Deep Discriminative Feature Learning for Document Image Manipulation Detection

Kais Rouis[1], Petra Gomez-Krämer[1], Mickaël Coustaty[1],
Saddock Kébairi[2] and Vincent Poulain d'Andecy[2]
[1]*L3i Laboratory, La Rochelle University, La Rochelle, France*
[2]*R&D Department, Yooz, Aimargues, France*

Keywords: Document Forgery, Manipulation Detection, Deep Neural Model, Residual Network.

Abstract: Image authenticity analysis has become a very important task in the last years with one main objective that is tracing the counterfeit content induced by illegal manipulations and forgeries that can be easily practiced using available software tools. In this paper, we propose a reliable residual-based deep neural network that is able to detect document image manipulations and copy-paste forgeries. We consider the perceptual characteristics of documents including mainly textual regions with homogeneous backgrounds. To capture abstract features, we introduce a shallow architecture using residual blocks and take advantage of shortcut connections. A first layer is implemented to boost the model performance, which is initialized with high-pass filters to forward low-level error feature maps. Manipulation experiments are conducted on a publicly available document dataset. We compare our method with two interesting forensic approaches that incorporate deep neural models along with first layer initialization techniques. We carry out further experiments to handle the forgery detection problem on private administrative document datasets. The experimental results demonstrate the superior performance of our model to detect image manipulations and copy-paste forgeries in a realistic document fraud scenario.

1 INTRODUCTION

Today lots of documents are used and processed in document flows in companies, banks and administrations. These documents can be original digital documents or scans of printed documents. The manipulation detection of these documents has become an important concern as the use of fraudulent documents can lead, for instance, to false identity documents, identity theft or fraudulently obtained credits. Several types of fraudulent manipulations exist. The authors of (Cruz et al., 2018) report four tampering operations in document images: the imitation of font characteristics to insert text into the document image, the copy and paste of a region from the same document image, the copy and paste of a region from another document image, and the deletion of information. Furthermore, the authors of (James et al., 2020) report additionally the tampering by pixel manipulations. This means that pixel values are modified to change the visual aspect of a character, e.g. to change the character "c" into the character "o".

Document images are qualified by their binary character, and strong contrasts and contours as well as poor textures compared to natural scene images (Gomez-Krämer, 2022). When dealing with structured data such as documents, featuring visual objects with similar salient shapes and homogeneous backgrounds, we may encounter learning issues along overly deep architectures. This is due to the *poor* enclosed information in terms of textures and pixel intensity variations. Therefore, outliers corresponding to manipulated image regions illustrate for most forgery operations similar perceptual properties to the original samples. The document image domain relates indeed to particular requirements compared to natural image statistics.

Taking into account the aforementioned statement, understanding the behaviour of deep neural models is a key step to represent different levels of forensic features. In (Bayar and Stamm, 2018), a constrained convolutional neural network (CNN) architecture was designed to adaptively learn image manipulation features, and to determine the type of ungenuine image editing. The model allows to learn low-level prediction residual features, using a constrained layer placed at the top of the proposed network, while deeper layers learn high-level manipulation features and two fully connected layers of 200 neurons classify the output features. In a similar vein, an interesting CNN first layer initialization method (Castillo Camacho and Wang, 2022) was introduced

Rouis, K., Gomez-Krämer, P., Coustaty, M., Kébairi, S. and Poulain d'Andecy, V.
Deep Discriminative Feature Learning for Document Image Manipulation Detection.
DOI: 10.5220/0012410800003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 567-574
ISBN: 978-989-758-679-8; ISSN: 2184-4321

based on statistical properties of the training data. The aim is to properly scale the used high-pass filters as convolutional kernels. The authors studied the impact of variance stability of the first-layer output features on the detection accuracy. The used network is a less deeper version of (Bayar and Stamm, 2018). Experiments were conducted on a benchmarking digital image forensics dataset, where the original uncompressed images have undergone manipulations including filtering, resampling and compression among several parameters.

