Partial-Attribution Instance Segmentation for Astronomical Source Detection and Deblending

Ryan Hausen  
Department of Computer Science and Engineering  
University of California, Santa Cruz  
Santa Cruz, CA 95064  
rhausen@ucsc.edu

Brant Robertson  
Department of Astronomy and Astrophysics  
University of California, Santa Cruz  
Santa Cruz, CA 95064  
brant@ucsc.edu

Abstract

Astronomical source deblending is the process of separating the contribution of individual stars or galaxies (sources) to an image comprised of multiple, possibly overlapping sources. Astronomical sources display a wide range of sizes and brightnesses and may show substantial overlap in images. Astronomical imaging data can further challenge off-the-shelf computer vision algorithms owing to its high dynamic range, low signal-to-noise ratio, and unconventional image format. These challenges make source deblending an open area of astronomical research, and in this work, we introduce a new approach called Partial-Attribution Instance Segmentation that enables source detection and deblending in a manner tractable for deep learning models. We provide a novel neural network implementation as a demonstration of the method.

1 Introduction

Astronomical images can contain tens of thousands of stars and galaxies (sources). Forthcoming telescopes including the Vera Rubin Observatory [11][12], James Webb Space Telescope [26], and Nancy Grace Roman Space Telescope [24][25] will push the current limits of observational astronomy and dramatically increase the number of sources to analyze. To measure accurate properties for these sources, we must detect sources by identifying statistically significant local maxima in an image and deblend sources by isolating the potentially overlapping flux distributions of each object. Consider a background-subtracted astronomical image $I \in \mathbb{R}^{h \times w \times b}$ in which $n$ sources are observed, where $h$ is the height, $w$ is the width, and $b$ indicates the number of astronomical passbands. The image $I$ can be decomposed into a sum of individual object contributions as

$$I = \sum_{i=1}^{N} S_i + \epsilon$$  (1)

where $S_i \in \mathbb{R}^{h \times w \times b}$ represents the flux contributed to $I$ by source $i$, and $\epsilon \in \mathcal{N}(0, \sigma)$ is the approximate noise distribution in the image. The process of decomposing an image into the form of Equation 1 represents the core challenge of source deblending. This submission presents a deep learning-based method to perform detection and deblending on astronomical images.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).
### Table 1: Detection and deblending method categorization

<table>
<thead>
<tr>
<th>Name</th>
<th>Detection Capacity</th>
<th>Deblend Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SExtractor[2]</td>
<td>N</td>
<td>Disjoint</td>
</tr>
<tr>
<td>Morpheus[7]</td>
<td>N</td>
<td>Disjoint</td>
</tr>
<tr>
<td>blend2mask2flux[3]</td>
<td>2</td>
<td>Intersecting/Discrete</td>
</tr>
<tr>
<td>Modified SRGAN[22]</td>
<td>0</td>
<td>Intersecting/Continuous</td>
</tr>
<tr>
<td>SCARLET[19]</td>
<td>0</td>
<td>Intersecting/Continuous</td>
</tr>
<tr>
<td>This Work</td>
<td>N</td>
<td>Intersecting/Continuous</td>
</tr>
</tbody>
</table>

### 1.1 Related Work

Source detection and deblending are well-studied problems in astronomy, and many approaches have been developed. Below, we highlight some popular and recent methods for source detection and deblending and point the interested reader to the review by Masias et al. [18].

Detection and deblending methods can be characterized by their detection capacity and deblend type. The detection capacity represents the number of sources a method can detect within a single image. For Equation 1, a detection capacity of \( N \) would indicate that a method could detect all sources appearing in an image. The deblend type indicates how the flux in a single pixel may be split between overlapping (blended) sources. A disjoint deblerder assigns all flux in a pixel to a single source exclusively. An intersecting/discrete deblerder can assign the flux to more than one source with uniform weighting across pixels. Finally, an intersecting/continuous deblerder can assign the flux to more than one source with variable weighting across pixels.


