MACHINE LEARNING APPROACHES TO STAR-GALAXY CLASSIFICATION

BY

JUNHYUNG KIM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Emeritus Jon J. Thaler, Chair
Professor Robert J. Brunner, Director of Research
Professor George D. Gollin
Assistant Professor Alexander G. Schwing

Abstract

Accurate star-galaxy classification has many important applications in modern precision cosmology. However, a vast number of faint sources that are detected in the current and next-generation ground-based surveys may be challenged by poor star-galaxy classification. Thus, we explore a variety of machine learning approaches to improve star-galaxy classification in ground-based photometric surveys. In Chapter 2, we present a meta-classification framework that combines existing star-galaxy classifiers, and demonstrate that our Bayesian combination technique improves the overall performance over any individual classification method. In Chapter 3, we show that a deep learning algorithm called convolutional neural networks is able to produce accurate and well-calibrated classifications by learning directly from the pixel values of photometric images. In Chapter 4, we study another deep learning technique called generative adversarial networks in a semi-supervised setting, and demonstrate that our semi-supervised method produces competitive classifications using only a small amount of labeled examples.

Acknowledgments

First and formorest, I would like to thank my advisor, Robert Brunner, for his guidance and support during my graduate studies at UIUC. I also wish to thank my committee members, Jon Thaler, George Gollin, and Alex Schwing, for their help and support. I am also grateful to Matias Carrasco Kind, Yiran Wang, Nacho Sevilla, William Biscarri, Samantha Thrush, Xinyang Lu, and Kelechi Ikegwu. A special thanks to my father and my wife who endured this long process with me.

— Edward Junhyung Kim

This work was supported in part by the National Science Foundation Grant No. AST-1313415. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. This work is based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. CFHTLenS data processing was made possible thanks to significant computing support from the NSERC Research Tools and Instruments grant program. Funding for the DEEP2 survey has been provided by NSF grants AST-0071048,

AST-0071198, AST-0507428, and AST-0507483. Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, and the U.S. Department of Energy Office of Science. The SDSS-III web site is http://www.sdss3.org/. This work used data from the VIMOS Public Extragalactic Redshift Survey (VIPERS). VIPERS has been performed using the ESO Very Large Telescope, under the "Large Programme" 182.A-0886. The participating institutions and funding agencies are listed at http://vipers.inaf.it/. This work used data from the VIMOS VLT Deep Survey, obtained from the VVDS database operated by Cesam, Laboratoire d'Astrophysique de Marseille, France.

Table of Contents

Chapter 1

Introduction

1.1 Star-galaxy Classification in Photometric Surveys

Currently ongoing and upcoming large-scale surveys, such as the Dark Energy Survey (DES) and the Large Synoptic Survey Telescope (LSST), are purely photometric surveys, where digital images of the sky are obtained and subsequently analyzed. To quantify the brightness of a source in a photometric image, we count the number of photons from the source within a fixed aperture (e.g., a circle or a two-dimensional Gaussian). This brightness measurement is expressed in units of magnitude — a logarithmic unit in which the fainter a source appears the larger its magnitude. In mathematical terms, the apparent magnitude m in a spectral band λ is given by

$$m_\lambda = -2.5 \log_{10} \frac{F_\lambda}{F_{\lambda,0}}, \tag{1.1}$$

where F_λ is the observed flux using the photometric filter λ , and $F_{\lambda,0}$ is the reference flux (i.e., zero-point) for that filter. Photometric surveys use filters on telescopes to allow only light around a specific wavelength to pass. Figure 1.1 shows the wavelengths of the five filters (named u, g, r, i, z) of the Sloan Digital Sky Survey (SDSS). Before these photometric data can be used for a scientific analysis, however, they must be classified, which for most sources is either a star or a galaxy.

Stars are in our Milky Way galaxy and are close to us compared to distant galaxies. Due to their small physical size, however, almost all stars appear as compact point sources in photometric images. Galaxies, despite being farther away, generally subtend a larger angle, and

thus appear as extended sources. However, as Figure 1.2 demonstrates, it becomes increasingly difficult to separate stars from galaxies due to a large number of unresolved galaxies at faint magnitudes. Since the number of galaxies grows exponentially with magnitude, this implies that the majority of sources that are detected in the current and next-generation ground-based surveys may be challenged by poor star-galaxy classification. Furthermore, due to the sheer number of stars and galaxies, this classification has to be automated. For example, the SDSS has obtained photometric observations of more than 3×10^8 objects (Eisenstein et al., 2011), and the LSST will produce a catalog of 2×10^{10} galaxies and a similar number of stars (Ivezic et al., 2008). Thus, there is a need for a robust, automated classification technique for large ground-based photometric surveys.

