Noname manuscript No.
(will be inserted by the editor)

DocScanner: Robust Document Image Rectification with Progressive Learning

Hao Feng · Wengang Zhou · Jiajun Deng · Qi Tian · Houqiang Li

Received: date / Accepted: date

Abstract Compared with flatbed scanners, portable smartphones provide more convenience for physical document digitization. However, such digitized documents are often distorted due to uncontrolled physical deformations, camera positions, and illumination variations. To this end, we present DocScanner, a novel framework for document image rectification. Different from existing solutions, DocScanner addresses this issue by introducing a progressive learning mechanism. Specifically, DocScanner maintains a single estimate of the rectified image, which is progressively corrected with a recurrent architecture. The iterative refinements make DocScanner converge to a robust and superior rectification performance, while the lightweight recurrent architecture ensures the running efficiency. To further improve the rectification quality, based on the geometric priori between the distorted and the rectified images, a geometric regularization is introduced during training to further improve the performance. Extensive experiments are conducted on the Doc3D dataset and the DocUNet Benchmark dataset, and the quantitative and qualitative evaluation results verify the effectiveness of DocScanner, which outperforms previous methods on OCR accuracy, image similarity, and our proposed distortion metric by a considerable margin. Furthermore, our DocScanner shows superior efficiency in runtime latency and model size.

Keywords Document image rectification · Progressive learning · Segmentation · OCR · Image similarity

1 Introduction

Document digitization refers to the creation of a digital image backup of a document file, which is frequently applied in many formal affairs. Thanks to the rapid advances in portable cameras and smartphones, document digitization becomes much more accessible than before. However, such captured document images commonly suffer from various levels of distortions, due to uncontrolled camera position, uneven illumination, and various paper sheet deformations (*i.e.*, folded, curved, and crumpled). These distortions make the digital files unqualified on many occasions. Besides, they also bring difficulties to many downstream processings, such as automatic text recognition (Yuan et al., 2022; Peng et al., 2022), content understanding (Zhong et al., 2019; Kim et al., 2022), and question answering (Mathew et al., 2021), editing, and preservation. To address these problems, document image rectification has been actively researched in recent years.

One direction of the early attempts to document image rectification is developed based on the recon-

Hao Feng
University of Science and Technology of China
E-mail: fh1995@mail.ustc.edu.cn

Wengang Zhou
University of Science and Technology of China
E-mail: zhwg@ustc.edu.cn

Jiajun Deng
The University of Sydney
E-mail: jiajun.deng@sydney.edu.au

Qi Tian
Huawei Cloud & AI
E-mail: tian.qi1@huawei.com

Houqiang Li
University of Science and Technology of China
E-mail: lihq@ustc.edu.cn

struction of the 3D shape of the deformed pages. Those methods heavily rely on auxiliary hardware (Brown and Seales, 2001; Brown et al., 2007; Zhang et al., 2008; Meng et al., 2014) or multiview shooting (Brown and Seales, 2004; Yamashita et al., 2004; Koo et al., 2009; You et al., 2018), limiting their further applications. Some other methods (Lavialle et al., 2001; Wu and Agam, 2002; Chew Lim Tan et al., 2006; He et al., 2013) assume a parametric model on the curved pages and optimize the model with specific attributes, such as shading, boundaries, and textlines. Nevertheless, the oversimplified parametric models of such approaches usually lead to limited performance as well as non-negligible computational costs for model optimization.

Recently, deep learning has been introduced to document image rectification with promising performance as well as a significant reduction in computational cost. In deep learning based methods (Ma et al., 2018; Das et al., 2019; Li et al., 2019; Liu et al., 2020; Xie et al., 2020; Das et al., 2020; Markovitz et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Feng et al., 2022; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022), document image rectification is approached as the regression of a dense 2D vector field (warping flow) that samples the pixels from distorted images to rectified ones. Typically, DocUNet (Ma et al., 2018) first demonstrates the potential of deep learning for document image rectification with a stacked U-Net (Ronneberger et al., 2015). Then, DewarpNet (Das et al., 2019) models the 3D shape of a deformed document in the network, while DocGeoNet (Feng et al., 2022) and RDGR (Jiang et al., 2022) leverage the curved textlines to guide the rectification. DocProj (Li et al., 2019) and PWUNet (Das et al., 2021) consider distinct local deformation fields and stitch them together to obtain an improved restoration. Although they report superior performance on the challenging benchmark dataset (Ma et al., 2018), the rectified images remain unsolved distortions and these advanced solutions are still limited by difficulties such as large sheet deformations and distorted textlines. In contrast, we propose to conduct distortion rectification in a progressive manner, aiming to obtain a robust and superior rectification result.

In this work, we introduce DocScanner, a novel deep network framework for document image rectification. Different from existing solutions, DocScanner approaches the task by introducing a progressive learning mechanism. Specifically, DocScanner takes a recurrent structure that corrects the document distortion via iterative and progressive refinements. During training, at each iteration, DocScanner takes the rectification results of the previous iteration as input, aggregates them, and learns to refine the current rectified image toward a distortion-free one. Note that such a refinement operation can be applied iteratively during inference without divergence. In this way, the distortions in the input document images are progressively corrected and finally converge to a relatively steady status, achieving an accurate and robust rectification.

Our DocScanner exhibits a novel design, discussed next. Firstly, our recurrent rectification architecture maintains a single estimate of the rectified image that is refined iteratively. This is different from the intuitive strategy that a rectified image pyramid is supervised to refine the output in a multi-scale way (Zamir et al., 2021; Yang et al., 2021), where large degradations/deformations are recovered at low resolution, while small ones are recovered at high resolution, which may have difficulty in recovering from early errors. Secondly, at each iteration, DocScanner aggregates the results predicted at the previous iteration, including the features of the original distorted image and the current rectified image. Then, a convolution-based gated recurrent unit takes the aggregated features and the current hidden state as input, and outputs the refined rectified image. Thirdly, the recurrent architecture is lightweight with only 4.1M parameters, which ensures efficiency under multiple iterations. Fourthly, we propose a circle-consistency loss as a geometric regularization to further relieve the rectified distortion, which imposes straight-line constraints on rectified images.

Moreover, we propose a new evaluation metric for document image rectification. Based on the dense SIFT-flow (Liu et al., 2011) between ground truth image to rectified one, the typical metric Local Distortion (LD) (You et al., 2018) computes the average displacement of all matched pixels. We observe that LD focuses more on the distortion of local areas. Inspired by this, we propose Line Distortion (Li-D) as a supplementary metric to further evaluate the global distortion of the rectified images, by computing the average deformation of the row and column pixels in rectified images.

Extensive experiments on the Doc3D dataset (Das et al., 2019) and DocUNet Benchmark dataset (Ma et al., 2018) demonstrate the effectiveness of our method as well as its superiority over state-of-the-art methods. In addition, we validate various design choices of DocScanner through extensive ablation studies. We summarize the strengths of DocScanner as follows,

- *State-of-the-art performance*: DocScanner sets several state-of-the-art records on the challenging DocUNet Benchmark dataset (Ma et al., 2018), including metrics MultiScale Structural SIMilarity (MS-SSIM), Local Distortion (LD), our proposed Line Distortion (Li-D), Edit Distance (ED), and Character Error Rate (CER).

- *Superior efficiency*: DocScanner processes 1080P document images at 10.03 FPS on a 2080Ti GPU. Moreover, the parameter number of DocScanner is about **1/5** of the best-published method.
- *Strong generalization ability*: DocScanner exhibits strong generalization ability, demonstrated by the robustness experiments on background, viewpoint, illumination, and document type.

2 Related Work

In this section, we broadly categorize the research on document image rectification into two different directions: (a) rectification by 3D shape reconstruction, and (b) rectification based on deep learning. In the following, we discuss them separately.

Rectification by 3D shape reconstruction. Traditionally, some methods utilize auxiliary equipments to reconstruct a 3D shape of the deformed document, followed by flattening the surface to correct the distortions. Brown and Seales (Brown and Seales, 2001) acquire the 3D representation of document shape with a light projector and then flatten this representation via a mass-spring particle system. Later, they (Brown et al., 2007) acquire a 3D scan of the document surface with a 3D scanning system, and conformal mapping (Lévy et al., 2002) is used to rectify the geometric distortion by mapping the 3D surface to a plane. Zhang *et al* (Zhang et al., 2008) use a laser range scanner and perform restoration by using a physical modeling technique. Meng *et al* (Meng et al., 2014, 2017) introduce structured beams illuminating upon the deformed document page to recover curves of the page surface.

In addition to the methods that rely on the auxiliary hardwares, some other methods utilize two or more multiview images for 3D shape reconstruction. Brown *et al* (Brown and Seales, 2004) utilize a calibrated mirror system to obtain the 3D surface using multiview stereo. Yamashita *et al* (Yamashita et al., 2004) detect the stereo corresponding points between two images based on the normalization cross correlation method. Tsoi *et al* (Tsoi and Brown, 2007) transform the multiple views into a common coordinate frame based on the document boundaries to correct the distortions. Koo *et al* (Koo et al., 2009) estimate the unfolded surface by the corresponding points between two images by SIFT (Lowe, 2004). You *et al* (You et al., 2018) propose a ridge-aware 3D reconstruction method to rectify a paper sheet from a few of images. However, both the auxiliary equipments and multiple images are unavailable in common situations, which limits their applicability.

Moreover, techniques of the third subcategory reconstruct the 3D shape from a single view. They commonly assume a parametric model on the document surface, like a cylindrical surface, and fit the model based on the extraction of specific representations. Among them, some methods (Wada et al., 1997; Chew Lim Tan et al., 2006; Courteille et al., 2007; Zhang et al., 2009) obtain a document shape based on shape from shading technique. Some algorithms are designed based on the priori that the textlines are horizontally or vertically aligned in well-rectified images, thus the distorted images can be corrected based on the detection of textlines. In early work, the detected textlines are modeled as cubic B-splines by Lavialle *et al* (Lavialle et al., 2001; Meng et al., 2011), non-linear curve by Wu and Agam (Wu and Agam, 2002), and polynomial approximation by (Mischke and Luther, 2005; Kim et al., 2015; Kil et al., 2017). Moreover, features about boundaries (Brown and Tsoi, 2006; He et al., 2013), characters (Zandifar, 2007), interline spacing and textline orientation (Koo and Cho, 2010) are extracted to estimate the rectification. Liang *et al* (Liang et al., 2008) estimate the 3D document shape from texture flow information obtained directly from the image. Tian *et al* (Tian and Narasimhan, 2011) compute the 3D deformation up to a scale factor using SVD. Meng *et al* (Meng et al., 2018) estimate the 3D shape model through weighted majority voting on the vector fields. Das *et al* (Das et al., 2019) innovatively model the 3D shape of a document with a convolutional network and then regress the warping flow for rectification.

