The Astrophysical Journal, 877:58 (19pp), 2019 May 20 https://doi.org/10.3847/1538-4357/ab16d9

Automated Lensing Learner: Automated Strong Lensing Identification with a Computer Vision Technique

Camille Avestruz[1,2], Nan Li[2,3,4,5], Hanjue Zhu (朱涵珏)[6], Matthew Lightman[7], Thomas E. Collett[8], and Wentao Luo[9]

[1] Enrico Fermi Institute, The University of Chicago, Chicago, IL 60637, USA; avestruz@uchicago.edu
[2] Kavli Institute for Cosmological Physics, The University of Chicago, Chicago, IL 60637, USA
[3] Department of Astronomy & Astrophysics, The University of Chicago, Chicago, IL 60637, USA
[4] High Energy Physics Division, Argonne National Laboratory, Lemont, IL 60439, USA
[5] School of Physics and Astronomy, University of Nottingham, University Park, Nottingham, NG7 2RD, UK; nan.li@nottingham.ac.uk
[6] The University of Chicago, Chicago, IL 60637, USA
[7] JPMorgan Chase, Chicago, IL 60603, USA
[8] Institute of Cosmology and Gravitation, University of Portsmouth, Burnaby Road, Portsmouth, PO1 3FX, UK
[9] Center for Astronomy and Astrophysics, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, People's Republic of China

Received 2017 April 7; revised 2019 March 18; accepted 2019 April 4; published 2019 May 24

Abstract

Forthcoming surveys such as the Large Synoptic Survey Telescope (LSST) and Euclid necessitate automatic and efficient identification methods of strong lensing systems. We present a strong lensing identification approach that utilizes a feature extraction method from computer vision, the Histogram of Oriented Gradients (HOG), to capture edge patterns of arcs. We train a supervised classifier model on the HOG of mock strong galaxy–galaxy lens images similar to observations from the *Hubble Space Telescope* (*HST*) and LSST. We assess model performance with the area under the curve (AUC) of a Receiver Operating Characteristic (ROC) curve. Models trained on 10,000 lens and non-lens containing images exhibit an AUC of 0.975 for an *HST*-like sample, 0.625 for one exposure of LSST, and 0.809 for 10 yr mock LSST observations. Performance appears to continually improve with the training set size. Models trained on fewer images perform better in the absence of the lens galaxy light. However, with larger training data sets, information from the lens galaxy actually improves model performance, indicating that HOG captures much of the morphological complexity of the arc-finding problem. We test our classifier on data from the Sloan Lens ACS Survey and find that small-scale image features reduce the efficiency of our trained model. However, these preliminary tests indicate that some parameterizations of HOG can compensate for differences between observed mock data. One example best-case parameterization results in an AUC of 0.6 in the F814 filter image, with other parameterization results equivalent to random performance.

Key words: galaxies: elliptical and lenticular, cD – gravitational lensing: strong – methods: data analysis – methods: numerical – methods: statistical – surveys

1. Introduction

Gravitational lensing occurs when intermediate fluctuations in the matter density field deflect light from background sources (see Kneib & Natarajan 2011, for a review). Strong gravitational lensing can manifest as visible giant arcs magnifying high-redshift galaxies (Lynds & Petrosian 1986; Gladders et al. 2003), multiply imaged quasars (Walsh et al. 1979), multiply imaged galaxies (Sharon et al. 2005), and arclets (Bezecourt et al. 1998). Lensing signatures probe the underlying dark matter distribution of the lens (Warren & Dye 2003), and high-redshift galaxy formation (Allam et al. 2007). Strong lenses also provide a geometric test of cosmology via comparison of predicted arc abundances with observed abundances (Kochanek 1996; Chae 2003; Linder 2004), via time-delay between signals from multiply imaged quasars (Suyu et al. 2014, 2017; Bonvin et al. 2017), and via distance ratios in lenses with sources at multiple redshifts (Jullo et al. 2010; Collett et al. 2012; Collett & Auger 2014).

The application of strong gravitational lensing to constrain the mass distribution of strong lenses, such as early-type galaxies (ETGs), necessitates large samples of galaxy–galaxy strong lensing systems. Miralda-Escude & Lehar (1992) first suggested that massive ellipticals would likely be frequent strong lensing sources in optical surveys. These systems contain a background source galaxy that the lens galaxy deflects into a partial or full arc shaped Einstein ring. The strong lensing signature directly probes the underlying matter. The identification of such systems in upcoming surveys is the first step in constraining the mass-to-light ratio for a large number of objects in this mass range.

Over the last decade, infrastructure for both large scale visual and automated image classification emerged. By nature, the human eye is one of the best discriminators for image classification. Visual arc identification has been effective through the use of citizen science platforms. *SpaceWarps* is an example of citizen science based image classification of strong lensing systems in Canada–France–Hawaii Telescope Science (CFHTLS) telescope observations (Marshall et al. 2016; More et al. 2016). These platforms are quite successful for a data set like CFHTLS; here, 3000 candidate images were identified in eight months, resulting in 89 final candidates. However, future data sets like Euclid (Oguri & Marshall 2010) and LSST expect to find hundreds of thousands of galaxy-scale strong lenses (Collett 2015). The volume of upcoming data challenges the scalability of a pure citizen science approach.

Recent efforts have focused on the development of automated methods with performance comparable to or better than humans. *SpaceWarps* is a part of the Zooniverse Project (Marshall et al. 2016), which also includes *Galaxy Zoo*, the citizen science-based image classification of galaxy types (Lintott et al. 2008). Galaxy classification is an early example in which machine learning algorithms successfully trained models to classify astronomical images with comparable performance to humans (Dieleman et al. 2015).

Earlier efforts on automated, or "robot," identification of strong lensing systems have two distinct generalized steps. The first enhances and extracts characteristic features, and the second uses some form of pattern recognition in the features to classify lens and non-lens containing systems. Among others, selected features might include shape parameterization (see Alard 2006; Kubo & Dell'Antonio 2008; Xu et al. 2016; Lee 2017), colors in multi-band imaging (Gavazzi et al. 2014; Maturi et al. 2014), light profiles (Brault & Gavazzi 2015), and characteristics of potential lens galaxies (Marshall et al. 2009). Pattern recognition often incorporates cutoffs in selected parameter space distributions (Lenzen et al. 2004; Joseph et al. 2014; Paraficz et al. 2016). A number of these have been publicly distributed, with specific end applications. For example, *ArcFinder* is one such code that finds arcs in groups or cluster scale lens (Seidel & Bartelmann 2007), and *RingFinder* is an analogous tool in searching for multiply imaged quasars (Gavazzi et al. 2014). Codes like these have been complementary to human identification (More et al. 2016).

Pattern recognition methods have transitioned to using "machine learning" algorithms in place of manual cutoffs. With machine learning, we can train a model to separate a data set into a known set of classes, such as "lensing systems" and "non-lensing systems." However, many classic machine learning algorithms do not work well in image space, i.e., directly using the raw pixel values of the image, but first require a process of *feature engineering*. *Feature engineering* uses domain expertise to extract variables that are more directly related to the classification task at hand. An example would be the application of an edge detector. The optimal weights and cutoffs for these derived variables that are used to determine the class label of an image are found automatically by the algorithm.

Some more recent works have made use of a subset of machine learning algorithms called neural networks to classify images, either from derived image parameters or directly in image space. In Estrada et al. (2007), authors used derived shape parameters to train neural nets to identify arc candidates. Agnello et al. (2015) used neural networks trained on data from multi-band magnitudes, and Bom et al. (2017) used extracted morphological parameters.

The most advanced use of neural networks for strong lensing classification operate directly in image space. (Petrillo et al. 2017) used mock Kilo Degree Survey (KiDS) data to train convolutional neural networks, with a training set size of six million images. Lanusse et al. (2018) used state-of-the-art deep residual neural networks to also work directly in the image space with minimal image pre-processing. A major strength of the Lanusse et al. (2018) implementation is that in comparison to deep convolutional neural networks, deep residual neural networks have been found to be easier to train and perform better on simulated data (Metcalf et al. 2018).

A lens-finding challenge conducted in 2017 compared results of various lens-finding methods from several teams (Metcalf et al. 2018). The challenge included both ground- and space-based data, respectively, using mock and real KiDS data, and mock Euclid data. Using the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) curve (see Section 3.1 for more details) as a metric, the performance for ground-based-like data was as follows. Neural networks performed best with a top AUC = 0.98 (e.g., Jacobs et al. 2017; Lanusse et al. 2018), human inspection with an AUC = 0.89, and the method we present in this paper at AUC = 0.84. AUC values closer to 1 indicate better identification of lenses. It is worth noting, however, that the AUC worsened for all methods when evaluated on true ground-based data alone.

To date, the most significant challenge has been in translating lens-finding algorithms to perform well on observed data. Jacobs et al. (2017) provided the first example of a trained neural network successfully applied to data from the Canada–France–Hawaii Telescope Legacy Survey. The success in application to observations stems from the creation of training sets that incorporate real survey galaxies and real survey backgrounds.

While there has been a recent surge in the use of deep neural networks applied to image classification problems in astronomy, it is not always easy to scale these techniques to large data sets, nor are the necessary computational resources and hardware, such as graphical processing units (GPUs), easily accessible to the entire scientific community. We present a first application of the Histogram of Oriented Gradients (HOG) as a feature extraction method and classify strong lensing systems with a basic Logistic Regression algorithm (LR). HOG is a fast feature extraction procedure that quantifies edges in images, commonly used to identify humans in security software. The authors know of only one other use of HOG in astronomy, in the recent context of spectral line observations to characterize atomic and molecular gas (Soler et al. 2019). LR is a linear classifier, making its scalability relatively straightforward with existing open source tools. This paper also serves as both a presentation of the mock data set. We test the methods on mock *Hubble Space Telescope* (*HST*) and LSST data, which will also be made publicly available. Results are reproducible on personal computers, as both the pipeline and the data will be publicly distributed.

We show results of the pipeline on mock galaxy–galaxy lens systems observed by *HST* and LSST as respective examples of classifier performance on optical space- and ground-based observations. We train and test our pipeline on subsamples from 10,000 mock observed strong lensing systems and 10,000 non-strong lensing systems, each centered on a potential lens galaxy. We additionally assess model performance on observed data from The Sloan Lens ACS Survey (SLACS; Bolton et al. 2008). We discuss limitations of our methods in the context of what simulations are able to capture.

Our paper is organized as follows. In Section 2 we briefly describe the methods to generate the mock *HST* and LSST data and our overall image processing and classification pipeline. We present our results in Section 3, and our summary and discussions in Section 4.

2. Methodology

2.1. Mock Images

To train and test our model, we create mock observations using two different codes. We generate mock *Hubble Space Telescope*[10] (*HST*) with *PICS* (Pipeline for Images of Cosmological Strong lensing) (Li et al. 2016). From PICS, we

[10] https://www.spacetelescope.org/

also have simulated strong lenses without PSF and noise, on which we then apply *LensPop* (Collett 2015) to the *PICS* generated images to perform mock *Large Synoptic Survey Telescope*[11] (LSST) observations. Note that both codes are used because the mock observing module in *PICS* does not include LSST. We specifically use *LensPop* to implement the mock observing for LSST based on the simulated images created by PICS.

There are 10,000 lens containing mock observations and 10,000 non-lens containing mock observations for running our parameter search. We keep a holdout set of 1000 lens containing mock observations and 1000 non-lens containing mock observations on which we evaluate the final trained model.

Mock observations of lensing systems include the lens galaxy, lensed images of the source galaxy, and galaxies along the line of sight. Mock observations of non-lensing systems include all but the images of a lensed source galaxy. We convolve each of the 2 × 10,000 train/test images and 2 × 1000 holdout images to produce three separate sets of mock "observations." These each have equal numbers of lens and non-lens containing systems: (1) *HST*-like observations, (2) best single exposure LSST-like observations, and (3) LSST-like observations over the span of 10 years. We respectively label these observations as *HST*, LSST-best, and LSST10.

2.1.1. Modeling the Mass Distribution of Lens Galaxies

To produce simulated lensed galaxies, we first model the mass of lens galaxies as a Singular Isothermal Ellipsoid (SIE). This model is both analytically tractable and is consistent with models of individual lenses and lens statistics on the length scales relevant for strong lensing (e.g., Koopmans et al. 2006; Gavazzi et al. 2007; Dye et al. 2008; Li & Chen 2009).

The normalized convergence map of the SIE model is defined as:

$$\kappa = \frac{\theta_{\rm E}}{2}\frac{1}{\sqrt{x_1^2/q + x_2^2 q}}, \tag{1}$$

where $\theta_{\rm E}$ is the Einstein Radius and q is the axis ratio. The Einstein radius can be calculated from the redshift of the lens, the redshift of the source, and the velocity dispersion of the lens galaxy as follows,

$$\theta_{\rm E} = 4\pi\left(\frac{\sigma_v}{c}\right)^2\frac{D_{ls}}{D_s}, \tag{2}$$

Here, c is the speed of light, σ_v is the velocity dispersion of the lens galaxy, D_{ls} and D_s are respectively the angular diameter distances from the source plane to the lens plane and from the source plane to the observer.

To rotate the lenses with random orientation angle ϕ, we adopt the transformation below:

$$\begin{bmatrix} x_1{}' \\ x_2{}' \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{3}$$

From Equations (1)–(3), there are five independent parameters from which we can derive the lensing map: velocity dispersion σ_v, axis ratio or ellipticity q, orientation angle ϕ, redshift of the lens z_l, and redshift of the source z_s.