Furthermore, few forensic approaches have been particularly suggested for document images. For instance, the method in (Nandanwar et al., 2021) explores the DCT coefficients to analyze the effect of tampering distortions. Reconstructed images from the inverse DCT are filtered by Laplacian kernels to point out the difference between textual regions and surrounding pixels. The authors in (Abramova and Böhme, 2016) examine feature representations proposed in the literature with respect to the correct detection of copied text document segments. To model contextual information (Cruz et al., 2017), multiple descriptors of forged regions can be combined based on locally extracted texture features. On the one hand, the performance of such methods strongly depends on the document content characteristics, i.e. processing forgeries with high or low visual distortion levels. On the other hand, the related works proposed for natural images cannot be straightforwardly applied to document images seeing the dissimilar perceptual properties.

In this paper, we propose a new residual-based network architecture to detect document image manipulations and forgeries. First, we demonstrate that our scheme achieves higher performance to classify manipulation operations, in comparison to related methods (Bayar and Stamm, 2018; Castillo Camacho and Wang, 2022). In fact, the application of these methods is quite convenient to our objectives, as we aim to define a robust *shallow* residual CNN model, which is able to detect realistic fraudulent blocks within document regions.

Our deep learning strategy is entirely distinct from preceding solutions which are based on hierarchical stacked convolutional layers. We afford a shallower architecture by means of shortcut connections within residual blocks as basic network units. Afterwards, we conduct experiments on a private administrative document dataset, to prove a consistent hypothesis: if a deep model can efficiently determine applied manipulations, it is possible to train the model using only transformed original samples to predict finally the fraudulent ones. More precisely, we propose to train our model to separate between compressed versions of the inputs with varied quality factors. In this part, each compression level represents a particular class during the training step.

We choose the compression as a non-geometric transformation with regard to copy-paste forgeries. Also, we use only original samples of image blocks without content modification (copy-paste operation) in the learning. The features of engendered compression noise will have an unnatural distribution for fake image content. Consequently, if the model fails to recognize the compression levels of a test sample, then we judge it as a fraud, otherwise it is genuine.

Our approach differs entirely from double compression detection. Here, the compression has a different impact on forged content producing unnatural artefact distributions. Therefore, the model will not be able to predict the right compression ratio (as a class) for such distributions. We do not rely on the use of compressed images particularly, but on the compression as a normal or abnormal pixel alteration.

The rest of the paper is organized as follows. Section 2 reviews related works on document image manipulation detection. In Section 3, we introduce the theoretical background about the shallow deep model principle. Section 4 describes the details of the proposed residual-based architecture. Experiments in Section 5 demonstrate the performance of our method to classify manipulation operations (public document dataset). Copy-paste forgery detection is further investigated on a private document dataset. Finally, some conclusions are drawn in the last section.

2 RELATED WORKS

Documents can be secured using active approaches introducing security elements into the document. These approaches insert an extrinsic fingerprint into the document to check their authenticity afterwards. Typical approaches are watermarks (Brassil et al., 1999; Huang et al., 2019) or digital signatures (Tan and Sun, 2011; Gomez-Krämer et al., 2023). However, the security elements have to be added to the document during its production.

That is why passive approaches have gained in interest. These methods look for intrinsic characteristics in the document that indicate a modification of the document content. Printer identification (Joshi and Khanna, 2020; Choi et al., 2013) aims at identifying the printer that was used to produce the document. However, many documents are transmitted in original digital format or printed by the user itself. Thus, the basic assumption that the printer used to create the document is known is no longer valid.

Lately, document tampering detection methods have appeared, but until now quite few work has been presented for this task. Early methods aim at detecting graphical signs of manipulations such as slope, size and alignment variations of a character with respect to the others (Bertrand et al., 2013), font or spacing variations of characters or in a word (Bertrand et al., 2015), the variation of geometric distortions of characters introduced by the printer (Shang et al., 2015), or the text-line rotation and alignment (Beusekom et al., 2013). These methods only apply to a specific type of manipulation.

The authors of (Ahmed and Shafait, 2014) use distortions in the varying parts of the documents (not the template ones) through a pair-wise document alignment to detect forgery. Hence, the method needs several samples of a class (template). In (Abramova and Böhme, 2016) a block-based method is proposed for copy and move forgery detection based on the detection of similar characters using Hu and Zernike moments, as well as PCA and kernel PCA combined with a background analysis.