The classification of stars vs. galaxies has many important applications in precision cosmology. As a basic example, in a homogeneous universe with a Euclidean geometry for three-dimensional space, the number counts of galaxies as a function of magnitude follows

$$N\left(m_\lambda\right) \propto 10^{0.6\left(m_\lambda - m_{\lambda,0}\right)}. \tag{1.2}$$

By comparing this relation with the predictions of a Friedmann-Robertson-Walker (FRW) universe (i.e., the standard model of cosmology), Yasuda et al. (2001) show that our universe does not have a Euclidean geometry for three-dimensional space. Without a reliable method for separating stars from unresolved galaxies, we risk underestimating the number density of galaxies by rejecting all unresolved galaxies, while including them could result in significant contamination of the galaxy sample. Furthermore, the accurate separation of stars and galaxies in faint samples significantly improves our ability to (i) measure auto-correlation functions of luminous galaxies (Ross et al., 2011), (ii) control the systematic errors in the weak lensing shear measurement (Soumagnac et al., 2015), (iii) map the signature of baryon acoustic oscillations (Anderson et al., 2014), and (iv) identify electromagnetic counterparts to gravitational wave sources (Miller et al., 2017), among other things.

Given the importance of this classification problem, it is not surprising that a variety of different strategies have been developed. The most commonly used method to classify stars and galaxies in large sky surveys is the morphological separation (Sebok, 1979; Kron, 1980; Valdes, 1982; Yee, 1991; Vasconcellos et al., 2011; Henrion et al., 2011). It relies on the assumption that stars appear as point sources while galaxies appear as resolved sources. For example, a popular technique in the weak lensing community (Kaiser et al., 1995) makes a hard cut in the space of photometric attributes as shown in Figure 1.3. As the Figure shows, there is a distinct locus produced by point sources in the half-light radius vs. the i-band magnitude plane. (The half-light radius is the effective radius at which half of the total light of an object is contained.) A rectangular cut in this size-magnitude plane separates point sources (which are presumed to be stars) from resolved sources (which are presumed to be galaxies).

However, such a hard cut in a low-dimensional parameter space has disadvantages: it does not break down gracefully; its treatment of measurement uncertainties is too simplistic; it uses a rather limited subset of the full information available; and it ignores a priori information like the expected demographics of the source populations. Furthermore, currently ongoing and upcoming large photometric surveys will detect a vast number of unresolved galaxies at faint magnitudes. Near a survey's limit, the photometric observations cannot reliably separate stars from unresolved galaxies by morphology alone without leading to incompleteness and contamination in the star and galaxy samples.

1.2 Machine Learning

1.2.1 Supervised Learning

The systematic misclassification of sources can be mitigated by using machine learning algorithms. Machine learning methods have the advantage that it is easier to include extra information, such as shape information or different model magnitudes. Machine learning

techniques are usually categorized into two main types: supervised and unsupervised learning approaches. In the supervised learning approach, the input attributes (i.e., the values that describe the properties of each objects e.g., magnitudes), $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, are provided along with the desired output values (e.g., star or galaxy), $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$, in a labeled set of input-output pairs $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. Here, $\mathbf{D}$ is the training set, and N is the number of training examples. The goal of supervised learning is then to estimate a function that maps $f : \mathbf{X} \to \mathbf{y}$. As we discuss in the following chapters, it is desirable for the algorithm to return a probability. To emphasize the need for probabilistic predictions, we formulate the goal of supervised learning as folows: a probabilistic supervised learning algorithm infers the probability distribution $P(\mathbf{y}|\mathbf{X}, \mathbf{D})$ over possible labels, given the input $\mathbf{X}$ and training set $\mathbf{D}$. We use the conditioning bar | to explicitly show that the probability is conditional on both the input $\mathbf{X}$ and the training set $\mathbf{D}$. When we have a set of multiple models to choose from, we explicitly condition the probability on the set of models and write $P(\mathbf{y}|\mathbf{X}, \mathbf{D}, \mathbf{M})$, where $\mathbf{M}$ is the set of models. However, if it is clear from the context which model we use to make predictions, we drop $\mathbf{M}$ although it is implied that the probability is conditional on the form of model.

