deepCR: Cosmic Ray Rejection with Deep Learning

Keming Zhang*
Department of Astronomy
University of California at Berkeley
Berkeley, CA 94720
kemingz@berkeley.edu

Joshua S. Bloom
Department of Astronomy
University of California at Berkeley
Berkeley, CA 94720
joshuabloom@berkeley.edu

Abstract

We present deepCR, a deep learning based method for cosmic ray (CR) identification and replacement in astronomical imaging data. We train and evaluate models on Hubble Space Telescope ACS/WFC images of sparse extragalactic fields, globular clusters, and resolved galaxies. At a false positive rate of 0.5%, deepCR achieves close to 100% detection rates in both extragalactic and globular cluster fields, and 91% in resolved galaxy fields, which is a significant improvement over the current state-of-the-art LACosmic. Compared to a parallel CPU implementation of LACosmic, deepCR CR mask predictions run up to 6.5 times faster on CPU and 90 times faster on a single GPU. For CR inpainting, the mean squared errors of deepCR predictions are 20 times lower in globular cluster fields, 5 times lower in resolved galaxy fields, and 2.5 times lower in extragalactic fields, compared to the best performing non-neural technique tested. This framework was first presented in Zhang and Bloom [2019] along with an open-source implementation in Python.[2]

1 Introduction

Astronomical imaging and spectroscopy data are frequently contaminated by cosmic rays (CR), which are high-energy charged particles that create excess flux in the pixels hit when passing through solid state detectors. CR artifacts must be identified and either masked or replaced, before further analysis could be done with the image. It is straightforward to identify these artifacts by taking a sequence of images of the same astronomical target. In such cases, a median image could be calculated from the individual exposures, which is effectively free of cosmic ray randomly distributed in each individual frame. Each one of the single images is then aligned and compared with the median image to identify CR artifacts (cf. Windhorst et al. 1994).

However, many cases call for algorithms to detect CRs in single images. For example, when detector read-out times are non-negligible, or when sources of interest are transient, causing them to be excluded from the median image, CR rejection with multiple exposures is sub-optimal, or in the latter case, infeasible. In single images, CR artifacts can be distinguished from astronomical objects such as stars and galaxies by the sharpness of their edges, because photons from astronomical targets are smeared out when passing through Earth's atmosphere and the telescope optics, which doesn't affect CR. The current state-of-the-art method is LACosmic [van Dokkum, 2001], which identifies CR artifacts by Laplacian edge detection. However, LACosmic has two drawbacks. First, it creates many false detections in under-sampled images where stars look as sharp as CRs, such as the case of Hubble Space Telescope imaging. On the other hand, large CRs with smoother edges tend to be missed. In addition, LACosmic is computation heavy, requiring substantial run-time even for the most recent parallel implementation, astroscrappy [McCully et al., 2018]. In this paper, we present a

