

Deblending and Classifying Astronomical Sources with Mask R-CNN Deep Learning

Colin J. Burke,^{1,2★} Patrick D. Aleo,^{1,3†} Yu-Ching Chen,^{1,2} Xin Liu,^{1,2}
John R. Peterson,⁴ Glenn H. Sembroski,⁴ Joshua Yao-Yu Lin⁵

¹*Department of Astronomy, University of Illinois at Urbana-Champaign, 1002 West Green Street, Urbana, IL 61801, USA*

²*National Center for Supercomputing Applications, 1205 West Clark Street, Urbana, IL 61801, USA*

³*Advanced Visualization Laboratory, National Center for Supercomputing Applications, 1205 West Clark Street, Urbana, IL 61801, USA*

⁴*Department of Physics and Astronomy, Purdue University 525 Northwestern Avenue, West Lafayette, IN 47907, USA*

⁵*Department of Physics, University of Illinois at Urbana-Champaign, 1110 West Green Street, Urbana, IL 61801, USA*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We apply a new deep learning technique to detect, classify, and deblend sources in multi-band astronomical images. We train and evaluate the performance of an artificial neural network built on the Mask R-CNN image processing framework, a general code for efficient object detection, classification, and instance segmentation. After evaluating the performance of our network against simulated ground truth images for star and galaxy classes, we find a precision of 92% at 80% recall for stars and a precision of 98% at 80% recall for galaxies in a typical field with ~ 30 galaxies/arcmin². We investigate the deblending capability of our code, and find that clean deblends are handled robustly during object masking, even for significantly blended sources. This technique, or extensions using similar network architectures, may be applied to current and future deep imaging surveys such as LSST and WFIRST. Our code, *Astro R-CNN*, is publicly available at https://github.com/burke86/astro_rcnn.

Key words: techniques: image processing – methods: data analysis – galaxies: general

1 INTRODUCTION

The next generation of astronomical surveys such as the Large Synoptic Survey Telescope (LSST; Ivezić et al. 2019), the Wide-Field Infrared Survey Telescope (WFIRST; Spergel et al. 2013), and Euclid (Amiaux et al. 2012) will produce unprecedented amounts of imaging data throughout the 2020s. This quickly-approaching era demands efficient, uniform, and robust techniques to detect, classify, and analyze sources in images.

The task of star/galaxy classification is a long-standing problem in astronomy, dating back to the likes of Messier (1781). Until recently, a technique known as morphological separation (Sebok 1979; Valdes 1982) was the popular choice, which involved a simple assumption: galaxies are resolved sources, and stars point sources. Sebok (1979) and Valdes (1982) pioneered a Bayesian approach focusing on classifying objects by maximizing the probability of object class models matching the observed pixel intensity distributions. In contrast, Jarvis & Tyson (1981) use a parametric

method, where clustering of data points of measured pixel intensity distribution determines the classification.

Next-generation ground-based surveys, such as LSST, will detect numerous unresolved and marginally-resolved galaxies, particularly near the photometric limit. In this regime, a strictly morphology-based approach will not be able to consistently differentiate between stars and galaxies (Kim et al. 2015). Thus, several studies have introduced machine learning methods such as decision trees (Vasconcellos et al. 2011), a blend of different learning approaches (Kim et al. 2015; Soumagnac et al. 2015), and deep learning (Serra-Ricart et al. 1996; Kim & Brunner 2017). See Cheng et al. (2019) for an overview and comparison of different machine learning galaxy classification techniques.

As the both the sensitivity and depth of surveys increase, we will encounter larger numbers of blended (overlapping) sources due to line-of-sight projection or source interaction (i.e. galaxy-galaxy mergers). The detection of both fainter galaxies and more extended regions of objects will increase the probability of blending. If blends are not identified, they will bias results from pipelines that assume object isolation. Some important examples include photometric redshifts (Boucaud et al. 2019) and weak lensing (Arneson 2013). Once LSST begins its survey, efficient deblending

★ E-mail: colinjb2@illinois.edu

† E-mail: paleo2@illinois.edu

techniques will be a necessity, and thus been recognized a high priority in the field.

Current estimates place the fraction of significantly blended galaxies (3" center-to-center distance) in LSST images at roughly 50% (Dawson et al. 2016; Dawson & Schneider 2014; Chang et al. 2013). Chang et al. (2013) estimates roughly 10% of galaxies to be blends with a 1" center-to-center distance in a typical region of the sky (around 37 galaxies per arcmin²). In all, if effective deblending algorithms are not put in place during the ten-year LSST survey, roughly 200 million galaxies could be discarded each year according to Reiman & Göhre (2019).

Even in current surveys, such as the Dark Energy Survey (DES; Abbott et al. 2018), crowded fields are challenging for the current pipeline. For this reason, a deblending code was developed by Zhang et al. (2015) for DES images of clusters of galaxies. See Sevilla-Noarbe et al. (2018) for a summary of star/galaxy classification techniques used in DES. The Hyper Suprime-Cam (HSC) Subaru Strategic Program (Aihara et al. 2018) also suffers from poor photometry in crowded fields due to significant blending (Huang et al. 2018; Aihara et al. 2019). Several existing codes for source detection and classification incorporate simple deblending algorithms into their frameworks, such as FOCAS (Jarvis & Tyson 1981), SExtractor (Bertin & Arnouts 1996), and NEXT (Andreon et al. 2000). However, these approaches are highly sensitive to configuration parameters, such as the density of sources in the field. Additionally, these codes can be inefficient compared to machine learning approaches.

Recently, new algorithmic techniques have been developed in the context of LSST such as the work of Lupton (2014) and SCARLET (Melchior et al. 2018). Similar to Astro R-CNN, SCARLET takes advantage of multi-band imaging when there are overlapping sources. It achieves this by utilizing non-negative matrix factorization, where one matrix factor stores the source color and the other spatial shape information. Then, the likelihood function is minimized with respect to the inherent non-negativity constraints via a proximal gradient descent update.

The recent work of Hausen & Robertson (2019) combined deep learning semantic segmentation with mask separation algorithms to classify and deblend galaxies in the Hubble Space Telescope images. Their code, *Morpheus*, is based on the U-Net (Ronneberger et al. 2015) architecture, and like Astro R-CNN, performs pixel-level classifications. Further, both codes uniquely identify source and background pixels, allowing for a singular, cohesive analysis of object detection and classification. Although, *Morpheus* utilizes user-supplied segmentation maps and utilizes the classic watershed transform algorithm (Couprie & Bertrand 1997) to separate source and background pixels. These source pixels are subsequently classified in terms of their morphology (background, disk, spheroid, irregular, point source/compact).

Some recent works have used more experimental artificial neural network techniques for deblending and shown promising results (Reiman & Göhre 2019; Boucaud et al. 2019; González et al. 2018). Zhang & Bloom (2019) used deep learning to mask cosmic rays. With recent ubiquity of powerful machines with multiple graphics processing units (GPUs) and a plethora of machine learning libraries, these techniques have never been more accessible or appealing.

Many classification and deblending codes operate

monochromatically, not taking into account source color gradient information. In the context of DES and LSST, their uniform, multi-band data should be exploited to assist with source classification and deblending. In addition, the source spectral energy distribution (SED) information can be used to help discriminate between stars and unresolved galaxies or quasars. Multi-band images can also be used with convolutional neural networks to estimate photometric redshifts of galaxies and quasars (Pasquet et al. 2019; D’Isanto & Polsterer 2018).

In this work, we develop a new deep learning method based on the Mask Region-based Convolutional Neural Network (Mask R-CNN) framework (He et al. 2017) to perform all tasks of source detection classification, and deblending in a single machine learning framework. We train and validate our network using simulated images and catalogs, and test its performance using DECam Legacy Survey (DECaLS; Dey et al. 2019) images of a crowded field. After training, this method is extremely efficient, detecting, classifying, and segmenting a 512×512 pixel² image in $\lesssim 100$ milliseconds using a single NVIDIA Tesla V100 GPU. This work may be extended to other telescopes and surveys using transfer learning. The code is open source and available at the Astro R-CNN GitHub repository (Burke et al. 2019).

This paper is organized as follows. In §2, we introduce the Mask R-CNN framework and describe the architecture of our implementation. We also explain our training procedure using transfer learning and simulated images. In §3, we present the results of our trained neural network, validate the results using the simulated catalog as a ground truth, and evaluate its performance. We present results using real DECam images. In §4, we discuss the implications of our method and its benefits and drawbacks compared to existing work. In §5, we summarize our findings and conclude.

2 NETWORK & TRAINING

Object detection, classification, and instance segmentation is an active area of research in the field of computer vision. There are several machine learning-based solutions that perform semantic/instance segmentation, such as YOLO (Redmon et al. 2015), YOLACT (Bolya et al. 2019), PANet (Liu et al. 2018), and TeraNet (Iglavikov & Shvets 2018).

Recently, He et al. (2017) developed a novel and general framework for instance segmentation called Mask R-CNN. This extends earlier work of Fast/Faster R-CNN (Girshick 2015; Ren et al. 2015), a deep convolutional neural network for classification and bounding box recognition. Mask R-CNN adds a parallel branch for instance segmentation (Fig. 1). The Mask R-CNN framework is highly efficient and robust to occlusion. Several recent works have applied Mask R-CNN to different fields from cellular biology (Tsai et al. 2019) to remote sensing (Zhang et al. 2018). We select this framework for its recent ubiquity and maturity. Even more recent works have built-upon Mask R-CNN (e.g. Mask Scoring R-CNN; Huang et al. 2019) and Zimmermann & Siems (2018), but are not explored in this work.

