# Augraphy: Data Augmentation for Document Images*

Alexander Groleau[1], Kok Wei Chee, and Stefan Larson[2]

[1] Sparkfish, Addison TX, USA
[2] SkySync, Ann Arbor MI, USA
[3] ck91wei@gmail.com

**Abstract.** This short paper introduces *Augraphy*, a Python package for data augmentation pipelines for document image analysis. Augraphy uses many different augmentation strategies to produce augmented versions of clean document images that appear as if they have been distorted due to noisy paper printing, faxing, scanning, or copy machine processes.

**Keywords:** Document Analysis · Denoising · Data Augmentation.

## 1 Introduction

Data augmentation is a widely used strategy in various areas of machine learning, including computer vision, image processing, natural language processing, and audio applications. Data augmentation can be used to generate new training samples data by applying transformations, rotations, noise, and other modifications to training data. Alternatively, data augmentation can be used to create noisy or challenging evaluation data from clean data, in which case it can be used for robustness testing or image denoising.

This paper introduces Augraphy,[4] a Python library for document image data augmentation. Augraphy uses highly-configurable pipelines to apply adjustments to document images to create augmented versions that appear old or noisy, as if they had been printed on dirty laser or inkjet printers, scanned by dirty or low-quality office scanners, or otherwise mistreated by real-world paper handling office equipment. This paper highlights some of the features of Augraphy, and demonstrates how it can be used effectively to produce challenging synthetic document denoising data.

## 2 Augraphy

***Related Work.*** Several data augmentation libraries exist for image tasks. General purpose image augmentation libraries include Albumentations [3], Aug-

* Supported by Sparkfish LLC.
[4] https://github.com/sparkfish/augraphy

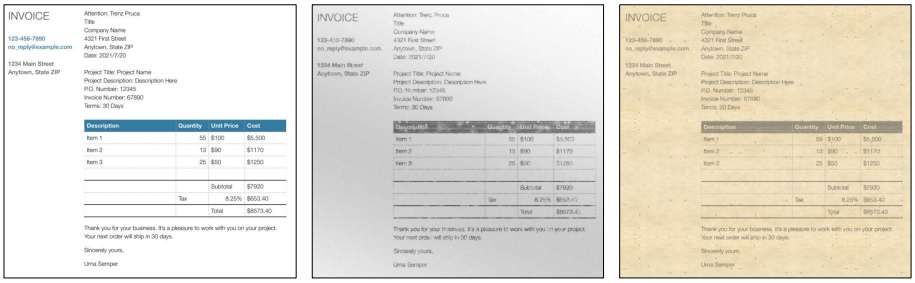

**Fig. 1.** Example input image (left) with two augmented versions (middle and right).

mentor [1], Augly [2], and imgaug [5]. Augmentation techniques from these general purpose libraries include rotations, translations, warps, and color transformations, yet none of these libraries provide augmentations targeted at imitating the types of transformations seen in document analysis corpora.

A notable exception is DocCreator [4], which is a document synthesizing tool that provides several transformation strategies as part of its synthesis pipeline. DocCreator's augmentations target imitating artefacts seen in historical (e.g., ancient or medieval) manuscripts, and hence do not address more modern causes of noise, such as noise introduced by document scanners. DocCreator is written in C++ and is meant to be used as a what-you-see-is-what-you-get tool; with no scripting or API interface, it is not easily amenable to being used in broader machine learning model development pipelines. In contrast, Augraphy is written in Python and has a simple interface to allow for seamless use with other Python libraries and data pipelines.

***The Augraphy Package.*** Augraphy is a lightweight Python package. It is registered on the Python Package Index (PyPI) and can be installed using `pip install augraphy` . Augraphy requires only a few other commonly-used Python scientific computing or image handling packages in order to run, such as NumPy and Pillow. Augraphy has been tested on Windows, Linux, and Mac computing environments. Listing 1 shows how easy it is to get Augraphy up and running to create a straightforward augmentation pipeline and apply it to an image.

```
import augraphy; import cv2
pipeline = augraphy.default_augraphy_pipeline()
img = cv2.imread("image.png")
data = pipeline.augment(img)
augmented = data["output"]
```

**Listing 1.1.** Transforming an image with Augraphy.

Examples of output generated by Augraphy can be seen in Figure 1, which shows augmentations mimicking low printer ink and fuzzy, low-resolution text (middle image), and other paper surfaces (right image). We also show several of Augraphy's individual augmentation features in Figure 2. Importantly, these in-

dividual augmentation strategies can be composed together in an augmentation pipeline to create even more realistic looking, noisy output.

