Can You Read Me Now? Content Aware Rectification using Angle Supervision Supplementary Material

Amir Markovitz*, Inbal Lavi*, Or Perel, Shai Mazor, and Roee Litman

Amazon Web Services
{amirmak, ilavi, orperel, smazor, rlit}@amazon.com

1 Introduction

The supplementary material provides a comparison to the method proposed by Li et al. [16] (Section 2), details regarding the dataset used for training (Section 3), and more information regarding the Cartesian to Polar conversion done during the calculation of our loss (Section 4). In addition, we present a side-task providing word segmentation masks (Section 5) and discuss the usefulness of the MS-SSIM metric (Section 6) and provide more visual results from our method.

2 Comparison to Li et al. [16]

A recent related work by Li et al. [16] presented a method for document rectification focused on uneven background illumination and gently folded documents. The work took a patch-based approach for inferring local flow fields followed by a graph-cut model for stitching the patches back to the complete flow. The results obtained using this method both using their publicly available model and by models re-trained using our data are availabe in Table. 1.

While this method shares several similarities with work presented and compared to in this paper, it is not suitable for the kind of deformations found in our dataset and handled both by our model and by DewarpNet [6]. Faced with complex deformations that render non-planar patches, it often resorts to inconsistent stitching patterns, as seen in Fig. 1.

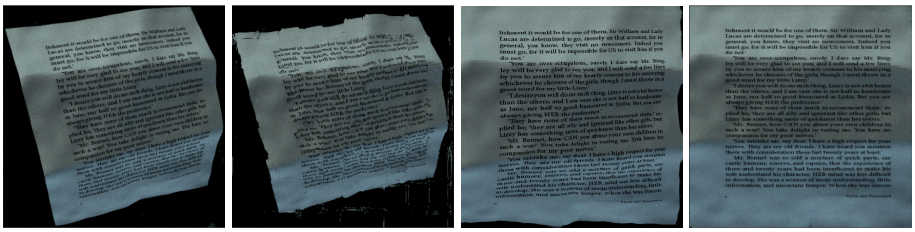
Due to the method's incomparable results we have decided to include it separately, and evaluate both the authors' publicly available pre-trained model[1], and the averaged results of 5 models trained using our dataset.

* - Equal Contribution

[1] https://github.com/xiaoyu258/DocProj

Table 1: Benchmark Comparison to [16] on Synthetic Data. Mean results and standard deviation over the test set. (†) denotes the author's pre-trained model. (*) denotes models retrained using our training set. For SSIM, higher is better. For E_d and EPE, lower is better.

	ED	SSIM	EPE
Li et al. [16]†	0.683	0.281	0.205
Li et al. [16]*	0.652 ± 0.027	0.263 ± 0.003	0.184 ± 0.003
CREASE	0.178 ± 0.003	0.411 ± 0.002	0.043 ± 0.002



Input Li et al. [16] Ours Ground-Truth

Fig. 1: **Results From Li et al. [16].** Left to right: Input image, Results from [16], Our results, and the result of rectifying the input image using the ground-truth backward map. Notice uneven boundaries and visible stitches in the patch based method.

3 Training Dataset

Generation of our dataset. The data was generated in Blender [5] using 10,000 document images and 8000 meshes. The doucment images were extracted from PDFs collected from open-access magazines, books, academic papers, in multiple formats (one, two and three columns, advertisements with a single text blob, etc.) that include diverse images, figures and text. The 8,000 meshes are those used in [6], and were kindly provided by the authors. In addition to the warped images, 3D world coordinates, and UV maps provided by the renderer, we extracted the content meta-data from each PDF document, including text and word bounding boxes, and used them to create the flattened binary text masks. In the following step, each flat mask was warped using the generated backward map. Unlike previous works that rendered relatively low resolution images, here images and annotations were rendered in a 1600×1600 pixel resolution, useful for fine grained OCR evaluations and closer to real world scanning resolutions.

Doc3D The authors of [6] presented the Doc3D dataset, that was also generated in Blender in a similar manner to ours. In Doc3D, however, a few limitations prohibited us from using it during our training and evaluation protocols: (i) At the time of writing, the former dataset is no longer publicly available, except for the meshes used for generation; (ii) The dataset was generated in a $448 \times$

448 resolution, significantly below the required threshold for OCR and even unreadable by people for commonly used font sizes. We thus retrained the models of [6] using our dataset and the training parameters from the publicly available implementation[2].

4 Cartesian to Polar Coordinate Conversion

In order to apply our angular deformation estimation based loss, we predict the rotation angle of each of the two axes. The use of a separate angle for each axis corresponds to both rotation and shear. In other words, during the deformation, both axes rotate differently on a per-pixel level. Axes are rotated individually, and are no longer orthogonal as in a flat surface. The predicted maps account for the magnitudes of the change of each axis, in each direction.