In this section, we describe our Mask R-CNN implementation for the purpose of star/galaxy detection, classification, and deblending. We also outline our training, validation, and test dataset generation using simulated im-

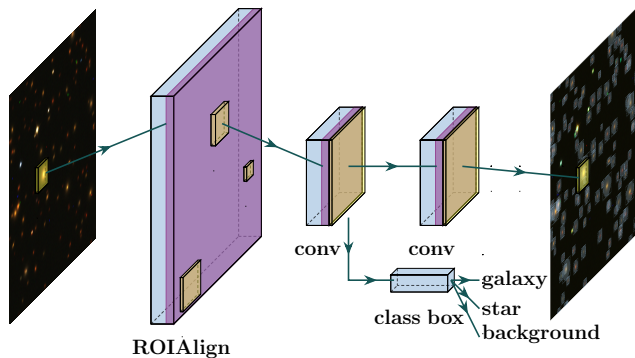


Figure 1. A simple schematic of our Mask R-CNN framework for instance segmentation, based on Figure 1 of He et al. (2017). The feature map in RoIAlign has a few region of interest (RoI) examples, shown in yellow, where one galaxy is propagated through the network. In the convolutional layers this RoI is a fixed size feature map. Mask R-CNN extends Faster R-CNN by parallelizing the branch for segmentation mask prediction with the branch for classification and bounding box regression.

ages. Our training procedure is a form of supervised learning where simulated images and labeled masks are used to train the network. We describe this process in detail below.

2.1 Implementation

The code developed in this work extends the Python language implementation of Mask R-CNN from Abdulla (2017). This code is built on the Keras library (Chollet et al. 2015) using a TensorFlow (Abadi et al. 2016) backend. We allow multi-band flexible image transport system (FITS) files (Pence et al. 2010) as image input during both the training and detection (inference) modes. The final segmentation masks are saved as multi-extension FITS files. The problem of source detection, classification, and deblending in astronomical surveys is well-suited to the Mask R-CNN framework which performs all tasks in one cohesive package.

Our Mask R-CNN implementation uses the standard 101-layer deep residual network (ResNet-101; He et al. 2016) architecture as its backbone. On its own, ResNet-101 is a feature extractor, wherein earlier layers detect low-level features (e.g. corners or edges) and later levels detect high-level features (e.g. galaxies or stars) using residual learning. By using shortcut connections in 3-layer deep residual blocks, ResNet-101 is able to solve the *degrading accuracy problem*: with increasing network depth, accuracy becomes saturated and degrades quickly (Bengio et al. 1994; He et al. 2016). This occurs when back-propagating the gradient, where repeated multiplications involving small weights tend to create smaller and smaller gradients to the point of ineffectiveness (Huang et al. 2016). This is referred to as the *vanishing gradient problem*.

As with Faster R-CNN, Mask R-CNN is a feature pyramid network (FPN) which defines hierarchically-sized *anchors* for multi-scale object recognition. This FPN adds a second pyramid network that allows high level features to be passed down to lower levels and vice-versa, so that any layer has an awareness of both low- and high-end features

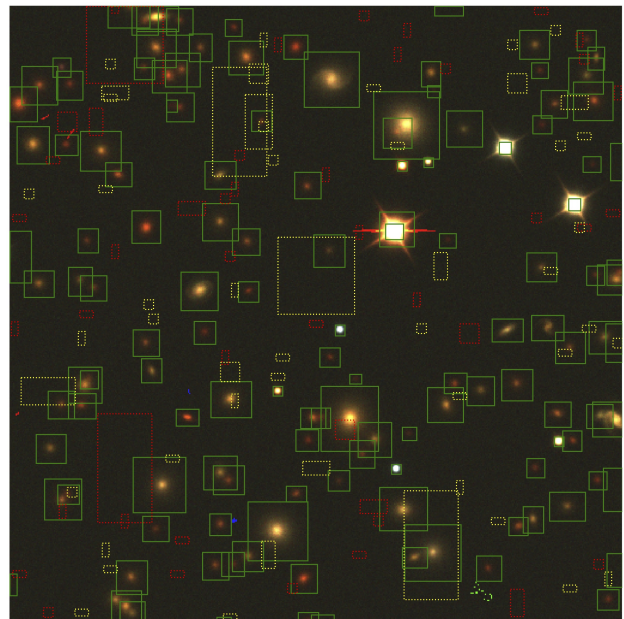


Figure 2. Example of a simulated DECam image cutout with RoI bounding boxes overlaid. Positive RoIs (solid green) correspond to a region where a significant detection is identified. For clarity, only 50 neutral (dashed yellow) and 50 negative (dashed red) RoIs are shown, and the RoI bounding boxes are enlarged. Note that the RPN scans over the backbone feature map and not the image directly as shown here; this is done for illustrative purposes to develop an intuition of how the RPN will identify features for extraction.

(Lin et al. 2017). With a feature map at each level of the second pyramid (instead of a single backbone feature in the top layer of the first pyramid), the one most appropriate for the size of the object is chosen at runtime, ultimately enabling better feature extraction (Abdulla 2017; Lin et al. 2017).

Once the appropriate backbone feature map is chosen, this is fed to the region proposal network (RPN) for scanning (Ren et al. 2015). The RPN is a neural network which scans in a sliding window fashion over thousands of anchors—overlapping areas of different sizes and aspect ratios—to generate two outputs for each anchor: an associated anchor class and bounding box refinement. For anchor class, the anchor will be assigned as *foreground/positive* or *background/negative*, where the former suggests the existence of an object contained within it, and the latter does not (a third type, *neutral*, does not contribute to training). If assigned positive, the bounding box refinement will generate a suggested shifted bounding box to place the object at its center. These predictions most likely to contain objects are called proposals and sent to the next stage as regions of interest (RoIs; Fig. 2).

In this work we implement three RoIs classifications: star, galaxy, or background. However, the background class is trivial and will be ignored in our analysis. This essentially makes our model a simple binary classifier. Now, each RoI is assigned another bounding box. Unfortunately, RoI boxes are likely to be of all different sizes, which confuses classifiers. To circumvent this, RoIAlign samples the feature maps and

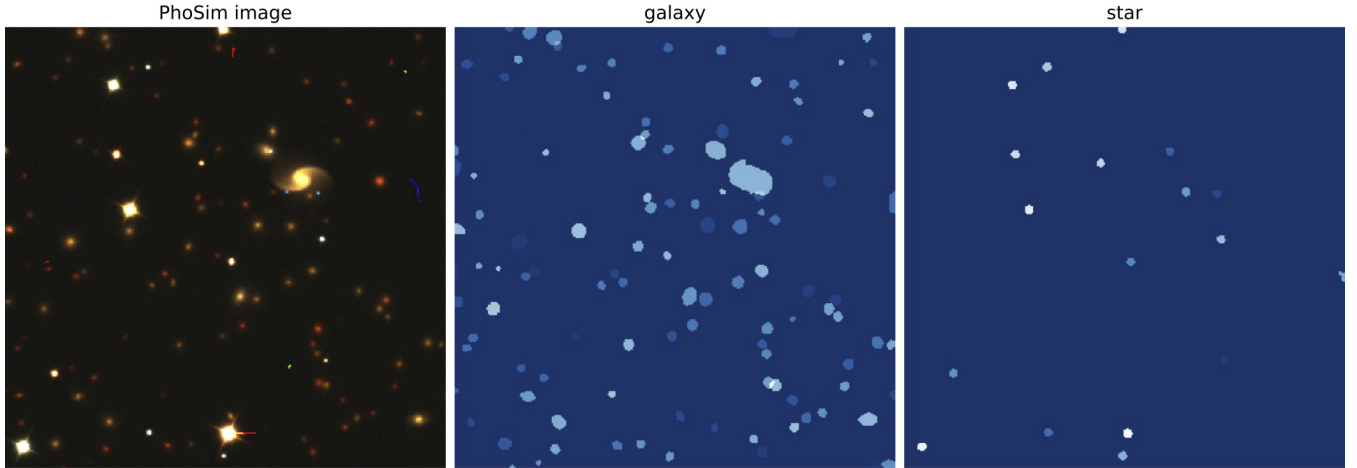


Figure 3. Typical example of PhoSim DECam training data. The simulated color image of a crowded field (*left panel*) and masks corresponding to both galaxy (*center panel*) and star (*right panel*) classes are pictured. Note the realism of the simulated images: the distribution and colors of sources, complex galaxy morphology, diffraction and blooming (most noticeably for the bright stars), bleed trails, and cosmic rays are clearly visible.

applies bilinear interpolation¹ such that there is a fixed input size (represented by the first layer after RoIAlign in Fig 1). RoIs are aligned with the RoIAlign layer and piped through the CNN, a process founded by He et al. (2017) to improve AP scores over the standard RoIPool (Girshick 2015) due to its preservation of exact spatial locations critical to feature extraction (He et al. 2017).

In final instance segmentation (masking) stage, each pixel is assigned a class which can be used to mask sources and obtain (deblended) cutouts from full-scale images. Mask shapes are stored at bounding box positions using the “mini masks” feature to optimize the training. We use a fixed image size of 512×512 (image sub-region) ×3 (*g,r,z* bands) pixel³ to simplify the training. These image sub-regions are used in our extension of Mask R-CNN, and the output can be tiled to full-scale images or mosaics.

2.2 Training Set

A common problem of artificial neural network techniques is the lack of a sufficiently-sized, uniform, and unbiased training set. For example, one may wish to train on astronomical sources which are too faint for current surveys or too rarely observed. We alleviate these problems by using simulated images of crowded extragalactic fields. Importantly, using simulated images gives us a large training set of a known catalog of stars and galaxies. This data is used as a truth to test our network’s performance without the misclassification errors in real catalogs.

We invoke The Photon Simulator (PhoSim; Peterson et al. 2015) to simulate DECam images. PhoSim is an *ab initio* photon Monte Carlo code originally developed for LSST. We use PhoSim with the Blanco 4-m DECam telescope/instrument options adapted from Flaugh et al.

(2015). The PhoSim DECam implementation includes a full description of the optical prescription and focal plane, as described by Cheng (2017). PhoSim can quickly simulate images of pseudorandom star and galaxy catalogs under a distribution of typical observing conditions. PhoSim includes all relevant physics of the atmosphere, telescope and camera optics, and detector. Below, we detail our procedure for generating a large, uniform, and realistic training set of PhoSim DECam images.