Rectification based on deep learning. Although the above methods achieve encouraging results, the strong assumptions on surface geometry, contents, and illumination limit their applicabilities. DocUNet (Ma et al., 2018) is the first model to demonstrate the potential of deep learning for document image rectification. It predicts a dense forward warping flow with a stacked U-Net (Ronneberger et al., 2015) to unwarp the distorted document image. DocProj (Li et al., 2019) predicts the warping flow of the cropped distorted document image patches first, rather than the entire image, and then stitches them together to generate a fully rectified image. However, the estimation and subsequent stitching of the warping flow patches heavily increase the computational cost. AGUN (Liu et al., 2020) develops a pyramid encoder-decoder architecture, which predicts the forward warping flow at multiple resolutions in a coarse-to-fine fashion. However, directly feeding the distorted images with complex backgrounds to the network for rectification estimation is difficult, due to the involvement of extra implicit learning to identify the foreground document. Based on Fully Convolutional Network (Long et al., 2015), Xie *et al* (Xie et al., 2020) perform a foreground/background classification

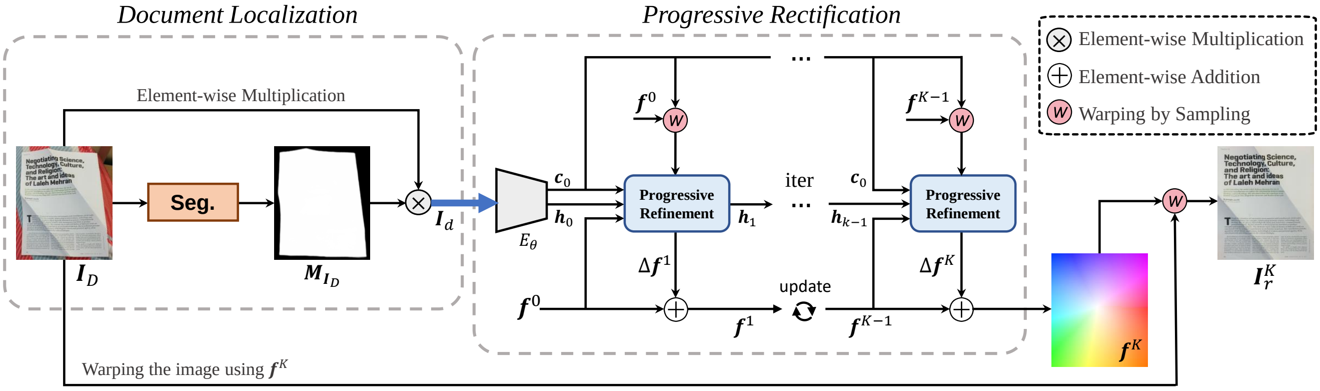


Fig. 1: **An overview of the proposed DocScanner.** It decouples the task into a foreground document localization and a geometric distortion rectification. Given the distorted document image $\boldsymbol{I}_D$, the document localization module first separates the foreground document from the noisy background by predicting a binary mask $\boldsymbol{M}_{\boldsymbol{I}_D}$ of the foreground document. Then, the background-excluded image $\boldsymbol{I}_d$ is fed into the progressive rectification module, which progressively corrects the geometric distortion in an iterative manner. It maintains a single estimate of the warping flow that is refined at each iteration, and output the rectified image by warping the image $\boldsymbol{I}_D$ with the output warping flow $\boldsymbol{f}^K$ at the last iteration.

as a post-processing to refine the predicted forward warping flow on boundary regions of the document. To learn a powerful representation for the document image, DocTr (Feng et al., 2021) first introduces the self-attention mechanism (Vaswani et al., 2017) from the natural language processing tasks to the filed. To improve the running efficiency, DDCP (Xie et al., 2021) only estimates several pairs of control points to conduct rectification. PWUNet (Das et al., 2021) concentrates on the distinct distortion of local regions for improved global rectification. DocGeoNet (Feng et al., 2022) extracts global and local geometric representations to improve rectification, by the prediction of 3D shape and textlines. To extract the structural information of a deformed document, FDRNet (Xue et al., 2022) focuses on high-frequency components in the Fourier space to improve restoration. Marior (Zhang et al., 2022) considers the rectification of the document images with large background regions and gradually rectifies them to a robust state. RDGR (Jiang et al., 2022) first detect textlines and boundaries in a document image, and then perform the rectification by solving an optimization problem with the proposed grid regularization. To improve the generalization ability of the network, PaperEdge (Ma et al., 2022) involves the real-world document images in the training.

Although the field of document image rectification has witnessed rapid progress in recent years, the rectified results of such advanced methods still remain unsolved distortions and are unsatisfactory. In this work, we propose DocScanner, a new deep architecture for document image rectification, aiming to achieve an accurate and robust distortion rectification.

3 METHODOLOGY

In this section, we present our design of DocScanner to facilitate the geometric correction of distorted document images. As shown in Fig. 1, DocScanner consists of a document localization module and a progressive rectification module. Given a distorted document image $\boldsymbol{I}_D$, the document localization module estimates a foreground mask $\boldsymbol{M}_{\boldsymbol{I}_D}$ to exclude the background. Then, the image with only foreground document $\boldsymbol{I}_d$ is fed into the progressive rectification module, which maintains a single estimate of warping flow and refines it across K iterations. The final output warping flow $\boldsymbol{f}^K$ is used to rectify the input image $\boldsymbol{I}_D$. Additionally, to further relieve the distortion of rectified images, we propose a regularization loss to regularize the training of the progressive rectification module. In the following, we elaborate the key components of DocScanner, including the document localization module, the progressive rectification module, and the training strategy.

3.1 Document Localization Module

The goal of the document localization module is to remove the noisy background. It makes the subsequent rectification network focuses on geometric rectification, without extra learning on localizing the document. Following prior work (Xie et al., 2021; Feng et al., 2021; Zhang et al., 2022), we formulate the foreground document segmentation as a saliency detection problem, and address it with a nested U-structure network (Qin et al., 2020). As shown in Fig. 1, given a distorted document image $\boldsymbol{I}_D \in \mathbb{R}^{H \times W \times 3}$, where H and W are the

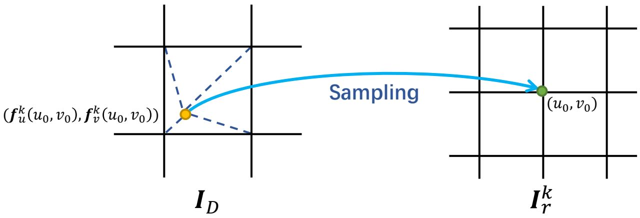


Fig. 2: Visualization of the rectification process of a certain pixel on the rectified image based on the warping flow. $\boldsymbol{I}_D$ and $\boldsymbol{I}_r^k$ are the input distorted image and output rectified image, respectively.

height and width of the image, we predict a confidence map of the foreground document. This map is further binarized with a threshold τ to obtain the binary document region mask $\boldsymbol{M}_{\boldsymbol{I}_D}$. Then, the background of $\boldsymbol{I}_D$ can be removed by element-wise matrix multiplication with broadcasting along the channels of $\boldsymbol{I}_D$. It should be noted that this module can also be replaced with other alternative segmentation networks. The document localization module is trained with a binary cross-entropy loss (De Boer et al., 2005) as follows,

$$\mathcal{L}_{seg} = -\sum_{i=1}^{N_p} \left[y_i \log(\hat{p_i}) + (1-y_i)\log(1-\hat{p_i}) \right], \tag{1}$$

where N_p is the number of the pixels of the distorted image $\boldsymbol{I}_D$, $y_i \in \{0,1\}$ and $\hat{p_i} \in [0,1]$ denote the ground-truth and the predicted confidence, respectively.

3.2 Progressive Rectification Module

Given the background-excluded image, the progressive rectification module progressively corrects it toward a distortion-free one. Specifically, we design a compact recurrent architecture to refine the rectification result estimated at the previous iteration. Through iterative refinements, the distortions in the input distorted document images are progressively corrected and finally converge to a relatively steady and accurate status.

As shown in Fig. 1, given the background-excluded image $\boldsymbol{I}_d \in \mathbb{R}^{H\times W\times 3}$ obtained by the document localization module, we estimate the warping flow iteratively and get the sequence $\{\boldsymbol{f}^1, \cdots, \boldsymbol{f}^K\}$, where $\boldsymbol{f}^k = (\boldsymbol{f}_u^k, \boldsymbol{f}_v^k)$ is the predicted warping flow at the k^{th} iteration, and K is the total iteration number. Note that the two channel of the warping flow $\boldsymbol{f}^k \in \mathbb{R}^{H\times W\times 2}$ denote the horizontal and the vertical coordinate mapping (*i.e.*, $\boldsymbol{f}_u^k$ and $\boldsymbol{f}_v^k$), respectively. With $\boldsymbol{f}^k$ predicted at the k^{th} iteration, as illustrated in Fig. 2, the rectified image $\boldsymbol{I}_r^k$ can be obtained by the warping operation based on the bilinear sampling as follows,

$$\boldsymbol{I}_r^k(u_0,v_0) = \boldsymbol{I}_D(\boldsymbol{f}_u^k(u_0,v_0), \boldsymbol{f}_v^k(u_0,v_0)), \tag{2}$$

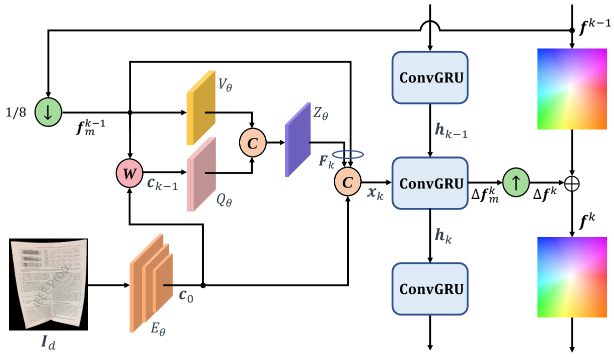


Fig. 3: Illustration of the warping flow estimation at the k^{th} iteration. Given the distorted features $\boldsymbol{c}_0$ and predicted warping flow $\boldsymbol{f}^{k-1}$, it outputs the current warping flow $\boldsymbol{f}^k$. $\boldsymbol{W}$ represents the bilinear sampling operation of warping. "$\boldsymbol{C}$" and "+" denote concatenation over channel and element-wise addition, respectively. "↓" and "↑" denote the bilinear downsampling and the learnable upsampling module, respectively.

where (u_0, v_0) is the integer pixel coordinate in rectified image, and $(\boldsymbol{f}_u^k(u_0,v_0), \boldsymbol{f}_v^k(u_0,v_0))$ is the predicted decimal pixel coordinate in distorted image.

For convenience of understanding, we divide the progressive rectification module into three blocks, including (1) distorted feature encoder, (2) rectified feature generator, and (3) warping flow updater. In the following, we separately detail the three blocks.

Distorted feature encoder. Given the input image $\boldsymbol{I}_d \in \mathbb{R}^{H\times W\times 3}$, we use a convolutional network E_θ to extract features from distorted image $\boldsymbol{I}_d$. E_θ consists of 6 residual blocks (He et al., 2016) and stride the feature maps every two blocks, followed by two parallel convolutional layers. The two parallel layers produce features $\boldsymbol{c}_0 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times D}$ and $\boldsymbol{h}_0 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times D}$, respectively, where we set channel dimension $D = 128$. $\boldsymbol{c}_0$ denotes the distorted features, and $\boldsymbol{h}_0$ serves as the initial hidden state for warping flow updater. Note that both $\boldsymbol{c}_0$ and $\boldsymbol{h}_0$ need to be calculated only once.

Rectified feature generator. As shown in Fig. 3, we take the k^{th} iteration as an example for illustration. Given the distorted features $\boldsymbol{c}_0$ from the distorted feature encoder and the warping flow $\boldsymbol{f}^{k-1}$ predicted at the $(k-1)^{th}$ iteration, we first downsample $\boldsymbol{f}^{k-1}$ and get the warping flow $\boldsymbol{f}_m^{k-1} = (\boldsymbol{f}_{mu}^{k-1}, \boldsymbol{f}_{mv}^{k-1})$ at 1/8 resolution. Then, we unwarp the feature maps $\boldsymbol{c}_0$ toward rectified domain using predicted $\boldsymbol{f}_m^{k-1}$ based on bilinear sampling, and obtain features $\boldsymbol{c}_{k-1}$ as follows,

$$\boldsymbol{c}_{k-1}(x,y) = \boldsymbol{c}_0(\boldsymbol{f}_{mu}^{k-1}(x,y), \boldsymbol{f}_{mv}^{k-1}(x,y)), \tag{3}$$

where (x,y) is the integer pixel coordinate in $\boldsymbol{c}_{k-1}$, and $(\boldsymbol{f}_{mu}^{k-1}(x,y), \boldsymbol{f}_{mv}^{k-1}(x,y))$ is the predicted decimal pixel

Table 1: The structure of the distorted feature encoder and rectified feature generator in DocScanner-B.

