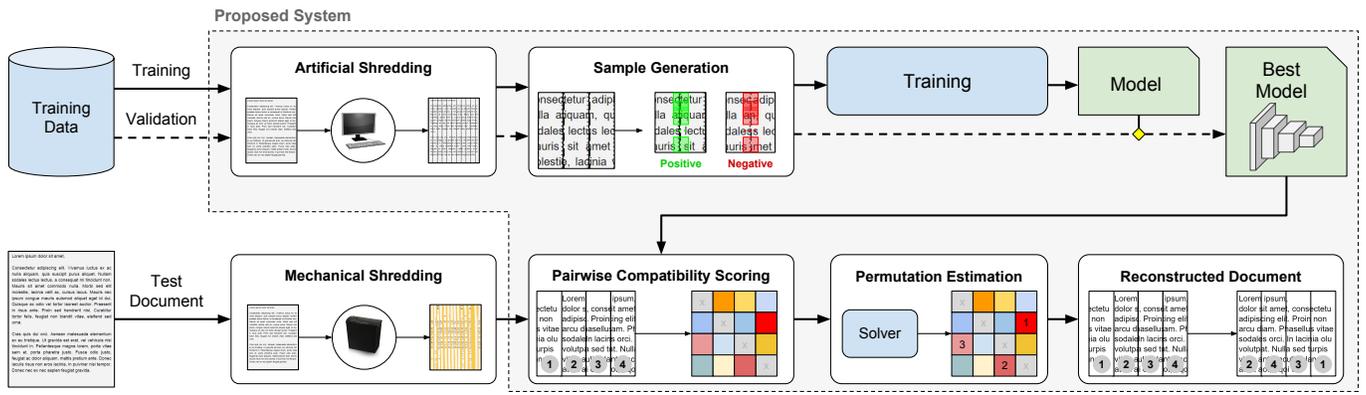


Fig. 3. Overview of the proposed system. There are two major pipelines: training of the deep learning model and reconstruction of documents. The first pipeline (top flow) comprises the generation of training data from artificially-shredded documents followed by the training of the model, where the best model is chosen using a validation set. The reconstruction pipeline (bottom flow) comprises the evaluation on mechanically-shredded documents (including the creation of a new database), where the best model is used to compute the compatibility score between two strips. After the matrix computation, a solver is employed finally outputting a permutation that best reconstructs the document.

pioneering work of Ukovich *et al.* [5] leverages MPEG-7 features to match strips in a content-based retrieval system. Specific text-based features were late introduced in [6]. The reconstruction step itself, i.e., the optimization process, is not addressed in any of these works. Unlikewise, Skeoch [7] conducted a more extensive investigation of the problem. The author evaluated several dissimilarity measurements and color models for compatibility evaluation, and also applied Genetic Algorithm [8] to obtain a final reconstruction. The study concluded that color comparison along strips boundaries makes feasible the reconstruction of documents rich in color, which is not the case with text documents that are nearly black-and-white.

The works in [9]–[20] address the reconstruction of text documents with compatibility evaluation designed specifically to this domain. As the focus was in optimization techniques, evaluation was performed only on simulated-shredded instances without accounting for noise, which does not provide insights about the performance on real scenarios. Lin and Fan-Chiang [21] used a single real-shredded document (i.e., obtained mechanically with a paper shredder) for tests, and were able to reconstruct nearly half of the document. Marques and Freitas [22] produced a collection of 200 strip-shredded test instances, which includes 60 text documents. Their method explores color dissimilarities, and, likewise Skeoch’s investigation, the poorest performance was reported for plain text documents.

Xing and Zhang [23] proposed a machine-learning approach for reconstruction of mechanically-shredded documents. They used a local dataset of 50 documents, where 40 are used to collect character features statistics (training stage), and 10 were used for testing. In spite of promising results, their method is restricted to documents in the Chinese language whose characters have a particular shape structure. Andaló *et al.* [24] address the text document reconstruction as a particular instance of the 2-D puzzle solving with regular shapes. Although both problems are strongly correlated, the

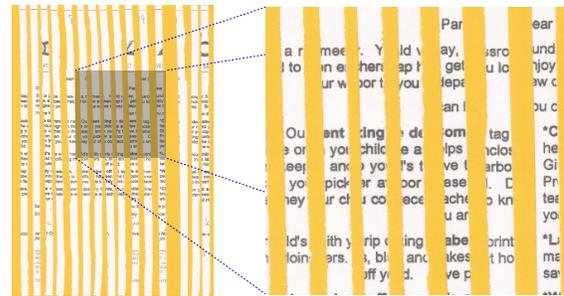


Fig. 4. Document sample of our local dataset. Note that strips can be curved, and how the borders are trimmed.

regularity assumption for shredded documents is not verified in real situations, as illustrated in Figure 4.

In summary, literature has focused mostly on the optimization part of the pipeline, and most of the experimental results has been reported for artificial data. In this work, we address the robust reconstruction of real strip-shredded text documents in a deep learning-based approach. To this purpose, fully-convolutional neural networks are leveraged to estimate how fitting (compatible) two strips are. Training data is fully automatically extracted and labeled from an OCR documents benchmark, which requires no mechanical shredding of documents to generate the extensive training data. Our full reconstruction system, i.e., the compatibility score coupled to an optimization tool, achieved, on average, 94.58% of accuracy in experiments with real-shredded data. Following the same optimization procedure, the second-best performing compatibility score yielded 73.77% of average accuracy. Two shredded-documents datasets were used in the experiments, being one of them produced in the context of our research and made freely available to the scientific community.