Simulated galaxies are described using PhoSim’s **serSicComplex** three-dimensional galaxy model. This model includes three-dimensional ellipsoidal Sérsic profiles (Sérsic 1963) for both the bulge and disk morphology, along with additional parameters for describing irregular knots and spiral structure and their three-dimensional orientations. By sampling these parameters from realistic distributions, PhoSim can simulate images with spiral, elliptical, or irregular galaxies. The approximate number density and population of galaxies is derived from the cosmic star-formation history described in Madau & Dickinson (2014). Galaxies are given random, typical SEDs for the bulge and disk, derived from Mollá et al. (2009). A simple model accounting for the different stellar ages and metallicities of the bulge and disk is considered (Peletier & Balcells 1996; Gallazzi et al. 2005).

Milky Way stars are simulated as point sources whose number density distribution varies as a function of galactic latitude. The stellar population follows the initial mass function described in Kroupa (2001). Stellar SEDs are derived from Castelli & Kurucz (2003) and Kurucz (1993). Stellar metallicities and atmosphere parameters are derived from Allende Prieto et al. (2004) and Prugniel et al. (2011).

Next, we outline our procedure for generating the training set data. The only additional change made to PhoSim is to use an increased number density of galaxies to simulate a more dense extragalactic field, such as a cluster of galaxies. We use a 4× overdensity of galaxies which works out to about 30 galaxies/arcmin² in our images. We execute PhoSim in two stages, which are outlined as follows:

¹ The code we adopted uses TensorFlow’s *crop_and_resize* function as a numerically-efficient approximation.

(i) Simulate 512×512 pixel² (≈ 5 arcmin²) DECam images in 3 bands (g, r, z) of a pseudorandom crowded extragalactic field with a 150 s integration time. We simulate stars and galaxies according to PhoSim’s realistic distributions between the typical magnitude range $12 < g < 23$. This integration time and limiting magnitude roughly approximates typical DECaLS data release 7 coadds (Dey et al. 2019). Given the various observing conditions, the typical minimum signal-to-noise ratio limit for sources in our images is $S/N \sim 2$. Fainter objects will ultimately contribute to objects missed in Mask R-CNN detection.

(ii) For every object in the field, simulate a 512×512 pixel² g -band image with no background using the same catalog from step (i). This second stage is used to produce the object masks. This way, occlusion can be handled from the simulated masks and the network can be trained to identify separate masks for blended sources. Multiple instances of PhoSim are run simultaneously to parallelize this stage.

To speed-up the simulations, we use PhoSim’s perfect optics configuration with a simple ~ 1 arcsec PSF model. This excludes all higher-order optical perturbations and atmospheric effects. Although our simulations are idealized and do not capture all systematics, the neural network only needs to capture the basic morphology of objects and noise in reduced images. Instead, it is sufficient to employ data augmentation afterwards, described in §2.4, to vary the images and reduce overfitting. Each image has ~ 150 object masks corresponding to a star or galaxy. We generate 1,000 simulated DECam images in our training set, for a total training set of approximately 150,000 astronomical sources. An example of a typical PhoSim training set image with its galaxy and star masks is shown in Fig. 3. We show the distribution of object sizes (in pixels) in our images in Fig. 4.

An additional validation set with 250 images is generated using PhoSim in the same manner as the training set. The validation set is used to reduce overfitting and tune the hyperparameters to find the best model. During the training phase, the model is tested on the validation set to ensure that it is generalizing sufficiently. If the model is solely classifying well on the training set but not the validation set, this is a sign of overfitting. Similarly, we also generate a test dataset of size 50 to get an unbiased evaluation of the performance of the network.

2.3 Data Standardization

The color values in each training image are assigned to re-scaled values in each band according to the z-score normalization prescription,

$$\begin{aligned} R &= A(z - \bar{z})/\sigma_z \\ G &= A(r - \bar{r})/\sigma_r \\ B &= A(g - \bar{g})/\sigma_g \end{aligned} \quad (1)$$

where R is pixel values in the red channel, $\bar{z}$ is the z -band mean value, and σ_z is the z -band standard deviation (and similarly for the green G and blue B channels using the r and g -bands respectively). The scale factor A can be adjusted, for example to correct for exposure times or changes in the instrument sensitivity. We fix $A = 10^4$ for the training. It is important to preserve the relative values between color

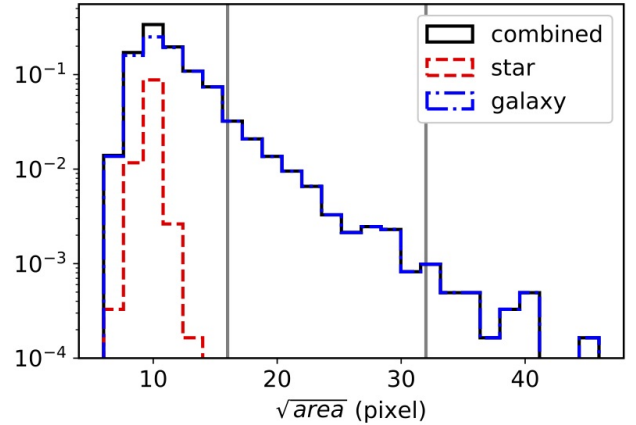


Figure 4. Frequency histogram of bounding box area for objects in the test dataset. The star and galaxy histograms are normalized to the combined stars+galaxies distribution. This figure shows the distribution of object sizes in our images. The vertical grey lines show the 16^2 and 32^2 pixel² boundaries used for the analysis in section §3.1.

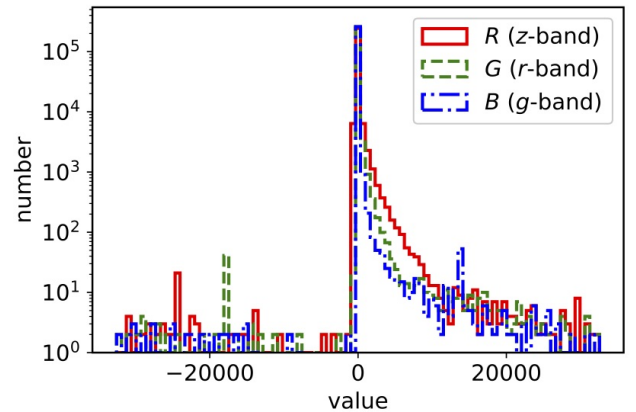


Figure 5. Histogram of normalized pixel values following Eq. 1 in a typical simulated training image. The red R , green G , and blue B channels are shown in their respective colors.

channels so that the source color information is retained. We perform this standardization on each set of images (an example is shown in Fig. 5), because the distribution of values can vary greatly in astronomical images depending on objects in the field and observing setup/conditions. This standardization of the image values means our network is not sensitive to the exposure time, gain of the detector, or the final normalization of the reduced images from an image reduction pipeline. This same data standardization is performed on real images during inferencing.

2.4 Data Augmentation

We employ the technique of data augmentation (Krizhevsky et al. 2012; Dieleman et al. 2015) to reduce network overfitting. We perform several image transformations to increase the robustness of our network. These data augmentations preserve object masks and classes. In this work, we augment



Figure 6. Example of Mask R-CNN person segmentation masks from the MS COCO database (images taken from He et al. (2017), Fig. 7). We employ transfer learning by initializing our training procedure with MS COCO precomputed weights before retraining on simulated astronomical images.

the data by randomly applying zero to four of the following augmentations:

- Rotate: The image and masks are rotated 90, 180, or 270 degrees.
- Mirror: The image and masks arrays are mirrored left-right or up-down.
- Blur: Smooth the image using a two-dimensional Gaussian kernel with size σ (in pixels) sampled from a random uniform distribution in the interval [2.0, 6.0). This blurs the image and masks, mimicking different PSF sizes.
- Add: Add or subtract random values element-wise to each image channel. The possible values are restricted to a range of $\pm 10\%$ times the maximum value in the image.

These simple augmentations mimic additional observing setups and conditions at little computational expense. With the addition to the random stars and galaxies in the PhoSim images, our network learns rotational invariance. These image augmentations help our network generalize its results to real images or images with slightly different features than the training set.

2.5 Transfer Learning

Transfer learning is a technique in machine learning where networks can generalize knowledge of one task to complete a different but related task. (See Tan et al. (2018) for an overview of deep transfer learning.) In one example of transfer learning, pre-trained weights from one dataset are used as initial conditions for training on another dataset. This improves the speed of training and reduce overfitting of the network. We use Mask R-CNN weights provided by Abdulla (2017) trained on the Microsoft Common Objects in Context (MS COCO) dataset (Lin et al. 2014) as the starting point for our training procedure. MS COCO is a dataset of $\sim 328,000$ images with 91 classes of everyday objects (Fig. 6).

2.6 Training

Our network is trained in two stages using stochastic gradient descent (Kingma & Ba 2014). Stochastic gradient descent updates the model parameters (weights) θ_j by minimizing the cost function $J(\theta)$ in the equation

$$\theta_{j+1} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j), \quad (2)$$

where α is the learning rate. The learning rate is a hyperparameter that is fine-tuned so that the model avoids trapping in local minima and achieves convergence. The first stage performs a re-training of the head layers with a learning rate of $\alpha = 10^{-3}$. The second stage trains all layers with a learning rate starting at $\alpha = 10^{-4}$ which we decrease progressively to $\alpha = 10^{-6}$. We use 50 learning epochs in total (see Appendix A). Nearly all 1,000 sets of training images and 250 validation images are processed per epoch. By re-scaling our training data to 16-bit integer arrays, we can read-in several sets of images and masks per GPU simultaneously using ResNet-101.

When training, each sampled RoI has an associated multi-task loss, following the form $L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}$ (He et al. 2017). Here, L_{cls} is the classification loss $-\log p_u$ for ground-truth class u and discrete probability distribution per RoI $p = (p_0, \dots, p_K)$ over $K + 1$ classes (Girshick 2015). L_{box} is the bounding-box loss as defined in Girshick (2015). Because the mask branch contains K binary masks of resolution $m \times m$ for each of the K classes, a per-pixel sigmoid is applied and thus defines L_{mask} as the average binary cross-entropy loss. This specific form of the loss was chosen by He et al. (2017) instead of the more commonly used per-pixel softmax and multinomial cross-entropy in fully convolutional network implementation to allow the network to generate masks across classes without competition between them. This is vital for decoupling class and mask prediction, a key feature of the Mask R-CNN architecture.