Layer	Output size	Operation
E_θ	$256 \times 36 \times 36$	$7 \times 7, 64, stride\ 2$ $3 \times 3, 64, stride\ 1$ $3 \times 3, 64, stride\ 1$ $3 \times 3, 96, stride\ 1$ $3 \times 3, 96, stride\ 2$ $3 \times 3, 128, stride\ 1$ $3 \times 3, 128, stride\ 2$ $1 \times 1, 256, stride\ 1$
V_θ	$64 \times 36 \times 36$	$7 \times 7, 128, stride\ 1$ $3 \times 3, 64, stride\ 1$
Q_θ	$192 \times 36 \times 36$	$1 \times 1, 224, stride\ 1$ $3 \times 3, 192, stride\ 1$
Z_θ	$128 \times 36 \times 36$	$3 \times 3, 128, stride\ 1$

coordinate in $\boldsymbol{c}_0$. Note that the initial warping flow $\boldsymbol{f}^0 \in \mathbb{R}^{H \times W \times 2}$ is initialized as the coordinate map of the pixels in $\boldsymbol{I}_d$. In addition, the warping operation is implemented based on bilinear interpolation. Therefore, we can compute the gradients to the input feature map $\boldsymbol{c}_0$ and warping flow $\boldsymbol{f}_m^{k-1}$ for backpropagation, according to classic STN (Jaderberg et al., 2015), and the module can be trained in an end-to-end manner. Then the warped feature map $\boldsymbol{c}_{k-1}$ is processed by a convolutional module Q_θ which consists of two convolutional layers, and produce features $Q_\theta(\boldsymbol{c}_{k-1}) \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D_q}$, where we set $D_q = 192$.

Additionally, another convolutional module V_θ that consists of two convolutional layers is used to extract features from the predicted warping flow $\boldsymbol{f}_m^{k-1}$, and output features $V_\theta(\boldsymbol{f}_m^{k-1}) \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D_v}$, where we set $D_v = 64$. Then, we concatenate $Q_\theta(\boldsymbol{c}_{k-1})$ and $V_\theta(\boldsymbol{f}_m^{k-1})$ along the channel dimension into a single feature map, which is fused by a following convolutional layer Z_θ. Finally, we concatenate the output features and the downsampled warping flow $\boldsymbol{f}_m^{k-1}$ to generate the rectified feature map $\boldsymbol{F}_k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$. It carries the content and the structural information of the current rectified image estimated at the previous iteration, which is differentiated and processed by the following updater to estimate a further refinement.

Warping flow updater. As shown in Fig. 3, we concatenate distorted features $\boldsymbol{c}_0$ and the rectified features $\boldsymbol{F}_k$ along the channel dimension into a single feature map $\boldsymbol{x}_k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 2D}$, which serves as the input of the recurrent unit at the k^{th} iteration. We use a convolution-based gated recurrent unit (GRU) as the recurrent unit as many other tasks (Tokmakov et al., 2017; Teed and Deng, 2020; Zhou et al., 2021). As shown in Fig. 4, it is a variant of GRU (Cho et al., 2014), in which the fully connected layers are replaced by the convolutional layers. For the k^{th} iteration, it processes the input features $\boldsymbol{x}_k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 2D}$ and the hidden state $\boldsymbol{h}_{k-1} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$, and outputs the hidden states $\boldsymbol{h}_k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$ as follows,

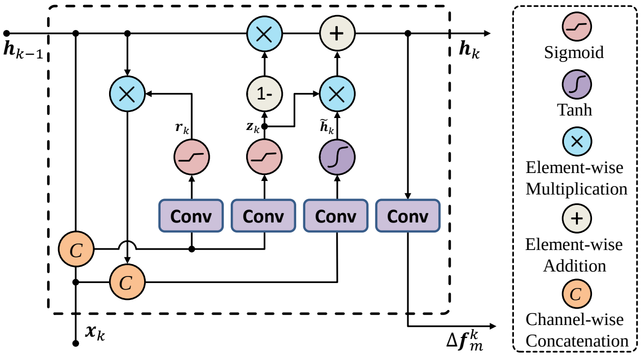


Fig. 4: Inner structure of the ConvGRU, a modified version of GRU (Cho et al., 2014).

$$\boldsymbol{z}_k = \sigma(Conv_{3\times3}([\boldsymbol{h}_{k-1}, \boldsymbol{x}_k], \boldsymbol{W}_z)), \tag{4}$$

$$\boldsymbol{r}_k = \sigma(Conv_{3\times3}([\boldsymbol{h}_{k-1}, \boldsymbol{x}_k], \boldsymbol{W}_r)), \tag{5}$$

$$\widetilde{\boldsymbol{h}}_k = tanh(Conv_{3\times3}([\boldsymbol{r}_k \odot \boldsymbol{h}_{k-1}, \boldsymbol{x}_k], \boldsymbol{W}_h)), \tag{6}$$

$$\boldsymbol{h}_k = (1 - \boldsymbol{z}_k) \odot \boldsymbol{h}_{k-1} + \boldsymbol{z}_k \odot \widetilde{\boldsymbol{h}}_k, \tag{7}$$

where σ and $\odot$ represent Sigmoid function and element-wise multiplication operation, respectively. Followed by $\boldsymbol{h}_k$ is two convolutional layers that produce the residual displacement $\Delta\boldsymbol{f}_m^k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 2}$.

To upsample the obtained 1/8 scale $\Delta\boldsymbol{f}_m^k$ to full resolution $(H \times W)$, we introduce a learnable upsampling module (Feng et al., 2021). Specifically, we first exploit two convolutional layers (stride 1) to process the hidden state $\boldsymbol{h}_k \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$, and reshape the output to a $\frac{H}{8} \times \frac{W}{8} \times 8 \times 8 \times 9$ map. Then, we perform softmax on the last dimension of it and get the weight matrix. Next, using the obtained weight matrix, we take a weighted combination over the 3×3 neighborhood of each pixel in $\Delta\boldsymbol{f}_m^k$. Finally, the obtained $\frac{H}{8} \times \frac{W}{8} \times 8 \times 8 \times 2$ map is permuted and reshaped to the full resolution residual displacement map $\Delta\boldsymbol{f}^k \in \mathbb{R}^{H \times W \times 2}$.

After that, $\Delta\boldsymbol{f}^k$ is used to update the current warping flow $\boldsymbol{f}^k$ as follows,

$$\boldsymbol{f}^k = \boldsymbol{f}^{k-1} + \Delta\boldsymbol{f}^k. \tag{8}$$

As shown in Fig. 1, after K iterations, based on Equation (2), we obtain the rectified image $\boldsymbol{I}_r^K$ by warping the distorted image $\boldsymbol{I}_D$ with the final predicted $\boldsymbol{f}^K$.

3.3 Training Loss Function

During the training of the progressive rectification module, the loss is calculated over all K iterations as follows,

$$\mathcal{L} = \sum_{k=1}^{K} \lambda^{K-k} \mathcal{L}^{(k)}, \tag{9}$$

where λ^{K-k} is the weight of the k^{th} iteration which increases exponentially ($\lambda < 1$). At the k^{th} iteration, the loss is defined as the weighted summation of the warping flow regression loss $\mathcal{L}_f^{(k)}$ and the proposed circle-consistency loss $\mathcal{L}_{line}^{(k)}$ as follows,

$$\mathcal{L}^{(k)} = \mathcal{L}_f^{(k)} + \alpha \mathcal{L}_{line}^{(k)}, \tag{10}$$

where α is a constant weighting factor. $\mathcal{L}_f^{(k)}$ is defined as the L_1 distance between the predicted warping flow $\boldsymbol{f}^k$ and its given ground truth $\boldsymbol{f}_{gt}$ as follows,

$$\mathcal{L}_f^{(k)} = \left\| \boldsymbol{f}_{gt} - \boldsymbol{f}^k \right\|_1 . \tag{11}$$

The proposed circle-consistency loss $\mathcal{L}_{line}^{(k)}$ works as a regularizer, which imposes straight-line constraint along rows and columns in rectified image. We detail it in the following.

3.3.1 Circle-consistency Loss

Equation (2) shows that during the rectification process, the pixel in rectified image $\boldsymbol{I}_r^k$ is filled with the corresponding pixel sampled in distorted image $\boldsymbol{I}_D$ based on the predicted warping flow $\boldsymbol{f}^k$. This predicted warping flow $\boldsymbol{f}^k$ is termed as backward warping flow. We further introduce forward warping flow $\boldsymbol{g} = (\boldsymbol{g}_x, \boldsymbol{g}_y)$ from the dataset, which maps the pixel (x_0, y_0) in distorted image $\boldsymbol{I}_D$ to pixel $(\boldsymbol{g}_x(x_0, y_0), \boldsymbol{g}_y(x_0, y_0))$ in rectified image $\boldsymbol{I}_r^k$ as follows,

$$\boldsymbol{I}_r^k(\boldsymbol{g}_x(x_0, y_0), \boldsymbol{g}_y(x_0, y_0)) = \boldsymbol{I}_D(x_0, y_0), \tag{12}$$

where (x_0, y_0) is the integer pixel coordinate in the distorted image $\boldsymbol{I}_D$, while $(\boldsymbol{g}_x(x_0, y_0), \boldsymbol{g}_y(x_0, y_0))$ is the corresponding decimal pixel coordinate in the rectified image $\boldsymbol{I}_r^k$.

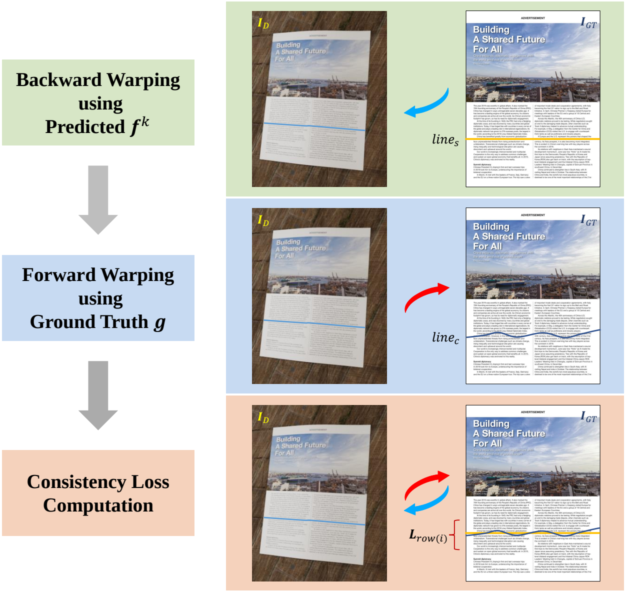

Backward Warping using Predicted $\boldsymbol{f}^k$
Forward Warping using Ground Truth $\boldsymbol{g}$
Consistency Loss Computation
$\boldsymbol{I}_D$
$\boldsymbol{I}_{GT}$
Building A Shared Future For All
$line_s$
$line_c$
$\mathcal{L}_{row(i)}$

Fig. 5: Illustration of the circle-consistency loss. After warping a line pixels $line_s$ using the predicted backward warping flow $\boldsymbol{f}^k$ and ground truth forward warping flow $\boldsymbol{g}$, the output line $line_c$ should be consistent with itself $line_s$ under perfect prediction. Based on this observation, the circle-consistency loss is defined by computing the distortion of $line_c$.

We propose circle-consistency loss, based on the circle-consistency introduced by the backward warping and the forward warping operations. It consists of two terms, along the row and column direction respectively. As shown in Fig. 5, we take the row distortion term as an example. Specifically, we first map the pixels of i^{th} row (*i.e.*, $line_s$) in ground truth document image to $\boldsymbol{I}_D$, based on the predicted backward warping flow $\boldsymbol{f^k}$. Secondly, we map these pixels back to the ground truth document image again, using the ground truth forward warping flow $\boldsymbol{g}$. After the above two steps, we get a curved line $line_c$, which shall be the straight line $line_s$ when the backward warping flow in the first step is perfectly estimated. Hence, we define the i^{th} row circle-consistency loss $\mathcal{L}_{row(i)}$ as the deviation of row coordinate of the estimated curved line $line_c$ as follows,

$$\mathcal{L}_{row(i)} = \frac{1}{W} \sum_{k=1}^{W} \| x(i,k) - \overline{x_i} \|_2^2, \tag{13}$$

where $\overline{x_i}$ denotes the averaged row coordinate of the curved line $line_c$, and $x(i,k)$ denotes the row coordinate of the k^{th} pixel of $line_c$. $\mathcal{L}_{row(i)}$ measures the distortion of the i^{th} row which should be zero in case of perfect rectification.

