



A Conditional GAN Based Approach for Distorted Camera Captured Documents Recovery

Mohamed Ali Souibgui¹(✉), Yousri Kessentini², and Alicia Fornés¹

¹ Computer Vision Center Computer Science Department,
Universitat Autònoma de Barcelona, Bellaterra, Spain
{msouibgui,afornes}@cvc.uab.es

² Digital Research Center of Sfax, 3021 MIRACL Laboratory,
Sfax University, Sfax, Tunisia
yousri.kessentini@crns.rnrt.tn

Abstract. Many of the existing documents are digitized using smart phone's cameras. These are highly vulnerable to capturing distortions (perspective angle, shadow, blur, warping, etc.), making them hard to be read by a human or by an OCR engine. In this paper, we tackle this problem by proposing a conditional generative adversarial network that maps the distorted images from its domain into a readable domain. Our model integrates a recognizer in the discriminator part for better distinguishing the generated images. Our proposed approach demonstrates to be able to enhance highly degraded images from its condition into a cleaner and more readable form.

Keywords: Mobile phone captured images · Document enhancement · Generative adversarial networks

1 Introduction

With the increasing daily use of smartphones and the advancement of its applications, they start replacing other tools and machines in many different tasks, such as scanning. Nowadays, smartphones could be used to digitize a document paper by simply taking a photo from its camera. Indeed, smartphones allow to scan anywhere compared to a classic scanning machine that is not mobile due to its size and weight. However, despite the mobility advantage, problems are occurring in most of the camera based scans: bad perspective angles, shadows, blur, light unbalance, warping, etc. [17]. Consequently, the extracted text from these document images by directly using a standard Optical Character Recognition (OCR) system becomes unreliable. Lately, thanks to the success of deep and machine learning models, some recent works show a higher robustness when reading distorted documents (at line level). Anyway, some of these methods apply a preprocessing step to segment the scanned text images into separated

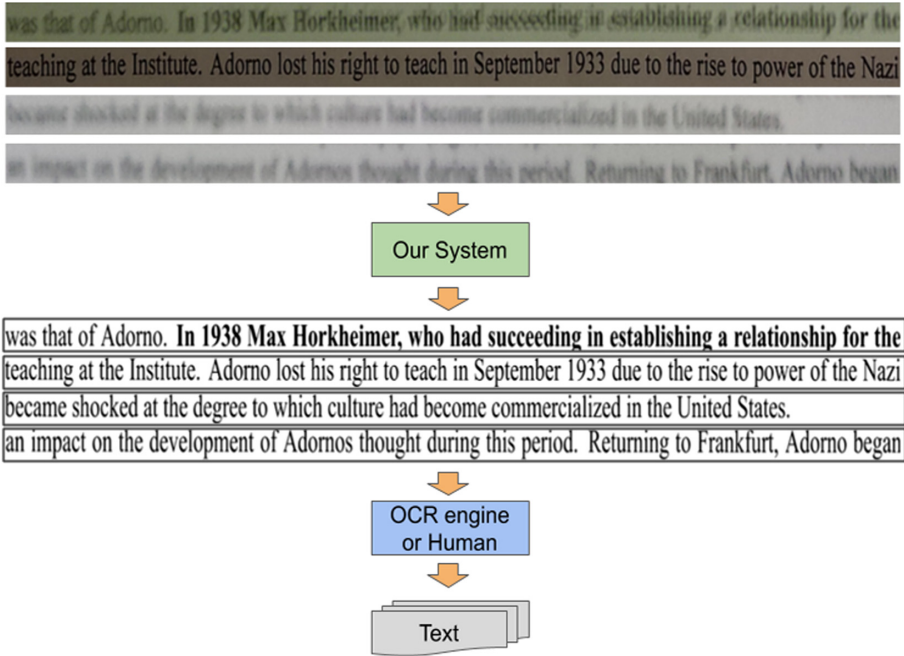


Fig. 1. The proposed reading process: the role of our system is to preprocess the images to be read by an OCR system or by a human.

