TorMentor: Deterministic dynamic-path, data augmentations with fractals

Anguelos Nicolaou[1], Vincent Christlein[2], Edgar Riba[3], Jian Shi[3], Georg Vogeler[1], Mathias Seuret[2]
[1]University of Graz, [2]Friedrich-Alexander-Universität Erlangen-Nürnberg, [3]kornia.org

Abstract

We propose the use of fractals as a means of efficient data augmentation. Specifically, we employ plasma fractals for adapting global image augmentation transformations into continuous local transforms. We formulate the diamond square algorithm as a cascade of simple convolution operations allowing efficient computation of plasma fractals on the GPU. We present the TorMentor image augmentation framework that is totally modular and deterministic across images and point-clouds. All image augmentation operations can be combined through pipelining and random branching to form flow networks of arbitrary width and depth. We demonstrate the efficiency of the proposed approach with experiments on document image segmentation (binarization) with the DIBCO datasets. The proposed approach demonstrates superior performance to traditional image augmentation techniques. Finally, we use extended synthetic binary text images in a self-supervision regiment and outperform the same model when trained with limited data and simple extensions.

1. Introduction

In the era of deep learning, computer vision-based neural networks need extreme quantities of annotated data. This has led to Image Data Augmentation (IDA) being employed in any state-of-the-art training pipeline to various degrees. Specialised domains such as Document Image Analysis (DIA) or medical imaging usually are much more restricted in the size of the available datasets making data augmentation an integral part of the training pipeline. When the use case exceeds typical image classification such as image segmentation, object detection, landmark detection etc., data augmentation becomes more challenging as the ground truth needs to be modified along with the input sample.

Figure 1. An MS-COCO [11] sample and its ground truth augmented through TorMentor along with the automatically generated validity mask

Image data augmentation is the introduction of noise in data samples, essentially creating new plausible samples in order to train a ML model with more data. This causes the model to become invariant to the specific kind of noise. Each noise operation of an IDA tries to mimic distortions that could occur naturally in an image and thus defining and tuning it requires domain-specific knowledge.

IDA operators can also be employed in self-supervision scenarios, test-time augmentation, and ablation studies.

One could argue that defining all the invariants that need to be learned, is essentially defining the problem itself.

2. Plasma Fractals for Image Augmentation

Elastic transforms have proved quite important for augmenting data, especially textual images [24] but they require employing large Gaussian filters. The diamond-square algorithm was proposed [5] in 1982 as an algorithm for generating random height-maps for computer graphics. The algorithm can be used to generate cloud-looking fractals called plasma-fractals. While the popular algorithm requires sparse memory access, and square images of size $(2^n+1)\times(2^n+1)$ for any $n > 2$, we propose a version of the algorithm that can be implemented with convolutions.

In algorithm 1, the python-numpy inspired pseudo-code description of the proposed algorithm can be seen. The algorithm consists of two steps: the first **ONEDS** is applying a diamond and square step cascade on an existing image and some weighted random pixels effectively quadrupling its resolution. The second step **DS** invokes **ONEDS** recursively in order to grow the plasma fractal to an arbitrary size. It should be pointed out that in line 19, plasma could be initialized to any image of odd dimension sizes larger than 3, but if a dimension is initialized with a size grater than 3, then it will be missing some low frequencies. The $roughness$ parameter, through e, controls the ratio between the existing pixels and random pixels added. In practice other than the desired resolution (recursion steps), it is the only parameter controlling the fractal generation and can be perceived

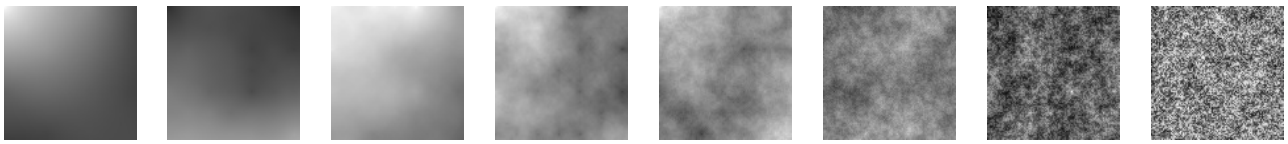

Figure 2. Generated plasma fractals 129×129 pixels with a roughness of 0.2 to 0.9