Similarly, we can calculate the column circle-consistency term $\mathcal{L}_{col(j)}$ for the j^{th} column. Then, the total circle-consistency loss $\mathcal{L}_{line}$ is calculated over all rows and columns as follows,

$$\mathcal{L}_{line} = \frac{1}{W} \sum_{i=1}^{W} \mathcal{L}_{row(i)} + \frac{1}{H} \sum_{j=1}^{H} \mathcal{L}_{col(j)}, \tag{14}$$

where (H, W) is the shape of the predicted warping flow $\boldsymbol{f}^k$.

4 Experiments

4.1 Datasets

We train our DocScanner on the Doc3D dataset (Das et al., 2019) and evaluate it on the DocUNet Benchmark dataset (Ma et al., 2018). In the following, we elaborate the two datasets respectively.

Doc3D. Doc3D dataset (Das et al., 2019) is the largest dataset to date for document image rectification. It is created by the real document data and rendering software, *i.e.*, Blender[1]. The dataset consists of 100k distorted document images. For each distorted image, there are corresponding ground truth 3D coordinate map, albedo map, normals map, depth map, forward warping flow map, and backward warping flow map.

DocUNet Benchmark. The challenging DocUNet Benchmark dataset (Ma et al., 2018) is a widely-used dataset for document image rectification. It comprises 130 photos of real paper documents captured by mobile cameras. The documents include various types such as receipts, letters, fliers, magazines, academic papers, and books, *etc.* Besides, their distortion and background are various to cover different levels of difficulty.

Notably, we observe that the 127^{th} and 128^{th} distorted document images are rotated by 180 degrees, which do not match the ground truth documents. This inconsistency is ignored by existing methods (Ma et al., 2018; Das et al., 2019; Liu et al., 2020; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022). In our experiments, we use the corrected dataset.

4.2 Evaluation Metrics

We use three evaluation schemes to quantitatively evaluate the performance of DocScanner in terms of (a) rectification distortion, (b) Optical Character Recognition (OCR) accuracy, and (c) image similarity. Firstly, for rectification distortion, we use Local Distortion (LD) (You et al., 2018) as recommended in (Ma et al., 2018; Das et al., 2019; Liu et al., 2020; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022). Moreover, we propose a new metric, namely Line Distortion (Li-D), to further evaluate the global distortion of the rectified document images. Secondly, for OCR accuracy, we choose Edit Distance (ED) (Levenshtein, 1966) and Character Error Rate (CER) (Morris et al., 2004) to evaluate the utility of our method on text recognition, following (Ma et al., 2018; Das et al., 2019, 2021; Feng et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022). Thirdly, for image similarity, we use Multi-Scale Structural Similarity (MS-SSIM) (Wang et al., 2003) as previous works (Ma et al., 2018; Das et al., 2019; Liu et al., 2020; Das et al., 2020; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022) suggest.

Local Distortion. Local distortion (LD) (You et al., 2018) first registers the rectified image with the ground truth one using a dense SIFT-flow (Liu et al., 2011) $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y})$, where $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta y}$ denote the horizontal and vertical displacement map of the matched pixels from the ground truth image to the rectified one, respectively. Then, LD is calculated as the mean value of the L_2 distance among all matched pixels, which measures the average local deformation of the rectified image. Note that, for a fair comparison, all the rectified images and the ground truth images are resized to a 598,400-pixel area, as suggested in (Ma et al., 2018; Das et al., 2019; Liu et al., 2020; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022).

Line Distortion. We propose Line Distortion (Li-D) as a supplementary metric to further evaluate the global distortion of the rectified images. Specifically, the dense SIFT-flow (Liu et al., 2011) $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y})$ from the ground truth scanned image to the rectified one is first computed. Then, we calculate the standard deviation of all column vectors in the $\boldsymbol{\Delta x}$ and all row vectors in the $\boldsymbol{\Delta y}$, which measure the deformation of a certain rectified row and column pixels, respectively. Finally, we take the mean of all the standard deviation values to obtain the overall Line Distortion (Li-D) value.

Compared to the typical metric Local Distortion (LD) (You et al., 2018), the proposed Line Distortion (Li-D) computes the average deformation of the row and column pixels. In another word, Li-D focuses more on the global distortions. The less distortion of the rectified image, the lower the value. Note that if there is only global misalignment (i.e. scaling and translation) between two images, the Li-D should be 0 but such global misalignments are not considered for this task.

ED and CER. Edit Distance (ED) (Levenshtein, 1966) quantifies how dissimilar two strings are to one another. It is defined based on the minimum number of operations required to transform one string into the reference one, which can be efficiently calculated using the dynamic programming algorithm. Specifically, the involved operations include deletions (d), insertions (i), and substitutions (s). Then, Character Error Rate

[1] https://www.blender.org/

Table 2: Quantitative comparisons of the existing learning-based rectification methods in terms of image similarity, distortion metrics, OCR performance, and running efficiency on the DocUNet Benchmark dataset (Ma et al., 2018). "↑" indicates the higher the better and "↓" means the opposite.

Methods	**Venue**	**MS-SSIM ↑**	**LD ↓**	**Li-D ↓**	**ED ↓**	**CER ↓**	**FPS ↑**	**Para.(M)**
Distorted	-	0.2459	20.51	5.66	2111.56/1552.22	0.5352/0.5089	-	-
DocUNet (Ma et al., 2018)	*CVPR'18*	0.4103	14.19	3.19	1933.66/1259.83	0.4632/0.3966		58.6
AGUN (Liu et al., 2020)	*PR'18*	-	-	-	-	-	-	-
DocProj (Li et al., 2019)	*TOG'19*	0.2946	18.01	5.00	1712.48/1165.93	0.4267/0.3818	-	47.8
FCN-based (Xie et al., 2020)	*DAS'20*	0.4477	7.84	2.04	1792.60/1031.40	0.4213/0.3156	1.49	23.6
DewarpNet (Das et al., 2019)	*ICCV'19*	0.4735	8.39	2.31	885.90/525.45	0.2373/0.2102	8.17	86.9
PWUNet (Das et al., 2021)	*ICCV'21*	0.4915	8.64	2.34	1069.28/743.32	0.2677/0.2623	-	-
DocTr (Feng et al., 2021)	*MM'21*	0.5105	7.76	2.11	724.84/464.83	0.1832/0.1746	7.74	26.9
DDCP (Xie et al., 2021)	*ICDAR'21*	0.4729	8.99	2.20	1442.84/745.35	0.3633/0.2626	**14.09**	13.3
DocGeoNet (Feng et al., 2022)	*ECCV'22*	0.5040	7.71	2.22	713.94/**379.00**	0.1821/**0.1509**	8.57	24.8
Marior (Zhang et al., 2022)	MM'22	0.4780	**7.44**	2.03	776.22/593.80	0.1928/0.2136	-	-
RDGR (Jiang et al., 2022)	*CVPR'22*	0.4968	8.51	2.12	729.52/420.25	**0.1717**/0.1559	-	-
PaperEdge (Ma et al., 2022)	*SIGGRAPH'22*	0.4724	7.99	**1.83**	777.76/**375.60**	0.2014/0.1541	**13.95**	36.6
DocScanner-T	-	0.5123	7.92	2.04	809.46/501.82	0.2068/0.1823	10.81	**2.6**
DocScanner-B	-	**0.5134**	7.62	1.88	**671.48**/434.11	0.1789/0.1652	10.03	**5.2**
DocScanner-L	-	**0.5178**	**7.45**	**1.86**	**632.34**/390.43	**0.1648**/**0.1486**	9.52	8.5

(CER) can be computed as follows,

$$CER = (d + i + s)/N_c, \tag{15}$$

where N_c is the character number of the reference string. It represents the percentage of characters in the reference text that was incorrectly recognized in the distorted image. The lower the CER value (with 0 being a perfect score), the better the performance of the rectification method. We use Tesseract (v5.0.1) (Smith, 2007) as the OCR engine to recognize the text string of the rectified image and the ground truth image, as recommended in previous works (Ma et al., 2018; Das et al., 2019; Liu et al., 2020; Das et al., 2020; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022).

MS-SSIM. The Structural SIMilarity (SSIM) (Zhou Wang et al., 2004) measures the similarity of mean value and variance within each image patch between two images. Considering that the perceivability of image details depends on the sampling density of the image, Multi-Scale Structural Similarity (MS-SSIM) (Wang et al., 2003) builds a Gaussian pyramid for the rectified image and the corresponding ground truth image, respectively. Then, MS-SSIM is calculated as the weighted summation of SSIM (Zhou Wang et al., 2004) across multiple scales. Specifically, all the rectified and ground truth flatbed-scanned images are first resized to a 598,400-pixel area, as recommended in DocUNet (Ma et al., 2018). Then, we build a 5-level-pyramid for MS-SSIM and the weight for each level is set as 0.0448, 0.2856, 0.3001, 0.2363, 0.1333, which is inherited from the original implementation of MS-SSIM (Wang et al., 2003).

4.3 Training Details

The whole framework of DocScanner is implemented in Pytorch (Paszke et al., 2017). We train the document localization module and progressive rectification module independently on the Doc3D dataset (Das et al., 2019). We detail their training in the following.

Document localization module. During training, to generalize well to real data with complex background environments, we randomly replace the background of the distorted image with the texture images from Describable Texture Dataset (DTD) (Cimpoi et al., 2014). We use Adam optimizer (Kingma and Ba, 2014) with a batch size of 32. The initial learning rate is set as 1×10^{-4}, and reduced by a factor of 0.1 after 30 epochs. After 45 epochs, the training loss converges. The training is conducted on two NVIDIA RTX 2080 Ti GPUs. The threshold τ for binarizing the confidence map in Sec. 3.1 is empirically set as 0.5.

Progressive rectification module. During training, we remove the background of distorted images using the ground truth masks of the foreground document regions. In other words, the documents are within a clean background. To generalize well to real data with complex illumination conditions, we then add a jitter in the HSV color space to magnify illumination and document color variations. We use AdamW optimizer (Loshchilov and Hutter, 2017) with a batch size of 12. The total training iteration is set as 560k, and the learning rate reaches the maximum 1×10^{-4} after 27k iterations for learning rate warm-up. We set the hyperparameters $K = 12, \lambda = 0.85$ (in Equation (9)), $\alpha = 0.5$ (in Equation (10)). Experiments are all performed on a single NVIDIA GTX 1080 Ti GPU.

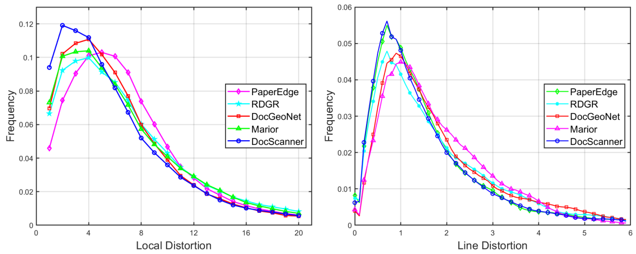


Fig. 6: Comparisons of the distortion distribution curve of DocScanner-L with the state-of-the-art methods (Ma et al., 2022; Jiang et al., 2022; Feng et al., 2022; Zhang et al., 2022). The x-coordinate denotes the distortion extent, and the y-coordinate shows their frequency distribution among the total DocUNet Benchmark dataset (Ma et al., 2018).

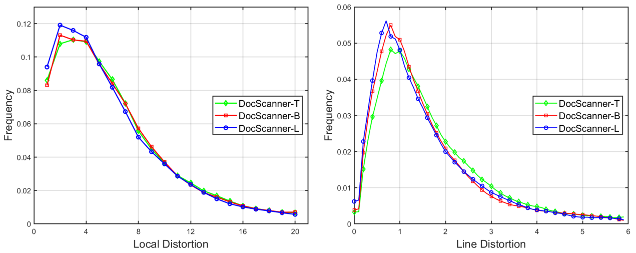


Fig. 7: Comparisons of the distortion distribution curve of the three varieties of DocScanner, *i.e.*, DocScanner-T, DocScanner-B, and DocScanner-L. The x-coordinate denotes the distortion extent, and the y-coordinate shows their frequency distribution among the total DocUNet Benchmark dataset (Ma et al., 2018).

