The Astrophysical Journal Supplement Series, 248:20 (37pp), 2020 May https://doi.org/10.3847/1538-4365/ab8868

Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data

Ryan Hausen[1] and Brant E. Robertson[2,3]
[1] Department of Computer Science and Engineering, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA; rhausen@ucsc.edu
[2] Department of Astronomy and Astrophysics, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA; brant@ucsc.edu
[3] Institute for Advanced Study, 1 Einstein Drive, Princeton, NJ 08540, USA
Received 2019 June 26; revised 2020 April 2; accepted 2020 April 6; published 2020 May 12

Abstract

We present *Morpheus*, a new model for generating pixel-level morphological classifications of astronomical sources. *Morpheus* leverages advances in deep learning to perform source detection, source segmentation, and morphological classification pixel-by-pixel via a semantic segmentation algorithm adopted from the field of computer vision. By utilizing morphological information about the flux of real astronomical sources during object detection, *Morpheus* shows resiliency to false-positive identifications of sources. We evaluate *Morpheus* by performing source detection, source segmentation, morphological classification on the Hubble Space Telescope data in the five CANDELS fields with a focus on the GOODS South field, and demonstrate a high completeness in recovering known GOODS South 3D-HST sources with $H < 26$ AB. We release the code publicly, provide online demonstrations, and present an interactive visualization of the *Morpheus* results in GOODS South.

Unified Astronomy Thesaurus concepts: Galaxy classification systems (582); Galaxies (573); Extragalactic astronomy (506); Convolutional neural networks (1938); Computational methods (1965); GPU computing (1969)

Supporting material: machine-readable table

1. Introduction

Morphology represents the structural end state of the galaxy formation process. Since at least Hubble (1926), astronomers have connected the morphological character of galaxies to the physics governing their formation. Morphology can reflect the initial conditions of galaxy formation, dissipation, cosmic environment and large-scale tidal fields, merger and accretion history, internal dynamics, star formation, the influence of supermassive black holes, and a range of other physics (e.g., Binney 1978; Dressler 1980; Binney & Tremaine 1987; Djorgovski & Davis 1987; Dressler et al. 1987; Bender et al. 1992; Tremaine et al. 2002). The development of morphological measures for galaxies, therefore, comprises an important task in observational astronomy. To help realize the potential of current and future surveys for understanding galaxy formation through morphology, this work presents *Morpheus*, a deep learning-based model for the simultaneous detection and morphological classification of objects through the pixel-level semantic segmentation of large astronomical image data sets.

The established connections between morphology and the physics of galaxy formation run deep, and the way these connections manifest themselves observationally depends on the measures of morphology used. Galaxy size and surface brightness profile shape have served as common proxies for morphology, as quantitatively measured from the light distribution of objects (Vaucouleurs 1959; Sérsic 1968; Peng et al. 2010). Size, radial profile, and isophotal shape or ellipticity vary with stellar mass and luminosity (e.g., Kormendy 1977; Roberts & Haynes 1994; Shen et al. 2003; Sheth et al. 2010; Bruce et al. 2012; van der Wel et al. 2012, 2014; Morishita et al. 2014; Huertas-Company et al. 2015; Allen et al. 2017; Jiang et al. 2018; Miller et al. 2019; Zhang et al. 2019). When controlled for other variables, these measures of galaxy morphology may show variations with cosmic environment (Dressler et al. 1997; Smail et al. 1997; Cooper et al. 2012; Huertas-Company et al. 2016; Kawinwanichakij et al. 2017), redshift (Abraham & van den Bergh 2001; Trujillo et al. 2004, 2006; Conselice et al. 2005; Elmegreen et al. 2005; Lotz et al. 2008; van Dokkum et al. 2010; Patel et al. 2013; Shibuya et al. 2015), color (Franx et al. 2008; Yano et al. 2016), star formation rate or quiescence (Toft et al. 2007; Zirm et al. 2007; Wuyts et al. 2011; Bell et al. 2012; Lee et al. 2013; Whitaker et al. 2015), internal dynamics (Bezanson et al. 2013), the presence of active galactic nuclei (Kocevski et al. 2012; Bruce et al. 2016; Powell et al. 2017), and stellar age (Williams et al. 2017). The presence and size of bulge, disk, and bar components also vary with mass and redshift (Sheth et al. 2008; Simmons et al. 2014; Margalef-Bentabol et al. 2016; Dimauro et al. 2018), and they provide information about the merger rate (e.g., Lofthouse et al. 2017; Weigel et al. 2017). Galaxy morphology encodes a rich spectrum of physical processes and can augment what we learn from other galaxy properties.

While complex galaxy morphologies may be easily summarized with qualitative descriptions (e.g., "disky," "spheroidal," "irregular"), providing quantitative descriptions of this complexity represents a long-standing goal for the field of galaxy formation and has motivated ingenuitive analysis methods including measures of galaxy asymmetry, concentration, flux distribution (e.g., Abraham et al. 1994, 1996; Conselice et al. 2000; Conselice 2003; Lotz et al. 2004), shapelet decompositions (Kelly & McKay 2004, 2005), morphological principal component analyses (Peth et al. 2016), and unsupervised morphological hierarchical classifications (Hocking et al. 2018). These measures provide well-defined characterizations of the surface brightness distribution of galaxies and can be connected to their underlying physical state by, e.g., calibration through numerical simulation (Huertas-Company et al. 2018). The complementarity between these quantitative measures and qualitative morphological descriptions of galaxies means that developing both classes of characterizations further can continue to improve our knowledge of galaxy formation physics.

Characterizing large numbers of galaxies with descriptive classifications simultaneously requires domain knowledge of galaxy morphology (“expertise”), the capability to evaluate quickly each galaxy (“efficiency”), a capacity to work on significant galaxy populations (“scalability”), some analysis of the data to identify galaxy candidates for classification (“preprocessing”), a presentation of galaxy images in a format that enables the characteristic structures to be recognized (“data model”), and an output production of reliable classifications (“accuracy”). Methods for the descriptive classification of galaxy morphology have addressed these challenges in complementary ways.

Perhaps the most important and influential framework for galaxy morphological classification to date has been the Galaxy Zoo project (Lintott et al. 2008; Willett et al. 2013, 2017), which enrolls the public in the analysis of astronomical data including morphological classification. This project has addressed the expertise challenge by training users in the classification of galaxies and statistically accounting for the distribution of users’ accuracies. The efficiency of users varies, but by leveraging the power of the public interest and enthusiasm, and now machine learning (Beck et al. 2018; Walmsley et al. 2020), the project can use scalability to offset variability in the performance of individual users. The preprocessing and delivery of suitable images to the users has required significant investment and programming, but has led to a robust data model for both the astronomical data and the data provided by user input. Science applications of Galaxy Zoo include quantitative morphological descriptions of $\sim$50,000 galaxies (Simmons et al. 2017) in the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011), probes of the connection between star formation rate and morphology in spiral galaxies (Willett et al. 2015), and measuring galaxy merger rates (Weigel et al. 2017).

Other efforts have emphasized different dimensions of the morphological classification task. Kartaltepe et al. (2015) organized the visual classification of $\sim$10,000 galaxies in CANDELS by a team of dozens of professional astronomers. This important effort performed object detection and source extraction on the CANDELS science data, assessed their completeness, and provided detailed segmentation maps of the regions corresponding to classified objects. The use of high-expertise human classifiers leads to high accuracy but poses a challenge for scalability to larger samples. The work of Kartaltepe et al. (2015) also leveraged a significant investment in the preprocessing and presentation of the data to their users with a custom interface with a high-quality data model for the results.

Leveraging human classifiers, be they highly expert teams or well-calibrated legions, to provide descriptive morphologies for forthcoming data sets will prove challenging. These challenges motivate a consideration of other approaches, and we present two salient examples in James Webb Space Telescope (JWST; Gardner et al. 2006) and the Large Synoptic Survey Telescope (LSST; Ivezić et al. 2019; LSST Science Collaboration et al. 2009).

JWST enables both sensitive infrared imaging with *NIRCam* and multi-object spectroscopy with *NIRSpec* free of atmospheric attenuation. The galaxy population discovered by JWST will show a rich range of morphologies, star formation histories, stellar masses, and angular sizes (Williams et al. 2018), which makes identifying *NIRCam*-selected samples for spectroscopic follow-up with *NIRSpec* challenging. The efficiency gain of parallel observations with *NIRCam* and *NIRSpec* will lead to programs where the timescale for constructing *NIRCam*-selected samples will be very short ($\sim$2 months) to enable well-designed parallel survey geometries. For this application, the ability to generate quick morphological classifications for thousands of candidate sources will enhance the spectroscopic target selection in valuable space-based observations.

LSST presents a challenge of scale, with an estimated 30 billion astronomical sources, including billions of galaxies over $\sim$17,000 deg^2 (LSST Science Collaboration et al. 2009). The morphological classification of these galaxies will require the development of significant analysis methods that can both scale to the enormity of the LSST data set and perform well enough to allow imaging data to be reprocessed in pace with the LSST data releases. Indeed, morphological classification methods have been identified as keystone preparatory science tasks in the LSST Galaxies Science Roadmap (Robertson et al. 2017, see also Robertson et al. 2019).

Recently, advances in the field of machine learning called “deep learning” have enjoyed success in morphological classification. Dieleman et al. (2015, hereafter D15) and Dai & Tong (2018) use deep learning to classify the Galaxy Zoo Survey. Huertas-Company et al. (2015) used a deep learning model derived from D15 and the classifications from K15 to classify the CANDELS survey. González et al. (2018) used deep learning to perform galaxy detection and morphological classification, an approach that has also been used to characterize Dark Energy Survey galaxy morphologies (Tarsitano et al. 2018). Deep learning models have been further applied to infer the surface brightness profiles of galaxies (Tuccillo et al. 2018), measure their fluxes (Boucaud et al. 2020), and now to simulate entire surveys (Smith & Geach 2019).

Here, we extend previous efforts by applying a semantic segmentation algorithm to both classify pixels and identify objects in astronomical images using our deep learning framework called *Morpheus*. The software architecture of the *Morpheus* framework is described in Section 2, with the essential convolutional neural network and deep learning components reviewed in Appendix A. The *Morpheus* framework has been engineered by using TensorFlow (Abadi et al. 2016) implementations of these components to perform convolutions and tensorial operations, and it is not a port of existing deep learning frameworks or generated via “transfer learning” (e.g., Pratt 1993) of existing frameworks pre-trained on nonastronomical data such as ImageNet (Deng et al. 2009).

We train *Morpheus* using multiband Flexible Image Transport System (FITS; Wells et al. 1981) images of CANDELS galaxies visually classified by Kartaltepe et al. (2015) and their segmentation maps derived from standard *sextractor* analyses (Bertin & Arnouts 1996). The training procedure is described in Section 3, including the “loss function” used to optimize the *Morpheus* framework. Since *Morpheus* provides local estimates of whether image pixels contain source flux, the *Morpheus* output can be used to perform source segmentation and deblending. We present fiducial segmentation and deblending algorithms for *Morpheus* in Section 4.

We then apply *Morpheus* to the Hubble Legacy Fields (HLF; Illingworth et al. 2016) reduction of the CANDELS and GOODS data in the GOODS South region and to the v1.0 data release (Grogin et al. 2011; Koekemoer et al. 2011) for the other four CANDELS regions, and we generate FITS data files of the same pixel format as the input FITS images, each containing the pixel-by-pixel model classifications of the image data into *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* classes, as described in Section 6. We publicly release these *Morpheus* pixel-level classification data products and detail them in Appendix D. We evaluate the performance of *Morpheus* in Section 7, including tests that use the catalog of 3D-HST photometric sources (Skelton et al. 2014; Momcheva et al. 2016) to measure the completeness of *Morpheus* in recovering sources as a function of source magnitude. We find that *Morpheus* is highly complete (>90%) for sources up to one magnitude fainter than objects used to train the model. Using the *Morpheus* results, we provide estimates of the morphological classification of 3D-HST sources as a public value-added catalog, described in Section 8. In Section 9, we discuss applications of *Morpheus* and semantic segmentation, which extend well beyond morphological classification, and connect the capabilities of *Morpheus* to other research areas in astronomical data analysis. We publicly release the *Morpheus* code, provide online tutorials for using the framework via Jupyter notebooks, and present an interactive website to visualize the *Morpheus* classifications and segmentation maps in the context of the HLF images and 3D-HST catalog. These software and data releases are described in Appendices B–D. A summary of our work is presented with some conclusions in Section 10. Throughout the paper, we have used the AB magnitude system (Oke & Gunn 1983) and assumed a flat ΛCDM universe ($\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$) with a Hubble parameter $H_0 = 70\ \mathrm{km\ s^{-1}\ Mpc^{-1}}$ when necessary.

2. *Morpheus* Deep Learning Framework

Morpheus provides a deep learning framework for analyzing astronomical images at the pixel level. Using a semantic segmentation algorithm, *Morpheus* identifies which pixels in an image are likely to contain source flux and separates them from "background" or sky pixels. *Morpheus*, therefore, allows for the definition of corresponding segmentation regions or "segmentation maps" by finding contiguous regions of source pixels distinct from the sky. Within the same framework, *Morpheus* enables for further classification of the source pixels into additional "classes." In this paper, we have trained *Morpheus* to classify the source pixels into morphological categories (*spheroid*, *disk*, *irregular*, *point source/compact*, and *background*) approximating the visual classifications performed by the CANDELS collaboration in K15. These source pixel classes identified by *Morpheus* could, in principle, be trained to reproduce other properties of the galaxies, such as, e.g., photometric redshift, provided a sufficient training data set is available. In the sections below, we describe the architecture of the *Morpheus* deep learning framework. Readers unfamiliar with the primary computational elements of deep learning architectures may refer to Appendix A where more details are provided.

2.1. Input Data

We engineered the *Morpheus* deep learning framework to accept astronomical image data as direct input for pixel-level analysis. *Morpheus* operates on science-quality FITS images, with sufficient pipeline processing (e.g., flat-fielding, background subtraction, etc.) to enable photometric analysis. *Morpheus* accepts multiband imaging data, with a FITS file for each of the n_b bands used to train the model (see Section 3). The pixel format of the input FITS images (or image region) matches the format of FITS images used to perform training, reflecting the size of the convolutional layers of the neural network determined before training. *Morpheus* allows for arbitrarily large images to be analyzed by subdividing them into regions that the model processes in parallel, as described in Section 2.3 below.

For the example application of morphological classification presented in this paper, we use the $F606W(V)$, $F850LP(z)$, $F125W(J)$, and $F160W(H)$ band images from the Hubble Space Telescope for training, testing, and our final analysis. Our training and testing images were FITS thumbnails and segmentation maps provided by Kartaltepe et al. (2015). Once trained, *Morpheus* can be applied to arbitrarily large images via a parallelization scheme described below in Section 2.3. We have used the CANDELS public release data (Grogin et al. 2011; Koekemoer et al. 2011) in additional performance tests and the HLF v2.0 data (Illingworth et al. 2016) for our *Morpheus* data release.

We note that the approach taken by *Morpheus* differs from deep learning models that use traditional image formats, e.g., three-color Portable Network Graphics (PNG) or Joint Photographic Experts Group (JPEG) images as input. Using PNG or JPEG files as input is convenient because deep learning models trained on existing PNG or JPEG data sets, such as ImageNet (Deng et al. 2009; Russakovsky et al. 2015), can be retrained via transfer learning to classify galaxies. However, the use of these inputs requires additional pre-processing beyond the science pipeline, including arbitrary decisions about how to weight the FITS images to represent the channels of the multicolor PNG or JPEG. With the goal of including *Morpheus* framework analyses as part of astronomical pipelines, we have instead used FITS images directly as input to the neural network.

2.2. Neural Network

Morpheus uses a neural network inspired by the U-Net architecture (Ronneberger et al. 2015, see Appendix A.5) and is implemented using Python 3 (Rossum 1995) and the TensorFlow library (Abadi et al. 2016). We construct *Morpheus* from a series of "blocks" that combine multiple operations used repeatedly by the model. Each block performs a sequence of "block operations." Figure 1 provides an illustration of a *Morpheus* block and its block operations. Block operations are parameterized by the number Q of convolved output images, or "feature maps", they produce, one for each convolutional artificial neuron in the layer. We describe this process in more detail below.

Consider input data $\boldsymbol{X}$, consisting of K layers of images with $N \times M$ pixels. We define a block operation on $\boldsymbol{X}$ as

$$\mathrm{OP}_Q(\boldsymbol{X}) = \mathrm{ReLU}(\mathrm{CONV}_Q(\mathrm{BN}(\boldsymbol{X}))), \tag{1}$$

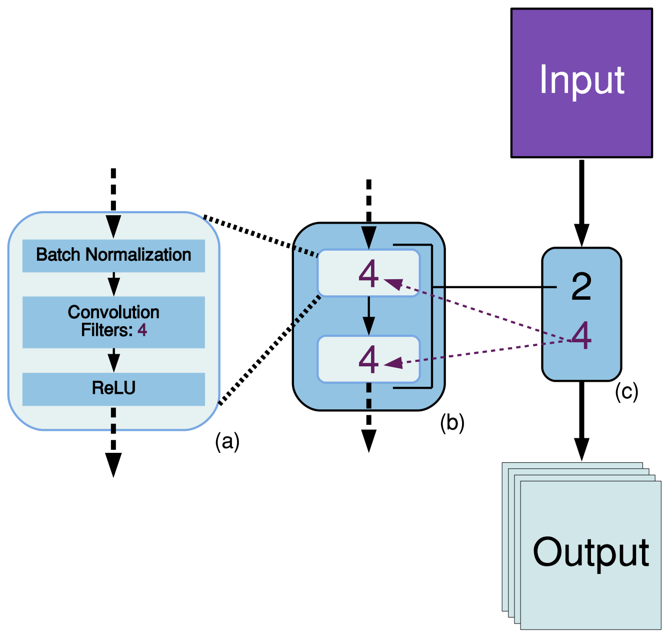

Figure 1. Diagram of a single block in the *Morpheus* neural network architecture (Figure 2). Panel (c) shows a single block from the architecture, parameterized by the number P (black) of block operations and the number Q (purple) of convolutional artificial neurons (CANs; Section A.3) in all of the convolutional layers within the block. Panel (b) shows an example zoom-in where there are $P = 2$ groups of $Q = 4$ block operations. Panel (a) shows a zoom-in on a block operation, which consists of batch normalization, $Q = 4$ CANs, and a rectified linear unit (ReLU). In the notation of Equation (1), this block operation would be written as $\mathrm{OP}_4(\boldsymbol{X})$.

where ReLU is the Rectified Linear Unit activation function (ReLU; Hahnloser et al. 2000; Lecun et al. 2015, see also Appendix A.1), CONV_Q is a convolutional layer (see Appendix A.3) with a number Q of convolutional artificial neurons (see Appendix A.3), and BN is the batch normalization procedure (Ioffe & Szegedy 2015, and Appendix A.4.4). Note that the values of Q appearing in OP_Q and CONV_Q are equal. For example, OP_4 would indicate that the convolutional layer within the OP_4 function has four convolutional artificial neurons. Unless stated otherwise, all inputs into a convolutional layer are zero-padded to preserve the width and height of the input, and all convolutional artificial neurons have kernel dimensions 3×3. Given Equation (1), for an input $\boldsymbol{X}$ with dimensions $N \times M \times K$, the output of the function $\mathrm{OP}_4(\boldsymbol{X})$ would have dimensions $N \times M \times 4$.

Equation (1) allows for a recursive definition of a function describing a series of block operations, where the input data to one block operation consist of the output from a previous block operation. This recursion can be written as

$$\mathrm{OP}_Q^P(\boldsymbol{X}) = \begin{cases} \boldsymbol{X}, & \text{if } P = 0 \\ \mathrm{ReLU}(\mathrm{CONV}_Q(\mathrm{BN}(\mathrm{OP}_Q^{P-1}(\boldsymbol{X})))) & \text{if } P > 0. \end{cases} \tag{2}$$

Equation (2) introduces a new parameter P, shown with a superscript in OP_Q^P. The parameter P establishes the conditions of a base case for the recursion. Note that in Equation (2), the input $\boldsymbol{X}$ is processed directly when $P = 1$, and when $P > 1$, the input to the OP_Q^P function is the output from OP_Q^{P-1}. It can be seen from the formulation of Equations (1) and (2) that $\mathrm{OP}_Q(\boldsymbol{X}) = \mathrm{OP}_Q^1(\boldsymbol{X})$.

Since a block performs a number P block operations, we can define a block mathematically as

$$\mathrm{BLOCK}(Q, P, \boldsymbol{X}) = \mathrm{OP}_Q^P(\boldsymbol{X}). \tag{3}$$

An example block and its block operations can be seen diagrammatically in Figure 1. With these definitions, we can present the neural network architecture used in *Morpheus*.

Like the U-Net architecture, the *Morpheus* architecture consists of a contraction phase and an expansion phase. The contraction phase consists of three blocks with parameters ($P = 4$, $Q = 8$), ($P = 4$, $Q = 16$), and ($P = 4$, $Q = 32$). Each block is followed by a max-pooling operation with size $= (2 \times 2)$ (see Appendix A.4.1), halving the width and height of its input. After the contraction phase, there is a single intermediary block preceding the expansion phase with the parameters ($P = 1$, $Q = 16$). The expansion phase consists of three blocks with the parameters ($P = 2$, $Q = 8$), ($P = 2$, $Q = 16$), and ($P = 2$, $Q = 32$). Each block is preceded by a bicubic interpolation operation that doubles the width and the height of its input. Importantly, the output from each block in the contraction phase is concatenated (see Appendix A.4.3) with the output from the bicubic interpolation operation in the expansion phase whose output matches its width and height (see Figure 2). The output from the final block in the expansion phase is passed through a single convolutional layer with five convolutional artificial neurons. A softmax operation (see Equation (4)) is performed on the values in each pixel, ensuring that the values sum to unity. The final output is a matrix with the same width and height as the input into the network, but where the last dimension, five, now represents a classification distribution describing the confidence that the corresponding pixel from the input belongs to one of the five specified morphological classes.

The blocks in *Morpheus* are organized into the U-Net structure, shown in Figure 2. The model proceeds clockwise, starting from "Input" on the upper left through to "Output" on the lower left. The very first step involves the insertion of the input FITS images into the model. Each FITS image is normalized to have a mean of 0 and unit variance before processing by *Morpheus*. We will refer to the number of input bands as n_b, and in the application presented here, we take $n_b = 4$ (i.e., *VzJH*). The input images each have pixel dimensions $N \times M$, and we can, therefore, consider the astronomical input data to have dimensions $N \times M \times n_b$. Only the first block operation takes the FITS images as input, and every subsequent block operation in the model takes the output from previous blocks as input.

The first convolution in the first block operation convolves the normalized $N \times M \times n_b$ astronomical data with three-dimensional kernels of size $n_k^2 \times n_b$, and each element of the kernel is a variable parameter of the model to be optimized. The convolutions operate only in the two pixel dimensions, such that n_b convolutions are performed, one for each $N \times M$ pixel image, using a different $n_k \times n_k$ kernel for each convolution. The n_b convolved images are then summed pixel by pixel to create an output feature map of size $N \times M$. The convolutional layer repeats this process Q times with different kernels, generating Q output feature maps and an output data set of size $N \times M \times Q$. For the first block in *Morpheus*, we use $Q = 8$ (see Figure 2). After the first convolution on the

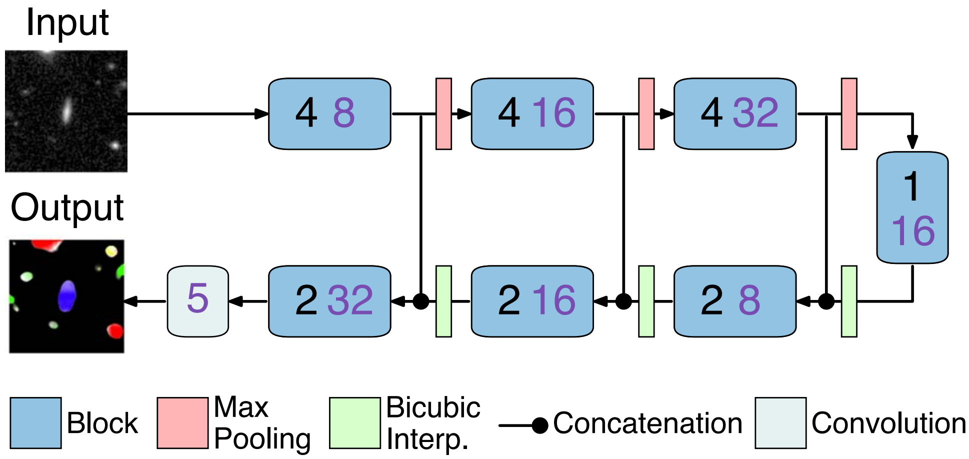

Figure 2. Neural network architecture of the *Morpheus* deep learning framework, following a U-Net (Ronneberger et al. 2015) configuration. The input to the model *Morpheus* consists of astronomical FITS images in n_b bands (upper left). These images are processed through a series of computational blocks (sky blue rectangles), each of which applies P (black numbers) block operations consisting of a batch normalization and multiple convolutional layers producing Q (purple numbers) feature maps. The blocks are described in more detail in Figure 1. During the contraction phase of the model, max-pooling layers (salmon rectangles) are applied to the data to reduce the pixel size of the images by taking local maxima of 2×2 regions. The contraction phase is followed by an expansion phase where the output feature maps from each block are expanded by a 2×2 factor via bicubic interpolation (green rectangles) and concatenated with the output from the corresponding block in the contraction phase. The output from the last block is processed through a set of convolutional layers (light blue box with $Q = 5$) that result in a feature map for each classification in the model. These "classification images" are normalized to sum to unity pixel by pixel. In this paper, the classification images are *spheroid*, *disk*, *irregular*, *point source/compact*, and *background*.

astronomical data, every subsequent convolution in the first block has both an input and output data of size $N \times M \times Q$.

