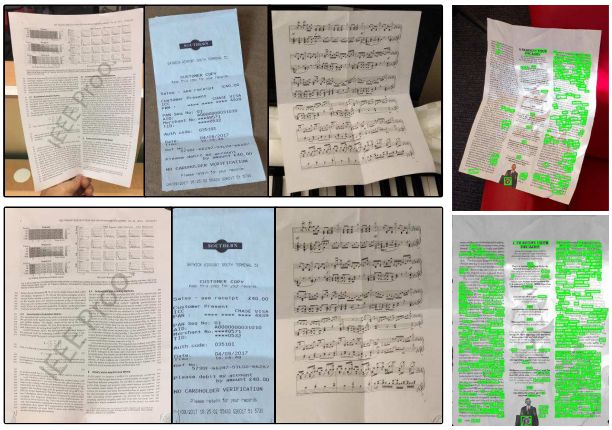
DocUNet: Document Image Unwarping via A Stacked U-Net

Ke Ma[1] Zhixin Shu[1] Xue Bai[2] Jue Wang[2] Dimitris Samaras[1]
[1]Stony Brook University [2]Megvii Inc.
[1]{kemma, zhshu, samaras}@cs.stonybrook.edu [2]{baixue, wangjue}@megvii.com

Abstract

Capturing document images is a common way for digitizing and recording physical documents due to the ubiquitousness of mobile cameras. To make text recognition easier, it is often desirable to digitally flatten a document image when the physical document sheet is folded or curved. In this paper, we develop the first learning-based method to achieve this goal. We propose a stacked U-Net [25] with intermediate supervision to directly predict the forward mapping from a distorted image to its rectified version. Because large-scale real-world data with ground truth deformation is difficult to obtain, we create a synthetic dataset with approximately 100 thousand images by warping non-distorted document images. The network is trained on this dataset with various data augmentations to improve its generalization ability. We further create a comprehensive benchmark[1] that covers various real-world conditions. We evaluate the proposed model quantitatively and qualitatively on the proposed benchmark, and compare it with previous non-learning-based methods.



(a) (b)

Figure 1. **Document image unwarping and its usefulness.** (a) shows some results from our network. The top row shows input images and the bottom shows the outputs. (b) Our network greatly improves the effectiveness of state-of-the-art text detection systems [10]: more words (shown in green) can be detected from the rectified document image (bottom) than the original folded one (top).

1. Introduction

Document digitization serves as an important means to preserve existing printed documents, making them easier to access anywhere at anytime. Traditionally, documents are digitized with flat-bed scanners, which are non-portable, expensive devices. Recently, with the increasing popularity of mobile cameras, taking pictures of physical documents has become the easiest way for scanning physical documents. Once captured, images can be further processed by text detection and recognition pipelines for content analysis and information extraction.

A common practical problem when taking document images is that the document sheets are not in an ideal condition for scanning: they may be curved, folded or crumpled, or laying on a complex background. Think about a crumpled receipt taken out from a pocket. All these factors could cause significant problems for automatic document image analysis procedures down-stream, as shown in Fig. 1. It is thus desirable to digitally flatten such documents in the captured images.

Flattening document images has been addressed before, using various approaches. Some vision systems rely on carefully designed, well calibrated hardware such as stereo cameras [35, 29], or structured light projectors [1, 21] to measure the 3D distortion of the documents. They produce high quality results, but the additional hardware limits their application. Other work [28, 36] obviate the need for extra hardware by utilizing multi-view images to reconstruct the 3D shape of distorted document sheets. Others aim to recover the rectified document by analyzing a single image, based on various hand-crafted low-level features such as illumination/shading [32, 37], text-lines [27, 20] etc.

This paper presents a novel learning-based method to recover arbitrarily curved and folded paper documents captured in the wild. Unlike previous techniques, our method

[1]http://www.cs.stonybrook.edu/~cvl/docunet.html

is the first end-to-end learning-based method to directly predict document distortion. Previous methods have only used learning for feature extraction, while the final image recovery was still based on traditional optimization procedures. Our method instead relies on Convolutional Neural Networks (CNNs) for end-to-end image recovery. Compared to optimization-based methods, the feedforward network is very efficient in the testing phase. Moreover, this data-driven method can be better generalized to many document types (text, figures, handwriting, etc.) if proper training data is provided.

We formulate this task as seeking the appropriate 2D image warping that can rectify a distorted document image. Our network predicts a mapping field that moves a pixel in the distorted source image $S(u, v)$ to (x, y) in the result image D:

$$D(x, y) = S(u, v). \tag{1}$$

Formulating the problem in this way, we find this task shares some commonalities with semantic segmentation. For the latter, the network assigns a class label to each pixel. Similarly, our network assigns a 2-dimensional vector to each pixel. This inspires us to use U-Net [25] in our network structure, which is widely known for its success in semantic segmentation. To adapt it to our regression problem, we define a novel loss function to drive the network to regress the coordinate (x, y) in D for each pixel in S.