lines. For example, [2] applied the Long Short-Term Memory (LSTM) networks directly on the gray-scale text line images (without removing the distortions), to avoid error-prone binarization of blurred documents, as the authors claimed. Similarly, [3] used Convolutional Neural Networks (CNN) to extract the features from the lines and pass it through a LSTM to read the text. All these approaches lead indeed to a better performance comparing to using a standard OCR engine in this specific domain (i.e. distorted line image). Those neural networks could be seen as direct mapping functions from the distorted lines to the text. This means that they are not providing a clean version of the image lines to be read by a human, or by the widely used OCR systems that are much powerful when dealing with clean images because they are trained on a huge amount of data from different domains and languages. For this reason, we believe that restoring the lines images, i.e. mapping it from the distorted domain to a readable domain (by an OCR system or by a human) is a better solution. Figure 1 illustrates our approach: a preprocessing module to improve the posterior reading step (either manual or automatic).

Knowing that the OCR accuracy has largely depended on the preprocessing step since it was generally the first step in any pattern recognition problem, a lot of research has addressed the preprocessing stage (i.e. document binarization and enhancement) during the last decades. The goal of this step is to transform the

document image into a better or cleaner version. In our case, this means to remove (or minimize) the distortions in these lines (e.g. shadows, blur and warping). The most common step to clean a text image is binarization, which is usually done by finding either locally or globally thresholds to separate the text pixels from the distorted ones (including background noise) using the classic Image Processing (IP) techniques [18, 19, 21]. These approaches could be used to remove the shadows and fix the light distortion, but, they usually fail to restore the blurred images or to fix the baselines. Thus, machine learning techniques for image domain translation have been recently used for this purpose. These methods mainly consist of CNN auto-encoders [4, 14, 16] and Generative Adversarial Networks (GANs) [7, 11, 22, 23]. The latter is leading to a better performance comparing to the classic IP techniques because they can handle more complex distortion scenarios like: dense watermarked [22], shadowed [8, 13], highly blurred [10, 22] and warped [15] document images.

But, despite the success of the mentioned machine learning approaches for images domain translation, they are still addressing those distortion scenarios separately. Contrary, in this paper we are providing a single model to solve different types of degradation in camera captured documents [6, 17]. Moreover, in those image domain translation approaches, the goal is mapping an image to a desired domain depending only on the visual pixels information loss. In our case, when translating the text images, they should not only look clean, but also, legible. It must be noted that, sometimes, the model could consider the resultant images as text, but in fact they are just random pixels that emulate the visual shape characteristics of text, or random text characters that are constructing a wrong and random script. For this reason, current machine learning text generation models are using a recognition loss in addition to the visual loss to validate the readability of a generated text image [1, 12]. Similarly, we add a recognizer in our proposed conditional GAN model to guide the generator in producing readable images (by the human or the OCR) when translating them from the distortion domain to the clean domain. This simple idea shall lead to a better recovery of our distorted lines.

The rest of the paper is organized as follows. Our proposed model is described in the next Section. Afterwards, we evaluate it comparing with related approaches in Sect. 3. Finally, a brief conclusion is added in Sect. 4.

2 Proposed Method

The proposed architecture is illustrated in Fig. 2. It is mainly composed of three components: A regular generator G , a discriminator D (with the assigned trainable parameters θ_G and θ_D , respectively) and an OCR system R , which will not be trainable since it will only be used to validate the generations. It must be noted that we used the same generator and discriminator architectures as [22], because of the superiority that they showed in document enhancement tasks.

