ShabbyPages: A Recipe for Repeatable, Synthetic, Modern Document Images

Anonymous ICDAR 2023 submission

No Institute Given

Abstract. The ShabbyPages document image dataset is produced using the Augraphy document image augmentation tool. The development of the computation pipeline used to generate the corpus is discussed, and the results presented. The final corpus contains over 6000 "born-digital" ground truth document images, sourced on the Web, with synthetically-noised counterparts ("shabby pages") that appear to have been printed and faxed, photocopied, or otherwise altered through physical processes. The release of this dataset and the recipe for its production attempts to address a growing need for labeled training document images for supervised learning tasks. The results of initial experiments are discussed, in which the corpus is used to train performant convolutional denoisers which remove real noise features with a high degree of human-perceptible fidelity, establishing baseline performance for a new ShabbyPages benchmark.

1 Introduction

Denoising is a fundamental concern for many document processing workflows, wherein unwanted artifacts introduced to a document image via noisy processes like scanning are removed. The effectiveness of the denoising stage of document processing pipelines has implications for downstream tasks like optical character recognition (OCR) and layout parsing [need to cite]. Recent work in supervised machine learning has yielded promising results at the denoising task [need to cite], increasing the importance of access to large volumes of high-quality training and evaluation data.

Prior work has introduced denoising and binarization datasets, but these are often small in scale, limited in the types of noise features present, not diverse enough to train robust models, and impossible to recreate.

- The DIBCO datasets — widely recognized as critical data for document denoising and binarization — are far too small for large-scale training; each DIBCO dataset ranges between 10-20 images total.
- The NoisyOffice denoising dataset [9] is also quite small, having *72* training samples, and contains only a few types of noise — primarily artificial wrinkles and stains, with little variation.
- The DDI-100 corpus [**?**] is larger in scale than NoisyOffice and the DIBCO datasets, but is again limited in the degree of feature diversity [**?**].

In this paper, we introduce a new dataset designed primarily for the document denoising task, which addresses each of the concerns listed above. This new dataset, *ShabbyPages*, is large-scale, having *6202* training and *mmm* [TODO: do we need this?] testing samples. *ShabbyPages* is more diverse than alternatives like NoisyOffice, as it contains documents from various language groups and contains documents with graphical elements like tables and figures. In the next sections, we provide motivation and context for ShabbyPages' creation, compare it at length to other important datasets in the same space, describe the corpus construction process, and finally demonstrate its utility by benchmarking document denoising models trained on the dataset.