Obtaining large-scale data with ground truth labels is the first challenge for deep supervised learning. To train our network, we need to obtain a large number of document images distorted in various degrees as input, as well as the corresponding deformations that can lead to perfect rectifications. Currently no such dataset exists. Obtaining ground truth deformation in the physical world is very challenging. We thus resort to synthetic data for training. We synthesize 100K images by randomly warping perfectly flat document images, so that the perturbed image is the input and the mesh we used to warp the image is the inverse deformation that we aim to recover.

There is no comprehensive benchmark publicly available to evaluate document unfolding. Previous methods either evaluate on a small number of images, or the datasets only contain one or two types of distortion such as smooth curving. We fill this vacuum by creating a benchmark of 130 images that contains large variations in document type, degree and type of distortion, as well as capture conditions.

Our primary contributions include:

i) The first end-to-end, learning-based approach for document image unwarping. We propose a stacked U-Net [25] with intermediate supervision. It is trained in an end-to-end manner to predict the forward mapping that can rectify the distorted document.

ii) A technique to synthesize images of curved or folded paper documents. Using this method, we create a large-scale dataset containing approximately 100K images for training.

iii) A diverse evaluation benchmark dataset with ground truth, on which we evaluate our method and compare against previous methods.

2. Related Work

Rectifying documents has been studied in the literature. We roughly categorize previous methods into two groups:

3D shape reconstruction. To reconstruct the 3D shape of the paper document, Brown and Seales [1] used a visible light projector-camera system. Zhang et al. [38] utilized a more advanced range/depth sensor, and also took into consideration the physical properties of paper for shape recovery. More recently, Meng et al. [21] set up a platform with two structured laser beams to acquire the document curl. Besides additional hardware, other work relied on multi-view images for 3D shape reconstruction. Ulges et al. [29] computed the disparity map between two images via image patch matching. Yamashita et al. [35] parameterized the 3D shape with Non-Uniform Rational B-Splines (NURBS). Tsoi and Brown [28] did not require a well calibrated stereo vision system. They utilized the boundary information from multi-view images and composed these images together to generate the rectified image. Similarly, Koo et al. [13] used two uncalibrated images from different views to measure 3D shape by SIFT matching. Östlund et al. [24] proposed a mesh parameterization method that can reconstruct the 3D shape of a deformable surface given correspondences with a reference image. Recently, You et al. [36] reconstructed the 3D shape of the document by modeling the creases on the paper from multi-images.

Shape from low-level features. Low level features include illumination/shading, text lines etc. Wada et al. [32] formulated the problem using Shape from Shading (SfS). The distorted document exhibits varying shading under a directional light source. Courteille et al. [5] extended this work by using a camera instead of a scanner and estimate perspective shape from shading. Zhang et al. [37] proposed a more robust SfS system that can handle shadows and background noise. Some other methods relied on analyzing the document content. A prevalent strategy is to trace textlines [8, 30, 18, 21, 11, 16], under the assumption that textlines in the rectified document should be horizontal and straight. In particular, Cao et al. [2] modeled the curved document on a cylinder and Liang et al. [14] used a developable surface. Tian and Narasimhan [27] optimized over textlines as the horizontal clue and character strokes as the vertical clue to produce the 3D mesh. These works can be seen as special cases of the broader Shape from Texture

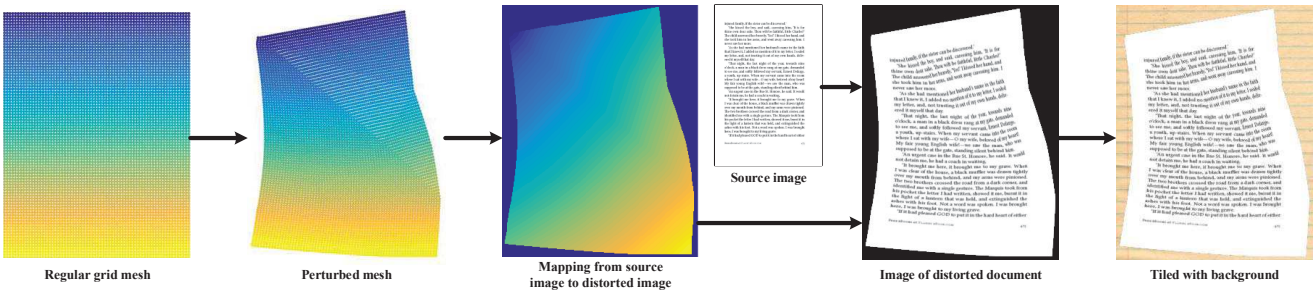


Figure 2. **Synthetic distorted document image in 2D.**

problem (SfT) [34, 19, 9]. Recently, Das et al. [6] applied a CNN to detect paper creases for rectification. However, CNN was used only as one step in their optimization process rather than in an end-to-end manner.