During training, the generator is taking as an input the distorted image, noted by I_d and outputting a generated image I_g , hence: $I_g = G_{\theta_G}(I_d)$. Then,

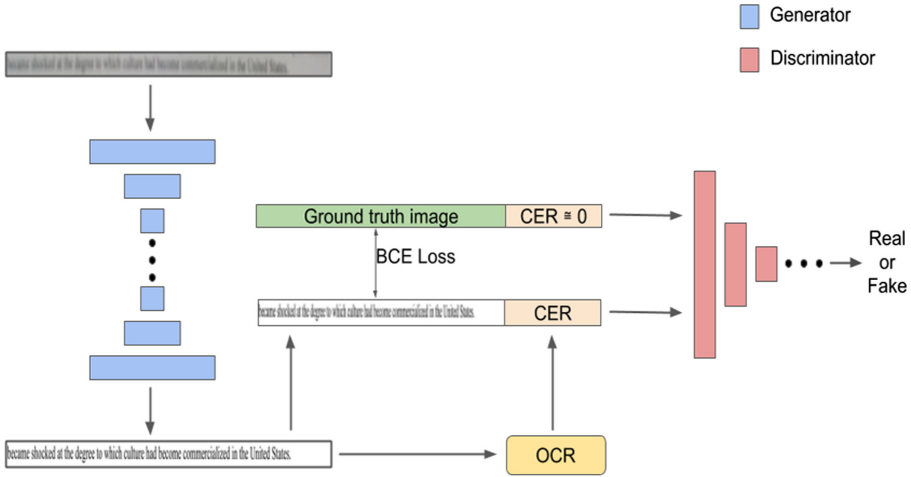


Fig. 2. The proposed architecture.

the generated image is passed through the recognizer (OCR system) to get the recognition accuracy measured by the Character Error Rate (CER) $CER_g = R(I_g)$. After that, a matrix having the same shape of I_g is created and filled with the resultant CER_g . The matrix is concatenated with I_g over the depth and passed to the discriminator with the label Fake to train it. The discriminator is looking, of course, to the Ground Truth (GT) images I_{gt} which are concatenated with a CER that is close to zero and labeled as real.

Clearly, concatenating a matrix with the same number of pixels as the generated image could be replaced by attaching a simple loss directly to the CER and force it to be reduced. However, the choice of a CER matrix was done to let the method be extendable on measuring the error rate from each word (even character or pixel) separately. Thus, we can provide a better feedback to the model, so that it can focus on enhancing the parts with high CER in the image (which could be known from the matrix), while keeping the parts of the image line that were correctly recovered (with low CER in the matrix).

The discriminator is then used to predict the degree of ‘reality’ (i.e. how realistic) of the generated image, where $P(Real) = D_{\theta_D}(I_g, CER_g)$. We noted that it is better to assign a high CER for the GT images at the beginning of the training stage and then starting to decrease it after some epochs. Thus, we start with a weak discriminator that we progressively enhance it in parallel with the generator to get a better adversarial training. The whole adversarial training could be formalized, hence, with the following loss:

$$L_{GAN}(\theta_G, \theta_D) = \mathbb{E}_{I_d, I_{gt}} \log[D_{\theta_D}(I_d, I_{gt}, CER \approx 0)] + \mathbb{E}_{I_d} \log[1 - D_{\theta_D}(I_d, G_{\theta_G}(I_d), CER_g)] \quad (1)$$

To speed up the convergence of the generator parameters θ_G , we use an additional loss which is the usual Binary Cross Entropy (BCE) between the generated images and the ground truth images. The whole Loss becomes:

$$L(\theta_G, \theta_D) = \min_{\theta_G} \max_{\theta_D} L_{GAN}(\theta_G, \theta_D) + BCE(\theta_G) \quad (2)$$

For a better understanding, we describe in what follows each architecture of the used components.

2.1 Generator

Similar to [22], the used generator is following the U-net encoder-decoder architecture detailed in [20]. It consists of 17 fully convolutional layers with the encoder-decoder fashion, 8 layers for the encoder (down-sampling with max-pooling every two layers) until getting to the 9th layer, followed by a 10th for the decoder (up-sampling every two layers), with an employed skip connections (a concatenation between the layers). Table 1 presents the architecture. As it can be seen, the output is an image with 1 channel since we are providing a grey scale image.

Table 1. Generator architecture: the channels and skip connections are presented. In the U-net model all the convolutions have the kernel size 3×3 .