*kemingz@berkeley.edu
[2]https://github.com/profjsb/deepCR

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

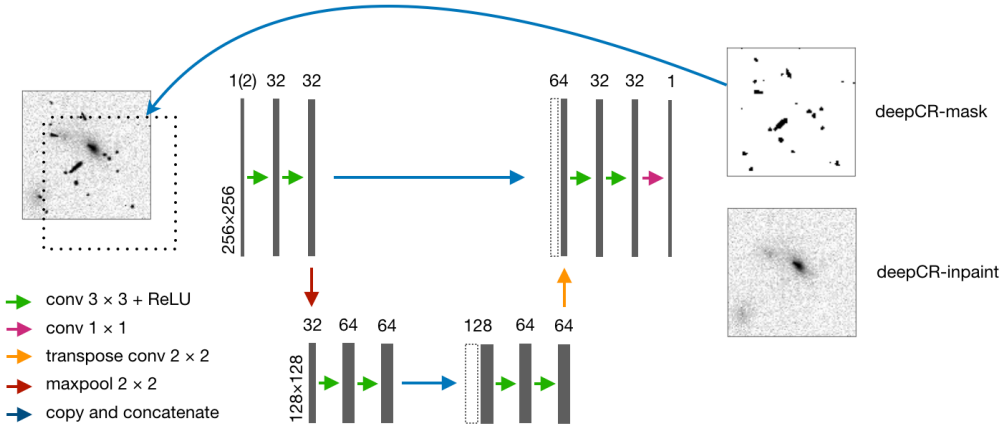

Figure 1: Neural network architecture of `deepCR`. Feature maps are represented by gray boxes while the number of channels and feature map dimensions are indicated on the top of and to the left of each feature map, respectively. Unfilled boxes to the right of blue arrows represent feature maps directly copied from the left, which are to be concatenated with the adjacent feature map.

summary of Zhang and Bloom [2019] which presented a novel deep learning based method named `deepCR` for cosmic ray identification and replacement that significantly improved the aforementioned issues. The framework and trained models are presented as an open-source project in `Python`.

2 Model Architecture

`deepCR` is formulated as a sequence of two independent deep neural networks, `deepCR`-mask and `deepCR`-inpaint: `deepCR`-mask, given an input image, predicts a probabilistic map of each pixel being affected by CRs, and `deepCR`-inpaint predicts what the pixel values would have been had they not been affected by CRs. The probabilistic CR map is then turned into a binary CR mask by setting a threshold. The basic architecture of both `deepCR`-mask and `deepCR`-inpaint is a modification of the UNet [Ronneberger et al., 2015], which is an encoder-decoder CNN with skip connections between each depth of the encoder and decoder (Figure 1). The skip connections are important in that they allow the decoder to have direct access to pixel level features such as edges, in addition to semantic features attained from the network bottleneck, such as the approximate location of CR artifacts.

Binary cross-entropy loss is used in training `deepCR`-mask,

$$\mathcal{L}_{\mathrm{F}} = \mathbb{E}[\mathrm{M} \times \log(1 - \mathrm{F}(\mathrm{X})) + (1 - \mathrm{M}) \times \log(\mathrm{F}(\mathrm{X}))], \tag{1}$$

where F refers to the `deepCR`-mask, M the ground truth CR mask, and $\mathbb{E}$ is the expectation with respect to training data. For `deepCR`-inpaint, mean squared error (MSE) loss is taken between the predicted pixels under the inpainting mask ($\mathrm{M_I}$) and the ground truth values. Instead of using the ground truth cosmic ray masks (M) as the inpainting mask, which would require having ground truth pixel values of CR affected pixels, we create custom inpainting mask for each image by sampling and adding 1-9 CR masks from the rest of the dataset. As long as the $\mathrm{M_I}$ do not overlap with M, we have the ground truth pixel values *a priori*. This naturally allows for data augmentation. Although using median images as ground truth allow for higher S/N training targets, Lehtinen et al. [2018] showed that training on noisy targets is comparable to training on clean targets both in terms of convergence speed and final performance. The inpainting loss is then formulated as,

$$\mathcal{L}_{\mathrm{G}} = \mathbb{E}[(\mathrm{G}(\mathrm{X}, \mathrm{M_I}) \circ \mathrm{M_I} \circ (1 - \mathrm{M}) - \mathrm{X} \circ \mathrm{M_I} \circ (1 - \mathrm{M}))^2], \tag{2}$$

where $\circ$ denotes element-wise multiplication. Here, G refers to `deepCR`-inpaint, X the input image, M the cosmic ray mask, and $\mathrm{M_I}$ the inpainting mask. Element-wise multiplication by $(1-\mathrm{M})$ ensures that the loss is not computed for parts of the inpainting mask that overlap with CR artifacts. Loss is also not computed outside of the inpainting mask; otherwise `deepCR`-inpaint is forced to also learn an identity function for those regions, which is both unnecessary and degrades model performance at a fixed model capacity.