Our proposed method can be classified into the second category. It however differs from all previous methods as a purely data-driven method. The network is trained end-to-end to predict the forward mapping for the warping, so no hand-crafted low-level features are used. There is no optimization process during the test phase. Our network has two benefits: i) it can handle various document types and conditions as long as large-scale training data is provided. ii) it can be deployed as an efficient method in real-world applications.

3. Dataset

The proposed method is based on a CNN, which needs to be trained with large-scale training data. In this task, document deformation can be represented as a 3D mesh, surface normals, 2D flow etc. Accurately capturing it in any form in the real world is difficult. Additional hardware like a range camera or calibrated stereo vision system is required, while the accuracy of the estimated deformation usually depends on the cost of the hardware. Furthermore, it is almost impossible to manually fold/distort document papers that can cover all real-world situations.

We consider using synthetic data for training, a common step in recent deep learning systems [31, 26]. This allows full control over the variations in the dataset, such as 3D mesh shape, illumination, material, etc.

One straightforward idea is to directly render distorted documents in 3D rendering engines. This is however impractical due to the following reasons. First, physically-correct 3D paper meshes are hard and slow to generate using physical simulation [22]. Second, rendering via path tracing is also time-consuming, *e.g.* it takes around 1 minute to render an image in [6]. Rendering 100K images would take more than two months.

3.1. Distorted Image Synthesis in 2D

We directly synthesize training images in 2D. Though the underlying physical modeling is neglected, manipulating a 2D mesh is much easier and the images are faster to generate. As our purpose is to map the distorted paper to the rectified one, the data synthesis is the inverse process, that is, we warp the rectified image into different distortions.

When creating the distortion maps, we follow the following empirical guidelines:

i) A piece of real paper is a locally-rigid fabric. It will not expand or compress. The deformation at one point will propagate spatially.

ii) There are two kinds of distortions: folds and curves generating creases and paper curls. In the real world there is usually a mix of these two basic distortions.

We first collect a large amount of flat digital documents including papers, books and magazine pages. We then warp these images as shown in Fig. 2. The procedure is detailed below.

Perturbed mesh generation: Given an image I, we impose an $m \times n$ mesh M on it to provide control points for warping. A random vertex $\boldsymbol{p}$ is selected on M as the initial deformation point. The direction and strength of the deformation is denoted as $\boldsymbol{v}$ and is also randomly generated. Finally, based on observation i), $\boldsymbol{v}$ is propagated to other vertices by weight w. The vertices on the distorted mesh are computed as $\boldsymbol{p_i} + w\boldsymbol{v}, \forall i$.

It is crucial to define w. As $\boldsymbol{p}$ and $\boldsymbol{v}$ define a straight line, we first compute the normalized distance d between each vertex and this line, and define w as a function of d. Based on observation ii), we define a function for each distortion type. For folds:

$$w = \frac{\alpha}{d + \alpha}, \tag{2}$$

and for curves:

$$w = 1 - d^{\alpha}, \tag{3}$$

where α controls the extent of the deformation propagation. Overall, a larger α leads w towards 1, which means all other

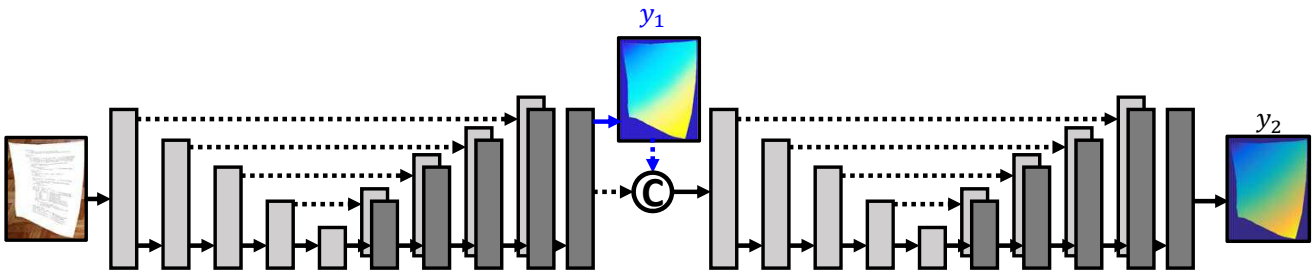
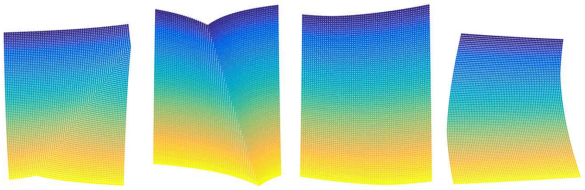