Each block performs a number of block operations P, resulting in output data with dimensions of $N \times M \times Q$ emerging from the block. The number of feature maps Q changes with each block. For a block producing Q filters, if the data entering the block has size $N \times M \times Q'$ with $Q' \neq Q$, then the first convolutional layer in the first block operation will have Q kernels of size $n_k^2 \times Q'$. All subsequent convolutional layers in the block will then ingest and produce data of size $N \times M \times Q$ by using kernels of size $n_k^2 \times Q$.

We can apply further operations on the data in between the blocks, and the character of these operations can affect the dimensions of the data. The first half of the model is a contraction phase, where each block is followed by a max-pooling operation (Cireşan et al. 2012, and Appendix A.4.1). The max-pooling operation is applied to each feature map output by the block, taking the local maximum over small areas within each feature map (in the version of *Morpheus* presented here, a 2×2 pixel region) and reducing the size of the data input to the next block by the same factor. For this paper, the contraction phase in the *Morpheus* framework uses three pairs of blocks plus max-pooling layers.

After the contraction phase, the model uses a series of blocks, bicubic interpolation layers, and data concatenations in an expansion phase to grow the data back to the original format. Following each block in the expansion phase, a bicubic interpolation layer expands the feature maps by the same areal factor as the max-pooling layers applied in the contraction phase (2×2 in the version of *Morpheus* presented here). The output feature maps from the interpolation layers are concatenated with the output feature maps from the contraction phase blocks where the data have the same format. Finally, the output from the last block in the expansion phase is input into a convolutional layer that produces the final output images that we call "*Morpheus* classification images," one image for each class. The pixel values in these images contain the model estimates for their classification, normalized such that the element-wise sum of the classification images equals unity. For this paper, where we are performing galaxy morphological classification, there are five classification images (*spheroid*, *disk*, *irregular*, *point source/compact*, and *background*).

As the data progresses through the model, the number of feature maps and their shapes change owing to the max-pooling and interpolation layers. For reference, in Table 1, we list the dimensions of the data at each stage in the model, assuming input images in n_b bands, each with $N \times M$ pixels, and a total of n_c classification images produced by the model.

2.3. Parallelization for Large Images

While the *Morpheus* neural network performs semantic segmentation on pixels in FITS images with a size determined by the training images, the model can process and classify pixels in arbitrarily large images. To process large images, *Morpheus* uses a sliding window strategy by breaking the input FITS files into thumbnails of size $N \times M$ (the size of the training images) and classifying them individually. *Morpheus* proceeds through the large-format image, first column by column, and then row by row, shifting the active $N \times M$ window by a unit pixel stride and then recomputing the classification for each pixel.

As the classification process continues with unit pixel shifts, each pixel is deliberately classified many times. We noticed heuristically that the output *Morpheus* classification of pixels depended on their location within the image and that the pixel classifications were more accurate relative to our training data when they resided in the inner $n_p = (N - B) \times (M - B)$ region of the classification area, where the less accuracte region consisted of a border about $B \sim 5$ pixels wide on each side. Outside of the very outer B pixels in the large-format image, *Morpheus* classifies each pixel n_p times. For the large FITS data images used in this paper, this repetition corresponds to $n_p = 900$ separate classifications per pixel per output class, where each classification occurs when the pixel lies at a different location within the active window. This substantial

Table 1
Computational Steps in the *Morpheus* Deep Learning Framework

Layer	Input	Output Dimensions
Input Images	n_b Bands, $N \times M$ Pixels	$[N, M, n_b]$
Block 1a	Input Images	$[N, M, 8]$
Block 1b	Block 1a	$[N, M, 8]$
Block 1c	Block 1b	$[N, M, 8]$
Block 1d	Block 1c	$[N, M, 8]$
Max Pooling 1	Block 1d	$[N/2, M/2, 8]$
Block 2a	Max Pooling 1	$[N/2, M/2, 16]$
Block 2b	Block 2a	$[N/2, M/2, 16]$
Block 2c	Block 2b	$[N/2, M/2, 16]$
Block 2d	Block 2c	$[N/2, M/2, 16]$
Max Pooling 2	Block 2d	$[N/4, M/4, 16]$
Block 3a	Max Pooling 2	$[N/4, M/4, 32]$
Block 3b	Block 3a	$[N/4, M/4, 32]$
Block 3c	Block 3b	$[N/4, M/4, 32]$
Block 3d	Block 3c	$[N/4, M/4, 32]$
Max Pooling 3	Block 3d	$[N/8, M/8, 32]$
Block 4a	Max Pooling 3	$[N/8, M/8, 16]$
Interpolation 1	Block 4a	$[N/4, M/4, 16]$
Block 5a	Interp. 1 + Block 3d	$[N/4, M/4, 8]$
Block 5b	Block 5a	$[N/4, M/4, 8]$
Interpolation 2	Block 5b	$[N/2, M/2, 8]$
Block 6a	Interp. 2 + Block 2d	$[N/2, M/2, 16]$
Block 6b	Block 6a	$[N/2, M/2, 16]$
Interpolation 3	Block 6b	$[N, M, 16]$
Block 7a	Interp. 3 + Block 1d	$[N, M, 32]$
Block 7b	Block 7a	$[N, M, 32]$
Convolution	Block 7b	$[N, M, n_c]$

Note. For each layer (left column), we list its input (center column) and the output shape of its data (right column). The model takes as its starting input a set of images in n_b bands, each with $N \times M$ pixels. The final output of the model is a set of n_c classification images, each with $N \times M$ pixels. The *Morpheus* block structures are illustrated in Figure 1. The "+" symbol denotes a concatenation between two layer outputs, as shown in Figure 2.

additional information can be leveraged to improve the model, but storing the full "distribution" of classifications produced by this method would increase our data volume by roughly three orders of magnitude.

While *Morpheus* would enable full use of these distributions, for practical considerations, we instead record some statistical information as the computation proceeds and do not store the entire set of n_p samples. To avoid storing the full distribution, we track running estimates of the mean and variance of the distribution.[4] Once the mean for each class for each pixel is computed, we normalize the means across classes to sum to unity. We further record a statistic we call *rank voting*, which is a tally of the number of times each output class was computed by the model to be the top class for each pixel. The sum of rank votes across classes for a single pixel equals the number of times *Morpheus* processed the pixels (i.e., n_p for most pixels). After the computation, the rank votes are normalized to sum to unity across the classes for each pixel.

The strips of classified regions produce 15 output images, containing the mean and variance estimators for the classification distribution and normalized rank votes for each class. This striped processing of the image can be performed in parallel across multiple *Morpheus* instances and then stitched back together. The weak scaling of this processing is, in principle, trivial and is limited only by the number of available GPUs and the total memory of the computer used to perform the calculation.

3. Model Training

The training of deep learning frameworks involves important decisions about the training data, the metrics used to optimize the network, the numerical parameters of the model, and the length of training. We provide some rationale for these choices below.

3.1. Training Data

To train a model to perform semantic segmentation, we require a data set that provides both information on the segmentation of regions of interest and classifications associated with those regions. For galaxy morphological classification, we use 7629 galaxies sampled from the K15 data set. Their two-epoch CANDELS data provide an excellent combination of multiband FITS thumbnails, segmentation maps in FITS format, and visually classified morphologies in tabulated form. The K15 classifications consisted of votes by expert astronomers, between 3 and 60 per object, who inspected images of galaxies and then selected from several morphological categories to assign to the object. The number of votes for each category for each object are provided, allowing *Morpheus* to use the distribution of votes across classifications for each object when training. We downloaded and used the publicly available K15 thumbnail FITS files for the *F*606*W*, *F*850*LP*, *F*125*W*, and *F*160*W* bands as input into the model for training and testing. In training Morpheus to reproduce the K15 classifications, multiband data approximates the information provided to the astronomers who performed the K15 classifications. Morpheus is trained using the same *V*-, *z*-, *J*-, and *H*-band image thumbnails used in the K15 classification process. Other bands or different numbers of bands could be used for training as necessary, and *Morpheus* allows for reconfiguration and retraining depending on the available training images. Of the K15 data set, we used 80% of the objects to form our training sample and 20% to form our test sample. Various statistical properties of the test and training samples are described throughout the rest of the paper.

The primary K15 classifications *spheroid, disk, irregular*, and *point source/compact* were used in the example *Morpheus* application presented here. We added one additional class, *background*, to represent sky pixels without a significant source flux. We classify pixels as belonging to the background category if those pixels fell outside the K15 segmentation maps. Pixels inside the segmentation maps were assigned the distribution of classifications provided by the K15 experts.

The K15 classification scheme also included an *unknown* class for objects. Since *Morpheus* works at the pixel level and could provide individual pixel classifications that were locally accurate within a source but that collectively could sum to an object whose morphology expert astronomers might classify as *unknown*, we were posed with the challenge of how to treat the K15 *unknown* class. Given our addition of the *background* class constructed from large image regions dominated by sky, one might expect overlap in the features of regions that are mostly noise and amorphous regions classified as *unknown*. Since one might also expect overlap between *unknown* and

[4] See, e.g., http://people.ds.cam.ac.uk/fanf2/hermes/doc/antiforgery/stats.pdf for an example of running mean and variance estimators.

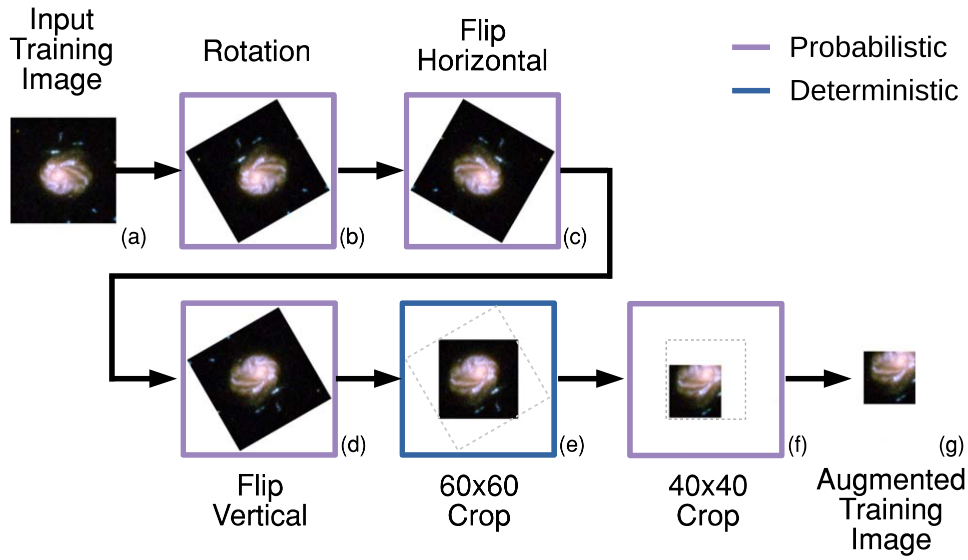

Figure 3. Data augmentation pipeline used during neural network training. Each training image is processed by the data augmentation pipeline before being presented to the neural network during training. The pipeline can be described in seven stages (annotated "(a)–(g)" above). First, an image from the training set is selected (panel (a)). A number of augmentation operations are then applied to the image. The image is rotated by a random angle $\theta \in [0, 2\pi]$ (panel (b)), flipped horizontally with 50% probability (panel (c)), and flipped vertically with a 50% probability (panel (d)). The centermost 60×60 subset of the resulting image is cropped (panel (e)), and then a random 40×40 subset is selected from the cropped image (panel (f)). The output 40×40 rotated, flipped, and cropped image is then used for training. This procedure increases the available images for training by a factor of ~574,400. Using this process helps reduce overfitting, particularly in the cases of data sets with limited training sample sizes.

irregular classifications, we wanted to preserve some distinction in the object classes. We, therefore, removed the *unknown* class by removing any sources that had *unknown* as their primary classification from the training sample (213 sources). For any sources where the nondominant K15 classifications included *unknown*, we redistributed the *unknown* votes proportionally to the other classes.

3.2. Data Augmentation

To increase the effective size of the training data set, *Morpheus* uses a data augmentation method. Augmentation supplements the input training data set by performing transformations on the training images to alter them with the intent of adding similar but not identical images with known classifications. Augmentation has been used successfully in the context of galaxy morphological classification (e.g., Dieleman et al. 2015), and *Morpheus* adopts a comparable approach to previous implementations.

During training, *Morpheus* produces a series of 40×40 pixel augmented versions of the training images. The augmentation approach is illustrated in Figure 3. For each band in the original training image, the image is collectively rotated by a random angle $\phi \in [0, 2\pi]$, flipped horizontally with a random 50% probability, and then flipped vertically with a random 50% probability. A crop of the inner 60×60 pixels of the resulting image is produced, and then a random 40×40 pixel subset of the image is selected and passed to the model for training. This method allows us to increase the effective number of images available for training by a factor of ~574,400 and helps ameliorate over-training on the original training image set.

3.3. Loss Function

A standard method for training deep learning frameworks is to define a loss function that provides a statistic based on the output classifications to optimize via stochastic gradient descent with gradients computed using back-propagation (Rumelhart et al. 1986). Here, we describe how the *Morpheus* loss function is constructed.

The first task is to assign a distribution of input classifications on a per-pixel basis, choosing between the n_c classes available to the *Morpheus* model. For this work, we choose $n_c = 5$ (*background*, *disk*, *spheroid*, *irregular*, and *point source/compact*), but *Morpheus* can adopt an arbitrary number of classes. We use the index k to indicate a given class, with $k \in [1,\ n_c]$. Consider an $N \times M$ image of an astronomical object that has been visually classified by a collection of experts, and a segmentation map defining the extent of the object in the image. Outside the segmentation map of the object, the pixels are assumed to belong to the sky and are assigned the *background* class. Inside the segmentation map, pixels are assigned the distribution of *disk*, *spheroid*, *irregular*, and *point source/compact* classifications determined by the experts for the entire object. For each pixel ij, with $i \in [1, N]$ rows and $j \in [1, M]$ columns, we then have the vector q_{ij} whose elements q_{ijk} contain the input distribution of classifications. Here, the index k runs over the number of classes n_c and $\sum_k q_{ijk} = 1$ for each pixel with indices ij. The goal of the model is to reproduce this normalized distribution q_{ij} of discrete classes for each pixel of the training images. We wish to define a *total loss function* $L_{\rm tot}$ that provides a single per-image statistic for the model to optimize when attempting to reproduce q_{ij}. *Morpheus* combines a weighted *cross entropy* loss function with a Dice loss (Milletari et al. 2016; Novikov et al. 2018) for its optimization statistic, which we describe below.

At the end of the *Morpheus* data flow, as outlined in Figure 2, the raw output of the model consists of $N \times M$ vectors x_{ij} with n_c elements per-pixel estimates that represent unnormalized approximations to the input per-pixel distributions q_{ij}. The model outputs x_{ij} for each pixel are then normalized to form a probability distribution p_{ij} using the *softmax* function

$$p_{ij} = \frac{\exp(x_{ij})}{\sum_{k=1}^{n_c} \exp(x_{ijk})}, \quad \text{for} \quad k \in [1, n_c]. \tag{4}$$

The distribution p_{ij} then represents the pixel-by-pixel classifications computed by *Morpheus* for each of the $k \in [1, n_c]$ classes. For a pixel with indices ij, we can define the per-pixel cross entropy loss function as

$$L_{ij}(p_{ij}, q_{ij}) = -\sum_{k=1}^{n_c} p_{ijk} \log(q_{ijk}) \tag{5}$$

where p_{ij} and q_{ij} are again the two per-pixel probability distributions, with q_{ij} representing the true distribution of the input classifications for the pixel ij and p_{ij} representing the model output.

Equation (5) provides the per-pixel contribution to the entropy loss function. However, for many images, the majority of pixels lie outside the segmentation maps of sources identified in the training data and are therefore labeled as *background*. To overcome this imbalance and disincentivize the model from erroneously learning to classify pixels containing source flux as *background*, we apply a weighting to the per-pixel loss. We define an index $k_{ij}^{\max,q}$ that indicates which class is the maximum of the input classification distribution for each pixel, written as

$$k_{ij}^{\max,q} = \text{argmax } q_{ij} \tag{6}$$

with $1 \leqslant k_{ij}^{\max,q} \leqslant n_c$. For each class k, we then define a weight w_k that is inversely proportional to the number of pixels with $k_{ij}^{\max,q} = k$. We can write

$$w_k = \left[\sum_{i=1}^{N}\sum_{j=1}^{M} \max(q_{ij})\delta_{k,k_{ij}^{\max,q}}\right]^{-1}. \tag{7}$$

Here, $\delta_{i,j}$ is the Kronecker delta function. The vector w has size n_c and each of its elements w_k contain the inverse of the sum of $\max(q_{ij})$ for pixels with $k_{ij}^{\max,q} = k$. In a given image, we ignore any classes that do not appear in the input classification distribution (i.e., any class k for which $\sum_i \sum_j q_{ijk} = 0$).

Using w, we define a weighted cross entropy loss for each pixel as

$$L_{ij}^{w} = w_{k_{ij}^{\max}} L_{ij}(p_{ij}, q_{ij}). \tag{8}$$

A mean weighted loss function is then computed by averaging Equation (8) over all pixels as

$$\bar{L}^{w} = \frac{1}{N \times M} \sum_{i=1}^{N}\sum_{j=1}^{M} L_{ij}^{w}. \tag{9}$$

This mean weighted loss function serves as a summary statistic of the cross entropy between the output of *Morpheus* and the input classification distribution.

When segmenting images primarily comprised of *background* pixels, the classification distributions of the output pixels should be highly unbalanced, with the majority having *background* ≈ 1. In this case, the mean loss function statistic defined by Equation (9) will be strongly influenced by a single class. A common approach to handle unbalanced segmentations is to employ a Dice loss function to supplement the entropy loss function (e.g., Milletari et al. 2016; Sudre et al. 2017). The Dice loss function used by *Morpheus* is written as

$$L^{D}(\boldsymbol{b}, \boldsymbol{m}) = 1 - 2\frac{\sum_i \sum_j (S(\boldsymbol{b}) \circ \boldsymbol{m})_{ij}}{\sum_i \sum_j (S(\boldsymbol{b}) + \boldsymbol{m})_{ij}}. \tag{10}$$

Here, $S(\boldsymbol{b}) = (1 + \exp(-\boldsymbol{b}))^{-1}$ is the sigmoid function (see Equation (A3)) applied pixel-wise to the *background* classification image output by the model. The image $\boldsymbol{m}$ is the input mask with values $m = 1$ denoting *background* pixels and $m = 0$ indicating source pixels, defined, e.g., by a segmentation map generated using *sextractor*. The $\circ$ symbol indicates a Hadamard product of the matrices $S(\boldsymbol{b})$ and $\boldsymbol{m}$. Note that the output *background* matrix $\boldsymbol{b}$ has not yet been normalized using a softmax function, and so, $b_{ij} \in [-\infty, \infty]$ and $S(b_{ij}) \in [0, 1]$. The Dice loss then ranges from $L^D = 0$ if $S(\boldsymbol{b}) \approx \boldsymbol{m}$ and $L^D \sim 1$ when $S(\boldsymbol{b})$ and $\boldsymbol{m}$ differ substantially. The addition of this loss function helps to maximize the spatial coincidence of the output *background* pixels assigned $b_{ij} \approx 1$ with the nonzero elements of the input segmentation mask $\boldsymbol{m}$.

To define the total loss function optimized during the training of *Morpheus*, the cross entropy and Dice losses are combined as a sum weighted by two parameters λ_w and λ_D. The total loss function is written as

$$L_{\text{tot}} = \lambda_w L^w + \lambda_D L^D. \tag{11}$$

For the implementation of *Morpheus* used in this paper, the entropy and Dice loss functions are weighted equally by setting $\lambda_w = 1$ and $\lambda_D = 1$.

3.4. Optimization Method

To optimize the model parameters, the *Adam* stochastic gradient descent method (Kingma & Ba 2014) was used. The *Adam* algorithm uses the first and second moments of first-order gradients computed via back-propagation to find the minimum of a stochastic function (in this case, our loss function, see Section 3.3, which depends on the many parameters of the neural network). The *Adam* optimizer, in turn, depends on hyper-parameters that determine how the algorithm iteratively finds a minimum. Since the loss function is stochastic, the gradients change with each iteration, and *Adam* uses an exponential moving average of the gradients ($\hat{m}$) and squared gradients ($\hat{v}$) when searching for a minimum. Two dimensionless hyper-parameters (β_1 and β_2) set the decay rates of these exponential averages (see Algorithm 1 of Kingma & Ba 2014). As the parameters θ of the function being optimized are iterated between steps $t - 1$ and t, they are updated according to

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon). \tag{12}$$

Here, ϵ is a small, dimensionless safety hyper-parameter that prevents division by zero, and α is a small, dimensionless hyper-parameter that determines the magnitude of the iteration step. Table 2 lists the numerical values of the *Adam* optimizer hyper-parameters used by *Morpheus*. We use the default suggested values for β_1, β_2, and ϵ. After some experimentation, we adopted a more conservative step size for α than that used by Kingma & Ba (2014).

Table 2
Adam Optimizer (Kingma & Ba 2014) Hyper-parameter Values Used During the Training of the Neural Network Used in *Morpheus*

Adam Optimizer Hyper-parameters	
Hyper-parameter	Value
β_1	0.9
β_2	0.999
ϵ	10^{-8}
α	9.929×10^{-5}

Note. See the text for definitions of the hyper-parameters.

3.5. Model Evaluation

As training proceeds, the performance of the model can be quantified using various metrics and monitored to determine when training has effectively completed. The actual performance of *Morpheus* will vary depending on the classification scheme used, and here, we report the performance of the model relative to the CANDELS images morphologically classified in K15. Performance metrics reported in this section refer to pixel-level quantities, and we discuss object-level comparisons of morphological classifications relative to K15 in Section 5.

While the model training proceeds by optimizing the loss function defined in Section 3.3, we want to quantify the accuracy of the model in recovering the per-pixel classification and the overlap of contiguous regions with the same classification. First, we will need to define the index $k_{ij}^{\max}$ with maximum probability to reflect either the input classification q_{ij} or the output classification p_{ij}. We define an equivalent of Equation (6) for p_{ij} as

$$k_{ij}^{\max,p} = \operatorname{argmax} p_{ij}. \tag{13}$$

We can then define a percentage accuracy

$$\mathcal{A} = \frac{100}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M} \delta_{k_{ij}^{\max,p},k_{ij}^{\max,q}}. \tag{14}$$

The accuracy $\mathcal{A}$ then provides the percentage of pixels for which the maximum probability classes of the input and output distributions match.

In addition to accuracy, the intersection-over-union $\mathcal{I}_{\mathcal{U}}$ of pixels with *background* probabilities above some threshold is computed between the input q_{ij} and output p_{ij} distributions. If we define the index b to represent the *background* class, we can express the input *background* probabilities as $q_b = q_{ijb}$ for $i \in [1, N]$ and $j \in [1, M]$ and the equivalent for the output *background* probabilities p_b. We can refer to q_b and p_b as the input and output *background* images, and the regions of these images with values above some threshold B are expressed as $q_b(>B)$ and $p_b(>B)$, respectively. Note that the input q_b only contains values of zero or one, whereas the output p_b has continuous values between zero and one. We can then define the $\mathcal{I}_{\mathcal{U}}$ metric for threshold B as

$$\mathcal{I}_{\mathcal{U}}(B) = \frac{p_b(>B) \cap q_b(>B)}{p_b(>B) \cup q_b(>B)}. \tag{15}$$

Intuitively, this $\mathcal{I}_{\mathcal{U}}$ metric describes how well the pixels assigned by *Morpheus* as belonging to a source match up with the input source segmentation maps. A value of $\mathcal{I}_{\mathcal{U}} = 1$ indicates a perfect match between source pixels identified by *Morpheus* and the input segmentation maps, while a value of $\mathcal{I}_{\mathcal{U}} = 0$ would indicate no pixels in common between the two sets.

Table 3
Morpheus Training and Test Results for Accuracy $\mathcal{A}$, and Intersection-over-union $\mathcal{I}_{\mathcal{U}}$ as a Function of *Background* Threshold B

Morpheus Training and Test Results		
Metric	Training	Test
Accuracy $\mathcal{A}$		
Background	91.5%	91.4%
Disk	74.9%	75.1%
Irregular	80.6%	68.6%
Point source/compact	91.0%	83.8%
Spheroid	72.3%	71.4%
All Classes	86.8%	85.7%
Intersection-over-union $\mathcal{I}_{\mathcal{U}}$		
$B > 0.5$	0.899	0.888
$B > 0.6$	0.900	0.891
$B > 0.7$	0.902	0.893
$B > 0.8$	0.902	0.895
$B > 0.9$	0.900	0.896

As training proceeds, the accuracy $\mathcal{A}$ and intersection-over-union $\mathcal{I}_{\mathcal{U}}$ are monitored until they plateau with small variations. For the K15 training data, the model plateaued after about 400 epochs. The training then continues for another 100 epochs to find a local maximum in $\mathcal{A}$ and $\mathcal{I}_{\mathcal{U}}$, and the model parameters at this local maximum are adopted for testing. Table 3 summarizes the per-pixel performance of *Morpheus* in terms of $\mathcal{A}$ for each class separately, $\mathcal{A}$ for all classes, and $\mathcal{I}_{\mathcal{U}}(B)$ for $B = [0.5, 0.6, 0.7, 0.8, 0.9]$. We also report the performance of the training and testing samples separately. The pixel-level classifications are 70%–90% accurate depending on the class, and the intersection-over-union is $\mathcal{I}_{\mathcal{U}} \sim 0.9$ for all thresholds $B \geqslant 0.5$. The model shows some evidence for overfitting as accuracy declines slightly from the training to test sets for most classes.

