Supplementary Material for: UVDoc: Neural Grid-based Document Unwarping

Floor Verhoeven
ETH Zurich
Switzerland
floor.verhoeven@inf.ethz.ch

Tanguy Magne
ETH Zurich
Switzerland
tanguy.magne@inf.ethz.ch

Olga Sorkine-Hornung
ETH Zurich
Switzerland
sorkine@inf.ethz.ch

1 UVDOC DATASET: ORDERING THE GRID

Using the UV-lit image, where the printed grid is visible, we obtain the pixel coordinates of the grid points on the deformed piece of paper. We then need to compute their correspondences to the vertices of a regular grid, which is equivalent to ordering them as an 89×61 grid. We solve the ordering problem in 3 steps:

(1) *Finding the top-left corner.* We first find the top-left corner of the grid. We compute the two principal components of the detected grid points and define the diagonal direction of the grid as the sum of these two vectors. For each point, we draw a line orthogonal to this diagonal direction and we count the number of points on each side of the line. The top-left corner is then the point that has exactly zero points to its left. The process is illustrated in Fig. 1.

(2) *Ordering border points.* Next we detect all border points. To this end, we use a segmentation of the paper that we obtain by thresholding the UV-lit image. Based on this segmentation, we use OpenCV's findContours function to extract an *ordered* contour polyline. For each contour vertex, we find the nearest neighbor point in the set of grid points. We then define our grid border points as the 296 grid points — the number of points on the border of the grid — that are most frequently found as nearest neighbor. Finally, since the contour extracted using OpenCV is ordered, we can also order the detected grid border points.

(3) *Ordering interior points.* The final step is to order the points that lie in the interior of the grid. We iteratively identify all points $(i, j) \in [2, 88] \times [2, 60]$ in row-major ordering, starting from point $(2, 2)$ (the top-left interior grid point). We do this (for point (i, j)) by finding the three nearest yet-unordered grid points for each of the previously-ordered points $(i-1, j-1)$, $(i, j-1)$, and $(i-1, j)$ (the points to the top-left, top and left of the point we are currently trying to identify). The point that is in the intersection of these three nearest-neighbor sets is chosen as point (i, j). We use the average distance to the three reference points as a tiebreaker in case the intersection contains multiple points. This point is then considered ordered, and we move on to the next point.

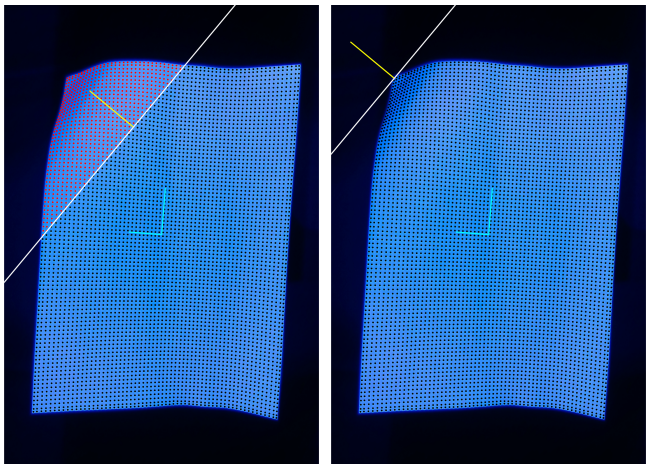

Figure 1: Illustration of the top-left corner identification step. Cyan lines represent the principal components of the grid points, the yellow line is the diagonal direction, and the white line is the orthogonal line defining the dividing half-space. Red points are towards the left of the line and black points towards its right. (Left) There are several red points, this is not the top-left corner. (Right) There are no red points, the top-left corner is the point on top of which the white is located.

2 TRAINING DETAILS

We obtain the ground-truth G and W for the Doc3D dataset by sampling the ground truth backward maps at a regular grid of 45×31 points covering the entire backward map. For our UVDoc dataset (see Sec. 3 of the main paper) we slice the available high-resolution ground truths by a factor of 2.