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Channels	64	64	128	128	256	256	512	512	512	512	512	256	256	128	128	64	64	2	1
Skip-con	-	16	-	14	-	12	-	10	-	8	-	6	-	4	-	2	-	-	-

2.2 Recognizer

We used Tesseract 4.0 as a recognizer. This OCR engine version is based on deep learning techniques (LSTM), which show a good recognition performance. The recognizer takes an image as input and outputs its predicted text. Anyway, it must be noted that any other OCR system could be used for this purpose.

2.3 Discriminator

The defined discriminator is composed of 6 convolutional layers described in Table 2, which outputs a 2D matrix containing probabilities of the generated image denoting its realistic degree. The discriminator receives three inputs: the degraded image, its cleaned version (ground truth or cleaned by the generator) and the obtained CER. Those inputs are concatenated together in a $H \times W \times 3$ shape. Then, the obtained volume is propagated in the model to end up in a $\frac{H}{16} \times \frac{W}{16} \times 1$ matrix in the last layer. This matrix contains probabilities that should be, to the discriminator, 1 if the clean image represents the ground truth and 0 if it is coming from the generator. Therefore, the last layer takes a sigmoid as an activation function. Once the training is finished, this discriminator is no longer used. Given a distorted image, we only use the generative network to recover it. However, the discriminator shall force the generator during training to produce a realistic result, in addition to the BCE loss with the GT images.

Table 2. Discriminator architecture. All the convolutions are with kernel size 4×4 . A max-pooling is performed after each layer, except the last one.

Layer	1	2	3	4	5	6
Channels	64	128	256	256	256	1

3 Experiments and Results

As mentioned above, the goal of this study is to provide a mapping from the distorted document into a clean and readable version. For evaluation, we compare our proposed approach with the relevant methods that can handle the same task in this Section.

3.1 Dataset and Training Details

For a fair comparison, all the methods will be tested on the same dataset containing the distorted lines images and its clean version. This data was taken from SmartDoc-QA [17], which is constituted from smartphone’s camera captured document images, under varying capture conditions (light, shadow, different types of blur and perspective angles). SmartDoc-QA is categorized in three subsets of documents: contemporary documents, old administrative documents and shop’s receipts. For computational reasons, we use only the contemporary documents category in our experiments. An example of those documents is presented in Fig. 3.

A preprocessing step was done to segment those documents at line level and construct our desired dataset. First, we extract the document paper from the background by applying a Canny edge detector [5] and finding the four corners of the document. Then, a geometric transformation is done for dewarping. Finally, the horizontal projection was applied to detect the lines. This results in 17000 lines images pairs (distorted and clean); from them, 10200 pairs were taken for training the different approaches and 6800 pairs for testing.

The training was done for 80 epochs with a batch size of 32, and the Adam optimization algorithm was used with a learning rate of $1e-4$.

3.2 Evaluated Approaches and Metrics

We study the performance of our developed method by comparing it with the following existing approaches, which were widely used for similar tasks:

- DE-GAN [22]: This method uses our same architecture, but without a recognizer (only a generator and a discriminator). In this way, we can evaluate if adding the recognizer helps to provide cleaner documents.