4. Segmentation and Deblending

To evaluate the completeness of *Morpheus* in object detection and to compute an object-level classification, segmentation maps must be constructed and then deblended from the *Morpheus* pixel-level output. *Morpheus* uses the *background* class from the output of the neural network described in Section 2.2 to create a segmentation map. The segmentation algorithm uses a watershed transform to separate *background* pixels from source pixels and then assigns contiguous source pixels a unique label. The deblending algorithm uses the flux from the input science images and the output of the segmentation algorithm to deblend under-segmented regions containing multiple sources. We summarize these procedures as Algorithms 1 and 2. Figure 4 illustrates the process for generating and deblending segmentation maps.

4.1. Segmentation

The segmentation algorithm operates on the output *background* classification image and identifies contiguous regions of low background as sources. The algorithm begins with the *background* image $b \equiv p_b$ defined in Section 3.5 and an

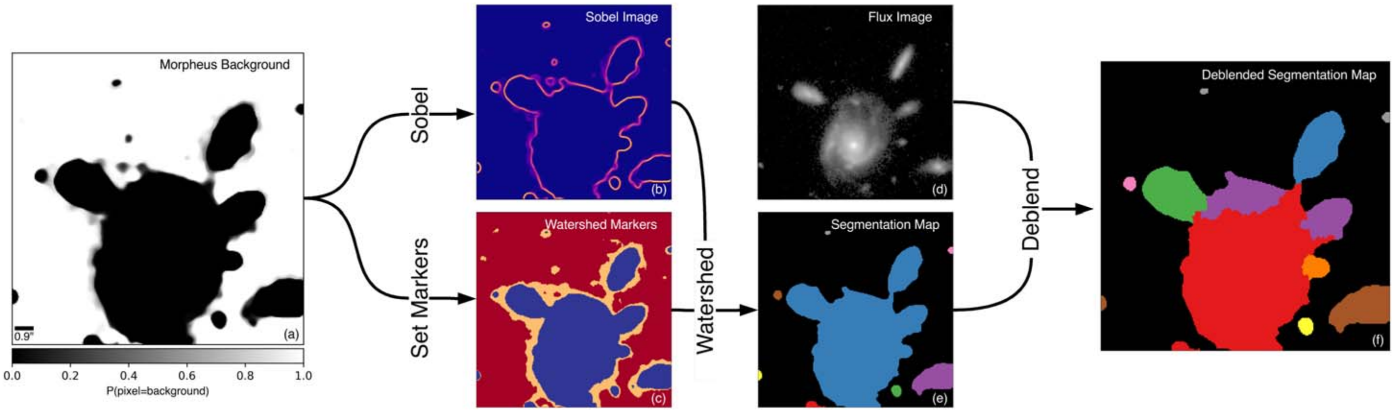

Figure 4. Segmentation and deblending process used by *Morpheus*, illustrating Algorithms 1 and 2. The *background* image (panel (a)) output from the *Morpheus* neural network is used as input to a Sobel-filtered image (panel (b)) and a discretized map marking regions of high and low *background* (panel (c)). These two images are input to a watershed algorithm to identify and label distinct, connected regions of low *background* that serve as the highest-level *Morpheus* segmentation map (panel (e)). This segmentation map represents the output of Algorithm 1. A flux image and a list of object locations (panel (d)) are combined with the high-level segmentation map to deblend multicomponent objects using an additional watershed algorithm by using the source locations in the flux image as generating points. The end result is a deblended segmentation map (panel (f)), corresponding to the output of Algorithm 2.

initially empty mask $m = 0$ of the same size. For every pixel in the image, if $b_{ij} = 1$, we set $m_{ij} = 1$, and if $b_{ij} = 0$, we set $m_{ij} = 2$. The *background* mask m then indicates extreme regions of b. The Sobel & Feldman (1968) algorithm is applied to the *background* image b to produce a Sobel edge image s. *Morpheus* then applies the watershed algorithm of Couprie & Bertrand (1997), using the Sobel image s as the "input image" and the *background* mask m as the "marker set." We refer the reader to Couprie & Bertrand (1997) for more details on the watershed algorithm; but in summary, the watershed algorithm collects regions together that have the same marker set value within basins in the input image. The Sobel image s provides these basins by identifying edges in the background, and the *background* mask m provides the marker locations for generating the individual sheds. The output of the watershed algorithm is then an image sm containing distinct regions generated from areas of low *background* that are bounded by edges where the *background* is changing quickly. The algorithm then visits each of the distinct regions in sm and assigns them a unique id, creating the segmentation map sm before deblending.

Algorithm 1. Segmentation

Input: Background probability map b, Specified marker set p (optional, same size as b)
Output: Labeled segmentation map sm
$m \leftarrow$ zero matrix same size as b
for m_{ij} in m do
  if $b_{ij} = 1$ then
    $m_{ij} \leftarrow 1$
  end
  else if $b_{ij} = 0$ or $p_{ij} > 0$ then
    $m_{ij} \leftarrow 2$
  end
end
$s \leftarrow$ SOBEL(b)
$sm \leftarrow$ WATERSHED(s, m)
$id \leftarrow 1$
for each contiguous set of pixels $y > 0$ in sm do
  for pixel y_{ij} in y do
    $y_{ij} \leftarrow id$
  end

(Continued)

  $id \leftarrow id + 1$
end
return sm

Where SOBEL is the Sobel algorithm (Sobel & Feldman 1968) and WATERSHED is the watershed algorithm (Couprie & Bertrand 1997). The optional parameter p allows for pixel locations to be specified, such as the locations of known sources, and used as generating points for the watersheding operation.

4.2. Deblending

The algorithm described in Section 4.1 provides a collection of segmented regions of contiguous areas, each with a unique index. Since this algorithm identifies contiguous regions of low *background*, neighboring sources with overlapping flux in the science images will be blended by the segmentation algorithm. The deblending algorithm used in *Morpheus* is *ad hoc* and is primarily designed to separate the segmented regions into distinct subregions containing a single pre-defined object. The locations of these objects may be externally specified, such as catalog entries from a source catalog (e.g., 3D-HST sources), or they may be internally derived from the science images themselves (e.g., local flux maxima).

The deblending algorithm we use applies another round of the watershed operation on each of the distinct regions identified by the segmentation algorithm, using the local flux distributions from the negative of a science image (e.g., $F160W$) as the basins to fill and object locations as the marker set. We assign the resulting subdivided segmentations a distinct *subid* in addition to their shared *id*, allowing us to keep track of adjacent deblended regions that share the same parent segmentation region. The *subid* of deblended sources is indicated by decimal values, and the parent *id* is indicated by the whole number of the *id*. For example, if a source with $id = 8$ was actually two sources, after deblending, the two deblended sources would have id values 8.1 and 8.2.

In testing *Morpheus*, we find that the deblending algorithm may shred extended sources like large disks or point-source diffraction spikes. However, the *Morpheus* algorithm

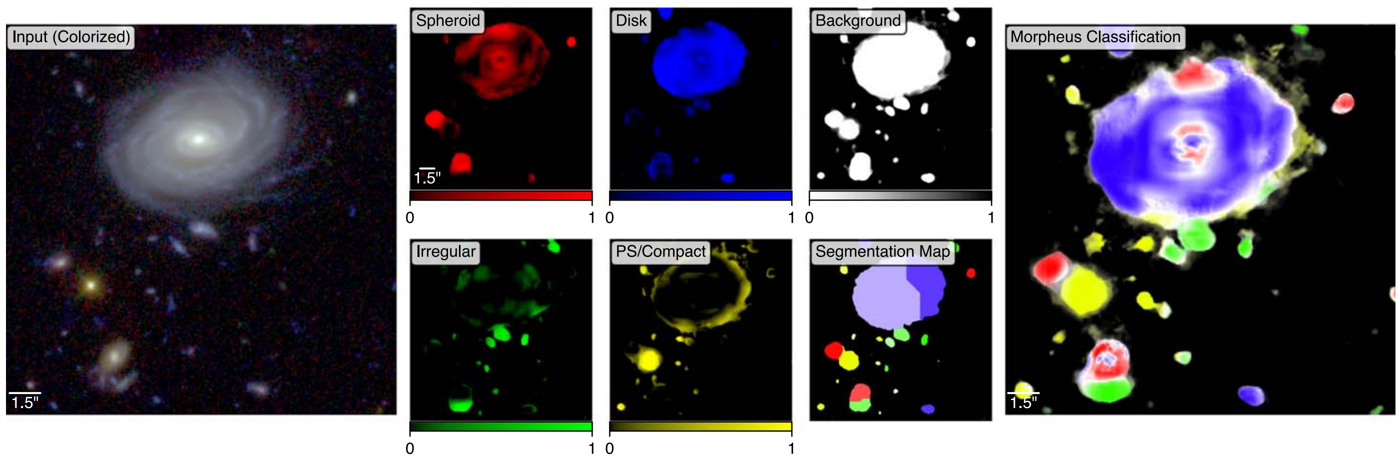

Figure 5. *Morpheus* morphological classification results for a region of the GOODS South field. The far left panel shows a three-color composite *VzH* image. The scale bar indicates 1$''$.5. The *V*, *z*, *J*, and *H* FITS images are supplied to the *Morpheus* framework, which then returns images for the *spheroid* (red–black panel), *disk* (blue–black panel), *irregular* (green–black panel), *point source/compact* (yellow–black panel), and *background* (white–black panel) classifications. The pixel values of these images indicate the local dominant *Morpheus* classification, normalized to sum to one across all five classifications. The panel labeled "Segmentation Map" is also generated by *Morpheus*, using the 3D-HST survey sources as generating locations for the segmentation Algorithm 1. The regions in the segmentation map are color-coded by their flux-weighted dominant class computed from the *Morpheus* classification values. The far right panel shows the *Morpheus* "classification color" image, where the pixel hues indicate the dominant morphological classification, and the intensity indicates 1 − *background*. The saturation of the *Morpheus* color image indicates the difference between the dominant classification value and the second-most-dominant classification, such that white regions indicate pixels where *Morpheus* returns a comparable result for multiple classes. See Section 6.1.6 for more details.

successfully deblends some small or faint sources proximate to bright sources that are missing from the 3D-HST catalog.

Algorithm 2. Deblending

Input: Segmentation map sm, flux image h, minimum radius between flux peaks r, maximum number of deblended subregions $nd_{\max}$, Specified marker set p (optional, same size as sm)
Output: Deblended segmentation map db
if p is not specified then
  $idc \leftarrow 10^{\lceil \log_{10} nd_{\max} \rceil}$ ($\lceil\rceil$ indicate ceiling operation)
  $sm \leftarrow idc \times sm$
end
for each contiguous set of source pixels $s > 0$ in sm do
  $h_{local} \leftarrow$ subset of h corresponding to s
  if p is specified then
    $p_{local} \leftarrow$ subset of p corresponding to s
    if p_{local} contains more than one id then
      $s \leftarrow$ WATERSHED$(-h_{local}, p_{local})$
    end
    else
      $s \leftarrow$ MAX(p_{local})
    end
  end
  else
    $idx \leftarrow$ PEAKLOCALMAXIMA(h_{local}, r, c)
    if COUNT$(idx) > 1$ then
      $subid \leftarrow 1$
      $m \leftarrow$ a zero matrix same size as s
      for indices i, j in idx do
        $m_{ij} \leftarrow subid$
        $subid \leftarrow subid + 1$
      end
      $s \leftarrow$ WATERSHED$(-h_{local}, m)$
    end
  end
end
if p is not specified then
  $db \leftarrow idc^{-1} \times sm$

(Continued)

  end
else
  $db \leftarrow sm$
end
return db

Where WATERSHED is the watershed algorithm (Couprie & Bertrand 1997). PEAKLOCALMAXIMA(x, y, z) returns a list of tuples marking the pixel locations of at most z local maxima in x that lie at least $2y$ pixels apart, as implemented by van der Walt et al. (2014). COUNT returns the number of elements in a collection. MAX returns the maximum element from a matrix. The optional parameter p allows for pixel locations to be specified, such as the locations of known sources, and used as generating points for the watersheding.

5. Object-level Classification

While *Morpheus* uses a semantic segmentation model to enable pixel-level classification of astronomical images using a deep learning framework, some applications, like the morphological classification of galaxies, additionally require object-level classification. *Morpheus* aggregates pixel-level classifications into an object-level classification by using a flux-weighted average.

Figure 5 shows the results of the *Morpheus* pixel-level classification for an example area of the CANDELS region of GOODS South. The leftmost panel shows a three-color *VzH* composite of the example area for reference, though *Morpheus* operates directly on the science-quality *VzJH* FITS images. The central panels show the output pixel classifications (i.e., q from Section 3.3) for the *background*, *spheroid*, *disk*, *irregular*, and *point source/compact* classes, with the intensity of each pixel indicating the normalized probability $q_{ijk} \in [0,\ 1]$. The segmentation map resulting from the algorithms described in Section 4 is also shown in as a central panel. The rightmost panel shows a color composite of the *Morpheus* pixel-level classification, with the color of each pixel indicating its dominant class and the saturation of the pixel being proportional to the difference Δq between the dominant and second-most-dominant class. White pixels then indicate regions where the model did not strongly distinguish between two classes, such as in transition regions in the image between two objects with different morphological classes. The pixel intensities in the pixel-level classification image are set to $1 - background$ and are not flux-weighted. The dominant classification for each object, as determined by *Morpheus*, is often clear visually. The brightest objects are well-classified and agree with the intuitive morphological classifications an astronomer might assign based on the *VzH* color composite image. Faint objects in the image have less morphological information available and are typically classified as *point source/compact*, in rough agreement with their classifications in the K15 training set. However, these visual comparisons are qualitative, and we now turn to quantifying the object-level classification from the pixel values.

Consider a deblended object y containing a total of n_o contiguous pixels of arbitrary shape within a flux image, and a single index $i = [1, n_o]$ scanning through the pixels in y. Each class $k \in [1, n_c]$ in the distribution of classification probabilities $\boldsymbol{Q}$ for the object is computed as

$$Q_k = \frac{\sum_{i=1}^{n_o} y_i q_{ik}}{\sum_{i=1}^{n_o} y_i}. \quad (16)$$

Here, y represents the pixel region in a science image assigned to the object, and y_i is the flux in the ith pixel of the object. The quantity q_{ik} is the kth classification probability of the ith pixel in y. Equation (16) represents object-level classification computed as the flux-weighted average of the pixel-level classifications in the object.

6. *Morpheus* Data Products

Before turning the quantifications of the object-level performance, we provide a brief overview of the derived data products produced by *Morpheus*. A more detailed description of the data products is presented in Appendix D, where we describe a release of pixel-level morphologies for the five CANDELS fields and 3D-HST value-added catalog, including object-level morphologies. The HLF (Illingworth et al. 2016) GOODS South v2.0 release and 3D-HST survey (Momcheva et al. 2016) are the primary focus of the analysis of the *Morpheus*' performance owing to their depth and completeness.

As described in Section 5, *Morpheus* produces a set of n_c "classification images" that correspond to the pixel-by-pixel model estimates q_{ij} for each class, normalized across classes such that $\sum_k q_{ijk} = 1$. The value of each pixel is, therefore, bounded ($q_{ijk} \in [0, 1]$). The classification images are stored in FITS format, and inherit the same ($N \times M$) pixel dimensions as the input FITS science images provided to *Morpheus*. When presenting classification images used in this paper, we represent *background* images in negative grayscale, *spheroid* images in black–red, *disk* images in black–blue, *irregular* images in black–green, and *point source/compact* images in black–yellow color scales. Figure 5 shows *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* images (central panels) for a region of CANDELS GOODS South.

Given the separate classification images, we can construct what we deem a "*Morpheus* morphological color image" that indicates the local dominant class for each pixel. To produce a Red–Green–Blue (RGB) false color image to represent the morphological classes visually, we use the Hue-Saturation-Value (HSV) color space and convert from HSV to RGB via standard conversions. In the HSV color space, the "hue" image indicates a hue on the color wheel, "saturation" provides the richness of the color (from white or black to a deep color), and "value" sets the brightness of a pixel (from dark to bright). On a color wheel of hues, $\mathcal{H} \in [0, 360]$ ranges from red ($\mathcal{H} = 0$) to red ($\mathcal{H} = 360$) through yellow ($\mathcal{H} = 120$), green ($\mathcal{H} = 180$), and blue ($\mathcal{H} = 240$), and we can assign hue pixel values corresponding to the dominant morphological class (*spheroid* as red, *disk* as blue, *irregular* as green, and *point source/compact* as yellow). We set the saturation of the image to be the Δq_{ijk} between the dominant class and the second-most-dominant class, such that cleanly classified pixels ($q_{ijk_{ij}^{\max}} \approx 1$, $\Delta q_{ijk} \approx 1$) appear as deep red, blue, green, or yellow, and pixels where *Morpheus* produces an indeterminate classification ($\Delta q_{ijk} \approx 0$) appear as white or desaturated. The "value" channel is set equal to $1 - background$, such that regions of low background containing sources are bright, and regions with high background are dark. Figure 5 also shows the *Morpheus* morphological color image (far right panel) for a region of CANDELS GOODS South.

6.1. Morphological Images for GOODS South

As part of our data products, we have produced *Morpheus* morphological images of the HLF v2.0 (Illingworth et al. 2016) reduction of GOODS South. These data products are used in Section 7 to quantify the performance of *Morpheus* relative to standard astronomical analyses, and we, therefore, introduce them here. The *Morpheus* morphological classification images for the HLF were computed as described in Section 2.3, feeding *Morpheus* subregions of the HLF *VzJH* images for processing and then tracking the distribution of output pixel classifications to select the best classification for each. The $\sim 10^8$ pixels in each classification image are then stitched back together to produce contiguous *background*, *spheroid*, *disk*, *irregular*, and *point source/compact* images for the entire HLF GOODS South.

6.1.1. Background Image

Figure 6 shows the *background* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *background* classification for each pixel is shown in negative grayscale, with black corresponding to $background = 1$ and white regions corresponding to $background = 0$. The *background* image is used throughout Section 7 to quantify the performance of *Morpheus* in object detection.

6.1.2. Spheroid Image

Figure 7 shows the *spheroid* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *spheroid* classification for each pixel is shown on a black-to-red colormap, with black corresponding to $spheroid = 0$ and red regions corresponding to $spheroid = 1$.

6.1.3. Disk Image

Figure 8 shows the *disk* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *disk* classification for each pixel is shown on a black-to-blue colormap, with black

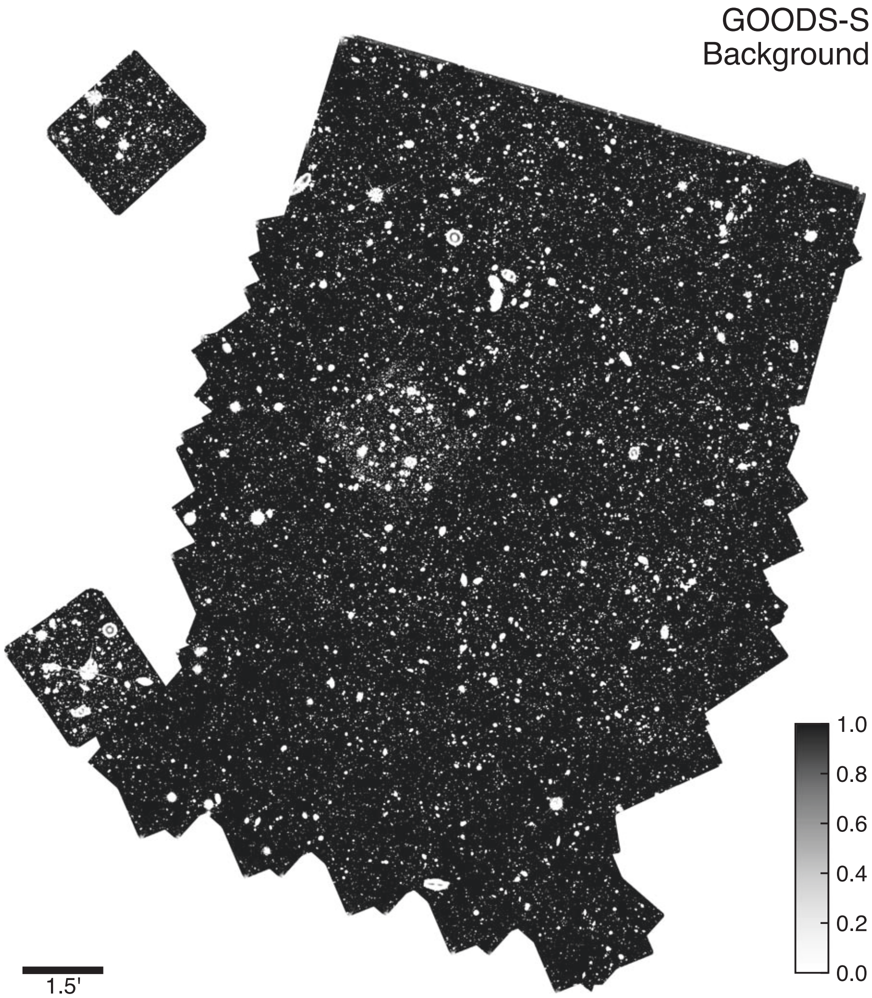

Figure 6. *Morpheus background* classification image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *background* class. The scale bar indicates $1\rlap{.}'5$. The color bar indicates the *background* $\in$ [0, 1], increasing from white to black. Correspondingly, the bright areas indicate regions of low background where sources were detected by *Morpheus*.

corresponding to *disk* = 0 and blue regions corresponding to *disk* = 1.

6.1.4. Irregular Image

Figure 9 shows the *disk* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *irregular* classification for each pixel is shown on a black-to-green colormap, with black corresponding to *irregular* = 0 and green regions corresponding to *irregular* = 1.

6.1.5. Point Source/Compact Image

Figure 10 shows the *point source/compact* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *point source/compact* classification for each pixel is shown on a black-to-yellow colormap, with black corresponding to *point source/compact* = 0 and yellow regions corresponding to *point source/compact* = 1.

6.1.6. Morphological Color Image

Figure 11 shows the *morphological color* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The false color image is constructed following Section 6, with the pixel intensities scaling with 1 − *background*, the pixel hues set according to the dominant class, and the saturation indicating the indeterminacy of the pixel classification. Pixels with a single dominant class appear as bright red, blue, green, or yellow for *spheroid*, *disk*, *irregular*, or *point source/compact* classifications, respectively. Bright white pixels indicate regions of the image where the model results were indeterminate in selecting a dominant class. Dark regions represent pixels that the model classified as *background*. We note that the pixel intensities are not scaled with the flux in the image, and the per-object classifications require a local flux weighting following Equation (16) and the process described in Section 5. This flux weighting usually results in a distinctive class for each object, since the bright regions of objects often have a dominant shared pixel classification. The outer regions of objects with low flux show more substantial variation in the per-pixel classifications, but these regions often do not contribute strongly to the flux-weighted per-object classifications computed from this morphological color image.

7. *Morpheus* Performance

Given the data products generated by *Morpheus*, we can perform a variety of tests to quantify the performance of the

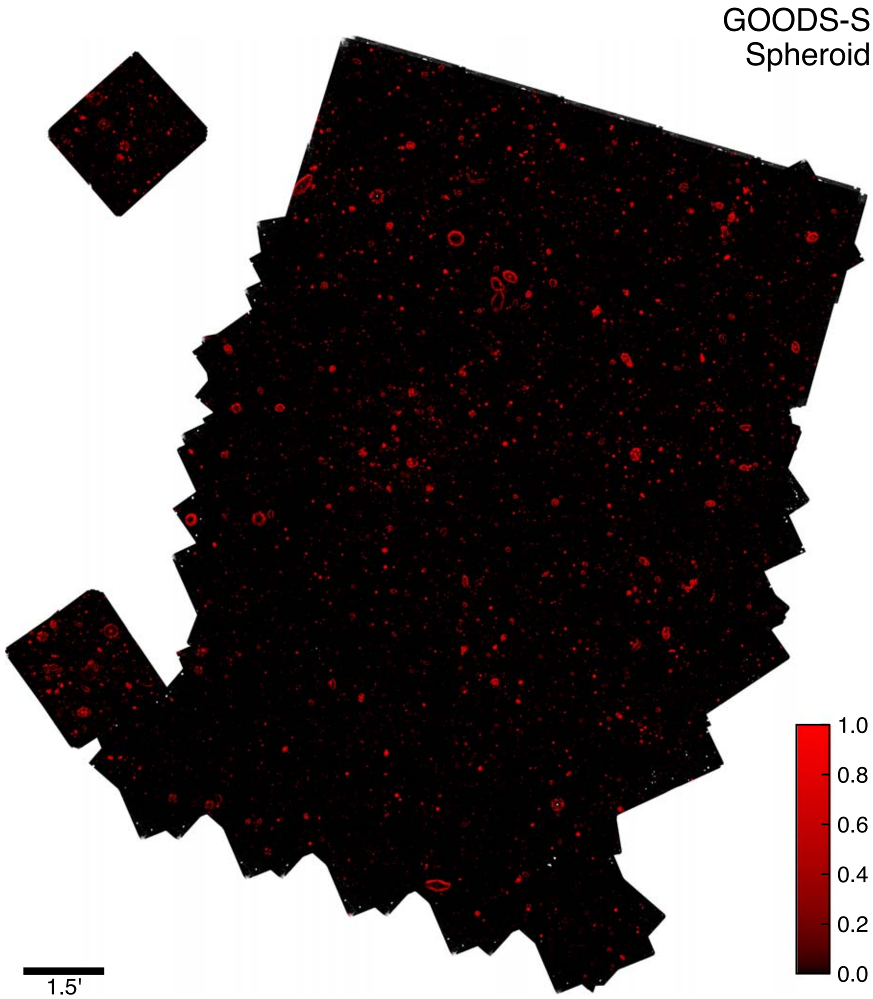

Figure 7. *Morpheus spheroid* classification image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *spheroid* class. The scale bar indicates $1\rlap{.}'5$. The color bar indicates the *spheroid* $\in$ [0, 1], increasing from black to red. Correspondingly, the bright red areas indicate pixels where *Morpheus* identified *spheroid* objects.

model. There are basic performance metrics relevant to how the model is optimized, reflecting the relative agreement between the output of the model and the training data classifications. However, given the semantic segmentation approach of *Morpheus* and the pixel-level classification it provides, there are additional performance metrics that can be constructed to mirror widely used performance metrics in more standard astronomical analyses including the completeness of sources detected by *Morpheus* as regions of low background. In what follows, we attempt to address both kinds of metrics and provide some ancillary quantifications to enable translations between the performance of *Morpheus* as a deep learning framework and as an astronomical analysis tool. In particular, we focus our analysis on the 3D-HST catalog and HLF reduction of the GOODS South region in the CANDELS Survey.