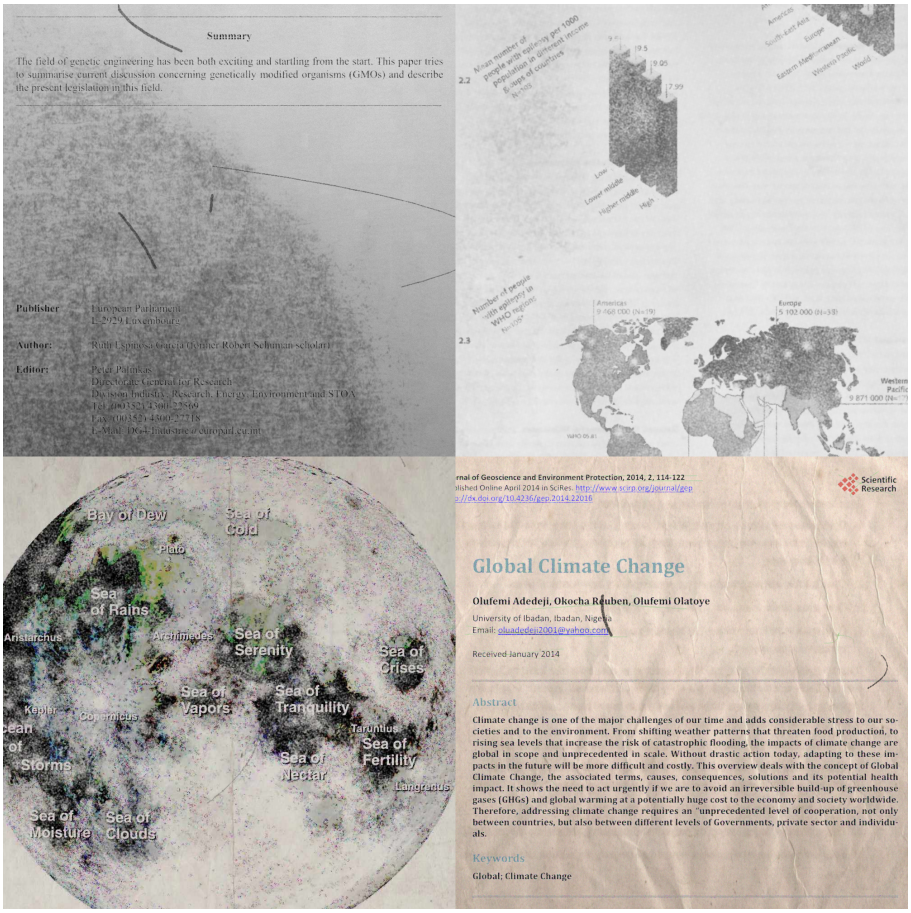

Fig. 1. Sample patches from ShabbyPages.

2 The Dataset

ShabbyPages joins a rapidly growing number of open-source datasets freely available online. Human activity today generates a tremendous amount of data, with many people choosing to publish theirs on websites like Kaggle [**?**], Hugging Face Hub [**?**], and many more. All of this data is arranged into categories: sets exist which contain text, images, audio, and data from other modalities, with each set often collected and designed for a purpose. Of the many image datasets available, the authors understand few to have been designed for groundtruthed supervised learning of document-specific tasks like binarization or denoising. In fact, we could find only one other set compiled specifically for doing so with images of *modern* documents: NoisyOffice [1].

2.1 Features

Modern artificial neural networks produce a latent representation of their training data, a space from which outputs are sampled. In deep networks, these representations live within multiple layers, each of which corresponds to a feature of the input data. Document images are a particularly interesting category, as the data within the documents each picture represents is itself frequently multi-modal; the document may contain an image and its description, a table with statistics, multiple regions of structured and formatted text, and even other documents, making their internal representations by neural networks often quite complicated.

Networks with a wider variety of latent representations frequently perform better when generalizing their learned functions to other tasks. For this reason, procuring a large-enough volume of robust-enough training data is paramount. Several sets of such images exist, many intended for processing by deep neural nets; Table [**?**] contains a comparison of some of the key hyperproperties of the ShabbyPages set compared with other popular document image datasets.

Diversity. We use the diversity metric as defined in [6,7] to help us measure intra-dataset variety. This metric is defined as

$$Diversity(X) = \frac{1}{|X|^2} \sum_{a \in X} \sum_{b \in X} \|f(a) - f(b)\|_2$$

where $f(\cdot)$ is an embedding function mapping the input image into an n-dimensional vector space. Here, we use CLIP's ViT model [8], which embeds each image into a 512-dimensional space. The intuition behind the diversity metric is that datasets where images are highly similar will have lower diversity scores, and datasets where images share less visual similarity will have higher diversity scores. A dataset with low diversity might not be representative enough of the real world, and low diversity may further correlate with task easiness.

2.2 Statistics

ShabbyPages contains documents from multiple classes, which contain many types of information.

Table [?] contains some statistics for the dataset, in grayscale, at a standard resolution of 150 ppi.

Table 1. ShabbyPages dataset statistics.

Statistic	**Value**
Num. Images	6202
DPI	150
Max image size	3336x12157
Min image size	532x532

2.3 Other Datasets

By now, hundreds of datasets exist for the document denoising task (several listed in [?]), the most similar to ShabbyPages being NoisyOffice [9]. NoisyOffice is very limited, however, consisting of two font weights, three font sizes, three font types, and four noise types, for a total of just 72 unique images. The NoisyOffice collection was created by permuting these font/size/base combinations, and the resulting set is feature-poor. NoisyOffice also lacks non-English and non-Latin characters, as well as graphical elements like images, logos, and tables. To add noise, the creators produced four different types of background noise (folds, wrinkles, coffee stains, and footprints), and then overlayed digitally-generated text onto these.

Another dataset, DDI-100 [?], was constructed in a manner similar to NoisyOffice. DDI-100 is far larger than NoisyOffice, but has been criticized for not containing enough diversity or noise to be an effective benchmark [?]. Additionally, the pipeline code for creating the noisy versions of both NoisyOffice and DDI-100 is not available, and thus these datasets cannot be reproduced.