Specifically, the 3D estimation model predicts 4 auxiliary channels, a pair for each axis, which we denote $(\phi_{xx}, \phi_{xy}, \phi_{yx}, \phi_{yy})$. The predictions of ϕ_{xy} provide the value of shift predicted for the X axis in the Y direction, and so forth. For each axis, we then calculate the angle θ_i, and the magnitude ρ_i for $i \in x, y$:

$$\theta_i = \text{arctan2}(\phi_{ix}, \phi_{iy}), \tag{1}$$

$$\rho_i = ||(\phi_{ix}, \phi_{iy})||_2, \tag{2}$$

where 'arctan2' is the four-quadrant variation of the arctangent operator (also referred to as 'atan2'). The calculated values are then used in our loss, as described in Section 3.2 in our paper.

5 Word Segmentation Output

We train an auxiliary word segmentation channel as part of the 3D estimation model. We show in Table 2 that this channel does not improve our results on the OCR metrics. However, this channel can quite accurately localize words and lines areas, as seen in Figure 2. This can be beneficial for the next task in the pipeline, e.g. text localization in the document.

Table 2: Text Segmentation Auxiliary Channel

	E_d	EPE	SSIM
No Text	0.178 ± 0.003	0.043 ± 0.002	0.411 ± 0.002
Text	0.182 ± 0.003	0.043 ± 0.002	0.409 ± 0.004

[2] https://github.com/cvlab-stonybrook/DewarpNet

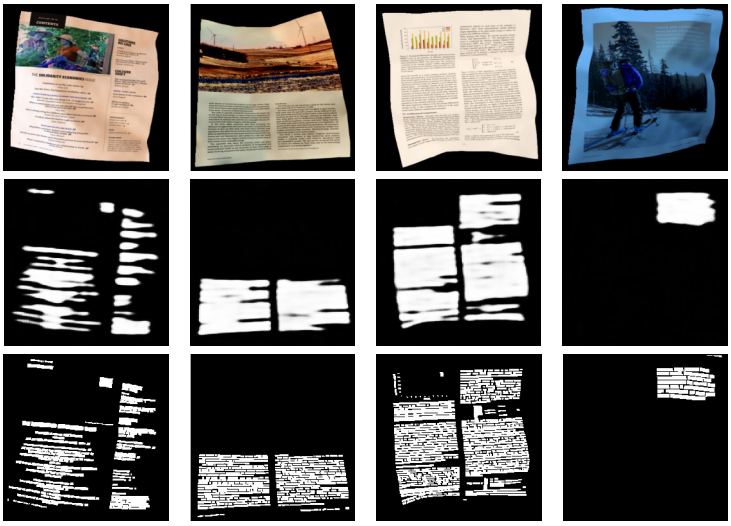


Fig. 2: A visualization of the text segmentation output provided by our 3D estimation model. Top row shows input images, middle row shows our model's predictions, and ground-truth predictions are presented in the bottom row. Robustness of this output can be seen in the right-most column, where text is properly segmented even in relatively low contrast areas.

6 The MS-SSIM Metric

The MS-SSIM [24] metric was used in this work and in other works to capture the rectified document's similarity to the ground truth, rectified variant of it. Given a small shift is to be expected even in very accurate predictions of the backward map, common per-pixel comparison metrics such as L_1 and L_2 are not useful for the task, as they would require an additional non-trivial registration step.

The MS-SSIM metric was chosen as an alternative for evaluating global similarity, specifically when used in multi-scale and applied over an image pyramid. SSIM is an alternative to L_1/L_2 in the sense that it is more correlative with human perception. SSIM still, however, suffers for the same need for exact registration the nultiscale MS-SSIM might deal with this issue more gracefully than the original counterpart.

However, the SSIM based metrics present their own disadvantages, and specifically, lack of sensitivity to fine-grained details and a bias towards even textures. As seen in Fig. 3, a document that wasn't rectified, but contains relatively even and flat textures (including background textures showing) exhibits a far superior similarity score to a properly rectified document that contains dense text. This comes to show that, while useful and intuitive for many applications, in the specific case of document rectification, the MS-SSIM metric isn't fully suitable for comparing the fine-grained details required for a useful, readable output.