To obtain the truth labels for the training data, we use spectroscopy to measure the spectrum of electromagnetic radiation from stars and galaxies. Although modern spectrometers are more complex, a spectrometer, in its most basic form, consists of a slit, a prism or diffraction grating (to split the light into its component colors), and a detector. We can use spectroscopy to measure many properties of distant stars and galaxies, such as their chemical composition, temperature, and distance, and thus spectral classification can be used as the ground truth for classifying sources in photometric images.

1.2.2 Neural Networks

As an example of a supervised learning algorithm, we provide a brief description of *artificial neural networks* (ANN)—the most widely used machine learning algorithm in astronomy.

The use of neural networks in astronomy goes as far back as the mid 1980s (Jeffrey and Rosner, 1986). ANN was first applied to the star-galaxy classification problem by Odewahn et al. (1992), and it has become a core part of the popular astronomical image processing software SExtractor (Bertin and Arnouts, 1996).

The original motivation for ANNs was to simulate neurons in the human brain. A neuron in the human brain receives signals from other neurons through synaptic connections. If the combination of these signals exceeds a certain threshold, the neuron will fire and send a signal to other neurons. Intelligence is believed to be the collective effect of approximately 10^{11} neurons firing. An artificial neuron in most artificial neural networks is represented as a mathematical function that models a biological neural structure (Figure 1.4a). Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a vector of inputs to a given neuron, $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ be a vector of weights, and b be the bias. Then, the output of the neuron is

$$y = \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right), \tag{1.3}$$

where σ is the activation function (or *non-linearity*). Common activation functions include the sigmoid function,

$$\sigma(x) = 1/\left(1 + e^{-x}\right), \tag{1.4}$$

the hyperbolic tangent function,

$$\sigma(x) = \tanh(x), \tag{1.5}$$

and the rectified linear unit (ReLU; Nair and Hinton, 2010),

$$\sigma(x) = \max(0, x). \tag{1.6}$$

Typical neurons are organized as layers, where each neuron in one layer is connected to the neurons of the subsequent layer. A schematic representation is shown in Figure 1.4b. All layers except the input and output layers are conveniently called hidden layers.

The training uses an algorithm to a set of weights and biases such that, given N samples, the output from the network $\mathbf{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$ approximates the desired output $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ as closely as possible for all input $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. We can formulate this as the minimization of a loss function $L(\mathbf{y}, \hat{\mathbf{y}})$ over the training data. A common form of the loss function is the *cross-entropy*,

$$L(y_j, \hat{y}_j) = -\frac{1}{N} \sum_{j=1}^{N} y_j \log_2 \hat{y}_j + (1 - y_j) \log_2 (1 - \hat{y}_j). \tag{1.7}$$

where y_j is the actual truth value (e.g., 0 or 1) of the j-th data, and $\hat{y}_j$ is the probability prediction made by the model.

To find the weights $\mathbf{w}$ and biases $\mathbf{b}$ which minimize the loss, we use a technique called *gradient descent*, where we use the following rules to update the parameters in each layer l:

$$\begin{aligned} \mathbf{w}_l \to \mathbf{w}_l' &= \mathbf{w}_l - \eta \frac{\partial L}{\partial \mathbf{w}_l} \\ \mathbf{b}_l \to \mathbf{b}_l' &= \mathbf{b}_l - \eta \frac{\partial L}{\partial \mathbf{b}_l}, \end{aligned} \tag{1.8}$$

where η is a small, positive number known as the *learning rate*. The gradients in 1.8 can be computed using the *backpropagation* procedure (Rumelhart et al., 1988), which is nothing more than an application of the chain rule for derivatives.

1.2.3 Unsupervised and Semi-supervised Learning

In contrast to supervised learning, in which the truth labels are provided, unsupervised learning does not utilize the desired output during the learning process. Instead, we are only given unlabeled inputs $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^{N}$, and the data is clustered into different classes or categories. In other words, unsupervised learning attempts to infer the probability distribution of the form $P(\mathbf{x}_i)$. Unsupervised machine learning techniques are less common, in part due to the successes of purely supervised learning. Semi-supervised learning falls between

supervised learning, where training data are completely labeled, and unsupervised learning, where all training data are unlabeled. Semi-supervised techniques make use of a large amount of unlabeled data, in conjunction with a small amount of labeled data, to better capture the underlying data distribution. We expect unsupervised and semi-supervised learning to become more important, since it is unclear if sufficient training data will be available in future ground-based photometric surveys and there will be many orders of magnitude more unlabeled than labeled data available in future ground-based imaging surveys.

