Supplementary Material for the paper “Evaluating Deep Neural Networks for Image Document Enhancement”

Lucas N. Kirsten
HP Inc. - R&D, Brazil
Porto Alegre, Brazil
lucas.nedel.kirsten@hp.com

Ricardo Piccoli
HP Inc. - R&D, Brazil
Porto Alegre, Brazil
ricardo.piccoli@hp.com

Ricardo Ribani
HP Inc. - R&D, Brazil
Barueri, Brazil
ricardo.ribani@hp.com

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction; Image processing; Neural networks.**

KEYWORDS

document analysis, image enhancement, deep learning

ACM Reference Format:

Lucas N. Kirsten, Ricardo Piccoli, and Ricardo Ribani. 2021. Supplementary Material for the paper “Evaluating Deep Neural Networks for Image Document Enhancement”. In *ACM Symposium on Document Engineering 2021 (DocEng '21)*, August 24–27, 2021, Limerick, Ireland. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3469096.3474938>

1 DOCUMENT IMAGE QUALITY ASSESSMENT (DIQA)

Document image quality assessment (DIQA) aims to measure the visual quality of document images, both in subject of reliability and visual perception overall (*i.e.* the aesthetic aspects). Li *et al.* [9] and Shen *et al.* [13] use Optical Character Recognition (OCR) engines for verifying aspects such as text alignment, text color and size, that are not relevant for the context of this work. Hussain *et al.* [7], on the other hand, proposed a metric that considers aesthetic aspects of the document image, however, we have not considered it since it is a reference-less metric. Alaei *et al.* [1] proposed a metric based on measuring human perception over quality and document readability, but we could not find an implementation available in order to reproduce the results of that paper.

2 RETINANEX MODEL OF COLOR CONSTANCY

If $E \in \mathbb{R}^3$ is an enhanced image with lighting-corrected content, then, according to [5], we have that:

$$E(x, y, c) = \mathfrak{F} \left(\frac{R(x, y, c)}{L(x, y, c)} \right), \quad (1)$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
DocEng '21, August 24–27, 2021, Limerick, Ireland
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8596-1/21/08...\$15.00
<https://doi.org/10.1145/3469096.3474938>

where $R \in \mathbb{R}^3$ is the raw (scanned) image, $L \in \mathbb{R}^3$ is the illumination surface of the raw image, and $\mathfrak{F} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is an enhancement function. Assuming that $\mathfrak{F}$ is a linear mapping function, we can obtain an image with added illumination by:

$$R(x, y, c) = E(x, y, c) \cdot L(x, y, c). \quad (2)$$

This is the base for our augmentation strategy of synthetically add illumination surfaces to the enhanced images of our training set. Figure 1 shows an example of this process.

Figure 1: Lighting surface augmentation method. First column: original raw image and enhanced image with algorithm from Fan [5], respectively. Top: lighting surfaces extracted from other images; Remaining images at the Bottom: resulting augmented raw images using the upper lighting surfaces and the enhanced image.

3 TRAINING SETUP

All networks were trained using two Nvidia RTX 2080 GPUs with 12 GB of VRAM and implemented using the Keras Framework [3] with Tensorflow 1 [4] as the backend. The augmentation procedures were applied on-the-fly during the models training. In the following we provide training details and the parameters used.

3.1 U-Net

We trained the default U-Net architecture with a reduced number of neurons for each layer: 15, 30, 60 and 120 respectively for both the encoder and decoder segments, due to hardware limitations.

The network was trained with the Adam optimizer [8], a learning rate of $5 \cdot 10^{-2}$, mean squared error (MSE) as the loss function, and batch size of 5 images.

3.2 EDSR

We trained the default architecture¹ with 8 improved ResNet blocks containing 64 neurons in each convolutional layer. However, we removed the up-sample blocks, since we aimed to maintain the input shape and leverage only their feature extraction module. The network was trained with the Adam optimizer [8], a learning rate of $1 \cdot 10^{-6}$, MSE as the loss function, and a batch size of 22 images.