4.4 Experimental Results

We evaluate the performance of DocScanner on the DocUNet Benchmark dataset (Ma et al., 2018) by quantitative and qualitative evaluation. Table 2 shows the comparisons of our method with the existing learning-based methods on image similarity, distortion metrics, OCR accuracy, and inference efficiency. Note that for OCR accuracy evaluation, following DewarpNet (Das et al., 2019) and DocTr (Feng et al., 2021), we select 50 and 60 images from the DocUNet Benchmark dataset (Ma et al., 2018) respectively, where the text makes up the majority of content. This is because if the text is rare in an image, the character number N_c (numerator) in Equation (15) is a small number, leading to a large variance for CER.

For DocUNet (Ma et al., 2018), DewarpNet (Das et al., 2019), FCN-based (Xie et al., 2020), DocTr (Feng et al., 2021), PWUNet (Das et al., 2021), Marior (Zhang et al., 2022), RDGR (Jiang et al., 2022), DocGeoNet (Feng et al., 2022), and PaperEdge (Ma et al., 2022), we obtain the results based on the rectified document images of DocUNet Benchmark dataset (Ma et al., 2018) from the authors or the public results. For AGUN (Liu et al., 2020), there is no public official code. Due to the two problematic samples in the DocUNet Benchmark dataset (Ma et al., 2018), we can not obtain the performance. For DocProj (Li et al., 2019) and DDCP (Xie et al., 2021), we report the results based on the official code and their public pre-trained models.

Comparison with state-of-the-art methods. As shown in Table 2, DocScanner sets several state-of-the-art records on DocUNet Benchmark dataset (Ma et al., 2018). Here we build three varieties of our DocScanner with different model sizes (*i.e.*, DocScanner-T, DocScanner-B, and DocScanner-L). Note that different from other methods, DocProj (Li et al., 2019) is a patch-based method that predicts the distortion flow on document patches rather than the entire image. Therefore, the rectified boundaries are still distorted due to the uncropped distorted images in the DocUNet Benchmark dataset (Ma et al., 2018), leading to a limited performance on distortion metrics. Compared with the classic DewarpNet (Das et al., 2019), DocScanner-B achieves a relative improvement on MS-SSIM by 8.43%, Li-D by 18.61%, and CER by 24.61%/21.41%, respectively, with only 1/16 parameters. Moreover, compared with DocTr (Feng et al., 2021) based on the powerful transformer (Vaswani et al., 2017), our larger DocScanner-L shows a relative improvement on Li-D by 11.85% and CER by 10.04%/14.89%, with 1/3 parameters. Compared with the recent state-of-the-art method Marior (Zhang et al., 2022) and PaperEdge (Ma et al., 2022), DocScanner-L yields a sizeable improvement on metric MS-SSIM and LD. Such lower distortion and superior OCR performance demonstrate that DocScanner can effectively restore both the structure and content of distorted document images.

As shown in Fig. 6, we compare the distortion frequency distribution curves of DocScanner with the state-of-the-art methods (Ma et al., 2022; Jiang et al., 2022; Feng et al., 2022; Zhang et al., 2022). Specifically, for the Local Distortion distribution curve (left), the x-coordinate denotes the L_2 distance of the matched pixels between the rectified image and the GT image, while the y-coordinate denotes their frequency distribution among the total DocUNet Benchmark dataset (Ma et al., 2018). We can see that, for DocScanner the pixels with small deformations take up the majority of the rectified images and the pixels with large deformations have a smaller proportion, compared to other methods. In another word, the rectified images of DocScanner have smaller local deformations. In addition, for

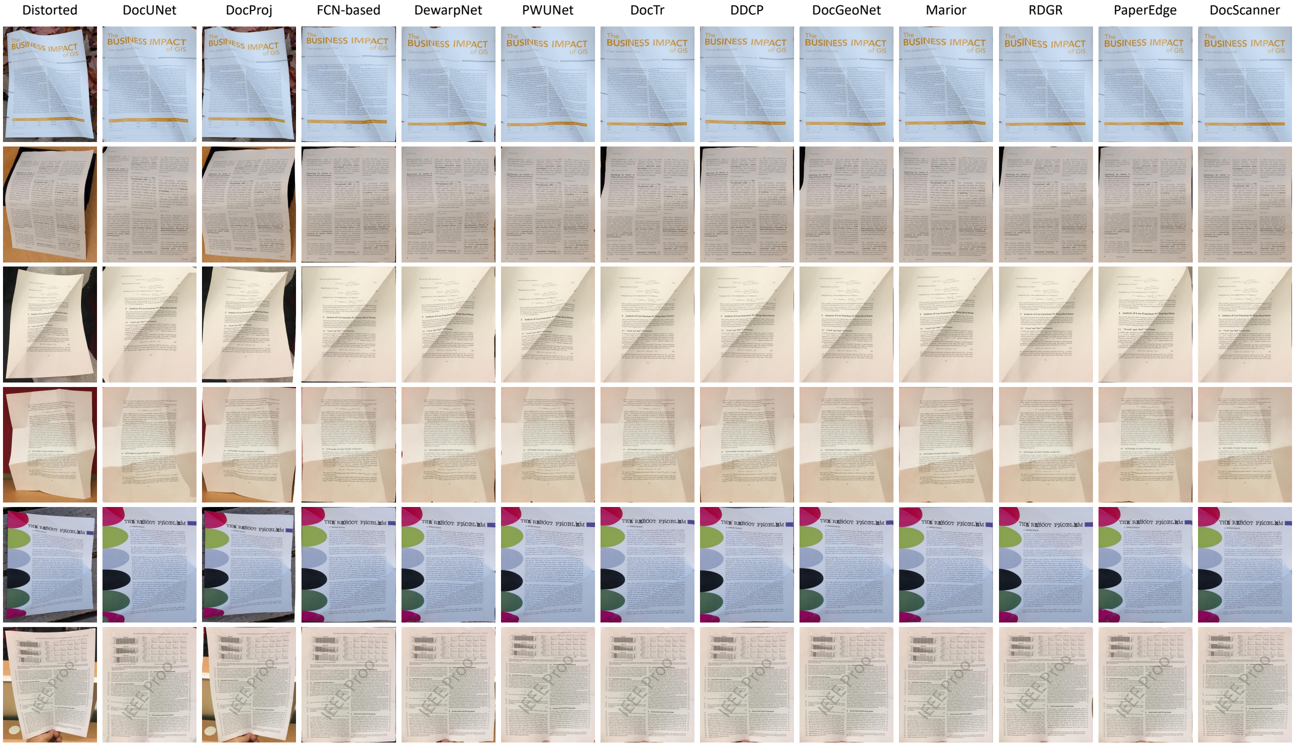


Fig. 8: Qualitative comparisons with existing learning-based methods, including DocUNet (Ma et al., 2018), DocProj (Li et al., 2019), FCN-based (Xie et al., 2020), DewarpNet (Das et al., 2019), PWUNet (Das et al., 2021), DocTr (Feng et al., 2021), DDCP (Xie et al., 2021), DocGeoNet (Feng et al., 2022), Marior (Zhang et al., 2022), RDGR (Jiang et al., 2022), and PaperEdge (Ma et al., 2022). The rectified images of DocScanner show less distortions than the other rectification methods. Zoom in for the best view.

the Line Distortion distribution curve (right), the x-coordinate denotes the standard deviation of the rectified row and column pixels. Similarly, the y-coordinate denotes their frequency distribution among the total DocUNet Benchmark dataset (Ma et al., 2018). The obtained curve (right) presents similar distributions to the Local Distortion distribution curve (left), which demonstrates that the rectified images of DocScanner have smaller global deformations. Such results show the superior rectification performance of DocScanner over the state-of-the-art methods.

In Fig. 7, we compare the distortion frequency distribution curves of DocScanner-T, DocScanner-B, and DocScanner-L. As we can see, DocScanner-L reveals smaller local and global deformations, compared with DocScanner-T and DocScanner-B.

To better demonstrate the effectiveness of our proposed DocScanner, we further conduct qualitative comparisons with existing methods (Ma et al., 2018; Li et al., 2019; Das et al., 2019; Xie et al., 2020; Das et al., 2021; Feng et al., 2021; Xie et al., 2021; Feng et al., 2022; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022). Concretely, as shown in Fig. 8, we first compare the rectified images. The results reveal that the rectified images of our DocScanner show less distortions than the other rectification methods. Second, as shown in Fig. 9, we randomly crop some local patches to compare the local rectification details. We can see that the rectified textlines of DocScanner are much straighter than other rectification methods. Such outstanding visual performances agree with the above quantitative results.

Table 3: Running time of processing a 1080P image and parameter count of the document localization module and the progressive rectification module.

Module of DocScanner-B	Time (s)	Parameters (M)
document localization	0.014	1.13
progressive rectification	0.085	4.10
total	0.099	5.23

Efficiency comparison. As shown in Table 2, we also conduct efficiency comparisons, on the running time of processing a 1080P resolution image and the network parameter numbers. The evaluation is performed on a single RTX 2080Ti GPU. Note that we only compare the methods with the released codes and the computed FPS does not involve the time spent on reading images.

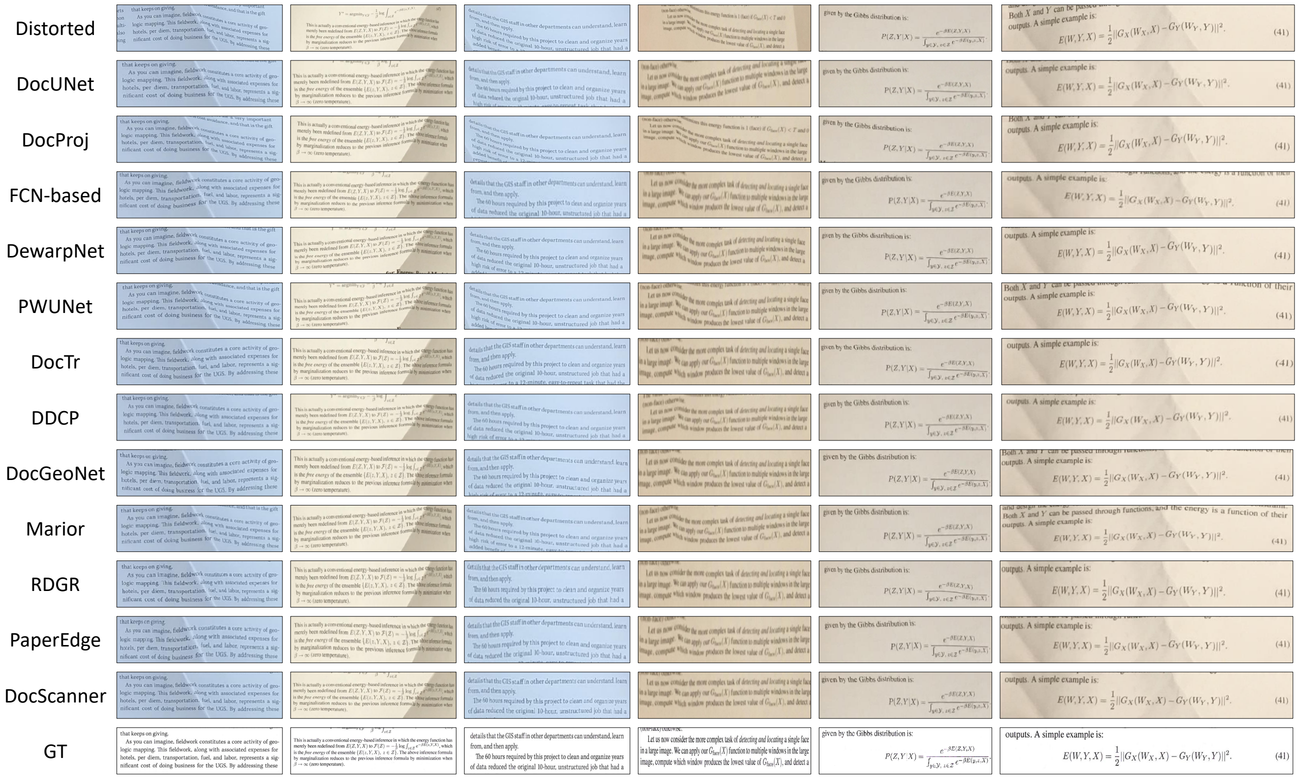


Fig. 9: Qualitative comparisons of the local rectified textlines with existing learning-based methods, including DocUNet (Ma et al., 2018), DocProj (Li et al., 2019), FCN-based (Xie et al., 2020), DewarpNet (Das et al., 2019), PWUNet (Das et al., 2021), DocTr (Feng et al., 2021), DDCP (Xie et al., 2021), DocGeoNet (Feng et al., 2022), Marior (Zhang et al., 2022), RDGR (Jiang et al., 2022), and PaperEdge (Ma et al., 2022). The rectified horizontal textlines of the proposed DocScanner are much straighter than the other rectification methods.