The method of (Cruz et al., 2017) is more general. It is based on an analysis of LBP textures to detect discontinuities in the background around characters, residuals of the image tampering. In (Nandanwar et al., 2021) DCT coefficients are used to detect distorsions caused by the tampering of the text. Recently methods based on neural networks have emerged. The method of (Joren et al., 2022) uses a graph neural network with optical character recognition bounding boxes as nodes to detect copied and moved characters. However, the results strongly depend on the quality of the optical character recognition which performs poorly on noisy documents such as printed and scanned documents.

Although significant works have been presented for document tampering detection, the methods lack of generality as they focus on a specific type of manipulation or content. Furthermore, noisy documents such as printed and scanned documents are still a challenge. For this reason, we present in this article a flexible and efficient method for manipulation detection which does not depend on the manipulation or content type.

3 DEEP RESIDUAL-BASED LEARNING

CNN architectures have seen a regular increase of the number of layers in the last few years, looking forward to improve the model performance. As we stack more layers together, training deep models has several risks such as exploding/vanishing gradients and degradation. The ultimate purpose of deep learning lies in the ability to capture abstract features as the signal moves into deeper layers. In a typical CNN architecture, hidden convolutional layers extract features from each lower-level output map. The convolution operation between a convolutional layer and the feature maps is given by:

$$\mathbf{x}_l^k = \sum_{j=1}^{N} \mathbf{x}_{l-1}^j * \mathbf{w}_l^{jk} + b_l^k, \qquad (1)$$

where $\mathbf{x}_l^k$ is the k^{th} output feature map of the l^{th} layer, $\mathbf{x}_{l-1}^j$ is the j^{th} channel of the $(l-1)^{th}$ layer, $\mathbf{w}_l^{jk}$ is the j^{th} channel in the k^{th} filter of the l^{th} layer, and b_l^k is the bias term. We consider the stochastic gradient descent (SGD) solver (Robbins and Monro, 1951) to form parameter updates that attempt to improve the loss. The iterative update rule, given in Eq. 2, is used for kernel coefficients during the backpropagation pass.

$$\mathbf{w}_{l+1}^{jk} = \mathbf{w}_l^{jk} - \overbrace{\left(\gamma\,.\,\frac{\partial \mathcal{E}}{\partial \mathbf{w}_l^{jk}} - \alpha\,.\,\nabla \mathbf{w}_l^{jk} + \lambda\,.\,\gamma\,.\,\mathbf{w}_l^{jk}\right)}^{\nabla \mathbf{w}_{l+1}^{jk}}, \qquad (2)$$

where $\mathcal{E}$ is the average loss between the true class labels and the network outputs. $\nabla \mathbf{w}_l^{jk}$ denotes the gradient of $\mathbf{w}_l^{jk}$ and γ is the learning rate. In the experiments, α and λ are used for fast convergence and correspond to decay and momentum, respectively. One can clearly notice how the optimization process of the overall parameters becomes considerably difficult for a large number of CNN hidden layers.

Useful techniques were suggested to relieve the optimization process including initialization strategies (He et al., 2015) and skip connections (Raiko et al., 2012). In this work, we opt for the concept of residual blocks along with skip connections to remove the learning degradation problem. A residual block incorporates a set of *few* convolutional layers. The non-linearity is applied after adding the output feature map of a layer to another deeper layer in the same block. Nevertheless, the technique of skip connections, alternatively called shortcut connections, consists of skipping *some* of the network layers and feeds the output of the previous layer to the current position (He et al., 2016). Let us consider the input feature maps $\mathbf{x}$ of a residual block and a residual function $\mathcal{R}(\mathbf{x})$. The expected outputs of this block can be defined as the underlying mapping to be fit: $\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x}) + \mathbf{x}$. Hence, assuming that the input and output channels of the residual block are of the same dimensions, the group of considered layers try to learn the new mapping function from the difference (i.e. residual) between inputs and expected outputs: $\mathcal{R}(\mathbf{x}) \doteq \mathcal{M}(\mathbf{x}) - \mathbf{x}$. Basically, $\mathcal{R}(\mathbf{x})$ would have two stacked convolutional layers. For instance, with a shortcut connection from layer l to $l+2$, the activation of layer $l+2$ can be computed as:

$$\begin{aligned} \mathbf{x}_{l+2} &= \mathcal{A}\Big(\mathbf{x}_l + \mathbf{y}_{l+2}\Big), \\ \text{with } \mathbf{y}_{l+2} &= \mathbf{x}_{l+2} * \mathbf{w}_{l+2} + b_{l+2}, \end{aligned} \tag{3}$$

where $\mathcal{A}$ is a ReLU activation function, $\mathbf{x}_l$ (shortcut connection) and $\mathbf{x}_{l+2}$ are respectively the input and output feature maps of the stacked layers within a residual block. If $\mathbf{x}_{l+2}, \mathbf{b}_{l+2} \mapsto 0$, then $\mathbf{x}_{l+2} = \mathcal{A}(\mathbf{x}_l)$. Since we use a ReLU activation, an *identity mapping* (shortcut connection) is created as $\mathbf{x}_{l+2} = \mathbf{x}_l$. Thanks to shortcut connections, the solver can drive easier the weights of the multiple nonlinear layers toward zero to approach identity mappings.

4 PROPOSED METHOD

It is more compelling for forensic applications to expand a conclusive representation of local residual noise distributions. To this end, we let our model learn residual features from a modeled noise, to finally predict manipulations and *unnatural* artefacts.

4.1 First Layer Kernel Initialization

At this step, we build a first convolutional layer, namely the Init-layer, to suppress the content and construct low-level error map features. We just set the Init-layer kernels with predefined high-pass filters to forward output error maps to the following residual blocks in the network. Actually, we do not normalize the initialized kernel weights as proposed in recent related works (Bayar and Stamm, 2018; Castillo Camacho and Wang, 2022). We are essentially interested in using a pre-processing step that will improve relatively the model performance. We implement the layer as *not trainable* to produce high-pass filtered inputs for each batch. We just use a tangent hyperbolic activation function. In our setting, we used 30 filter kernels generated by the Spatial Rich Model (SRM) (Fridrich and Kodovsky, 2012). These filters extract local noise features from adjacent pixels, capturing low-level content dependencies. The noise is modeled as the residual between a pixel (central filter value) and its estimated value by interpolating only neighboring pixels. The kernel size of the Init-layer is fixed to 5×5. Some of the SRM filters were accordingly padded with zeros (initially of size 3×3). We tested two configurations: using all 30 SRM filters and selecting randomly 3 SRM filters (over multiple runs).

4.2 Residual-Based Network Architecture

We introduced in the previous section the residual-based learning background to design a shallow CNN network. Figure 1 illustrates our proposed architecture which consists of three main stages: (1) extraction of low-level error maps (Init-layer), (2) residual feature learning (one convolutional layer + three residual blocks ResBlock$_i$) and (3) classification from the previously learned features.

In Figure 1, ResBlock$_i$ ($i = 1,2,3$) are detailed below the baseline network. The blue and red dashed lines surrounding ResBlock$_i$ represent two different types of shortcut connections (shown as dashed arrows). The first residual block (ResBlock$_1$) is preceded by a convolutional layer with 16 filters of input size $3 \times 3 \times 30$ and a stride of 1. Batch-Normalization (BN) and ReLU activation are applied to each layer input feature map. Their outputs are added to the outputs of two stacked convolutional layers. It is worth to mention that identity mapping does not add extra parameters. The network can still be trained by an SGD solver with backpropagation (end-to-end). The input of ResBlock$_1$ (N$_i$=16) and the second layer in the main path have the same dimensions ($\mathbf{x}_l$ and $\mathbf{y}_{l+2}$ in Eq. 3, respectively). In this case, the mapping is defined by an *identical-shortcut*.

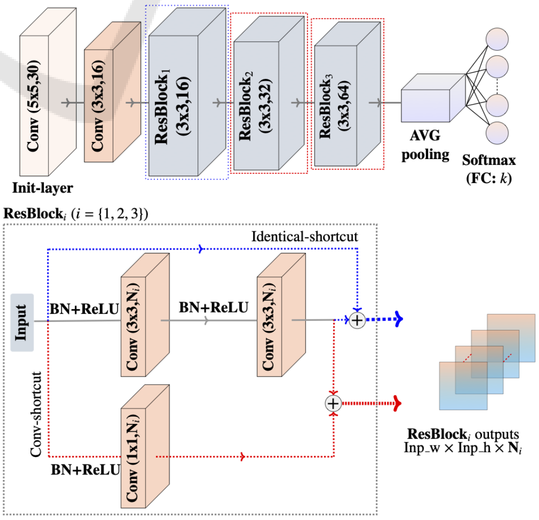


Figure 1: Proposed residual-based network architecture for manipulation detection.