## II. PROPOSED SYSTEM

The proposed system is divided into two major pipelines, as illustrated in Figure 3: training of the deep learning model

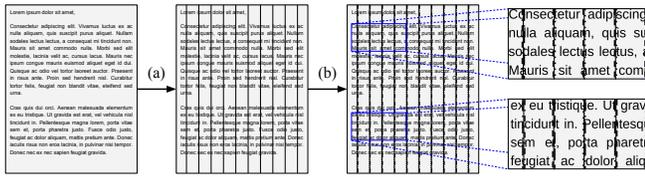


Fig. 5. Artificial Shredding. The document is artificially shredded based on the strip-cut (a) and random noise is added (b) to the edges to simulate the noise usually found in digitized mechanically-shredded documents.

and reconstruction of documents. The first pipeline (top flow) comprises the generation of training data from artificially-shredded documents followed by the training of the network, where the best model is chosen using a validation set. The reconstruction pipeline (bottom flow) assumes that the document to be reconstructed (test document) has been previously shredded, scanned, and segmented before the beginning of its operation. The best model obtained in the training pipeline is used to compute the compatibility score between every two strips. The resulting score values, arranged as a square matrix, are the inputs for the optimization procedure that estimates a solution (permutation of strips) corresponding to the reconstructed document. The details of the training and reconstruction pipelines are presented below alongside a more in-depth description of the system.

#### A. Learning from Artificially-Shredded Documents

A key step in the reconstruction process is the computation of a compatibility score for every pair of strips, where such value quantifies the likelihood of that pair to be a valid one. In the proposed system, a convolutional neural network is trained on artificially-produced data to estimate the compatibility of pairs of real-shredded strips. The training procedure of the neural model comprises three main steps. First, each document of the database undergoes a simple simulated shredding process (overview in Figure 5). After that, a samples generation process takes place. Finally, a convolutional neural network is trained using these samples. Each step is detailed below.

*a) Artificial Shredding:* This process consists in slicing an input digital image in 30 rectangular regions with same width and height (the same as the input image), being this quantity an approximation of the number of strips produced by usual shredders for A4 paper sheets. These simulated strips, however, present clean edges, which is very unlikely to happen in real-world shredding. To cope with that, in the second part, a noise  $\gamma \sim U(0, 255)$  is added to the edges of the strips (2 pixels in each direction) and the resulting images are clipped to avoid overflow. An overview of the process is depicted in Figure 5, where the effect of the noise addition can be seen as well. This shredding procedure is applied over a set of 780 (of a total of 800) images comprising reports, business letters, and legal documents in binary format from the well-known ISRI-Tk OCR database [25]. The 20 documents left out were used to produce a test dataset, as further explained in Section III. The use of simulated-produced data in training

is first justified by the lack of freely available databases containing real-world textual shredded documents. Second, the generation of such kind of database is tedious (error-prone) and highly demanding (requires printing, using a machine to shred the documents, manually organizing and digitalizing the strips, and post-processing them).

*b) Samples Generation:* The input of this step is a document-wise set of strips and the outputs are samples to be used for the model training. This step is applied to every pair of strips in the database in order to generate true (positive) and fake (negative) samples, i.e., samples generated from a valid adjacent pair of strips and others generated from a non-adjacent pair of strips. As the strips were generated by the system in the previous step, the samples will be automatically annotated based on which strips were side-by-side in the original document. Positive and negative samples were generated using the same procedure: given a pair of strips, a sample is represented by a rectangular region of  $31 \times 31$  pixels, i.e., 31 rows of the 15 rightmost pixels of the left strip and 31 rows of the 16 leftmost pixels of the right strip. The strips were sampled every 2 pixels along the vertical axis. A limit of 1,000 positive samples was set per document, and negative samples are limited to the number of positive samples collected on the same document to produce balanced sets. In practice, this means that both positive and negative sets will be balanced per document, as well as for the entire collection. Before extraction, positive and negative strips pairs are firstly shuffled to ensure sampling in different regions of the document since the number of samples per document is limited. The sample dimensions ( $31 \times 31$ ) were chosen because of three reasons: (i) sample diversity, (ii) locality, and (iii) performance. Using small samples (i.e., using  $31 \times 31$  instead of 31 by the height of the strip) allows for more diverse patterns, which means greater coverage of the sample space. In addition, small samples capture local features instead of global structure, which results in a more generic classifier (less dependent on the document structure). At last, small samples imply a lower training time. This generation procedure is repeated for every artificially-shredded document obtained in the previous step, one document at a time, generating equally-sized sets of positive and negative samples.