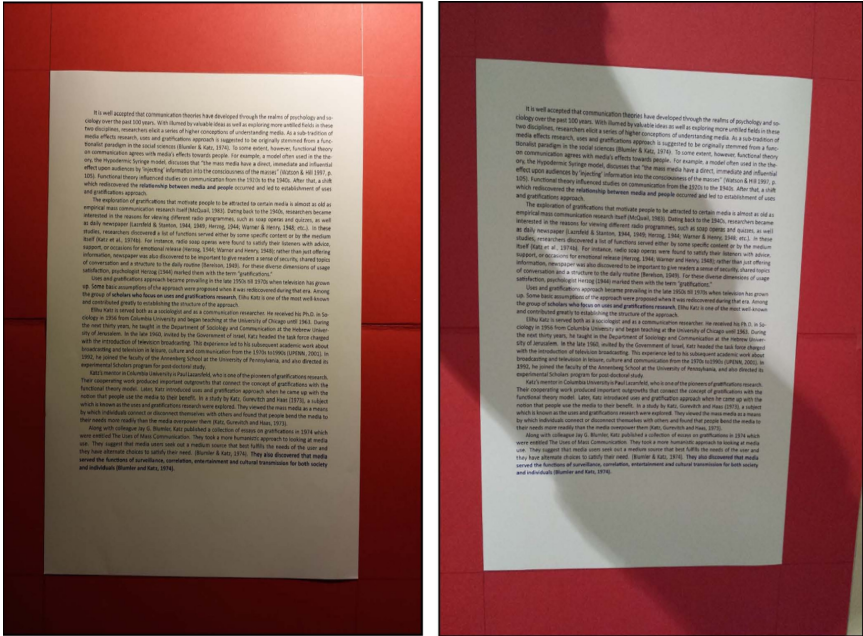


Fig. 3. Examples of two documents from QmartDoc-QA dataset

- Pix2Pix-HD [23]: This method extends [11] to provide a higher resolution and more realistic images. Anyway, both methods falls in the set of the widely used approaches to translate images between different domains.
- CNN [10]: In this approach, a CNN is used to clean the image. Concretely, it was proposed for the goal of text images deblurring.

The comparison is performed using two types of metrics: The first type is for measuring the visual similarity between the predicted images and the GT images. For this purpose, we use the Peak signal-to-noise ratio (PSNR) and Structural Similarity Index Measure (SSIM) [9]. The second metric type is for measuring the readability of the provided image. For this purpose, we simply use the CER metric after passing the cleaned images through Tesseract 4.0. The CER metric is defined as $CER = \frac{S+D+I}{N}$, where S is the number of substitutions, D of deletions, I of insertions and N the ground-truth's length. So, the lower the CER value, the better.

3.3 Results

Table 3. Comparative results between the different approaches.

Approach	SSIM	PSNR	CER
Distorted lines	0.33	9.08	0.25
CNN [10]	0.54	13.54	0.29
DE-GAN [22]	0.51	12.03	0.26
Pix2pix-HD [23]	0.45	11.45	0.66
Our approach	0.52	12.26	0.18

The obtained results are presented in Table 3. As it can be seen, cleaning the distorted images using the different approaches leads to a higher SSIM and PSNR compared to the degraded lines (without any cleaning). This means that we are able to recover a visually enhanced version of the lines images using any of these approaches, with a slightly better performance using the CNN [10] approach. But, this does not mean that all these approaches are leading to better versions of the line images. Because, the text is also an important factor to evaluate the cleaning.

Anyway, the CER of the distorted images is much better than the cleaned ones when using the CNN, pix2pix-HD and DE-GAN approaches. As stated before, the reason is that the text in those methods is degrading during the mapping to the clean space. Since the model is only enhancing the visual form of the distorted line images. Contrary, when using our proposed approach, we observe that the CER is also boosted with 7% compared to the distorted images. This demonstrates the utility of using the recognition rate input in our proposed model, which cleans the image while taking the text preservation into account. Thus, from the found results, we can conclude that our model is the best way to perform the distorted to clean text image mapping among the different compared approaches.

Moreover, To illustratively compare the performance of the different methods, we show in what follows some qualitative results. In Fig. 4, we present the recovering of a slightly distorted line. This means that it could be correctly read by the OCR even without any preprocessing, since the distortion is only consists in the baseline due to the warped document and in the background color. It could be observed from the figure that applying the CNN, pix2pix-HD and DE-GAN methods is fixing the baseline and cleaning the background, but deteriorating the text quality and leads to some character errors when reading by the OCR. Contrary, our proposed approach is the one that mostly preserves the text readability while visually enhancing the text line. Another example of a slightly blurred and warped line is also presented in Fig. 5. Despite the fact that the OCR result on the distorted image is still similar to applying it on our generated line (with a clear superiority compared to the CNN and pix2pix-HD methods), it is