In contrary, the dimensions of the shortcut output (residual block input) and the stacked layers are different for ResBlock$_2$ (N$_i$=32) and ResBlock$_3$ (N$_i$=64). We define here a residual mapping through an additional convolutional layer (*conv-shortcut*) to resize the output of the shortcut path. We set indeed the kernel size of this layer to 1×1 with a stride of 1. The role of this layer is to apply a learned linear function that adjusts the input dimension. The output feature maps of ResBlock$_3$ undergo BN and ReLU operations. We use afterwards an average-pooling layer to reduce the dimensionality of these feature maps. A softmax layer of k elements, corresponding to the number of manipulation class labels, is finally used to convert the subsequent flattened tensor to a probability distribution.

4.3 One-Class Network Learning

A model that is able to predict image manipulations can be more explored in a one-class learning context to detect forgeries. We propose to train the network on manipulations that are only applied to original samples. The original class refers in this context to genuine document images, i.e. without forged textual content. For the experiments, we created fraudulent samples generated by copy-paste forgeries on private datasets, which are detected if the model fails to predict the correct manipulation class label depending on computed scores. We used as manipulations the compression of image blocks with different quality factors. Each quality factor corresponds to a class label. Inspired by the method in (Golan and El-Yaniv, 2018) which detects out-of-distribution images by learning geometric transformations (classification problem), we consider applied manipulations in our context as non-geometric transformations of the input images.

In fact, we address the problem of forgery detection as a learning process of a scoring function (Schlegl et al., 2017). Let S be the set of original samples and $C = \{C_0, C_1, ..., C_{k-1}\}$ the set of compression quality factors. For any original image sample $x \in S$, j is the true label of the manipulated sample $C_j(x)$. We use the set $S_C := \{(C_j(x), j) : x \in S, C_j \in C\}$ to learn our shallow deep k-class classification model $f_{\mathbf{w}}$ ($\mathbf{w}$ are the model parameters). The model is trained over S_C using the cross-entropy loss function. A normality scoring function can be then defined assuming that all of the conditional distributions are independent: $\mathcal{N}_S(x) \doteq \sum_{j=0}^{k-1} \log p\Big[\text{softmax}\Big(f_{\mathbf{w}}(C_j(x))\Big)\Big|C_j\Big]$. Higher scores would correspond to the set of original samples. To predict the manipulation class at inference time, unseen original and fraudulent samples undergo each of the applied compression factors on the training set. We compute the final score based on the vector of softmax layer responses:

$$\mathcal{N}_S(x) = \frac{1}{k} \sum_{j=0}^{k-1} \Big[\text{softmax}(C_j(x))\Big]_j. \tag{4}$$

The model performance is typically assessed by measuring the area under the receiver operating characteristic curve (AUC) metric. Besides the AUC metric, we expect in realistic applications to detect forged document content considering a predefined *threshold* condition. Therefore, the pre-trained model predicts if a sample x is fake when the model fails to predict the correct manipulation class label, i.e. $\mathcal{N}_S(x) <$ *threshold*.

5 EXPERIMENTAL RESULTS

We conduct experiments to evaluate our approach under different scenarios. Two main experiments are considered: (1) the multi-class classification problem of a set of manipulations and (2) the forgery detection problem of copy-paste operations.

Table 1: List of image manipulation operations. Resampling and compression parameters are randomly selected from the given set.