Other prior work includes several datasets for the related task of document image binarization, which seeks to convert a document image into a binary image, where foreground text is black and background pixels are white. Among the most important in this class are the DIBCO collections [?,?,?,?,?,?,?,?,?,?]. As summarized in Table 2, the DIBCO family of datasets is very small, ranging between 10 and 20 samples each (for a total of 106 images across DIBCO-9 through DIBCO-18). Similar to NoisyOffice, the DIBCO datasets do not contain graphical elements like images, tables, charts, etc. The Persian Heritage Image Binarization Dataset (PHIBD) [?,?] is similar to the DIBCO benchmarks, consisting of a small number of images of handwritten Persian writing. Others, like the Bickley Diary set [?] and the 2019 Time-Quality Binarization Competition set [?] have also been used for benchmarking document binarization approaches, but these do not appear to be publicly available [?]. Ultimately, the sizes of these document binarization corpora severly limit the types of algorithms and mod-

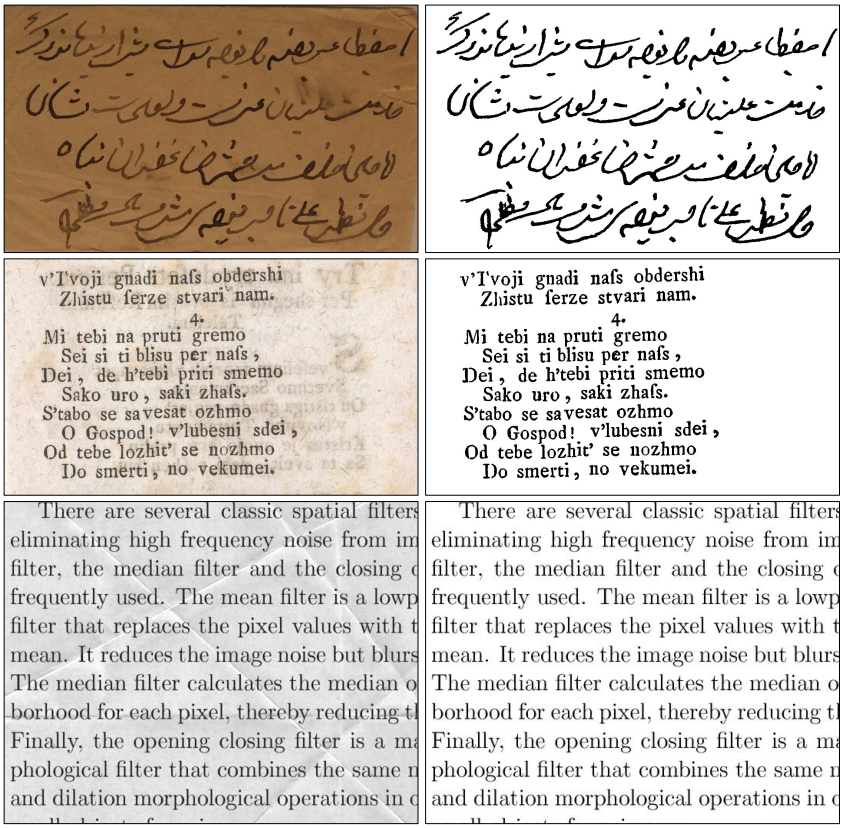

Fig. 2. Example noisy-clean pairs from PHIBD (top row), DIBCO-13 (middle row), and NoisyOffice (bottom row).

els that can be used. Examples of noisy-clean image pairs from some of these collections are displayed in Figure 2.

Other popular document image datasets — like RVL-CDIP [4], Tobacco-3482 [CITE], Tobacco-800 [3], and FUNSD [cite] — include large amounts of real noise, but none of these have corresponding clean ground-truth document images to train or evaluate document denoisiers and binarizers.

3 Construction

This section describes the dataset generation methodology; code for all of this is available on GitHub.

ShabbyPages was born out of a desire to address shortcomings in the popular datasets discussed in the previous section. Reproducibility was a primary concern, so ShabbyPages includes not only the groundtruth images, but also the original digital documents used to produce them, and the software for fully

Table 2. Summary of datasets for document denoising and binarization tasks.