3.3 WDSR

We trained the default architectures² with 8 ResNet blocks containing 32 neurons in the convolutional layers, 4-time expansion in the WDSR-A architecture and a 6-time expansion in the WDSR-B. However, we removed the up-sample blocks for the same reason as in the EDSR architecture. The networks were trained using a Weight Normalization Adam optimizer³ implementation [12], a learning rate of $1 \cdot 10^{-6}$, MSE as the loss function, and a batch size of 15 images.

3.4 SRGAN

In order to train this network for document enhancement, we replaced the default VGG16 [11] feature extractor model by an internal pre-trained model that classifies different types of documents using the MobileNet V1 [6] as the backbone. The expected result of this modification was to improve the feature loss propagation, since this internal model already was trained to extract features from document images (instead of natural images as pre-trained VGG on Imagenet). The up-sample blocks were also removed, for the same reasons as discussed for EDSR and WDSR.

For our use case, the generator network receives a raw document image as input and the discriminator classifies whether the generator's output was an enhanced document or not, based on examples of actual enhanced documents from our training set. Initially, the generator was pre-trained using a pixel-wise loss function with ground truth images (similarly to the other models) for 3 epochs with 1000 iterations per epoch. In this pre-training step, the Adam optimizer [8] was used with a fixed learning rate of $1 \cdot 10^{-6}$ and MSE as the loss function. Then, we continue by training the entire GAN network⁴. To evaluate the evolution of GAN training, the generator was evaluated after each 1000 iterations with the WaDIQaM metric [2], saving the results every time the network improved.

3.5 IRCNN

We used the default hyper-parameters for the architecture⁵ and added a subtraction layer at the end of the model, which receives

¹We used the following EDSR base model implementation: <https://github.com/krasserm/super-resolution/blob/master/model/edsr.py>.

²We used the following WDSR base model implementation: <https://github.com/krasserm/super-resolution/blob/master/model/wdsr.py>.

³We used the following Weight Normalization implementation: <https://github.com/openai/weightnorm/blob/master/keras/weightnorm.py>.

⁴The SRGAN training used the same hyper-parameters and procedures as presented in the following repository: <https://github.com/eriklindernoren/Keras-GAN#srgan>.

⁵We used the following IRCNN base model implementation: <https://github.com/lipengFu/IRCNN>.

the input image and the estimated residual image, resulting in the enhanced document image. For training, we used the Adam optimizer [8], a learning rate of $1 \cdot 10^{-4}$, MSE as the loss function, and a batch size of 20 images.

3.6 CycleGAN

We trained the model using default architecture and training hyper-parameters from an existing implementation⁶. The raw-to-enhanced generator was evaluated after each one thousand iterations with the WaDIQaM [2] metric, saving the results every time the network improved. However, we observed that this model is overly sensitive to the weights update, and often gets stuck in local minima. To address this, after the evaluation, if the model did not improve, the weights from the last best evaluation were re-applied on the model.

4 PSNR DEFINITION

The PSNR is defined as:

$$PSNR = 20 \cdot \log_{10} \left(\max_{p \in I} p \right) - 10 \cdot \log_{10} (MSE), \quad (3)$$

where $\max_{p \in I} p$ is the maximum possible pixel value of the image (e. g., 255 for 8-bit images) and MSE is the mean squared error. The PSNR measures the ratio between the maximum possible power of a signal and the power of the corrupting noise.

5 IQA INVESTIGATION

In order to evaluate which full-reference IQA metrics would be better suited for evaluating document image quality, we first used the algorithm from Fan [5] to enhance the captured images from our test set and manually selected the images that performed best using the same methodology presented in Section 3.3 of the original paper. We then compared each raw, enhanced and white images (used as a control image, as the dataset comprises only documents with white background) with their corresponding ground truth (the digital PDF version), storing the values returned by each of the metrics.

For this work, we assume that a proper quality metric should return better scores (*i.e.* return higher or lower value, depending on the metric) on enhanced images than on their captured (raw) or fully white counterparts. Similarly, the captured image should have better scores than the fully white image (as the white image can be seen as the original with all of its content erased). Therefore, for each metric, we defined two error measures:

raw errors as how many times the captured image produced a better score than the enhanced image;

white errors as how many times the fully white image produced a better score than the captured image.