Median filtering	$WindowSize = 3$
Gaussian blurring	$std = 0.5$, $WindowSize = 3$
Additive Gaussian noise	$std = 1.1$
Resampling	$Factor \in \{0.9, 1.1\}$
JPEG compression	$QualityFactor \in \{90, 91, ..., 100\}$

For (1), we use the publicly available PRImA dataset (Antonacopoulos et al., 2009) which presents a wide range of realistic contemporary documents (478 scanned documents). We consider the first scanned version that is saved in a lossless format as the reference quality and original content, with regards to eventually applied manipulations and forgeries. A PRImA document contains four types of annotated regions: text, image, table and graphical object. We extract randomly from each region 64×64 patches converted to grayscale. We follow (Castillo Camacho and Wang, 2022) to create manipulated patches using the operations listed in Table 1. We classify indeed six classes for each patch: original + 5 manipulated versions. Finally, the training/validation set includes about 83183 patches ($\approx$ 13863 per class), and the testing set comprises 9249 patches ($\approx$ 1540 per class). We define the training parameters as follows: the batch size equals 64, the total number of epochs is 60, the optimizer is based on SDG with the momentum=0.95 and the weight_decay=0.0005, the initial learning rate is 10^{-3} that decreases by a factor of 0.5 every six epochs.

We compare with two existing CNN models for image manipulation detection, namely (Bayar and Stamm, 2018) and (Castillo Camacho and Wang, 2022). We show in Figure 2 the evolution of test accuracy metric over 60 epochs of the model training. Our model outperforms the compared references and achieves promptly a stable high performance.

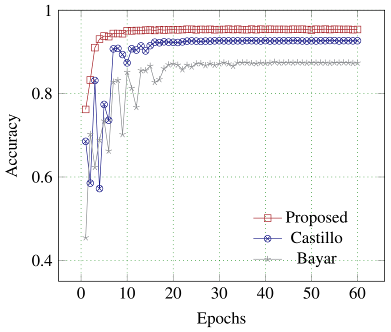


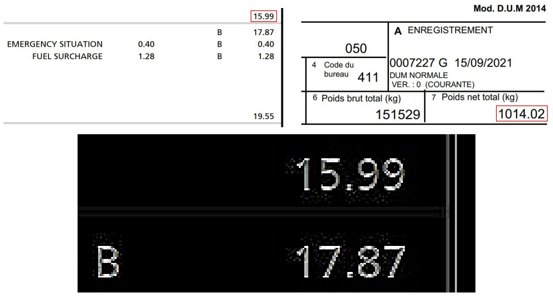
Figure 2: Evolution curves of test accuracy for the multi-class manipulation detection problem.

We compute the classification accuracy on the testing set (test accuracy) as the percentage of correctly predicted patches among all the test patches of different class labels. In Table 2, the accuracy percentages are shown for each model. We inspected our model performance under three scenarios: “Without Init-layer” without error map extraction at the first layer, “Default kernel filter initialization” using first layer kernel filter Xavier initialization (Glorot and Bengio, 2010) and “High-pass kernel filter initialization” as the original version of the proposed method.

Table 2: Test accuracy (in %, average of 5 runs) of the multi-class classification problem on each network. We report the higher accuracy of the proposed method according to different configurations.

Model	Without Init-layer	Default kernel filter initialization		High-pass kernel filter initialization	
		3 filters	30 filters	3 filters	30 filters
(Bayar and Stamm, 2018)	—	87.95	87.92	85.17	87.77
(Castillo Camacho and Wang, 2022)	—	88.84	89.18	92.83	93.07
Proposed	93.001	**92.75**	**92.82**	**94.04**	**95.26**

We have not excluded the first layer for the compared methods since it represents a main conceptual block in their networks during the training step. Even without Init-layer, the test accuracy of our model performs better with 93% against Xavier kernel initialization of the first layer. Besides, the use of high pass filters improved the performance except for the Bayar method which suggests specific filter computation over randomly initialized first layer weights. The proposed and Castillo models implicate 3 randomly selected SRM filters and the overall set of 30 filters. We can clearly notice the valuable impact of using an adaptive learning with respect to the document image characteristics. Typical deep CNN models could not be appropriate and the conception of initialization strategies should be carefully implemented. Our model achieved the best accuracy with 95.26% using 30 SRM kernel filters as Init-layer kernels.