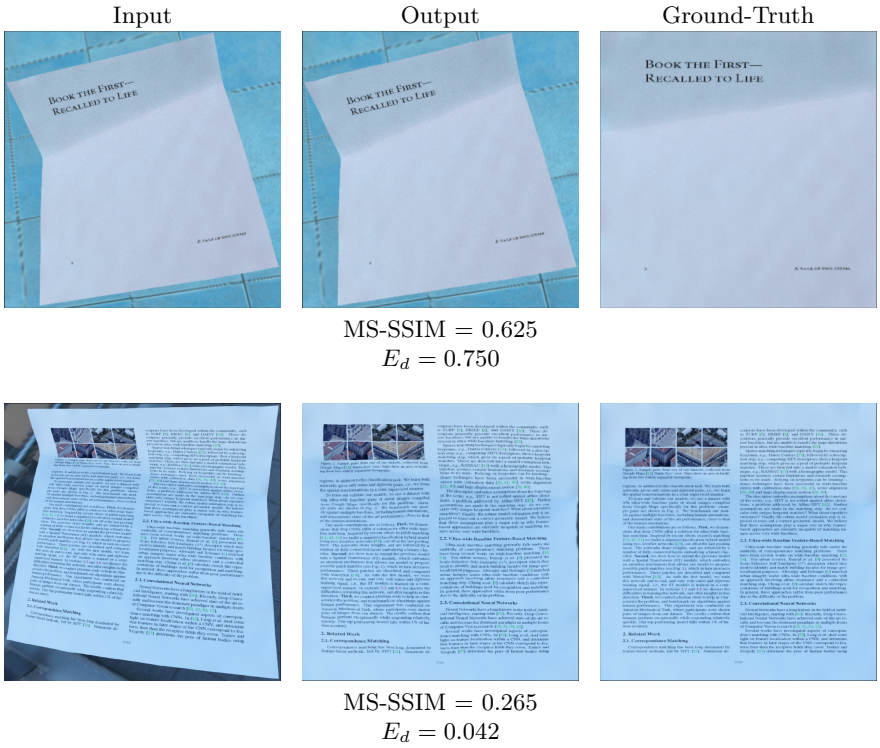


Fig. 3: **Bias in MS-SSIM.** *Top row:* we compare the input image directly, with and identity mapping (no rectification), to the ground-truth rectified image. The high SSIM score shows the bias of the metric towards even surfaces, even when rotated and when a large portion of background is showing. *Bottom row:* Input image is rectified using our method. It is then compared to the ground-truth rectified image. Notice, that even for a successful rectification yielding very low E_d, SSIM value is significantly lower.