GT	2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis
Tesseract	2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis
Distorted	<i>2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis</i>
Tesseract	2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis
CNN	<i>2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis</i>
Tesseract	U3 cve chicken broth — thag heecy mustant atic is the black pepper kosher saltet malesuede lames ec trp
DE-GAN	<i>2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis</i>
Tesseract	1/3 cup chicken proth — tbsp heney mustard stir in the biack pepper, koster “salts? malesuaiz tames ac turpls
Pix2pix-HD	<i>2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis</i>
Tesseract	1} mupdwldver broil tlre-honey mamend ow te the Keck pepuen iociier saber miiesowile lames to dunia
Ours	2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac turpis
Tesseract	2/3 cup chicken broth 1 tbsp honey mustard stir in the black pepper, kosher saltet malesuada fames ac tarpis

Fig. 4. Results of the different approaches for fixing a warped line image. Errors made by the Tesseract reading engine are shown at character level using the red color. (Color figure online)

clear that our model is producing a much easier image to read by a human, since it is successfully deblurring and unwarping it. We also note in this example that the use of the regular DE-GAN (our same architecture except the recognizer) is resulting in a weak discriminator, which could be fooled by a wrong generation. This can be observed from comparing the visual similarity between our approach and DE-GAN. But, when reading the text, more DE-GAN’s character errors are made compared to our generated text.

Next, we show the recovery of some highly distorted lines in Fig. 6. In this case, we tried to recover two distorted lines containing high blur, shadows and warping. Obviously, reading those lines directly with Tesseract is the worst option since it is clearly leading to a bad result by missing a lot of words which results in a high CER. However, by applying the different cleaning approaches, we are able to remove the distortion and produce a better text. Same as previous experiments, it can be seen that our proposed model is achieving the highest results by giving the best line image recovery. Our produced image is visually

GT	chocolate pieces that are poking up; it will make for a more attractive cookie. sprinkle lightly with sea salt and
Tesseract	chocolate pieces that are poking up; it will make for a more attractive cookie. sprinkle lightly with sea salt and
Distorted	<i>chocolate pieces that are poking up; it will make for a more attractive cookie. sprinkle lightly with sea salt and</i>
Tesseract	chocolate pieces that are poking up. H will mabe for c more attractive cookie yprinkle lightly with sea salt and
CNN	<i>chocolate pieces that are poking up; it will make for a more attractive cookie. sprinkle lightly with sea salt and</i>
Tesseract	Chocolate piece: that are poting te, @ wel mate fer 2 mere smrecmye cecte. tonnale betty with sea salt aed
DE-GAN	chocolate pieces that are poking up, it will make for a more attractive cookie. sprinkle lightly with sea salt and
Tesseract	chocolate pieces that are poking up, it will make for @ more stractive okie spnnikie iigitiy with ses salt end
Pix2pix-HD	<i>chocolate pieces that are poking up; it will make for a more attractive cookie. sprinkle lightly with sea salt and</i>
Tesseract	chontines tiecrs dum anegohmevy wil. mile tor amoue attmahe ovailtx. opmitly nemby-with iew.cait and
Ours	chocolate pieces that are poking up, it will make for a more attractive cookie. sprinkle lightly with sea salt and
Tesseract	chocolate plices that are poking up, it will male for a more attractive cookie. sprinkle lightly with sea salt anid

Fig. 5. Results of the different approaches for fixing a blurred and warped line image, errors made by Tesseract reading engine are shown in character level with the red color. (Color figure online)

closed to the GT image, with a preserved text, that can be seen from the low CER compared to different methods.

Finally, it is worth to mention that our proposed model was sometimes failing to produce readable lines. This was happening when dealing with the extremely distorted lines. Some examples of this particular case are presented in Fig. 7. As can be seen, despite the fact that some words have been correctly enhanced and recognized by the OCR after applying our method, the line images are still visually degraded and unsatisfactory. Of course, this is happening due to the extreme complexity of fixing such lines, which are hard to be read even by the human naked eye.