*c) Model Training:* At this point, two balanced sets (positive and negative) of  $31 \times 31$  samples are available for training the deep learning model. Two fully-convolutional architectures were considered in our system: SqueezeNet [26] and MobileNetV2 [27]. These architectures were chosen mainly because of three factors: (i) they are well-known architectures; (ii) there are efficient architectures, i.e., can achieve good performance with a small model size; and (iii) they are fully convolutional, a feature that is particularly interesting for the proposed system during the inference time, as discussed in the next subsection. In this work, both architectures were initialized with weights pre-trained on the ImageNet [28] using models publicly available. There are different pre-trained models available for the networks used, and the pre-trained models with input sizes  $227 \times 227$  and  $224 \times 224$  were

chosen for the SqueezeNet and MobileNetV2 respectively. Since the models are fully convolutional, the input size could be later reduced to fit the current training samples size without modifications in the architecture. To enable the initialization of the first layer with such weights, the black-and-white (single-channel) samples were replicated on the three channels of the input layer. Another modification was performed in the end of the networks. The order of the global average pooling and the last convolutional layer in the end of the MobileNetV2 was changed to match the order of the SqueezeNet v1.1. In addition, the last convolutional layer of both architectures was changed to have two filters (positive and negative) instead of the original 1,000 filters (ImageNet number of classes), and its weights were initialized using a Gaussian distribution with  $\mu = 0$  and  $\sigma = 0.01$ .

The trained model classifies whether a sample comes from a valid adjacent pair of strips (i.e., positive) or from a non-adjacent one (i.e., negative). With the architecture properly modified for the problem and the weights initialized, the training can begin. The database was split into training and validation sets at document level, i.e., all samples extracted from the same document are used exclusively either in training or to validate the model. A total of 702 documents (90%) are used in training, while 78 (10%) were set apart for validation. The model was trained using the Adam optimizer [30] ( $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ ) to minimize the categorical cross-entropy. Moreover, the model was trained during five epochs using the training set, and the best model was chosen based on the performance (accuracy) of the model on the validation set at the end of every epoch. Finally, the trained (best) model is ready to be deployed.

## B. Reconstruction of Real Documents

The proposed system assumes the strips of the document to be reconstructed were previously digitalized and individualized in image files at disk. After loading, strips are binarized by applying the Otsu thresholding technique [31] since images used to generate samples are black-and-white. Next, blank strips (i.e., those without black pixels) are discarded since they do not contain useful information for the reconstruction process, as discussed in [10]. Then, the system leverages the trained neural model to compute the pairwise compatibility score for the remaining (non-blank) strips. Based on these compatibilities, an optimization solver outputs a permutation that represents an estimated reconstruction for the input strips. These steps are detailed below.

*a) Pairwise Compatibility Scoring:* Given a set of non-blank strips  $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$  belonging to a single document, an ordered pair  $p_{ij} = (s_i, s_j)$  denotes the concatenation of a strip  $s_i$  to the left of  $s_j$ . The goal of this stage is to estimate a compatibility value for every ordered pair  $p_{ij}, i, j = 1, \dots, N, i \neq j$ , which can be arranged in a square matrix  $\mathbf{C}_{N \times N}$  where each entry  $(i, j)$  corresponds to the compatibility value computed to the ordered pair  $p_{ij}$ . In other words,  $\mathbf{C}_{ij}$  quantifies how likely  $s_j$  is the right neighbor of  $s_i$  in the original document. To estimate  $\mathbf{C}_{ij}$ , interest

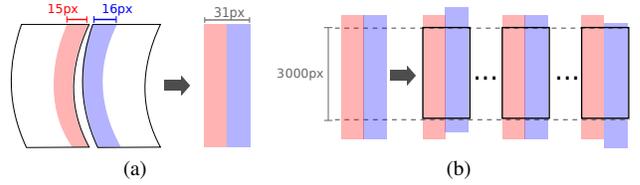


Fig. 6. Association images extraction: (a) rectified interest regions; (b) network visual field definition by cropping and sliding operations.

regions around the edges of  $s_i$  and  $s_j$  are first extracted, as illustrated in Figure 6a. The 15 rightmost pixels of  $s_i$  are joined (at left) with the 16 leftmost pixels of  $s_j$ , giving rise to a  $H \times 31$  association image, where  $H$  is the minimum height of both strips. To account for vertical misalignment, a total of 21 images are obtained from the association image by sliding its right part (blue area) over the left part. It was considered vertical offsets of  $-10, -9, \dots, 9, 10$  pixels, and a visual field of  $3000 \times 31$  pixels centered in relation to the left part of the association image (Figure 6b). The deployed neural network is then fed with 21-size batch containing this slides images, resulting in 21 pairs of positive/negative softmax probabilities.  $\mathbf{C}_{ij}$  is finally set to the maximum of those positive probabilities, which also defines automatically the  $p_{ij}$  vertical displacement. Note that network inference in this stage differs from training since now the input visual field is 3000 pixels height. This is only possible because the adopted architectures are fully convolutional, i.e., they have no fully connected layers, which are usually at the end of classification networks.