We choose σ_v, q, and ϕ randomly (flat prior) from typical ranges of observed galaxies: $\sigma_v \in [200, 320]$ km s^{-1}, $q \in [0.5, 1.0]$, and $\phi \in [0, 360]$. While a flat prior is not realistic, we use this as a starting point to test our pipeline (see Section 4 for more details on future work). We obtain the redshift of the lens galaxy by matching the velocity dispersion drawn to generate the simulations to the corresponding redshift from the catalog of elliptical galaxies in the COSMOS survey[12] from Zahid et al. (2015). This catalog contains both the redshift and velocity dispersion measurements from BOSS spectra[13] for massive ETGs in COSMOS that are analogs to our lens galaxies. This results in lens galaxy redshifts in the range, $z_l \in [0.2, 0.7]$. Note that the distribution of COSMOS galaxies leads to a selection of brighter and larger objects at higher redshifts.

2.1.2. Modeling Images of the Lens Galaxies

We model the light distributions of the lens galaxies with an elliptical Sérsic profile

$$I(R) = I_{\rm eff}\,\exp\left\{-b_n\left[\left(\frac{R}{R_{\rm eff}}\right)^{1/n} - 1\right]\right\}, \tag{4}$$

where $R = \sqrt{x_1^2/q + x_2^2 q}$, $R_{\rm eff}$ is the effective radius in units of arcseconds, $I_{\rm eff}$ is the intensity at the effective radius, n is the index of the Sérsic profile, and q is the ellipticity of the lens galaxy. We perform a similar transformation as Equation (3) to orient the source galaxies and assume that the distribution of light follows that of mass. The ellipticity and orientation are therefore the same as in the SIE model.

To create images of the lens galaxies, we also utilize the catalog to match the velocity dispersion with an assigned effective radius and effective luminosity to the light profile. As described in Section 2.1.1, the catalog consists of both COSMOS imaging and BOSS spectroscopy for velocity dispersion measurements of massive early-type galaxies, providing sufficient information to construct a fundamental plane of relations between all relevant quantities. We explicitly use the fundamental plane from these observations to relate σ_v in the lensing map with $R_{\rm eff}$ and $I_{\rm eff}$ in Equation (4).

To create images of the lens galaxies, we use the COSMOS morphological catalog (Zahid et al. 2015) to match the velocity dispersion with an assigned effective radius, effective luminosity, and index to the light profile. This catalog uses SDSS/BOSS selection criteria. Note that we explicitly use the fundamental plane from these observations to relate σ_v in the lensing map with $R_{\rm eff}$, $I_{\rm eff}$, and n in Equation (4).

While the galaxy distributions of the COSMOS data at higher redshifts are biased toward larger galaxies and the highest surface brightness galaxies for fixed size, this selection should mimic those in surveys such as the SLACS sample described in Section 3.4.3. We next assume the light center is on top of the mass center and create noiseless images of lens galaxies.

We construct galaxies along the line of sight by cutting light-cones from the Hubble Ultra Deep Field.[14] We create a composite of these images with the lens galaxy image and the

[11] https://www.lsst.org

[12] http://cosmos.astro.caltech.edu/

[13] http://www.sdss3.org/surveys/boss.php

[14] http://www.spacetelescope.org/science/deep_fields/

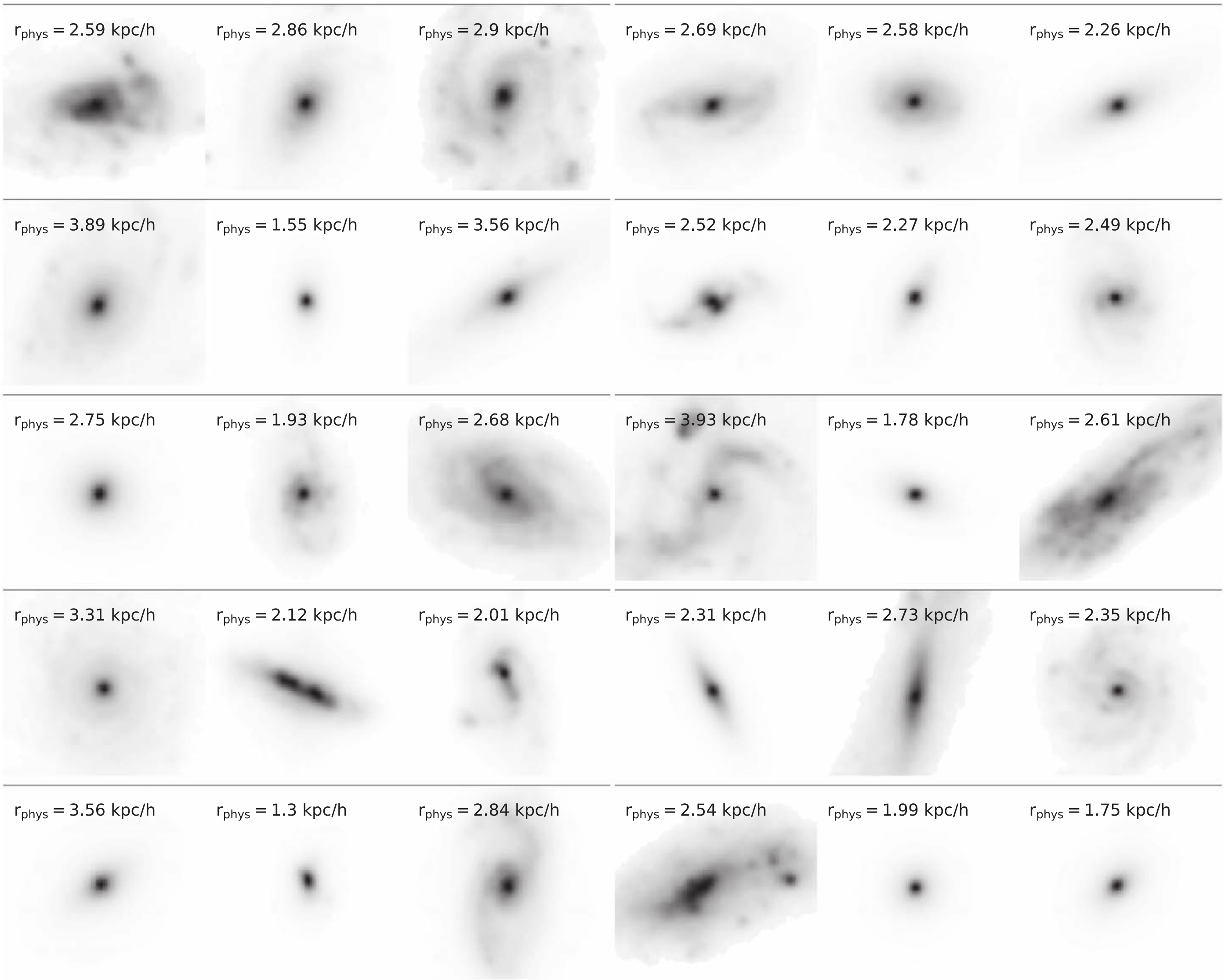
$r_{\rm phys} = 2.59$ kpc/h
$r_{\rm phys} = 2.86$ kpc/h
$r_{\rm phys} = 2.9$ kpc/h
$r_{\rm phys} = 2.69$ kpc/h
$r_{\rm phys} = 2.58$ kpc/h
$r_{\rm phys} = 2.26$ kpc/h
$r_{\rm phys} = 3.89$ kpc/h
$r_{\rm phys} = 1.55$ kpc/h
$r_{\rm phys} = 3.56$ kpc/h
$r_{\rm phys} = 2.52$ kpc/h
$r_{\rm phys} = 2.27$ kpc/h
$r_{\rm phys} = 2.49$ kpc/h
$r_{\rm phys} = 2.75$ kpc/h
$r_{\rm phys} = 1.93$ kpc/h
$r_{\rm phys} = 2.68$ kpc/h
$r_{\rm phys} = 3.93$ kpc/h
$r_{\rm phys} = 1.78$ kpc/h
$r_{\rm phys} = 2.61$ kpc/h
$r_{\rm phys} = 3.31$ kpc/h
$r_{\rm phys} = 2.12$ kpc/h
$r_{\rm phys} = 2.01$ kpc/h
$r_{\rm phys} = 2.31$ kpc/h
$r_{\rm phys} = 2.73$ kpc/h
$r_{\rm phys} = 2.35$ kpc/h
$r_{\rm phys} = 3.56$ kpc/h
$r_{\rm phys} = 1.3$ kpc/h
$r_{\rm phys} = 2.84$ kpc/h
$r_{\rm phys} = 2.54$ kpc/h
$r_{\rm phys} = 1.99$ kpc/h
$r_{\rm phys} = 1.75$ kpc/h

Figure 1. Images of our source galaxies from the CANDELS catalog. We annotate the physical size of each galaxy when redshifted to $z_s = 2$ in the top left corner of each thumbnail image. At the source redshift, each thumbnail is 0.96 arcsec across; the resolution of each is 128 pixels^2 at 0.0075 arcsec/pixel.

lensed source galaxy, calibrating the magnitude of all three components.

2.1.3. Modeling Images of the Source Galaxies

The background source galaxies come from a set of detailed images of low-redshift bright galaxies ($z \sim 0.45$) that have been extracted from mosaics produced by the CANDELS team (Grogin et al. 2011; Koekemoer et al. 2011) and selected from the CANDELS UDS catalog (Galametz et al. 2013). For these source images, we use the F606W band, which is the closest filter available in CANDELS data to the g-band of LSST. We rescale these background source galaxies to artificially redshift the sources to $z_s = 2$, and select source positions near caustics of the lensing system. The rescaling involves a correction to both the size and magnitude corresponding to the change in cosmological distances with $z_s = 2$. The rescaling did not involve a color adjustment, as our mock data is achromatic.

We add the caveat that the selection function of our galaxies is biased as this sample is not complete. Here, we select bright large objects with apparent magnitude <20, and effective radius above 3 arcsec. We define an effective radius as the radius enclosing pixels with values above 3σ of the background noise. While this leaves us with only 31 such sources, we leave a full exploration of the statistical bias of source galaxy distribution in training sets to future work. A thumbnail panel of 30 of these images is in Figure 1, each annotated with the physical size of the galaxy at $z_s = 2.0$. The physical sizes range from 1.3 to 3.93 kpc$/h$.

The selected positions are within a $2'' \times 2''$ square box centered on the lens galaxy center of mass. We then fix the projected position of the lens galaxy at the center of the field of view for each image.

Next, we perform ray-tracing simulations using the modeled galaxy images and parameters. The lens galaxy, source galaxy, and parameters of the image simulations (i.e., 300×300 pixels^2 with 0.03 arcsec/pixel for an *HST*-like image) are inputs to the simulation that produce ideal lensed images with the appropriate resolution.

2.1.4. Mock Observing

The final part is to perform the mock observation from the ideal images produced in the previous step. Both our *HST*- and LSST-like observations are monochromatic. The *HST*-like sample is in the F606W band and the LSST-like sample is in the *g*-band. We note that the use of a single band limits the potential performance of our models. We refer the reader to Metcalf et al. (2018) for an example of our model trained on multi-band data where we concatenated the feature vector from each band.

We create a composite image of the lens galaxy, source image, and galaxies along the line of sight. While we adjust the source size and magnitude to correspond to the change D_s when placed at z_s, we do not adjust colors. The distribution of lens properties is not realistic with a constant z_s, but provides a sufficient start in covering the feature space to test supervised classification methods. Incorporating the lens, source image, line of sight galaxies, and noise are particularly crucial in methods that use edge features.

For *HST*-like observations, we do the following for each component. The component that mimics along the line of sight galaxies is a cutout from the HUDF. The inclusion of this cutout results in an image where noise and the point-spread function (PSF) for *HST* is in the field of view. The lens galaxies have been convolved with the *HST* PSF, but their angular extent is significantly larger than the PSF of $\sim 0\overset{\prime\prime}{.}03$. Convolution of the lens galaxy component will not noticeably alter its appearance. The source images are from a ray traced CANDELS galaxy observed with *HST* PSF. This procedure does not capture the true clumpiness of these sources. We then create a composite image and magnitude calibrate all components to produce the final *HST*-like images, which are $300 \times 300\,\mathrm{pix}^2$ with 0.03 arcsec per pixel.

Note that we do not convolve the lensed image with the *HST* PSF in a final step after rescaling and ray-tracing. The original source image is a real *HST* galaxy that already has the *HST* PSF. We then magnify the source galaxy via the lensing procedure. It does not then make sense to perform an additional final convolution that would erase the clumpiness potentially captured in *HST* lensed arcs. Note that the observation of a true lensing effect would have a better PSF providing finer details than in the unlensed *HST* images that we have used.

For LSST-like observations, we use the *LensPop* software (Collett 2015). We resample images to match the detector pixel scale and convolve the resampled image with a circularly symmetric Gaussian PSF discretized at the same pixel scale. To generate a noisy realization of the image, we assume a Poisson model based on the sky plus signal, and an additional Gaussian read-out noise. Parameters for these simulations follow Collett (2015) and are based on the LSST observation simulator (Connolly et al. 2010).

To account for variations in seeing and sky-brightness over the course of the survey, we draw each simulated exposure from a stochastic distribution of these parameters. We then consider two different strategies to use the simulated exposures. First, we build one single-epoch image for each field (hereafter labeled as LSST-best) by keeping only the best seeing exposure. Second, we build another “worst-case” stacked image by degrading all individual exposures, 10 per filter per year, to match the one with the worst seeing and co-add all exposures to a single image (hereafter labeled as LSST10). These two sets of images will allow us to investigate the trade-off between resolution and signal-to-noise for our automated lens search.

Figure 2 illustrates sample mock observations with a strong lensing signature from each telescope. The leftmost column corresponds to a mock *HST* lensing system with a highly magnified source galaxy (top) and a less visible image of the source galaxy (bottom). For the *HST*-like data set, many arcs are visually obvious due to the exquisite spatial resolution and quality of space-based imaging.