15.99
B 17.87
EMERGENCY SITUATION 0.40 B 0.40
FUEL SURCHARGE 1.28 B 1.28
19.55

Mod. D.U.M 2014
A ENREGISTREMENT
050
4 Code du bureau 411
0007227 G 15/09/2021
DUM NORMALE
VER. : 0 (COURANTE)
6 Poids brut total (kg) 151529
7 Poids net total (kg) 1014.02

15.99
B 17.87

Figure 3: First row: Two examples of extracted document regions from Private_Set_1 (left) and Private_Set_2 (right). The red squares surround the forged content (fake information). Second row: The image gradient (Laplacian) of the forged region of the upperleft example. No particular noise can be visually observed around the forged region.

In the forgery detection experiments (2), we inspect our proposed residual-based network over two private document datasets received from a workflow of industrial partners. We are constrained to make the documents publicly available as they are subject of an confidentiality agreement. We consider Private_Set_1 (2374 authentic images) and Private_Set_2 (444 authentic images) of two companies to analyse the authenticity of their business bills. The images are in JPEG format that represents the common used extension of scanned business documents. We create a mixed dataset (Mixed_Private_Set) including all Private_Set_1 and Private_Set_2 documents. We use our implemented software tool to automatically generate copy-paste forgeries between blocks having relatively the same scale and dimension. XML annotation files are generated for each image including bounding box information of the original block (copied) and the fraudulent block (pasted in a new location). For Private_Set_2, we perform 5 generation runs to produce approximately the same number of patches as for Private_Set_1. The selection of copy-paste regions is random from one image to another. Hence, different blocks are selected at each run. We extract 64 × 64 patches in intersection with original and forged blocks. For Private_Set_1, the training and testing sets include 24241 patches (only originals) and 12312 patches (6090 originals + 6222 forged), respectively. For Private_Set_2, the training and testing sets comprise 24659 patches (only originals) and 12546 patches (6195 originals + 6351 forged), respectively. Mixed_Private_Set consists of 48983 original patches for training, and 24857 patches for testing (12202 originals + 12655 forged). We show two examples of forged blocks in Figure 3. In each set, the documents have similar layouts and tabular structures. As illustrated by the surrounded frauds, we cannot capture perceptually the forged blocks of such uniform textual patterns with homogeneous backgrounds.

Table 3: AUC and test accuracy (in %, average of 5 runs) of the forgery detection problem according to private document datasets.

Model	Private_Set_1		Private_Set_2		Mixed_Private_Set	
	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy
(Castillo Camacho and Wang, 2022)	93.43	84.09	**96.82**	89.72	**93.59**	84.03
Proposed	**94.72**	**88.43**	96.01	**91.69**	92.54	**85.32**

We train our model with only transformed original patches (manipulated explicitly). We have precisely 6 classes according to applied compression quality factors: $[70, 75, 80, 85, 90, 95]$. Each applied quality factor is associated to a class label. We use the same training parameters as in the manipulation detection experiments, with a total number of epochs equal to 30 (chosen empirically). We present in Table 3 the AUC and testing accuracy metrics. At the testing stage, we predict the compression class of both original and forged patches and then we take the average as in Eq. 4. The predicted scores and the ground truth labels of the original patches are used to compute the AUC, which represents the recognition performance of the genuine content. It is mandatory to avoid high percentages of false fraud alerts in a company workflow process. Moreover, the accuracy is evaluated using a predefined threshold. If the average of predicted scores of a given test patch is below 0.7 then we judge it as fraudulent content. This means that the model can not predict accurately the correct class for *abnormal* compression noise distributions. We compute the accuracy percentages based on the ground truth labels of original and patches. Compared to (Castillo Camacho and Wang, 2022), we have further achieved superior performance for the overall private sets.

6 CONCLUSION

In this paper, a shallow residual network is proposed with three basic residual blocks and shortcut connections. We demonstrated that our architecture is suitable to extract deep discriminative features with respect to perceptual document characteristics. Convolutional kernels of an additional first layer are initialized with high-pass filters to learn low-level prediction error features. A forgery detection method is additionally implemented based on a one-class learning process including only original samples during the training stage. In a realistic document flow, we experience much more authentic scanned images than manipulated or forged ones. Therefore, as a major advantage, we do not depend on a large fraud document dataset to provide a relevant pre-trained model. In the future we will perform further experiments to investigate other transformations besides compression. We are interested in extending the application of our model to detect various cases of forgeries.