Figure 3. **Network architecture.** Our network is a stack of two U-Nets. The network splits and produces a forward mapping y_1 from the output of the first U-Net. The same loss applied at y_2 is also applied at y_1. Then y_1 is concatenated with the output feature map of the first U-Net and serves as the input for the second U-Net.ⓒ represents the concatenation operator. y_2 can be directly used to generate the rectified image.



(a) (b) (c) (d)

Figure 4. **Effects of deformation functions.** (a) and (b) show the fold effect based on Eq. 2. (a) for large α, (b) for small α. (c) and (d) show the curve effect based on Eq. 3. (c) for large α and (d) for small α.

Figure 5. **Sample images in the synthetic dataset.**

vertices share the same deformation as $\boldsymbol{p}$ making the deformation more global, while a small α limits the deformation to the local area around p. The effects of the two functions are demonstrated in Fig. 4.

Perturbed image generation: The perturbed mesh provides a sparse deformation field. We interpolate it linearly to build a dense warping map at pixel level. The perturbed image can then be generated by applying the warping map to the original image. We synthesized 100K images on a single CPU in this way. Each image contains up to 19 synthetic distortions (30% are *curving* distortions and 70% *folding*). *Curving* needs to preserve that the Gaussian curvature should be 0 everywhere while *folding* is arbitrary. Some samples are shown in Fig. 5.

3.2. Data augmentation

Models trained on synthetic data may not generalize well to real data due to the gap between real and synthetic data. The problem can be eased by domain adaption, for example, using Generative Adversarial Networks (GAN) [26]. However, large scale real-world data are not available. We alleviate this problem by augmenting the synthesized images with various transformations. First, we use the texture images from the Describable Texture Dataset (DTD) [4] to produce various background textures. We then add jitter in the HSV color space to magnify illumination and paper color variations. A projective transform is further applied to cope with viewpoint change. Our experiments show that these data augmentation methods greatly improve the network generalization abilities.

4. DocUNet

4.1. Network Architecture

Similar to semantic segmentation, we design our network to enforce pixel-wise supervision. We select U-Net [25] as our base model due to its simplicity and effectiveness in semantic segmentation tasks. Basically U-Net is a Fully Convolutional Network [17]. It contains a series of downsampling layers followed by a series of upsampling layers. The feature maps are concatenated between downsampling layers and upsampling layers. Note that we modify the padding scheme in the original U-Net implementation to make the input and output of the network have the same spatial size.

However, the output of a single U-Net may not be satisfactory and should be refined. Inspired by the work of successive predictions and progressive refinement [23, 3], we stack another U-Net at the output of the first U-Net as a refiner.

In our network, as shown in Fig. 3, we have one layer to convert the deconvolutional features into the final output (x, y). The first U-Net splits after the last deconvolution layer. The deconvolution features of the first U-Net and the intermediate prediction $y1$ are concatenated together as the input of the second U-Net. The second U-Net finally gives a refined prediction y_2, which we use as the final output of our network. We apply the same loss function to both y_1

and y_2 during training. However, at test time, only $y2$ is used as the output of the network.

The input to the network is $S \in \Re^{M_s \times N_s \times 3}$ and the output is a mapping $F \in \Re^{M_s \times N_s \times 2}$ from the distorted image to the rectified image. Different from semantic segmentation which is a pixel-wise classification problem, computing F is a regression process. The output of the semantic segmentation network usually has C channels where C is the number of the semantic classes. The proposed network only outputs two channels for the (x, y) coordinates.

4.2. Loss Function

We define the loss function as a combination of an element-wise loss and a shift invariant loss. An $L2$ element-wise loss is:

$$L_e = \frac{1}{n} \sum_i (y_i - y_i^*)^2, \tag{4}$$

where n is the number of elements in F, y_i is the predicted value at index i and y_i^* is the corresponding ground truth value.