The middle column corresponds to the same simulated systems for LSST10. The right-most column corresponds to the simulated systems of LSST-best. These images have resolution $45 \times 45\,\mathrm{pixels}^2$ with $0.18\,\mathrm{arcsec/pixel}$. LSST10 images visually exhibit the improved signal-to-noise ratio, recovering the arc feature, albeit at a much lower resolution than with the *HST*-like image or the LSST-best image. The top images of LSST10 and LSST-best show a visible lensed source galaxy image. The bottom images are washed out in the bottom row, where the magnification of the source galaxy is not as large. The ground-based noise, PSF, and limited resolution of the LSST-best make visual giant arc identification difficult, except in systems with the most magnified source galaxies.

Our mock observations also include non-lens containing images. The procedure is similar to mock lensed images but we do not perform ray-tracing, so these images do not have lensed source galaxies.

Furthermore, we investigate the influence of light from the lens galaxies on the performance of our lens identification pipeline. We generate another set of each *HST*, LSST-best, and LSST10 images without the lens galaxy. We respectively label these nHST, nLSST-best, and nLSST10.

The final data set is then comprised of 6 × 10,000 lens containing images, and 6 × 10,000 non-lens containing images. We also keep a holdout set of 6 × 1000 lens containing images, and 6 × 1000 non-lens containing images.

2.2. Strong Arc Lensing Identification Pipeline

To perform our analysis, we have used tools from *Scikit-learn* (Pedregosa et al. 2012) to identify galaxy–galaxy strong lensing systems through supervised classification. Supervised classification is a class of machine learning where the class labels in the training set are known. In our case, the labels are “lens” and “non-lens” containing images.

The first step of our pipeline consists of a feature extraction stage, where our feature vector is a HOG (Dalal & Triggs 2005) that quantifies edges in the image. We describe the method and parameter search in Section 2.2.1. We then use LR, a machine learning algorithm described in Section 2.2.3, to train a classifier model on a subset of our images. LR requires a parameter search over the regression coefficient, C_{LogReg}, which we explore in Section 3.2. We briefly comment that our initial tests with a Support Vector Machine (SVM) using radial basis functions as an alternative machine learning algorithm yielded negligible performance improvement, and significantly increased computation cost. This indicated that the features of lens and non-lens images are relatively well separated by hyperplanes in feature space. For these reasons, we do not include SVM in our final analysis and comparisons and continue all discussions with a linear classifier.

Both the feature extractor, HOG, and the linear classifier, LR, contain parameters that must be tested and optimized for peak model performance. We use *GridSearchCV* from

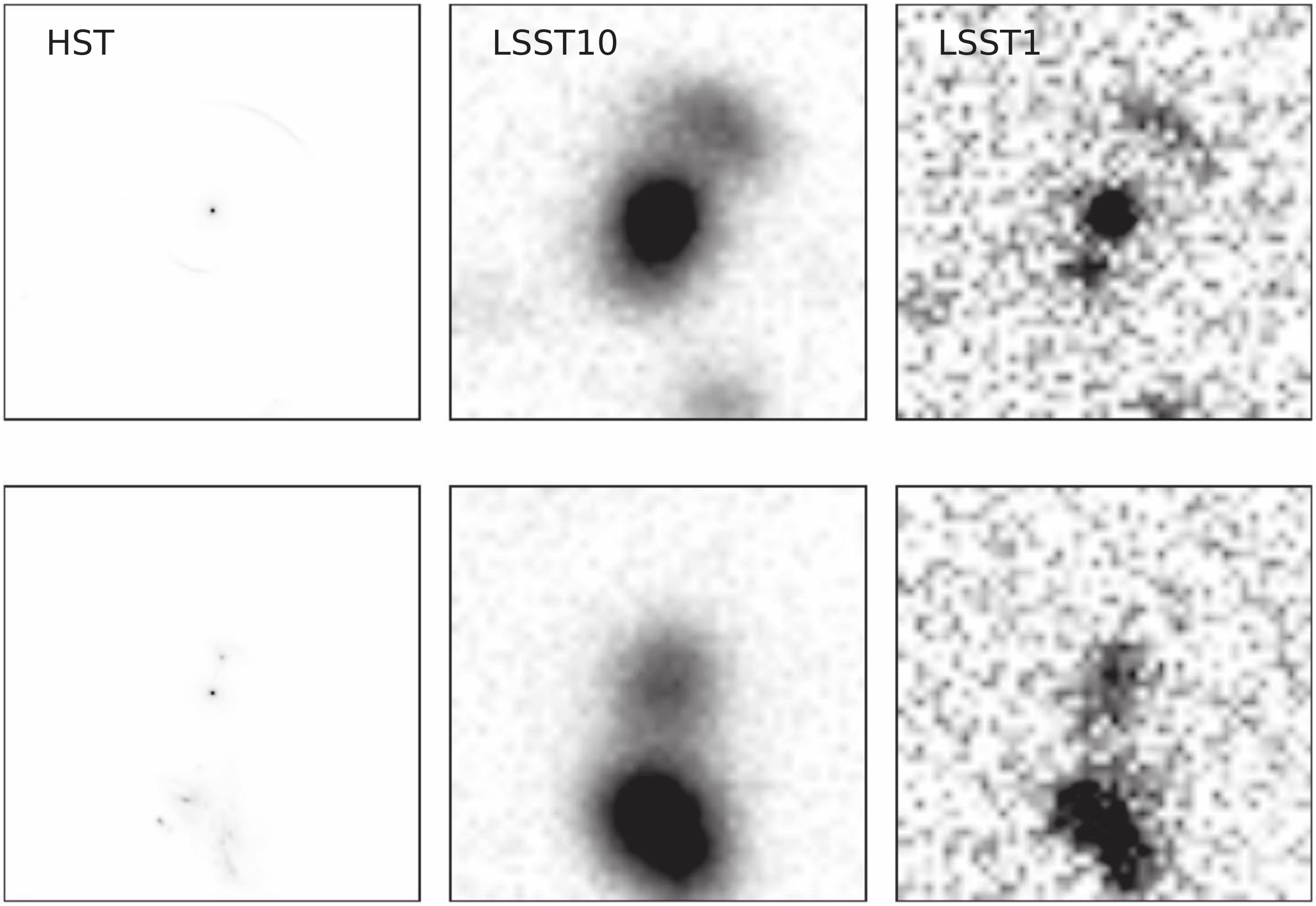

Figure 2. Left to right show example mock *HST*, LSST 10 year, and LSST-best images. The top row corresponds to a lensing system with a very visible arc signature, and the bottom row to a lensing system that is less obvious. Example mock *HST* images have $n_{\rm pix} \times n_{\rm pix} = 300 \times 300$. Example mock LSST images have $n_{\rm pix} \times n_{\rm pix} = 45 \times 45$. The resolution and noise of a ground-based telescope is noticeably worse. Visual identification of giant arcs in the LSST images in the bottom row is very difficult.

Scikit-learn to select cross-validated parameters, and discuss this step of our methodology in Section 2.2.2.

The second step of our analysis is to test our trained model on an independent subset of the images to assess the model performance. Here, we evaluate the model on each test image, predicting a likelihood ("score") between 0 and 1 that image contains a lensing system. This "holdout set" is not used in any of our parameter searches to keep our test metric independent of tuning.

2.2.1. Feature Extractor: Histogram Oriented Gradients

Originally created for human detection in computer vision, HOG is a feature extraction method that computes centered horizontal and vertical gradients. HOG is relatively robust to noise in the image, and is a fairly fast transform that describes edges. Details can be found in Dalal & Triggs (2005), but we describe the procedure here. The end result of HOG is a one-dimensional histogram computed as follows.

HOG first divides the image into blocks of 50% overlap. Each block contains $m \times m$ cells-per-block that each contain $n_{\rm pix} \times n_{\rm pix}$ pixels-per-cell. The computed gradient orientation is quantized into $N_{\rm orient}$ bins. Each gradient is computed using a $[-1,0,1]$ and $[-1,0,1]^T$ filter kernel to provide the x and y components of the gradient.

The orientation gradient of all pixels within each cell are binned into the quantized orientations, providing a net gradient description within that cell. As an example, for $N_{\rm orient} = 3$, our bins are centered at $\theta = 0, 2\pi/3, 4\pi/3$ in radians. If a cell only has a gradient in the $\theta = \pi/2$ direction, it will contribute 75% of its magnitude to the $\theta = 2\pi/3$ bin, and 25% of its magnitude to the $\theta = 0$ bin. The bins in all cells are then concatenated to make a larger feature vector that is $N_{\rm orient} \times N_{\rm cells}$.

The last step is a normalization procedure to control for illumination effects. Here, the sub-histograms of each cell within the same block are normalized with respect to one another before the transformation returns the final feature vector. The division of the image and the quantization of orientations are thus controlled by three parameters in HOG: $N_{\rm orient}$, cells-per-block, and pixels-per-cell. We discuss how we select parameters using cross-validation in Section 2.2.2.

2.2.2. Optimized Pipeline Parameters with a Grid Search

We run a grid search across parameters that should reasonably sample the arc edges in either the *HST*- or LSST-like mock observations, and illustrate the results in Table 1. The grid search procedure uses a three-fold cross-validation to help choose the best parameters for the different simulated data sets. Here, the three-fold cross-validation consists of splitting

Table 1
Grid Search of Pipeline Parameters

$N_{\rm orient}$	Pixels/Cell	Cells/ Block	$N_{\rm feat}$	$C_{\rm LogReg}$	Score
		(a) *HST*-like data			
9	(8, 8)	(4, 4)	166464	10	0.8764 ± 0.0064
9	(16, 16)	(4, 4)	32400	10	**0.9014 ± 0.0066**
9	(24, 24)	(4, 4)	11664	10	0.8939 ± 0.0084
9	(32, 32)	(3, 3)	3969	50	0.8764 ± 0.0105
5	(16, 16)	(4, 4)	18000	10	0.8945 ± 0.0084
7	(16, 16)	(4, 4)	25200	10	0.9003 ± 0.0096
5	(8, 8)	(1, 1)	6845	50	0.8456 ± 0.0080
5	(16, 16)	(1, 1)	1620	50	0.8381 ± 0.0113
5	(24, 24)	(1, 1)	720	50	0.8594 ± 0.0094
5	(32, 32)	(1, 1)	405	50	0.8581 ± 0.0111
5	(16, 16)	(1, 1)	1620	50	0.8381 ± 0.0113
7	(16, 16)	(1, 1)	2268	50	0.8488 ± 0.0092
9	(8, 8)	(1, 1)	12321	50	0.8539 ± 0.0084
		(b) LSST-like data (LSST10)			
4	(3, 3)	(3, 3)	6084	50	0.6155 ± 0.0049
3	(4, 4)	(3, 3)	2187	100	**0.6680 ± 0.0089**
4	(5, 5)	(3, 3)	1764	50	0.6472 ± 0.0039
4	(7, 7)	(3, 3)	576	100	0.6512 ± 0.0127
4	(4, 4)	(3, 3)	2916	100	0.6583 ± 0.0031
6	(4, 4)	(3, 3)	4374	50	0.6506 ± 0.0150
9	(7, 7)	(3, 3)	1296	100	0.6405 ± 0.0065
9	(3, 3)	(1, 1)	2025	100	0.5400 ± 0.0074
4	(4, 4)	(1, 1)	484	100	0.5867 ± 0.0175
4	(5, 5)	(1, 1)	324	50	0.5800 ± 0.0044
6	(7, 7)	(1, 1)	216	100	0.5936 ± 0.0097
3	(4, 4)	(2, 2)	1200	500	0.6567 ± 0.0082
4	(2, 2)	(1, 1)	1936	50	0.5597 ± 0.0045
6	(3, 3)	(1, 1)	1350	50	0.5525 ± 0.0032

Note. Panel (a) shows a subsample of the results of a grid search for *HST* across a range of HOG parameters, feature vector length, and regularization parameter for Logistic Regression, $C_{\rm LogReg}$, from Equation (6). Panel (b) shows a subsample of the results of a grid search for LSST10. Each use a data set of size 2 × 8000 for cross-validation to get the average scores and standard deviation. We explore different HOG parameters in each data set due to resolution and image size differences. We highlight the best performance from the grid search in bold.

the data into three parts; we train on two of the three parts and test on the third to get a score (accuracy), and rotate which two are trained versus tested. This is how we derived an errorbar on the grid search results in Table 1, the score (or the accuracy, which is the fraction of examples correctly classified) with threshold for classification of 0.5.

Recall, the *HST*-like images are 300 × 300 pixels per image, while the LSST-like mock observations are 45 × 45 pixels per image.

We first estimate the size of a cell that will contain a coherent arc feature. To first order approximation, subdivisions of cells that are 1/100 the area of the entire image should contain coherent arc edges that span an elongated shape within arc-containing cells. Therefore, we sample the pixels per cell parameter from (8, 8) to (32, 32) for the *HST*-like images, and (3, 3) to (7, 7) for the LSST-like images in our grid search.

Next, the cells-per-block parameter determines the normalization of each cell with respect to the neighboring cell. In general, this will downweight arc-like edges in cells that neighbor very bright cells, such as cells that cover the central lens galaxy. We therefore vary the cells-per-block parameter between (2, 2) and (4, 4) for the LSST-like images and between (1, 1) and (4, 4) for the *HST*-like images.

The number of orientations will determine the sampling of rounded edges. For example, if we only have two orientations, an arc-like feature in a cell directly north of the lensing galaxy will appear in our HOG visualization as a strong horizontal line (e.g., see top left in Figure 3), and an arc-like feature northeast of the lensing galaxy will appear as an L-shape. However, contributions from a cluster or LOS galaxy in the same cell will tend to contribute edges in all orientations of the histogram (e.g., bottom right in Figure 3).