*b) Permutation Estimation:* The final reconstruction can be represented by a sequence  $s_{\pi_1} s_{\pi_2} \dots s_{\pi_N}$  of elements in the strips set  $\mathcal{S}$ , where  $\pi = \pi_1 \pi_2 \dots \pi_N$  is a permutation of  $\{1, 2, \dots, N\}$ . In this stage, a permutation  $\pi$  is estimated from the previously computed compatibilities. To solve this, we adopted a graph-based optimization model based on Asymmetric Traveling Salesman Problem (ATSP) closely related to the works in [9], [10]. Since ATSP is a minimization problem, a distance (cost) matrix  $\mathbf{D}_{N \times N}$  is first derived from the compatibility matrix  $\mathbf{C}$  by setting  $\mathbf{D}_{ij} = \max(\mathbf{C}) - \mathbf{C}_{ij}$  for  $i, j = 1, 2, \dots, N, i \neq j$ , and  $+\infty$  for the diagonal elements of  $\mathbf{D}$ . The distance matrix can be viewed as a complete directed weighted graph  $G = (V = \{v_1, \dots, v_N\}, A, w)$ , where a vertex  $v_i \in V$  maps to a strip  $s_i \in \mathcal{S}$ ,  $A \in V^2$  is the set of arcs, and  $w : A \rightarrow \mathbb{R}$  is the weight function defined so that  $w((v_i, v_j)) = \mathbf{D}_{ij}$ . A solution for the reconstruction problem is straightly derived from the shortest Hamiltonian path of  $G$ . As the ATSP formulation intends to find a shortest cycle, we derive a new complete graph  $G'$  from  $G$  by adding a “dummy” vertex  $v'$  connected to all other vertices with zero weight arcs. Thus, if a cycle  $v_{\pi_1} v_{\pi_2} \dots v_{\pi_{k-1}} v_{\pi_k} v_{\pi_{k+1}} \dots v_{\pi_N}$ , with  $v' = v_{\pi_k}$ , is a solution of the ATSP problem for  $G'$ , it can be concluded that  $s_{\pi_{k+1}} \dots s_{\pi_N} s_{\pi_1} \dots s_{\pi_{k-1}}$  is the solution of the reconstruction problem. Finally, ATSP is solved directly by running the black-box LocalSolver optimization tool [32].

### III. EXPERIMENTAL METHODOLOGY

The experiments were conducted in two mechanically-shredded datasets<sup>2</sup>: a collection provided by the authors in [22] (D1) and a local dataset (D2) produced from ISRI-Tk OCR database [25]. The goal of the experiments is to evaluate the performance of both the compatibility scores and also the full reconstruction pipeline. This section describes the datasets, the conducted experiments, and the software/hardware platform used in the evaluation process.

#### A. Datasets

The first dataset (D1) was produced and provided by Marques and Freitas [22]. The entire collection comprises 200 mechanically-shredded documents, however we used only those the authors labeled as text documents, which includes 60 samples. To generate the dataset, Marques and Freitas used a Cadence FRG712 strip-cut paper-shredder, being the produced strips scanned at 300 dpi. The strips were made available already segmented, each one corresponding to a single JPEG file.

The second dataset (D2) was produced to meet the lack of public benchmarks for the specific type of documents addressed in this work. It was assembled from 20 text documents comprising business letters and legal documents from ISRI-Tk OCR database [25]. These documents were originally scanned at 300 dpi using 24 bits format (despite the graylevel appearance). Then, we print them out using a multi-functional HP Deskjet 1510 printer. The printed documents were undergone to a Leadership 7348 strip-cut shredder, and the resulting strips were spliced onto a thick yellow paper sheet, as depicted in Figure 4. The yellow color was chosen to help the strips semi-automatic segmentation step to be performed after the scanning process (details of the segmentation procedure are out of the scope of this work). The paper sheets containing the strips were scanned at 300 dpi (24-bits depth) using the scanner embedded in the multi-functional printer kit. Even both datasets were produced from real paper shredders, it was verified D1 presents more regular strips. The D2 documents frequently depict curved shape, and the original content around the borders are more corrupted.

#### B. Experiments

The intended goals of the evaluation process are to assess (i) the quality of the compatibility score and (ii) the performance of reconstructions delivered by the proposed system.

*a) Performance of the Compatibility Scoring:* Compatibility scoring leverages the deep learning models trained on a simulated shredding scenario for posterior evaluation on mechanically-shredded documents. To assess the coherence of the assigned compatibility values, the pairwise compatibility matrix was analyzed independently of the subsequent solving step. Given the compatibility matrix  $\mathbf{C}$ , and assuming that, for simplicity of mathematical description, the correct sequence of

strips is given by  $s_1 s_2 \dots s_N$ , the quality of a compatibility score ( $Q_C$ ) can be computed by

$$Q_C = \frac{1}{N-1} \sum_{i=1}^{N-1} F(i, i+1), \quad (1)$$

where  $F(i, j)$ , formalized in Equation 2, is a boolean-valued function representing whether the compatibility score assigned to the pair  $(s_i, s_j)$  is the highest in the  $i$ -th row and  $j$ -th column. The quality measure  $Q_C$  lies in the interval  $[0, 1]$ , and the higher its value, the better the compatibility scoring is.

$$F(i, j) = \underbrace{\left( C_{ij} = \max_{j' \in \mathcal{N} \setminus \{i\}} C_{ij'} \right)}_{\text{column-wise verification}} \wedge \underbrace{\left( C_{ij} = \max_{i' \in \mathcal{N} \setminus \{j\}} C_{i'j} \right)}_{\text{row-wise verification}}, \quad (2)$$

$$\mathcal{N} = \{1, 2, \dots, N\}.$$

*b) Performance of the Reconstruction Pipeline:* As a rule, the compatibility scoring for real-shredded documents is not perfect (i.e.,  $Q_C < 1$ ). Therefore, we also evaluate the performance of the reconstruction pipeline as a whole, where both the compatibility scoring and solver work together to find the best permutation. To this purpose, the neighbor comparison measure defined in [24] is used to quantify the accuracy of a permutation of strips. Assume, for simplicity, that  $s_1 s_2 \dots s_N$  is the ground-truth solution, i.e., the perfect reconstruction for a given document instance. The accuracy of an arbitrary solution induced by the permutation  $\pi = \pi_1 \pi_2 \dots \pi_N$  is given by

$$Acc_\pi = \frac{1}{N-1} \sum_i^{N-1} (\pi_i + 1 = \pi_{i+1}). \quad (3)$$

It can be proved that  $Acc_\pi \geq Q_C - \frac{1}{N-1}$  for reconstructions obtained with the simple Kruskal-based heuristic [35], an algorithm that builds solutions by greedily connecting pairs of strips at each step.