The shift invariant loss L_s does not care about the absolute value of y_i in F. It enforces that the difference between y_i and y_j should be close to that between y_i^* and y_j^*. So L_s can be written as:

$$L_s = \frac{1}{2n^2} \sum_{i,j} ((y_i - y_j) - (y_i^* - y_j^*))^2. \tag{5}$$

Assuming $d_i = y_i - y_i^*$, Eq. 5 can be written as:

$$\begin{aligned} L_s &= \frac{1}{2n^2} \sum_{i,j} (d_i - d_j)^2 \\ &= \frac{1}{2n^2} \sum_{i,j} (d_i^2 + d_j^2 - 2d_i d_j) \\ &= \frac{1}{n} \sum_i d_i^2 - (\frac{1}{n} \sum_i d_i)^2. \end{aligned} \tag{6}$$

The first term is just an element-wise loss. The second term decreases the loss if the distance between two elements is similar to that in the ground truth. This loss is also known as Scale-Invariant Error [7]. The weights of the two terms are the same in Eq. 5. We can assign different weights to these two terms during training. We also observe that the $L1$ loss is better than the $L2$ loss. So we rewrite the loss function as:

$$L^f = \frac{1}{n} \sum_i |d_i| - \frac{\lambda}{n} | \sum_i d_i |, \tag{7}$$

where λ controls the strength of the second term. We use 0.1 in all our experiments.

The elements in F corresponding to the background pixels in S have a constant negative value of -1 as described in section 3. So part of the loss in Eq. 7 is due to the background. Actually it is not necessary for the network to precisely regress these elements to -1. Any negative value should suffice. Thus, we use hinge loss for the background pixels:

$$L^b = \frac{1}{n} \sum_i \max(0, y_i), \tag{8}$$

while using Eq. 7 for the foreground pixels only.

5. Experiments

We first introduce our benchmark for the evaluation of the rectification of distorted paper documents in the wild from a single image. Then we evaluate our proposed learning-based method and compare the results with a prior art non-learning-based method [27].

5.1. Benchmark

Images: The images in this benchmark are photos of physical paper documents captured by mobile cameras. We collected 65 paper documents of various content/format, and for each document we took two photos, resulting in 130 images in total. The benchmark contains both the original photos and the tightly cropped ones. We use the latter one in our experiments because we focus on paper dewarping rather than localizing the document in the image. The benchmark is created considering the following factors:

i) Document types. Our selected documents include various types such as receipts, letters, fliers, magazines, academic papers and books. Most of them contain a mix of text and figures. Some of them are pure text and others contain only images. Some are in color and some are black and white. Most of the text is in English, while some is in Japanese and Chinese. Some music sheets are also included. Most of the documents have a white background, but documents with different background colors are also included.

ii) Distortions. The original flat paper documents were physically distorted by different people. Each document was deformed into two different shapes. In order to assure the diversity of the benchmark, we included both easy and hard cases. For easy cases, the document may only have one crease or one curl or a "common fold" [6]. For hard cases, the document may be heavily crumpled. We left few documents flat and untouched on purpose to test if a method will preserve them. In particular, 94.6% of the images contain *curving*; 31.7% contain more than 6 folds; 8.5% are paper crumples, which are challenging cases.

iii) Environments. The photos were taken by two people with two different cellphones, under various indoor and

(a) original photos

(b) document centered cropped images

(c) scans from a flatbed scanner

Figure 6. **Samples in the benchmark.**

outdoor scenes, with different illumination conditions: sunlight, indoor lights or the cellphone's built-in flash light. We also changed the viewpoint to some extent.

The resulting benchmark contains very diverse examples and can reasonably approximate the most common use cases.

Ground truth: Before folding the collected paper documents, we scanned them using a flatbed scanner. We adjusted the size and overall color of the obtained images to match the original flat documents as much as possible. Fig. 6 shows some examples in our benchmark.

Evaluation Scheme: Previous work suggests two different evaluation schemes. One is based on Optical Character Recognition (OCR) accuracy and the other is based on the measurement of image similarity. We choose the latter, as OCR accuracy heavily depends on which OCR module is used, and cannot deal with documents that mostly contain figures. For image similarity we use Multi-Scale Structural Similarity (MS-SSIM) [33], as the task is mainly about document structure rather than pixel-level color accuracy. We also adopt a method that uses dense SIFT flow [15] to evaluate the Local Distortion (LD) proposed by You et al. [36].

Single scale SSIM is calculated on all image patches in an image. It is based on the similarity of statistical measures including the mean value and variance within each image patch. The implementation of MS-SSIM builds a Gaussian pyramid of the input image and the target image. MS-SSIM is a weighted sum of the SSIM computed at each level. Besides the weighted sum, we also report single scale SSIM at different levels.

Method	MS-SSIM	LD	time(s)
Tian *et al.* [27]	0.13	33.69	212
Ours	**0.41**	**14.08**	**4**

Table 1. **Comparison between our method and [27].** Methods are evaluated by Multi-Scale Structural Similarity (MS-SSIM) and Local Distortion (LD). For MS-SSIM, higher value means higher similarity with ground truth. For LD, lower value means lower displacement. Our method achieves much better and much faster results compared to [27].