We use the ADAM optimizer [Kingma and Ba 2015] with a batch size of 8. The initial learning rate is set to 2×10^{-4} for 10 epochs and linearly decays to 0 over 10 further epochs. We alternate optimization steps based on a batch of Doc3D data with a batch of our UVDoc data, using the same loss function on both of them.

We visually augment both the Doc3D and our data with noise, color changes and other appearance transformations. Additionally, we augment our data with rotations, since our images are captured from a more uniform angle than the Doc3D data. All images are tightly cropped before being fed to the network.

Empirically, we find that the best set of weights to balance the influence of the individual loss terms as defined in Eq. 1 in the main paper are $\alpha = 5$ and $\beta = 5$. During training γ is set to 0.0 for the first 10 epochs (first half) and then to 1.0 for the remaining 10 epochs.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12.
https://doi.org/10.1145/3610548.3618174

Table 1: Ablations on training data. The reported values are averages and standard deviations over 10 repetitions of training with otherwise constant parameters. Settings used in our final model are underlined. We show performance on the DocUNet and UVDoc benchmarks. *Doc3D reduced* is a version of the Doc3D dataset with 20,000 samples removed to offset for the additional UVDoc samples. The underlined setting is the one we use.

	DocUNet					UVDoc					
Data	MS-SSIM ↑	LD ↓	AD ↓	CER ↓	ED ↓	MS-SSIM ↑	AD ↓	CER ↓	ED ↓	H-line ↓	V-line ↓
Doc3D	0.492±0.004	7.99±0.13	0.360±0.007	0.197±0.018	757±57	0.669±0.015	0.178±0.013	0.078±0.013	220±30	2.42±0.03	3.85±0.16
Doc3D reduced + UVDoc	0.535±0.004	7.01±0.20	0.331±0.008	0.206±0.019	797±69	**0.765±0.009**	0.138±0.011	0.073±0.010	217±25	**1.84±0.11**	2.65±0.13
Doc3D + UVDoc	**0.536±0.006**	**6.96±0.17**	**0.325±0.006**	**0.195±0.012**	**745±34**	0.762±0.014	**0.129±0.008**	**0.070±0.010**	**205±23**	1.85±0.06	**2.53±0.06**

We give a detailed graphical overview of our model architecture in Fig. 3.

3 EVALUATION METRICS

As explained in the main paper, we used image similarity metrics such as MS-SSIM, LD and AD as well as optical character recognition (OCR) performance measured with CER and ED. Details about these metrics are provided below.

The structural similarity measure (SSIM) [Wang et al. 2004] quantifies the visual similarity between two images by measuring the similarity of mean pixel values and variance within image patches between the two images. The multi-scale variant (MS-SSIM) repeats this process at multiple scales using a Gaussian pyramid and computes a weighted average over the different scales as its final measure. We use the same weights as described in the original implementation [Wang et al. 2003].

LD is computed using a dense SIFT flow mapping [Liu et al. 2008] from the ground truth image to the rectified image. Using this registration, LD is computed as the mean L_2 distance between mapped pixels [You et al. 2018], essentially measuring the average local deformation of the unwarped image.

Aligned distortion (AD) is a more robust variant of the LD metric, introduced in [Ma et al. 2022]. In contrast to LD, AD eliminates the error caused by a global translation and scaling of the image by factoring out the optimal affine transformation out of the SIFT flow distortion. Such a global affine transformation can cause large LD values but does not greatly impact human readability of the image. Additionally, AD weighs the error according to the magnitude of the gradient in the image, emphasizing interesting areas, such as text or image edges, rather than the background. Prior to computing these similarity metrics, we resize all images, both rectified and ground-truth, to a 598,400-pixel area, as suggested in [Ma et al. 2018].

In addition to the image similarity metrics, we evaluate OCR performance based on character error rate (CER) and editing distance (ED) [Navarro 2001]. The CER is defined as the ratio between the ED (the edit distance between the recognized and reference text) and the number of characters in the reference text. We obtain the reference text by extraction from the flatbed scans of the documents. The full definition for the CER then becomes: $\text{CER} = (s+i+d)/N$, where s, i, d are the number of substitutions, insertions and deletions, respectively, and N is the number of characters in the reference text.