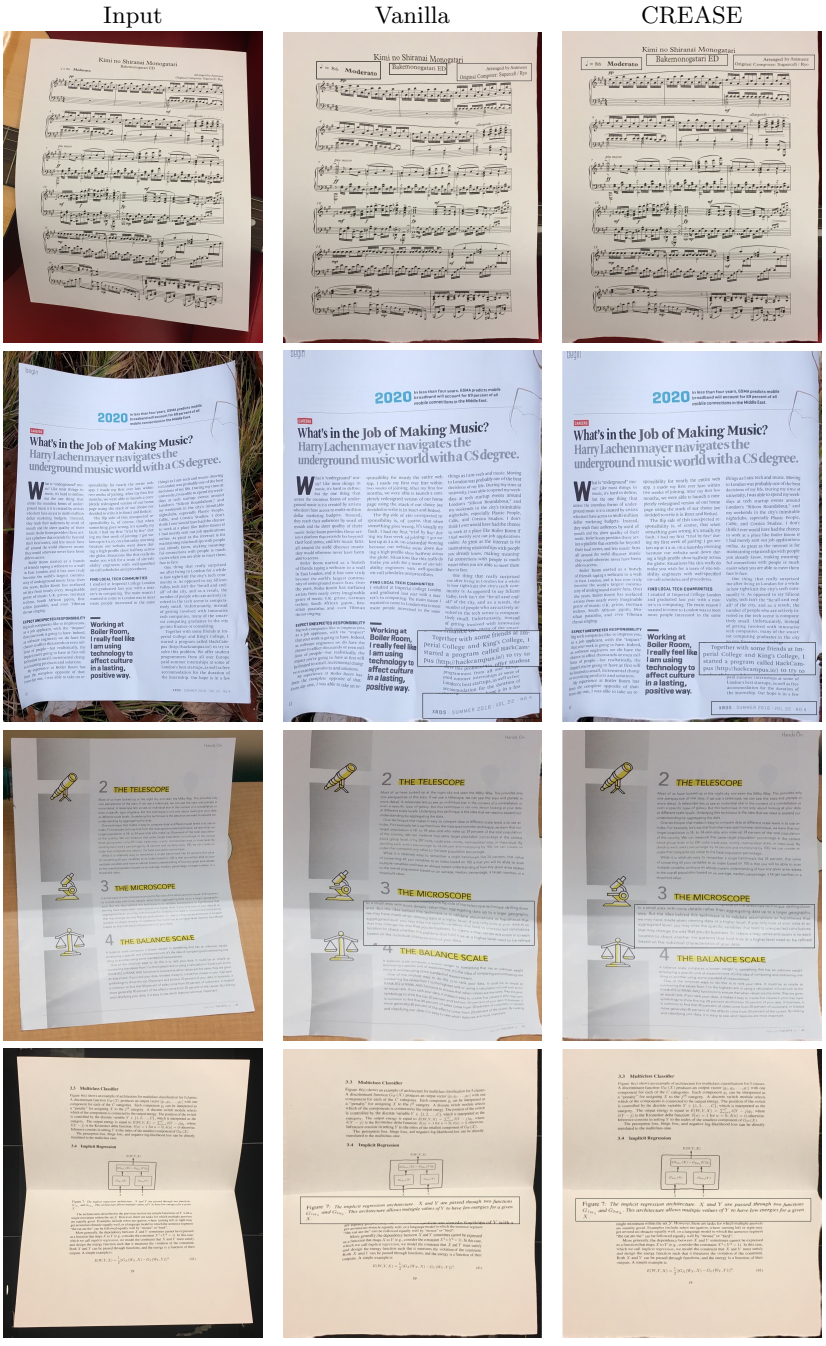


Fig. 4: **Additional Results from the real image dataset of [13].**

References

1. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: International Conference on Computer Vision (ICCV) (2019), to appear
2. Bajjer Ramanna, V.K., Bukhari, S.S., Dengel, A.: Document image dewarping using deep learning. In: The 8th International Conference on Pattern Recognition Applications and Methods. International Conference on Pattern Recognition Applications and Methods (ICPRAM-2019), February 19-21, Prague, Czech Republic. Insticc (2019)
3. Brown, M.S., Seales, W.B.: Image restoration of arbitrarily warped documents. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 1295–1306 (2004)
4. Burden, A., Cote, M., Albu, A.B.: Rectification of camera-captured document images with mixed contents and varied layouts. In: 2019 16th Conference on Computer and Robot Vision (CRV). pp. 33–40. IEEE (2019)
5. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), http://www.blender.org
6. Das, S., Ma, K., Shu, Z., Samaras, D., Shilkrot, R.: Dewarpnet: Single-image document unwarping with stacked 3d and 2d regression networks. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
7. Das, S., Mishra, G., Sudharshana, A., Shilkrot, R.: 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. p. 125–128. DocEng '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3103010.3121030, https://doi.org/10.1145/3103010.3121030
8. Grüning, T., Leifert, G., Strauß, T., Michael, J., Labahn, R.: A two-stage method for text line detection in historical documents. International Journal on Document Analysis and Recognition (IJDAR) p. 285–302 (Jul 2018)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
10. Huang, Z., Gu, J., Meng, G., Pan, C.: Text line extraction of curved document images using hybrid metric. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). pp. 251–255 (Nov 2015). https://doi.org/10.1109/ACPR.2015.7486504
11. Inc., A.: Amazon textract, https://aws.amazon.com/textract
12. Inc., G.: Detect text in images, https://cloud.google.com/vision/docs/ocr
13. Ke Ma, Zhixin Shu, X.B.J.W.D.S.: Docunet: Document image unwarping via a stacked u-net. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2018)
14. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals
16. Li, X., Zhang, B., Liao, J., Sander, P.V.: Document rectification and illumination correction using a patch-based cnn. ACM Transactions on Graphics (TOG) (11 2019)

17. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: Fots: Fast oriented text spotting with a unified network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Jun 2018)
18. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. Lecture Notes in Computer Science p. 71–88 (2018)
19. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia **20**(11), 3111–3122 (Nov 2018)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
22. Smith, R.: An overview of the tesseract ocr engine. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). vol. 2, pp. 629–633. IEEE (2007)
23. Sorkine-Hornung, O.: Laplacian mesh processing. In: Eurographics (2005)
24. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
25. You, S., Matsushita, Y., Sinha, S., Bou, Y.B., Ikeuchi, K.: Multiview rectification of folded documents. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**, 505–511 (2016)
26. Yousef, M., Bishop, T.E.: Origaminet: Weakly-supervised, segmentation-free, one-step, full page text recognition by learning to unfold. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
27. Zheng, Y., Kang, X., Li, S., He, Y., Sun, J.: Real-time document image super-resolution by fast matting. In: 2014 11th IAPR International Workshop on Document Analysis Systems. pp. 232–236. IEEE (2014)
28. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: An efficient and accurate scene text detector. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)