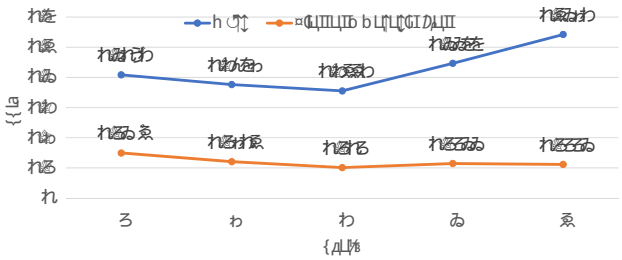


Figure 7. **Structural Similarity comparison at different scales.** At every scale, our method outperforms Tian and Narasimhan [27]. Scale 1 is the original image and scale 5 is the image at the top of the Gaussian pyramid.

5.2. Experiment Setup

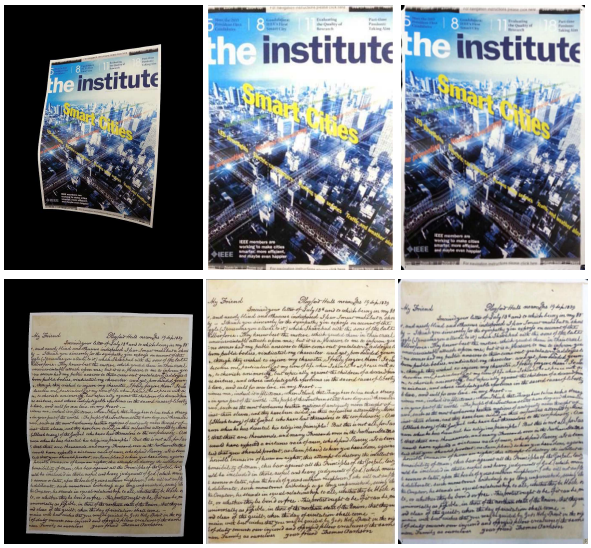
We use 90K images in our dataset for training and the rest for validation. We use Adam [12] as our optimizer. The training starts with a learning rate of 0.0002 and reduces to one fifth of that when the validation loss reaches a plateau. For fair comparison, none of the documents used in the benchmark are used to create the synthetic data for training. In fact, the benchmark contains more diverse data than the training set, as a way to test the generalization ability of our method.

We compare our results quantitatively with Tian and Narasimhan [27] on our benchmark using their publicly available code. We also compare our results qualitatively with two recent works by You et al. [36] and Das et al. [6]. Given that there is no publicly available code for these methods, we test our trained model on their examples shown in their papers.

We resize all output images and target images to have the same $598,400$ area, while keeping their aspect ratio. We use a 5-level-pyramid for MS-SSIM and the weight for each level is $0.0448, 0.2856, 0.3001, 0.2363, 0.1333$, which is inherited from the original implementation. For SIFT flow, we use the code provided by the author with default parameters.

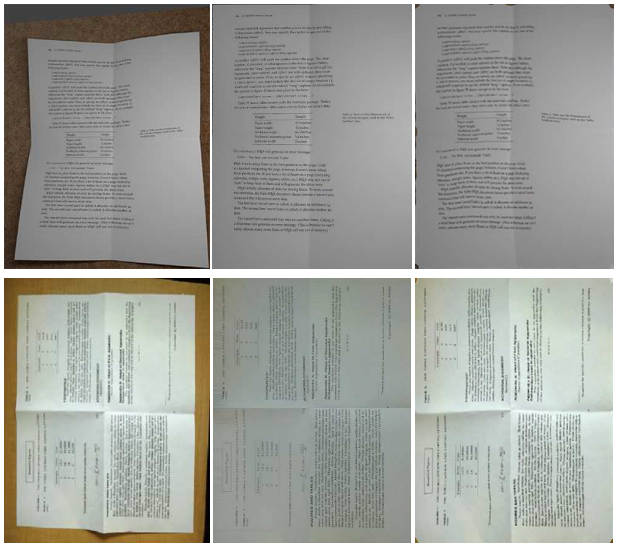
5.3. Results

Our benchmark evaluation shows that the proposed method outperforms the method proposed by Tian and



distorted image result of [36] our result

Figure 8. **Comparison with You et al. [36].** Even though [36] utilizes five to ten images, our method is still quite comparable.

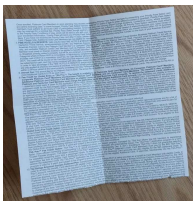


distorted image result of [6] our result

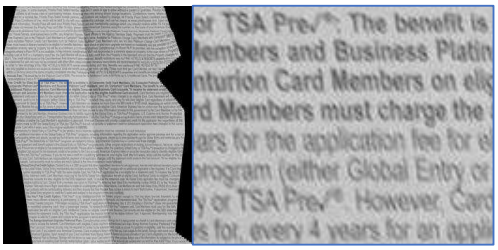
Figure 9. **Comparison with Das et al. [6].** The method in [6] is specially designed to work well for two folds condition. Our method also works well under this condition. The two images are from [6].