7.1. Object-level Morphological Classifications

The semantic segmentation approach of *Morpheus* provides classifications for each pixel in an astronomical image. These pixel-level classifications can then be combined into object-level classifications $\boldsymbol{p}$ using the flux-weighted average described by Equation (16). The *Morpheus* object-level classifications can then be compared directly with a test set of visually classified object morphologies provided by Kartaltepe et al. (2015).

To understand the performance of *Morpheus* relative to the K15 visual classifications, we present some summary statistics of the training and test sets pulled from the K15 samples. During training, the loss function used by *Morpheus* is computed relative to the distribution of input K15 classifications for each object and not only their dominant classification. The goal is to retain a measure of the uncertainty in visual classifications for cases where the morphology of an object is not distinct.

7.1.1. Distribution of Training Sample Classifications

Galaxies in the K15 training set have been visually classified by multiple experts, providing a distribution of possible classifications for each object in the sample. Figure 12 presents histograms of the fraction of K15 classifiers recording votes for *spheroid*, *disk*, *irregular*, and *point source/compact* classes for each object. Only classes with more than one vote are plotted.

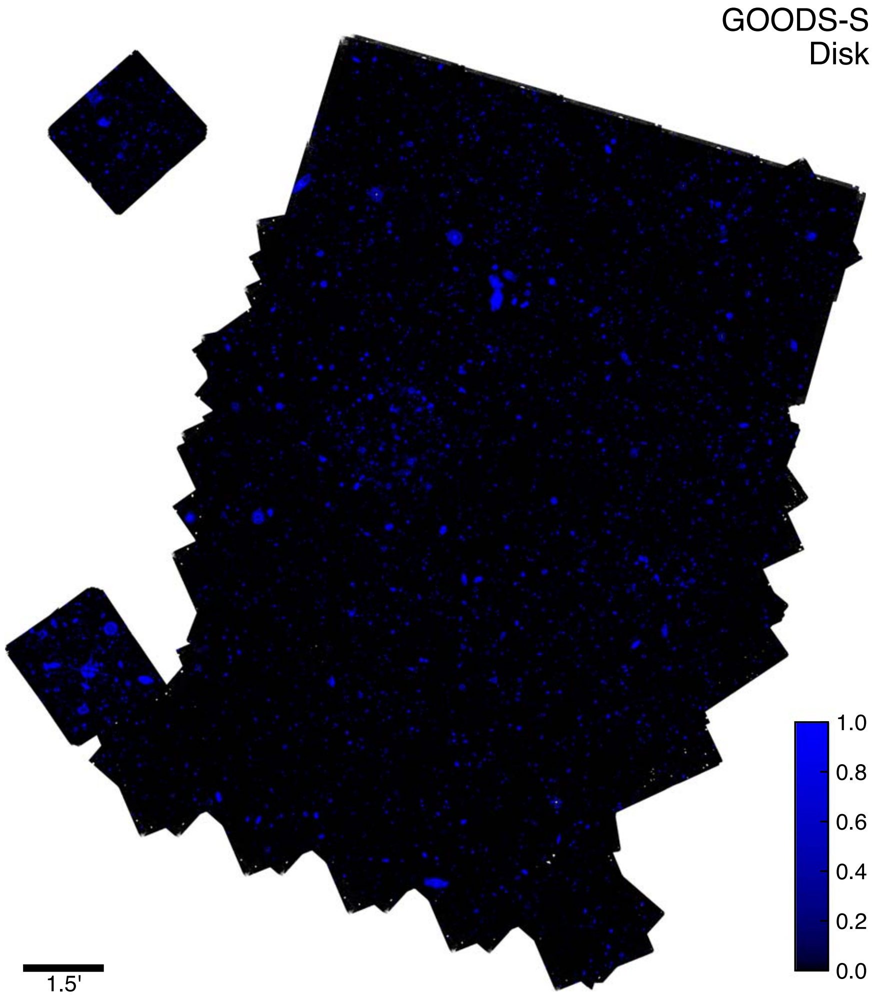

Figure 8. *Morpheus disk* classification image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *disk* class. The scale bar indicates $1'\!.5$. The color bar indicates the *disk* $\in [0, 1]$, increasing from black to blue. Correspondingly, the bright blue areas indicate pixels where *Morpheus* identified *disk* objects.

7.1.2. Classification Agreement in Training Sample

To aid these comparisons, we introduce the *agreement* statistic

$$a(\boldsymbol{p}) = 1 - \frac{H(\boldsymbol{p})}{\log(n_c)} \tag{17}$$

where $\boldsymbol{p}$ is the distribution of classifications and n_c is the number of classes. The quantity

$$H(\boldsymbol{p}) \equiv -\sum_{k=1}^{n_c} p_k \log p_k \tag{18}$$

is the self entropy. According to these definitions, $H(\boldsymbol{p}) \in [0, \log n_c]$ and $a(\boldsymbol{p}) \in [0, 1]$. The agreement $a(\boldsymbol{p}) \to 1$ when the distribution of classifications $\boldsymbol{p}$ is concentrated in a single class, and $a(\boldsymbol{p}) \to 0$ when the classifications are equally distributed. For reference, $a(\boldsymbol{p}) \approx 0.57$ for two equal classes and $a(\boldsymbol{p}) \approx 0.8$ for a 90%/10% split between two classes for $n_c = 5$ possible classes.

7.1.3. Training and Test Set Statistics

The K15 classifications have substantial variation in their agreement $a(\boldsymbol{p})$. Figure 13 shows histograms and the cumulative distribution of $a(\boldsymbol{p})$ for objects with *spheroid*, *disk*, *irregular*, and *point source/compact* dominant classes. These distributions of $a(\boldsymbol{p})$ are roughly bimodal, consisting of a single peak near $a(\boldsymbol{p}) = 1$ and a broader peak near $a(\boldsymbol{p}) \approx 0.5$ with a tail to larger $a(\boldsymbol{p})$. As the cumulative distributions indicate, roughly 20%–60% of objects in the K15 sample had perfect agreement in their morphological classification, with *disk* and *point source/compact* being the most distinctive classes.

The breadth in the agreement statistic for the input K15 data indicates substantial variation in how expert astronomers would visually classify individual objects. As these data are used to train *Morpheus*, understanding exactly what *Morpheus* should reproduce requires further analysis of the K15 data. An important characterization of the input K15 data is the confusion matrix of object classifications. This matrix describes the typical classification distribution for objects of a given dominant class. Figure 14 presents the confusion matrix for the K15 classifications, showing the typical spread in classifications for objects assigned *spheroid*, *disk*, *irregular*,

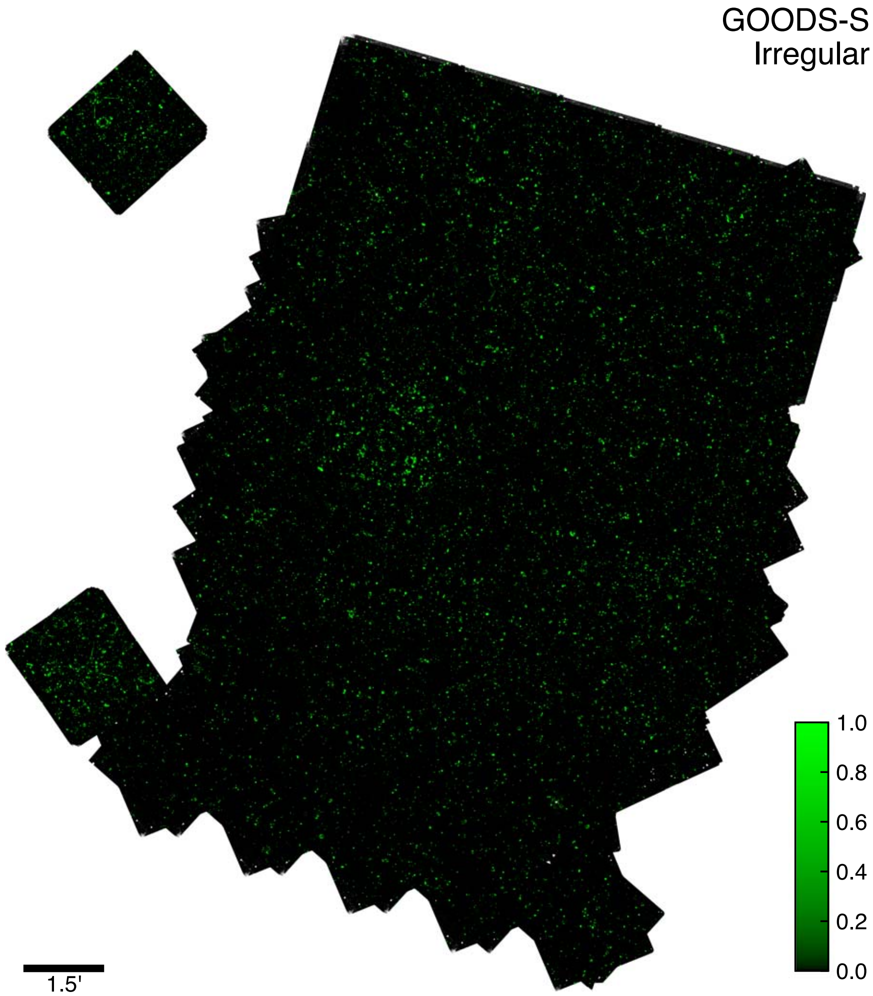

Figure 9. *Morpheus irregular* classification image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *irregular* class. The scale bar indicates $1\overset{\prime}{.}5$. The color bar indicates the *irregular* $\in [0, 1]$, increasing from black to green. Correspondingly, the bright green areas indicate pixels where *Morpheus* identified *irregular* objects.

or *point source/compact* dominant morphologies. For reference, a confusion matrix for a distribution with perfect agreement is the identity matrix. Figure 14 provides some insight into the natural degeneracies present in visually classified morphologies. Objects with a dominant *disk* classification are partially classified as *spheroid* (10%) and *irregular* (11%). The *irregular* objects frequently receive an alternative *disk* classification (19%). The *point source/compact* objects also are assigned *spheroid* classifications (14%). Objects with a dominant *spheroid* class have the highest variation and receive substantial *disk* (18%) and *point source/compact* (11%) classifications. This result is consistent with Figure 13, which shows a relatively large disagreement for objects with a dominant *spheroid* classification.

Since *Morpheus* is trained to reproduce the distribution of K15 classifications, the confusion matrix between the dominant *Morpheus* classifications and the K15 classification distributions should be similar to that in Figure 14. Indeed, Figure 15 shows that the distribution of K15 classifications for objects with a given dominant *Morpheus* classification agrees well with the input K15 distributions shown in Figure 14. This result demonstrates that *Morpheus* reproduces well the intrinsic uncertainty in the K15 classifications, as measured by the distribution of morphologies, recovered for a given K15 dominant classification.

The ability of *Morpheus* to reproduce the distribution of K15 classifications is not the only metric of interest, as it does not indicate whether the object-by-object *Morpheus* classifications agree with the K15 classifications for objects with distinctive morphologies. Figure 13 shows that 20%–60% of objects in the K15 classifications have an agreement $a(\boldsymbol{p}) = 1$, meaning that all K15 visual classifiers agreed on the object morphology. The confusion matrix for these distinctive objects constructed from the K15 data is diagonal, and the confusion matrix for these objects constructed from the *Morpheus* output should also be diagonal if *Morpheus* perfectly reproduced the object-by-object K15 classifications. Further, to ensure that *Morpheus* captures the distribution of the K15 morphologies, the cumulative distributions of dominant K15 morphologies and dominant *Morpheus* morphologies as a function of color were compared using a two-sample Kolmogorov–Smirnov test. For each morphology, the *p*-values ($p = 0.3$–0.99) indicate consistency between the Morpheus and K15 distributions as a function of color. These results suggest that *Morpheus* accurately captures the K15 representation of morphology without significant color bias.

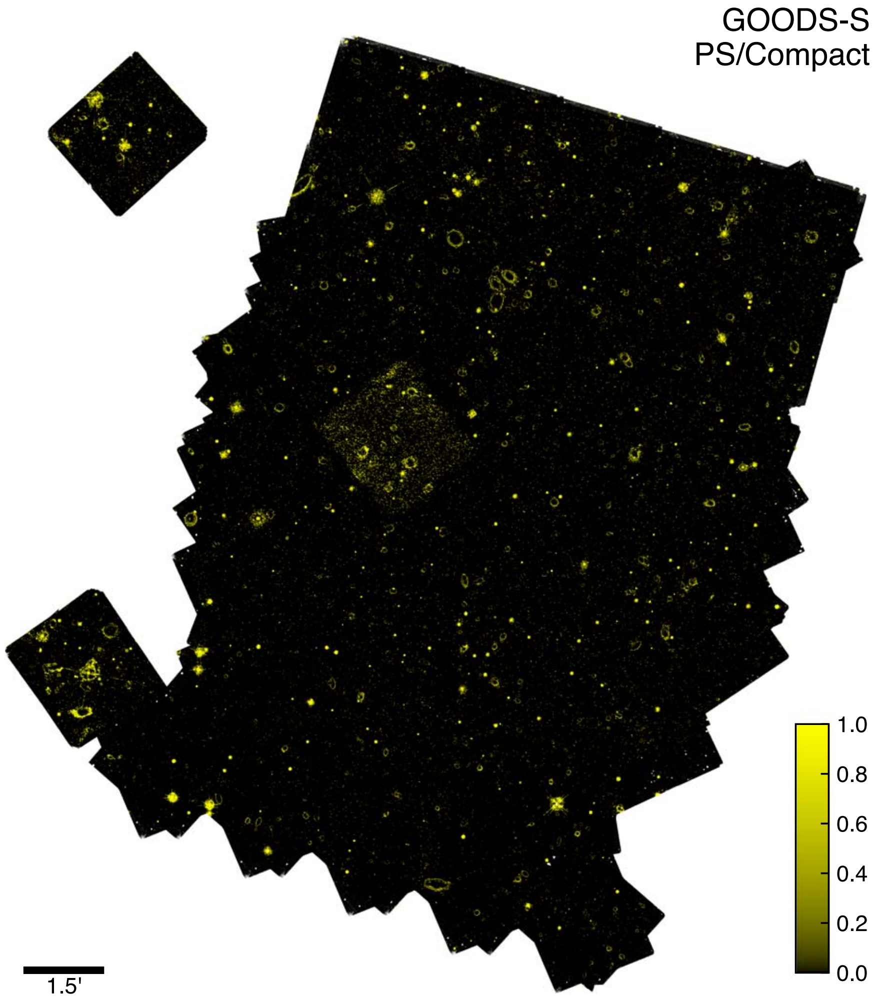

Figure 10. *Morpheus point source/compact* classification image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *point source/compact* class. The scale bar indicates 1.′5. The color bar indicates the *point source/compact* $\in$ [0, 1], increasing from black to yellow. Correspondingly, the bright yellow areas indicate pixels where *Morpheus* identified *point source/compact* objects.

To characterize the performance of *Morpheus* for the $a(\boldsymbol{p}) = 1$ K15 subsample, we used the *Morpheus* output classification images computed from the HLF GOODS South images. The flux-weighted *Morpheus* morphological classifications were computed following Equation (16) and using the K15 segmentation maps to ensure that the same pixels were being evaluated. Figure 16 presents the resulting confusion matrix showing the *Morpheus* dominant classification for each object's dominant classification determined by K15. As Figure 16 demonstrates, *Morpheus* achieves extremely high agreement ($\geqslant$90%) with K15 for *spheroid* and *point source/compact* objects, and good agreement ($\geqslant$80%) for *disk* and *irregular* objects with some mixing $\sim$15% between them. This performance is comparable to other object-by-object morphological classifications in the literature (e.g., Huertas-Company et al. 2015) but is constructed directly from a flux-weighted average of pixel-by-pixel classifications by *Morpheus* using real FITS image data of differing formats and depth.

7.1.4. Redshift Evolution of Morphology in CANDELS Galaxies

To illustrate the scientific applications of *Morpheus*, we examine the morphological distribution of $\sim$54,000 3D-HST sources in the five CANDELS fields as a function of redshift and stellar mass (Figure 17). We combine together the flux-weighted *Morpheus* classifications of galaxies identified in CANDELS with the 3D-HST stellar masses and redshift, dividing the sample into coarse redshift bins. The fraction of objects N/N_{tot} with a flux-weighted classification of *spheroid* (red), *disk* (blue), or *irregular* (green) is shown as a function of stellar mass for each redshift bin, along with Poisson uncertainties on the binned values. The well-established trends of increasing fractions of irregular objects at small masses and high redshifts are correctly reproduced by *Morpheus*, as well as the growth of the disk population at low redshifts. These results can be compared with the results reported in Figure 3 of Huertas-Company et al. (2016, HC16). To ensure comparable samples between HC16 and this work, the *Morpheus*-classified samples in Figure 17 are limited to objects with $H < 24.5$AB. Since HC16 and Morpheus use similar but not identical morphological classifications, we adapt the sample definitions used by HC16 to the *Morpheus* classification scheme. To be counted as a part of a morphological class, each galaxy's flux-weighted confidence value assigned by *Morpheus* must be greater than 0.7. This threshold ensures each classification is mutually exclusive but low enough to ensure a comparable sample size to HC16.

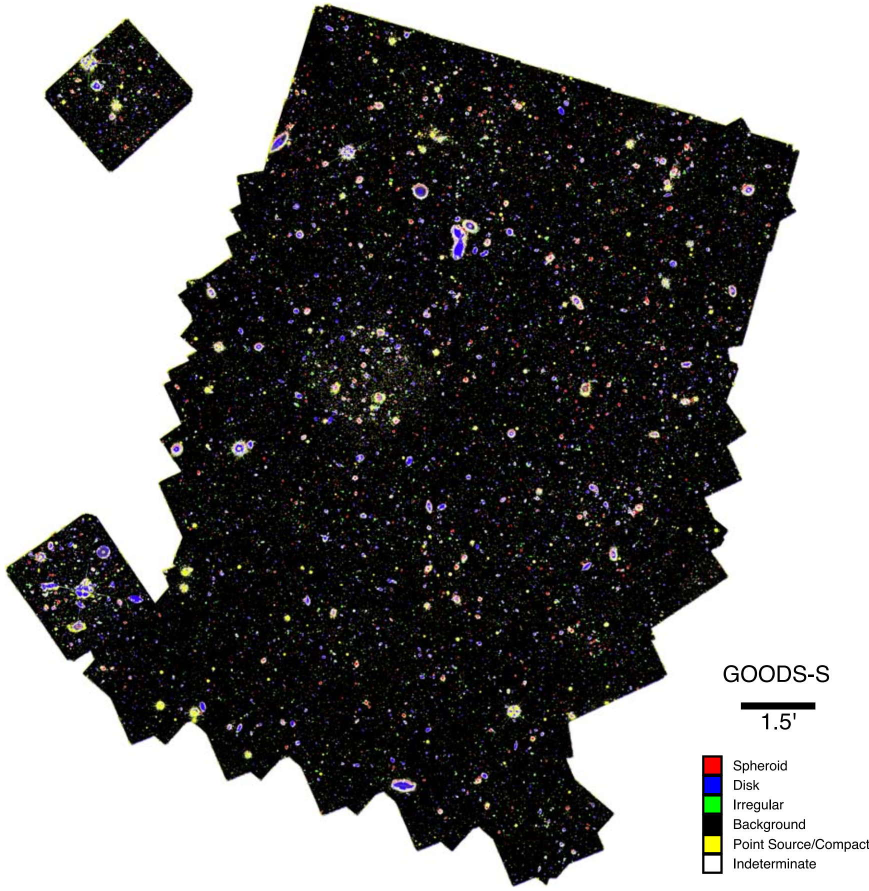

Figure 11. *Morpheus* morphological color image for the HLF (Illingworth et al. 2016) reduction of the CANDELS survey data (Grogin et al. 2011; Koekemoer et al. 2011) in GOODS South. The image intensity is set proportional to $1 - background$ for each pixel, such that regions of high background are black and regions with low background containing source pixels identified by *Morpheus* appear bright. The hue of each source pixel indicates its dominant classification, with *spheroid* shown as red, *disk* as blue, *irregular* as green, and *point source/compact* as yellow. The color saturation of each pixel is set to the difference between the first and second most dominant class values, such that regions with indeterminate morphologies as determined as *Morpheus* appear as white and regions with strongly determined classifications appear as deep colors. Note that the morphological color image is not flux-weighted, and the per-object classifications assigned by *Morpheus* include a flux-weighted average of the per-pixel classifications shown in this image.

The trends in Figure 17 agree with those found by HC16 in two important aspects. First, at lower redshifts, disks tend to dominate spheroids, and as redshift increases, spheroids tend to dominate disks. Second, irregular sources are a larger portion of the population than spheroids and disks at lower stellar masses and more become less abundant than spheroids and disks as stellar mass increases. The agreement between *Morpheus* and the results of HC16, which were based on object-level classifications, confirms the ability of *Morpheus* to capture source-level morphologies by aggregating pixel-level classifications.

7.2. Simulated Detection Tests

The *Morpheus* framework enables the detection of astronomical objects by producing a *background* classification image, with source locations corresponding to regions where *background* < 1. If generating points in the form of a source catalog are not supplied, the segmentation algorithm of *Morpheus* uses an even more restrictive condition that regions near sources must contain pixels with *background* $= 0$. Given that the semantic segmentation algorithm of *Morpheus* was trained on the K15 sample that has a completeness limit, whether the regions identified by *Morpheus* to have *background* $= 0$ correspond to an approximate flux limit should be tested. Similarly, whether noise fluctuations lead to regions assigned *background* ≈ 0 in error should also be evaluated.

Below, we summarize detection tests for *Morpheus* using simulated images. For these tests, a simulated sky background was generated using Gaussian random noise with rms scatter measured in $0\farcs5$ apertures after convolving with a model HST point-spread function (PSF) and scaled to that measured from the K15 training images. The Tiny Tim software (Krist et al. 2011) software was used to produce the PSF models appropriate for each band.

7.2.1. Simulated False-positive Test

For a large enough image of the sky, random sampling of the noise could produce regions with local fluctuation of some factor f above the rms background σ and lead to a false-positive detection. A classical extraction technique using aperture flux thresholds would typically identify such regions as a signal-to-noise ratio $(\mathrm{S/N}) = f$ source. Here, we evaluate whether *Morpheus* behaves similarly.

Using the Gaussian random noise field, single-pixel fluctuations were added to the *H*-band only such that the local flux measured in a $0\farcs5$ aperture after convolving with Tiny Tim corresponded to $\mathrm{S/N} = [0.5, 1, 2, 3, 4, 5, 6, 7, 10]$. The false signals were inserted at well-separated locations such that

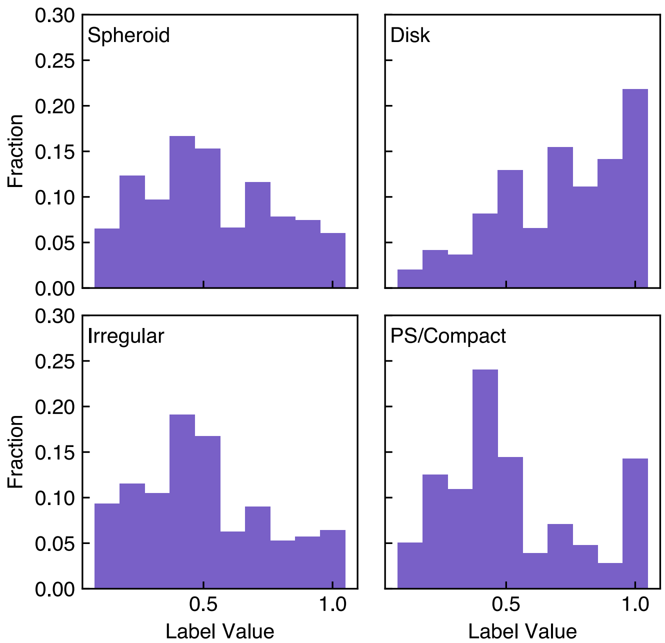

Figure 12. Distribution of morphological classifications in the Kartaltepe et al. (2015) sample, which serve as a training sample for *Morpheus*. Shown are histograms of the fraction of sources with a nonzero probability of belonging to the *spheroid* (upper left panel), *disk* (upper right panel), *irregular* (lower left panel), or *point source/compact* classes (lower right panel), as determined via visual classification by expert astronomers. The histograms have been normalized to show the distribution of classification probabilities for each class, and they consist of ≈7,600 sources.