Algorithm 1 Convolutional Diamond Square

procedure ONEDS($plasma$, e)
  $dfilter \leftarrow [[0.25, 0, 0.25], [0, 0, 0], [0.25, 0, 0.25]]$
  $sfilter \leftarrow [[0, 0.25, 0], [0.25, 0, 0.25,], [0, 0.25, 0]]$
  $oldw, oldh \leftarrow size(plasma)$
  $w, h \leftarrow (oldw - 1) * 2 + 1, (oldh - 1) * 2 + 1$
  $border \leftarrow ones(w, h)$
  $border[1 : -1, 1 : -1] \leftarrow ones(w - 2, h - 2)$
  $rnd \leftarrow random(w, h)$
  $dilated \leftarrow zeros(w, h)$
  $dilated[:: 2, :: 2] \leftarrow plasma + random(w, h) * e$
  $d \leftarrow dilated \circledast dfilter$
  $dc \leftarrow ispositive(d)$
  $dilated \leftarrow dilated + (1 - e) * d * dc + rnd * dc * e$
  $s \leftarrow dilated \circledast sfilter * border$
  $sc \leftarrow ispositive(s)$
  return $dilated + (1 - e) * d * sc + sc * rnd * e$
procedure DS($steps$,$roughness$)
  $e \leftarrow 1$
  $plasma \leftarrow rand(3, 3)$
  for $i \leftarrow 1, steps$ **do**
    $e \leftarrow e * roughness$
    $plasma \leftarrow OneDS(plasma, e)$
  return $plasma$

as a parameter controlling whether low or high frequencies will dominate. In Fig. 2, the resulting plasma fractals for different roughness values can be seen.

Other than the fact that **ONEDS** is differentiable with respect to $plasma$ and can be used as a valid neural network layer, the proposed algorithm can be computed efficiently on the GPU with PyTorch [13]. In Fig. 3, the performance with respect to the output image size is compared between a C++ implementation [9], a generic python-numpy version [7], and the proposed method in CPU and GPU mode. They were tested for resolutions of 65×65 up to 8193×8193. The proposed method on CPU (single thread) converges to being twice as slow as the C++ version while the GPU version is more than an order of magnitude faster.

3. The TorMentor augmentation framework

The extensive use of plasma-fractals to provide localized augmentation operations was coupled with some other

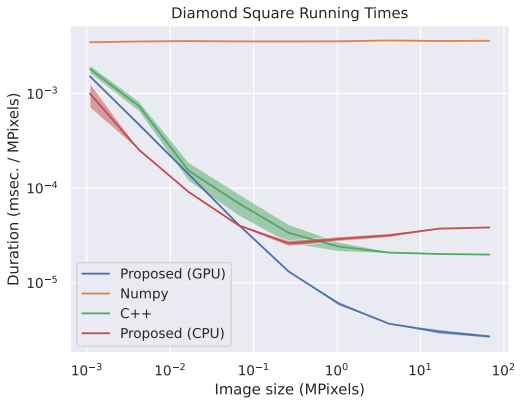

Figure 3. Proposed Diamond Square Benchmark.

design patterns into a coherent image data augmentation framework that called TorMentor. [1]

At the heart of the design lies the concept of ventral operations, which move things in the image, and dorsal augmentations, which modify pixels where they are [3]. Ventral operations affect images, masks, and points while dorsal operations affect only images, this allows for the creation of deep augmentation pipelines that are applicable at the same time on image pointclouds and masks.

An augmentation operation is a class that possess the random generation parameters, instances of that class are lightweight objects containing a random number generator seed as their only data member allowing each instance of the operation to be deterministic and serializable. Each augmentation class must implement a method sampling the random distributions that specify the augmentation parameters given the size of the data and a functional method that applies it on a sampling field if it is a ventral operation or directly on the image if it is a dorsal operation.

Finally, other than ventral and dorsal augmentations, a random choice between several augmentations, a cascade of several augmentations, and an identity augmentation are also defined. These preserve determinism for data of the same (image) size and allow for combinations of elementary augmentations into ones of arbitrary complexity, *e.g.* an operation known as random flip, can be implemented in tormentor as a choice between a vertical flip, a horizontal flip,

[1] `https://github.com/anguelos/tormentor`.