The results are presented in Table 1⁷. The metrics with the lowest total error (sum of *raw* and *white* errors) were, respectively: PIE [10], WaDIQaM [2] and the MS-SSIM [14]. For this reason, we used these metrics as a baseline to compare our results. We also used the PSNR error, because it is a traditional IQA metric, and is based on the MSE

⁶We used the same CycleGAN hyper-parameters as presented in the following repository: <https://github.com/eriklindernoren/Keras-GAN#cyclegan>.

⁷In the PDF version of this paper, you can click on each metric to navigate to the implementation used herein.

error that was used to train most of the neural networks studied in this work.

Table 1: Relative values from IQA results. Lower values indicate better performance. The best results are highlighted in bold. The *Total* column is the sum of the *Raw error* and the *White error* for each row.

Metric	Raw error	White error	Total
BSNR	4.0	16.2	20.2
PSNR	0	28.8	28.8
MAE	0	38.4	38.4
MSE	0	28.8	28.8
SNR	0	28.8	28.8
PAMSE	0	10.6	10.6
MS-SSIM	2.5	1.0	3.5
ERGAS	6.1	22.2	28.3
RMSE	0	33.3	33.3
RMSE-SW	0.5	36.9	41.9
SAM	35.9	37.4	73.3
SCC	22.7	0	22.7
UQI	0.5	37.4	42.4
VIFP	7.1	0	7.1
MAD	49.0	0.5	49.5
F-SSIM	15.2	0	15.2
PSNR HMA	6.6	0	6.6
SR SIM	37.4	0	37.4
GMSD	30.8	100	130.8
MDSI	6.1	0	6.1
HaarPSI	35.4	0	35.4
PIE	0.5	0	0.5
WaDIQaM	0	3.0	3.0

6 QUALITATIVE RESULTS

In this section, we present the visual (qualitative) results from the model outputs. Figure 2 compares the output of an image from each of the networks that returned poor enhancements with its ground truth. Meanwhile, Figures 3, 4, 5, and 6 show a visual comparison of the outputs from the networks with good enhancements with the outputs from the algorithm from Fan [5] as well.

REFERENCES

- [1] Alireza Alaei, Romain Raveaux, Donatello Conte, and Bela Stantic. 2018. "Quality" vs. "Readability" in Document Images: Statistical Analysis of Human Perception. In *2018 13th LAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 363–368.
- [2] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2017. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing* 27, 1 (2017), 206–219.
- [3] Francois et al. Chollet. 2015. *Keras*. <https://github.com/fchollet/keras>
- [4] Martin Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <http://tensorflow.org/> Software available from tensorflow.org.
- [5] Jian Fan. 2007. Enhancement of camera-captured document images with watershed segmentation. *CBDAR07* (2007), 87–93.
- [6] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
- [7] Mehdi Hussain, Ainuddin Wahid Abdul Wahab, Yamani Idna Bin Idris, Anthony TS Ho, and Ki-Hyun Jung. 2018. Image steganography in spatial domain: A survey. *Signal Processing: Image Communication* 65 (2018), 46–66.
- [8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [9] Hongyu Li, Fan Zhu, and Junhua Qiu. 2019. Towards Document Image Quality Assessment: A Text Line Based Framework and A Synthetic Text Line Image Dataset. *arXiv preprint arXiv:1906.01907* (2019).
- [10] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. 2018. PieAPP: Perceptual Image-Error Assessment Through Pairwise Preference. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [11] Hussam Qassim, Abhishek Verma, and David Feinzimer. 2018. Compressed residual-VGG16 CNN model for big data places image recognition. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 169–175.
- [12] Tim Salimans and Diederik P. Kingma. 2016. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In *Neural Information Processing Systems 2016*.
- [13] Aili Shen, Bahar Salehi, Timothy Baldwin, and Jianzhong Qi. 2019. A joint model for multimodal document quality assessment. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 107–110.
- [14] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

Figure 2: Example of network results that were not considered to be accurately enhanced by manual inspection.

Figure 3: Results for a black-and-white document image

Figure 4: Results for a colored mixed-content document image

Figure 5: Results for a mixed-content document image containing ambiguous content (i.e., shadows)

(a) Raw image

(b) Ground truth

(c) Fan [5]

(d) EDSR

(e) SRGAN

(f) WDSR

(g) U-Net

Figure 6: Results for a black-and-white document image captured under poor lighting conditions