4 ADDITIONAL EXPERIMENTS

Mixed training. As shown in the main paper, we find that training models on a combination of the Doc3D and UVDoc datasets yields improved performance compared to training on Doc3D alone. However, models trained on a combination of both datasets see more samples and thus more variety than the ones trained on Doc3D only. To verify that the increased number of unique samples is not the cause of the performance gain, we train on a combination of Doc3D and UVDoc datasets, removing 20,000 samples from the Doc3D dataset. This way, the models trained on a combination of the two datasets see equally many samples as the ones trained on Doc3D only. The results of these experiments, along with the results of models trained on Doc3D only and on a combination of the full Doc3D and UVDoc datasets are presented in Table 1.

The models trained on a combination of the reduced Doc3D dataset and UVDoc have slightly worse performance than the models trained on the full datasets. This is expected, as the models are trained with fewer samples. However, the difference between the two is very small. More importantly, the models trained on the full Doc3D dataset alone give very poor results in comparison. Replacing samples from the Doc3D dataset with higher-quality ones from our UVDoc dataset improves its overall performance.

5 LINE UNWARPING VISUALIZATION

Our new UVDoc benchmark, equipped with the ground-truth unwarping function, allows one to warp and unwarp not only the document image but any texture. We can warp the texture based on the ground truth deformation and unwarp it using the predicted deformation. This idea, which we apply to create our new line straightness metric, can also be used to better visualize the structural behavior of an unwarping function. By unwarping the unshaded document texture, we can remove the visual effect of shape-from-shading, giving a better visualization of the remaining geometric distortions. We apply this to visually compare our method with related works in Fig. 2.