Dataset	*Dataset Size*	*Synthetic Noise*	*Ground-Truths*	*Diversity*	*Font size*	*Paper styles*	*Multilingual*	*Contains graphics*	*Reproducible*
ShabbyPages (ours)	6,202	✓	✓	0.488	multi	multi	✓	✓	✓
NoisyOffice [1]	72	✓	✓	0.317	16pt	1	✗	✗	✗
DocCreator [5]		✓	✗						✓
DDI-100 [**?**]		✓	✓						✗
DIBCO-9 [**?**]									n/a
DIBCO-10 [**?**]	10	✗	✓		multi	multi	✗	✗	n/a
DIBCO-11 [**?**]	16	✗	✓		multi	multi	✓	✗	n/a
DIBCO-12 [**?**]	14	✗	✓		multi	multi	✗	✗	n/a
DIBCO-13 [**?**]	16	✗	✓		multi	multi	✓	✗	n/a
H-DIBCO-14 [**?**]	10	✗	✓		multi	multi	✓	✗	n/a
H-DIBCO-16 [**?**]	10	✗	✓		multi	multi	✓	✗	n/a
DIBCO-17 [**?**]	20	✗	✓		multi	multi	✓	✗	n/a
H-DIBCO-18 [**?**]	10	✗	✓		multi	multi	✓	✗	n/a
DIBCO-19 [**?**]		✗							n/a
Bickley Diary [**?**]				n/a					n/a
PHIBD									n/a
TQBC									n/a

recreating the set, allowing anyone to quickly produce a new member of the Shabby family, optionally re-exporting groundtruths at a different resolution beforehand. The 6202 images in the ShabbyPages release set are really just the first sample from the space of these documents; we encourage readers to build their own. NoisyOffice is the only other set we examined that contains its digital provenance, but the released database materials lack a means of reproducing the simulated printing method used to create the NoisyOffice training data; Augraphy's `PaperFactory` augmentation is such a tool. The other sets are not even theoretically extensible in this way.

3.1 Data Gathering

A team of workers searched the public internet for PDFs of many different kinds, from culturally-important sources like government agencies, NGOs, and large multinational corporations. 600 unique documents were retrieved, totaling 6202 pages. Care was taken to retrieve freely-available and attributable documents; a manifest is included in the dataset which contains an ordered table of the filenames, the languages present in the document, the download URL for the file, and an English name or acronym for the source organization or person.

A similar process was completed to collect paper textures on which to "print" the documents: 300 unique textures were gathered, all of which are either in the public domain or carry CC-0, CC-BY, or CC-BY-SA licenses; metadata about these textures appears in another manifest. We also manually reviewed the gathered documents to verify that no personal identity information was present within the corpus.

3.2 PDF to PNG

The pdftoppm tool from the poppler-utils package was used to split the PDF documents into individual pages. Each page was converted to a PNG image at 150dpi, a common printing resolution. Because the majority of these documents were created with the standard US Letter dimensions, this resulted in the most common image dimension being 1275 pixels wide by 1650 pixels tall.

```
pdftoppm document_name.pdf document_name -png -r 150
```

3.3 Developing the Pipeline

The Augraphy library presents an easy-to-use API for constructing feature pipelines, which has been designed for interoperability with other augmentation tools, and within the broader data ecosystem. While Augraphy's default pipeline has what we believe to be quite realistic defaults, we wanted a broader range of features from this dataset than those the default pipeline could produce. To address this, we broke all of the parameters for every augmentation constructor out into separate variables, tweaking these and committing the new pipeline to GitHub. We created an automated daily build in GitHub Actions to render a random selection of ground-truth images with the updated pipeline, then our team met frequently to discuss the output and make adjustments to the augmentation parameters, until we were consistently satisfied with the results.

The pipeline uses 23/26 augmentations from Augraphy; we excluded augmentations that would make it hard to do pixel to pixel comparisons against groundtruths.

PaperFactory was used with 300 textures, dramatically increasing the possible variations in the final dataset. We did not use Brightness or Brightness-Texturize, because these can interact poorly with some of the paper textures we chose. We also omitted the BookBinding augmentation, which can alter the document image geometry by adding additional pixels, making the result unsuitable for analysis by many common metrics.

Augraphy by design includes a high degree of randomness: most augmentations make several calls to a pseudorandom number generator, on which several transforms depend. All or none of these augmentations may have applied to each input image, and only one pipeline run was completed per image.