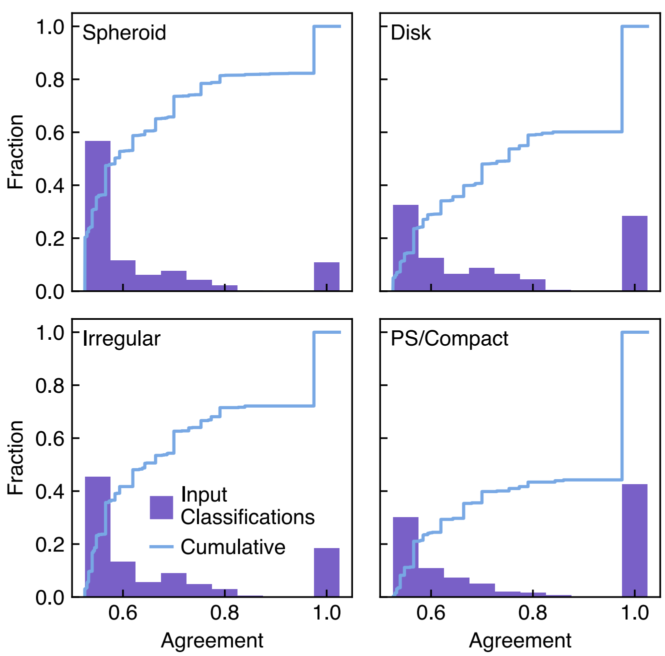

Figure 13. Histograms (purple) and cumulative distribution (blue lines) of agreement $a(\boldsymbol{p})$ for the Kartaltepe et al. (2015, K15) visual morphological classifications, for objects with *spheroid* (upper left panel), *disk* (upper right panel), *irregular* (lower left panel), and *point source/compact* (lower right panel) as their dominant classification. Agreement $a(\boldsymbol{p})$ (see Equation (17) for a definition) characterizes the breadth of the distribution of morphological classes assigned by the K15 classifiers for each object, with $a(\boldsymbol{p}) = 1$ indicating perfect agreement of a single class and $a(\boldsymbol{p}) = 0$ corresponding to perfect disagreement with equal probability among classes. The distribution of agreement in the K15 training classifications is roughly bimodal, with a strong peak near-perfect agreement and a broader peak near $a(\boldsymbol{p}) \approx 0.5$, close to the agreement value for an even split between two classes.

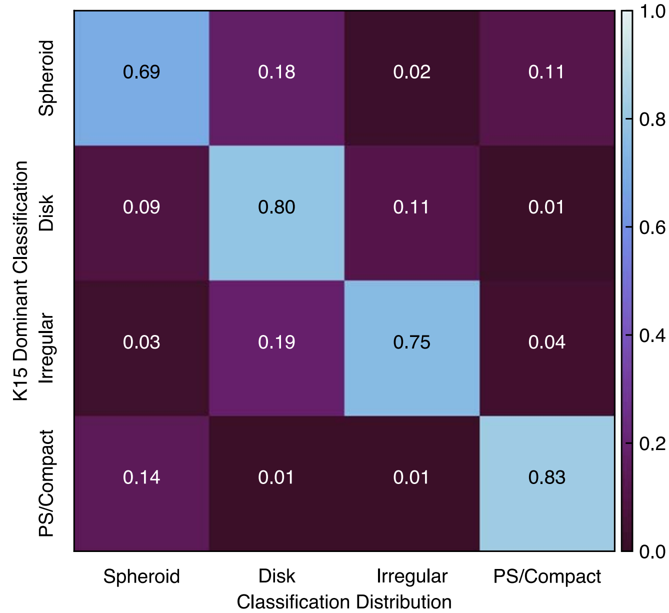

Figure 14. Confusion matrix for the distribution of K15 morphological classifications. Shown is the distribution of morphologies assigned by K15 visual classifiers for objects of a given dominant classification. Objects with a dominant *spheroid* class show the most variation, with frequent additional *disk* and *point source/compact* morphologies assigned. The most distinctive dominant class is *point source/compact*, which also receives a *spheroid* classification in 14% of objects. The off-diagonal components of the confusion matrix indicate imperfect agreement among the K15 classifiers, consistent with the distributions of the agreement statistic shown in Figure 13.

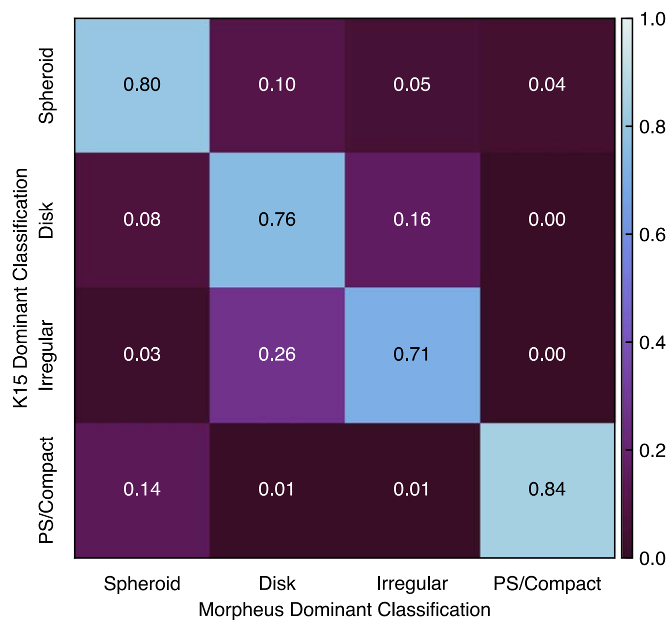

Figure 15. Confusion matrix showing the spread in *Morpheus* dominant classifications for objects with given K15 dominant classifications. The *Morpheus* framework is trained to reproduce the input K15 distributions, and this confusion matrix should, therefore, largely match that of Figure 14. The relative agreement between the two confusion matrices demonstrates that the *Morpheus* output can approximate the input K15 classification distributions.

Morpheus evaluated them independently. The V, z, and J images were left as blank noise, and then all four images were supplied to *Morpheus*. We find that *Morpheus* assigns none of these fake signals pixels with *background* $= 0$. However, the $\mathrm{S/N} = 7$ and $\mathrm{S/N} = 10$ regions have some *background* < 1

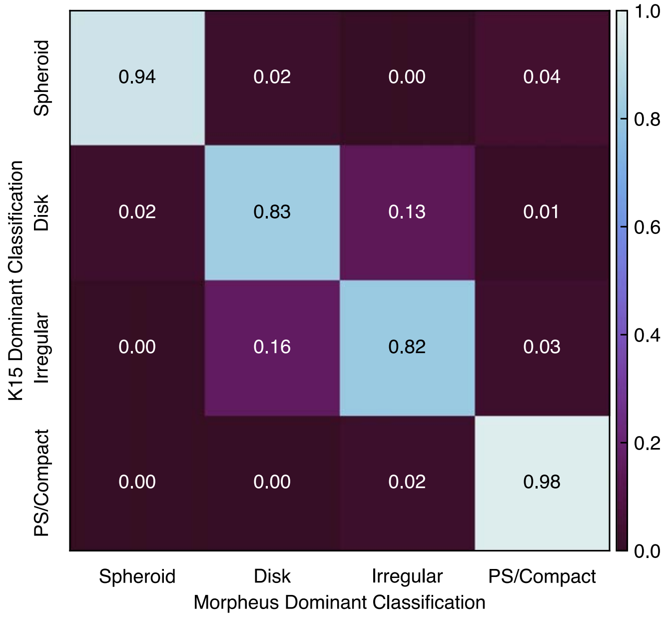

Figure 16. Confusion matrix quantifying the spread in *Morpheus* dominant classifications for K15 objects with a distinctive morphology. Shown are the output *Morpheus* classification distributions for K15 objects where all visual classifiers agreed on the input classification. The *Morpheus* pixel-by-pixel classifications computed for the HLF GOODS South images were aggregated into flux-weighted object-by-object classifications following Equation (16) using the K15 segmentation maps. The results demonstrate that *Morpheus* can reproduce the results of the dominant K15 visual classifications for objects with distinct morphologies, even as the *Morpheus* classifications were computed from per-pixel classifications using different FITS images of the same region of the sky.

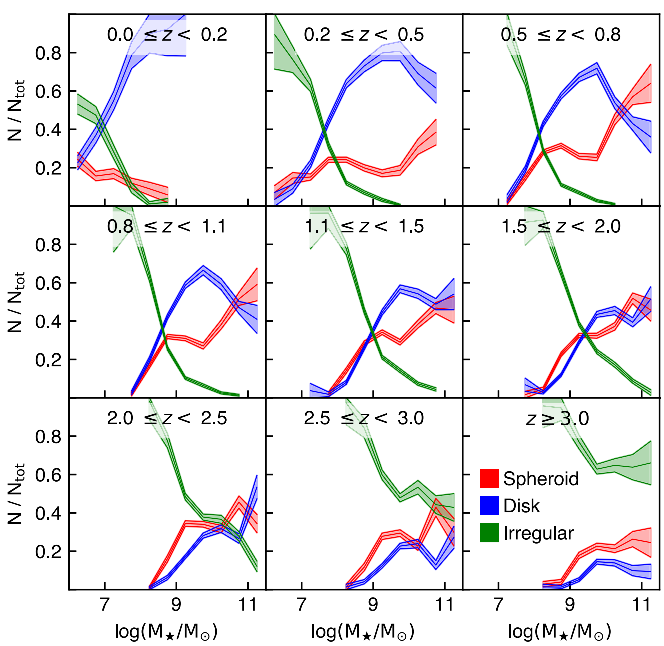

Figure 17. Morphology as a function of stellar mass and redshift for 54,000 sources in the five CANDELS fields. Sources included in the plot are those where $H < 24.5$AB and the *Morpheus* confidence for *spheroid*, *disk*, or *irregular* is greater than 0.7. See Section 7.1.4.

pixels, and while in the default algorithm, *Morpheus* would not assign these regions segmentation maps, a more permissive version of the algorithm could. An alternative test was performed by replacing the $S/N = 10$ noise fluctuation in the H-band image with a Tiny Tim H-band PSF, added after the convolution step with an amplitude corresponding to $S/N = 10$ measured in a $0\farcs5$ aperture. This test evaluates whether the shape of flux distribution influences the detection of single-band noise fluctuations. In this case, the minimum pixel values decreased to *background* ≈ 0.05 for a single band $S/N = 10$ fluctuation shaped like an H-band PSF, but did not lead to a detection. We conclude that *Morpheus* is robust to false positives arising from relatively large ($S/N \lesssim 7$) noise fluctuations.

7.2.2. False-negative Test

Given that *Morpheus* seems insensitive to false positives from noise fluctuations, it may also miss real but low S/N sources. By performing a similar test to that presented in Section 7.2.1 but with sources inserted in all bands rather than noise fluctuations inserted in a single band, the typical S/N where *Morpheus* becomes incomplete for real objects can be estimated.

Noise images were generated to have the same rms noise as the K15 images by convolving Gaussian random variates with the Tiny Tim (Krist et al. 2011) model for the HST PSF. An array of well-separated point sources modeled by the PSF were then inserted with a range of $S/N \in [1, 25]$ into all four input band images. The *Morpheus* model was then applied to the images, and the output *background* image analyzed to find regions with *background* below some threshold value. Figure 18 shows the number of pixels below various *background* threshold values assigned to objects with different S/Ns. Below about $S/N \sim 15$, the number of pixels identified as low *background* begins to decline rapidly. We therefore expect *Morpheus* to show incompleteness in real data for $S/N \lesssim 15$ sources. However, we emphasize that this limitation likely depends on the training sample used. Indeed, the K15 training data set is complete to $H = 24.5$AB in images with 5σ source sensitivities of $H \approx 27$AB. If trained on deeper samples, *Morpheus* may prove more complete to fainter magnitudes. We revisit this issue in Section 7.4 below, but we will explore training *Morpheus* on deeper training sets in future work.

7.3. Morphological Classification versus Surface Brightness Profile

In this paper, the *Morpheus* framework is trained on the K15 visual classifications to provide pixel-level morphologies for galaxies. The K15 galaxies are real astronomical objects with a range of surface brightness profiles for a given dominant morphology. Correspondingly, the typical classification that *Morpheus* would assign to idealized objects with a specified surface brightness profile is difficult to anticipate without computing it directly. Understanding how *Morpheus* would classify idealized galaxy models can provide some intuition about how the deep learning framework operates and what image features are related to output *Morpheus* classifications.

Figure 19 shows the output *Morpheus* classification distribution for simulated objects with circular Sérsic (1968) surface brightness profiles, for objects with $S/N = 20$, Sérsic indices $\eta \in [1, 9]$, and effective radii ranging from three to nine pixels. Synthetic FITS images for each object in each band were constructed by assuming zero color gradients and a flat f_ν spectrum, populating the image with a Sérsic profile object and

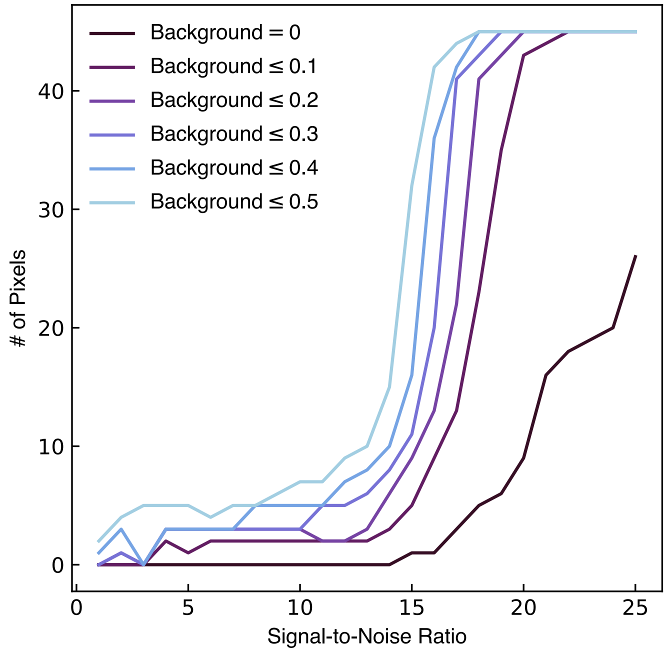

Figure 18. False-negative test for the *Morpheus* source detection scheme. Simulated sources with different S/Ns were inserted into a noise image and then recovered by *Morpheus*, which assigns a low *background* value to regions it identifies as containing source flux (see Section 7.2.2). Shown are lines corresponding to the number of pixels assigned to sources of different S/Ns, as a function of the *background* threshold. As trained on the K15 sample, *Morpheus* becomes incomplete for objects with S/N $\lesssim$ 15, and it is more complete if the threshold for identifying sources is made more permissive (i.e., at a higher *background* value).

noise consistent with the K15 images, and then convolving the images with a Tiny Tim point-spread function model appropriate for each input HST filter.

The results from *Morpheus* reflect common expectations for the typical Sérsic profile of morphological classes. Objects with $\eta = 1$ were typically classified as *disk* or *spheroid*, while intermediate Sérsic index objects (e.g., $\eta \approx 2$–3) were classified as *spheroid*. More compact objects, with Sérsic indices $\eta \geqslant 4$, were dominantly classified as *point source/compact*. Also, as expected for azimuthally symmetric surface brightness profiles, *Morpheus* did not significantly classify any objects as *irregular*. Figure 20 provides a complementary summary of the *Morpheus* classification of Sérsic profile objects, showing a matrix indicating the dominant classification assigned for each pair of [η, R_e] values. The *Morpheus* model classifies large objects with low η as *disk*, large objects with high η as *spheroid*, and small objects with high η as *point source/compact*.

Overall, this test indicates that for objects with circular Sérsic profiles, *Morpheus* reproduces the expected morphological classifications and that asymmetries in the surface brightness are needed for *Morpheus* to return an *irregular* morphological classification.

7.4. Source Detection and Completeness

The semantic segmentation capability of *Morpheus* allows for the detection of astronomical objects directly from the pixel classifications. In its simplest form, this object detection corresponds to regions of the output *Morpheus* classification images with low *background* class values. However, the *Morpheus* object detection capability raises several questions.

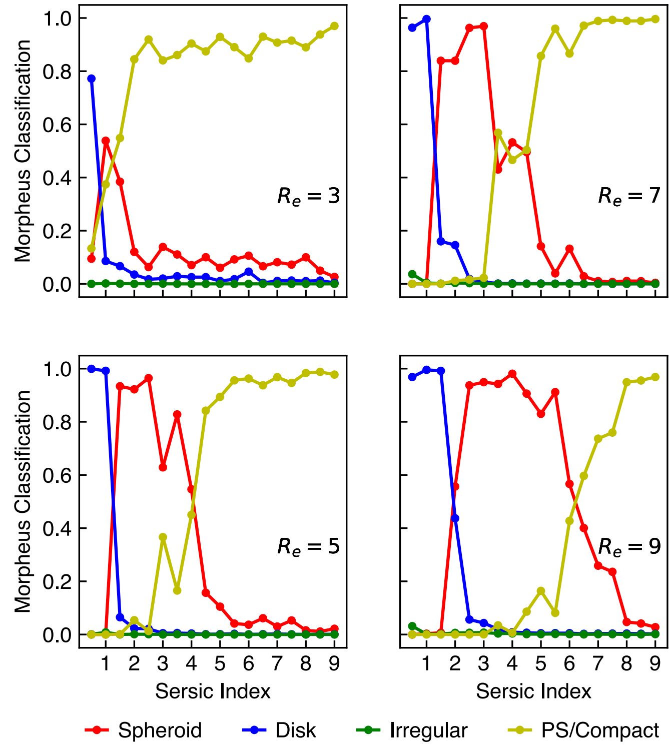

Figure 19. Morphological classifications as a function of simulated source surface brightness profile Sérsic index. Shown are the *Morpheus* classification distributions for simulated S/N = 20 objects with circular Sérsic (1968) profiles, as a function of the Sérsic index $\eta \in [1, 9]$. The experiment was repeated on objects with effective radii of three (upper left panel), five (upper right panel), seven (lower left panel), and nine (lower right panel) pixels. Objects with $\eta = 1$ were dominantly classified as *disk* or *spheroid*. Intermediate Sérsic profiles ($\eta \sim 2$–3) were mostly classified as *spheroid*. Objects with high Sérsic index ($\eta \geqslant 4$) were classified as *point source/compact*. These simulated objects with azimuthally symmetrical surface brightness profiles were assigned almost no *irregular* classifications by *Morpheus*.

The model was trained on the K15 sample, which has a reported completeness of $H = 24.5$AB, and given the pixel-by-pixel *background* classifications computed by *Morpheus*, it is unclear whether the object-level detection of sources in images would match the K15 completeness. In regions of low *background*, the transition to regions of high *background* likely depends on the individual pixel fluxes, but this transition should be characterized.

In what follows below, we provide some quantification of the *Morpheus* performance for identifying objects with different fluxes. To do this, we use results from the 3D-HST catalog of sources for the GOODS South (Skelton et al. 2014; Momcheva et al. 2016). Given the output *Morpheus background* classification images computed from the HLF GOODS South FITS images in $F606W$, $F850LP$, $F125W$, and $F160W$, we can report the pixel-by-pixel *background* values and typical *background* values aggregated for objects. These measurements can be compared directly with sources in the Momcheva et al. (2016) catalog to characterize how *Morpheus* detects objects and the corresponding completeness relative to 3D-HST.

In a first test, we can locate the Momcheva et al. (2016) catalog objects based on their reported coordinates in the *Morpheus background* image and then record the *background* pixel values at those locations. Figure 21 shows the two-dimensional histogram of the *Morpheus background* value and

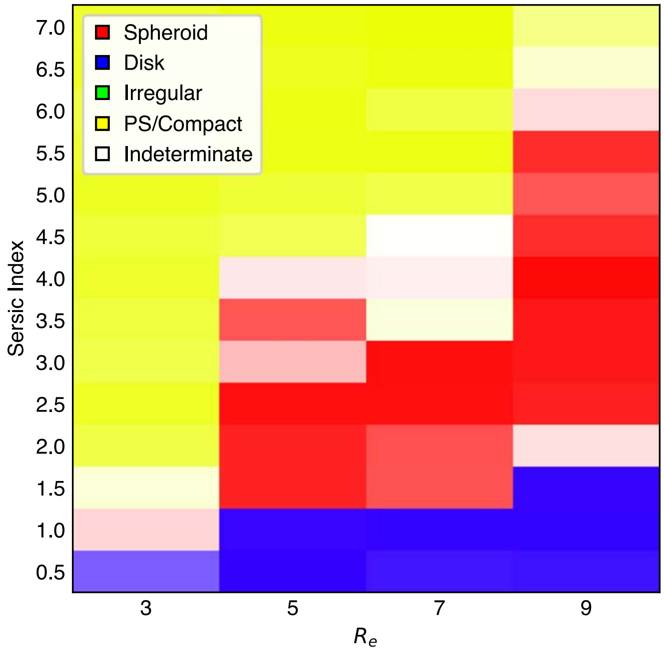

Figure 20. Dominant morphological classification as a function of simulated source surface brightness profile Sérsic index η and effective radius R_e in pixels. Each element of the matrix is color-coded to indicate the dominant *Morpheus* classification assigned for each $[\eta, R_e]$ pair, with the saturation of the color corresponding to the difference between the dominant and second *Morpheus* classification values. Large objects with low Sérsic index are classified as *disk* (blue). Large objects with high Sérsic index are classified as *spheroid* (red). Small objects with high Sérsic index are classified as *point source/compact* (yellow). None of the symmetrical objects in the test were classified as *irregular* (green).

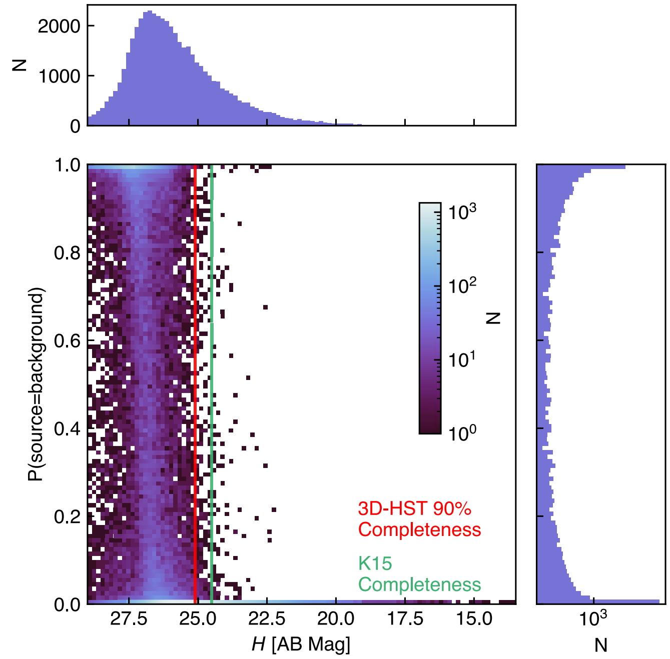

Figure 21. Two-dimensional histogram of *Morpheus background* values and 3D-HST source flux in GOODS South. Shown is the distribution of *background* at the location of 3D-HST sources (Skelton et al. 2014; Momcheva et al. 2016) in GOODS South of various *H*-band magnitudes, along with the marginal histograms for both quantities (side panels). For reference, the K15 completeness (green line) and 3D-HST 90% completeness (red line) flux limits are also shown. The 3D-HST sources most frequently have *background* = 0, and the majority of 3D-HST sources of any flux $H < 29$ have *background* < 0.5. The *background* values for objects where K15 and 3D-HST are complete is frequently zero. The *Morpheus background* values increase for many objects at flux levels $H > 26$AB.

3D-HST source *H*-band AB magnitude, along with the marginal distributions of both quantities. The figure also indicates the reported K15 sample and 3D-HST 90% completeness flux levels. The results demonstrate that for the majority of 3D-HST sources and for the vast majority of bright 3D-HST sources with $H < 25$, the local *Morpheus background* = 0. The low *background* values computed by *Morpheus* extend to extremely faint magnitudes (e.g., $H \approx 29$), indicating that for some faint sources, *Morpheus* reports *background* = 0 and that *background* is not a simple function of the local S/N of an object. For many objects with fluxes below the 3D-HST completeness, the *Morpheus background* value does increase with decreasing flux, and there is a rapid transition between detected sources at $H \approx 26.5$ to undetected sources at $H \leqslant 27.5$.

Owing to this transition in *background* with decreasing flux, the completeness of *Morpheus* relative to 3D-HST will depend on a threshold in *background* used to define a detection. Figure 22 shows the completeness of *Morpheus* in recovering 3D-HST objects as a function of *H*-band source flux for different *background* levels defining a *Morpheus* detection. The completeness flux limits for K15 and 3D-HST are indicated for reference. For magnitudes $H < 25$AB, where 3D-HST and K15 are complete, *Morpheus* is highly complete and recovers more than 99% of all 3D-HST sources. The *Morpheus* completeness declines rapidly at fluxes $H > 26.5$AB, where *Morpheus* is 90% relative to 3D-HST for *background* thresholds of $P \leqslant 0.5$. Perhaps remarkably, for all *background* thresholds $P \leqslant 0.01$–0.5, *Morpheus* detects some objects as faint as $H \approx 29$, about 100× fainter in flux than the training set objects.

We further examined the detection of 3D-HST sources as a function of color $(V - H)$ to evaluate bias that may have been inherited as result of the training data set. In our tests, we found that *Morpheus* is not biased with respect to color for those sources that are brighter than the K15 magnitude limit (Figure 23). When considering all sources within the 3D-HST catalog, *Morpheus* detects sources well, with a slight bias for bluer sources, but it performs less well for very red $((V - H) \geqslant 9)$ and $((V - H) < 0)$ sources. However, it should be noted that there are very few such sources in the training set, and with a more extensive training sample, *Morpheus* could be more complete.

7.5. Morphological Classification versus Source Magnitude

The tests of *Morpheus* on simulated Sérsic objects of different effective radii and the completeness study suggest that the ability of *Morpheus* to provide informative morphological information about astronomical sources will depend on the size and S/N of the object. While these are intuitive limitations on any morphological classification method, the distribution of morphological classifications with source flux determined by *Morpheus* should be quantified.