We use four state-of-the-art NVIDIA Tesla V100 GPUs (each with 5,120 cores and 16 GB high bandwidth memory) to train on 1,000 simulated color images. To speed-up the training, we employ model-based parallelism. Our model is copied onto each GPU where the training workload is divided equally before the weights are brought back together. Our training took ~ 3 hours to complete (wall time) and reached a total training set losses of $L_{\text{cls}} = 0.209$ $L_{\text{box}} = 0.208$ $L_{\text{mask}} = 0.311$. After this initial cost to train the network, detection and inference can be performed on images in less than a second. Our loss curve during training is shown in Appendix A.

3 RESULTS

To validate our trained Mask R-CNN network, we test its performance against simulated PhoSim images from the test dataset. The test dataset is not used in the training, thus it can be used to give an unbiased estimate of the network performance. Then, we assess our network using a real DECam image of a crowded field and present examples of deblending.

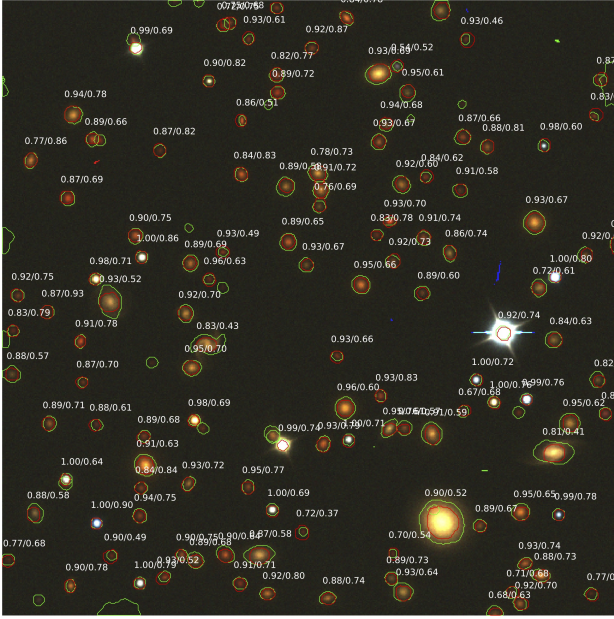


Figure 7. Typical example of detection inference (red) and ground truth (green) in the PhoSim validation dataset. The intersection over union is shown in white text near each positive detection. Note the successful recognition of the saturated star and several blended objects.

3.1 Network Performance

We use our simulated PhoSim catalog as a truth catalog to validate and evaluate the performance of our Mask R-CNN implementation. Throughout this section we use the test dataset, on which the network is not trained. Thus, we can avoid any misclassification bias that would come from using a real image and catalog. However, there may be systematic differences between the simulated and real images that are not taken into account in this section.

To quantify the performance of our network’s classification capability, we calculate the precision and recall for each image in the test dataset. The precision p (purity) and recall r (completeness) are defined as follows:

$$p = \frac{TP}{TP + FP} \quad (3)$$

$$r = \frac{TP}{TP + FN} \quad (4)$$

where TP=true positive, TN=true negative, FP=false positive, FN=false negative. A detection is considered positive if its ranked output (detection confidence) is greater than a given minimum detection confidence threshold (this value can be adjusted, but we choose 0.5 throughout this work). Note that the detection confidence is not the same as the Bayesian significance given by a S/N calculation. By comparing our network’s classification results to the PhoSim catalog, we can evaluate the precision at various recalls. In this analysis, we analyze the precision and recall for both star and galaxy classes as well as both classes jointly (combined).

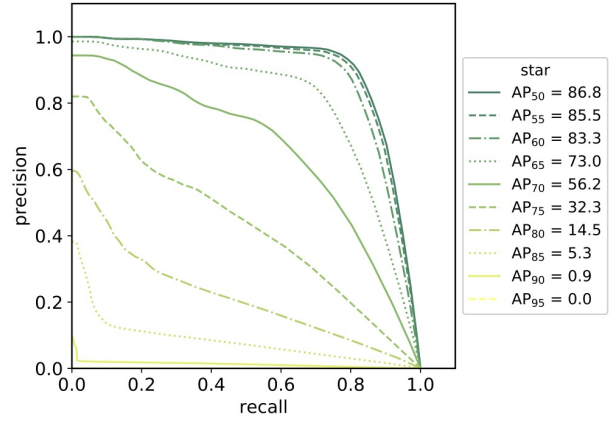


Figure 8. Precision-recall curves and mean AP scores calculated at varying IOU thresholds averaged over the test dataset for stars. We find IOU=0.5 gives a mean AP score of 86.8.

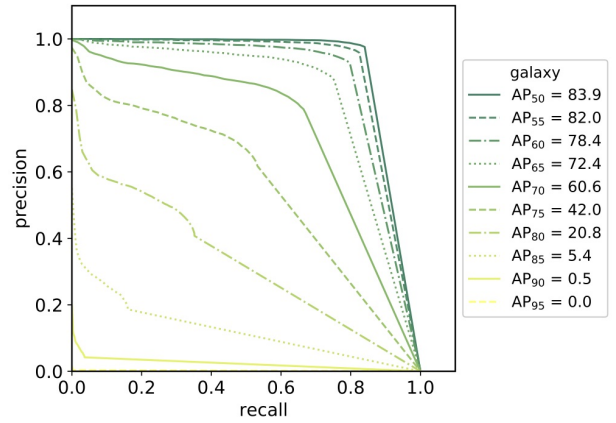


Figure 9. Precision-recall curves and mean AP scores calculated at varying IOU thresholds averaged over the test dataset for galaxies. We find IOU=0.5 gives a mean AP score of 83.9.

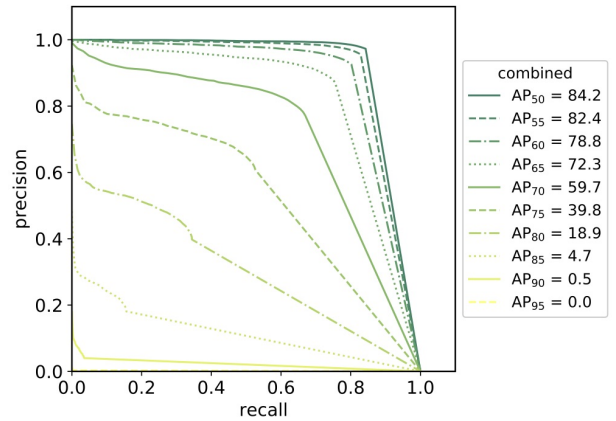


Figure 10. Precision-recall curves and mean AP scores calculated at varying IOU thresholds averaged over the test dataset calculated jointly for both stars and galaxies. We find IOU=0.5 gives a mean AP score of 84.2.

We adopt a metric commonly used in the machine learning community called the average precision (AP) score² (see [Everingham et al. 2010](#)). The AP score is simply the area under the precision-recall curve, averaged for each object in an image at discrete recall levels:

$$\text{AP} = \frac{1}{51} \sum_{r \in \{0, 0.02, \dots, 1.0\}} p(r) \quad (5)$$

where $p(r)$ is maximum the precision in bin Δr . We average the precision-recall curves and mean AP scores for all 50 images in the test dataset. This procedure is then done for the intersection over union (IOU) thresholds $\text{IOU} \in \{0.5, 0.55, \dots, 0.95\}$. The IOU is defined as the area of the intersection of the predicted and ground truth masks divided by the area of the union of the predicted and ground truth masks,

$$\text{IOU} = \frac{\text{area}(\text{mask}_{\text{predicted}} \cap \text{mask}_{\text{truth}})}{\text{area}(\text{mask}_{\text{predicted}} \cup \text{mask}_{\text{truth}})} \quad (6)$$

An RoI detection is considered positive if its IOU is greater than the IOU threshold. We therefore expect fewer positive detections at greater IOU thresholds. We show an example of detection masks overlayed with ground truth masks in Fig. 7 in a simulated test image. The results shown in Figs. 8, 9, and 10 summarize the performance of our network against PhoSim ground truth images in the test dataset. We also show the confusion matrix at IOU thresholds of 0.5 (Fig. 2) and 0.75 (Fig. 2).

In Table 1, we calculate various AP metrics to evaluate the performance of our work for stars, galaxies, and a joint star+galaxy (combined) evaluation. We use the standard AP score variants used in the computer vision community defined as follows:

- AP: AP score averaged over IOU thresholds of 0.5 to 0.95.
- AP₅₀: AP score at IOU threshold of 0.5 (50%).
- AP₇₅: AP score at IOU threshold of 0.75 (75%).
- AP_S: AP score of small sized ($\text{area} < 16^2 \text{ pixel}^2$) objects bounding box at IOU threshold of 0.5.
- AP_M: AP score of medium sized ($16^2 < \text{area} < 32^2 \text{ pixel}^2$) objects bounding box at IOU threshold of 0.5.
- AP_L: AP score of large sized ($\text{area} > 32^2 \text{ pixel}^2$) objects at IOU bounding box threshold of 0.5.

These metrics show how our Mask R-CNN implementation performs across mask accuracy (by varying the IOU threshold) and scales (by evaluating for different bounding box sizes), as well as facilitates comparison to other works. Note, that our definitions of AP_S, AP_M, and AP_L differ from those used by the MS COCO evaluation, because astronomical objects are generally much smaller in area in pixels.

Class	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
star	48.6	86.8	32.3	86.8	—	—
galaxy	49.6	83.9	42.0	75.3	7.6	0.3
combined	49.0	84.2	39.8	76.4	7.0	0.3

Table 1. Summary table of our AP score metrics calculated on the test dataset. Note their definitions in the text. Although our network performs well for small sources, its performance rapidly decreases for larger sources. This is likely due to the scarcity of sources with $\text{area} > 16^2 \text{ pixel}^2$ in the training images. All stars have bounding box sizes $\text{area} < 16^2 \text{ pixel}^2$.

		Predicted	
		star	galaxy
Truth	star	585	39
	galaxy	78	6302

Table 2. Confusion matrix calculated on the test dataset for stars/galaxies at an IOU threshold of 0.5.