Narasimhan [27]. Specifically, our method achieves 0.41 in terms of MS-SSIM and an average LD of 14.08 pixels. [27] only achieves 0.13 in MS-SSIM and 33.69 in LD. This is because their method is mainly designed for documents with textlines and it heavily depends on the quality of textline tracing. As a result, it cannot handle more complex documents that have mixed textlines and figures, or areas where textline tracing fails, as shown in Fig. 10.

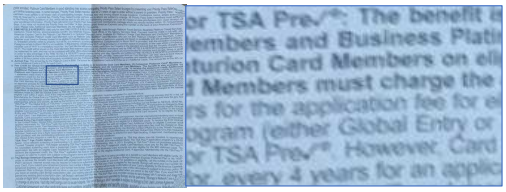
In terms of computational efficiency, [27] takes 3 to 4 minutes to process one image on a CPU with their Matlab



(a) input image



(b) result of [27]



(c) our result

Figure 10. **Comparison with Tian and Narasimhan [27].** The method in [27] relies substantially on the tracing of textlines. So if the textline can not be traced, it will be replaced by a black area in the rectified result. However, recovery of details is slightly better than ours.

implementation. Though it is not a fair comparison, our network can run at 28 fps on a GTX 1080 Ti GPU. The bottleneck lies in generating rectified images from the mapping. Our unoptimized Matlab implementation takes around 3 to 4 seconds on a CPU. Overall our method is an order of magnitude faster than [27]. The overall comparison is shown in Tab. 1. The single scale SSIM value is also shown in Fig. 7. Our method has better performance at every scale.

We also test our trained network on the examples provided in You et al. [36] and Das et al. [6]. The qualitative results show that our model can successfully generalize to many other cases. It is worth noting that [36] requires multi-view images as input and [6] only tackles a specific folding type. Our results are comparable to theirs even though our method uses only a single image and is designed to handle arbitrary distortions, as shown in Fig. 8 and Fig. 9. More qualitative results on our benchmark are shown in Fig. 11.

We justify our design of the network and the loss function by conducting a set of ablation experiments. We consider the following three configurations: 1) Stacked U-Net with L^f in Eq. 7 and L^b in Eq. 8 as loss functions. This is the configuration we used thorough our previous experiments. 2) Stacked U-Net with only the conventional $L1$ loss. 3) Single U-Net with $L1$ loss. Comparison results are

(a) input images

(b) output of the network

(c) ground truth

Figure 11. **More results on the benchmark.**

Configuration	MS-SSIM	LD
Stacked U-Net + L^f + L^b	**0.41**	**14.08**
Stacked U-Net + $L1$	0.40	14.85
Single U-Net + $L1$	0.35	15.73

Table 2. **Ablation experiments.** We compare three different configurations. The first one is a stacked U-Net with L^f and L^b as loss functions. The second configuration replace the loss function with $L1$ and the last one uses one single U-Net instead of a stacked U-Net.

in Tab. 2.

The configuration of the proposed network architecture and loss function achieves the best result on our benchmark. Using the $L1$ loss function leads a slight drop of MS-SSIM from 0.41 to 0.40 and LD increases from 14.08 to 14.85. Further replacing a stacked U-Net with a single U-Net decreases MS-SSIM to 0.35 and increases LD to 15.73.

6. Conclusion and Future Work

In this paper, we presented the first end-to-end neural network to flatten and rectify distorted document images. We proposed a stacked U-Net with intermediate supervision and trained it in an end-to-end manner to directly predict the mapping that can remove the distortion. We proposed a procedure to create synthetic training data. We also created a benchmark containing real world images taken under various conditions. Experimental results demonstrated the effectiveness and efficiency of our method.

Certain limitations exist in our work. For example, as shown in the fifth image in Fig. 11 (b), our network doesn't fully recover perspective distortion such that the three columns in the result still contain traces of perspective distortion. This should be addressed in future work. Also in future work, we would like to apply GAN to make our network generalize better to the real-world images. We would also like to incorporate an illumination model to remove highlights or shadows on the rectified image. On the other hand, we will optimize the code generating rectified documents from the mapping and make the whole pipeline real-time and deploy it on mobile devices.

Acknowledgements. This work started when Ke Ma was an intern at Megvii Inc. This work was supported by a gift from Adobe, Partner University Fund, and the SUNY2020 Infrastructure Transportation Security Center.