Finally, the resolution of the overall image will also limit the additional information that an increase in $N_{\rm orient}$ will provide. From the grid search, the best-case number of orientations for each data set is $N_{{\rm orient},\ HST} = 9$, $N_{\rm orient,LSST10} = 3$, and $N_{\rm orient,LSST\text{-}best} = 5$.

The image resolution affects the length of the HOG feature vector, which has a monotonically increasing relationship with the time required to train the model. Additionally, for fixed memory restrictions, there is a trade-off between the length of the feature vector and the size of the training set. We will discuss how the training set size affects the train time for each data set in Section 3.3.1.

2.2.3. Machine Learning Algorithm: LR

The problem of detecting gravitational lenses in images falls under the general category of *classification* in machine learning. In general, the task is to find a function that assigns data points x to one of two or more classes, denoted by the class label y. This is equivalent to specifying a decision boundary, or decision boundaries between the classes in the space of the data points. (Compare this to *regression* in which the task is to find a function $y = f(x)$, where y is a continuous, rather than discrete, variable.) In our case, we have two classes: lens and non-lens containing images, and the data points x are the HOG feature vectors extracted from the images. In this paper we use the Logisitic Regression (LR) algorithm, for which the decision boundary is a hyperplane. (The equivalent in the regression setting would be linear regression.) In LR we determine the optimal hyperplane by minimizing the objective function

$$L(A, b) = \sum_i \log[1 + \exp(-y_i(A \cdot x_i + b))], \qquad (5)$$

where x_i is a data point (HOG feature vector), y_i is the known label for that data point (1 for a lens containing image, −1 for a non-lens containing), and A and b are the parameters of the hyperplane. Equation (5) is to be minimized with respect to A and b. Other more complicated machine learning algorithms exist that do not necessarily produce a linear decision boundary, such as SVMs, *Random Forests*, and *Neural Networks* (Hastie et al. 2009).

The HOG feature vectors in this paper can be very high-dimensional. When dealing with high-dimensional data, where the number of dimensions becomes comparable to the number of data points, overfitting can become an issue. (An example of an extreme case of overfitting is fitting a degree n polynomial to n points. The polynomial would simply wiggle so that it goes through every point, and would have no predictive power if you tried to interpolate or extrapolate.) In machine learning, overfitting is avoided using different *regularization* techniques. A common choice for LR is to add a penalty term to

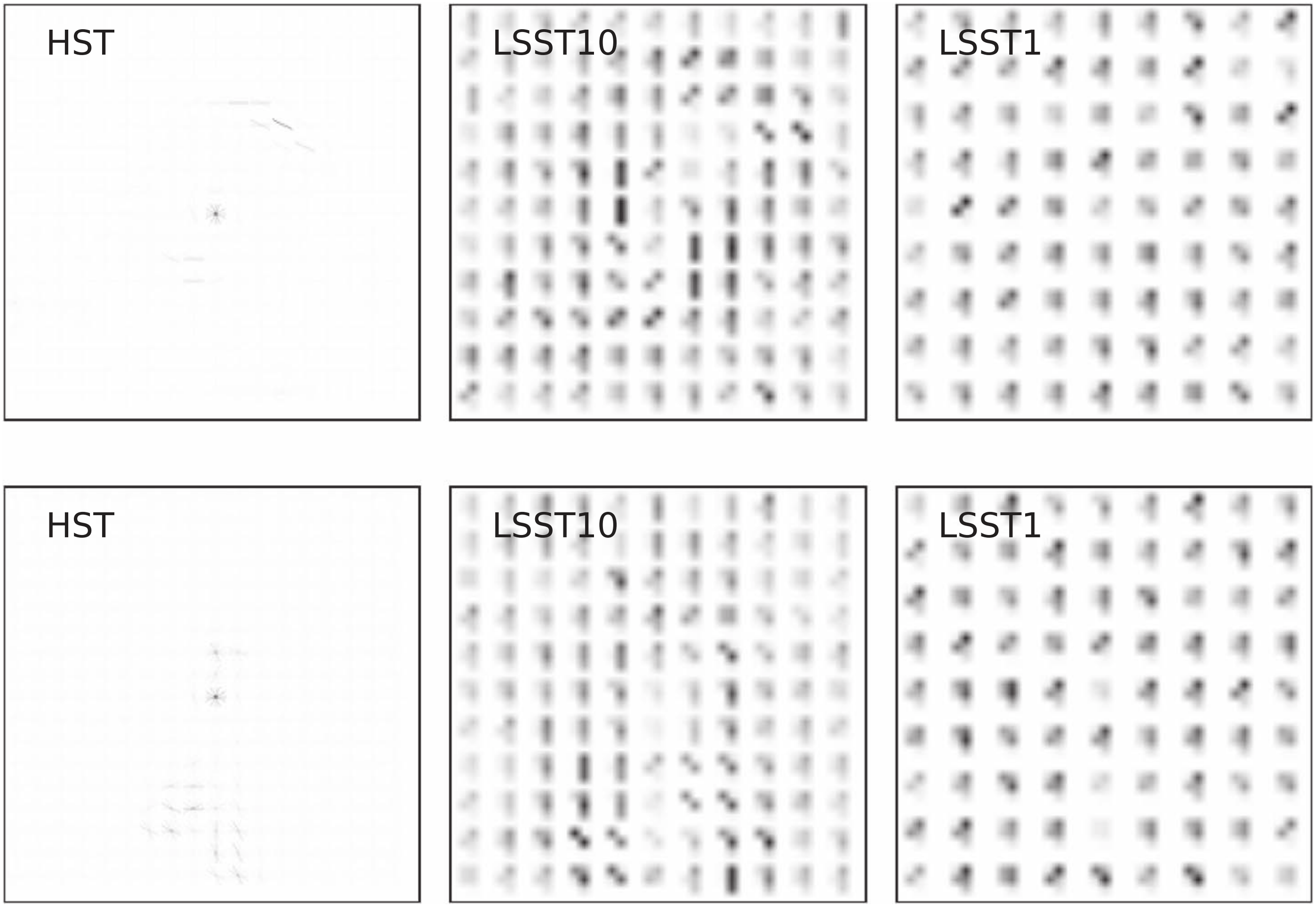

Figure 3. Left to right: example image transforms of mock images from Figure 2 with a visualized histogram of oriented gradients. The image transform picks up edge features, with arc features showing up as edges across radial orientations. Each of the oriented gradients within a cell is color-coded by magnitude, and represented as a line in the direction perpendicular to that gradient. The actual extracted features fill a one-dimensional feature vector comprised of the magnitudes of each of the oriented edges within the visualized cells.

Equation (5)

$$L_{\rm Reg}(A,\, b,\, C_{\rm LogReg}) = L(A,\, b) + \frac{1}{2C_{\rm LogReg}}\|A\|, \qquad (6)$$

where the norm $\|A\|$ is typically taken as either the sum of squares of the coefficients (L_2 norm) or the sum of absolute values of the coefficients (L_1 norm). In this paper we use the former.

The amount of regularization is controlled by the parameter $C_{\rm LogReg}$: larger values of $C_{\rm LogReg}$ correspond to increasing model complexity. If $C_{\rm LogReg}$ is too large then the model will overfit, and if it is too small the model will underfit. To determine whether a model is overfit or underfit, the model is trained, (i.e., Equation (6) is minimized), on a subset of the data called the *training set* and its performance (goodness of fit) is evaluated both on the training set and a separate *test set* that was not used in constructing the model. Figure 6 shows the performance of a model with selected HOG parameters as a function of the regularization parameter, $C_{\rm LogReg}$, for both the training and the test set. (The performance can be measured by the accuracy, i.e., percent of images correctly classified, or by some other metric, such as the area under the ROC curve described in Section 3.1.)

When $C_{\rm LogReg}$ is small, the performance of the model improves with increasing $C_{\rm LogReg}$ on both the training and test set, meaning that $C_{\rm LogReg}$ is still in the underfitting regime. Eventually the performance on the test set reaches a maximum and starts to decrease, even while the performance on the training set continues to increase. This means that the model is no longer generalizing well and is starting to overfit. The optimal $C_{\rm LogReg}$ occurs when the performance of the test set is at its maximum; this is the value of $C_{\rm LogReg}$ that should be used in the final model.

In practice, there is something of a trade-off between accuracy and computational resources because a larger value of $C_{\rm LogReg}$ will also increase the training time, since a larger $C_{\rm LogReg}$ corresponds to a less constrained parameter space being searched. We discuss the performance and training time dependence on $C_{\rm LogReg}$ in Section 3.2.

3. Results

We show results of the HOG and LR generated models using our mock *HST* and LSST data sets. For each of these, we also explore the performance of our models trained on mock data in absence of the lens galaxy as an idealized test of perfectly modeling out the lens. The data with removed lens are labeled as nHST, nLSST-best, and nLSST-10. In the final subsection of

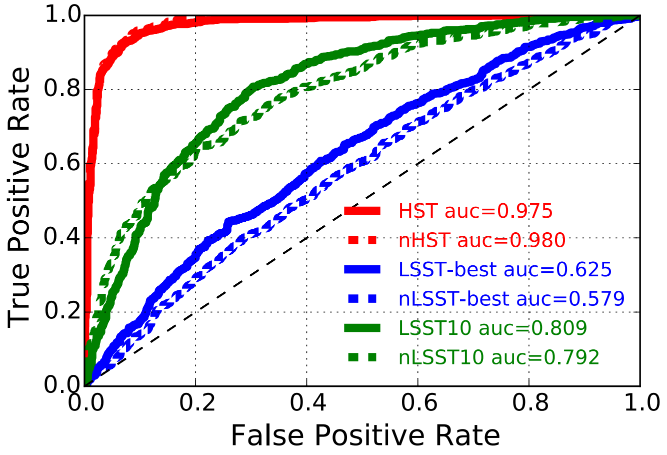

Figure 4. Red, blue, green: ROC curves for models trained on our whole 10,000 training set and tested on our holdout set of 1000. These respectively correspond to the *HST*, LSST 1 yr, and LSST 10 yr data. The solid lines are for data that include the lensing central galaxy, and the dashed lines for the data where there is no lensing galaxy, mimicking an ideal removal of the lens. Model performance can be summarized by the area under the curve (AUC), labeled in the legend. AUC = 1 is a perfect model, and AUC = 0.5 is a useless model.

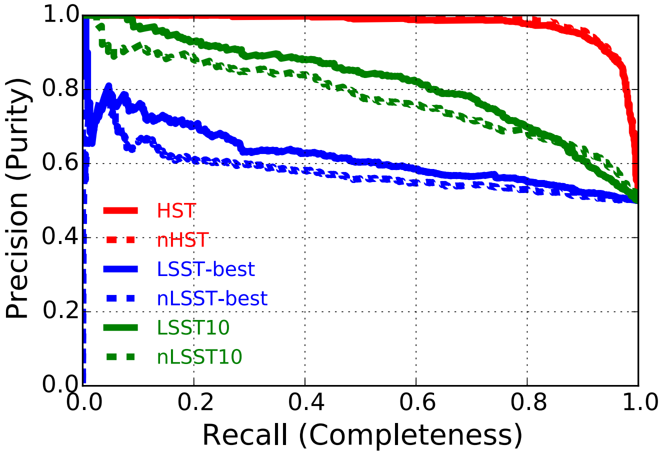

Figure 5. Red, blue, green: precision-recall (PR) curves for models trained on our whole 10,000 training set and tested on our holdout set of 1000. These respectively correspond to the same models and data shown in Figure 4. An ideal model would reach both a precision (purity) and recall (completeness) that equal 1. Note that this performance describes a data set with a 50–50 split between lens and non-lens containing images.

our results, we also examine the performance of models trained on mock *HST* data, and tested on real observed data from the SLACS (Bolton et al. 2008).

3.1. Receiver Operating Characteristic

In this section, we discuss the ROC curve (see Figure 4), which shows the true positive rate (tpr) as a function of false positive rate (fpr) for a given model and test set. The tpr is defined as the number of lenses correctly identified as positive divided by the total number of real lenses. The fpr is defined as the number of non-lenses incorrectly identified as positive divided by the total number of non-lenses. The ROC curve illustrates the performance of our trained model as we vary the discrimination threshold.

The classifier model assigns a score to each test image, which is a probability that the image is a strong lensing system. To construct the ROC curve, we rank the test images by probability, and calculate the tpr and fpr for decreasing discrimination threshold.

Higher discrimination thresholds correspond to higher tprs, but will have more false negatives (bottom left region of the ROC). For a very low discrimination threshold, we have fewer false negatives but more false positives (top right region of the ROC). The ideal model would have an ROC curve with data points that go from $(x, y) = (0, 0)$ to $(x, y) = (0, 1)$ to $(x, y) = (1, 1)$.

In the context of strong lensing systems, we wish to maximize the tpr so we have a representative count of the fraction of strong lensing systems in an observed volume of the universe. We also want to minimize the fpr. Positively identified strong lensing systems will require expensive spectroscopic follow-up for validation. The steepness of the ROC curve indicates how well the model will optimize the two. One way to characterize the performance of a model is with the AUC. The ideal model would have an AUC of 1. We show the ROC curves of our best performing models in each data set.

Figure 4 shows the ROC curves for models trained using the entire 10,000 training sample, with best-case HOG and regularization parameters. The models have been evaluated on a holdout set of 1000 images that were not used in the parameter search. We show the mock *HST*, LSST-best, and LSST10 results respectively in red, blue, and green. Solid lines correspond to a model trained and tested on images with the lensing galaxy. Dashed lines correspond to a model trained and tested on images where the lensing galaxy is excluded from the mock observation, simulating an ideal modeling and subtraction of the lensing galaxy, which has been one proposed method to improve the identification of strong lensing systems. The corresponding AUC is listed in the legend.