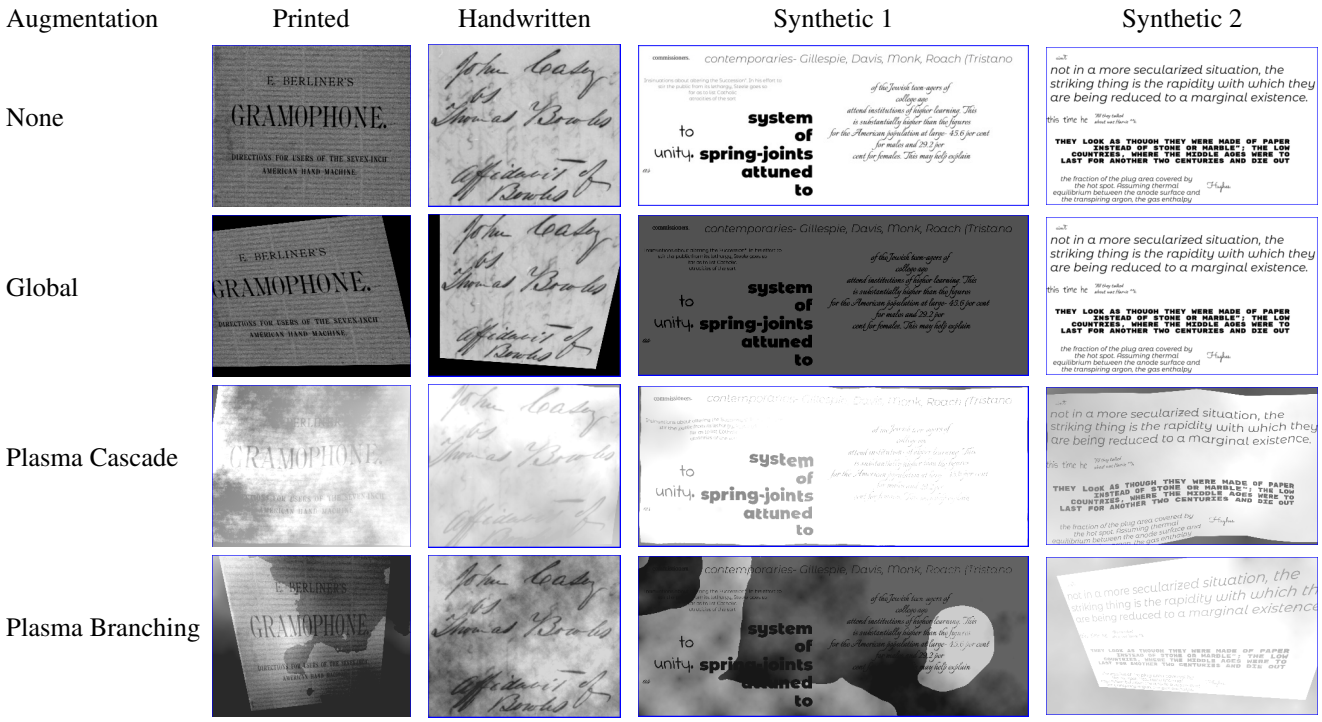

Figure 4. Samples augmented under the three augmentation regiments in the document segmentation experiment.

a cascade of both, or an identity. In essence, a TorMentor's data augmentation regiment defines a flow network where the input data are the source, the augmented data are the sink, and every augmentation instance combined with a specific input image size defines a random path from the source to the sink.

As with every other augmentation class, choices have their own categorical distributions from which they sample and these are easily tuned to control the probability of every branch in the graph.

TorMentor builds on top of Kornia [22] and thus it is differentiable and a PyTorch layer. This allows to employ augmentations anywhere inside an end-to-end model training regiment including between the generator and discriminator in a Generative Adversarial Network (GAN).

4. Document Image Segmentation Experiments

We performed several experiments for document image segmentation (binarization). Our intention was not to compare the models we employed to the state of the art but rather obtain insights about generalization abilities of a straightforward model, trained under different augmentation strategies. Document Image Segmentation (DIS) ground-truthing is extremely labor intensive and in the case of historical documents, paleographers might be needed. In most cases, training data are limited to at most a few pages.