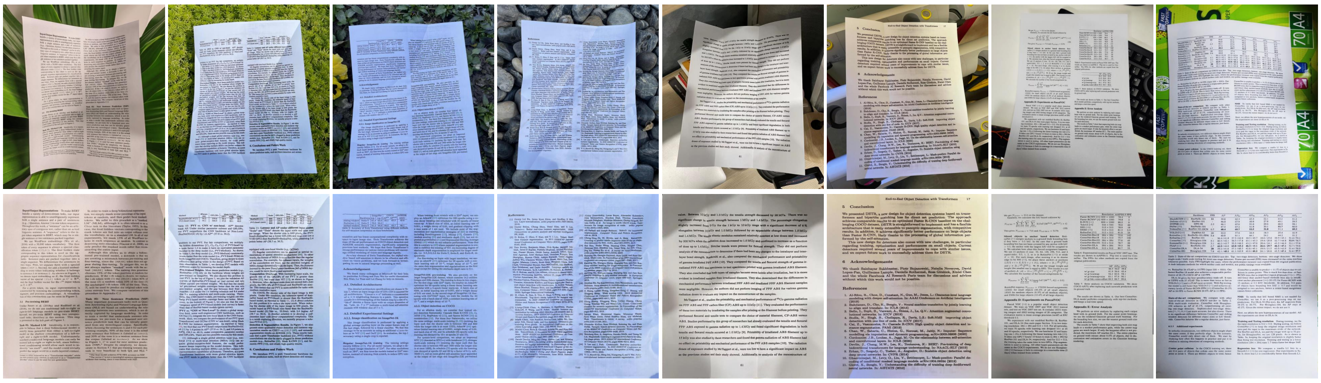


Fig. 10: Robustness illustration of DocScanner on background changes. The two rows show input distorted and corresponding rectified images, respectively. We capture the images of deformed documents, within cluttered backgrounds under outdoor or indoor scenes during the day or night.

The previous method DewarpNet (Das et al., 2019), DocTr (Feng et al., 2021), PaperEdge (Ma et al., 2022), and our DocScanner all directly predict backward warping flow for rectification. They show much higher efficiency than methods (Ma et al., 2018; Xie et al., 2020; Liu et al., 2020) that take the forward warping flow as the ground truth. This is because for the latter methods, during inference, the predicted forward warping flow needs to be converted to the backward warping flow first based on scattered data interpolation (Amidror, 2002), which is time-consuming. Besides, DocProj (Li et al., 2019) first estimates the warping flow of the image patches and then stitches them based on multilabel graph cuts (Boykov et al., 2001),

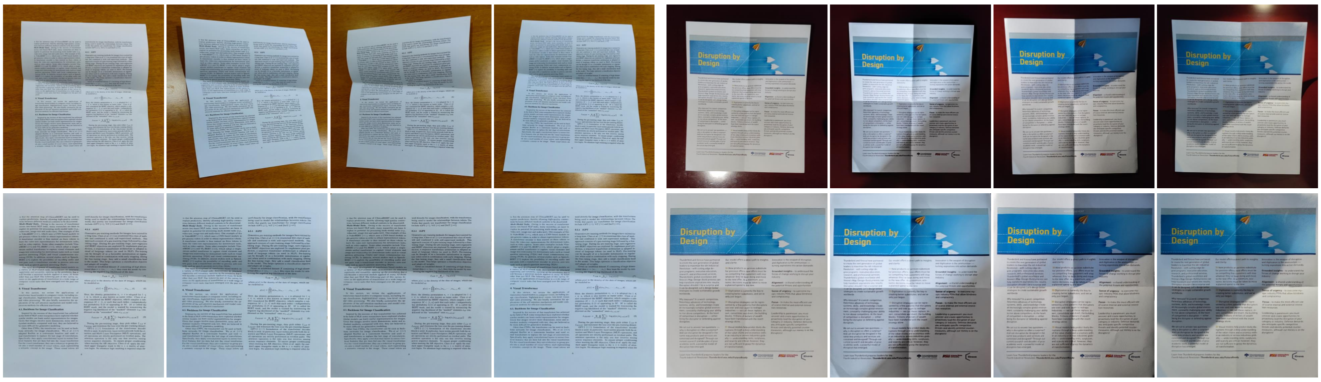


Fig. 11: Robustness illustration of DocScanner on viewpoint (left) and illumination (right) changes. The two rows show input distorted and corresponding rectified images. The input images are captured from different viewpoints (left) and under different illumination conditions (right).

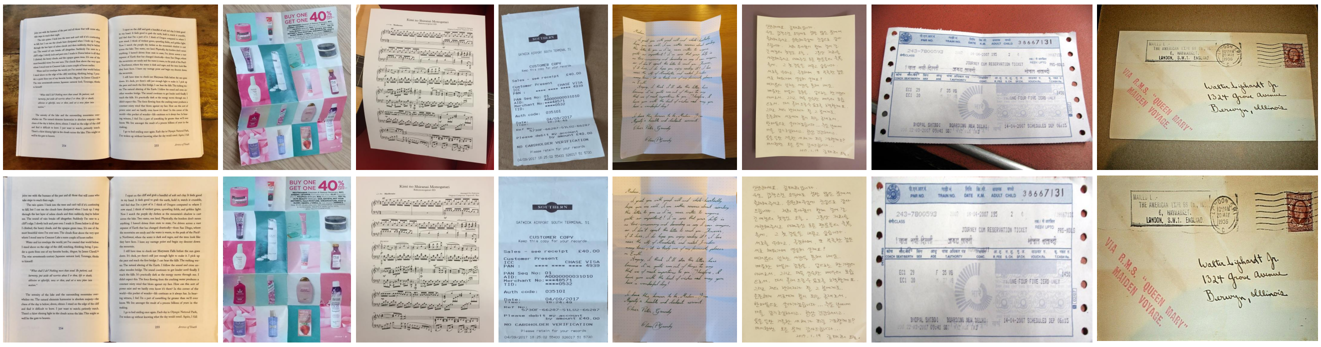


Fig. 12: Robustness illustration of DocScanner on document type changes. The two rows show input distorted and corresponding rectified images. The types of captured deformed document include a full view of books, advertisement, music sheet, receipt, hand-written letter, ticket, and envelop, from left to right.

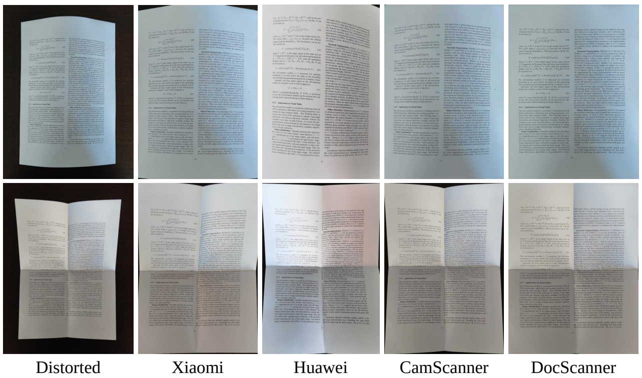


Fig. 13: Qualitative comparisons with the prevalent techniques in smartphones. The two rows show input distorted and corresponding rectified images. Different from the prevalent techniques in smartphones, our DocScanner can handle any irregular deformations.

which heavily increases the computational cost. Compared with them, DocScanner shows superior efficiency, though it involves iterations. This could be ascribed that DocScanner applies a compact recurrent rectification module. DDCP (Xie et al., 2021) regresses a set of control points and PaperEdge (Ma et al., 2022) estimates a sparse backward warping flow, showing higher efficiency. Moreover, since DocScanner ties the weights across iterations, it is the most lightweight method to date. As shown in Table 2, the parameter number of DocScanner-T only has 2.6M parameters, which is approximately 3% of DewarpNet (Das et al., 2019) and 7% of PaperEdge (Ma et al., 2022), and it achieves 10.81 FPS. As shown in Table 3, we further show the running efficiency and parameter count of the document localization module and progressive rectification module in DocScanner-B, respectively.

Comparison with the prevalent techniques. The prevalent algorithms built in smartphones commonly have a restriction that the document must be a regular quadrilateral. Such techniques first detect the corner points of the document to localize the document region and then perform a perspective transformation to rectify the image. Hence, these methods can not handle the situations when the captured document has any irregular deformations. As shown in Fig. 13, we compare our DocScanner with some prevalent techniques, including

Table 4: Ablation experiments of DocScanner-B in terms of image similarity, distortion metrics, OCR performance, and running efficiency on the DocUNet Benchmark dataset (Ma et al., 2018). "↑" indicates the higher the better and "↓" means the opposite.

Models	MS-SSIM ↑	LD ↓	Li-D ↓	ED ↓	CER ↓	FPS ↑	Para. (M)
DocScanner-B	**0.5134**	7.62	1.88	**671.48/434.11**	0.1788/0.1652	8.62	5.2 (1.1+4.1)
Document Localization → None	0.4738	9.22	2.23	668.26/436.50	**0.1734**/0.1668	**10.09**	**4.1**
Upsampling: Learnable → Bilinear	0.5072	8.03	1.87	674.01/452.15	**0.1763**/0.1679	8.69	4.8 (1.1+3.7)
ConvGRU → ConvLSTM	0.5131	7.92	**1.86**	684.26/448.01	0.1792/**0.1647**	8.53	5.7 (1.1+4.6)
Shared Weights → Unshared Weights	0.5087	**7.52**	1.92	680.62/459.52	0.1801/0.1693	-	38.7 (1.1+37.6)
Circle-consistency Loss → None	0.5117	7.99	1.99	663.76/445.40	0.1787/0.1674	-	5.2 (1.1+4.1)

Table 5: Ablation experiments of the rectified feature generator of DocScanner-B in terms of image similarity, distortion metrics, OCR performance, and running efficiency on the DocUNet Benchmark dataset (Ma et al., 2018). "↑" indicates the higher the better and "↓" means the opposite.

Components of $\boldsymbol{x}_k$				MS-SSIM ↑	LD ↓	Li-D ↓	ED ↓	CER ↓	FPS ↑	Para. (M)
$\boldsymbol{c}_0$	$Q_\theta(\boldsymbol{c}_{k-1})$	$V_\theta(\boldsymbol{f}_m^{k-1})$	$\boldsymbol{f}_m^{k-1}$							
✓				0.4736	9.09	2.70	1492.76/1006.88	0.3856/0.3687	9.17	3.9 (1.1+2.8)
✓	✓			0.4762	9.03	2.65	1503.82/922.05	0.3856/0.3645	9.13	4.7 (1.1+3.6)
✓		✓	✓	0.4968	8.01	1.97	697.59/461.92	**0.1741**/0.1668	8.70	4.4 (1.1+3.3)
✓	✓	✓	✓	**0.5134**	**7.62**	**1.88**	**671.48/434.11**	0.1788/**0.1652**	8.62	5.2 (1.1+4.1)

the CamScanner Application[2], the built-in document rectification system of Huawei Mate 30 Pro, and Xiaomi 11. We can see that DocScanner is capable of correcting various irregular deformations. This is because the predicted warping flow of our DocScanner defines a non-parametric transformation, thus being able to represent a wide range of distortions.

Robustness of DocScanner. To verify the robustness of DocScanner, we evaluate the rectification performance in four aspects, including the change of background, viewpoint, illumination, and document types.

Firstly, as shown in Fig. 10, our DocScanner can perform strongly when the captured documents are under various cluttered backgrounds. Note that these distorted images are real document photos captured by smartphones under outdoor or indoor scenes during the day or night. Secondly, we validate the rectification performance when the input distorted images are captured from different viewpoints and under different illumination conditions, respectively. The results are shown in Fig. 11. It can be seen that DocScanner shows high robustness in spite of the various viewpoints and illumination conditions. Thirdly, as shown in Fig. 12, we further evaluate the ability of DocScanner to process distorted images with different document types. The types of captured deformed documents involve a full view of book, advertisement, music sheet, receipt, hand-written letter, ticket, and envelope. Note that such document types are blind in the training dataset but they are well-rectified by our DocScanner. These results reveal the strong generalization ability of our method.