1.3 Thesis Structure

In the following chapters, we explore a variety of statistical and machine learning approaches to push the limits of star-galaxy classification in ground-based photometric surveys. Each chapter is self-contained and has its own references.

In Chapter 2, we present a novel meta-classification framework that combines and fully exploits different techniques to produce a more robust star-galaxy classification. To demonstrate this hybrid, ensemble approach, we combine a purely morphological classifier, a supervised machine learning method based on random forest, an unsupervised machine learning method based on self-organizing maps, and a hierarchical Bayesian template fitting method. Using data from the Canada-France-Hawaii Telescope Lensing Survey (CFHTLenS), we consider different scenarios: when a high-quality training set is available with spectroscopic labels, and when the demographics of sources in a low-quality training set do not match the demographics of objects in the test data set. We demonstrate that our Bayesian combination technique improves the overall performance over any individual classification method in these scenarios.

In Chapter 3, we present a star-galaxy classification framework that uses a supervised machine learning algorithm called convolutional neural networks (ConvNets). Most existing star-galaxy classifiers use the reduced summary information from catalogs, requiring careful

feature extraction and selection. Deep ConvNets allow a machine to automatically learn the features directly from images, minimizing the need for input from human experts. Using data from the SDSS and CFHTLenS, we demonstrate that ConvNets are able to produce accurate and well-calibrated probabilistic classifications that are competitive with conventional machine learning techniques.

In Chapter 4, we study the application of a deep learning technique called generative adversarial networks (GANs) to the star-galaxy classification problem in a semi-supervised setting. As current and forthcoming photometric surveys probe large cosmological volumes, the majority of photometric observations are too faint for a uniform spectroscopic follow-up. As a result, the number of unlabeled data available for training machine learning algorithms will be orders of magnitude greater than the number of labeled data. Semi-supervised learning techniques are of great interest since they are able to capture the underlying data distribution with only a small amount of labeled data. Using photometric images from the SDSS, we demonstrate that semi-supervised GANs are able to produce accurate and well-calibrated classifications using only a small amount of labeled examples. We also show that the number count distributions of the images generated by GAN follow a similar distribution to the SDSS photometric sample.

In Chapter 5, we outline our conclusions.

1.4 Figures and Tables

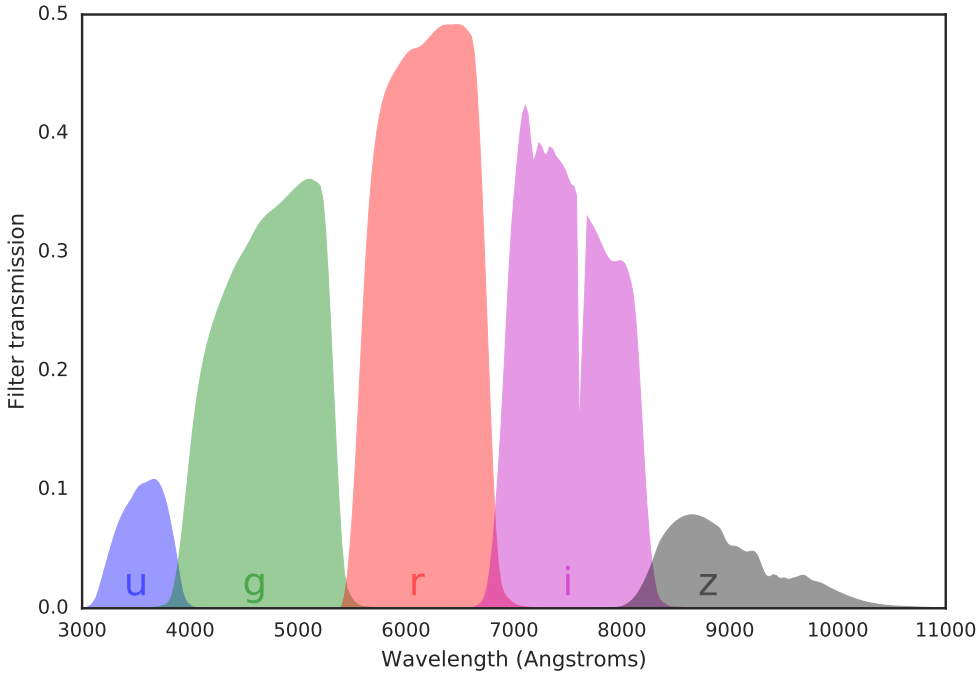


Figure 1.1: The SDSS *ugriz* filter transmission curves.