The conjunction of all these factors implies ShabbyPages is a sample from a massive space. Indeed, the cosine similarity computed over OpenAI's CLIP representations of each output image averages to 0.49, much higher than NoisyOffice's 0.31.

3.4 Processing Augraphy on a multicore system

Execution time is dependent on which augmentations are executed at runtime; a long Augraphy pipeline can take several seconds to process large images. The library is under active development with performance enhancements underway, but the time cost to generate large datasets sequentially is prohibitive when dealing with thousands of files, so we use a multi-process pooling technique to distribute the workload across many processor execution threads. For each process, a new pipeline with new probability coefficients was generated, representing a nontrivially-different sequence of document operations. Each of the 6202 base images was passed through a unique pipeline, with the output saved to a separate directory. Augraphy's API design made it very easy to write the code to accomplish this; the entire ShabbyPages generation script is fewer than 30 lines long including imports, with the most critical piece represented below:

```
def run_pipeline(filename):
  image = cv2.imread(filename)
  # returns the current ShabbyPages pipeline
  pipeline = get_pipeline()
  data = pipeline.augment(image)
  shabby_image = data['output']
  cv2.imwrite(filename + '-shabby.png',
              shabby_image)
```

Processing all 6202 images took less than half an hour on the 64 cores of a Graviton3 c7g.16xlarge instance on AWS. In testing, we rendered several tens of thousands of such images, with the cumulative computation time still less than one day due to Augraphy's efficiency.

3.5 Real Shabby Pages

In keeping with the spirit of NoisyOffice, our original inspiration, and to provide out-of-distribution test data for ours and other projects, we developed a separate but related dataset which we call ShabbyReal. To build this, we isolated reproduction steps for several physical operations, designed real-world augmentation pipelines with these, and then engaged a team of workers to use these instructions to produce several copies of printed born-digital documents with real noise.

Five individuals were recruited for this task, which involved printing 40 pages of documents and manually applying various effects such as folds, wrinkles, smudges, highlighters, pencils, crayons, splatters of ink or other liquids, printing, scanning, faxing and other creative and reasonable treatments determined by these workers. The process continued over a period of two weeks, incurring a cost of approximately $1,000 USD. While there are plenty of opportunities to streamline the operation, scaling this proce

Several challenges arose in the construction of this dataset. Old fax machines, faulty commercial printers, and other hardware suitable for inducing the intended effects were difficult to source, and frequently non-functional when they

could be found. Environmental concerns like the volume of paper and ink used, and the energy expenditure both in electricity and human terms, also factored into our decision to pursue synthetic augmentations for ShabbyPages. Manually producing a database of noisy documents is a time-consuming and repetitive task; creating just 100 real-world documents with real noise features required the labor of several people. Nevertheless, despite the limited size of the resulting dataset, the scarcity of real-world datasets with ground truth images makes the ShabbyReal corpus extremely valuable. This type of dataset is critical for benchmark comparison with synthetically-augmented document images, and for other tasks such as denoising and binarizing: we use ShabbyReal later for a human-perception evaluation of denoising performance.

4 Experimentation with the ShabbyPages Set

Denoising and binarization are two important techniques for the removal of unwanted data from images. Both approaches can be achieved by supervised learning of relationships between features of the input and output data. In this section, we train two NAFNet [2] instances as denoisers on both the NoisyOffice and the ShabbyPages datasets. We evaluate these models' predictions in a cross-validation test on both datasets, both by visual inspection and by metric computation. We then perform a similar experiment where the same NAFNets are trained to not only remove noise from the test images, but also to classify the foreground and backgorund pixels, predicting a binarized clean document. The model was "off-the-shelf"; besides rigging up the training inputs and choosing a reasonable training epoch limit, we made no significant alterations to the provided hyperparameters.

4.1 Denoising

Digital computation has enabled humanity to produce ever-growing amounts of both data and reasons to process it. Doing so is easiest when the data is free of unexpected outliers, the signal has less jitter, and perhaps most importantly, when our aesthetic sensibilities are un-challenged.

Denoising grayscale images is a challenging task, as models must determine not only which features should appear in the output, but also the 8-bit pixel intensity over the whole image. We randomly selected ten patches of 400x400 pixels from each page, then trained another NAFNet instance on this data for just 14 epochs (about 22 hours of compute time). For this task, we considered the structural similarity index (SSIM), a common image processing metric which takes into account visually-perceptible data such as luminance and contrast, appropriate for grayscale images. The results of the denoising cross-validation experiment are presented in the table below.