### 2 Partial-Attribution Instance Segmentation

Partial-Attribution Instance Segmentation (PAIS) is a new extension of the instance segmentation paradigm that allows for weighted, overlapping segmentation maps. PAIS differs from other segmentation schemes like cell segmentation [28], interacting surface segmentation [27], and amodal instance segmentation [17]. PAIS aims to isolate objects appearing in an image while preserving their measurable quantities within areas of overlap. For PAIS, we can approximate Equation 1 as

\[
\tilde{I} = \sum_{i=1}^{N} M_i \odot I
\]

where \( \tilde{I} \in \mathbb{R}^{h \times w \times b} \) estimates the background-subtracted flux image \( I \) in Equation 1. \( M_i \in [0, 1]^{h \times w \times b, st.} \sum_i M_{i,jkl} = 1 \) constitutes the pixel-level fractional contribution of source \( i \) to \( I \), and \( \odot \) symbolizes the Hadamard product. Equation 2 is tractable for deep learning models, allowing the model to learn the bounded quantities \( M_i \) rather than the unbounded source images \( S_i \).

The \( N \) number of sources setting the upper limit of the sum in Equation 2 can differ for each image.

To construct a PAIS format that can be represented by a CNN, we have to construct an encoding for the \( M_i \) in Equation 2 inspired by Cheng et al. [5] and Kendall et al. [13]. We propose an encoding for
Table 2: Partial Contribution Representation encoding efficacy

<table>
<thead>
<tr>
<th>Test</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Source Flux [e/s] (MAE)</td>
<td>1.97 ± 15.43</td>
</tr>
<tr>
<td>Two-Sample KS Test p-value</td>
<td>0.93 ± 0.22</td>
</tr>
</tbody>
</table>

the $M_i$ components called Partial Contribution Representation (PCR). The goal of PCR is to encode, for any single pixel $(j, k, l)$, the fractional contribution to its intensity from the closest $n$ sources. Using PCR, a variable number $N$ of sources can be encoded per image. PCR consists of three tensors: the Center-of-mass $C^c \in \{0, 1\}^{h \times w}$, Contribution-vectors $C^v \in \mathbb{R}^{h \times w \times n \times 2}$ and Contribution-maps $C^m \in [0, 1]^{h \times w \times b \times n}$. The center-of-mass encodes the locations of all the sources in an image. For any pixel, we set $C^c_{jk} = 1$ if that location indicates the center of a source and $C^c_{jk} = 0$ otherwise. The contribution-vector $C^v_{jk}$ encodes the Cartesian offset to the closest $n$ sources. The contribution-map $C^m_{jkl}$ connects the fractional contribution of the $n$ sources with the associated contribution-vectors $C^v_{jk}$. The fixed dimensionality of $C^c$, $C^v$, and $C^m$ make PCR tractable for deep learning algorithms.

3 Our Approach

Our approach consists of making a PAIS dataset leveraging PCR and is implemented using a novel neural network architecture. We summarize our dataset, model, and training method below.

3.1 Dataset

To generate the PAIS input samples, we used the Hubble Legacy Fields (HLF) GOODS-South F160W ($1.6 \mu m$) flux images [10], along with the 3D-HST source catalog [20]. The HLF images were split into training and test sets of $256 \times 256$ pixel subregions, with 2,000 training samples and 500 test samples. The input labels, as described in Section 2, consist of the center-of-mass images $C^c$, the contribution-vectors $C^v$, and the contribution-maps $C^m$. The center-of-mass training images are generated in a manner similar to Cheng et al. [5], by placing pixelated 2D Gaussians with standard deviation $\sigma = 8$ (pixels) at the locations of sources in the 3D-HST catalog. The contribution-vectors, an extension to the method by Cheng et al. [5], are generated from the Cartesian offset to the nearest $n = 3$ sources to each pixel. The contribution-maps require the $M_i$ values from Equation 2. To determine $M_i$, we use SCARLET [19] with the F125W, F160W, F606W, and F850LP flux and weight images from the HLF GOODS-South data and the TinyTim point-spread functions [15] to deblend the sources from the 3D-HST catalog. We then use PCR to encode the $M_i$ from SCARLET. The complete dataset generation routine can be found in our project repository (https://github.com/ryanhausen/morpheus-deblend/).

To evaluate the efficacy of PCR to encode $M_i$, we define two metrics. We use the mean difference between the total flux determined by the SCARLET encoded $M_i$ for each input source and that recovered by our encoding. We also use a two-sample Kolmogorov–Smirnov (KS) test to compare the normalized cumulative surface brightness profile within the radius encompassing 90% of the total flux of each source to evaluate the encoding of the spatial flux distribution. Table 2 reports the results and demonstrates that PCR encoding approximately preserves both the total flux and the spatial flux distribution for each source. With this verification, we can train a network to recover the PCR encoding for each input HST F160W image.