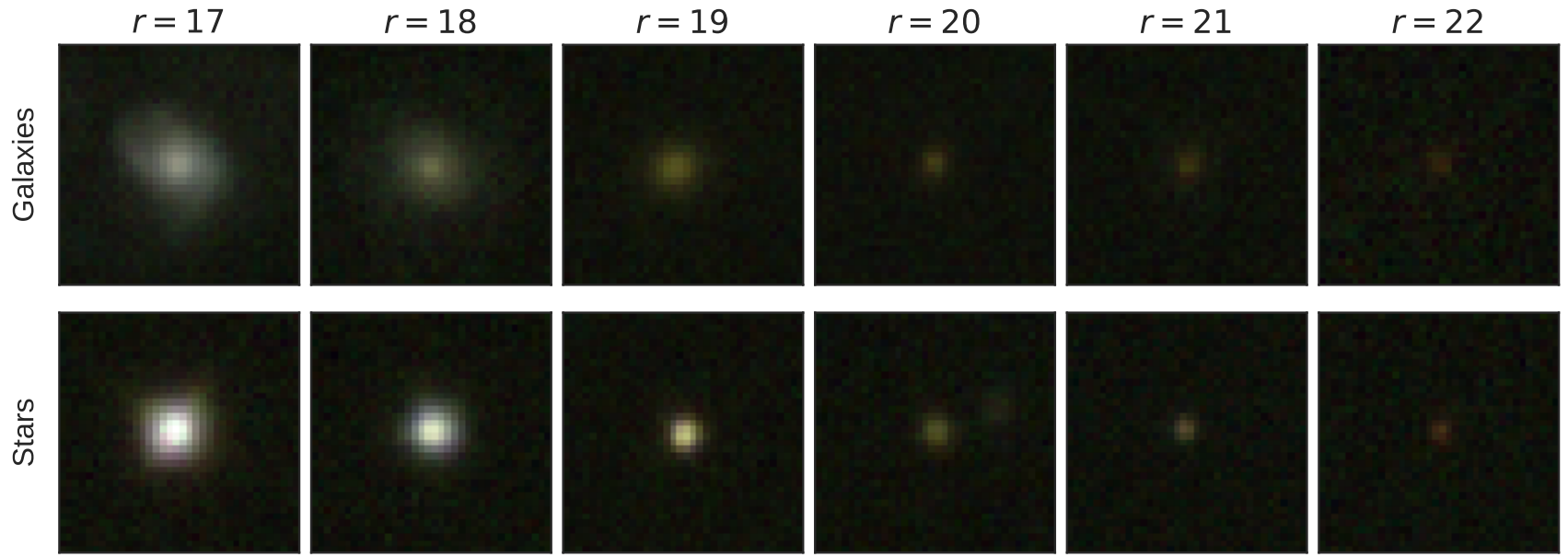


Figure 1.2: Sample images of stars (top row) and galaxies (bottom row) from the SDSS survey at different magnitudes (r-band). Note that it becomes increasingly difficult to classify sources at fainter magnitudes, where we have the majority of the detected sources.