*c) Comparative Analysis:* For the purpose of comparison, both evaluations (compatibility score and reconstruction pipeline) were also conducted for other reconstruction methods in literature: Marques et al. [22], Morandell et al. [9], Andaló et al. [24], Balme et al. [36], and Sleit et al. [14]. Since this work focuses on the design of a compatibility scoring procedure, the comparison at final reconstruction level is conducted using our ATSP-based solver. Besides, the accuracy of the Kruskal-based reconstruction [35] is also reported as a baseline performance. Two comparison approaches were considered to compare the reconstruction methods accuracy: realistic (fair) and unrealistic (unfair). In the realistic approach, the compared methods were configured just as recommended by their authors. However, since such methods are very sensitive to paper damage caused by mechanical shredding, we conducted an additional accuracy evaluation discarding  $d$  boundary pixels in each strip image row, following the idea in [22]. The idea is to

<sup>2</sup>The public available datasets in [33], [34] were not used since they do not fit to the class of documents addressed in this paper (i.e., machine printed text without meaningful color information).

compute a pool of 11 solutions for each document instance by setting  $d = 0, 1, \dots, 10$ , and then to report only the maximum accuracy one. This additional comparison is said to be unfair since the best solution is chosen based on the ground-truth solution, and a reconstruction system, by definition, is blind to this information. With exception of Andaló et al. [24], which provided their implementation, all the compared methods were reimplemented to enable the comparative analysis.

### C. Experimental Platform

The experiments were carried out on two different machines. First, the training (including dataset generation) was performed in an Microsoft Azure NC12 instance with 12 vCPU, 112GB RAM, and 2 GPUs (12GB each). Testing and analysis were performed in a 3.10 GHz PC with 4GB of RAM and a GTX 960 (4GB). The proposed system was developed in Python, using the TensorFlow framework for training and inference of the deep learning model, and the OpenCV library for image processing. The code, pre-trained models, and database are publicly available at <https://github.com/thiagopx/deeprec-sib18>.

## IV. RESULTS AND DISCUSSION

This section discusses the obtained results for the experiments described in the previous section. First, we evaluate solely the quality of compatibility scoring procedures for all the six aforementioned (including ours) methods. Posteriorly, the full reconstruction induced by these scoring approaches are evaluated based on the accuracy measure defined in Equation 3.

### A. Performance of the Compatibility Scoring

The performance of the compatibility scoring techniques is reported in Table I. The table columns show average and standard deviation values for  $Q_C$  measure (Equation 1) considering the entire documents collection ( $D1 \cup D2$ ), and for D1 and D2 in separate. Our compatibility score based on the SqueezeNet (**Proposed-SN**) achieved the highest average accuracy with the lowest relative standard deviation for both datasets, followed by the MobileNet-based score (**Proposed-MN**). Note that this quality measure serves as lower bound for the reconstruction problem. This means that even the naive Kruskal-based solver associated to our compatibility score is able to reach average accuracies around 92.13% and 87.29% for D1 and D2, respectively. However, despite the indicative of worse performance for the compared methods, no ultimate conclusion can be made without analyzing the full reconstruction results.

### B. Performance of the Reconstruction Pipeline

The performance of our full reconstruction system is depicted in Figure 7. In summary, the best performance was achieved by the proposed system with SqueezeNet and LocalSolver (LS) for both D1 and D2 datasets. Note, for this configuration, that the Kruskal-based heuristic (KBH) performs similarly to LS for D1 (less than 1% of difference on average),

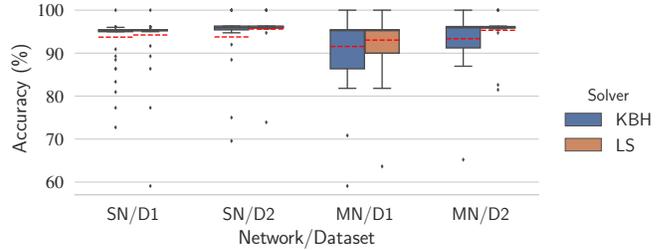


Fig. 7. Performance of the proposed system (median boxplots) for the datasets D1 and D2 measured in terms of accuracy (Equation 3). KBH and LS denote, respectively, the Kruskal-based reconstruction algorithm and the LocalSolver optimization tool. The red dashed lines represent the average accuracies.

which can be explained by the high  $Q_C$  lower bound (92.13%). For LS, five of the total of 80 documents were perfectly reconstructed. This low number (when compared to the overall accuracy of our system) is correlated to the fact that 72 of the 80 obtained reconstructions have the last strip merged to the left of the first strip (first and last are in relation to the ground-truth). Moreover, this kind of mistake is the only one in 69 of these 72 cases, which means that fixing this mistake would lead to 74 perfect reconstructions. This phenomenon is common in literature, and can be justified in our neural model context by the positive bias for white-to-white associations, which frequently happens when pairing the right border of the last strip with the left border of the first. A possible way to deal with this is to include neutral (white) training samples for both positive and negative to reduce the bias factor. Figure 8 shows two reconstructions delivered by our system configured with LocalSolver.