Figure 24 shows the fraction of 3D-HST objects detected and classified by *Morpheus* as *spheroid*, *disk*, *irregular*, and *point source/compact* as a function of their *H*-band magnitude. Most of the brightest objects in the image are nearby stars, classified as *point source/compact*. At intermediate

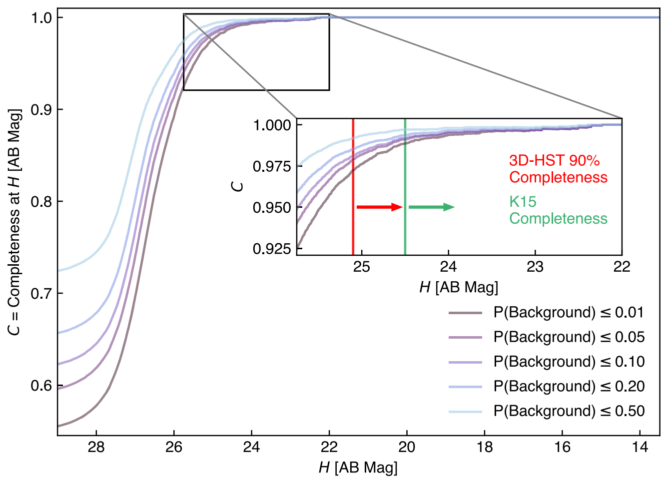

Figure 22. Completeness of *Morpheus* in source detection relative to 3D-HST (Skelton et al. 2014; Momcheva et al. 2016) in GOODS South. Shown is the fraction of 3D-HST sources in GOODS South detected by *Morpheus* brighter than some H-band source magnitude, for different *background* thresholds defining a detection (purple lines). The inset shows the *Morpheus* completeness for the brightest objects where 3D-HST (red line and arrow) and K15 (green line and arrow) are both highly complete. The completeness of *Morpheus* relative to 3D-HST is >90% where 3D-HST is highly complete. The completeness of *Morpheus* declines rapidly at faint magnitudes ($H \gtrsim 26.5$), but some objects are detected to $H \sim 29$, about 100× fainter than objects in the training set.

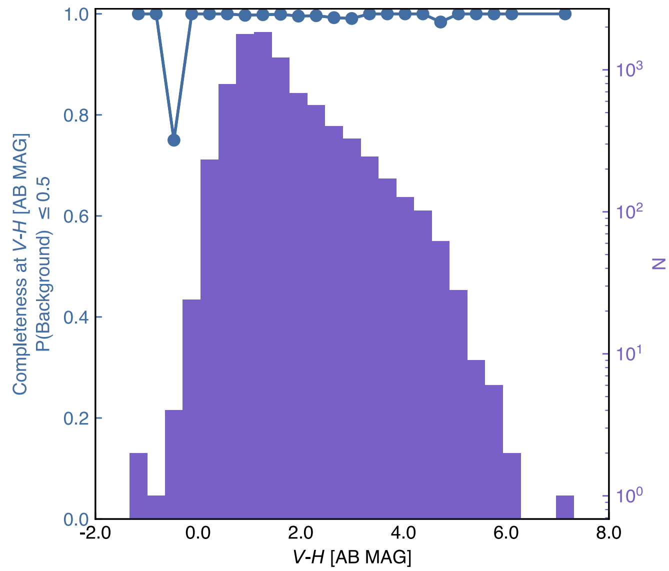

Figure 23. Source detection completeness as a function of color for sources with an H-band ($F160W$) AB magnitude of $H < 24.5$. Sources that had a V-band flux less than the V-band error had their flux replaced with three times the error value to limit unrealistically large $V - H$ values. *Morpheus* does not show bias in the detection of objects with respect to color. There is a dip in completeness at $V - H \sim 0.2$, where the completeness is ∼75%. However, this bin only has four sources, indicating that *Morpheus* only missed one source at this color.

magnitudes, *Morpheus* classifies the objects as primarily a mix of *disk* (∼50%) and *spheroid* (∼30%), with contributions from *irregular* (∼10%–30%) and *point source/compact* (∼5%–15%). For fainter objects, below the completeness limit of the K15 training sample, *Morpheus* increasingly classifies objects as *irregular* and *point source/compact*. This behavior is in part physical, in that many low-mass galaxies are irregular and distant galaxies are physically compact. In part, it reflects a natural bias in how the morphologies are defined during training. In K15, the class *point/source compact* can describe bright stars and compact unresolved sources (see Section 3.1 of K15). However, the trend also reflects how *Morpheus* becomes less effective at distinguishing morphologies in small, faint objects and returns either *point source/compact* and *irregular* for low S/N and compact sources. While training *Morpheus* on fainter objects with well-defined morphologies could enhance the ability of *Morpheus* to distinguish the features of faint sources, the results of this test make sense in the context of the completeness limit of the K15 training sample used.

7.6. False Positives in GOODS South

The segmentation and deblending of real astronomical data sets are challenging tasks. An important test of the efficacy of the *Morpheus* segmentation and deblending algorithms is to examine false positives generated when *Morpheus* is applied to a real image. To quantify the propensity for *Morpheus* to generate false positives, the segmentation and deblending algorithms were run on the HLF GOODS South image without the specified marker set parameter p (see Algorithms 1 and 2). For the purposes of this test, a false positive is then defined as a set of pixels classified by the *Morpheus* segmentation and deblending algorithms as a source, but one that does not contain a source from the 3DHST and CANDELS (Guo et al. 2013) catalogs. Additionally, since the edges of the GOODS

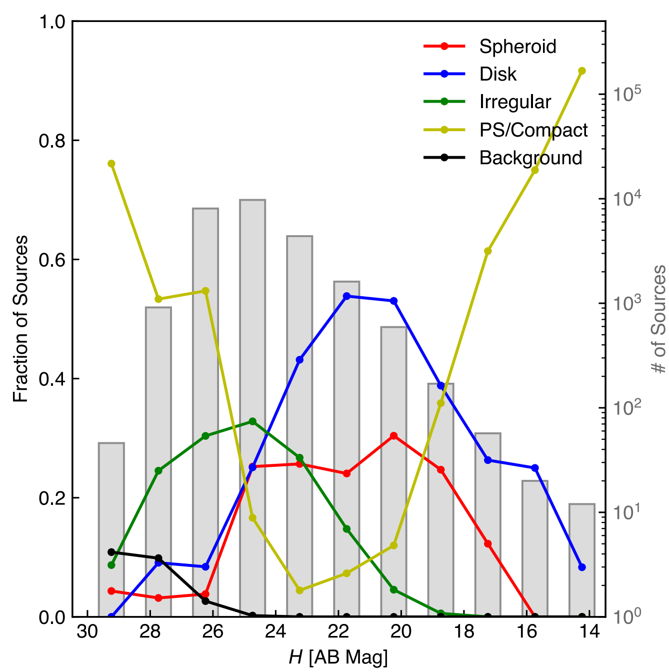

Figure 24. Morphological classification as a function of object flux in GOODS South. Shown are the fraction of 3D-HST objects (see left axis) with *Morpheus* dominant, flux-weighted classifications of *spheroid* (red line), *disk* (blue line), *irregular* (green line), and *point source/compact* (yellow line), each as a function of their H-band ($F160W$) AB magnitude. The brightest objects in the image are stars that are classified as *point source/compact*. The faintest objects in the image are compact faint galaxies classified as *point source/compact* or irregular. At intermediate fluxes, the objects are primarily classified as *disk* and *spheroid*. Also shown as a gray histogram (see right axis) is the number of 3D-HST objects detected and classified by *Morpheus* with source magnitude.

South classified image are a frayed mix of pixels, to minimize the effects of data artifacts, sources less than 20 pixels from the edge of the classified area were excluded from the analysis.

Table 4
Summary of Sources Identified by *Morpheus* in GOODS-S That Were Absent in the CANDELS or 3D-HST Catalogs

False Positives in GOODS South			
Category	Count	% of False Positives	% of All Sources
Image Artifact	27	21.95%	0.139%
Poor Deblend	31	25.20%	0.159%
Missed Source	47	38.21%	0.241%
Actual False Positive	18	14.64%	0.092%
Total	123	100%	0.631%

Note. Of the 19,481 sources identified by *Morpheus* in a subregion of GOODS-S, 123 sources did not have CANDELS or 3D-HST counterparts. Upon visual inspection, these objects could be categorized as either *image artifacts*, *poor deblends* where *Morpheus* had shredded sources, *missed sources* corresponding to real objects missed by CANDELS and 3D-HST, or *actual false positives* incorrectly identified by *Morpheus* as real sources. The false-positive rate for the *Morpheus* algorithm is only roughly 0.09%, defined relative to the CANDELS and 3D-HST catalogs. See Section 7.6 for more discussion.

Further, we conservatively use the "default" *Morpheus* algorithms that identify sources with $background = 0$, i.e., when *Morpheus* indicates a source detection with high confidence. With these choices, the sample used for the false-positive analysis was a total of 19,481 sources.

Among the objects classified by the segmentation and deblending algorithms, 123 sources were not present in the CANDELS or 3D-HST catalogs. Upon visual inspection of these sources, each can be categorized as an image artifact, a poor deblend, a missed source, or an actual false positive. We list the number of sources in each category in Table 4.

Sources in the *image artifiact* category are false positives caused by image artifacts. The *poor deblend* category represents false positives caused by the *Morpheus* deblending algorithm, where single sources in the CANDELS or 3D-HST catalogs were shredded into multiple *Morpheus* sources. The *missed sources* are *Morpheus* sources that upon visual inspection correspond to real objects missed by the 3D-HST or CANDELS catalogs. Sources in the *actual false-positive* category are false positives not associated with any image artifact or real source after visual inspection.

As Table 4 shows, *Morpheus* can identify real sources that other methods that are used to generate catalogs can miss; although, the algorithms used by *Morpheus* can very rarely cause actual false positives (at roughly the 0.1% rate). Given the delicate nature of deblending, this analysis suggests that the *Morpheus* deblending algorithm could be integrated with other methods to generate more robust segmentation maps.

8. Value-added Catalog for 3D-HST Sources with *Morpheus* Morphologies

The *Morpheus* framework provides a system for performing the pixel-level analysis of astronomical images and has been engineered to allow for the processing of large-format scientific FITS data. As described in Section 6.1, *Morpheus* was applied to the HLF (Illingworth et al. 2016) reduction of HST imaging in GOODS South[5] and a suite of morphological classification images produced. Using the *Morpheus background* in GOODS South, the detection efficiency of *Morpheus* relative to the Momcheva et al. (2016) 3D-HST catalog was computed (see Section 7.4) and a high level of completeness was demonstrated for objects comparably bright to the Kartaltepe et al. (2015) galaxy sample used to train the model. By segmenting and deblending the HLF images, *Morpheus* can then compute flux-weighted morphologies for all of the 3D-HST sources.

Table 5 provides the *Morpheus* morphological classifications for 50,506 sources from the 3D-HST catalog of Momcheva et al. (2016). This value-added catalog lists the 3D-HST ID, the source R.A. and decl., the $F160W$-band AB magnitude (or -1 for negative flux objects), and properties for the sources computed by *Morpheus*. The value-added properties include a flag denoting whether and how *Morpheus* detected the object, the area in pixels assigned to each source, and the *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* flux-weighted classifications determined by *Morpheus*. The size of the segmentation regions assigned to each 3D-HST object following Algorithms 1 and 2 is reported for all objects. If the segmentation region assigned to an object was smaller than a circle with a $0\farcs36$ radius, or the object was undetected, instead, we used a $0\farcs36$ radius aperture (about 109 pixels) to measure flux-weighted quantities. Only objects with joint coverage in the HLF V, z, J, and H FITS images are classified and receive an assigned pixel area. The full results for the *Morpheus* morphological classifications of 3D-HST objects are released as a machine-readable table. Appendix D describes the *Morpheus* Data Release associated with this paper, including FITS images of the classification images, the value-added catalog, and segmentation maps generated by *Morpheus* for the 3D-HST sources used to compute flux-weighted morphologies. Additionally, we release an interactive online map at https://morpheus-project.github.io/morpheus/, which provides an interface to examine the data and overlay the 3D-HST catalog on the *Morpheus* classification images, morphological color images, and segmentation maps.

9. Discussion

The analysis of astronomical imagery necessarily involves pixel-level information to be used to characterize sources. The semantic segmentation approach of *Morpheus* delivers pixel-level separation between sources and the background sky, and provides an automated classification of the source pixels. In this paper, we trained *Morpheus* with the visual morphological classifications from Kartaltepe et al. (2015). We then characterized the performance of *Morpheus* in reproducing the object-level classifications of K15 after aggregating the pixel information through flux-weighted averages of pixels in *Morpheus*-derived segmentation maps, and in detecting objects via completeness measured relative to the 3D-HST catalog (Momcheva et al. 2016). The potential applications of *Morpheus* extend well beyond object-level morphological classification. Below, we discuss some applications of the pixel-level information to understanding the complexities of galaxy morphology and future applications of the semantic segmentation approach of *Morpheus* in areas other than morphological classification. We also comment on some features of *Morpheus* specific to its application on astronomical images.

[5] Some bright pixels in the released HLF images are censored with zeros. For the purpose of computing the segmentation maps only, we replaced these censored pixels with nearby flux values.

Table 5
Morpheus + 3D-HST Value-added Catalog for GOODS South

ID	R.A. (deg)	Decl. (deg)	*H*160 (AB mag)	Detection Flag	Area (pixels)	*Spheroid*	*Disk*	*Irregular*	*PS/Compact*	*Background*	Min (*Background*)
1	53.093012	−27.954546	19.54	1	4408	0.092	0.797	0.106	0.003	0.003	0.000
2	53.089613	−27.959742	25.49	0	…	…	…	…	…	…	…
3	53.102913	−27.959642	25.37	1	121	0.013	0.033	0.894	0.025	0.034	0.000
4	53.101709	−27.958481	21.41	1	725	0.001	0.874	0.120	0.004	0.001	0.000
5	53.102277	−27.958683	24.62	1	144	0.098	0.003	0.020	0.746	0.133	0.000
6	53.090577	−27.958515	25.07	2	109	0.000	0.831	0.034	0.000	0.134	0.001
7	53.099964	−27.958278	23.73	1	266	0.000	0.712	0.284	0.000	0.003	0.000
8	53.096144	−27.957583	21.41	1	1322	0.001	0.752	0.238	0.003	0.006	0.000
9	53.091572	−27.958367	25.90	2	109	0.000	0.044	0.083	0.081	0.792	0.431
10	53.091852	−27.958181	25.88	2	109	0.000	0.000	0.038	0.186	0.776	0.570

Note. Column 1 provides the 3D-HST source ID. Columns 2 and 3 list the R.A. and decl. in degrees. Column 4 shows the $F160W$ AB magnitude of the 3D-HST source, with −1 indicating a negative flux reported by 3D-HST. Column 5 lists the detection flag, with 0 indicating that the object was not within the region of GOODS South classified by *Morpheus*, 1 indicating a detection with *background* = 0 at the source location, 2 indicating a possible detection with 0 < *background* < 1 at the source location, and 3 indicating a non-detection with *background* = 1 at the source location. Column 6 reports the area in pixels for the object determined by the *Morpheus* segmentation algorithm. For non-detections and objects with very small segmentation regions, we instead use a $0\farcs36$ radius circle (about 109 pixels) for their segmentation region. Columns 7–11 list the flux-weighted *Morpheus* morphological classifications of the objects within their assigned area. These columns are normalized such that the classifications sum to one for objects where the detection flag = 2. Column 12 reports the minimum *background* value within the segmentation region.

(This table is available in its entirety in machine-readable form.)

9.1. Pixel-level Morphology

The complex morphologies of astronomical objects have been described by both visual classification schemes and quantitative morphological measures for many years. Both Hubble (1926) and Vaucouleurs (1959) sought to subdivide broad morphological classifications into more descriptive categories. Quantitative morphological decompositions of galaxies (e.g., Peng et al. 2010) also characterize the relative strength of bulge and disk components in galaxies, and quantitative morphological classifications often measure the degree of object asymmetry (e.g., Abraham et al. 1994; Conselice et al. 2000; Lotz et al. 2004).

The object-level classifications computed by *Morpheus* provide a mixture of the pixel-level morphologies from the *Morpheus* classification images. The classification distributions reported in the *Morpheus* value-added catalog in GOODS South provide many examples of flux-weighted measures of morphological type. However, more information is available in the pixel-level classifications than the flux-weighted summaries provide.

Figure 25 shows an example object for which the *Morpheus* pixel-level classifications provide direct information about its complex morphology. The figure shows a disk galaxy with a prominent central bulge. The pixel-level classifications capture both the central bulge and the extended disk, with the pixels in each structural component receiving dominant bulge or disk classifications from *Morpheus*. Note that *Morpheus* was not trained to perform this automated bulge–disk decomposition, as in the training process, all pixels in a given object are assigned the same distribution of classifications as determined by the K15 visual classifiers. As the use of pixel-level morphological classifications becomes widespread, the development of standard data sets that include labels at the pixel-level will be needed to evaluate the efficacy of classifiers. Simulations of galaxy formation may be useful for generating such training data sets (e.g., Huertas-Company et al. 2019). We leave a more thorough analysis of automated morphological decompositions with *Morpheus* to future work.

9.2. Morphological Deblending

The ability of *Morpheus* to provide pixel-level morphological classifications has applications beyond the bulk categorization of objects. One potential additional application is the morphological deblending of overlapping objects, where the pixel-level classifications are used to augment the deblending process. Figure 26 shows an example of two blended objects, 3D-HST IDs 543 and 601, where the *Morpheus* pixel-level classifications could be used to perform or augment star-galaxy separation. As the figure makes clear, when *Morpheus* correctly assigns dominant classifications to pixels, there exists an interface region between regions with distinctive morphologies (in this case, *spheroid* and *point source/compact*) that could serve as an interface between segmented regions in the image. The deblending algorithm used in this work could include other forms of machine-learning (e.g., Masters et al. 2015; Hemmati et al. 2019) information in the deblending process. If *Morpheus* was trained on information other than morphology, such as photometric redshift, those pixel-level classifications could be used in the deblending process as well. We plan to explore this idea in future applications of *Morpheus*.

9.3. Classifications Beyond Morphology

The semantic segmentation approach of *Morpheus* allows for complex features of astronomical objects to be learned from the data, as long as those features can be spatially localized by other means. In this paper, we used the segmentation maps of K15 to separate source pixels from the sky, and then we assigned pixels within the segmentation maps the morphological classification determined by K15 on an object-by-object basis. In principle, this approach can be extended to identify regions of pixels that contain a wide variety of features. For instance, *Morpheus* could be trained to identify image artifacts, spurious cosmic-rays, or other instrumental or data effects that lead to distinctive pixel-level features in images. Of course, real features in images could also be identified, such as the pixels containing arcs in gravitational lenses, or perhaps low-surface

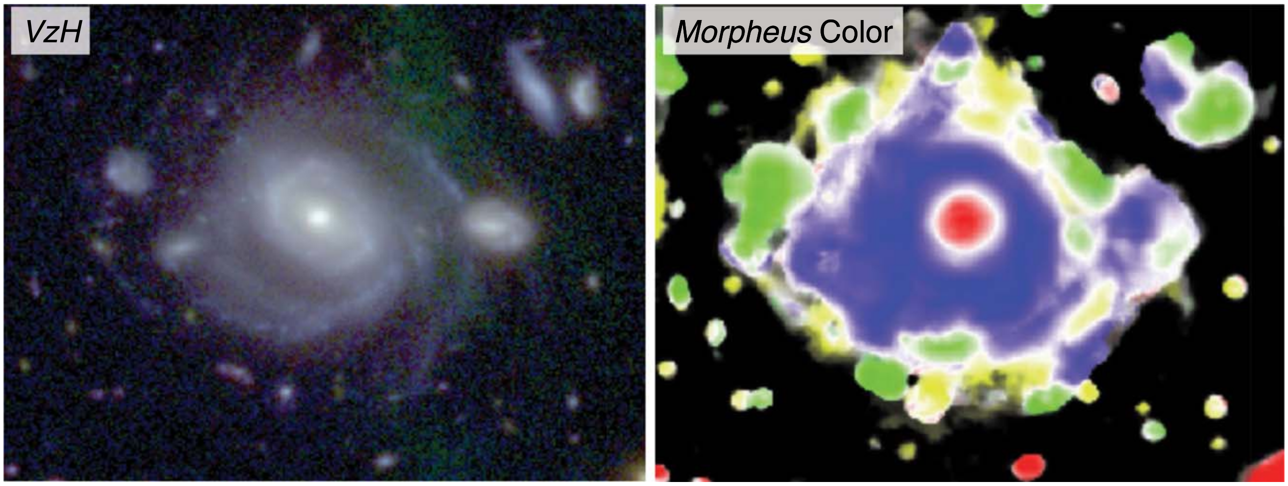

Figure 25. Example automated morphological decomposition by *Morpheus*. The left panel shows the *VzH* multicolor image of a galaxy in GOODS South from the HLF. The disk galaxy, 3D-HST ID 46386, has a prominent central bulge. The right panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The figure demonstrates that *Morpheus* correctly classifies the spheroid and disk structural components of the galaxy, even though the training process for *Morpheus* does not involve spatially varying morphologies for galaxy interiors. We note that there is a large-scale image artifact in *F*850*LP* that appears green in the left image but does not strongly affect the *Morpheus* pixel-level classifications.

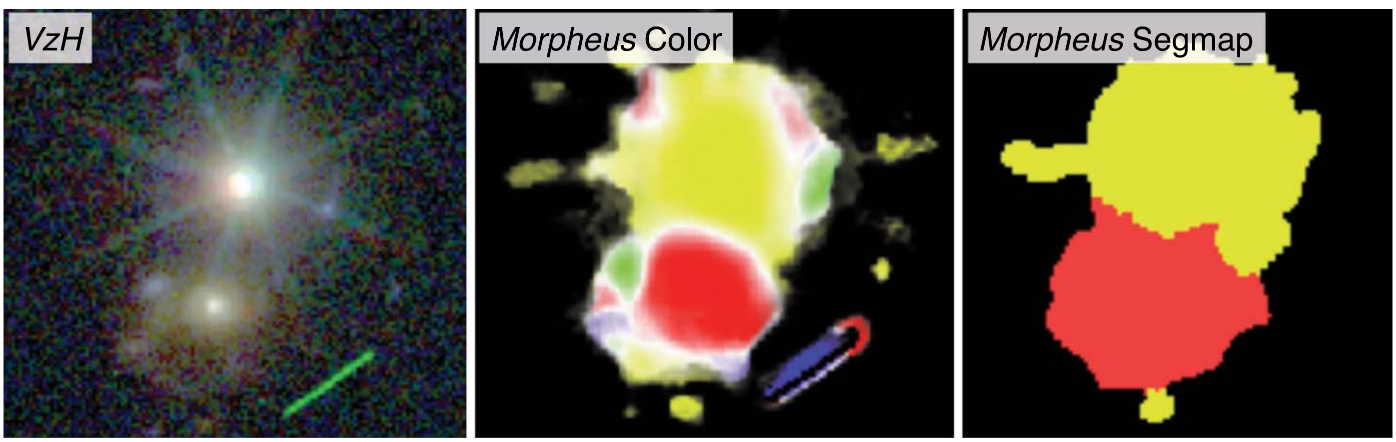

Figure 26. Example of morphological deblending by *Morpheus*. The leftmost panel shows the *VzH* image of a star-galaxy blend in GOODS South from the HLF. The star, 3D-HST ID 601, overlaps with a spheroidal galaxy 3D-HST ID 543. The center panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The pixel regions dominated by the star or spheroid are correctly classified by *Morpheus*. The right panel shows the resulting *Morpheus* segmentation map, illustrating that the dominant object classification in each segmentation region is also correct. The pixel-level classifications could be used to refine the segmentation to more precisely include only pixels that contained a single dominant class. The green feature in the left panel is an image artifact in *F*850*LP*.

brightness features in interacting systems and stellar halos. These pixel-level applications of *Morpheus* complement machine-learning-based methods already deployed, such as those that discover and model gravitational lenses (Agnello et al. 2015; Hezaveh et al. 2017; Morningstar et al. 2018, 2019). Pixel-level photometric redshift estimates could also be adopted by *Morpheus* and compared with existing methods based on SED fitting or other forms of machine learning (e.g., Masters et al. 2015; Hemmati et al. 2019).

9.4. Deep Learning and Astronomical Imagery

An important difference in the approach of *Morpheus*, where a purpose-built framework was constructed from TensorFlow primitives, compared with the adaptation and retraining of existing frameworks like Inception (e.g., Szegedy et al. 2016) is the use of astronomical FITS images as training, test, and input data rather than preprocessed PNG or JPG files. The incorporation of deep learning into astronomical pipelines will benefit from the consistency of the data format. The output data of *Morpheus* are also FITS classification images, allowing pixel-by-pixel information to be easily referenced between the astronomical science images and the *Morpheus* model images. As indicated in Section 2.2, the *Morpheus* framework is extensible and allows for any number of astronomical filter images to be used, as opposed to a fixed RGB set of layers in PNG or JPG files. The *Morpheus* framework has been engineered to allow for the classification of arbitrarily sized astronomical images. The same approach also provides *Morpheus* with a measure of the dispersion of the classifications of individual pixels, allowing the user to choose a metric for the “best” pixel-by-pixel classification. The combination of these features allows for immense flexibility in adapting the *Morpheus* framework to problems in astronomical image classification.