		Predicted	
		star	galaxy
Truth	star	330	68
	galaxy	333	6273

Table 3. Confusion matrix calculated on the test dataset for stars/galaxies at an IOU threshold of 0.75.

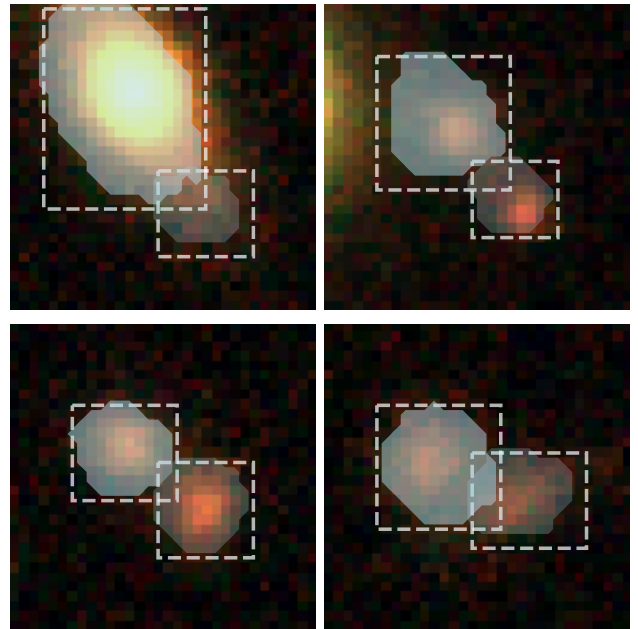


Figure 11. Examples of close blends identified as separate galaxies in a real DECaLS image using our deep learning technique. Bounding box edges are shown as white dashed lines, and masks are shown as transparent white regions. An additional S/N or detection confidence cutoff may be applied afterwards to remove false or spurious detections.

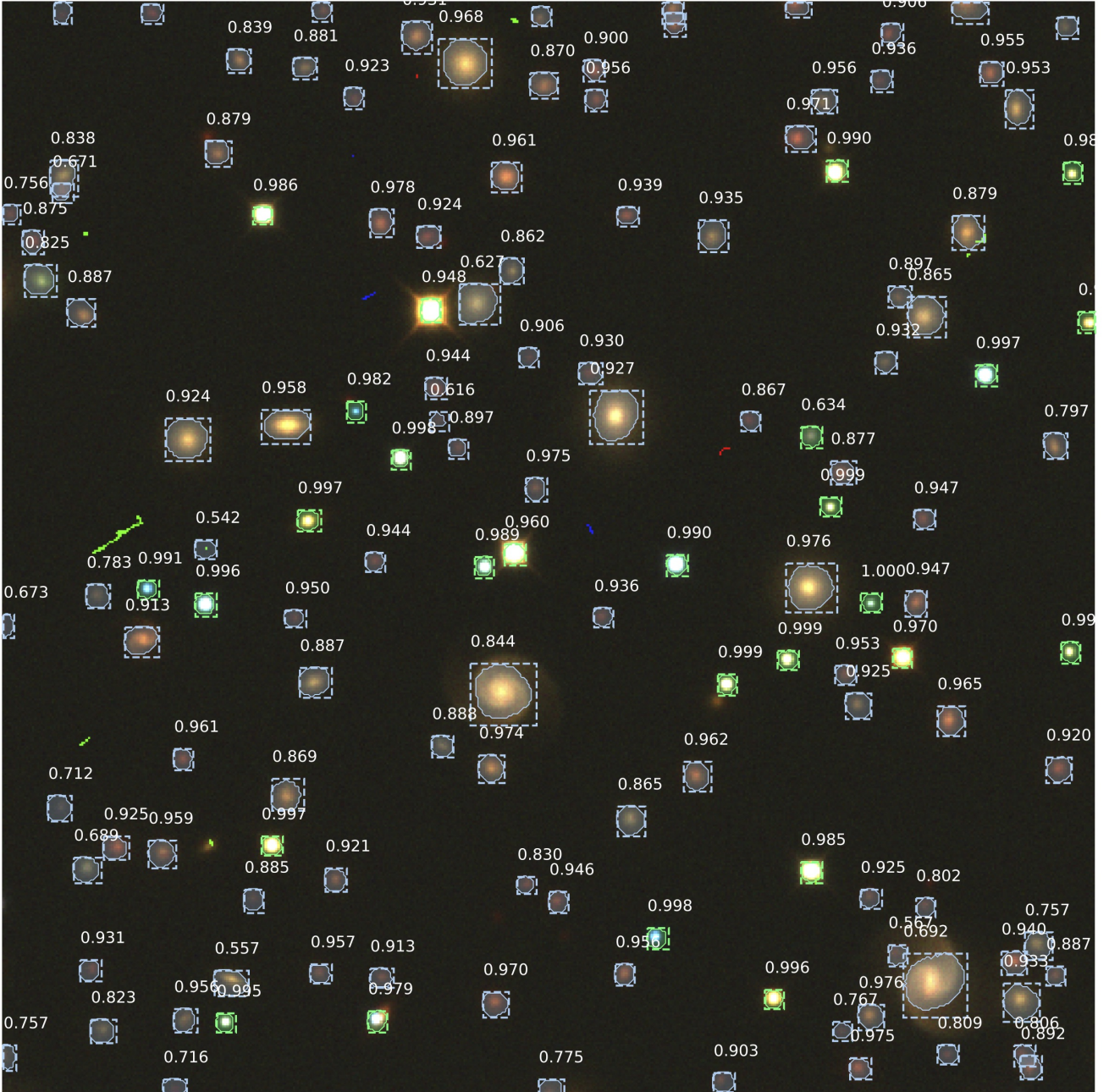


Figure 12. Typical example of detection inference in a full test dataset simulated image. Although this image is not used in the training, it is a typical example of what the training images look like (generated using the same PhoSim settings and procedure). Galaxy masks are shown in light blue and star masks are shown in green. The detection confidence the object belongs to that particular class is shown above each mask. Note several blends are successfully identified as separate objects, particularly at the lower-right.

3.2 Deblending

To test the deblending capability of our code, we simulate a PhoSim image with a high number density of galaxies to

² The AP score has largely superseded the older area under the receiver operating characteristic (ROC) curve in the computer vision community in favor of the AP score's greater sensitivity for high-performing networks.

mimic the DECaLS images. One relevant inference configuration parameter is `DETECTION_NMS_THRESHOLD`. This parameter sets the intersection threshold for non-maximum suppression (NMS). Mask R-CNN and other sliding window -based approaches typically result in several high-confidence detections for individual objects. NMS rejects low-confidence bounding boxes that have high IOU overlaps with another bounding box. We use an IOU threshold of 0.3 for typical images, although this threshold may be lowered

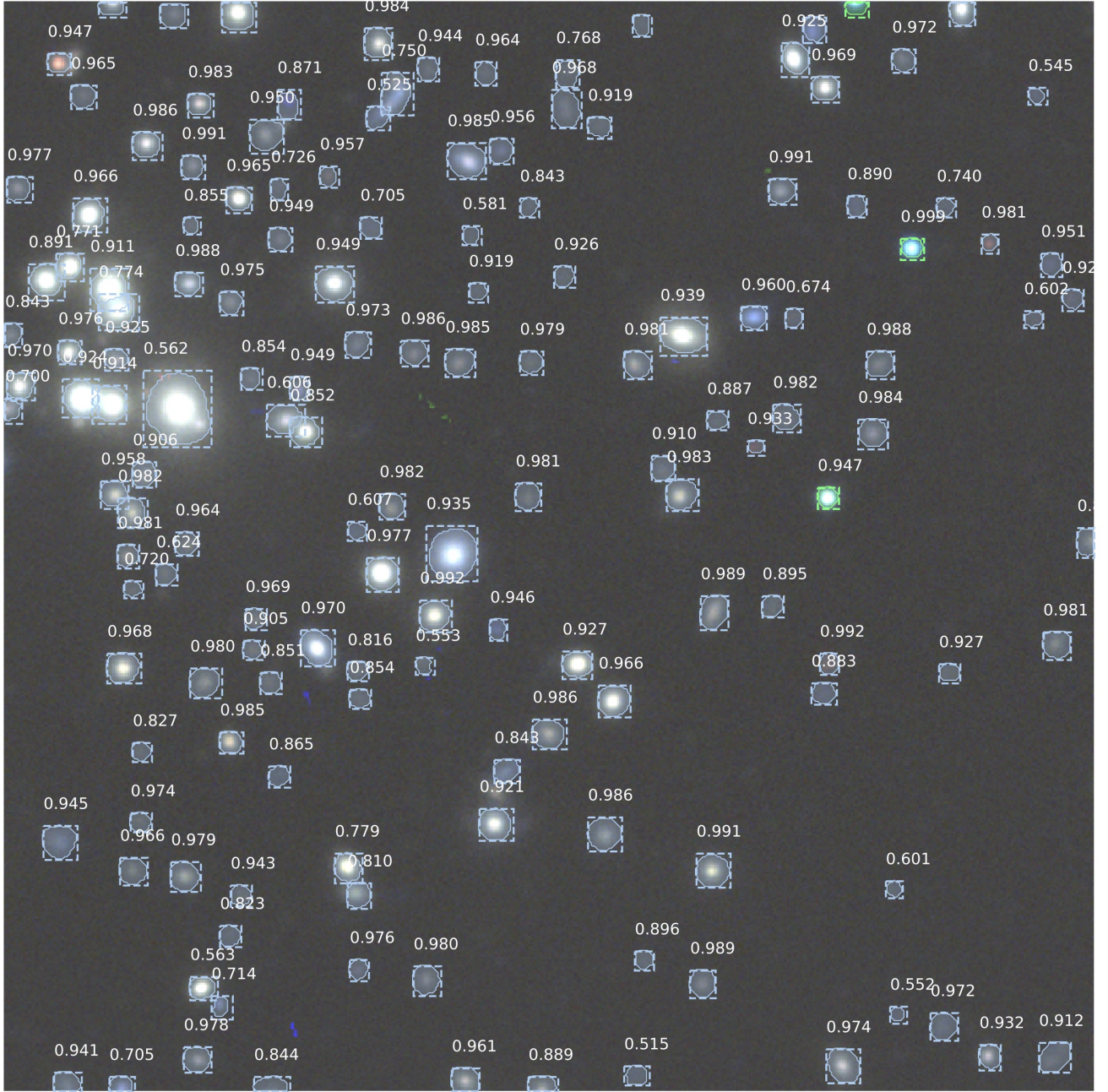


Figure 13. Example of detection inference in a real DECaLS image of ACO 1689. Galaxy masks are shown in light blue and star masks are shown in green. The detection confidence the object belongs to that particular class is shown above each mask. The central nucleus of the cluster is located in the upper-left of the image.

in very dense regions, such as the dense central nucleus of a cluster of galaxies, to increase the likelihood that close blends are identified as distinct objects.