References

[1] M. S. Brown and W. B. Seales. Document restoration using 3D shape: A general deskewing algorithm for arbitrarily warped documents. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2001. 1, 2

[2] H. Cao, X. Ding, and C. Liu. A cylindrical surface model to rectify the bound document image. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2003. 2

[3] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2017. 4

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014. 4

[5] F. Courteille, A. Crouzil, J.-D. Durou, and P. Gurdjos. Shape from shading for the digitization of curved documents. *Machine Vision and Applications*, 18(5):301–316, 2007. 2

[6] S. Das, G. Mishra, A. Sudharshana, and R. Shilkrot. The Common Fold: Utilizing the Four-Fold to Dewarp Printed Documents from a Single Image. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, DocEng '17, pages 125–128. ACM, 2017. 3, 5, 6, 7

[7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014. 5

[8] H. Ezaki, S. Uchida, A. Asano, and H. Sakoe. Dewarping of document image by global optimization. In *Proceedings of the International Conference on Document Analysis and Recognition*. IEEE, 2005. 2

[9] D. A. Forsyth. Shape from texture and integrability. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2001. 3

[10] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic Data for Text Localisation in Natural Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[11] B. S. Kim, H. I. Koo, and N. I. Cho. Document dewarping via text-line based optimization. *Pattern Recognition*, 48(11):3600–3614, 2015. 2

[12] D. Kinga and J. B. Adam. A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 6

[13] H. I. Koo, J. Kim, and N. I. Cho. Composition of a dewarped and enhanced document image from two view images. *IEEE Transactions on Image Processing*, 18(7):1551–1562, 2009. 2

[14] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):591–605, 2008. 2

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 6

[16] C. Liu, Y. Zhang, B. Wang, and X. Ding. Restoring camera-captured distorted document images. *International Journal on Document Analysis and Recognition*, 18(2):111–124, 2015. 2

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 4

[18] S. Lu and C. L. Tan. Document flattening through grid modeling and regularization. In *Proceedings of the International Conference on Pattern Recognition*. IEEE, 2006. 2

[19] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168, 1997. 3

[20] G. Meng, C. Pan, S. Xiang, and J. Duan. Metric rectification of curved document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):707–722, 2012. 1

[21] G. Meng, Y. Wang, S. Qu, S. Xiang, and C. Pan. Active flattening of curved document images via two structured beams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014. 1, 2

[22] R. Narain, T. Pfaff, and J. F. O'Brien. Folding and Crumpling Adaptive Sheets. *ACM Transactions on Graphics (TOG)*, 32(4):51:1–51:8, 2013. 3

[23] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*. Springer, 2016. 4

[24] J. Östlund, A. Varol, D. T. Ngo, and P. Fua. Laplacian meshes for monocular 3D shape recovery. In *Proceedings of the European Conference on Computer Vision*. Springer, 2012. 2

[25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer, 2015. 1, 2, 4

[26] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 4

[27] Y. Tian and S. G. Narasimhan. Rectification and 3D reconstruction of curved document images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011. 1, 2, 5, 6, 7

[28] Y.-C. Tsoi and M. S. Brown. Multi-view document rectification using boundary. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007. 1, 2

[29] A. Ulges, C. H. Lampert, and T. Breuel. Document Capture Using Stereo Vision. In *Proceedings of the 2004 ACM Symposium on Document Engineering*, DocEng '04, pages 198–200. ACM, 2004. 1, 2

[30] A. Ulges, C. H. Lampert, and T. M. Breuel. Document image dewarping using robust estimation of curled text lines. In *Proceedings of the International Conference on Document Analysis and Recognition*. IEEE, 2005. 2

[31] G. Varol, J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. 3

[32] T. Wada, H. Ukida, and T. Matsuyama. Shape from shading with interreflections under a proximal light source: Distortion-free copying of an unfolded book. *International Journal of Computer Vision*, 24(2):125–135, 1997. 1, 2

[33] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*. IEEE, 2003. 6

[34] A. P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, 1981. 3

[35] A. Yamashita, A. Kawarago, T. Kaneko, and K. T. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *Proceedings of the International Conference on Pattern Recognition*. IEEE, 2004. 1, 2

[36] S. You, Y. Matsushita, S. Sinha, Y. Bou, and K. Ikeuchi. Multiview Rectification of Folded Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2, 6, 7

[37] L. Zhang, A. M. Yip, M. S. Brown, and C. L. Tan. A Unified Framework for Document Restoration Using Inpainting and Shape-from-shading. *Pattern Recognition*, 42(11):2961–2978, 2009. 1, 2

[38] L. Zhang, Y. Zhang, and C. Tan. An improved physically-based method for geometric restoration of distorted document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):728–734, 2008. 2