The model performance for the mock *HST* data is AUC = 0.975 for images with the lens galaxy, and AUC = 0.98 for images without the lens galaxy (red solid and dashed). On the other hand, the model for our LSST-like data set for one year has an AUC = 0.625 with the lens galaxy and AUC = 0.579 without the lens galaxy (blue solid and dashed), and the model for our LSST-like data set for 10 yr has an AUC = 0.809 with the lens galaxy and AUC = 0.792 without the lens galaxy (green solid and dashed). Removal of the lens galaxy does not systematically perform better, and is actually dependent on the size of the training set. We discuss relative model performance and complexity for images with and without the lens galaxy in Section 3.3.1.

While the ROC curve is a standard metric for supervised classification in the machine learning community, we note that it does not fully capture the practicality of an algorithm since it is measured for a data set with equal lenses and non-lenses. This is not the true ratio between the two classes in the classification. To complement the ROC curve, we also discuss the Precision–Recall (PR) curve. The recall axis is the same as the tpr axis in the ROC curve (number of positively identified lenses divided by the number of real lenses), also called the completeness of a sample. Precision is the number of lenses correctly identified as positive divided by the total number of positive identifications, also known as the purity of a sample.

Figure 5 shows the PR curve for our models. Each point in the figure is calculated with a varying threshold for identification. Since our sample has a class balance of 50–50 between lens and non-lens, the most lenient threshold that classifies all objects as

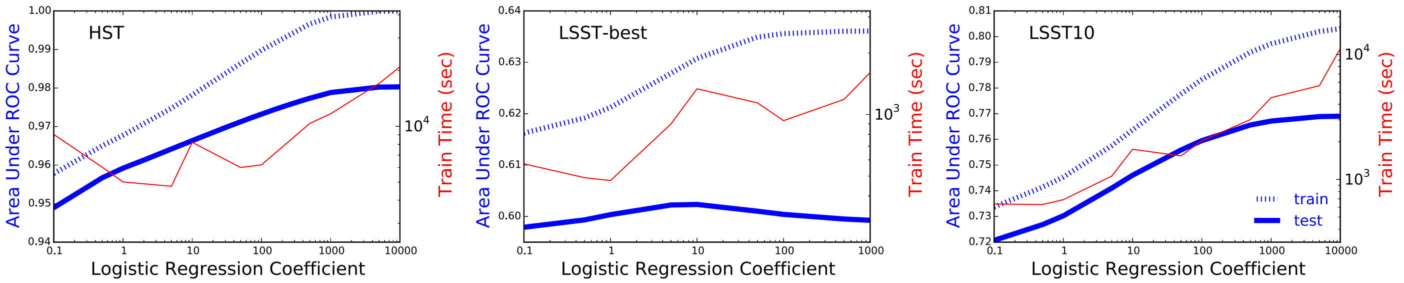

Figure 6. AUC of the model with varying LR regularization coefficient parameter, C_{LogReg}, used when training the model classifier. We use a subset of the 10,000 training images to search over the LR C_{LogReg} parameter, training on 8000 and testing on 1000. Each panel corresponds to a different mock observation. From left to right: *HST*, LSST for one year, and LSST for 10 yr. The solid blue lines correspond to the AUC of the test set, and the dotted blue lines to the AUC of the training set. To avoid overfitting, we choose the smallest parameter for which the AUC of the test set is maximal: 5000, 10, and 5000, respectively. In thin red solid lines, we show the train time of the model, which roughly increases in a log–log scaling with logistic regression coefficient parameter. The train time tick marks are on the right side of each figure in units of seconds.

positive would yield a precision of 0.5 at a recall of 1.0. It is important to note that this figure changes as the class balance changes; if we had 90% non-lenses and 10% lenses, the most lenient threshold would yield a precision of 1/9 at a recall of 1.

In full application to real data, the precision quantity is what determines the efficiency of follow-up by spectroscopic measurements or human inspection, and we expect the number of lenses to non-lenses to be 1–1000. For approximate comparison, a 30% recall (completeness) for our mock *HST* data set has a 0.9967 precision in the 50–50 class balance, which then corresponds to a 23% precision in the realistic class balance of 1–1000. However, for LSST10, a 30% recall corresponds has a 0.89 precision in the 50–50 balance, which then corresponds to 1% in precision in 1–1000. The precision in our *HST* data is relatively idealized, so we expect the purity to be an absolute upper limit estimate for how well the HOG/LR methods might be able to do in real observations. For comparison, the precision-recall values quoted in other simulation-based tests are 94%–100% in precision with 96%–100% recall in Jacobs et al. (2017) using convolutional neural networks, where the lens to non-lens ratio is approximately 1:1. As another example, in Gavazzi et al. (2014), the values are 29% in precision and 42% in recall using RingFinder in their sample. But these values are only directly comparable to tests where the ratio of lens to non-lens is the same.

3.2. Effects of Regularization on Model Performance

As described in Section 2.2.3, LR trains a model with complexity determined by the regularization parameter coefficient, C_{LogReg}. Larger values of C_{LogReg} are less regularized and allow for increased model complexity. The highest values of C_{LogReg} will better describe features in the training set. However, an overly complex model will overfit the training set at the expense of its performance on any independent test set. The regularization parameter ultimately defines the model performance, and we must perform a parameter search to identify the optimal value for C_{LogReg}.

Figure 6 shows the model performance as a function of regularization parameter for each data set *HST*, LSST-best, and LSST10. The solid and dotted blue lines respectively correspond to the model performance on the test and training set, with the AUC as a metric for performance. In red, we show the train time as a function of C_{LogReg}.

As expected, the training set AUC increases and asymptotes with C_{LogReg}. With increasing model complexity, the model better fits the training data set. This is analogous to fitting a seventh order polynomial to seven data points, where the fitting function will go through every point but will not likely predict additional points. With increasing model complexity, we are better able to capture features that are generally characteristic of strong lensing systems with arcs. However, past a certain C_{LogReg}, the model performance on the test set decreases or asymptotes, as it has overfit the training set. We use the scaling of AUC with C_{LogReg} when training 8000 out of our full 10,000 training data set to determine the best value for C_{LogReg}. However, the optimal parameter is also dependent on the size of the training set (see Section 3.3.1), so this choice is not generalizable.

For fixed training set size, the log of the train time roughly scales linearly with the log of C_{LogReg}. Since lower values of C_{LogReg} correspond to a more regularized model, there is a smaller volume in hyperparameter space to search for the best fit coefficients. The solution, on average, will converge more quickly, for more regularized models. The scaling is not purely monotonic because the fitting still has some randomness associated with the path it takes to convergence.

3.3. Data Set Size Dependence

3.3.1. Effects of Training Set Size on Performance and Train Time

In this section, we show the effects of training set size on model performance on the holdout test set of 1000 images. We also compare the improvement between images that include the lens galaxy and images with no lens galaxy.

Figure 7 shows how the AUC depends on the log of the size of the training set for both the LSST10 (LSST-best) data in the solid blue (red) line, and the nLSST10 (nLSST-best) data in the dashed blue (red) line. The AUC for models trained on the LSST10 data improves almost linearly with the log of the training set size, increasing from AUC = 0.705 to AUC = 0.788 when the train size is increased from 2 × 500 lens/non-lens images to 2 × 8000 lens/non-lens images. However, for the nLSST10 data, where the lens galaxy has been removed from the images, the improvement is less dramatic. With the same increase in training size, the AUC for nLSST10 changes from just below 0.77 to just below 0.78.

In the LSST10 case, the trained model can incorporate the additional information of the edges from the lens galaxy, which is correlated with the lensing cross-section and likelihood of the image being a lensing system. The nLSST10 images do not contain this information, but provide cleaner signals of the lensed image for lensed images that occur close to the lens

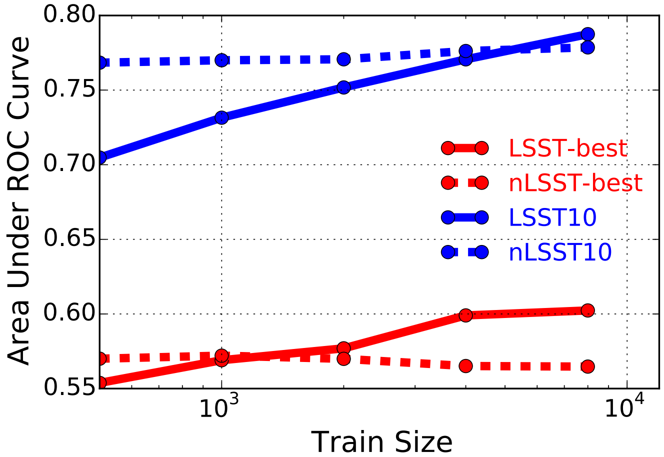

Figure 7. Solid (dotted) blue line: AUC for models with varying size of the training set for LSST10 (nLSST10). Solid (dotted) red line: same for LSST-best. The improvement of AUC scales roughly linearly with the log of the training set size. However, LSST10 and LSST-best have a steeper improvement with data training size. The performance of nLSST10 and nLSST-best models trained on smaller training data sets is better than the respective LSST10 and LSST-best models trained on the same size data, but the LSST10 and LSST-best models outperform with larger size training data set.

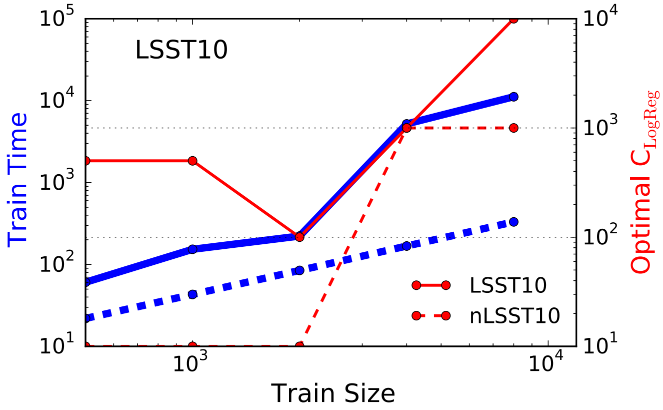

Figure 8. Solid (dotted) blue line: train time for models with varying size of the training set for LSST10 (nLSST10). Train time roughly scales logarithmically with the train size, but the train time is also affected by model complexity. Solid (dotted) red line: best regularization parameter as a function of train size. Note that LSST10 requires more model complexity to exceed the performance of nLSST10 (see blue solid and dotted lines in Figure 7), and therefore requires more training time for continual increase in performance.

galaxy. The cleaner signal in nLSST10 allows for better model performance for smaller training data set sizes ($N_{\rm train} \sim 5{,}000$). However, models trained on LSST10 improve more rapidly with train size, since the additional information from the lens galaxy better describes the lens containing images. For $N_{\rm train} \gtrsim 5{,}000$, the models trained on LSST10 outperform those trained on nLSST10.

In red, we see an analogous improvement for LSST-best and nLSST-best. Here, the crossover happens at train size of 2×1000, where an increase in larger training data set size does not help improve the AUC for nLSST-best.

Since the models that contain information from the lens galaxy edges in LSST10 are more complex, the models require a larger training set size for a better fit. While model performance for LSST10 appears to steadily increase, this comes at the cost of increased train time, which is two-fold. The train time will increase due to both an increase in data to fit, and also an increase in optimal $C_{\rm LogReg}$ where the volume of hyperparameter space for allowed solutions is larger (see red lines in Figure 6).

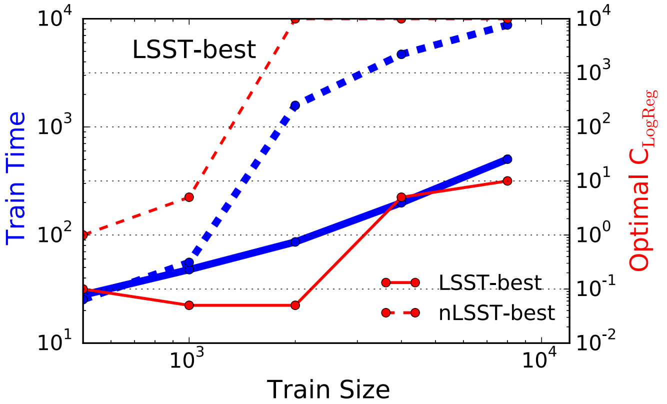

Figure 9. Same as Figure 8, but for LSST-best. Here, nLSST-best requires more model complexity than LSST-best. Note that LSST-best and nLSST-best have better seeing but reduced overall signal-to-noise than LSST10 and nLSST10. This corresponds to better resolution, and therefore sharper edge features that prominently correspond to arcs.

We illustrate the dependence of train time on both the size of the training set and model complexity in Figures 8 and 9. In red, Figure 8 shows that the optimal values for $C_{\rm LogReg}$. For LSST10, $C_{\rm LogReg}$ roughly scales logarithmically with the log of the train size, with an exception of the data point corresponding to train size of 2000. Generally, a larger training set allows for an increase in model complexity without reducing its ability to generalize. This is also true for the optimal $C_{\rm LogReg}$ dependence on the number of training images in the nLSST10 data. But, the required complexity is systematically less than for the LSST10 images.

In blue, Figure 8 also shows the train time for LSST10 and nLSST10 as a function of the size of the training set for the optimal regularization parameter for that subset of the training data. Each model uses features extracted with the same HOG parameterization from the grid search and the optimal regularization parameter for that subset of the training data. The train time of a given model generally increases for increasing regularization parameter. For the subset of train size $N_{\rm train} = 2000$ in LSST10, the optimal regularization parameter happened to be $C_{\rm LogReg} = 100$, whereas it was $C_{\rm LogReg} = 500$ for the subset of train size 1000, and $C_{\rm LogReg} = 1000$ for the subset of train size 4000. This makes the train time increase at train size $N_{\rm train} = 2000$ for LSST10 less dramatic than the average log–log slope of approximately 2.