In Fig. 11, we demonstrate several examples of deblending performed using our Mask R-CNN technique. The network recognizes close blends and generates segmentation maps that encompass each object. Additional post-processing may be done in some scenarios, such as a S/N cutoff or a smoothing refinement of the masks. Presently, we allow some pixels to overlap with multiple objects after

NMS is applied. Ultimately, the pixels that lie inside each masked region can simply be used to isolate objects.

3.3 Inference on real images

To investigate the performance of our network on real images, we use the public data release 7 DECaLS coadds avail-

able on the NOAO science archive³. We wish to assess our network in the limit of a crowded extragalactic field. Therefore, we use images centered on clusters of galaxies in the Abell Catalog (Olowin 1988; Abell et al. 1989). Specifically, we use DECaLS images of ACO 1689. ACO 1689 is a rich (class 4; Abell et al. 1989) cluster with a type II-III Bautz-Morgan classification (Bautz & Morgan 1970) at $z = 0.183$. ACO 1689’s richness and the fact that it has been extensively studied (e.g. Taylor et al. 1998; Tyson & Fischer 1995) and imaged by DECam makes it a good target for our tests. We show the result of a detection/inference on a simulated image (Fig. 12) and a real image (Fig. 13) for comparison.

We note that the PSF appears significantly larger in the real images compared to the simulated images. This may be likely due to choosing idealized simulated images that excludes some atmospheric or optical perturbations that enlarge the PSF. Another contributing factor is the DECaLS images are coadds taken in varying seeing conditions, which may result in a larger PSF. Despite this, our neural network is not overly-sensitive to the PSF size because we employ a Gaussian blur augmentation. However, very bright stars which have significant blooming can result in a badly-fitting mask or misclassification as a galaxy and a low IOU score for that detection.

3.4 Ultra-Deep Fields

We investigate how the richness (source density) of the field affects our networks performance. In general, a deeper and more crowded field is likely to contain a larger number of overlapping objects. Following the procedure outlined in §2.2, we simulate additional images with PhoSim using an exposure time of 30 hours with galaxies as faint as $g = 28$. This deep field mimicks the depth of the Subaru HSC ultra-deep field or ten-year LSST coadds, containing ~ 750 galaxies/arcmin². For these simulations, we turn off charge saturation and cosmic rays in PhoSim. The result is shown in (Fig. 14). We then calculate the mean AP score in this image at an IOU threshold of 0.5, following the same procedure outlined in §3.1. We find $AP = 52$ (stars) and $AP = 7.6$ (galaxies). Our network is not sensitive to the number of density of sources in the image, because the number of objects detected increases in proportion to the number in the simulations. However, galaxies with redshifted SEDs that are too faint to appear in the training images are not detected. We could mitigate this effect and increase the AP score by training on images with longer exposure times with higher-redshift galaxies or allowing a lower detection confidence threshold.

4 DISCUSSION

One of the main strengths of this technique is its speed and ease of use after training. After training is completed, we find individual images can be processed in 100 milliseconds or less depending on the GPU(s) used (not including time to read/write the results). This and similar machine learning

techniques have this important strength, which is particularly relevant for large astronomical surveys. In addition, our Mask R-CNN package performs all tasks of object detection, classification, and segmentation in parallel in a simple conceptual framework. Finally, our package is insensitive to configuration parameters, such as the density of sources in the field, PSF size, and exposure time. Instead, a large and diverse set of training images and additional data augmentations is needed.

Here we describe briefly the new and unique contributions of this work:

- Establish a novel technique for instance segmentation (masking) in astronomical images.
- Show how existing deep learning techniques in the field of computer vision can be used to solve difficult problems such as deblending in astronomy.
- Use of multi-band information to perform star/galaxy classification and deblending.
- Train a neural network on large, realistic simulated images approaching the size of whole CCDs.

There are several limitations to the machine learning techniques used in this work. First, it can be hard to correct for the systematic differences between simulated and real images. Issues arise if the real images do not mimic the simulated images. This can occur if the data in one band is missing or noisy (confusing the colors of objects). However, transfer learning could be used to perform a fine-tuning re-training for these scenarios. Occasionally, the neural network will confuse bright galaxy bulges for stars if the network has not been trained to a sufficient loss. This can be solved by training more extensively, employing non-maximum suppression to quickly reject these stars, or perhaps using simulated galaxies more brighter or more realistic bulges. Finally, it can be challenging to generate a realistic training set that accounts for the vast differences in images in different regions of the sky. For example, globular clusters, near the galactic center of the Milky Way, or images of large/extended galaxies may be challenging regimes. However, we expect that these problems can be mitigated if these types of objects were added to the simulated training data. Although care must be taken when simulating images of these ultra-dense regions. Because PhoSim works by generating a catalog from a three-dimensional space, we must not train for detections on sources that are completely obscured by another object.

In the future, we will explore masking additional features such as cosmic rays and bleed trails. These distinct features should be straightforward for Mask R-CNN to learn. We may also explore different network architectures for instance segmentation, such as the popular U-Net (Ronneberger et al. 2015). U-Net may be better-suited to full-scale astronomical images because it does not create regions around each object, which can lead to inefficiencies in the training. The down-side is that U-Net can only inherently perform semantic segmentation (i.e., not distinguishing masks of individual objects of the same class). Therefore, these methods must eventually fall-back on traditional techniques to handle mask occlusion, and may suffer from the same issues current pipelines face in crowded fields.

Moreover, we will explore (a) executing and tiling results on a full-scale dataset with images taken under a variety of conditions, such as DES or HSC deep coadd images;

³ <http://archive.noao.edu>

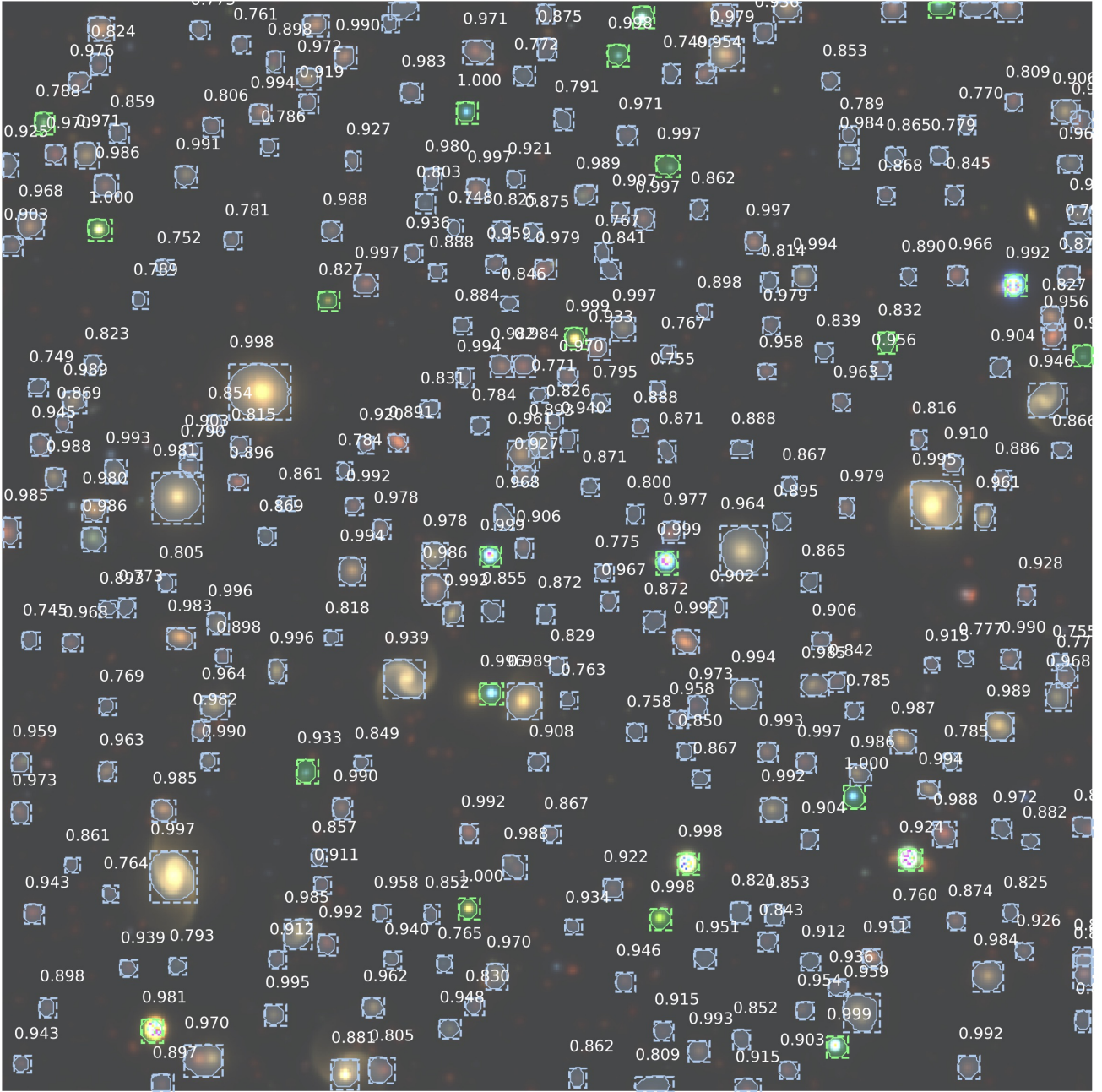


Figure 14. An example of an ultra-deep 30 hour simulation and detection inference. Galaxy masks are shown in light blue and star masks are shown in green. The detection confidence the object belongs to that particular class is shown above each mask. Many of the fainter and higher redshift galaxies are not detected with a confidence threshold of 0.5 or higher because they appear redder than the galaxies in our training data. However, nearly all of more typical sources are detected.