As can be seen from Table [**?**], the ShabbyPages-trained denoiser generalized to NoisyOffice far better than the NoisyOffice-trained model generalized to ShabbyPages. This result is consistent with the much higher diversity present within the ShabbyPages set, relative to features found in the NoisyOffice database.

Table 3. Document image denoising cross-test performance of NAFNet models trained on ShabbyPages and NoisyOffice.

Training Set	Test Set	SSIM↑
ShabbyPages	NoisyOffice-real	**0.96**
NoisyOffice-sim	ShabbyPages	0.83

Machine learning is ultimately about getting computers to complete tasks which normally require human labor; as further validation for the efficacy of ShabbyPages in training performant denoisers, we also used our ShabbyPages-trained NAFNet model to denoise the ShabbyReal set, and present some examples of its performance below.

4.2 Binarization

In this section, we discuss results obtained by training a NAFNet denoiser to both restore augmented images to their ground truths and to classify pixels as foreground.

We first performed an ensemble binarization preprocessing step on the clean groundtruth images. For each image, we computed the Niblack, Sauvola, and Otsu thresholds, using these to binarize three copies of the input. The average over the resulting matrices was taken elementwise, producing the final binarized groundtruth.

One instance of the model was trained on the full SimulatedNoisyOffice corpus, while the other was trained on a 1050-image subset of ShabbyPages, cropped to a single random 500x500 patch. In both cases, the prediction target was the relevant ensemble-binarized groundtruths, and each instance trained for 100 epochs. As a test, these models were used to denoise and binarize both the full NoisyOfficeReal dataset and a 450-image testing subset taken from ShabbyPages which does not overlap the training subset.

Multiple common image processing metrics were applied to the predictions made by each model, comparing these to the preprocessed groundtruths. In addition to the structural similarity metric from the denoising task, we add the peak signal to noise ratio (PSNR) and the root mean squared error (RMSE). Both of these metrics are more useful for determining the differences between binary images: together, they give some idea of the degree to which noise artifacts remain in the image after denoizing and binarization affect the visual signal, and how many such artifacts exist. The averages of each metric over all predictions was taken, with the results presented in Table [**?**].

We cross-validate these models by predicting cleaned and binarized images using the other testing set as inputs. The ShabbyPages-trained model was able to classify foreground and background pixels in the NoisyOffice test set with a substantially higher degree of accuracy than the NoisyOffice-trained model could classify ShabbyPages, even with less-favorable training data. ShabbyPages contains a much higher feature diversity than NoisyOffice, so these results were