Since nLSST10 does not contain the lens galaxies, fewer of the extracted features describe the lens system, requiring decreased model complexity. The increase in train time for nLSST10 as a function of train size is mostly due to only having more data to fit in the regression, leading to a steady and slow increase of train time with number of training images with log–log slope of approximately 1.

For LSST-best, Figure 9 shows the same relationships between train time and train size in blue, and the optimal regularization parameter (or model complexity) in red. Contrary to what we found when comparing LSST10 to nLSST10, nLSST-best requires more model complexity than LSST-best. Recall that LSST-best and nLSST-best correspond to single epoch simulated exposures with the best seeing; these images exhibit better resolution but worse signal-to-noise. The

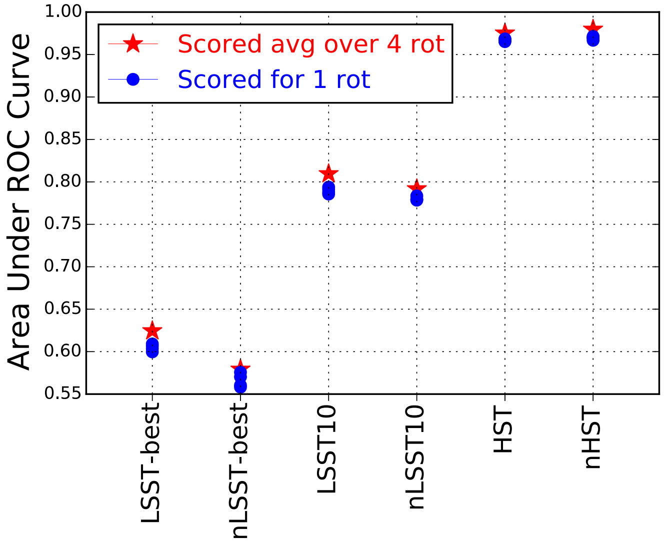

Figure 10. Our summary figure: the AUCs of models trained on the full 10,000 and tested on the holdout 1000. Blue circles: AUC calculated from scores of images at a given rotation (e.g., 0°, 90°, 180°, and 270°). Red stars: AUC calculated from the average score of all rotations of each image. The average score produces an improved AUC in all data sets. We expect the AUC to further improve with increased train size.

switch in required model complexity corresponds to a trade-off between features from the lens galaxy providing additional information or swamping the signal from a strong lensing arc.

3.3.2. *Effects of Rotation on AUC*

To augment our training set, we rotated each image in the set by multiples of 90°. Since our feature extraction method of HOG is not rotationally invariant, augmenting our data by a factor of four naturally optimizes the use of available training data. This has an equivalent improvement to the study illustrated in Figure 7.

We also tested the effects of evaluating our model on all four rotations of the test set, and using the average score of each test image to calculate the AUC. In Figure 10, we show the AUC for different orientations of the data sets. The x-axis corresponds to each of our three data sets, with and without the lens galaxy. The y-axis shows the AUC. The filled blue circles correspond to the four AUCs calculated when the model evaluated images at each of the four rotations. The filled red stars correspond to the AUC calculated from the average of all four test scores, which are systematically higher than any one rotation. The average score across all rotations for each image is likely to be less noisy for the whole test sample, giving an improved AUC.

Figure 10 also summarizes the best-case results of our models trained on our entire 10,000 training sets, and tested on our holdout 1000 test set. Recall, however, that we expect model performance on images containing the lens galaxy to improve further with larger training sets (see Figure 7).

3.4. *Image Classification Performance*

3.4.1. *Populating the ROC Curve*

In this section, we discuss the different image types that our model is most and least able to successfully classify. We have six paradigms of model performance on the mock images based on the score an image receives when evaluated by the trained model, and its true label. From highest scoring to lowest scoring: true Positives (tp), False Positives (fp), Borderline Positives (bp), Borderline Negatives (bn), False Negatives (fn), and True Negatives (tn).

Figure 11 illustrates four images from each paradigm for the holdout test set of LSST10. The trained model used the entire 10,000 image training set. The left two columns show lens systems, and the right two columns show non-lens systems. The annotation in the top left corner of each images shows the score.

In general, the true positives (the highest scoring lens systems) have lensed images with large magnification. The true negatives (the lowest scoring non-lens systems) have small lens galaxies with galaxies along the line of sight that are rounded. The successful classification of these two paradigms are least sensitive to the threshold. On the other hand the failed classification of the false positives (the highest scoring non-lens systems) and the false negatives (the lowest scoring lens systems) are also least sensitive to the threshold. False negatives are typically lens systems with lensed images of smaller magnification and minor distortions that mimic along the line of sight galaxies that the model has learned to ignore. False positives are often non-lens systems with elongated, elliptical, or “fuzzy,” galaxies along the line of sight whose signal blends with the lens galaxy contributing to the fpr even for conservative thresholds. Visually, these false positives are virtually indistinguishable from true arcs, and would require spectroscopic follow-up.

The middle two rows of Figure 11 illustrate the borderline positives and borderline negatives. The successful classification of the borderline positives and negatives are most sensitive to the threshold, and would be the first candidates for alternative classification methods, such as visual follow-up. Thresholds set around these scores yield a tpr and fpr of tpr $\approx$ 0.8 and fpr $\approx$ 0.25, respectively.

3.4.2. *Dependence on Lens–Model Parameters*

Here, we examine lens-model parameters that affect how well our pipeline can classify the system. The lens-model parameters we examined are the redshift, ellipticity, orientation angle, and velocity dispersion of the lensing galaxy, and also the magnification of the lensed image compared with its original size. We found that the magnification of the lensed image is the most correlated lens-model parameter with our trained model performance, with a secondary and related correlation with lens galaxy velocity dispersion that is encapsulated in the Einstein radius. The more strongly lensed an image is, the larger its magnification, and the easier it is for our trained model to classify.

In Figure 12, we show the classification score as a function of Einstein radius and image magnification of the source galaxy in each of our samples. In an *HST*- or LSST10-like observation, lensing systems with images that have magnification $\gtrsim$7 will likely be classified as positive with threshold scores above $\gtrsim$0.5. These systems are also those that are most easily classified by eye. However, our trained model has varying performance for systems with lower magnifications.

Highly magnified systems typically have lenses that are at higher redshifts of and/or lens galaxies with velocity dispersions larger than $\sigma_v \gtrsim 230\,\mathrm{km\,s^{-1}}$. Lensing galaxies with smaller velocity dispersions are less massive and therefore

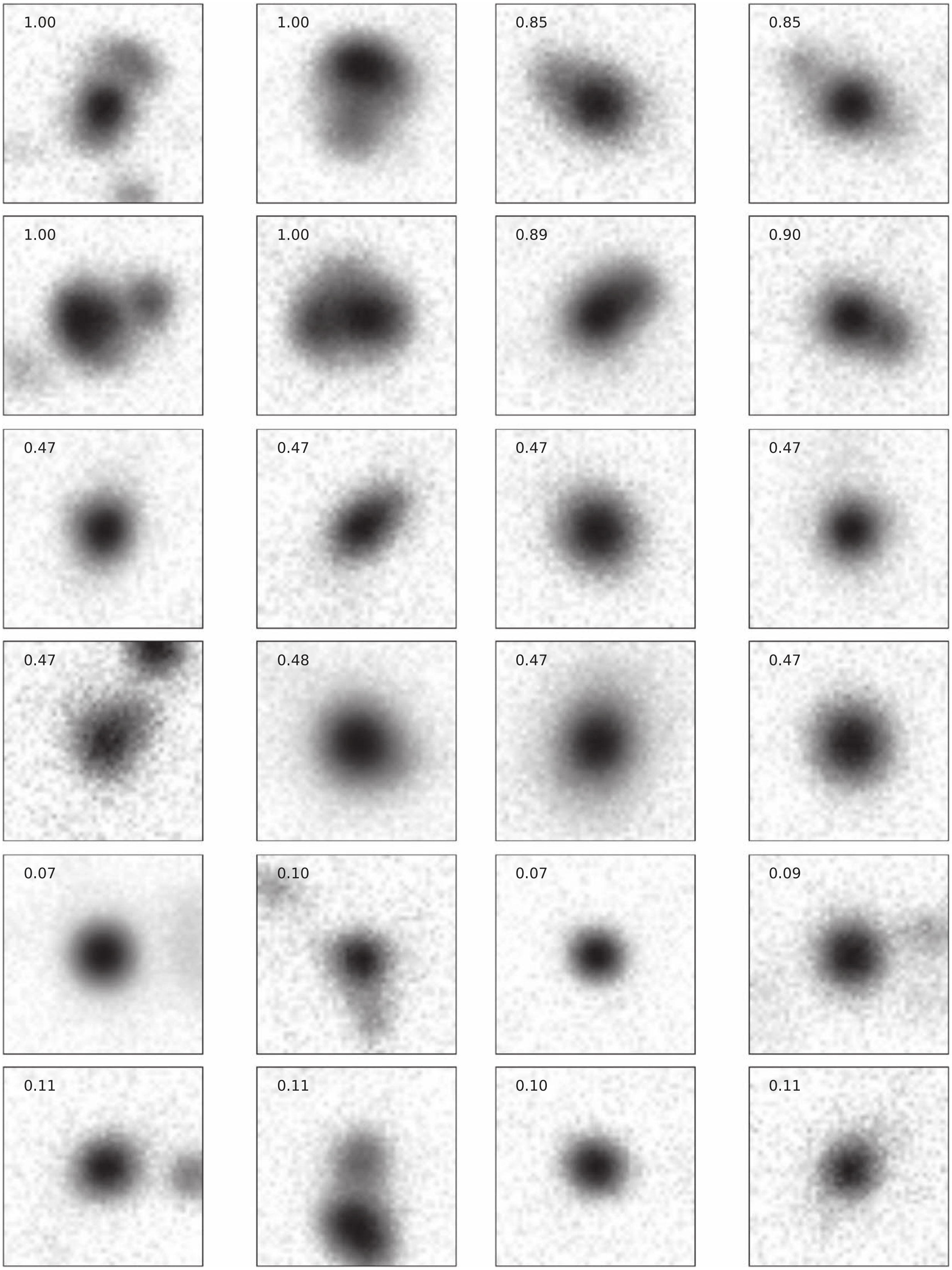

Figure 11. LSST 10 yr mock images. Left two columns: lens containing images, annotated with the image score assigned by our trained classifier. Right two columns: non-lens containing images, annotated with the image score. The top two rows show characteristic images that will be accepted with a high threshold for classification, contributing to the bottom left of the ROC curve in Figure 4. The middle two rows show characteristic images that will be accepted with a moderate threshold, contributing to the knee of the ROC curve, with true positive rates and false positive rates of tpr ≈ 0.8 and fpr ≈ 0.25, respectively. The bottom two rows show characteristic images that will only be accepted with an extremely lenient threshold, contributing to the top right area of the ROC curve.

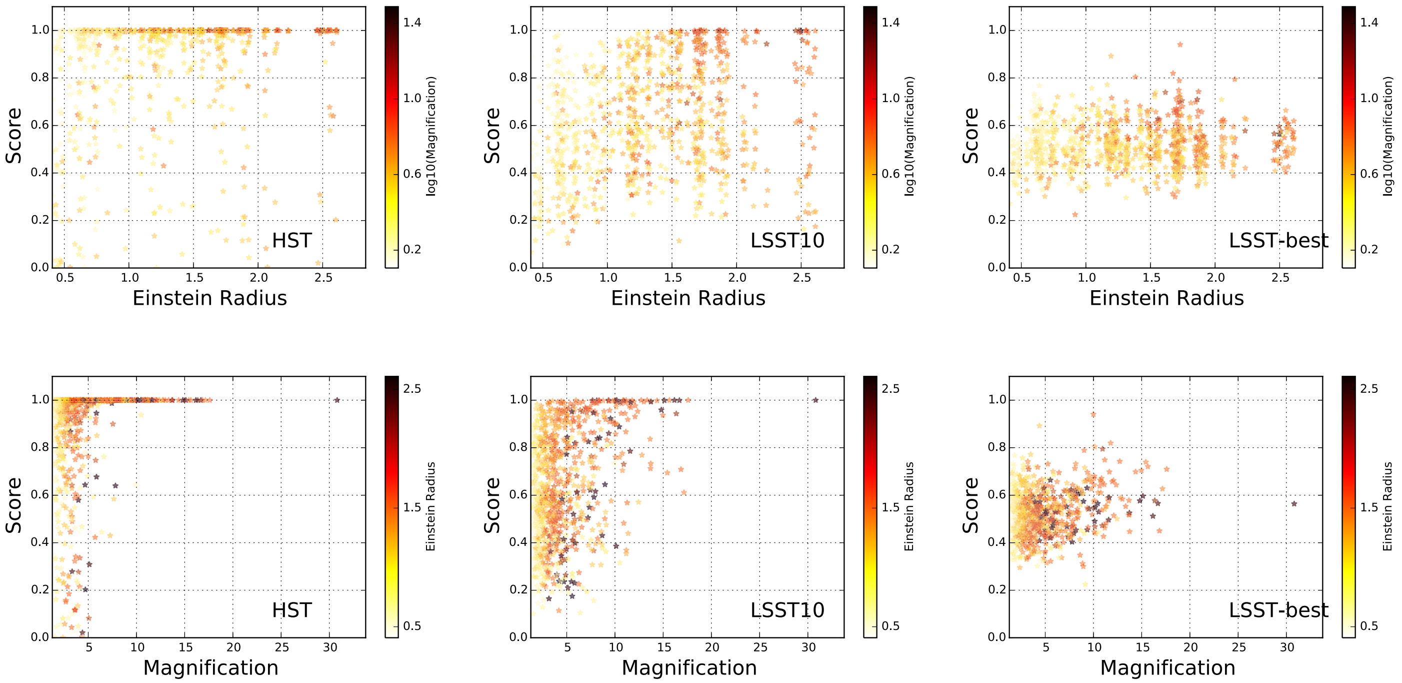

Figure 12. Left to right: image classification score on *HST*, LSST10, and LSST-best mock observed lensing images as a function of Einstein radius (top) and magnification (bottom). The top panels are color-coded by magnification, and the bottom by Einstein radius to visualize the combined effects. Magnification is the strongest indicator of how likely a lensing system will be successfully classified. The Einstein radius has a weaker correlation with how easily a lensing system might be classified, since it contains information on both the velocity dispersion of the lens and the redshifts of the lens and source galaxies. High velocity dispersion lens galaxies are likely to produce high magnification images of the source galaxies.

have a smaller efficiency of lensing cross-section. The smaller efficiency means that a background galaxy is less likely to be strongly lensed with high magnification. Also, due to hierarchical structure formation, galaxies are less massive at higher redshifts, so the trends of model performance with these three parameters are somewhat degenerate with one another.