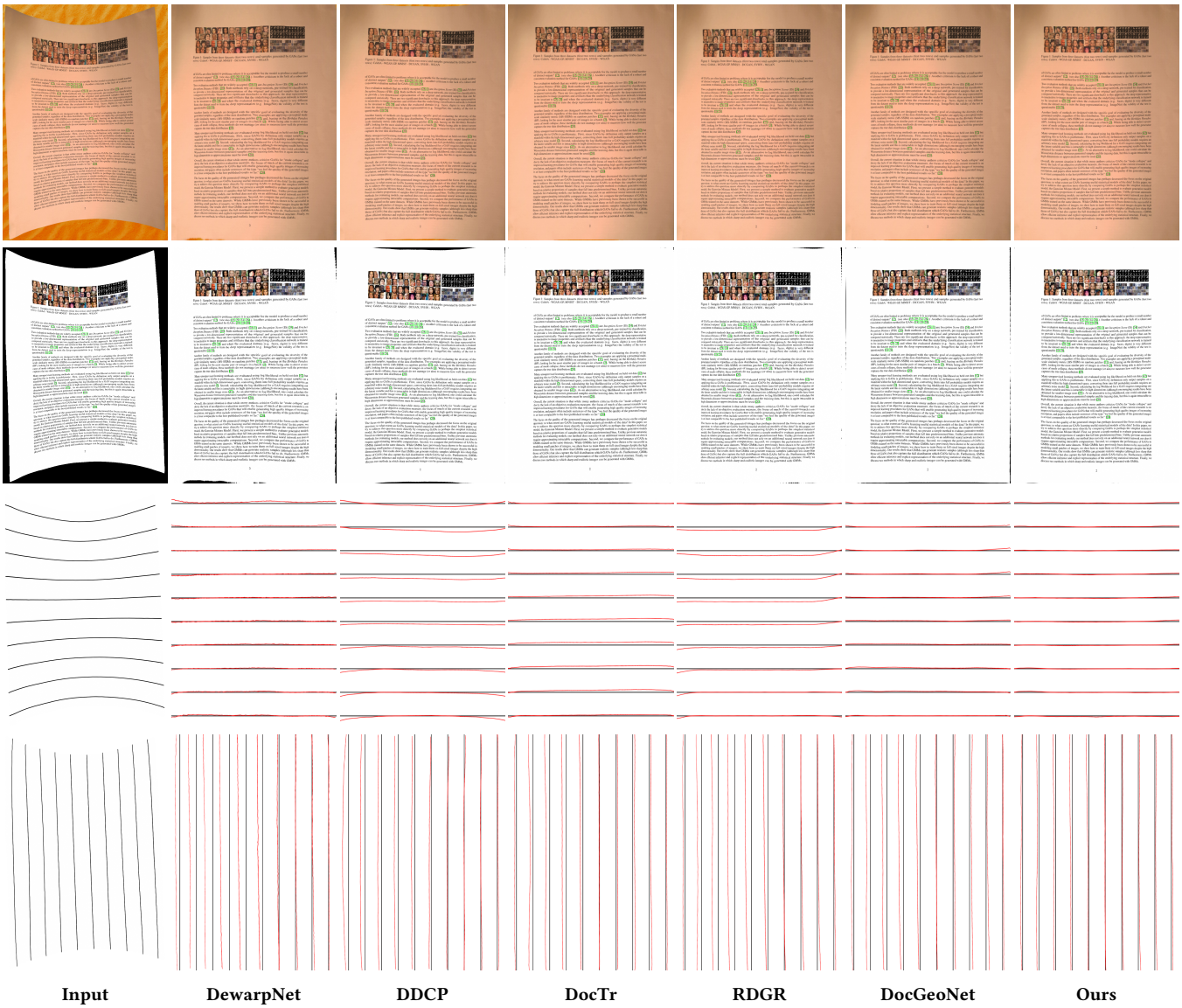

Figure 2: Results on our UVDoc benchmark. From top to bottom: shaded image, unshaded document texture, horizontal lines, vertical lines. The black lines represent the ground-truth and the red lines are the unwarped ones. From left to right: input, DewarpNet [Das et al. 2019], DDCP [Xie et al. 2021], DocTr [Feng et al. 2021], RDGR [Jiang et al. 2022], DocGeoNet [Feng et al. 2022], ours.