(b) predicting probability contours for objects by training using different mask sizes or IOU cutoffs, add photometric redshift prediction, adding additional galaxy classes in place of the generic “galaxy” classification, such as spiral, elliptical, irregular, and lensed; (c) implementing a more robust interface for training and configurations. Mask R-CNN can also be used on video for real-time instance segmentation, which may have interesting applications for LSST and time-domain astronomy.

There may also be additional ways to optimize our Mask R-CNN implementation (both training and detection) on astronomical image data. For example, it is possible that different contrast stretching such as [Lupton et al. \(2004\)](#) (see [González et al. 2018](#)) or backbone architecture will produce better results on astronomical images. The standard techniques adopted in this work (linear scaling with the ResNet-101 backbone) used on terrestrial images in the MS COCO dataset may not be as optimal for astronomical data which

typically have smaller objects at lower S/N . Based on our evaluation, using training images which are *less* realistic, but include a larger variety of objects and with a more uniform distribution of sizes may be appropriate.

Because our network works well on real images, this work also acts as a unique and real-world validation of PhoSim itself. Several additional telescopes/instruments are implemented in PhoSim, including LSST and Subaru HSC. Therefore, extensions of this work to other telescopes would be straightforward. We would welcome further extensions of the *Astro R-CNN* code repository (Burke et al. 2019) to improve the training, performance, and usability. Transfer learning from this work to other telescopes or surveys could also be used to shorten training time and with little required GPU resources.

5 CONCLUSIONS

In this work, we develop a new deep-learning method for classifying and deblending sources in astronomical images. Using PhoSim, we generate simulated images and catalogs used as a ground truth comparison for supervised machine learning. Our code, *Astro R-CNN* efficiently performs all tasks of source detection, classification, and deblending in one package. The network is robust to object overlaps and mask occlusion, resulting in clean deblends for significantly blended sources.

We evaluate the performance of our network using the AP score metric, and show the precision and recall curves for stars and galaxies. Our network performs well at moderate IOU thresholds. We measure a precision of 92% at 80% recall for stars and a precision of 98% at 80% recall for galaxies in a typical field with 30 galaxies/arcmin² with an IOU threshold of 0.5 and a minimum detection confidence threshold of 0.5. We show multiple examples of deblending in real and simulated images, including in the central nucleus of an Abell cluster of galaxies. Because of the region-based nature of the Mask R-CNN neural network, the results are insensitive to the density of sources in the image therefore may be used at different galactic latitudes. If care is taken to configure and train *Astro R-CNN* properly, it may be used on a variety of real images. The depth and quantity of images in current and future deep optical surveys demands efficient and robust techniques such as this one. We suggest future endeavors use this work as an example to apply even newer and rapidly-advancing techniques from the computer vision community to astronomy.

ACKNOWLEDGEMENTS

This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign. We thank Dr. Volodymyr Kindratenko and Dr. Dawei Mu at the National Center for Supercomputing Applications for their assistance with the GPU cluster used in this work. This work is based on projects started during the 2019 graduate-level AI seminar at the Department of Astronomy, University of Illinois at Urbana-Champaign. We thank the anonymous referees for helpful comments.

We acknowledge use of Matplotlib (Hunter 2007), a community-developed Python library for plotting. This research made use of Astropy,⁴ a community-developed core Python package for Astronomy (Astropy Collaboration et al. 2013; Price-Whelan et al. 2018). This research has made use of NASA's Astrophysics Data System.

NOAO is operated by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the National Science Foundation.

This project used data obtained with the Dark Energy Camera (DECam), which was constructed by the Dark Energy Survey (DES) collaboration. Funding for the DES Projects has been provided by the U.S. Department of Energy, the U.S. National Science Foundation, the Ministry of Science and Education of Spain, the Science and Technology Facilities Council of the United Kingdom, the Higher Education Funding Council for England, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, the Kavli Institute of Cosmological Physics at the University of Chicago, Center for Cosmology and Astro-Particle Physics at the Ohio State University, the Mitchell Institute for Fundamental Physics and Astronomy at Texas A&M University, Financiadora de Estudos e Projetos, Fundacao Carlos Chagas Filho de Amparo, Financiadora de Estudos e Projetos, Fundacao Carlos Chagas Filho de Amparo a Pesquisa do Estado do Rio de Janeiro, Conselho Nacional de Desenvolvimento Cientifico e Tecnolico and the Ministerio da Ciencia, Tecnologia e Inovacao, the Deutsche Forschungsgemeinschaft and the Collaborating Institutions in the Dark Energy Survey. The Collaborating Institutions are Argonne National Laboratory, the University of California at Santa Cruz, the University of Cambridge, Centro de Investigaciones Energeticas, Medioambientales y Tecnologicas-Madrid, the University of Chicago, University College London, the DES-Brazil Consortium, the University of Edinburgh, the Eidgenossische Technische Hochschule (ETH) Zurich, Fermi National Accelerator Laboratory, the University of Illinois at Urbana-Champaign, the Institut de Ciencies de l'Espai (IEEC/CSIC), the Institut de Fisica d'Altes Energies, Lawrence Berkeley National Laboratory, the Ludwig-Maximilians Universitat Munchen and the associated Excellence Cluster Universe, the University of Michigan, the National Optical Astronomy Observatory, the University of Nottingham, the Ohio State University, the University of Pennsylvania, the University of Portsmouth, SLAC National Accelerator Laboratory, Stanford University, the University of Sussex, and Texas A&M University.

REFERENCES

- Abadi M., et al., 2016, in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp 265–283
- Abbott T. M. C., et al., 2018, *ApJS*, **239**, 18
- Abdulla W., 2017, Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, https://github.com/matterport/Mask_RCNN
- Abell G. O., Corwin Jr. H. G., Olowin R. P., 1989, *ApJS*, **70**, 1
- Aihara H., et al., 2018, *PASJ*, **70**, S8
- Aihara H., et al., 2019, arXiv e-prints, p. arXiv:1905.12221