The relationship between our model performance and lens parameters indicates that the magnification of lensed images is the most relevant, and the distribution of image magnification in a data set will impact trained model performance. We did not find strong correlations in model performance with other lens parameters. It is also useful to keep in mind that lensed galaxy images with magnification $\lesssim$5 are often visually indistinguishable from edge-on disk galaxies along the line of sight, which can lead to false positives. Since the latter can lead to false positives, the model has learned to downweight the related features. Note that the model would be more sensitive to systems with lower magnification without galaxies along the line of sight.

In a forthcoming paper, we will discuss how class imbalance, or differences between the lens-model parameter distributions in the training and test sets affect model performance and will explore a method to correct for this.

3.4.3. Methods Applied to Real Data

We examine the performance of the HOG/LR methodology on data from SLACS, real space-based *HST* observations Bolton et al. (2008). The main conclusion from our tests is that HOG is a feature extraction method where parameters *can* be varied to compensate for imperfections and details that the mock training data does not capture. However, the HOG parameterization that best captures the geometric features of an arc and lens galaxy will vary depending on the quality of the image. This subsection also examines how potential further steps in using HOG might mitigate differences between simulated and real data with a focus on *HST* images. For shorthand, we provide the HOG parameterization of $n_{\rm pix} \times n_{\rm pix}$ pixels-per-cell and $m \times m$ cells-per-block as ppc-$n_{\rm pix}$-cpb-m.

We note that a test of our model on real ground-based data is plotted in Figure 8 of Metcalf et al. (2018). Consistent with other methods explored in Metcalf et al. (2018), the performance of our method decreases when evaluated only on the subset of real data. We do not explore how HOG parameterizations impact the performance on the real data evaluated in Metcalf et al. (2018). Instead, we focus on SLACS data, which is most similar to our mock *HST* data set that we have for model training and testing.

The SLACS data set is comprised of images selected for high-redshift emission lines and a lower redshift continuum in a single spectrum from the Sloan Digital Sky Survey. In the data set we examined, there are 64 clear lensing systems, and 27 non-lensing systems as classified by the authors through visual examination in the direct images and model-subtracted residual images. We do not use ambiguously classified systems from their sample in our test. Each system has an image in at least one of three filters: F814W, F555W, and F435W. For 34 of these systems, there is another exposure at the same filter.

The model trained on mock *HST* images using the best grid search output parameterization, ppc-16-cpb-1, does not uniformly perform well on all images from the SLACS sample. For example, the evaluated score for SLACS J0956+5100 imaged in the F814W filter increased from 0.436 to 0.959 with ppc-12-cpb-3. From left to right, the top row of Figure 13 shows the image, HOG visualization for ppc-16-cpb-1, HOG visualization for ppc-12-cpb-3, histogram for ppc-16-cpb-1, and histogram for ppc-12-cpb-3. As a counter-example, the bottom row of Figure 13 shows SLACS J1420+6019 imaged

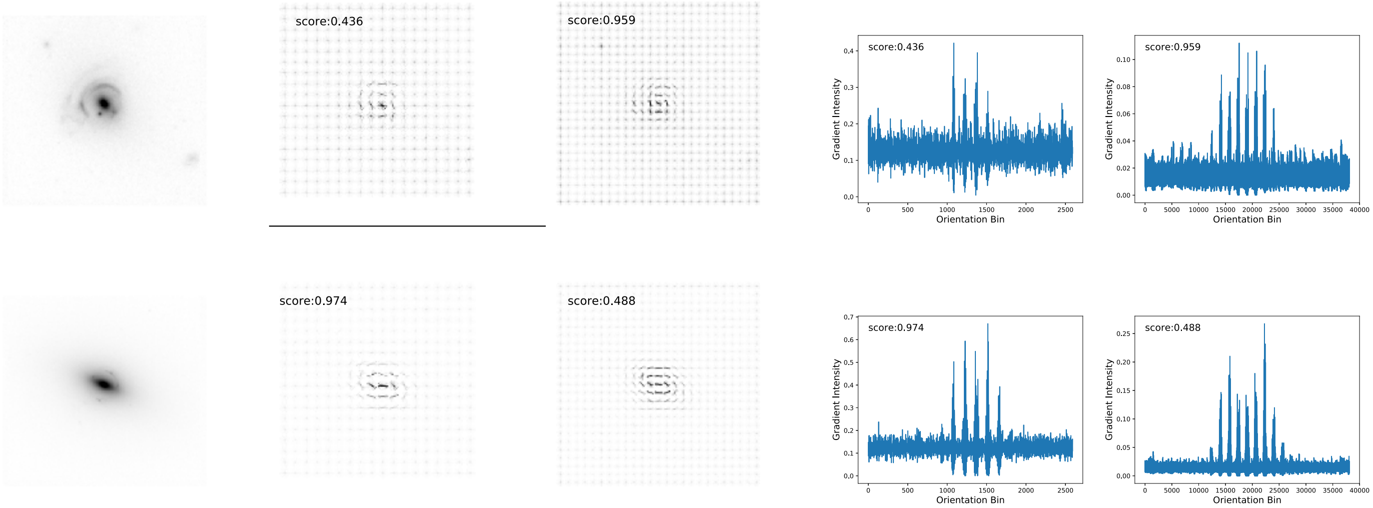

Figure 13. Top row: from left to right, the image of SLACS J0956+5100 in the F814W filter, its HOG visualization for ppc-16-cpb-1, for ppc-12-cpb-3, the histogram for ppc-16-cpb-1, and the histogram for ppc-12-cpb-3. The asymmetric arc features are better captured in the latter HOG parameterization. Bottom row: from left to right, the image of SLACS J1420+6019 in the F555W filter and the analogous HOG visualizations and histograms. For this image, the arc features are better captured in the former HOG parameterization, where arc features contribute to the right side of the histogram.

in F555W filter and its transition went from a score of 0.974 to 0.488 in ppc-12-cpb-3.

The image graininess present in real data can impact how well a given HOG parameterization can capture the morphological features of an arc. In Figure 14, SLACS J1205+4910 is an example of a visibly clear lens that is highly scored for its image in the F814W and F555W filters, but has a significantly lower score in the F435W filter. We also show the feature vector and the visualization of the HOG in the middle and right panels. The grainy features correspond to a higher normalization in additional bins of oriented edges, swamping the signal from the lens edge.

To assess the variations in how well a given HOG parameterization can capture lens parameters, we trained models on only 1000 lensed and non-lensed mock *HST* images in eight different sets of HOG parameterizations in the spirit of a grid search: ppc-6-cpb3, ppc-8-cpb-2, ppc-8-cpb3, ppc-8-cpb-4, ppc-8-cpb-10, ppc-12-cpb-3, ppc-16-cpb-1, and ppc-16-cpb-3. We tested each of these models on a separate 1000 lensed and non-lensed mock images, deriving an AUC on the mock data set to assess the robustness of these models within the simulated data. We then measured the AUC of each model when evaluated on the SLACS data divided into bands. The results of our test are summarized in Table 2. Note that the AUCs of the mock test data are not as high as the results quoted in Figure 10 because of the difference in training set size.

First, we should note that our overall sample to test on is relatively small, so the uncertainty of the AUC values for the SLACS data set is rather large. To quantify this, we compute the bootstrap average and standard deviation, which are shown in the parenthesis of the table column for the AUCs. Note, when subselecting on individual bands, the standard deviation can be almost 25%.

Next, of the HOG parameterizations we tested, ppc-12-cpb3 performed the best on all of the SLACS data, and in particular on images from the F814W filter. Finally, the main point we would like to emphasize is that the models that were trained on mock *HST* data do not contain the same artifacts or variations in graininess as the real observations in the SLACS sample. But, for a given image, a HOG parameterization can be selected that ignores grainy features and keeps features that correspond to arcs. The current problem is that the SLACS sample is too small to run a robust systematic search. In absence of a single ideal HOG parameterization for a SLACS-like sample, we could identify a handful of HOG parameterizations that describe subsets of the data and concatenate the vectors from each HOG parameterization for the feature vector. Feature vector concatenation is how we leveraged multi-band data from the ground-based sample in Metcalf et al. (2018). Another alternative would be to quantify the quality of an image by a metric that correlates best with HOG parameterization.

4. Summary and Discussions

We have presented a supervised classification pipeline to automatically identify galaxy–galaxy strong lensing systems using a HOG as a feature extractor, and LR as a machine learning algorithm. Our pipeline can easily be extended to identify other strong lensing features, such as multiply imaged quasars, and to test alternative features and/or machine learning algorithms.

We have also made use of a new sophisticated set of mock observations, which will be made publicly available. The lensing systems have lens galaxies generated with a realistic redshift distribution and along the line of sight galaxies drawn from Hubble Ultra-Deep field observations. We have explored results from mock *HST*, 1 year LSST, and 10 year LSST observations.

We summarize key points below:

1. We have designed our pipeline to easily select and add image pre-processing and feature extraction methods, and to select a machine learning algorithm for classification. Additionally, the user can easily perform parameter searches to train a model with the best parameters for a given problem.
2. We have tested and run parameter tests for a HOG as an efficient and effective feature extractor for galaxy–galaxy strong lensing systems in both a space-based (*HST*-like) and ground-based (LSST-like) observation. We have also

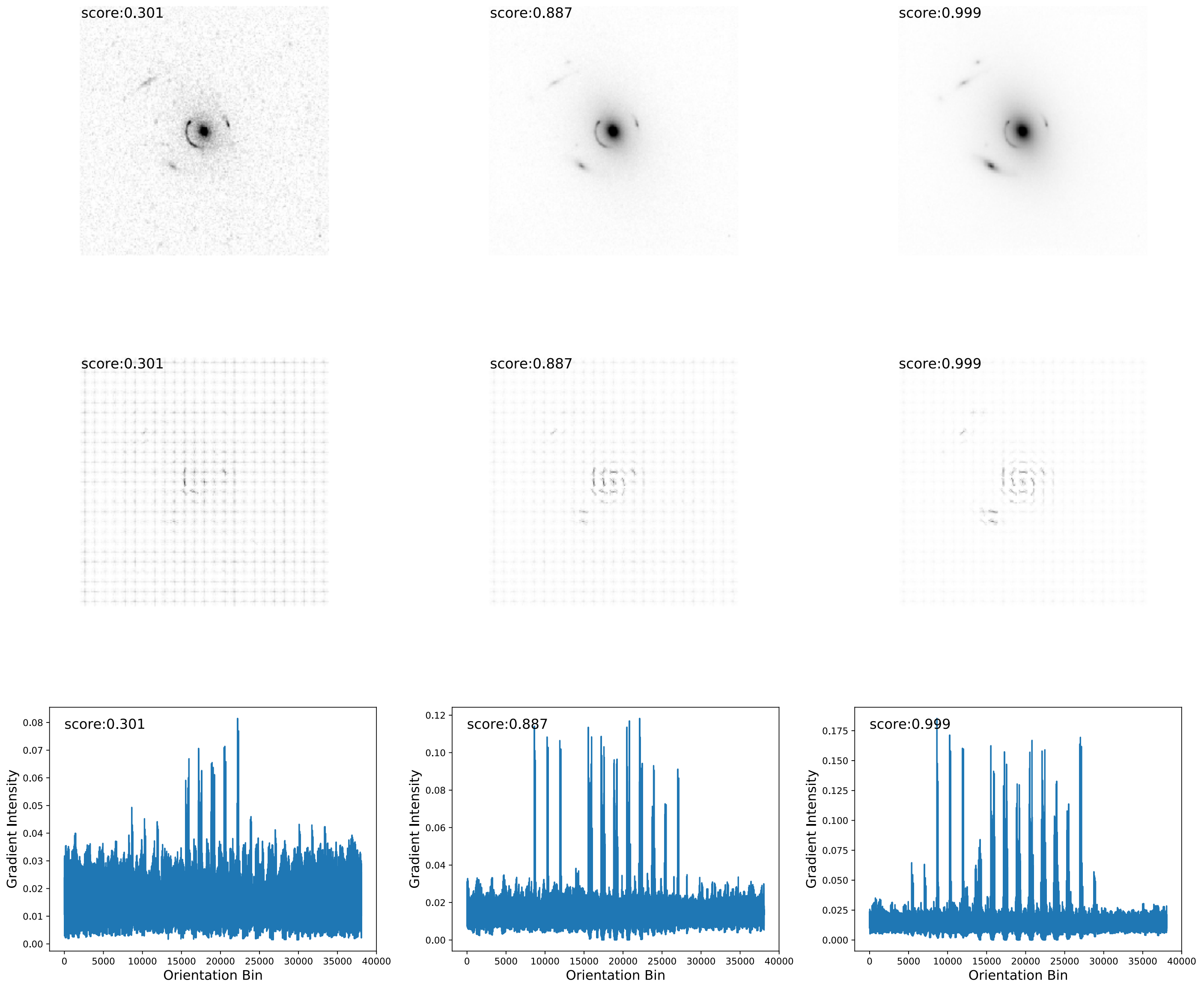

Figure 14. Top row: image of SLACS J1205+4910 in the F435W, F555W, and F814W filters. Middle row: corresponding visualization of the HOG of each image with ppc-12-cpb3. Bottom row: the histogrammed edges. Grainy features in F435W increases the histogram values in every direction, swamping the edge signatures in the left side of the histogram and lowering the score.

tested and run parameter tests for LR as a scalable, cheap, and effective machine learning algorithm.