For the sake of comparison, Table II presents the comparative performance with methods set to their default configuration (fair comparison). Both Kruskal-based heuristic (KBH) and LocalSolver (LS) average accuracies, alongside the respective standard deviation were reported. The highlighted rows exhibit our system performance. It can be noticed that our approach induced better solutions with lower relative standard deviation than the compared methods for both datasets. In addition, our system performs very similarly (and with high accuracy) for KBH and LS, which is a strong quality evidence of the compatibility scoring procedure. The same, in general, is not verified for the compared methods, which means that they require more effort at the optimization side of the pipeline. It

TABLE I  
PERFORMANCE OF THE COMPATIBILITY SCORING METHODS:  $Q_C \pm \sigma$  (%).

Method	D1 $\cup$ D2	D1	D2
<b>Proposed-SN</b>	<b>90.92 <math>\pm</math> 15.55</b>	<b>92.13 <math>\pm</math> 15.75</b>	<b>87.29 <math>\pm</math> 14.72</b>
<b>Proposed-MN</b>	<b>79.58 <math>\pm</math> 18.89</b>	<b>80.53 <math>\pm</math> 19.84</b>	<b>76.72 <math>\pm</math> 15.81</b>
Andaló	41.36 $\pm$ 26.43	42.50 $\pm$ 28.39	37.94 $\pm$ 19.62
Morandell	35.84 $\pm$ 25.31	42.91 $\pm$ 24.67	14.63 $\pm$ 11.88
Balme	18.84 $\pm$ 29.06	24.34 $\pm$ 31.66	2.33 $\pm$ 3.92
Sleit	14.54 $\pm$ 9.06	15.81 $\pm$ 9.70	10.75 $\pm$ 5.37
Marques	12.71 $\pm$ 19.14	12.16 $\pm$ 19.02	14.38 $\pm$ 19.90

TABLE II  
FULL RECONSTRUCTION PERFORMANCE (ORIGINAL COMPATIBILITY METHODS + OUR ATSP-BASED SOLVER):  $Acc_{\pi} \pm \sigma$  (%).

Method	D1 $\cup$ D2		D1		D2	
	KBH	LS	KBH	LS	KBH	LS
<b>Proposed-SN</b>	<b>93.74 <math>\pm</math> 5.63</b>	<b>94.58 <math>\pm</math> 5.40</b>	<b>93.72 <math>\pm</math> 4.73</b>	<b>94.23 <math>\pm</math> 5.40</b>	<b>93.77 <math>\pm</math> 7.88</b>	<b>95.62 <math>\pm</math> 5.39</b>
<b>Proposed-MN</b>	<b>92.01 <math>\pm</math> 7.07</b>	<b>93.61 <math>\pm</math> 5.58</b>	<b>91.56 <math>\pm</math> 6.81</b>	<b>93.04 <math>\pm</math> 5.72</b>	<b>93.37 <math>\pm</math> 7.81</b>	<b>95.32 <math>\pm</math> 4.85</b>
Marques	56.46 $\pm$ 23.72	73.77 $\pm$ 19.76	55.46 $\pm$ 23.26	72.06 $\pm$ 20.27	59.45 $\pm$ 25.43	78.91 $\pm$ 17.63
Morandell	55.32 $\pm$ 22.79	65.94 $\pm$ 22.22	58.60 $\pm$ 23.31	68.93 $\pm$ 21.63	45.49 $\pm$ 18.36	56.96 $\pm$ 22.05
Andaló	71.17 $\pm$ 20.31	58.29 $\pm$ 24.14	72.75 $\pm$ 19.22	58.25 $\pm$ 24.91	66.44 $\pm$ 23.17	58.41 $\pm$ 22.26
Balme	48.23 $\pm$ 22.95	59.93 $\pm$ 22.47	53.39 $\pm$ 23.48	64.38 $\pm$ 22.12	32.75 $\pm$ 11.92	46.57 $\pm$ 18.13
Sleit	26.57 $\pm$ 11.68	23.04 $\pm$ 9.17	28.02 $\pm$ 12.58	24.46 $\pm$ 9.48	22.23 $\pm$ 7.00	18.76 $\pm$ 6.68

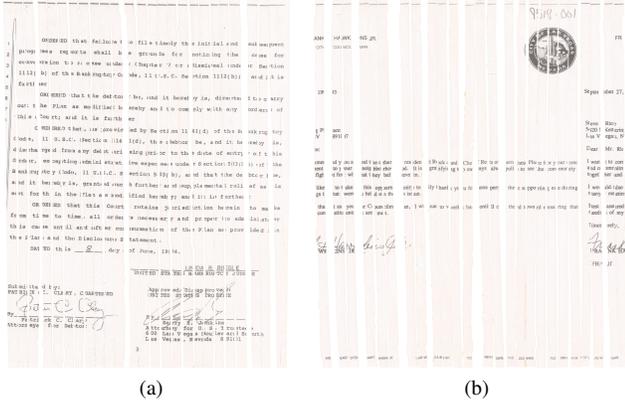


Fig. 8. Reconstructions obtained with our system configured with LocalSolver: (a) A perfect reconstruction; (b) A reconstruction case with accuracy of 73.91%.

is also important to stress that lower bounds depicted in Table I are too restrictive, and may not indicate precisely the ability of a compatibility score in inducing good solutions. Note that Marques reached 78.91% of accuracy for D2 (Table II, column LS), while the quality value was 14.38% (Table I).