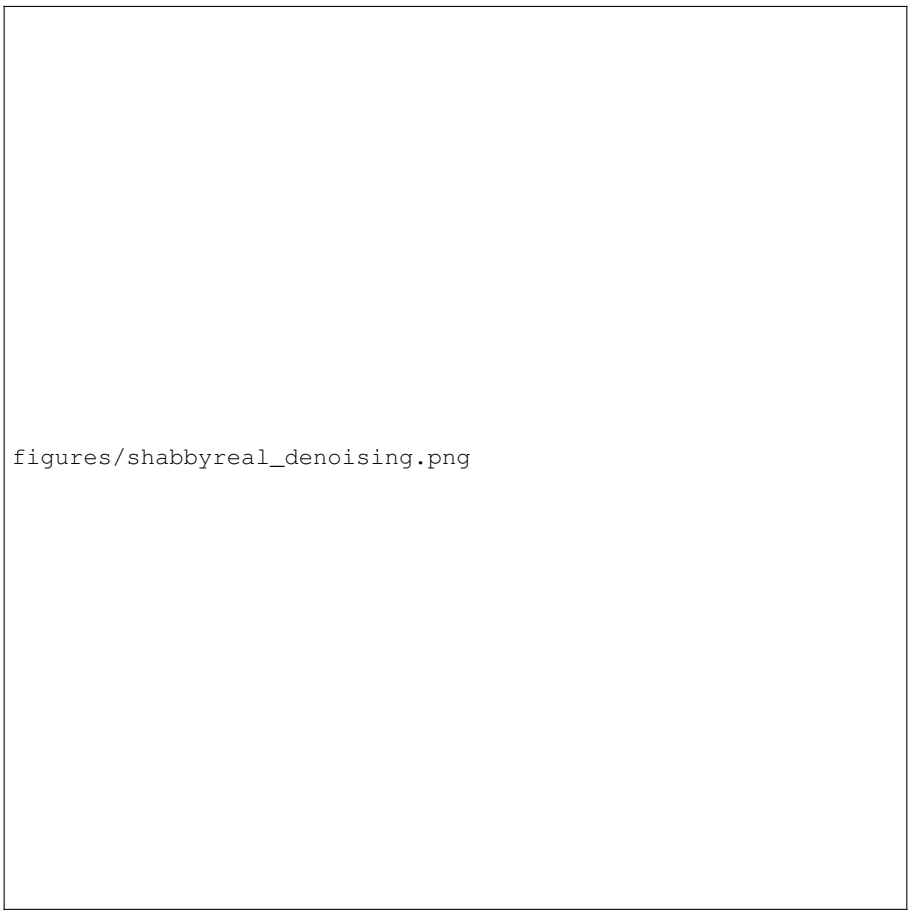

Fig. 3. Sample ShabbyReal images and their Shabby_NAFNet-denoised versions.

Table 4. Document image binarization performance of a NAFNet model trained and tested on ShabbyPages and NoisyOffice.

Training Set	Test Set	SSIM↑	PSNR↓	RMSE↓
ShabbyPages	NoisyOffice-real	**0.947**	**38.098**	**3.205**
NoisyOffice-sim	ShabbyPages	0.811	34.562	5.384

unsurprising, although we did expect our model to achieve worse results than it did, since it only saw a single patch from each full page. The ShabbyPages-trained model performs well on a much wider range of features, handling tables, graphs, and even simple images quite well. This is a strong indicator that neural networks trained on ShabbyPages can outperform those trained on NoisyOffice when generalizing to other datasets.

5 Conclusion

Supervised learning requires a collection of training data and accompanying "labels", which in the graphical modality are clean images, free of noise or other degradations. We presented ShabbyPages, a large dataset composed of 6202 clean/noisy pairs of synthetically-noisy images taken from 600 digital documents. Information about the set's contents and its relation to other common datasets was explored. The noisy images were created from their clean origins by the application of a pipeline developed with the `Augraphy` library; the production of ShabbyPages was detailed in the third section, with all source code and input materials made openly available. ShabbyPages was used to train performant denoising and binarization models which generalize well to document images containing real noise.

References

1. Castro-Bleda, M.J., España-Boquera, S., Pastor-Pellicer, J., Zamora-Martínez, F.: The NoisyOffice Database: A Corpus to Train Supervised Machine Learning Filters for Image Processing. The Computer Journal **63**(11), 1658–1667 (11 2019). https://doi.org/10.1093/comjnl/bxz098, https://doi.org/10.1093/comjnl/bxz098
2. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. arXiv preprint arXiv:2204.04676 (2022)
3. Doermann, D.: Tobacco 800 dataset, https://tc11.cvc.uab.es/datasets/Tobacco800_1
4. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: International Conference on Document Analysis and Recognition (ICDAR) (2015)
5. Journet, N., Visani, M., Mansencal, B., Van-Cuong, K., Billy, A.: Doccreator: A new software for creating synthetic ground-truthed document images. Journal of Imaging **3**(4) (2017). https://doi.org/10.3390/jimaging3040062, https://www.mdpi.com/2313-433X/3/4/62
6. Kang, Y., Zhang, Y., Kummerfeld, J.K., Tang, L., Mars, J.: Data collection for dialogue system: A startup perspective. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers) (2018), https://aclanthology.org/N18-3005

7. Larson, S., Mahendran, A., Lee, A., Kummerfeld, J.K., Hill, P., Laurenzano, M.A., Hauswald, J., Tang, L., Mars, J.: Outlier detection for improved data quality and diversity in dialog systems. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (2019), https://aclanthology.org/N19-1051
8. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021), https://arxiv.org/pdf/2103.00020.pdf
9. Zamora-Martínez, F., España Boquera, S., Castro-Bleda, M.: Behaviour-based clustering of neural networks applied to document enhancement. In: Computational and Ambient Intelligence (2007), https://link.springer.com/chapter/10.1007/978-3-540-73007-1_18