10. Summary and Conclusions

In this paper, we presented *Morpheus*, a deep learning framework for the pixel-level analysis of astronomical images. The architecture of *Morpheus* consists of our original implementation of a U-Net (Ronneberger et al. 2015) convolutional neural network. *Morpheus* applies the semantic segmentation technique adopted from computer vision to enable pixel-by-pixel classifications, and by separately identifying background and source pixels, *Morpheus* combines object detection and classification into a single analysis. *Morpheus* represents a new approach to astronomical data analysis, with wide applicability in enabling per-pixel classification of images where suitable training data sets exist. Important results from this paper include:

1. *Morpheus* provides pixel-level classifications of astronomical FITS images. By using user-supplied segmentation maps during training, the model learns to distinguish *background* pixels from pixels containing source flux. The pixels associated with astronomical objects are then classified according to the classification scheme of the training data set. The entire *Morpheus* source code has been publicly released, and a Python package installer for *Morpheus* provided. Further, we have a citable "frozen" version of code available through Zenodo (Hausen 2020).
2. As a salient application, we trained *Morpheus* to provide pixel-level classifications of galaxy morphology by using the Kartaltepe et al. (2015) visual morphological classifications of galaxies in the CANDELS data set (Grogin et al. 2011; Koekemoer et al. 2011) as our training sample.
3. Applying *Morpheus* to the HLF (Illingworth et al. 2016) v2.0 reduction of the CANDELS data in GOODS South and the v1.0 data (Grogin et al. 2011; Koekemoer et al. 2011) for COSMOS, EGS, GOODS North, and UDS, we generated morphological classifications for every pixel in the HLF mosaics. The resulting *Morpheus* morphological classification images have been publicly released.
4. The pixel-level morphological classifications in GOODS South were then used to compute and publicly release a value-added catalog of morphologies for all objects in the public 3D-HST source catalog (Skelton et al. 2014; Momcheva et al. 2016).
5. The CANDELS HLF and 3D-HST data were used to quantify the performance of *Morpheus*, both for morphological classification and its completeness in object detection. As trained, the *Morpheus* code shows high completeness at magnitudes $H \lesssim 26.5$AB. We demonstrate that *Morpheus* can detect objects in astronomical images at flux levels up to $100\times$ fainter than the completeness limit of its training sample ($H \sim 29$AB).
6. Tutorials for using the *Morpheus* deep learning framework have been created and publicly released as Jupyter notebooks.
7. An interactive visualization of the *Morpheus* model results for GOODS South, including the *Morpheus* segmentation maps and pixel-level morphological classifications of 3D-HST sources, has been publicly released.

We expect that semantic segmentation will be increasingly used in astronomical applications of deep learning, and *Morpheus* serves as an example framework that leverages this technique to identify and classify objects in astronomical images. We caution that *Morpheus* may be most effective at wavelengths similar to the data on which the model was trained (i.e., the $F606W$, $F850LP$, $F125W$, and $F160W$ bands). However, Domínguez Sánchez et al. (2019) have shown recent success in applying transfer learning on astronomical data sets with morphological labels. With the advent of large imaging data sets such as those provided by the Dark Energy Survey (Dark Energy Survey Collaboration et al. 2016) and Hyper Suprime-Cam (Aihara et al. 2018a, 2018b), and next-generation surveys to be conducted by Large Synoptic Survey Telescope (Ivezić et al. 2019; Robertson et al. 2019), Euclid (Laureijs et al. 2011; Rhodes et al. 2017), and the Wide Field Infrared Survey Telescope (Akeson et al. 2019), pixel-level analysis of massive imaging data sets with deep learning will find many applications. While the details of the *Morpheus* neural network architecture will likely change and possibly improve, we expect the approach of using semantic segmentation to provide pixel-level analyses of astronomical images with deep learning models will be broadly useful. The public release of the *Morpheus* code, tutorials, and example data products should provide a basis for future applications of deep learning for astronomical data sets.

B.E.R. acknowledges a Maureen and John Hendricks Visiting Professorship at the Institute for Advanced Study, and NASA contract NNG16PJ25C. R.D.H. acknowledges the Eugene V. Cota-Robles Fellowship. The authors acknowledge use of the lux supercomputer at UC Santa Cruz, funded by NSF MRI grant AST 1828315. This work is based on observations taken by the CANDELS Multi-Cycle Treasury Program with the NASA/ESA HST, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5-26555. This work is based on observations taken by the 3D-HST Treasury Program (GO 12177 and 12328) with the NASA/ESA HST, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5-26555.

Software: NumPy (van der Walt et al. 2011), astropy (Price-Whelan et al. 2018), scikit-learn (Pedregosa et al. 2011), matplotlib (Hunter 2007), TensorFlow (Abadi et al. 2016), Tiny Tim (Krist et al. 2011).

Appendix A
Deep Learning

The *Morpheus* deep-learning framework incorporates a variety of technologies developed for machine-learning applications. The following descriptions of deep-learning techniques complement the overview of *Morpheus* provided in Section 2 and are useful for understanding optional configurations of the model.

A.1. Artificial Neuron

The basic unit of the *Morpheus* neural network is the artificial neuron (AN), which transforms an input vector $\boldsymbol{x}$ to a single output AN($\boldsymbol{x}$). The AN is designed to mimic the activation of a neuron, producing a nonlinear response to an input stimulus value when it exceeds a rough threshold.

The first stage of an AN consists of a function

$$z(\boldsymbol{x}) = \sum_{i=1}^{n} w_i x_i + b \tag{A1}$$

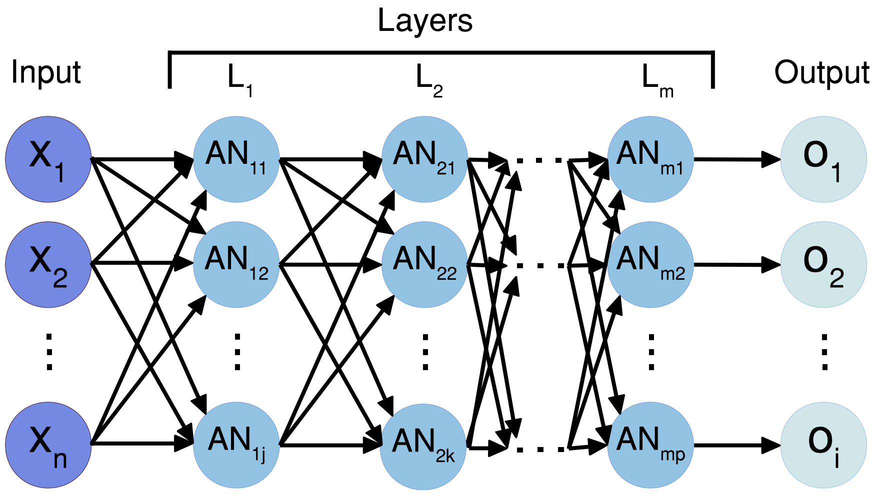

Figure A1. Schematic of a simple neural network. Given an input vector $\boldsymbol{x}$, the neural network applies a series of reductions and nonlinear transformations through a collection of layers $\boldsymbol{L}$ to produce an output $\boldsymbol{o}$. Each layer L consists of a set of artificial neurons AN that perform a linear rescaling of their input data, followed by a nonlinear transformation via the application of an activation function (see Equation (A2)). The activation function may vary across layers.

that adds the dot product of the n-element vector $\boldsymbol{x}$ with a vector of weights $\boldsymbol{w}$ to a bias b. The values of the $\boldsymbol{w}$ elements and b are parameters of the model that are set during optimization. The function $z(\boldsymbol{x})$ is equivalent to a linear transformation on input data $\boldsymbol{x}$.

In the second stage, a nonlinear function a is applied to the output of $z(\boldsymbol{x})$. We write

$$AN(\boldsymbol{x}) \equiv a(z(\boldsymbol{x})), \tag{A2}$$

where $a(z)$ is called the activation function. The *Morpheus* framework allows the user to specify the activation function, including the sigmoid

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}, \tag{A3}$$

the hyperbolic tangent

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}, \tag{A4}$$

and the rectified linear unit

$$\text{relu}(z) = \max(0, z). \tag{A5}$$

These functions share a thresholding behavior, such that the function activates a nonlinear behavior at a characteristic value of z, but the domains of these functions differ. For the morphological classification problem presented in this paper, the rectified linear unit (Equation (A5)) was used as the activation function.

A.2. Neural Networks

Increasingly complex computational structures can be constructed from ANs. Single ANs are combined into layers, which are collections of distinct ANs that process the same input vector x. A collection of layers forms a neural network (NN), with the layers ordered such that the outputs from one layer provide the inputs to the neurons in the subsequent layer. Figure A1 shows a schematic of an NN and how the initial input vector x is processed by multiple layers. As shown, these layers are commonly called "fully connected" since each neuron in a given layer receives the outputs z from all neurons in the previous layer.

A.3. Convolutional Neural Networks

The *Morpheus* framework operates on image data with a convolutional neural network (CNN). A CNN includes at least one layer of ANs whose z function uses a discrete cross-correlation (convolution) in place of the dot product in Equation (A1). For a convolutional artificial neuron (CAN), we write

$$z(\boldsymbol{X}) = (\boldsymbol{X} * \boldsymbol{W}) + b\boldsymbol{J}, \tag{A6}$$

where $\boldsymbol{X} * \boldsymbol{W}$ represents the convolution of an input image $\boldsymbol{X}$ and a kernel $\boldsymbol{W}$. The elements of the kernel $\boldsymbol{W}$ are parameters of the model, and $\boldsymbol{W}$ may differ in dimensions from $\boldsymbol{X}$. In *Morpheus*, the dimensions of $\boldsymbol{W}$ are set to be 3×3 throughout. The bias b is a scalar as before, and $\boldsymbol{J}$ represents a matrix of 1s with the same dimensions as the result of the convolution. In *Morpheus*, the convolution is zero-padded to maintain the dimensions of the input data.

The activation function of the neuron is computed element-wise after the convolution and bias have been applied to the input. We write

$$\text{CAN}(\boldsymbol{X}) \equiv a(z(\boldsymbol{X})). \tag{A7}$$

We refer to the output from a CAN as a feature map.

As with fully connected layers, convolutional layers consist of a group of CANs that process the same input data $\boldsymbol{X}$. Convolutional layers can also be arranged sequentially such that the output from one convolutional layer serves as input to the next. *Morpheus*' neural network architecture, being U-Net based, is comprised of CANs (see Figure A2 for a schematic). In typical convolutional neural network topologies, CANs are used to extract features from input images. The resulting feature maps are eventually flattened into a single vector and processed by a fully connected layer to produce the output classification values.

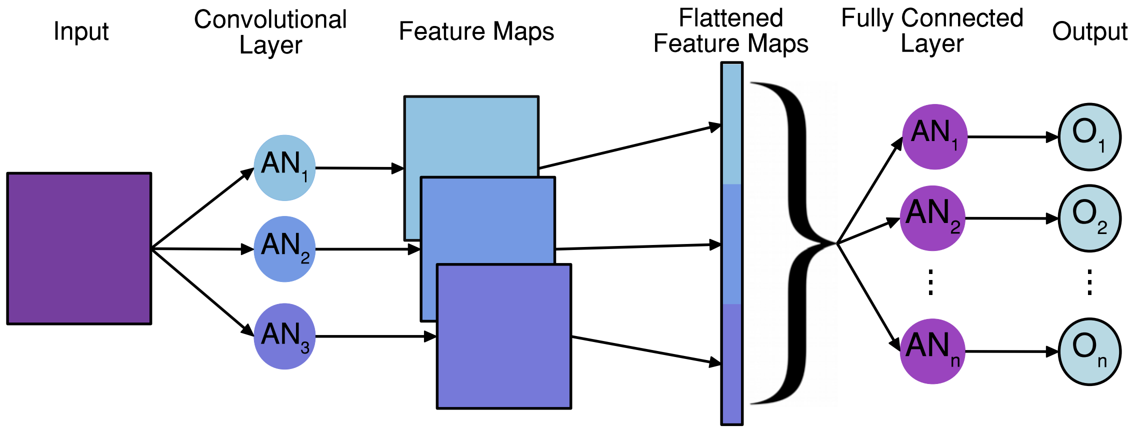

Figure A2. Schematic of a convolutional neural network (CNN). Shown is a simplified CNN consisting of a convolutional layer feeding a fully connected layer. Each artificial neuron (AN) in the convolutional layer outputs a feature map as described by Equation (A7). Each output feature map is flattened and concatenated into a single vector. This vector is processed by each AN in the fully connected layer (see Equation (A2)). The curly brace represents connections from all elements of the vector input.

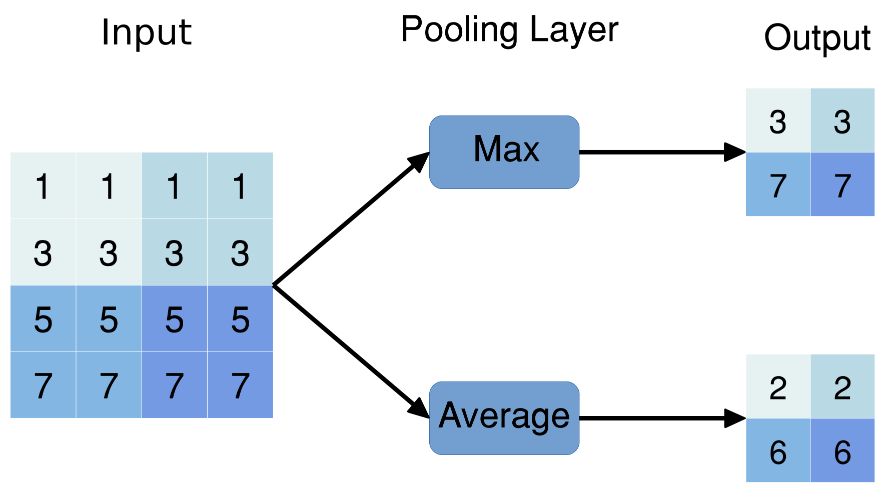

Figure A3. Comparison of max and average pooling layers. Pooling layers perform reductions on subsets of feature maps, providing a local average or maximum of data elements in a window (2×2 in this schematic). Shown are cells of an input feature map (left), color-coded within a window to match the corresponding regions of the output feature map (right). The pooling layers perform a simple reduction with these windows, taking either a maximum (upper branch) or average (lower branch).

A.4. Other Functions in Neural Networks

The primary computational elements of *Morpheus* are a convolutional neural network (Appendix A.3) and a fully connected layer (Appendix A.2). In detail, other layers are used to reformat or summarize the data, renormalize it, or combine data from different stages in the network.

A.4.1. Pooling

Pooling layers (Figure A3) are composed of functions that summarize their input data to reduce its size while preserving some information. These layers perform a moving average (average pooling) or maximum (max pooling) over a window of data elements, repeating these reductions as the window scans through the input image with a stride equal to the window size. In the morphological classification tasks described in this paper, *Morpheus* uses 2×2 windows and max pooling.

A.4.2. Up-sampling

Up-sampling layers expand the size of feature maps by a specified factor through an interpolation between input data elements. The up-sampling layers operate in the image dimensions of the feature map and typically employ bicubic and bilinear interpolation. In the morphological classification application explored in this paper, *Morpheus* used 2×2 up-sampling and bicubic interpolation.

A.4.3. Concatenation

Concatenation layers combine multiple feature maps by appending them without changing their contents. For instance, the concatenation of RGB channels into a three-color image would append three $N \times M$ images into an RGB image with dimensions $N \times M \times 3$. This operation is used in *Morpheus* to combine together data from the contraction phase with the output from bicubic interpolations in the expansion phase (see Figure 2).

A.4.4. Batch Normalization

A common preprocessing step for neural network architectures is to normalize the input data x using, e.g., the operation

$$\hat{x} = (x - \mu)/\sqrt{\sigma^2} \tag{A8}$$

where $\hat{x}$ is the normalized data, and μ and σ are parameters of the model. Ioffe & Szegedy (2015) extended this normalization step to apply to the inputs of layers within the network, such that activations (AN) and feature maps (CAN) are normalized over each batch. A batch consists of a subset of the training examples used during the training process. Simple normalization operations like Equation (A8) can reduce the range of values represented in the data provided to a layer, which can inhibit learning. Ioffe & Szegedy (2015) addressed this issue by providing an alternative normalization operation that introduces additional parameters to be learned during training. The input data elements x_i are first rescaled as

$$\hat{x}_i = \frac{x_i - \mu_{\boldsymbol{x}}}{\sqrt{\sigma_{\boldsymbol{x}}^2 + \epsilon}}. \tag{A9}$$

Here, x_i is a single element from the data output by a single AN or CAN over a batch, $\mu_{\boldsymbol{x}}$ is their mean, and $\sigma_{\boldsymbol{x}}^2$ is their variance. The parameter ϵ is learned during optimization. The new normalization $BN_{\hat{x}_i}$ is then taken to be a linear transformation

$$BN_{\hat{x}_i} = \gamma_x \hat{x}_i + \beta_x. \tag{A10}$$

The parameters γ_x and β_x are also learned during optimization. Ioffe & Szegedy (2015) demonstrated that batch normalization, in the form of Equation (A10), can increase overall accuracy and decrease training time, and we adopt this approach in the *Morpheus* framework.

A.5. U-Net Architecture

The *Morpheus* framework uses a U-Net architecture, first introduced by Ronneberger et al. (2015). The U-Net architecture was originally designed for segmentation of medical imagery but has enjoyed success in other fields. The U-Net takes as input a set of images and outputs a classification image of pixel-level probability distributions. The architecture begins with a contraction phase composed of a series of convolutional and pooling layers, followed by an expansion phase composed of a series of convolutional and up-sampling layers. Each of the outputs from the down-sampling layers is concatenated with the output of an up-sampling layer when the height and width dimensions of the feature maps match. These concatenations help preserve the locality of learned features in the output of the NN.

Appendix B
Code Release

The code for *Morpheus* has been release via GitHub (https://github.com/morpheus-project/morpheus). *Morpheus* is also available as a python package installable via pip (https://pypi.org/project/morpheus-astro/) and as Docker images available via Docker Hub (https://hub.docker.com/r/morpheusastro/morpheus). *Morpheus* includes both a Python API and a command-line interface, the documentation of which can be found online at https://morpheus-astro.readthedocs.io/en/latest/.

Appendix C
Code Tutorial

An online tutorial demonstrating the *Morpheus* Python API in the form of a Jupyter notebook can be found at https://github.com/morpheus-project/morpheus/blob/master/examples/example_array.ipynb. The tutorial walks through the classification of an example image. Additionally, the tutorial explores other features of *Morpheus*, including generating segmentation maps and morphological catalogs.

Appendix D
Data Release

The data release associated with this work consists of multiple data products. For each field in the CANDELS survey, we provide the following data products: pixel-level morphological classifications, segmentation maps, and value-added catalogs (see also Section 8) for the 3D-HST catalogs. Tables D1–D5 provide the URLs for each of the data products; these data products are also archived on Zenodo [doi:10.5281/zenodo.3746665]. Each of the fields has two types of segmentation maps, a segmentation map informed by the 3D-HST survey and a segmentation map informed only by the background values provided by *Morpheus* (see Algorithm 1). The classifications for the EGS and UDS fields may vary as a result of using the $F814W$ band in place of the $F850LP$ due to availability. Further, Figures D1–D4 show color morphological images (see Section 6.1.6) for the COMOS, EGS, GOODS North, and UDS fields.

An interactive online visualization of the HST images, *Morpheus* classification images, and 3D-HST sources is available at https://morpheus-project.github.io/morpheus/.

Table D1
Data Release Files Generated by Morpheus and Associated URLs for the COSMOS CANDELS Field

Morpheus Data Products for the COSMOS Field	
File Name	URL
Pixel-level Morphological Classifications	
morpheus_COSMOS_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/spheroid.html
morpheus_COSMOS_disk.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/disk.html
morpheus_COSMOS_irregular.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/irregular.html
morpheus_COSMOS_ps_compact.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/ps_compact.html
morpheus_COSMOS_background.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/background.html
morpheus_COSMOS_mask.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/mask.html
Segmentation Maps	
morpheus_COSMOS_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/segmap.html
morpheus_COSMOS_3dhst-segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/cosmos/3dhst-segmap.html
3D-HST Value Added Catalog	

Table D1
(Continued)

Morpheus Data Products for the COSMOS Field	
File Name	URL
morpheus_COSOMS_3dhst_catalog.v1.0.csv	morpheus-project.github.io/morpheus/data-release/cosmos/value-added-catalog.html
morpheus_COSOMS_3dhst_catalog.v1.0.txt	morpheus-project.github.io/morpheus/data-release/cosmos/value-added-catalog-mrt.html
All Files	
morpheus_COSMOS_all.v1.0.tar.gz	morpheus-project.github.io/morpheus/data-release/cosmos/all.html

Note. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_COSMOS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image we are classified by *Morpheus*. The *segmentation map* files are named according to the following scheme `morpheus_COSMOS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_COSMOS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 5. Additionally, a link to an archive containing all of the files associated with the COSMOS field is available in an additional section called *All Files*. See Appendix D for details.

Table D2
Data Release Files Generated by Morpheus and Associated URLs for the EGS CANDELS Field

Morpheus Data Products for the EGS Field	
File Name	URL
Pixel-level Morphological Classifications	
morpheus_EGS_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/spheroid.html
morpheus_EGS_disk.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/disk.html
morpheus_EGS_irregular.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/irregular.html
morpheus_EGS_ps_compact.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/ps_compact.html
morpheus_EGS_background.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/background.html
morpheus_EGS_mask.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/mask.html
Segmentation Maps	
morpheus_EGS_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/segmap.html
morpheus_EGS_3dhst-segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/egs/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_EGS_3dhst_catalog.v1.0.csv	morpheus-project.github.io/morpheus/data-release/egs/value-added-catalog.html
morpheus_EGS_3dhst_catalog.v1.0.txt	morpheus-project.github.io/morpheus/data-release/egs/value-added-catalog-mrt.html
All Files	
morpheus_EGS_all.v1.0.tar.gz	morpheus-project.github.io/morpheus/data-release/egs/all.html

Note. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_EGS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image we are classified by *Morpheus*. The *segmentation map* files are named according to the following scheme `morpheus_EGS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_EGS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 5. Additionally, a link to an archive containing all of the files associated with the EGS field is available in an additional section called *All Files*. See Appendix D for details.

Table D3
Data Release Files Generated by Morpheus and Associated URLs for the GOODS North CANDELS Field

Morpheus Data Products for the GOODS North Field	
File Name	URL
Pixel-level Morphological Classifications	
morpheus_GOODS-N_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/spheroid.html
morpheus_GOODS-N_disk.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/disk.html
morpheus_GOODS-N_irregular.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/irregular.html
morpheus_GOODS-N_ps_compact.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/ps_compact.html
morpheus_GOODS-N_background.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/background.html
morpheus_GOODS-N_mask.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/mask.html
Segmentation Maps	
morpheus_GOODS-N_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/segmap.html

Table D3
(Continued)

Morpheus Data Products for the GOODS North Field	
File Name	URL
morpheus_GOODS-N_3dhst-segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-n/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_GOODS-N_3dhst_catalog.v1.0.csv	morpheus-project.github.io/morpheus/data-release/goods-n/value-added-catalog.html
morpheus_GOODS-N_3dhst_catalog.v1.0.txt	morpheus-project.github.io/morpheus/data-release/goods-n/value-added-catalog-mrt.html
All Files	
morpheus_GOODS-N_all.v1.0.tar.gz	morpheus-project.github.io/morpheus/data-release/goods-n/all.html

Note. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-N_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image we are classified by *Morpheus*. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-N_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-N_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 5. Additionally, a link to an archive containing all of the files associated with the GOODS North field is available in an additional section called *All Files*. See Appendix D for details.

Table D4
Data Release Files Generated by Morpheus and Associated URLs for the GOODS South CANDELS Field

Morpheus Data Products for the GOODS South Field	
File Name	URL
Pixel-level Morphological Classifications	
morpheus_GOODS-S_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/spheroid.html
morpheus_GOODS-S_disk.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/disk.html
morpheus_GOODS-S_irregular.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/irregular.html
morpheus_GOODS-S_ps_compact.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/ps_compact.html
morpheus_GOODS-S_background.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/background.html
morpheus_GOODS-S_mask.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/mask.html
morpheus_GOODS-S_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/spheroid.html
Segmentation Maps	
morpheus_GOODS-S_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/segmap.html
morpheus_GOODS-S_3dhst_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/goods-s/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_GOODS-S_3dhst_catalog.v1.0.csv	morpheus-project.github.io/morpheus/data-release/goods-s/value-added-catalog.html
morpheus_GOODS-S_3dhst_catalog.v1.0.txt	morpheus-project.github.io/morpheus/data-release/goods-s/value-added-catalog-mrt.html
All Files	
morpheus_GOODS-S_all.v1.0.tar.gz	morpheus-project.github.io/morpheus/data-release/goods-s/all.html

Note. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-S_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image we are classified by *Morpheus*. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-S_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-S_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 5. Additionally, a link to an archive containing all of the files associated with the GOODS South field is available in an additional section called *All Files*. See Appendix D for details.

Table D5
Data Release Files Generated by Morpheus and Associated URLs for the UDS CANDELS Field

Morpheus Data Products for the UDS Field	
File Name	URL
Pixel-level Morphological Classifications	
morpheus_UDS_spheroid.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/spheroid.html
morpheus_UDS_disk.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/disk.html
morpheus_UDS_irregular.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/irregular.html
morpheus_UDS_ps_compact.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/ps_compact.html
morpheus_UDS_background.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/background.html
morpheus_UDS_mask.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/mask.html
Segmentation Maps	
morpheus_UDS_segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/segmap.html
morpheus_UDS_3dhst-segmap.v1.0.fits	morpheus-project.github.io/morpheus/data-release/uds/3dhst-segmap.html

Table D5
(Continued)

Morpheus Data Products for the UDS Field	
File Name	URL
3D-HST Value Added Catalogs	
morpheus_UDS_3dhst_catalog.v1.0.csv	morpheus-project.github.io/morpheus/data-release/uds/value-added-catalog.html
morpheus_UDS_3dhst_catalog.v1.0.txt	morpheus-project.github.io/morpheus/data-release/uds/value-added-catalog-mrt.html
All Files	
morpheus_UDS_all.v1.0.tar.gz	morpheus-project.github.io/morpheus/data-release/uds/all.html

Note. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_UDS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image we are classified by *Morpheus*. The *segmentation map* files are named according to the following scheme `morpheus_UDS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_UDS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 5. Additionally, a link to an archive containing all of the files associated with the UDS field is available in an additional section called *All Files*. See Appendix D for details.