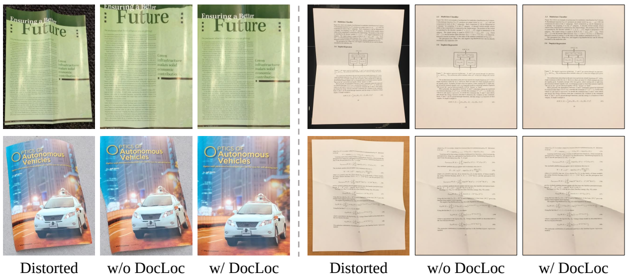


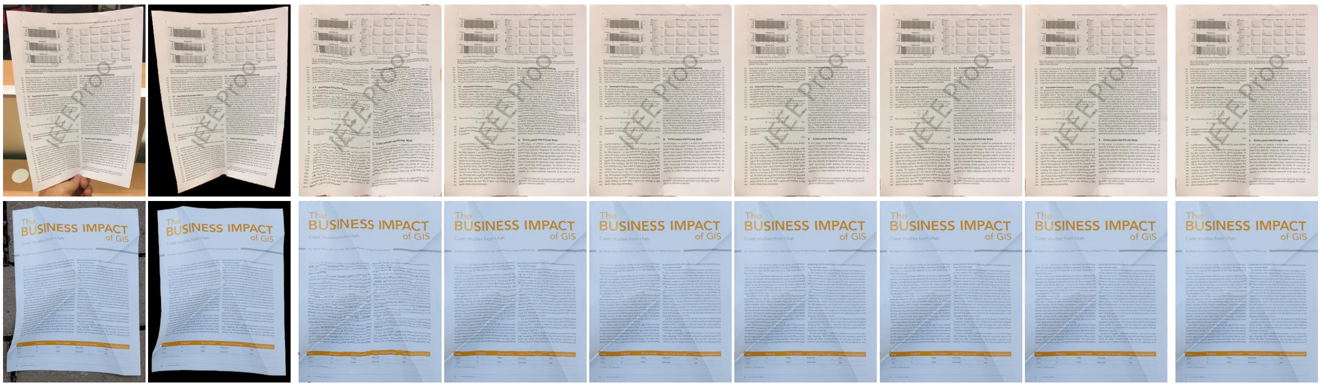
Fig. 14: Examples of the results from DocScanner-B to illustrate the impact of the document localization module (abbreviated as "DocLoc" in this figure). The two failure cases (left) and two successful cases (right) demonstrate that document localization is auxiliary and indispensable for building a robust document image rectification system.

4.5 Ablation Studies

We conduct ablation studies to verify the effectiveness of each component in DocScanner, including the document localization module, the progressive rectification module, and the circle-consistency loss. Several intriguing properties are observed.

Document localization module. Removing the noisy backgrounds or localizing the foreground document is an effective technique for improving the performance, and is widely adopted in the state-of-the-art methods (Xie et al., 2020; Feng et al., 2022; Zhang et al., 2022; Jiang et al., 2022; Ma et al., 2022) or the above built-in software in smartphones. To test its impact on our DocScanner, we train a baseline network without the document localization module, where a distorted

[2] https://www.camscanner.com/


Distorted

Background Removed

Iteration 1

Iteration 3

Iteration 5

Iteration 7

Iteration 9

Iteration 11

Output Rectified

Fig. 15: Visualization of the rectification process of DocScanner-B. We show the rectified document images at the selected odd iterations. It can be seen that DocScanner progressively corrects the document distortion and finally converges to a stable rectification result. It is better viewed in color.

Table 6: Performance of DocScanner-B on selected iterations on the DocUNet Benchmark dataset (Ma et al., 2018) during inference. DocScanner does not diverge even when iteration is up to 200. Settings used in our final model are underlined. "↑" indicates the higher the better and "↓" means the opposite.

iters	distorted	1	2	3	4	5	6	9	<u>12</u>	18	24	36	100	200
LD ↓	20.51	8.82	8.51	8.02	7.99	7.93	7.96	7.74	7.62	**7.55**	7.57	7.56	7.60	7.62
Li-D ↓	5.66	3.15	2.60	2.35	2.21	2.09	2.07	1.92	1.88	**1.83**	1.84	1.86	1.87	1.87

document image with a cluttered background is directly fed to the progressive rectification module. As shown in Table 4, with the document localization module, the performance of DocScanner increases 17.35% (from 9.22 to 7.62) and 15.70% (from 2.23 to 1.88) on metric LD and Li-D, respectively. These gains can be ascribed that in DocScanner, the distorted image feature $\boldsymbol{c}_0$ is taken as the input of the warping flow updater at every iteration, as shown in Fig. 1 and Fig. 3. If we do not localize the foreground documents before conducting the progressive rectification, the background noise will be injected and accumulated along with the iterations, which will disturb the rectification process.

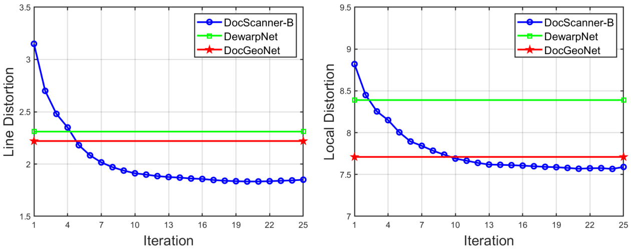


Fig. 16: Performance of DocScanner-B on metric Local Distortion (left) and Line Distortion (right) from the 1^{th} to 25^{th} iteration on the DocUNet Benchmark dataset (Ma et al., 2018) during inference. The lower the values of LD and Li-D, the better the performance. For our DocScanner, the superior performance is obtained after convergence.

Interestingly, as shown in Table 4, the improvement in OCR performance is not remarkable. To provide a more specific view of the impact of the document localization, we showcase four examples in Fig. 14. As illustrated in the left two examples, without document localization, the rectified image fails to cover the complete document region (upper example), and the boundaries are corrupted with background (lower example). In contrast, the right two examples show excellent rectification quality, despite no document localization. We take a deeper analysis by counting the number of failed cases belonging to these failure situations, and find that they only account for **7.69**% of the total test samples in the DocUNet Benchmark dataset (Ma et al., 2018). We observe that the images for OCR evaluation do not involve such cases, which has less influence on the OCR evaluation. These quantitative and qualitative results verify that taking the whole distorted image as the input involves an extra burden to localize the foreground document region besides geometric rectification. More importantly, for our DocScanner, document localization is an auxiliary but indispensable part for building a robust document image rectification system.

Progressive rectification module. In the following, we first validate the major components in the progressive rectification module, including the rectified feature generator, the learnable upsampling module for warp-

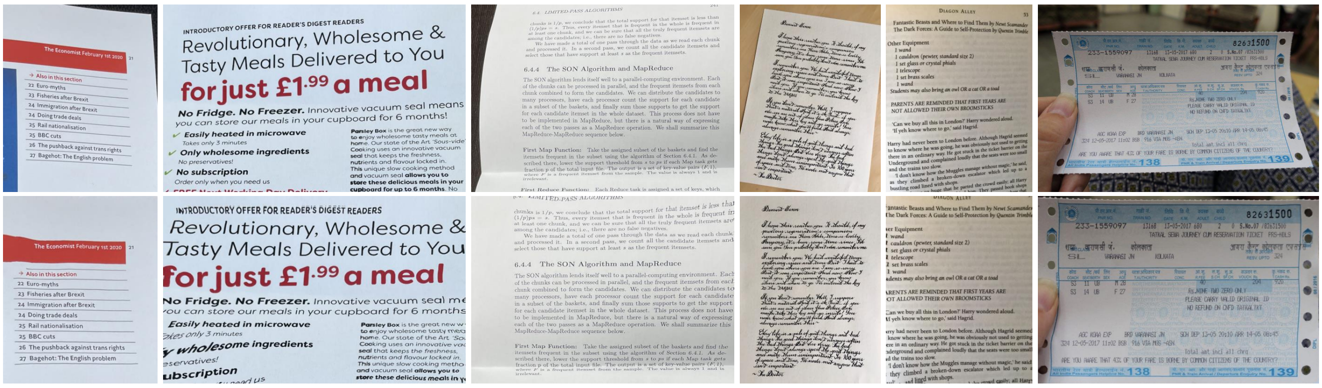


Fig. 17: Example results of the limitation discussion. The two rows show input distorted and corresponding rectified images of DocScanner-B, respectively. When the input distorted images have only incomplete or no document boundaries, the rectified images of DocScanner remain partial distortion.

ing flow, and the warping flow updater. Then, we verify the effectiveness of our progressive learning strategy.

We first validate the compositions of the feature $\boldsymbol{x}_k$ that is fed into the warping flow updater at each iteration. Specifically, as shown in Table 5, we train a baseline model that directly takes the distorted feature $\boldsymbol{c}_0$ as the input $\boldsymbol{x}_k$ to the warping flow updater. That is, the baseline model does not have the rectified feature generator. Then, we integrate the warped feature $(Q_\theta(\boldsymbol{c}_{k-1}))$ and the flow feature $(V_\theta(\boldsymbol{f}_m^{k-1})$ and $\boldsymbol{f}_m^{k-1})$, respectively. The performances obtain a 8.8% and 33.3% gain on metric Line Distortion, respectively. The improvement of former ablation could be ascribed that the warped feature encodes the structure and content information of the predicted rectified image, which is differentiated and processed to estimate the further refinement. For the latter ablation, the flow feature represents the pixel displacement information, which can facilitate the learning of the residual regression for refinement. At last, DocScanner fuses the warped feature and the flow feature to generate the input feature $\boldsymbol{x}_t$. The performance gains are further enhanced due to the strong feature representations.

At each iteration, the warping flow updater outputs the displacement residual $\Delta\boldsymbol{f}_m^k$ at 1/8 resolution. Next, we compare the bilinear upsampling to our learnable upsampling module for $\Delta\boldsymbol{f}_m^k$. As shown in Table 4, the performances are slightly better using the learnable upsampling module. The reason is that, the coarse bilinear upsampling operation for $\Delta\boldsymbol{f}_m^k$ likely can not recover the small deformations of the distorted document.

The default updater unit in DocScanner is ConvGRU. We replace the ConvGRU with ConvLSTM, a modified version of standard LSTM (Hochreiter and Schmidhuber, 1997). In Table 4, while ConvLSTM shows comparable performance, ConvGRU produces higher efficiency on inference time and parameter number. By default, we tie the weights across the total K iterations. Then, we test a version of our approach where each update operator learns a separate set of weights. Performances are slightly better when the weights are untied while the parameters significantly increase.

To provide a more specific view of the rectification process, we provide the results of the selected iteration numbers on the metric LD and Li-D in Table 6. The metric LD and Li-D capture the local and global distortion of the rectified document images, respectively. We can see that the main rectification lies in the top 1∼5 iterations, while the later iterations fine-tune the performance. Besides, the performance does not diverge even when the iteration number K is increased to 200, which illustrates the robustness of our method. As shown in Fig. 15, we further visualize the rectification process and show the corresponding rectified document images at odd iterations. It can be seen that, during the rectification process, the curved textlines in the input distorted document images are progressively corrected and finally converge to a relatively steady position, leading to a stable rectification performance.

As shown in Fig. 16, we further show the performance on the DocUNet Benchmark dataset (Ma et al., 2018) from the 1^{th} to 25^{th} iteration during the inference process on the metric LD (left) and Li-D (right), respectively. For DocScanner, the superior performance is obtained after convergence. Note that our DocScanner-B outperforms DewarpNet (Das et al., 2019) after about 4 iterations, and DocGeoNet (Feng et al., 2022) after about 8 iterations. In our final model, we set the iteration number K=12 to stride a balance between the accuracy and the running efficiency. These quantitative and qualitative results demonstrate the effectiveness and the robustness of the progressive learning strategy.