Since the compared methods are known to be sensitive to the damage imposed on the strips borders, we also report results considering the removal of  $d$  border pixels (unfair comparison, Section III). Figure 9 depicts the LocalSolver performance for our system against the compared methods in (a) their default configuration, and (b) optimally adjusted in relation to  $d$ . This result shows clearly that the performance of the compared methods can be drastically changed depending on what pixel border is considered for calculations. Despite remarkable improvements for both datasets, their performances are still worse than that achieved by our reconstruction system. When compared to Marques, the best-performed method among those used for comparison, our system yields accuracies +22.17 and +16.71% superior for D1 and D2, respectively. Note that, although the training data was extracted from documents more similar to D2, our deep learning approach achieved better results for both D1 and D2 datasets.

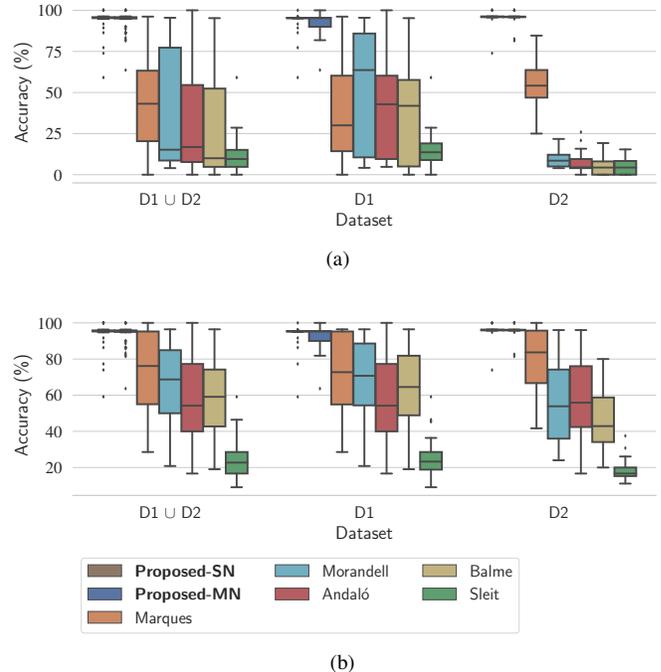


Fig. 9. Performance comparison obtained with LocalSolver: (a) Default methods configuration (fair comparison); (b) Optimal selection of  $d$  parameter for the compared methods (unfair comparison).

## V. CONCLUSION

In a typical document reconstruction framework, the most critical step for obtaining accurate solutions is to properly estimate the pairwise compatibility between paper fragments. This is even more challenging for text documents since the absence of color hinders the matching of fragments. In this paper, we proposed a novel deep learning-based compatibility model to be applied in the reconstruction of strip-shredded text documents. To circumvent the lack of (real) mechanically-shredded datasets, we exploited the fact that text documents are locally similar to train the model from small image patches, which are extracted from simulated-shredded documents, instead of using the entire strip. Experiments were performed to assess the quality of the proposed compatibility scoring approach (using deep learning models) in isolation, and also in conjunction with the optimization step to provide the performance of the reconstruction as a whole. Moreover,

five other compatibility scoring methods of the literature were used for comparative evaluation. The reconstruction with our deep learning approach (considering the best model) achieved 94.23% and 95.62% for the dataset in [22] and for a local dataset, respectively, outperforming by a great margin the compared methods (even considering their best performance in relation to a specific parameter). Future work should investigate the effectiveness of this method when reconstructing several documents at a time.

#### ACKNOWLEDGMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Cloud computing resources were provided by a Microsoft Azure for Research award. In addition, we would like to acknowledge the scholarships of Productivity on Research (grants 311120/2016-4 and 311504/2017-5) supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil); and, a scholarship supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Brazil).