3.2 Model

To recover the PCR for training images, we developed a novel neural network architecture inspired by Cheng et al. [5], based on the Fast Attention Network [9] and implemented in TensorFlow [11]. The model features two decoders that share a single encoder. The first decoder, called the spatial decoder, predicts values for $C^c$ and $C^m$. The second decoder, called the attribution decoder, predicts values for $C^v$. The complete model code can be found in the repository for this project (https://github.com/ryanhausen/morpheus-deblend/). An end-to-end example of the model can be seen in Figure 3.2.
3.3 Training

To train the model to recover the PCR of the input images, we use the Adam Optimizer [14] with a learning rate of $5 \times 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.9999$, $\epsilon = 1 \times 10^{-7}$, and a batch size of 100. The model was trained for 1000 epochs using an NVIDIA V100 32GB GPU, taking 31 hours. The loss function for the model is composed of three functions. The spatial decoder outputs for $C^c$ and $C^v$ are penalized according to mean squared error (MSE) and the mean absolute error (MAE), respectively. The attribution decoder output $C^m$ is penalized using cross-entropy loss with an additional entropy regularization term. In practice, we found that the additional entropy regularization helped incentivize the network to learn information about multiple sources in $C^m$. Each loss term is weighted and combined into a single loss function described by

$$L_{\text{total}} = \lambda_{C^c} L_{C^c} + \lambda_{C^v} L_{C^v} + \lambda_{C^m} L_{C^m} + \lambda_S L_S,$$

where $L_{C^c}$ is the MSE loss calculated between the model output and input label with $\lambda_{C^c} = 15$, $L_{C^v}$ is the MAE loss calculated between the model output and input label with $\lambda_{C^v} = 0.06$, $L_{C^m}$ is a cross-entropy loss calculated between the model output and input label with $\lambda_{C^m} = 4$, and $L_S$ is the entropy regularization on the model $C^m$ output with $\lambda_S = 2$. See Table 3 for a summary of the training results, demonstrating a good balance between test and training error. A complete log of training experiments is available at [https://www.comet.ml/ryanhausen/morpheus-deblend/](https://www.comet.ml/ryanhausen/morpheus-deblend/).

4 Discussion and Future Work

In this work, we introduced the Partial Attribution Instance Segmentation (PAIS) scheme for astronomical source deblending. We presented Partial Contribution Representation (PCR) as a method for implementing PAIS within deep learning-based models. We demonstrated the efficacy of PCR for encoding the results of existing astronomical deblenders, and developed a novel neural network architecture to recover the PCR from input flux images. While we demonstrated deblending for single band (F160W) images, PCR can be extended to multiband images. As with many supervised
methods, our model requires labeled training data. To apply this method on other survey datasets may require the use of transfer learning [6,21] or retraining.

Acknowledgments and Disclosure of Funding

5 Acknowledgements

RDH would like to thank Roberto Manduchi for helpful conversations. BER acknowledges support from NASA contract NNG16PJ25C and grant 80NSSC18K0563. The authors acknowledge use of the lux supercomputer at UC Santa Cruz, funded by NSF MRI grant AST 1828315.

6 Broader Impact

This work develops a novel method for separating source signals in astronomical images. Due to the specialized format and problem setting, the authors do not see any broader negative societal impacts as a result of this work.

References


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] The claim in the abstract is that we formed a novel approach to deblending. That claim is supported in Section 1.1 a brief survey of related work and in Section 2 describing the new formulation of the problem and in Section 3.2 describing the novel architecture.
   (b) Did you describe the limitations of your work? [Yes] See Section 4 we note dependence on training data and potential requirement to apply transfer learning/retraining on another astronomical image dataset.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In Sections 3.1 and 3.2, we reference the github repo for the project which contains the code to reproduce the dataset, the model, and train the model.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Section 3.3, we include the training hyperparameters along with a URL to all of our experiment logs.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars for both the model (see Table 3) and the encoding (see Table 2).

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The type of GPUs used is mentioned in Section 3.3, and the particular compute resources are mentioned in the acknowledgements.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] In Section 3.1, we cite the resources used to generate the dataset. In Section 3.2, we cite TensorFlow as the software used to write the model. If the reviewers would like, we can add a software section to the paper to include references to more common software like Python. I didn’t see any guidance on this in the NeurIPS style guide.

(b) Did you mention the license of the assets? [No] We are not reproducing any of the assets.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code to reproduce the dataset and the model is available in the repository for the project which is listed in the paper. The code is available under the MIT license.

(d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]