We used the whole range of the DIBCO datasets that were publicly available [6, 14–21]. In all experiments we performed, either we used DIBCO2009 as a train set or no "real" data at all. We used as a metric for each page, the FScore, the harmonic mean of precision and recall as defined in [6], and averaged it across all samples in each dataset regardless of the sample size. While DIBCO has different tracks and modalities in several years, we averaged the FScore across all samples in all cases among the dataset. We also created a synthetic dataset with 30 images containing black text on various fonts and sizes rendered as small text-blocks on white pages; these images are employed in a self supervision task being both input and output but augmenting the input. While synthetic data has been used for text image classification [8], to the authors knowledge, they have not been used for text-image segmentation. In Fig. 4, DIBCO and synthetic samples can be seen on the first row while rows 2, 3, and 4 demonstrate how they are augmented through the different augmentation regiments.

We chose a lightweight segmentation model: iUNets [4] which are reversible [10] UNets [23]. Networks such as the UNet are quite small from the perspective of parameters but the memory they require is vast and proportional to the size of the input image. Reversible networks allow to economize memory needed for caching the forward pass by an order of magnitude allowing to use the UNet in a fully convolutional way without the use of patches and their associated complications such as stitching and border artifacts. As the

Table 1. Document Image Segmentation Experiment

Method	Augmentation	FScore % on DIBCO Dataset									
		2009	2010	2011	2012	2013	2014	2016	2017	2018	2019
DIBCO Participant centile 100 %		91.24	91.78	93.33	92.85	92.7	96.88	88.72	91.04	89.37	72.88
DIBCO Participant centile 75 %		86.17	87.98	91.68	90.035	89.78	94.17	88.14	86.38	82.91	62.76
DIBCO Participant centile 50 %		84.57	85.06	88.99	89.38	89.06	89.51	87.61	83.10	78.46	57.66
Otsu	–	78.60	85.43	82.10	75.07	80.03	86.59	77.74	51.46	51.455	47.83
Thr. Oracle	–	87.86	87.70	85.88	87.81	87.61	90.66	85.15	88.74	88.741	81.13
BIUnet	None	–	90.52	87.86	**89.92**	**90.42**	91.39	84.62	87.59	66.43	62.03
BIUnet	Global	–	75.91	86.07	79.50	82.76	82.05	79.71	82.75	59.81	51.35
BIUnet	Plasma Cascade	–	**91.92**	85.52	88.83	89.41	**94.03**	85.00	86.30	**83.42**	63.62
Synth BIUnet	Plasma Branching	87.85	87.78	**88.45**	87.35	89.34	90.39	**89.07**	**89.69**	82.82	**69.84**

point of the experiments was not to compete with the state of the art in DIS, we chose a light-weigh architecture with 1,597,698 parameters, we also omitted connected component post-processing heuristics that typically improve the method outputs.

We also measured the performance of two global thresholds as reference, Otsu's threshold [12] and the Threshold oracle [1].

Finally for reference, we provide quantiles 100 % (best), 75 %, and 50 %(median) from all participating methods as reported by the competition organisers.

All images were converted to grayscale and three augmentation regiments were defined: a traditional one, called *Global*, selecting randomly between a perspective transformation, a brightness modification, and Gaussian additive noise. A cascade of three plasma-based augmentations: a plasma-brightness operation followed by plasma-wrapping, followed by a linear color space manipulation. And a *Branching* regiment where a cascade of several choices between mostly plasma-based augmentations was performed.

The results presented in Tab. 1 allow several observations. The *Global* regiment performs worse than no augmentation at all. The model trained on synthetic images achieves the best performance in 4 out of 9 datasets, compared to competition participants it would rank above or near the top 25 % and in recent years where the data became more challenging even higher.

Even when training on 10 images and without augmentation, the BIUnet could not overfit the data as the images were much larger than its receptive field demonstrating the merit of training fully convolutionally instead of patched-based. The DIBCO datasets are quite different from year to year and demonstrate very well that there is no such thing as the overall best DIS method.

5. Conclusion Discussion and Future Work

TorMentor is developed as a technology demonstrator that enables the encoding of domain-specific expert knowledge to generate plausible distortions. While defining an augmentation is constrained by its formalism, the fact that an augmentation is defined once and automatically applied on images, pointclouds, etc., makes it much easier to maintain than popular frameworks such as Albumentations [2] that redefine operations for every kind of data. The choice and cascade augmentations are also created through the '^' and '|' operators respectively allowing for readable and dense pythonic constructs of complicated augmentation graphs. The fact that it could be used to successfully train a UNet strictly on augmented ground truth maps proves its potential.