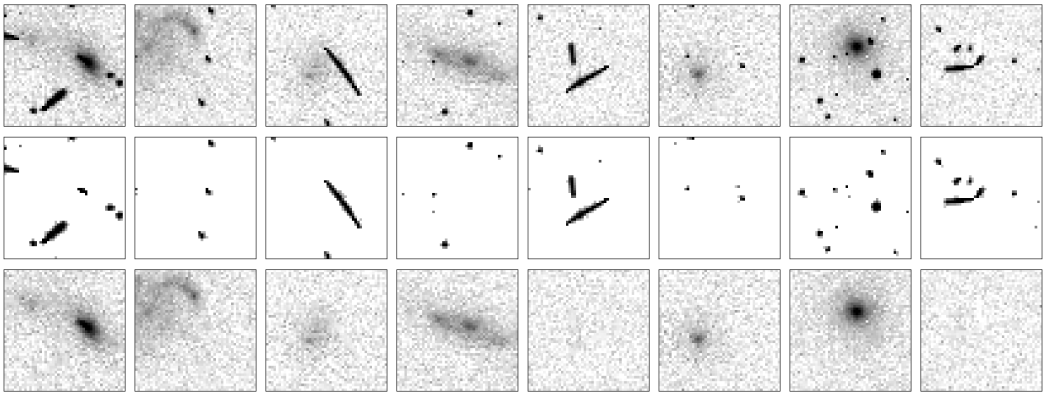

Figure 2: Top row shows details of cosmic ray artifacts in Hubble image of galaxy cluster EMACSJ2316.6. Middle panel shows cosmic ray as identified by `deepCR`-mask-2-32. Bottom panel shows images with artifact pixels replaced by `deepCR`-inpaint-3-32.

Training `deepCR`-mask and `deepCR`-inpaint requires a set of training data consisting of cosmic ray affected images (X) and ground truth cosmic ray masks (M). We use the `astrodrizzle` data pipeline [Hack et al., 2012] to construct our training set from the Hubble legacy archive, which includes multiple exposures of the same fields that provides the basis for us to derive accurate ground truth cosmic ray labels. We also create bad pixel and saturation masks for which we do not backpropagate gradients. Finally, we segment images and masks into chunks of 256×256, which results in a training set of 8190 image stamps and a test set of 3360 image stamps. Test set images are chosen to have different target fields than the training images and a wide range of exposure times.

3 Results

We train and evaluate models with different hyperparameters to examine the trade off between accuracy and run-time, both important metrics for us. Each model variant is labeled with network depth and number of channels at the initial layer, e,g., deepCR-2-32 would be a depth-2 network with 32 channels at the first convolution layer (same as Fig. 1). During training, sky background levels are augmented by adding up to 3 times and subtracting up to 0.9 times the original background flux. Such a data augmentation scheme helps prevent over-fitting on the discrete and limited range of exposure times seen in the training set and improves test time performance, especially for exposure times unseen during training.

We benchmark deepCR against baseline models separately on three categories of test data: extragalactic field, globular cluster, and local group galaxies for which the stellar population is well resolved; the density of astronomical sources, which are also sources of confusion during CR identification, increases in that order.

3.1 Cosmic Ray Identification: `deepCR`-mask

Figure 3 shows CR detection ROC curves of `deepCR`-2-32 (hereafter `deepCR`) and the baseline model, `LACosmic`. Since imaging noise makes it impossible to create objective ground truth cosmic ray masks, the characteristic way by which `astrodrizzle` create cosmic ray masks could be learned by `deepCR` during training but not `LACosmic`. To make a fairer comparison, we calculated additional ROC curves with true positive rates evaluated from 3×3 kernel dilated predictions, which results in ROC curves (shown in red) that does not penalize a model for not finding the exact cosmic ray shape matching the ground truth. This removes any advantage `deepCR` may have gained.