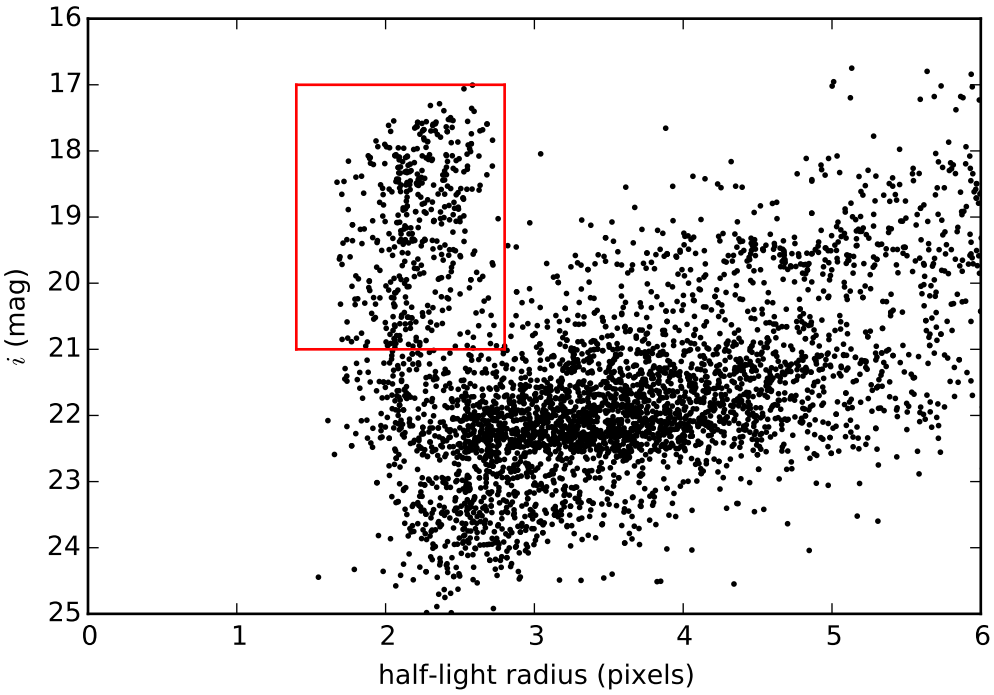
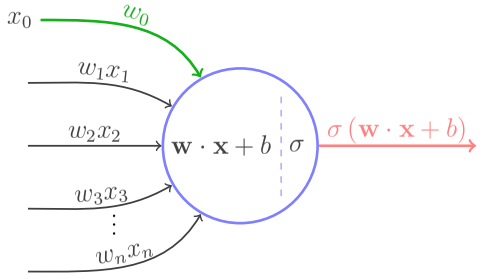
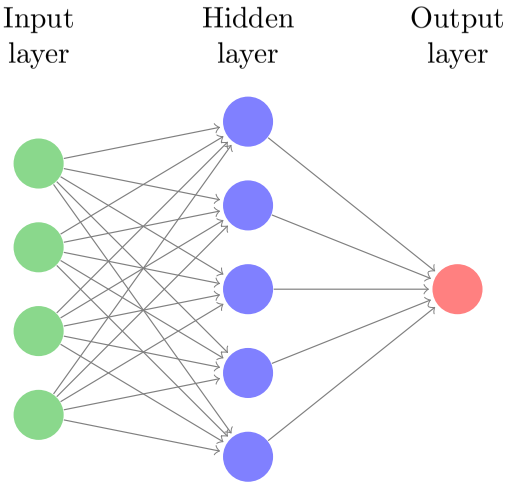


Figure 1.3: Half-light radius vs. magnitude.



(a)



(b)

Figure 1.4: (a) A mathematical model of a biological neuron. (b) A schematic diagram of a neural network with one hidden layer.

1.5 References

Lauren Anderson, Éric Aubourg, Stephen Bailey, Florian Beutler, Vaishali Bhardwaj, Michael Blanton, Adam S Bolton, J Brinkmann, Joel R Brownstein, Angela Burden, et al. The clustering of galaxies in the sdss-iii baryon oscillation spectroscopic survey: baryon acoustic oscillations in the data releases 10 and 11 galaxy samples. *Monthly Notices of the Royal Astronomical Society*, 441(1):24–62, 2014.

E. Bertin and S. Arnouts. Sextractor: Software for source extraction. *A&AS*, 117:393–404, June 1996.

Daniel J Eisenstein, David H Weinberg, Eric Agol, Hiroaki Aihara, Carlos Allende Prieto, Scott F Anderson, James A Arns, Éric Aubourg, Stephen Bailey, Eduardo Balbinot, et al. Sdss-iii: Massive spectroscopic surveys of the distant universe, the milky way, and extra-solar planetary systems. *AJ*, 142(3):72, 2011.

M. Henrion, D. J. Mortlock, D. J. Hand, and A. Gandy. A Bayesian approach to star-galaxy classification. *MNRAS*, 412:2286–2302, April 2011.

Zeljko Ivezic, JA Tyson, B Abel, E Acosta, R Allsman, Y AlSayyad, SF Anderson, J Andrew, R Angel, G Angeli, et al. Lsst: from science drivers to reference design and anticipated data products. *arXiv preprint arXiv:0805.2366*, 2008.