In future work, we intend to replace the visual augmentation tuning tool seen in Fig. 5 with adversarial training to mimic existing data. Extensive experiments allowing a quantitative estimation of the question "How many fewer samples do I need to annotate if I properly tune my data augmentation?".

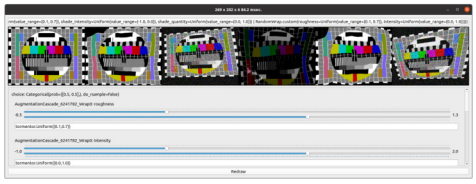

Figure 5. Tool for tuning Tormentor parameters visually

Acknowledgements

This work has been supported by the ERC Advanced Grant 101019327 'From Digital to Distant Diplomatics'. We would also like to thank Dmytro Mishkin and Aikaterini Symeonidi for their insights and support.

References

[1] Jonathan T Barron. A generalization of otsu's method and minimum error thresholding. In *European Conference on Computer Vision*, pages 455–470. Springer, 2020. 4

[2] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. 4

[3] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. Why do self-supervised models transfer? investigating the impact of invariance on downstream tasks. *arXiv preprint arXiv:2111.11398*, 2021. 2

[4] Christian Etmann, Rihuan Ke, and Carola-Bibiane Schönlieb. iunets: learnable invertible up-and downsampling for large-scale inverse problems. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020. 3

[5] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Communications of the ACM*, 25(6):371–384, 1982. 1

[6] Basilis Gatos, Konstantinos Ntirogiannis, and Ioannis Pratikakis. Icdar 2009 document image binarization contest (dibco 2009). In *2009 10th International conference on document analysis and recognition*, pages 1375–1382. IEEE, 2009. 3

[7] Christian Hill. Cloud images using the diamond-square algorithm. `https://scipython.com/blog/cloud-images-using-the-diamond-square-algorithm/`, 2016. last access: 24 March 2022. 2

[8] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 3

[9] Julien Jean. Diamondsquare. `https://github.com/ROUKIEN/DiamondSquare`, 2016. last access: 24 March 2022. 2

[10] Sil C. van de Leemput, Jonas Teuwen, Bram van Ginneken, and Rashindra Manniesing. Memcnn: A python/pytorch package for creating memory-efficient invertible neural networks. *Journal of Open Source Software*, 4(39):1576, 7 2019. 3

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[12] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979. 4

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[14] I Pratikakis et al. Icdar 2011 document image binarization contest. In *ICDAR*. ICDAR, 2010. 3

[15] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. H-dibco 2010-handwritten document image binarization competition. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 727–732. IEEE, 2010. 3

[16] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. Icfhr 2012 competition on handwritten document image binarization (h-dibco 2012). In *2012 international conference on frontiers in handwriting recognition*, pages 817–822. IEEE, 2012. 3

[17] Ioannis Pratikakis, Basilis Gatos, and Konstantinos Ntirogiannis. Icdar 2013 document image binarization contest (dibco 2013). In *2013 12th International Conference on Document Analysis and Recognition*, pages 1471–1476. IEEE, 2013. 3

[18] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. Icfhr2016 handwritten document image binarization contest (h-dibco 2016). In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 619–623. IEEE, 2016. 3

[19] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. Icdar2017 competition on document image binarization (dibco 2017). In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1395–1403. IEEE, 2017. 3

[20] Ioannis Pratikakis, Konstantinos Zagoris, Panagiotis Kaddas, and Basilis Gatos. Icfhr2018 competition on handwritten document image binarization contest (h-dibco 2018). In *International Conference on Frontiers in Handwriting Recognition*, pages 1–1, 2018. 3

[21] Ioannis Pratikakis, Konstantinos Zagoris, Xenofon Karagiannis, Lazaros Tsochatzidis, Tanmoy Mondal, and Isabelle Marthot-Santaniello. Icdar 2019 competition on document image binarization (dibco 2019). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1547–1556, 2019. 3

[22] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020. 3

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[24] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003. 1