As seen in Figure 3, `deepCR` achieved nearly 100% true positive rates (TPR) for both extragalactic fields and globular cluster fields and $> 90\%$ for resolved galaxy fields, at a false positive rate (FPR) of 0.5%. While mask dilation increased true detection rate drastically for `LACosmic`, `deepCR` still leads a significant edge in all three types of fields, especially in resolved galaxy fields and at very low FPRs. Quantitative comparison is presented in Table 1, which also lists runtime performance. We

Table 1: Cosmic ray detection true positive rates (TPR) at a fixed false positive rate (FPR) of 0.5%. TPRs in parenthesis show detection rates after mask dilation, as described in the main text. The last two columns show runtime required to predict CR masks for one hundred 256×256 images.

Model	extragalactic field TPR (0.5%)	globular cluster TPR (0.5%)	resolved galaxy TPR (0.5%)	runtime [sec] CPU	GPU
`deepCR-2-4`	94.0% (96.9%)	96.2% (98.7%)	80.6% (89.7%)	**1.4**	**0.1**
`deepCR-2-32`	**98.7%** (99.2%)	**99.5%** (99.7%)	**91.2%** (93.7%)	7.9	0.2
`LACosmic`	69.5% (95.4%)	73.9% (95.2%)	53.4% (77.4%)	9.0	n/a

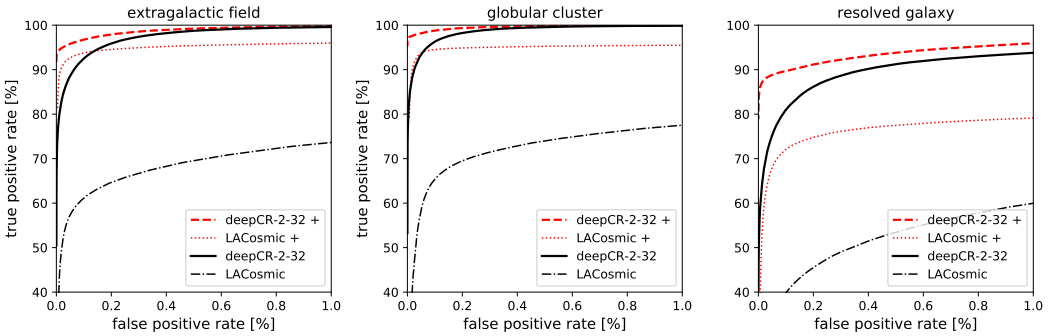

Figure 3: Receiver operating characteristic (ROC) curves of cosmic ray detection. Line shapes and the corresponding methods are indicated in the legends to the lower right. Red ROC curves (labeled with + in the legend) differ from black ones in that true positive rates are calculated from a 3×3 kernel dilated predicted mask, with false positive rates kept the same at the non-dilated values. The red curves therefore does not penalize a model for not finding the exact cosmic ray shape matching the ground truth.

find that the smaller variant, `deepCR-2-4`, runs on one Nvidia Titan X GPU around 90 times faster than the parallel CPU implementation of `LACosmic` (`astroscrappy`), and 6.5 times faster on CPU. CPU implementation of each model is well parallelized and fully utilized 4 cores on an Intel Xeon CPU.

3.2 Replacing Masked Pixels: `deepCR`-inpaint

We evaluate `deepCR`-inpaint against non-neural baseline models including biharmonic interpolation and masked median sampling (5×5 filter; `medmask`) that is used by `LACosmic`. Evaluation is done on 9600 images stamps randomly chosen from the test set, with different generated inpainting masks applied to each one of them. Inpainting performances and runtimes are listed in Table 2, which shows that `deepCR` achieves MSEs 20 times lower in globular cluster fields, 5 times lower in resolved galaxy fields, and 2.5 times lower in extragalactic fields, compared to the best performing non-neural method we tested, biharmonic inpainting. The superior MSE performance of the `deepCR`-inpaint over generic interpolation schemes is not surprising, given that our model is trained on the semantically constrained domain of astronomical images. Our model is also drastically faster than non-neural models on GPU. We do caution that further testing with astronomically meaningful metrics (such as photometric fidelity) is required, before the inpainted pixels can be used for science.