⁴ <http://www.astropy.org>

- Allende Prieto C., Barklem P. S., Lambert D. L., Cunha K., 2004, *A&A*, **420**, 183
- Amiaux J., et al., 2012, Euclid Mission: building of a reference survey, doi:10.1117/12.926513, <https://doi.org/10.1117/12.926513>
- Andreon S., Gargiulo G., Longo G., Tagliaferri R., Capuano N., 2000, *MNRAS*, **319**, 700
- Arneson R. A., 2013, Master's thesis, University of California, Irvine
- Astropy Collaboration et al., 2013, *A&A*, **558**, A33
- Bautz L. P., Morgan W. W., 1970, *ApJ*, **162**, L149
- Bengio Y., Simard P., Frasconi P., 1994, *Trans. Neur. Netw.*, **5**, 157
- Bertin E., Arnouts S., 1996, *A&AS*, **117**, 393
- Bolya D., Zhou C., Xiao F., Lee Y. J., 2019, arXiv e-prints, p. [arXiv:1904.02689](https://arxiv.org/abs/1904.02689)
- Boucaud A., et al., 2019, arXiv e-prints, p. [arXiv:1905.01324](https://arxiv.org/abs/1905.01324)
- Burke C. J., Aleo P. D., Chen Y.-C., 2019, Astro R-CNN: Instance Segmentation in Astronomical Images using Mask R-CNN Deep Learning, https://github.com/burke86/astro_rcnn
- Castelli F., Kurucz R. L., 2003, in Piskunov N., Weiss W. W., Gray D. F., eds, IAU Symposium Vol. 210, Modelling of Stellar Atmospheres. p. A20 ([arXiv:astro-ph/0405087](https://arxiv.org/abs/astro-ph/0405087))
- Chang C., et al., 2013, *MNRAS*, **434**, 2121
- Cheng J., 2017, PhD thesis, Purdue University
- Cheng T.-Y., et al., 2019, arXiv e-prints, p. [arXiv:1908.03610](https://arxiv.org/abs/1908.03610)
- Chollet F., et al., 2015, Keras, <https://keras.io>
- Couprie M., Bertrand G., 1997, in Melter R. A., Wu A. Y., Latecki L. J., eds, Proc. SPIE Vol. 3168, Vision Geometry VI. pp 136–146, doi:10.1117/12.292778
- D'Isanto A., Polsterer K. L., 2018, *A&A*, **609**, A111
- Dawson W., Schneider M., 2014, Complementarity of LSST and WFIRST: Regarding Object Blending
- Dawson W. A., Schneider M. D., Tyson J. A., Jee M. J., 2016, *ApJ*, **816**, 11
- Dey A., et al., 2019, *AJ*, **157**, 168
- Dieleman S., Willett K., Dambre J., 2015, *MNRAS*, **450**, 1441
- Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A., 2010, *International Journal of Computer Vision*, **88**, 303
- Flaugher B., et al., 2015, *AJ*, **150**, 150
- Gallazzi A., Charlot S., Brinchmann J., White S. D. M., Tremonti C. A., 2005, *MNRAS*, **362**, 41
- Girshick R., 2015, in 2015 IEEE International Conference on Computer Vision (ICCV). pp 1440–1448, doi:10.1109/ICCV.2015.169
- González R. E., Muñoz R. P., Hernández C. A., 2018, *Astronomy and Computing*, **25**, 103
- Hausen R., Robertson B., 2019, arXiv e-prints, p. [arXiv:1906.11248](https://arxiv.org/abs/1906.11248)
- He K., Zhang X., Ren S., Sun J., 2016, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778
- He K., Gkioxari G., Dollár P., Girshick R. B., 2017, 2017 IEEE International Conference on Computer Vision (ICCV), pp 2980–2988
- Huang G., Sun Y., Liu Z., Sedra D., Weinberger K., 2016. pp 646–661, doi:10.1007/978-3-319-46493-0_39
- Huang S., et al., 2018, *PASJ*, **70**, S6
- Huang Z., Huang L., Gong Y., Huang C., Wang X., 2019, arXiv e-prints, p. [arXiv:1903.00241](https://arxiv.org/abs/1903.00241)
- Hunter J. D., 2007, *Computing in Science & Engineering*, **9**, 90
- Iglovikov V., Shvets A., 2018, arXiv e-prints, p. [arXiv:1801.05746](https://arxiv.org/abs/1801.05746)
- Ivezić Ž., et al., 2019, *ApJ*, **873**, 111
- Jarvis J. F., Tyson J. A., 1981, *AJ*, **86**, 476
- Kim E. J., Brunner R. J., 2017, *MNRAS*, **464**, 4463
- Kim E. J., Brunner R. J., Carrasco Kind M., 2015, *MNRAS*, **453**, 507
- Kingma D. P., Ba J., 2014, arXiv e-prints, p. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Pereira F., Burges C. J. C., Bottou L., Weinberger K. Q., eds, , Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp 1097–1105
- Kroupa P., 2001, *MNRAS*, **322**, 231
- Kurucz R., 1993, ATLAS9 Stellar Atmosphere Programs and 2 km/s grid. Kurucz CD-ROM No. 13. Cambridge, Mass.: Smithsonian Astrophysical Observatory, 1993., **13**
- Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C. L., 2014, in European Conference on Computer Vision (ECCV). Zürich
- Lin T.-Y., Dollár P., Girshick R. B., He K., Hariharan B., Belongie S. J., 2017, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 936–944
- Liu S., Qi L., Qin H., Shi J., Jia J., 2018, arXiv e-prints, p. [arXiv:1803.01534](https://arxiv.org/abs/1803.01534)
- Lupton R., 2014, Joint Source Detection, Deblending, and Measurement for WFIRST-AFTA and LSST, NASA WPS Proposal
- Lupton R., Blanton M. R., Fekete G., Hogg D. W., O'Mullane W., Szalay A., Wherry N., 2004, *PASP*, **116**, 133
- Madau P., Dickinson M., 2014, *ARA&A*, **52**, 415
- Melchior P., Moolekamp F., Jerdee M., Armstrong R., Sun A. L., Bosch J., Lupton R., 2018, *Astronomy and Computing*, **24**, 129
- Messier C., 1781, Technical report, Catalogue des Nébuleuses et des Amas d'Étoiles (Catalog of Nebulae and Star Clusters)
- Mollá M., García-Vargas M. L., Bressan A., 2009, *MNRAS*, **398**, 451
- Olowin R. P., 1988, *PASP*, **100**, 1354
- Pasquet J., Bertin E., Treyer M., Arnouts S., Fouchez D., 2019, *A&A*, **621**, A26
- Peletier R. F., Balcells M., 1996, *AJ*, **111**, 2238
- Pence W. D., Chiappetti L., Page C. G., Shaw R. A., Stobie E., 2010, *A&A*, **524**, A42
- Peterson J. R., et al., 2015, *ApJS*, **218**, 14
- Price-Whelan A. M., et al., 2018, *AJ*, **156**, 123
- Prugniel P., Vauglin I., Koleva M., 2011, *A&A*, **531**, A165
- Redmon J., Divvala S., Girshick R., Farhadi A., 2015, arXiv e-prints, p. [arXiv:1506.02640](https://arxiv.org/abs/1506.02640)
- Reiman D. M., Göhre B. E., 2019, *MNRAS*, **485**, 2617
- Ren S., He K., Girshick R., Sun J., 2015, in Cortes C., Lawrence N. D., Lee D. D., Sugiyama M., Garnett R., eds, , Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp 91–99
- Ronneberger O., Fischer P., Brox T., 2015, in MICCAI.
- Sebok W. L., 1979, *AJ*, **84**, 1526
- Serra-Ricart M., Gaitan V., Garrido L., Perez-Fournon I., 1996, *A&AS*, **115**, 195
- Sérsic J. L., 1963, Boletín de la Asociacion Argentina de Astronomia La Plata Argentina, **6**, 41
- Sevilla-Noarbe I., et al., 2018, *MNRAS*, **481**, 5451
- Soumagnac M. T., et al., 2015, *MNRAS*, **450**, 666
- Spergel D., et al., 2013, arXiv e-prints, p. [arXiv:1305.5422](https://arxiv.org/abs/1305.5422)
- Tan C., Sun F., Kong T., Zhang W., Yang C., Liu C., 2018, A Survey on Deep Transfer Learning: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III. pp 270–279, doi:10.1007/978-3-030-01424-7_27
- Taylor A. N., Dye S., Broadhurst T. J., Benítez N., van Kampen E., 1998, *ApJ*, **501**, 539
- Tsai H.-F., Gajda J., Sloan T. F., Rares A., Shen A. Q., 2019, *SoftwareX*, **9**, 230
- Tyson J. A., Fischer P., 1995, *ApJ*, **446**, L55
- Valdes F., 1982, in Instrumentation in Astronomy IV. pp 465–472, doi:10.1117/12.933489
- Vasconcellos E. C., de Carvalho R. R., Gal R. R., LaBarbera F. L.,

Capelato H. V., Frago Campos Velho H., Trevisan M., Ruiz R. S. R., 2011, *AJ*, **141**, 189
 Zhang K., Bloom J. S., 2019, arXiv e-prints, p. [arXiv:1907.09500](https://arxiv.org/abs/1907.09500)
 Zhang Y., McKay T. A., Bertin E., Jeltima T., Miller C. J., Rykoff E., Song J., 2015, *PASP*, **127**, 1183
 Zhang W., Witharana C., Liljedahl A. K., Kanevskiy M., 2018, *Remote Sensing*, **10**
 Zimmermann R. S., Siems J. N., 2018, arXiv e-prints, p. [arXiv:1809.07069](https://arxiv.org/abs/1809.07069)

APPENDIX A: LOSS CURVES

We show the training loss versus epoch following the learning schedule described in §2.6 in Figures A1, A2, A3, A4. The total loss L is defined as the sum of the class, bounding box, and mask losses: $L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}$ (He et al. 2017). We stop training at 50 epochs. After ~50 epochs, the asymptotic loss curves show diminishing returns. We note that the validation loss is consistent with the training loss. Scheduling the learning rate to progressively decrease during the training enables refinement of the network performance without overfitting on the training set.

This paper has been typeset from a $\text{\LaTeX}$ file prepared by the author.

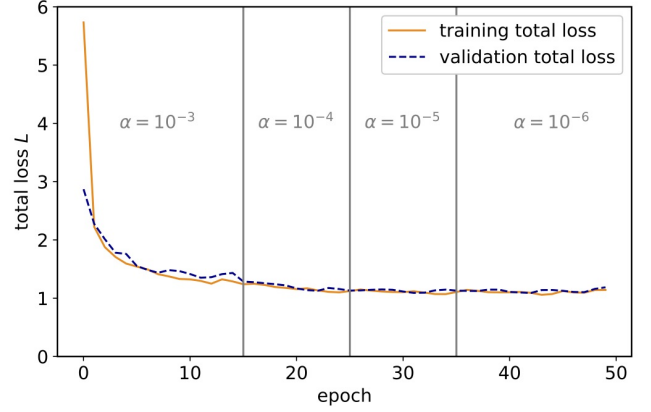


Figure A1. Mask R-CNN total loss versus training epoch.

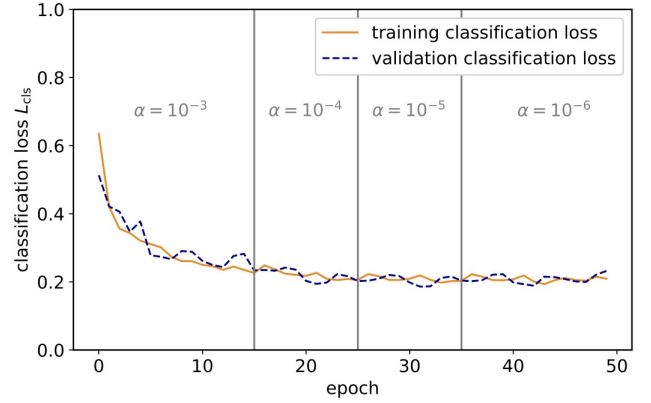


Figure A2. Mask R-CNN class loss versus training epoch.

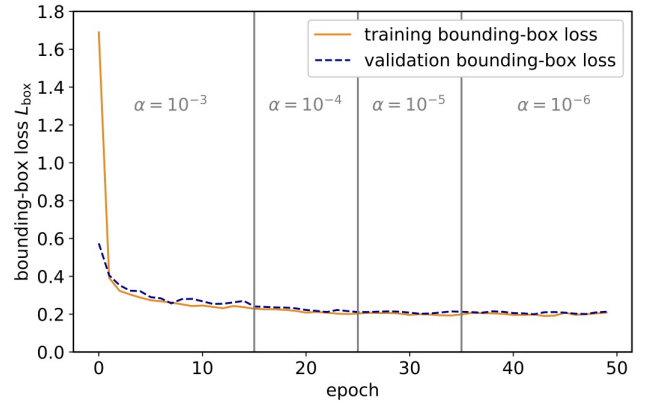


Figure A3. Mask R-CNN bounding box loss versus training epoch.

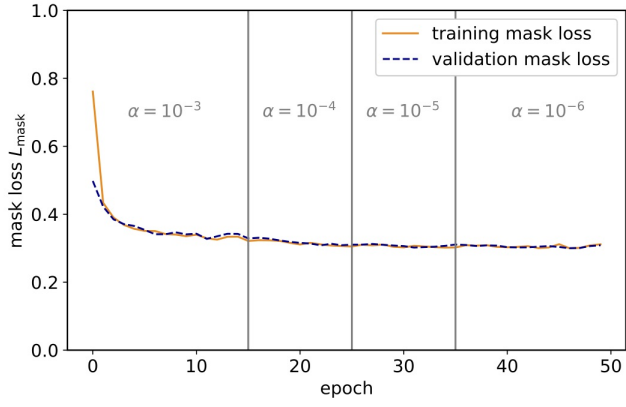


Figure A4. Mask R-CNN mask loss versus training epoch.