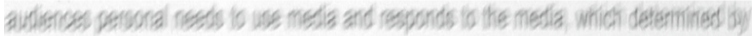

GT	audiences personal needs to use media and responds to the media, which determined by
Distorted	
Tesseract	WS 0 WBE Ia and responds (6 the media. which determined by
CNN	audiences personal needs to use media and responds to the media. which determined by
Tesseract	guciences setsona needs io use media and responcs io the media. wich deternmed ie
DE-GAN	audiences personal needs to use media and responds to the media, which determined by
Tesseract	audiences personal needs io woe medie and responds to the medi, wri detennined ny
Pix2pix-HD	audiences personal needs to use media and responds to the media, which determined by
Tesseract	1} audiences perional reads fo use ments and rszeonde io the medis, witiich determined tia
Ours	audiences personal needs to use media and responds to the media, which determined by
Tesseract	auciences persond! needs fo use medie and responds to the media, which deterttiined yr
<hr/>	
GT	mounds of dough (the size of generous golf balls) onto baking sheet, making sure to turn horizontally any
Distorted	
Tesseract	wen horizontally any
CNN	mounds of dough (the size of generous golf balls) onto baking sheet, making sure to turn horizontally any
Tesseract	moune: of Gowgh fihe mae of pemereet get balai ees bate thet mating rere te ture hormontaily any
DE-GAN	mounds of dough (the size of generous golf balls) onto baking sheet, making sure to turn horizontally any
Tesseract	mounds of dough {the sie et cenereat gol! ssl sem bakig sheet, mskinz sure te tum homentaily ary
Pix2pix-HD	mounds of dough (the size of generous golf balls) onto baking sheet, making sure to turn horizontally any
Tesseract	predt a? bout. .the due of geewerios gut! neckd wir tating cheer, muting aare te lum keotteerath, an
Ours	mounds of dough (the size of generous golf balls) onto baking sheet, making sure to turn horizontally any
Tesseract	mounds of dough ithe sire ef generaus goll bald este bakiog sheet, making sure te tum horivontally any

Fig. 6. Results of the different approaches for fixing two distorted line images. Errors made by the Tesseract reading engine are shown at character level in red color. (Color figure online)

GT	minutes, or until a skewer comes out clean when tested. cool on a wire rack. 1 tsp baking powder 1 cup caster
Distorted	
Tesseract	—
Ours	minutes, or until a skewer comes out clean when tested. cool on a wire rack. 1 tsp baking powder 1 cup caster
Tesseract	cupune: ar emtledeSrer domescet dex wherenes caolong mrenct 'rogeleog sever Lae cele?
<hr/>	
GT	by which individuals connect or disconnect themselves with others and found that people bend the media to
Distorted	
Tesseract	by wich indivichuails Commact or Pacaniert themncetver mith others ancl Mound at yeogie bend Sa mena
Ours	by which individuals connect or disconnect themselves with others and found that people bend the media to
Tesseract	by which inciattiueds connect or dscernect themselves with other amnd fawnd that peaoie bend the media ns
<hr/>	
GT	satisfaction, psychologist Herzog (1944) marked them with the term "gratifications."
Distorted	
Tesseract	-
Ours	satisfaction, psychologit hermg (1944) merled them with the tens 'gratifications.'
Tesseract	vatislacrian, ocychoiogia nenmng (1S44) merled ther weh the tens 'we-officelwes'

Fig. 7. Results of our approach for cleaning the extremely distorted line images. Errors made by the Tesseract reading engine are shown at word level in red color. (Color figure online)

4 Conclusion

In this paper we have proposed an approach for recovering distorted camera captured documents. The goal is to provide a clean and readable version of the document images. Our method integrates an OCR to cGAN model to preserve the readability while translating the document domain. As a result, our method leads to a better CER compared to the widely used methods for this task.

As future work, our proposed model could be extended to handle full pages instead of lines. Furthermore, the CER matrix provided for the discriminator could include the error rates at local level instead of passing the CER of the whole text line. This could help the discriminator to provide a better feedback to

the generative model, and thus, improve the overall model performance. Finally, it will be interesting to test the model on historically handwritten degraded documents, using of course, a Handwritten Text Recognition system instead of the OCR system.