ACKNOWLEDGEMENTS

This work was supported by the Nouvelle-Aquitaine region under the grant number 2019-1R50120 (CRASD project) and AAPR2020-2019-8496610 (CRASD2 project) and by the LabCom IDEAS under the grant number ANR-18-LCV3-0008.

REFERENCES

Abramova, S. and Böhme, R. (2016). Detecting copy–move forgeries in scanned text documents. *Electronic Imaging*, 2016(8):1–9.

Ahmed, A. G. H. and Shafait, F. (2014). Forgery detection based on intrinsic document contents. In *International Workshop on Document Analysis Systems (DAS)*, pages 252–256.

Antonacopoulos, A., Bridson, D., Papadopoulos, C., and Pletschacher, S. (2009). A realistic dataset for performance evaluation of document layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 296–300.

Bayar, B. and Stamm, M. C. (2018). Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706.

Bertrand, R., Gomez-Krämer, P., Terrades, O. R., Franco, P., and Ogier, J.-M. (2013). A system based on intrinsic features for fraudulent document detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 106–110.

Bertrand, R., Terrades, O. R., Gomez-Krämer, P., Franco, P., and Ogier, J.-M. (2015). A conditional random field model for font forgery detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 576–580.

Beusekom, J., Shafait, F., and Breuel, T. M. (2013). Text-line examination for document forgery detection. *International Journal of Document Analysis and Recognition*, 16(2):189–207.

Brassil, J. T., Low, S., and Maxemchuk, N. F. (1999). Copyright protection for the electronic distribution of text documents. *Proceedings of the IEEE*, 87(7):1181–1196.

Castillo Camacho, I. and Wang, K. (2022). Convolutional neural network initialization approaches for image manipulation detection. *Digital Signal Processing*, 122:103376.

Choi, J.-H., Lee, H.-Y., and Lee, H.-K. (2013). Color laser printer forensic based on noisy feature and support vector machine classifier. *Multimedia Tools and Applications*, 67:363–382.

Cruz, F., Sidere, N., Coustaty, M., d'Andecy, V. P., and Ogier, J.-M. (2017). Local binary patterns for document forgery detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1223–1228.

Cruz, F., Sidère, N., Coustaty, M., Poulain D'Andecy, V., and Ogier, J. (2018). Categorization of document image tampering techniques and how to identify them. In *International Workshop on Computational Forensics (IWCF)*, pages 117–124.

Fridrich, J. and Kodovsky, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256.

Golan, I. and El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9781–9791.

Gomez-Krämer, P. (2022). Verifying document intergrity. In *Multimedia Security 2: Biometrics, Video Surveillance and Multimedia Encryption*, volume 2, pages 59–89. Wiley-ISTE.

Gomez-Krämer, P., Rouis, K., Diallo, A. O., and Coustaty, M. (2023). Printed and scanned document authentication using robust layout descriptor matching. *Multimedia Tools and Applications*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Huang, K., Tian, X., Yu, H., Yu, M., and Yin, A. (2019). A high capacity watermarking technique for the printed document. *Electronics*, 8(12):1403.

James, H., Gupta, O., and Raviv, D. (2020). OCR graph features for manipulation detection in documents. *CoRR*, abs/2009.05158.

Joren, H., Gupta, O., and Raviv, D. (2022). Learning document graphs with attention for image manipulation detection. In *International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI)*, pages 263–274.

Joshi, S. and Khanna, N. (2020). Source printer classification using printer specific local texture descriptor. *IEEE Transactions on Information Forensics and Security*, 15:160–171.

Nandanwar, L., Shivakumara, P., Pal, U., Lu, T., Lopresti, D., Seraogi, B., and Chaudhuri, B. B. (2021). A new method for detecting altered text in document images. In *International Conference of Pattern Recognition and Artificial Intelligence (ICPRAI)*, pages 93–108.

Raiko, T., Valpola, H., and LeCun, Y. (2012). Deep learning made easier by linear transformations in perceptrons. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 924–932.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging (IPMI)*, pages 146–157.

Shang, S., Kong, X., and You, X. (2015). Document forgery detection using distortion mutation of geometric parameters in characters. *Journal of Electronic Imaging*, 24(2):1 – 10.

Tan, L. and Sun, X. (2011). Robust text hashing for content-based document authentication. *Information Technology Journal*, 10(8):1608–1613.