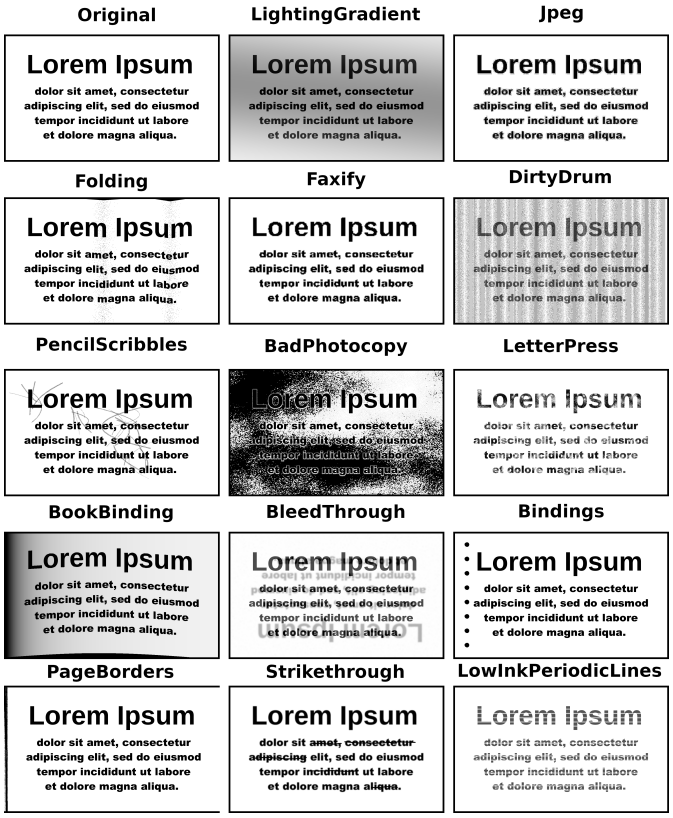


**Fig. 2.** Various individual augmentation types available in Augraphy. These individual augmentations can also be composed together.

***Qualitative Case Study: Document Denoising.*** In this section we highlight the effectiveness of Augraphy by creating a new evaluation set for the task of document denoising. Document denoising is the task of removing noisy artifacts from a document image, and one recent dataset that has emerged for this task is the NoisyOffice dataset [6], which itself generated noisy versions of clean documents by applying several augmentations. However, both the original documents and the augmentations in NoisyOffice are quite limited, so it is natural to wonder if a model trained on NoisyOffice data can generalize to more diverse data inputs for the denoising task.

In Figure 3 we show example test inputs (left) to a convolutional autoencoder, which we trained on the NoisyOffice dataset. The model's outputs are

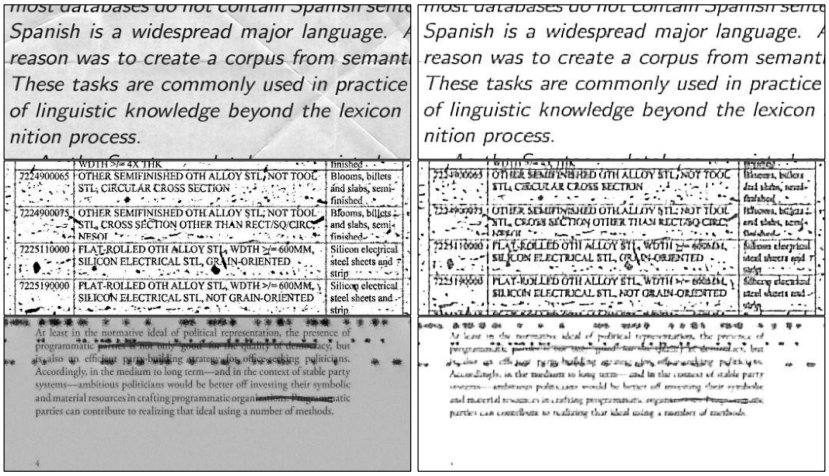


**Fig. 3.** Inputs (left) and outputs (right) to a denoising model. A NoisyOffice [6] sample is shown in the top row. Augraphy samples are shown in the bottom two rows.

shown on the right side of Figure 3. We see that the model does well on the NoisyOffice input (top row), but underperforms on data that was augmented by Augraphy (bottom two rows), showing that Augraphy's augmentations are effective at producing challenging testing data for analyzing the robustness of denoising models.

## 3 Conclusion

This paper introduces Augraphy, a new data augmentation package for document analysis tasks.

## References

1. Bloice, M., Roth, P., Holzinger, A. Biomedical image augmentation using Augmentor. *Bioinformatics*, 35(21):4522-4524, 2019.
2. Papakipos, Z., Bitton, J. AugLy: Data Augmentations for Robustness. *arXiv preprint*, 2022.
3. Buslaev, A., Iglovikov, V., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.
4. 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):62, 2017.
5. Jung, A., et al. Imgaug. 2020.
6. Castro-Bleda, M., 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, 2020.