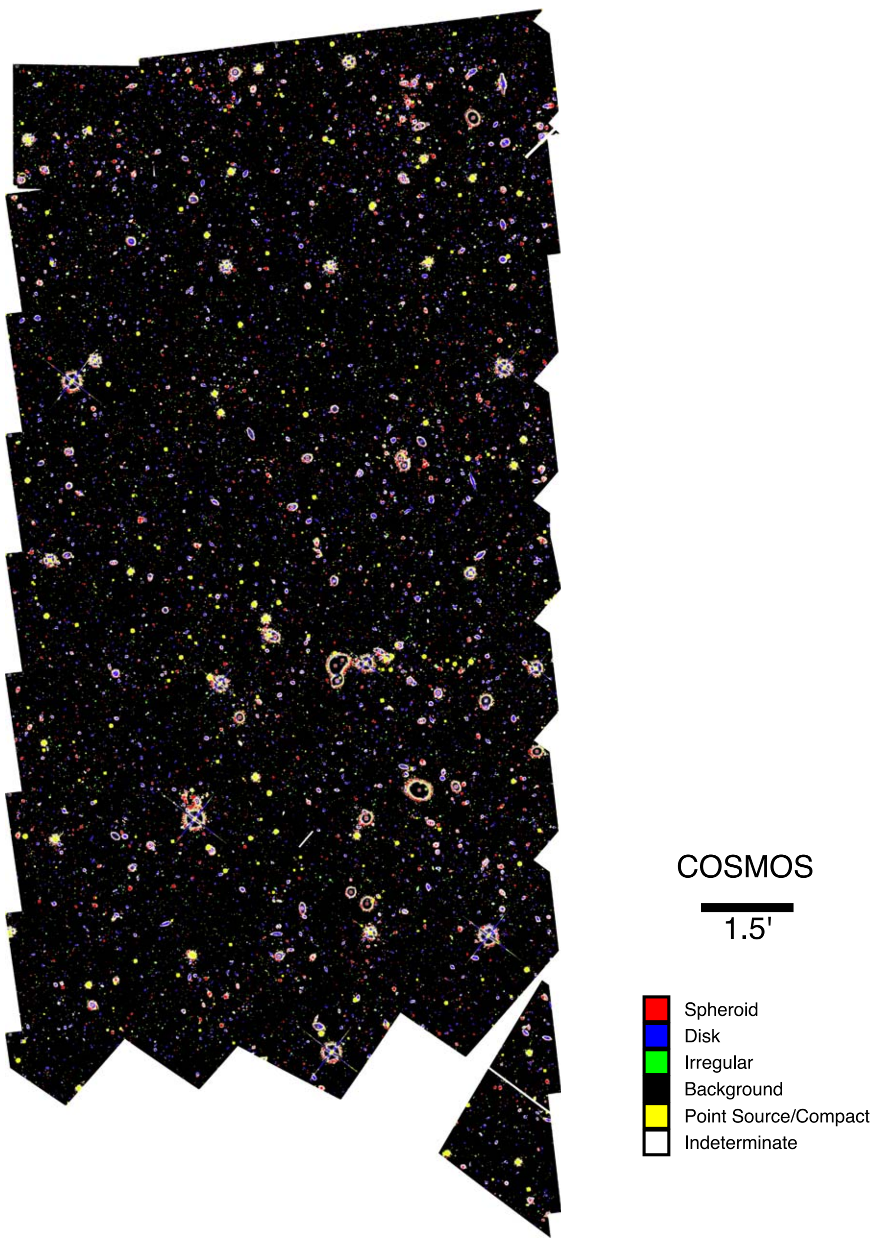

Figure D1. Color composite of the *Morpheus* morphological classifications for the COSMOS field from the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011).

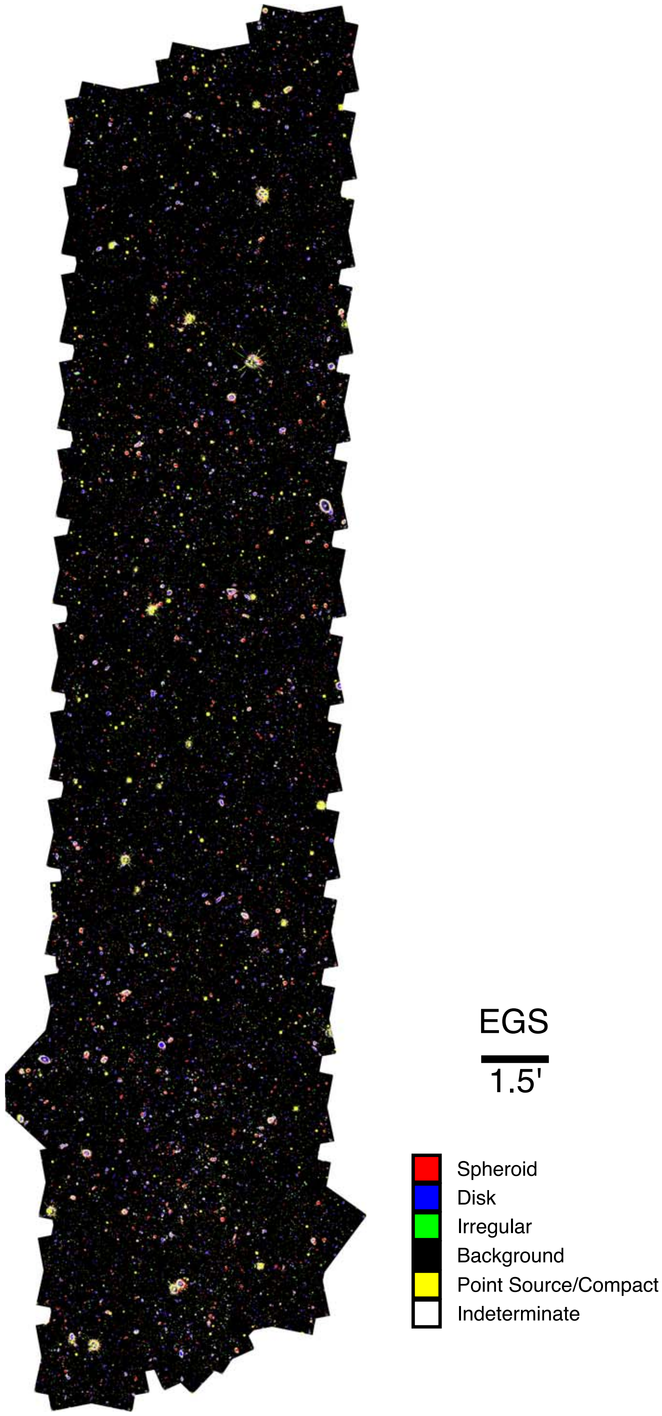

Figure D2. Color composite of the *Morpheus* morphological classifications for the EGS field from the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011).

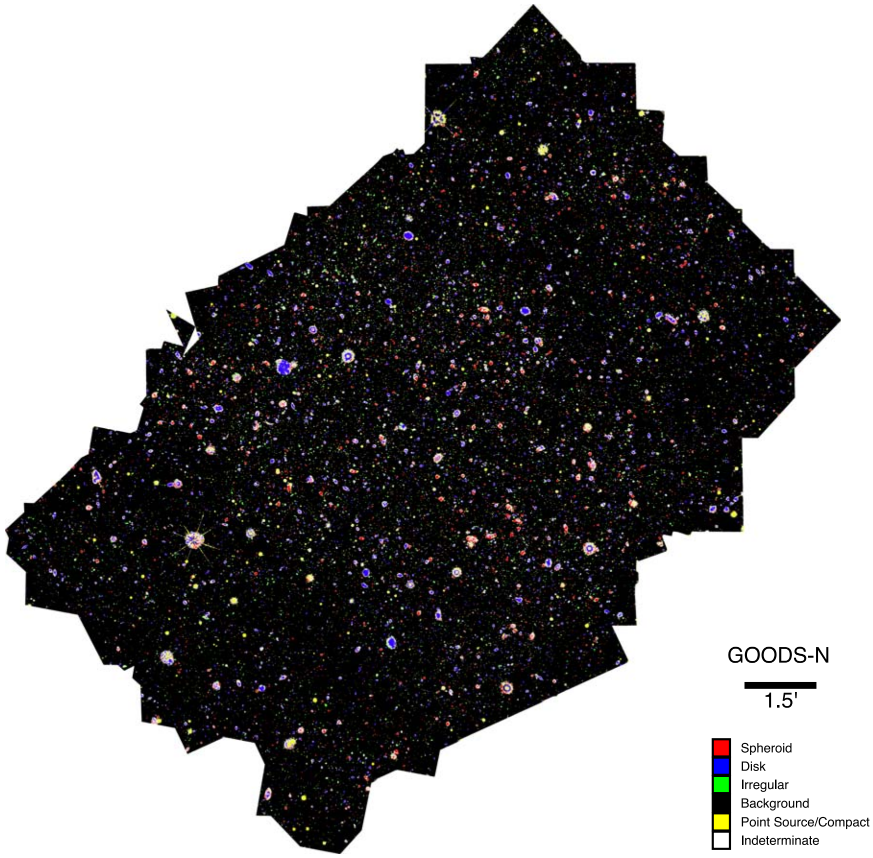

Figure D3. Color composite of the *Morpheus* morphological classifications for the GOODS North field from the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011).

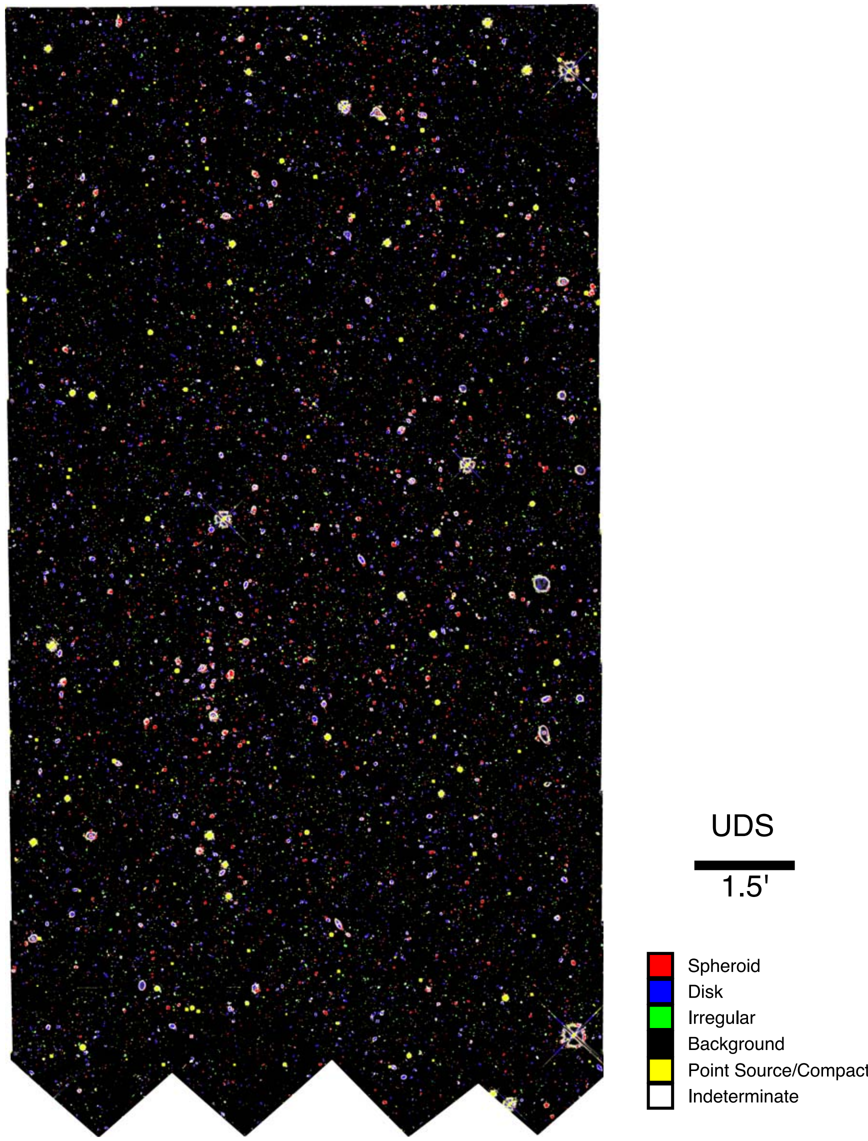

Figure D4. Color composite of the *Morpheus* morphological classifications for the UDS field from the CANDELS survey (Grogin et al. 2011; Koekemoer et al. 2011).

ORCID iDs

Ryan Hausen https://orcid.org/0000-0002-8543-761X
Brant E. Robertson https://orcid.org/0000-0002-4271-0364

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Abraham, R. G., Tanvir, N. R., Santiago, B. X., et al. 1996, MNRAS, 279, L47
Abraham, R. G., Valdes, F., Yee, H. K. C., & van den Bergh, S. 1994, ApJ, 432, 75
Abraham, R. G., & van den Bergh, S. 2001, Sci, 293, 1273
Agnello, A., Kelly, B. C., Treu, T., & Marshall, P. J. 2015, MNRAS, 448, 1446
Aihara, H., Arimoto, N., Armstrong, R., et al. 2018a, PASJ, 70, S4
Aihara, H., Armstrong, R., Bickerton, S., et al. 2018b, PASJ, 70, S8
Akeson, R., Armus, L., Bachelet, E., et al. 2019, arXiv:1902.05569
Allen, R. J., Kacprzak, G. G., Glazebrook, K., et al. 2017, ApJL, 834, L11
Beck, M. R., Scarlata, C., Fortson, L. F., et al. 2018, MNRAS, 476, 5516
Bell, E. F., van der Wel, A., Papovich, C., et al. 2012, ApJ, 753, 167
Bender, R., Burstein, D., & Faber, S. M. 1992, ApJ, 399, 462
Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393
Bezanson, R., van Dokkum, P. G., van de Sande, J., et al. 2013, ApJL, 779, L21
Binney, J. 1978, MNRAS, 183, 501
Binney, J., & Tremaine, S. 1987, Galactic Dynamics (Princeton, NJ: Princeton Univ. Press)
Boucaud, A., Huertas-Company, M., Heneka, C., et al. 2020, MNRAS, 491, 2481
Bruce, V. A., Dunlop, J. S., Cirasuolo, M., et al. 2012, MNRAS, 427, 1666
Bruce, V. A., Dunlop, J. S., Mortlock, A., et al. 2016, MNRAS, 458, 2391
Cireşan, D., Meier, U., & Schmidhuber, J. 2012, in Proc. IEEE Comput Vision Pattern Recogn. (Piscataway, NJ: IEEE), 3642, doi:10.1109/CVPR.2012.6248110
Conselice, C. J. 2003, ApJS, 147, 1
Conselice, C. J., Bershady, M. A., & Jangren, A. 2000, ApJ, 529, 886
Conselice, C. J., Blackburne, J. A., & Papovich, C. 2005, ApJ, 620, 564
Cooper, M. C., Griffith, R. L., Newman, J. A., et al. 2012, MNRAS, 419, 3018
Couprie, M., & Bertrand, G. 1997, Proc. SPIE, 3168, 136
Dai, J.-M., & Tong, J. 2018, arXiv:1807.10406
Dark Energy Survey Collaboration, Abbott, T., Abdalla, F. B., et al. 2016, MNRAS, 460, 1270
Deng, J., Dong, W., Socher, R., et al. 2009, in Proc. IEEE Conf. Comput. Vision Pattern Recogn. (Piscataway, NJ: IEEE), 248, doi:10.1109/CVPR.2009.5206848
Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441
Dimauro, P., Huertas-Company, M., Daddi, E., et al. 2018, MNRAS, 478, 5410
Djorgovski, S., & Davis, M. 1987, ApJ, 313, 59
Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., et al. 2019, MNRAS, 484, 93

Dressler, A. 1980, ApJ, 236, 351
Dressler, A., Lynden-Bell, D., Burstein, D., et al. 1987, ApJ, 313, 42
Dressler, A., Oemler, A., Jr., Couch, W. J., et al. 1997, ApJ, 490, 577
Elmegreen, D. M., Elmegreen, B. G., Rubin, D. S., & Schaffer, M. A. 2005, ApJ, 631, 85
Franx, M., van Dokkum, P. G., Förster Schreiber, N. M., et al. 2008, ApJ, 688, 770
Gardner, J. P., Mather, J. C., Clampin, M., et al. 2006, SSRv, 123, 485
González, R. E., Muñoz, R. P., & Hernández, C. A. 2018, A&C, 25, 103
Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, ApJS, 197, 35
Guo, Y., Ferguson, H. C., Giavalisco, M., et al. 2013, ApJS, 207, 24
Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. 2000, Natur, 405, 947
Hausen, R. 2020, morpheus-project/morpheus: Zenodo Release, v0.4.6, Zenodo, doi:10.5281/zenodo.3601680
Hemmati, S., Capak, P., Masters, D., et al. 2019, ApJ, 877, 117
Hezaveh, Y. D., Perreault Levasseur, L., & Marshall, P. J. 2017, Natur, 548, 555
Hocking, A., Geach, J. E., Sun, Y., & Davey, N. 2018, MNRAS, 473, 1108
Hubble, E. P. 1926, ApJ, 64, 321
Huertas-Company, M., Bernardi, M., Pérez-González, P. G., et al. 2016, MNRAS, 462, 4495
Huertas-Company, M., Gravet, R., Cabrera-Vives, G., et al. 2015, ApJS, 221, 8
Huertas-Company, M., Primack, J. R., Dekel, A., et al. 2018, ApJ, 858, 114
Huertas-Company, M., Rodriguez-Gomez, V., Nelson, D., et al. 2019, MNRAS, 489, 1859
Hunter, J. D. 2007, CSE, 9, 90
Illingworth, G., Magee, D., Bouwens, R., et al. 2016, arXiv:1606.00841
Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
Jiang, D., Liu, F. S., Zheng, X., et al. 2018, ApJ, 854, 70
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Kartaltepe, J. S., Mozena, M., Kocevski, D., et al. 2015, ApJS, 221, 11
Kawinwanichakij, L., Papovich, C., Quadri, R. F., et al. 2017, ApJ, 847, 134
Kelly, B. C., & McKay, T. A. 2004, AJ, 127, 625
Kelly, B. C., & McKay, T. A. 2005, AJ, 129, 1287
Kingma, D. P., & Ba, J. 2014, in Proc. Int. Conf. Learn. Represent., ed. Y. Bengio & Y LeCun (Wadern: dblp), https://dblp.uni-trier.de/rec/bibtex/journals/corr/KingmaB1
Kocevski, D. D., Faber, S. M., Mozena, M., et al. 2012, ApJ, 744, 148
Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, ApJS, 197, 36
Kormendy, J. 1977, ApJ, 218, 333
Krist, J. E., Hook, R. N., & Stoehr, F. 2011, Proc. SPIE, 8127, 81270J
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, arXiv:1110.3193
Lecun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436
Lee, B., Giavalisco, M., Williams, C. C., et al. 2013, ApJ, 774, 47
Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179
Lofthouse, E. K., Kaviraj, S., Conselice, C. J., Mortlock, A., & Hartley, W. 2017, MNRAS, 465, 2895
Lotz, J. M., Davis, M., Faber, S. M., et al. 2008, ApJ, 672, 177
Lotz, J. M., Primack, J., & Madau, P. 2004, AJ, 128, 163
LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv:0912.0201
Margalef-Bentabol, B., Conselice, C. J., Mortlock, A., et al. 2016, MNRAS, 461, 2728
Masters, D., Capak, P., Stern, D., et al. 2015, ApJ, 813, 53
Miller, T. B., van Dokkum, P., Mowla, L., & van der Wel, A. 2019, ApJL, 872, L14
Milletari, F., Navab, N., & Ahmadi, S.-A. 2016, arXiv:1606.04797
Momcheva, I. G., Brammer, G. B., van Dokkum, P. G., et al. 2016, ApJS, 225, 27
Morishita, T., Ichikawa, T., & Kajisawa, M. 2014, ApJ, 785, 18
Morningstar, W. R., Hezaveh, Y. D., Perreault Levasseur, L., et al. 2018, arXiv:1808.00011
Morningstar, W. R., Perreault Levasseur, L., Hezaveh, Y. D., et al. 2019, ApJ, 883, 14
Novikov, A. A., Lenis, D., Major, D., et al. 2018, IEEE Trans. Medical Imaging, 37, 1865
Oke, J. B., & Gunn, J. E. 1983, ApJ, 266, 713
Patel, S. G., Fumagalli, M., Franx, M., et al. 2013, ApJ, 778, 115
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Machine Learn. Res., 12, 2825, http://jmlr.org/papers/v12/
Peng, C. Y., Ho, L. C., Impey, C. D., & Rix, H.-W. 2010, AJ, 139, 2097
Peth, M. A., Lotz, J. M., Freeman, P. E., et al. 2016, MNRAS, 458, 963
Powell, M. C., Urry, C. M., Cardamone, C. N., et al. 2017, ApJ, 835, 22
Pratt, L. Y. 1993, in Advances in Neural Information Processing Systems 5, ed. S. J. Hanson, J. D. Cowan, & C. L. Giles (San Mateo, CA: Morgan Kaufmann Publishers), 204, http://papers.nips.cc/paper/641-discriminability-based-transfer-between-neural-networks.pdf
Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., et al. 2018, AJ, 156, 123
Rhodes, J., Nichol, R. C., Aubourg, É., et al. 2017, ApJS, 233, 21
Roberts, M. S., & Haynes, M. P. 1994, ARA&A, 32, 115
Robertson, B. E., Banerji, M., Brough, S., et al. 2019, NatRP, 1, 450
Robertson, B. E., Banerji, M., Cooper, M. C., et al. 2017, arXiv:1708.01617
Ronneberger, O., Fischer, P., & Brox, T. 2015, in Proc. Int. Conf. Medical Image Comput. Computer-Assisted Intervention, MICCAI 2015, ed. N. Navab (Cham: Springer), 234, doi:10.1007/978-3-319-24574-4_28
Rossum, G. 1995, Python Reference Manual, Tech. Rep. CS-R9525
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, Natur, 323, 533
Russakovsky, O., Deng, J., Su, H., et al. 2015, Int. J. Comput. Vision, 115, 211
Sérsic, J. L. 1968, Atlas de Galaxias Australes (Cordoba: Observatorio Astronomico)
Shen, S., Mo, H. J., White, S. D. M., et al. 2003, MNRAS, 343, 978
Sheth, K., Elmegreen, D. M., Elmegreen, B. G., et al. 2008, ApJ, 675, 1141
Sheth, K., Regan, M., Hinz, J. L., et al. 2010, PASP, 122, 1397
Shibuya, T., Ouchi, M., & Harikane, Y. 2015, ApJS, 219, 15
Simmons, B. D., Lintott, C., Willett, K. W., et al. 2017, MNRAS, 464, 4420
Simmons, B. D., Melvin, T., Lintott, C., et al. 2014, MNRAS, 445, 3466
Skelton, R. E., Whitaker, K. E., Momcheva, I. G., et al. 2014, ApJS, 214, 24
Smail, I., Dressler, A., Couch, W. J., et al. 1997, ApJS, 110, 213
Smith, M. J., & Geach, J. E. 2019, MNRAS, 490, 4985
Sobel, I., & Feldman, G. 1968, An Isotropic 3x3 Image Gradient Operator
Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Cardoso, M. J. 2017, in Proc. Int. Workshop on Deep Learning in Medical Image Analysis, CLMIA 2017 ML-CDS, ed. M. J. Cardisi et al. (Berlin: Springer), 240, doi:10.1007/978-3-319-67558-9_28
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, in Proc. IEEE Conf. Comput. Vision Pattern Recognition (Piscataway, NJ: IEEE), 2818, doi:10.1109/CVPR.2016.308
Tarsitano, F., Hartley, W. G., Amara, A., et al. 2018, MNRAS, 481, 2018
Toft, S., van Dokkum, P., Franx, M., et al. 2007, ApJ, 671, 285
Tremaine, S., Gebhardt, K., Bender, R., et al. 2002, ApJ, 574, 740
Trujillo, I., Förster Schreiber, N. M., Rudnick, G., et al. 2006, ApJ, 650, 18
Trujillo, I., Rudnick, G., Rix, H.-W., et al. 2004, ApJ, 604, 521
Tuccillo, D., Huertas-Company, M., Decencière, E., et al. 2018, MNRAS, 475, 894
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, CSE, 13, 22
van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., et al. 2014, PeerJ, 2, e453
van der Wel, A., Bell, E. F., Häussler, B., et al. 2012, ApJS, 203, 24
van der Wel, A., Franx, M., van Dokkum, P. G., et al. 2014, ApJ, 788, 28
van Dokkum, P. G., Whitaker, K. E., Brammer, G., et al. 2010, ApJ, 709, 1018
Vaucouleurs, G. 1959, HDP, 53, 275
Walmsley, M., Smith, L., Lintott, C., et al. 2020, MNRAS, 491, 1554
Weigel, A. K., Schawinski, K., Caplar, N., et al. 2017, ApJ, 845, 145
Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, A&AS, 44, 363
Whitaker, K. E., Franx, M., Bezanson, R., et al. 2015, ApJL, 811, L12
Willett, K. W., Galloway, M. A., Bamford, S. P., et al. 2017, MNRAS, 464, 4176
Willett, K. W., Lintott, C. J., Bamford, S. P., et al. 2013, MNRAS, 435, 2835
Willett, K. W., Schawinski, K., Simmons, B. D., et al. 2015, MNRAS, 449, 820
Williams, C. C., Curtis-Lake, E., Hainline, K. N., et al. 2018, ApJS, 236, 33
Williams, C. C., Giavalisco, M., Bezanson, R., et al. 2017, ApJ, 838, 94
Wuyts, S., Förster Schreiber, N. M., van der Wel, A., et al. 2011, ApJ, 742, 96
Yano, M., Kriek, M., van der Wel, A., & Whitaker, K. E. 2016, ApJL, 817, L21
Zhang, H., Primack, J. R., Faber, S. M., et al. 2019, MNRAS, 484, 5170
Zirm, A. W., van der Wel, A., Franx, M., et al. 2007, ApJ, 656, 66