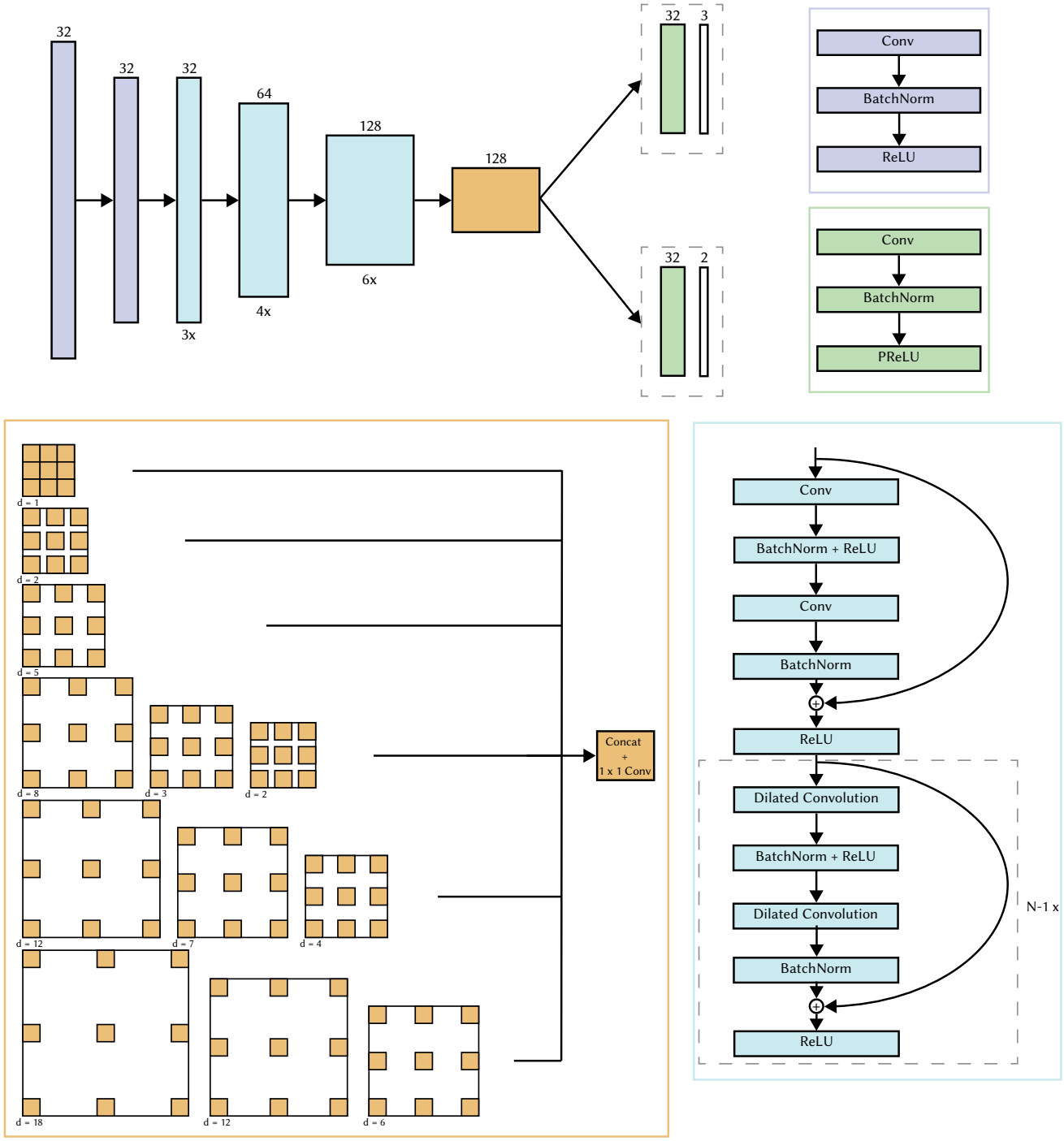

Figure 3: An overview of the architecture of our network.

REFERENCES

Sagnik Das, Ke Ma, Zhixin Shu, Dimitris Samaras, and Roy Shilkrot. 2019. DewarpNet: Single-Image Document Unwarping With Stacked 3D and 2D Regression Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 131–140. https://doi.org/10.1109/ICCV.2019.00022

Hao Feng, Yuechen Wang, Wengang Zhou, Jiajun Deng, and Houqiang Li. 2021. DocTr: Document Image Transformer for Geometric Unwarping and Illumination Correction. In *Proceedings of the ACM International Conference on Multimedia*. 273–281. https://doi.org/10.1145/3474085.3475388

Hao Feng, Wengang Zhou, Jiajun Deng, Yuechen Wang, and Houqiang Li. 2022. Geometric Representation Learning for Document Image Rectification. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 475–492. https://doi.org/10.1007/978-3-031-19836-6_27

Xiangwei Jiang, Rujiao Long, Nan Xue, Zhibo Yang, Cong Yao, and Guisong Xia. 2022. Revisiting Document Image Dewarping by Grid Regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4533–4542. https://doi.org/10.1109/CVPR52688.2022.00450

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. 13 pages.

Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. 2008. Sift flow: Dense correspondence across different scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 28–42. https://doi.org/10.1007/978-3-540-88690-7_3

Ke Ma, Sagnik Das, Zhixin Shu, and Dimitris Samaras. 2022. Learning From Documents in the Wild to Improve Document Unwarping. In *Proceedings of ACM SIGGRAPH*. 9 pages. https://doi.org/10.1145/3528233.3530756

Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. 2018. DocUNet: Document Image Unwarping via a Stacked U-Net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4700–4709. https://doi.org/10.1109/CVPR.2018.00494

Gonzalo Navarro. 2001. A Guided Tour to Approximate String Matching. *Comput. Surveys* 33, 1 (2001), 31–88. https://doi.org/10.1145/375360.375365

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *Proceedings of The Asilomar Conference on Signals, Systems Computers*, Vol. 2. 1398–1402. https://doi.org/10.1109/ACSSC.2003.1292216

Guo-Wang Xie, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. 2021. Document Dewarping with Control Points. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*. 466–480. https://doi.org/10.1007/978-3-030-86549-8_30

Shaodi You, Yasuyuki Matsushita, Sudipta Sinha, Yusuke Bou, and Katsushi Ikeuchi. 2018. Multiview Rectification of Folded Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 2 (2018), 505–511. https://doi.org/10.1109/TPAMI.2017.2675980