Circle-consistency loss. With the circle-consistency loss, as shown in Table 4, DocScanner (full model) obtains an important gain on all metrics on the DocUNet Benchmark dataset (Ma et al., 2018). The results illustrate the effectiveness of the straight-line constraint along the rows and columns, relieving the global distortions of the rectified document images.

4.6 Limitation Discussion

In this section, we discuss the limitation of our method. As shown in Fig. 17, when the input distorted images have only incomplete or no document boundaries, the rectified images remain partial distortion.

Interestingly, we can see that the proposed DocScanner still shows a certain rectification capacity for such images, though our training dataset does not contain document images with incomplete boundaries. In fact, for a distorted document image, its rectification cue mainly comes from three aspects, including document boundaries, textlines, and its illumination distribution. When document boundaries are incomplete in an image, the rectification network still can extract the geometric information from the other aspects.

Such distorted images are also common in real life and will be explored in our future work.

5 Conclusion

In this work, we present DocScanner, an effective cascaded system for document image rectification. It localizes the document first and then progressively corrects the document distortion in an iterative manner. With the progressive and iterative correction, DocScanner achieves superior rectification performance and set several state-of-the-art scores on the challenging benchmark dataset. Extensive experiments are conducted to validate the merits of our method. Moreover, we propose a new distortion metric for the field that evaluate the global distortion of the rectified document images. In the future, we will explore the rectification of document images with incomplete boundaries. Besides, considering that DocScanner focuses on the geometric distortion problem of the document images, we intend to further concentrate on the illumination distortion to enhance the visual quality and improve the OCR accuracy. We will seek the solution in further investigations.

References

Amidror I (2002) Scattered data interpolation methods for electronic imaging systems: a survey. Journal of Electronic Imaging 11(2):157–176 12

Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11):1222–1239 12

Brown MS, Seales WB (2001) Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents. In: Proceedings of the IEEE International Conference on Computer Vision, vol 2, pp 367–374 2, 3

Brown MS, Seales WB (2004) Image restoration of arbitrarily warped documents. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(10):1295–1306 2, 3

Brown MS, Tsoi YC (2006) Geometric and shading correction for images of printed materials using boundary. IEEE Transactions on Image Processing 15(6):1544–1554 3

Brown MS, Sun M, Yang R, Yun L, Seales WB (2007) Restoring 2D content from distorted documents. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11):1904–1916 2, 3

Chew Lim Tan, Li Zhang, Zheng Zhang, Tao Xia (2006) Restoring warped document images through 3D shape modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(2):195–208 2, 3

Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:14091259 6

Cimpoi M, Maji S, Kokkinos I, Mohamed S, Vedaldi A (2014) Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3606–3613 9

Courteille F, Crouzil A, Durou JD, Gurdjos P (2007) Shape from shading for the digitization of curved documents. Machine Vision and Applications 18(5):301–316 3

Das S, Ma K, Shu Z, Samaras D, Shilkrot R (2019) DewarpNet: Single-image document unwarping with stacked 3D and 2D regression networks. In: Proceedings of the International Conference on Computer Vision, pp 131–140 2, 3, 8, 9, 10, 11, 12, 13, 16

Das S, Sial HM, Baldrich R, Vanrell M, Samaras D (2020) Intrinsic decomposition of document images in-the-wild. In: Proceedings of the British Machine Vision Conference 2, 8, 9

Das S, Singh KY, Wu J, Bas E, Mahadevan V, Bhotika R, Samaras D (2021) End-to-end piece-wise unwarping of document images. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4268–4277 2, 4, 8, 9, 10, 11, 12

De Boer PT, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. Annals of Operations Research 134(1):19–67 5

Feng H, Wang Y, Zhou W, Deng J, Li H (2021) DocTr: Document image transformer for geometric unwarping and illumination correction. In: Proceedings of the ACM International Conference on Multimedia, pp 273–281 2, 4, 6, 8, 9, 10, 11, 12

Feng H, Zhou W, Deng J, Wang Y, Li H (2022) Geometric representation learning for document image rectification. In: Proceedings of the European Conference on Computer Vision 2, 4, 9, 10, 11, 12, 14, 16

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778 5

He Y, Pan P, Xie S, Sun J, Naoi S (2013) A book dewarping system by boundary-based 3D surface reconstruction. In: Proceedings of the International Conference on Document Analysis and Recognition, pp 403–407 2, 3

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780 16

Jaderberg M, Simonyan K, Zisserman A, et al. (2015) Spatial transformer networks. In: Proceedings of the Neural Information Processing Systems, pp 2017–2025 6

Jiang X, Long R, Xue N, Yang Z, Yao C, Xia GS (2022) Revisiting document image dewarping by grid regularization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4543–4552 2, 4, 8, 9, 10, 11, 12, 14

Kil T, Seo W, Koo HI, Cho NI (2017) Robust document image dewarping method using text-lines and line segments. In: Proceedings of the International Conference on Document Analysis and Recognition, vol 1, pp 865–870 3

Kim BS, Koo HI, Cho NI (2015) Document dewarping via text-line based optimization. Pattern Recognition 48(11):3600–3614 3

Kim G, Hong T, Yim M, Nam J, Park J, Yim J, Hwang W, Yun S, Han D, Park S (2022) OCR-free document understanding transformer. In: Proceedings of the European Conference on Computer Vision, pp 498–517 1

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 9

Koo HI, Cho NI (2010) State estimation in a document image and its application in text block identification and text line extraction. In: Proceedings of the European Conference on Computer Vision, pp 421–434 3

Koo HI, Kim J, Cho NI (2009) Composition of a dewarped and enhanced document image from two view images. IEEE Transactions on Image Processing 18(7):1551–1562 2, 3

Lavialle O, Molines X, Angella F, Baylou P (2001) Active contours network to straighten distorted text lines. In: Proceedings of the International Conference on Image Processing, vol 3, pp 748–751 2, 3

Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol 10, pp 707–710 8

Lévy B, Petitjean S, Ray N, Maillot J (2002) Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics 21(3):362–371 3

Li X, Zhang B, Liao J, Sander PV (2019) Document rectification and illumination correction using a patch-based cnn. ACM Transactions on Graphics 38(6):1–11 2, 3, 9, 10, 11, 12

Liang J, DeMenthon D, Doermann D (2008) Geometric rectification of camera-captured document images. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(4):591–605 3

Liu C, Yuen J, Torralba A (2011) SIFT flow: Dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5):978–994 2, 8

Liu X, Meng G, Fan B, Xiang S, Pan C (2020) Geometric rectification of document images using adversarial gated unwarping network. Pattern Recognition 108:107576 2, 3, 8, 9, 10, 12

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440 3

Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:171105101 9

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110 3

Ma K, Shu Z, Bai X, Wang J, Samaras D (2018) DocUNet: Document image unwarping via a stacked U-Net. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4700–4709 2, 3, 8, 9, 10, 11, 12, 14, 15, 16, 17

Ma K, Das S, Shu Z, Samaras D (2022) Learning from documents in the wild to improve document unwarping. In: Proceedings of the ACM SIGGRAPH Conference, pp 1–9 2, 4, 8, 9, 10, 11, 12, 13, 14

Markovitz A, Lavi I, Perel O, Mazor S, Litman R (2020) Can you read me now? Content aware rectification using angle supervision. In: Proceedings of the European Conference on Computer Vision, pp 208–223

2

Mathew M, Karatzas D, Jawahar C (2021) DocVQA: A dataset for VQA on document images. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp 2200–2209 1

Meng G, Pan C, Xiang S, Duan J, Zheng N (2011) Metric rectification of curved document images. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(4):707–722 3

Meng G, Wang Y, Qu S, Xiang S, Pan C (2014) Active flattening of curved document images via two structured beams. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3890–3897 2, 3

Meng G, Xiang S, Pan C, Zheng N (2017) Active rectification of curved document images using structured beams. International Journal of Computer Vision 122(1):34–60 3

Meng G, Su Y, Wu Y, Xiang S, Pan C (2018) Exploiting vector fields for geometric rectification of distorted document images. In: Proceedings of the European Conference on Computer Vision, pp 172–187 3

Mischke L, Luther W (2005) Document image dewarping based on detection of distorted text lines. In: Proceedings of the International Conference on Image Analysis and Processing, pp 1068–1075 3

Morris AC, Maier V, Green P (2004) From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In: Proceedings of the International Conference on Spoken Language Processing 8

Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch 9

Peng D, Jin L, Liu Y, Luo C, Lai S (2022) PageNet: Towards end-to-end weakly supervised page-level handwritten chinese text recognition. International Journal of Computer Vision 130(11):2623–2645 1

Qin X, Zhang Z, Huang C, Dehghan M, Zaiane OR, Jagersand M (2020) U2-Net: Going deeper with nested u-structure for salient object detection. Pattern Recognition 106:107404 4

Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention, pp 234–241 2, 3

Smith R (2007) An overview of the tesseract ocr engine. In: Proceedings of the International Conference on Document Analysis and Recognition, vol 2, pp 629–633 9

Teed Z, Deng J (2020) RAFT: Recurrent all-pairs field transforms for optical flow. In: Proceedings of the European Conference on Computer Vision, pp 402–419 6

Tian Y, Narasimhan SG (2011) Rectification and 3D reconstruction of curved document images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 377–384 3

Tokmakov P, Alahari K, Schmid C (2017) Learning video object segmentation with visual memory. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4481–4490 6

Tsoi YC, Brown MS (2007) Multi-view document rectification using boundary. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8 3

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proceedings of the Neural Information Processing Systems, pp 6000–6010 4, 10

Wada T, Ukida H, Matsuyama T (1997) 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 3

Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In: Proceedings of the Asilomar Conference on Signals, Systems Computers, vol 2, pp 1398–1402 8, 9

Wu C, Agam G (2002) Document image de-warping for text/graphics recognition. In: Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, pp 348–357 2, 3

Xie G, Yin F, Zhang X, Liu C (2020) Dewarping document image by displacement flow estimation with fully convolutional network. In: Proceedings of the International Workshop on Document Analysis Systems, pp 131–144 2, 3, 8, 9, 10, 11, 12, 14

Xie GW, Yin F, Zhang XY, Liu CL (2021) Document dewarping with control points. In: Proceedings of the International Conference on Document Analysis and Recognition, pp 466–480 2, 4, 8, 9, 10, 11, 12, 13

Xue C, Tian Z, Zhan F, Lu S, Bai S (2022) Fourier document restoration for robust document dewarping and recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4573–4582 4

Yamashita A, Kawarago A, Kaneko T, Miura KT (2004) 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, vol 1, pp 482–485 2, 3

Yang S, Lin C, Liao K, Zhang C, Zhao Y (2021) Progressively complementary network for fisheye image rectification using appearance flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6348–6357 2

You S, Matsushita Y, Sinha S, Bou Y, Ikeuchi K (2018) Multiview rectification of folded documents. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(2):505–511 2, 3, 8

Yuan Y, Liu X, Dikubab W, Liu H, Ji Z, Wu Z, Bai X (2022) Syntax-aware network for handwritten mathematical expression recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4553–4562 1

Zamir SW, Arora A, Khan S, Hayat M, Khan FS, Yang MH, Shao L (2021) Multi-stage progressive image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 14821–14831 2

Zandifar A (2007) Unwarping scanned image of Japanese/English documents. In: Proceedings of the International Conference on Image Analysis and Processing, pp 129–136 3

Zhang J, Luo C, Jin L, Guo F, Ding K (2022) Marior: Margin removal and iterative content rectification for document dewarping in the wild. In: Proceedings of the ACM International Conference on Multimedia, pp 2805–2815 2, 4, 8, 9, 10, 11, 12, 14

Zhang L, Zhang Y, Tan C (2008) An improved physically-based method for geometric restoration of distorted document images. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(4):728–734 2, 3

Zhang L, Yip AM, Brown MS, Tan CL (2009) A unified framework for document restoration using inpainting and shape-from-shading. Pattern Recognition 42(11):2961–2978 3

Zhong X, Tang J, Yepes AJ (2019) PublayNet: Largest dataset ever for document layout analysis. In: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, pp 1015–1022 1

Zhou Z, Fan X, Shi P, Xin Y (2021) R-MSFM: Recurrent multi-scale feature modulation for monocular depth estimating. In: Proceedings of the IEEE International Conference on Computer Vision, pp 12777–12786 6

Zhou Wang, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4):600–612 9