Acknowledgment. This work has been partially supported by the Swedish Research Council (grant 2018-06074, DECRYPT), the Spanish project RTI2018-095645-B-C21, the Ramon y Cajal Fellowship RYC-2014-16831 and the CERCA Program/Generalitat de Catalunya.

References

1. Alonso, E., Moysset, B., Messina, R.: Adversarial generation of handwritten text images conditioned on sequences. In: 15th International Conference on Document Analysis and Recognition (ICDAR) (2019). <https://doi.org/10.1109/ICDAR.2019.00083>
2. Asad, F., Ul-Hasan, A., Shafait, F., Dengel, A.: High performance OCR for camera-captured blurred documents with LSTM networks. In: 12th IAPR Workshop on Document Analysis Systems (DAS) (2016). <https://doi.org/10.1109/DAS.2016.69>
3. El Bahi, H., Zatni, A.: Text recognition in document images obtained by a smart-phone based on deep convolutional and recurrent neural network. *Multimed. Tools Appl.* **78**(18), 26453–26481 (2019). <https://doi.org/10.1007/s11042-019-07855-z>
4. Calvo-Zaragoza, J., Gallego, A.J.: A selectional auto-encoder approach for document image binarization. *Pattern Recogn.* **86**, 37–47 (2019)
5. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 679–698 (1986). <https://doi.org/10.1109/TPAMI.1986.4767851>
6. Chabchoub, F., Kessentini, Y., Kanoun, S., Eglin, V., Lebourgeois, F.: SmartATID: a mobile captured Arabic text images dataset for multi-purpose recognition tasks. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 120–125 (2016)
7. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. Fan, H., Han, M., Li, J.: Image shadow removal using end-to-end deep convolutional neural networks. *Appl. Sci.* **9**, 1–17 (2019). <https://doi.org/10.3390/app9051009>
9. Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 20th International Conference on Pattern Recognition (ICPR) (2010). <https://doi.org/10.1109/ICPR.2010.579>
10. Hradiš, M., Kotera, J., Zemčík, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: British Machine Vision Conference (BMVC), pp. 6.1–6.13, September 2015. <https://doi.org/10.5244/C.29.6>
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
12. Kang, L., Riba, P., Wang, Y., Rusiñol, M., Fornés, A., Villegas, M.: GANwriting: content-conditioned generation of styled handwritten word images. *Arxiv preprint* (2020)
13. Le, H., Samaras, D.: Shadow removal via shadow image decomposition. In: The IEEE International Conference on Computer Vision (ICCV), October 2019

14. Lore, K.G., Akintayo, A., Sarkar, S.: LLNet: a deep autoencoder approach to natural low-light image enhancement. *Pattern Recogn.* **61**, 650–662 (2017)
15. Ma, K., Shu, Z., Bai, X., Wang, J., Samaras, D.: DocUNet: document image unwarping via a stacked U-Net. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018
16. Meng, G., Yuan, K., Wu, Y., Xiang, S., Pan, C.: Deep networks for degraded document image binarization through pyramid reconstruction. In: *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 2379–2140 (2017). <https://doi.org/10.1109/ICDAR.2017.124>
17. Nayef, N., Luqman, M.M., Prum, S., Eskenazi, S., Chazalon, J., Ogier, J.M.: SmartDoc-QA: a dataset for quality assessment of smartphone captured document images - single and multiple distortions. In: *13th International Conference on Document Analysis and Recognition (ICDAR)* (2015). <https://doi.org/10.1109/ICDAR.2015.7333960>
18. Niblack, W.: *An Introduction to Digital Image Processing*. Strandberg Publishing Company, Birkerød (1985)
19. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979). <https://doi.org/10.1109/TSMC.1979.4310076>
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. *Arxiv preprint* (2015)
21. Sauvola, J., Pietik, M.: Adaptive document image binarization. *Pattern Recogn.* **33**, 225–236 (2000)
22. Souibgui, M.A., Kessentini, Y.: DE-GAN: a conditional generative adversarial network for document enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020). <https://doi.org/10.1109/TPAMI.2020.3022406>
23. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)