#### REFERENCES

- [1] A. A. Low, "Waste-paper receptacle," U.S. Patent 929960, 1909.
- [2] J. D. Derian, "Arms, Hostages, and the Importance of Shredding in Earnest: Reading the National Security Culture (II)," *Social Text*, no. 22, pp. 79–91, 1989.
- [3] Fraunhofer IPK. Automated Virtual Reconstruction of Ripped Stasi Files. Accessed 2018-06-15. [Online]. Available: <https://www.ipk.fraunhofer.de/en/divisions/automation-technology/departments/security-technology/core-rd-activities-of-the-security-technology-department/virtual-reconstruction/automated-virtual-reconstruction-of-ripped-stasi-files/>
- [4] P. De Smet, J. De Bock, and W. Philips, "Semiautomatic reconstruction of strip-shredded documents," in *Image and Video Communications and Processing*, vol. 5685, 2005, pp. 239–248.
- [5] A. Ukovich, G. Ramponi, H. Doulaverakis, Y. Kompatsiaris, and M. Strintzis, "Shredded document reconstruction using MPEG-7 standard descriptors," in *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology*, 2004, pp. 334–337.
- [6] A. Ukovich and G. Ramponi, "Feature extraction and clustering for the computer-aided reconstruction of strip-cut shredded documents," *Journal of Electronic Imaging*, vol. 17, no. 1, p. 013008, 2008.
- [7] A. Skeoch, "An investigation into automated shredded document reconstruction using heuristic search algorithms," Ph.D. dissertation, University of Bath, UK, 2006, unpublished thesis.
- [8] J. H. Holland, "Genetic algorithms," *Scientific American*, 1992.
- [9] W. Morandell, "Evaluation and reconstruction of strip-shredded text documents," Master's thesis, Vienna University of Technology, 2008.
- [10] M. Prandtstetter and G. R. Raidl, "Combining Forces to Reconstruct Strip Shredded Text Documents," in *Hybrid Metaheuristics*, 2008, pp. 175–189.
- [11] —, "Meta-heuristics for reconstructing cross cut shredded text documents," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 2009, pp. 349–356.
- [12] C. Schauer, M. Prandtstetter, and G. R. Raidl, "A memetic algorithm for reconstructing cross-cut shredded text documents," in *Hybrid Metaheuristics*, 2010, pp. 103–117.
- [13] B. Biesinger, C. Schauer, B. Hu, and G. R. Raidl, "Enhancing a Genetic Algorithm with a Solution Archive to Reconstruct Cross Cut Shredded Text Documents," in *International Conference on Computer Aided Systems Theory*, 2013, pp. 380–387.
- [14] A. Sleit, Y. Massad, and M. Musaddaq, "An alternative clustering approach for reconstructing cross cut shredded text documents," *Telecommunication Systems*, vol. 52, no. 3, pp. 1491–1501, 2013.
- [15] P. Li, X. Fang, L. Pan, Y. Piao, and M. Jiao, "Reconstruction of Shredded Paper Documents by Feature Matching," *Mathematical Problems in Engineering*, vol. 2014, no. Article ID 514748, 9 pages, 2014.
- [16] Y. Wang and D.-C. Ji, "A Two-Stage Approach for Reconstruction of Cross-Cut Shredded Text Documents," in *Tenth International Conference on Computational Intelligence and Security (CIS)*, 2014, pp. 12–16.
- [17] H. Xu, J. Zheng, Z. Zhuang, and S. Fan, "A Solution to Reconstruct Cross-Cut Shredded Text Documents Based on Character Recognition and Genetic Algorithm," in *Abstract and Applied Analysis*, vol. 2014, no. Article ID 829602, 11 pages, 2014.
- [18] T. Phienthrakul, T. Santitewagun, and N. Hnoohom, "A Linear Scoring Algorithm for Shredded Paper Reconstruction," in *11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2015, pp. 623–627.
- [19] S. Guo, S. Lao, J. Guo, and H. Xiang, "A semi-automatic solution archive for cross-cut shredded text documents reconstruction," in *International Conference on Image and Graphics*, 2015, pp. 447–461.
- [20] Y.-J. Gong, Y.-F. Ge, J.-J. Li, J. Zhang, and W. Ip, "A splicing-driven memetic algorithm for reconstructing cross-cut shredded text documents," *Applied Soft Computing*, vol. 45, pp. 163–172, 2016.
- [21] H.-Y. Lin and W.-C. Fan-Chiang, "Reconstruction of shredded document based on image feature matching," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3324–3332, 2012.
- [22] M. Marques and C. Freitas, "Document decipherment-restoration: Strip-shredded document reconstruction based on color," *IEEE Latin America Transactions*, vol. 11, no. 6, pp. 1359–1365, 2013.
- [23] N. Xing and J. Zhang, "Graphical-character-based shredded Chinese document reconstruction," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12871–12891, 2017.
- [24] F. A. Andalo, G. Taubin, and S. Goldenstein, "PSQP: Puzzle solving by quadratic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 385–396, 2017.
- [25] T. A. Nartker, S. V. Rice, and S. E. Lumos, "Software tools and test data for research and testing of page-reading OCR systems," in *Electronic Imaging 2005*, 2005, pp. 37–47.
- [26] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.
- [27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
- [29] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference for Learning Representations (ICLR)*, 2015.
- [31] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [32] T. Benoist, B. Estellon, F. Gardi, R. Megel, and K. Nouioua, "Local-solver 1. x: a black-box local-search solver for 0-1 programming," *4OR*, vol. 9, no. 3, p. 299, 2011.
- [33] DARPA. (2011) Darpa shredder challenge. Accessed 2017-05-25. [Online]. Available: <http://archive.darpa.mil/shredderchallenge>
- [34] P. Saboia and S. Goldenstein, "Assessing Cross-Cut Shredded Document Assembly," in *Iberoamerican Congress on Pattern Recognition*, 2014, pp. 272–279.
- [35] R. Ranca and I. Murray, "A Composable Strategy for Shredded Document Reconstruction," in *International Conference on Computer Analysis of Images and Patterns*, 2013, pp. 324–331.
- [36] J. Balme, "Reconstruction of shredded documents in the absence of shape information," Working paper, Dept. of Computer Science, Yale University, USA, Tech. Rep., 2007.