3. We find AUC values of ROC curves of optimized classifier models to yield AUC = 0.975 for the *HST*-like data, AUC = 0.809 for the stacked LSST-like data. Model performance exhibits continual increase with the training size.
4. While removal of the lens galaxy improves model performance for smaller size training samples, features from the lens galaxy improve model performance for larger training data sets.
5. Images that were easiest for our model to classify typically were lens systems that had high lensed image magnification and a lens galaxy with large velocity dispersion or non-lens systems with lens galaxies with smaller velocity dispersion and non-elongated along the line of sight galaxies.
6. We have explored the potential of HOG/LR in mitigating the problem where simulations are not able to capture imperfections and details in real data from the SLACS sample. The results indicate that different HOG parameterizations can be robust to different amounts of noise/defects that are not captured by our simulations. However, no single HOG parameterization is able to maximally perform on the ensemble of SLACS images. With a larger data set, a systematic study that couples image quality to the best HOG parameterizations would be possible.

We emphasize that simple linear classifiers, such as LR, are scalable and relatively easy to parallelize with open source tools such as *Apache Spark*.[15] Our work indicates that HOG feature extraction plus a linear classifier captures much of the morphological complexity in the arc finding problem. We have also tested how HOG feature extraction can be parameterized to be less sensitive to real image quality variations that are not

[15] http://spark.apache.org/

Table 2
Effects of HOG Parameterization on Model Evaluation

PPC	CPB	$\mathrm{AUC}_{\mathrm{mock}}$	$\mathrm{AUC}_{\mathrm{SL}}$	AUC_{814}	AUC_{435}	P_{mock}	R_{mock}	P_{SL}	R_{SL}	P_{814}	R_{814}	P_{435}	R_{435}
(6, 6)	(3, 3)	0.901	0.469 (0.470 ± 0.052)	0.470 (0.471 ± 0.066)	0.311 (0.312 ± 0.099)	0.713	0.916	0.741	0.496	0.758	0.588	0.333	0.125
(8, 8)	(2, 2)	0.889	0.540 (0.541 ± 0.053)	0.543 (0.543 ± 0.066)	0.417 (0.417 ± 0.108)	0.670	0.936	0.781	0.678	0.789	0.750	0.563	0.375
(8, 8)	(3, 3)	0.920	0.566 (0.566 ± 0.053)	0.573 (0.573 ± 0.066)	0.426 (0.426 ± 0.113)	0.712	0.954	0.777	0.603	0.779	0.663	0.538	0.292
(8, 8)	(4, 4)	0.934	0.574 (0.575 ± 0.052)	0.578 (0.579 ± 0.064)	0.423 (0.422 ± 0.117)	0.731	0.955	0.795	0.545	0.800	0.650	0.333	0.083
(10, 10)	(3, 3)	0.925	0.567 (0.566 ± 0.053)	0.548 (0.548 ± 0.068)	0.433 (0.432 ± 0.111)	0.729	0.924	0.787	0.612	0.500	0.208	0.787	0.738
(12, 12)	(3, 3)	0.932	0.612 (0.611 ± 0.052)	0.607 (0.607 ± 0.065)	0.503 (0.505 ± 0.107)	0.724	0.943	0.817	0.554	0.809	0.688	0.333	0.042
(16, 16)	(1, 1)	0.820	0.509 (0.509 ± 0.056)	0.481 (0.481 ± 0.067)	0.474 (0.475 ± 0.108)	0.581	0.933	0.782	0.711	0.769	0.750	0.667	0.500
(16, 16)	(3, 3)	0.942	0.580 (0.580 ± 0.052)	0.576 (0.574 ± 0.063)	0.439 (0.437 ± 0.116)	0.698	0.963	0.797	0.620	0.787	0.738	0.400	0.083

Note. A table summary of a grid-search-like test to check if a given HOG parameterization might best capture arc morphology in the varying levels of graininess in observed data. The columns correspond to the pixels-per-cell HOG parameterization, cells-per-block, the AUC of the ROC curve of mock test data, the AUC of the ROC curve of all of the SLACS data (SL), the subselection of F814W filter images (814), and the subselection of F435W filter images 435 in addition to the respective precision and recall values at a threshold of 0.5. We do not provide AUC values in the F555W filter because there are no non-lens images in the filter. The AUC columns for the SLACS data include the average and standard deviation of the AUC from bootstrapping. We also include precision and recall values at the 0.5 threshold.

captured by simulations, but further tests on a larger sample of data will be necessary. The methods also scale to large data sets on a computing cluster, if needed.

One major caveat to our results is the fact that our mock data does not describe the full distribution of lens and non-lens images that will be observed, a shortcoming to be addressed in future mock data work. For example, the galaxies along the line of sight in our training and test images all come from the Hubble Ultra Deep field, sampling a smaller range of potential contaminants that are not associated with a lensed image. A limitation of the CANDELS sources is that the sources are observed with *HST* PSF, and would not resolve the clumpiness within a true *HST* arc. Also, the redshift of the source galaxies have been fixed to $z_s = 2$. Varying the source redshifts affects the relative brightness between the lens and the source. Note that in our lens-classification method, the HOG image processing step normalizes the contrast of local histograms within blocks of the image, providing an option to enable results that are more invariant to changes in brightness across the image.

Finally, we must take the class balance, or relative number of lens to non-lens systems, into account when assessing a metric. The Precision–Recall (purity-completeness) metric is sensitive to the ratio of lens and non-lens systems in the data. Both of our training and test data sets have 50% lensed and 50% unlensed images, which is not expected in observations. Again, the precision of a method is what would enable efficient spectroscopic or human-based follow-up.

For images selected for a massive elliptical, the number of non-lenses will outnumber lenses by at least 1000 to 1. Therefore, a sample that is 50% pure requires a classifier with an fpr of 0.001. Looking at the solid green line in Figure 5, we can set a high classification threshold and obtain a sample with close to 100% purity and up to ∼10% recall of all of our lenses, before contamination from non-lenses leaks into the selection. While information from other bands will certainly improve the model performance, a maximally large and pure sample from our method would likely require further filtering, e.g., by citizen science, or modification to how the HOG features are used by machine learning classifiers. However, neural networks have the current best performance in application to the strong lens finding problem and are the best single approach to the pure lens-finding problem (Metcalf et al. 2018).

The ROC curve metric is insensitive to the ratio, but is sensitive to the sampling. Given alternative lens and non-lens sample splittings, our true positive and fprs in the ROC curves would stay the same, making the ROC curve a more standard metric in the literature. On the other hand, the ROC curves show a representative rate for lens and source distributions that are evenly sampled. We do not expect this sampling to be representative of what we might expect from an observational survey. We leave these additional challenges to future work.

C.A. acknowledges support from the Enrico Fermi Institute at the University of Chicago, and the University of Chicago Provost's Office. N.L. would like to thank the funding support from NSFC, grant No. 11503064, as well as a UK Science and Technology Facilities Council research grant. W.L. thanks the funding support from Shanghai Natural Science Foundation, grant No. 15ZR1446700. This work was also supported in part by the Kavli Institute for Cosmological Physics at the University of Chicago through grant NSF PHY-1125897 and an endowment from the Kavli Foundation and its founder Fred Kavli. The authors thank the referee for a careful reading of the manuscript and constructive comments that have significantly improved the paper, Adam Bolton for providing the SLACS data, and Mike Gladders for useful discussions.

ORCID iDs

Camille Avestruz https://orcid.org/0000-0001-8868-0810
Nan Li https://orcid.org/0000-0001-6800-7389
Hanjue Zhu (朱涵珏) https://orcid.org/0000-0003-0861-0922
Thomas E. Collett https://orcid.org/0000-0001-5564-3140
Wentao Luo https://orcid.org/0000-0003-1297-6142

References

Agnello, A., Kelly, B. C., Treu, T., & Marshall, P. J. 2015, MNRAS, 448, 1446
Alard, C. 2006, arXiv:astro-ph/0606757
Allam, S. S., Tucker, D. L., Lin, H., et al. 2007, ApJL, 662, L51
Bezecourt, J., Pello, R., & Soucail, G. 1998, A&A, 330, 399
Bolton, A. S., Burles, S., Koopmans, L. V. E., et al. 2008, ApJ, 682, 964
Bom, C. R., Makler, M., Albuquerque, M. P., & Brandt, C. H. 2017, A&A, 597, A135
Bonvin, V., Courbin, F., Suyu, S. H., et al. 2017, MNRAS, 465, 4914
Brault, F., & Gavazzi, R. 2015, A&A, 577, A85
Chae, K.-H. 2003, MNRAS, 346, 746
Collett, T. E. 2015, ApJ, 811, 20
Collett, T. E., & Auger, M. W. 2014, MNRAS, 443, 969
Collett, T. E., Auger, M. W., Belokurov, V., Marshall, P. J., & Hall, A. C. 2012, MNRAS, 424, 2864
Connolly, A. J., Peterson, J., Jernigan, J. G., et al. 2010, Proc. SPIE, 7738, 77381O
Dalal, N., & Triggs, B. 2005, in IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 886
Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441
Dye, S., Evans, N. W., Belokurov, V., Warren, S. J., & Hewett, P. 2008, MNRAS, 388, 384
Estrada, J., Annis, J., Diehl, H. T., et al. 2007, ApJ, 660, 1176
Galametz, A., Grazian, A., Fontana, A., et al. 2013, ApJS, 206, 10
Gavazzi, R., Marshall, P. J., Treu, T., & Sonnenfeld, A. 2014, ApJ, 785, 144
Gavazzi, R., Treu, T., Rhodes, J. D., et al. 2007, ApJ, 667, 176
Gladders, M. D., Hoekstra, H., Yee, H. K. C., Hall, P. B., & Barrientos, L. F. 2003, ApJ, 593, 48
Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, ApJS, 197, 35
Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. 2009, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics (New York: Springer)
Jacobs, C., Glazebrook, K., Collett, T., More, A., & McCarthy, C. 2017, MNRAS, 471, 167
Joseph, R., Courbin, F., Metcalf, R. B., et al. 2014, A&A, 566, A63
Jullo, E., Natarajan, P., Kneib, J.-P., et al. 2010, Sci, 329, 924
Kneib, J.-P., & Natarajan, P. 2011, A&ARv, 19, 47
Kochanek, C. S. 1996, ApJ, 473, 595
Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, ApJS, 197, 36
Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, ApJ, 649, 599
Kubo, J. M., & Dell'Antonio, I. P. 2008, MNRAS, 385, 918
Lanusse, F., Ma, Q., Li, N., et al. 2018, MNRAS, 473, 3895
Lee, C.-H. 2017, PASA, 34, 14
Lenzen, F., Schindler, S., & Scherzer, O. 2004, A&A, 416, 391
Li, N., & Chen, D.-M. 2009, RAA, 9, 1173
Li, N., Gladders, M. D., Rangel, E. M., et al. 2016, ApJ, 828, 54
Linder, E. V. 2004, PhRvD, 70, 043534
Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179
Lynds, R., & Petrosian, V. 1986, BAAS, 18, 1014
Marshall, P. J., Hogg, D. W., Moustakas, L. A., et al. 2009, ApJ, 694, 924
Marshall, P. J., Verma, A., More, A., et al. 2016, MNRAS, 455, 1171
Maturi, M., Mizera, S., & Seidel, G. 2014, A&A, 567, A111
Metcalf, R. B., Meneghetti, M., Avestruz, C., et al. 2018, arXiv:1802.03609
Miralda-Escude, J., & Lehar, J. 1992, MNRAS, 259, 31P
More, A., Verma, A., Marshall, P. J., et al. 2016, MNRAS, 455, 1191
Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579
Paraficz, D., Courbin, F., Tramacere, A., et al. 2016, A&A, 592, A75

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2012, arXiv:1201.0490
Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2017, MNRAS, 472, 1129
Seidel, G., & Bartelmann, M. 2007, A&A, 472, 341
Sharon, K., Ofek, E. O., Smith, G. P., et al. 2005, ApJL, 629, L73
Soler, J. D., Beuther, H., Rugel, M., et al. 2019, A&A, 622, A166
Suyu, S. H., Bonvin, V., Courbin, F., et al. 2017, MNRAS, 468, 2590
Suyu, S. H., Treu, T., Hilbert, S., et al. 2014, ApJL, 788, L35
Walsh, D., Carswell, R. F., & Weymann, R. J. 1979, Natur, 279, 381
Warren, S. J., & Dye, S. 2003, ApJ, 590, 673
Xu, B., Postman, M., Meneghetti, M., et al. 2016, ApJ, 817, 85
Zahid, H. J., Damjanov, I., Geller, M. J., & Chilingarian, I. 2015, ApJ, 806, 122