4 Summary

We have presented a summary of the main results from `deepCR` [Zhang and Bloom, 2019], a novel deep learning based method for cosmic ray rejection, and showed that after proper training it outperforms `LACosmic` in terms of both speed and accuracy in mask prediction and image inpainting, on HST ACS/WFC imaging data. We also expect `deepCR` to work well on ground-based or spectroscopic data which are more homogeneous. To facilitate the use of the `deepCR` framework in real-world data reduction pipelines, we have made our code with the ACS/WFC F606W trained models available as

Table 2: Image inpainting metrics of `deepCR`-inpaint as compared to baseline models. Runtime refer to time required to inpaint one hundred 256×256 images. MSE is mean squared error.

Model	extragalactic MSE	globular cluster MSE	resolved galaxy MSE	runtime [sec] CPU	GPU
deepCR-2-32	**0.012**	0.034	0.503	7.5	**0.2**
deepCR-3-32	**0.012**	**0.033**	**0.479**	12.7	0.3
medmask	0.105	1.511	5.946	**1.0**	n/a
biharmonic	0.041	0.669	2.578	109.5	n/a

an open source project, and we encourage the community to contribute by training additional models that allow `deepCR` to be used in a wide range of detector configurations.

Acknowledgments

It is a pleasure to thank Dan Weisz for helpful conversations and Stéfan van der Walt both for his insights on image inpainting and for comments on a draft of this manuscript. Keming Zhang thanks the LSSTC Data Science Fellowship Program, which is funded by LSSTC, NSF Cybertraining Grant #1829740, the Brinson Foundation, and the Moore Foundation; his participation in the program has benefited this work. This work is supported by a Gordon and Betty Moore Foundation Data-Driven Discovery grant.

References

W. J. Hack, N. Dencheva, A. S. Fruchter, A. Armstrong, R. Avila, S. Baggett, E. Bray, M. Droettboom, M. Dulude, S. Gonzaga, N. A. Grogin, V. Kozhurina-Platais, R. A. Lucas, J. Mack, J. MacKenty, L. Petro, N. Pirzkal, A. Rajan, L. J. Smith, C. Sontag, and L. Ubeda. AstroDrizzle: More than a New MultiDrizzle. In *American Astronomical Society Meeting Abstracts #220*, volume 220 of *American Astronomical Society Meeting Abstracts*, page 135.15, May 2012.

J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2noise: Learning Image Restoration without Clean Data. In *International Conference on Machine Learning*, pages 2965–2974, July 2018. URL `http://proceedings.mlr.press/v80/lehtinen18a.html`.

C. McCully, S. Crawford, G. Kovacs, E. Tollerud, E. Betts, L. Bradley, M. Craig, J. Turner, O. Streicher, B. Sipocz, T. Robitaille, and C. Deil. astropy/astroscrappy: v1.0.5 zenodo release, Nov. 2018. URL `https://doi.org/10.5281/zenodo.1482019`.

O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI*, 2015. URL `http://arxiv.org/abs/1505.04597`. arXiv: 1505.04597.

P. G. van Dokkum. Cosmic-Ray Rejection by Laplacian Edge Detection. *Publications of the Astronomical Society of the Pacific*, 113:1420–1427, Nov. 2001. ISSN 0004-6280. doi: 10.1086/323894. URL `http://adsabs.harvard.edu/abs/2001PASP..113.1420V`.

R. A. Windhorst, B. E. Franklin, and L. W. Neuschaefer. Removing Cosmic-Ray hits from Multi-Orbit HST wide field camera images. *Publ. Astro. Soc. Pac.*, 106:798, July 1994.

K. Zhang and J. S. Bloom. deepCR: Cosmic Ray Rejection with Deep Learning. *arXiv e-prints*, 1907:arXiv:1907.09500, July 2019. URL `http://adsabs.harvard.edu/abs/2019arXiv190709500Z`.