W Jeffrey and R Rosner. Optimization algorithms-simulated annealing and neural network processing. *The Astrophysical Journal*, 310:473–481, 1986.

N. Kaiser, G. Squires, and T. Broadhurst. A Method for Weak Lensing Observations. *ApJ*, 449:460, August 1995.

R. G. Kron. Photometry of a complete sample of faint galaxies. *ApJS*, 43:305–325, June 1980.

AA Miller, MK Kulkarni, Y Cao, RR Laher, FJ Masci, and JA Surace. Preparing for advanced ligo: A star–galaxy separation catalog for the palomar transient factory. *AJ*, 153(2):73, 2017.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys, and W. A. Zumach. Automated star/galaxy discrimination with neural networks. *AJ*, 103:318–331, January 1992.

AJ Ross et al. Ameliorating systematic uncertainties in the angular clustering of galaxies: a study using the sdss-iii. *MNRAS*, 417(2):1350–1373, 2011.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

W. L. Sebok. Optimal classification of images into stars or galaxies - A Bayesian approach. *AJ*, 84:1526–1536, October 1979.

M. T. Soumagnac et al. Star/galaxy separation at faint magnitudes: Application to a simulated dark energy survey. *MNRAS*, 450:666–680, jun 2015.

F. Valdes. Resolution classifier. In *Instrumentation in Astronomy IV*, volume 331 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 465–472, October 1982.

E. C. Vasconcellos, R. R. de Carvalho, R. R. Gal, F. L. LaBarbera, H. V. Capelato, H. Frago Campos Velho, M. Trevisan, and R. S. R. Ruiz. Decision Tree Classifiers for Star/Galaxy Separation. *AJ*, 141:189, June 2011.

Naoki Yasuda, Masataka Fukugita, Vijay K Narayanan, Robert H Lupton, Iskra Strateva, Michael A Strauss, Željko Ivezić, Rita SJ Kim, David W Hogg, David H Weinberg, et al. Galaxy number counts from the sloan digital sky survey commissioning data. *The Astronomical Journal*, 122(3):1104, 2001.

H. K. C. Yee. A faint-galaxy photometry and image-analysis system. *PASP*, 103:396–411, April 1991.

Chapter 2

A Hybrid Ensemble Learning Approach to Star-galaxy Classification

2.1 Introduction

The problem of source classification is fundamental to astronomy and goes as far back as Messier (1781). A variety of different strategies have been developed to tackle this long-standing problem, and yet there is no consensus on the optimal star-galaxy classification strategy. The most commonly used method to classify stars and galaxies in large sky surveys is the morphological separation (Sebok, 1979; Kron, 1980; Valdes, 1982; Yee, 1991; Vasconcellos et al., 2011; Henrion et al., 2011). It relies on the assumption that stars appear as point sources while galaxies appear as resolved sources. However, currently ongoing and upcoming large photometric surveys, such as the Dark Energy Survey (DES) and the Large Synoptic Survey Telescope (LSST), will detect a vast number of unresolved galaxies at faint magnitudes. Near a survey's limit, the photometric observations cannot reliably separate stars from unresolved galaxies by morphology alone without leading to incompleteness and contamination in the star and galaxy samples.

The contamination of unresolved galaxies can be mitigated by using training based algorithms. Machine learning methods have the advantage that it is easier to include extra

This chapter contains material from the following previously published article:

- E. J. Kim, R. J. Brunner, and M. Carrasco Kind. A hybrid ensemble learning approach to star-galaxy classification. *MNRAS*, 453(1):507–521, 2015

information, such as concentration indices, shape information, or different model magnitudes. However, they are only reliable within the limits of the training data, and it can be difficult to extrapolate these algorithms outside the parameter range of the training data. These techniques can be further categorized into supervised and unsupervised learning approaches.

In supervised learning, the input attributes (e.g., magnitudes or colors) are provided along with the truth labels (e.g., star or galaxy). Odewahn et al. (1992) pioneered the application of neural networks to the star-galaxy classification problem, and it has become a core part of the astronomical image processing software SExtractor (Bertin and Arnouts, 1996). Other successfully implemented examples include decision trees (Weir et al., 1995; Suchkov et al., 2005; Ball et al., 2006; Sevilla-Noarbe and Etayo-Sotos, 2015) and Support Vector Machines (Fadely, Hogg, and Willman, 2012). Unsupervised machine learning techniques are less common, as they do not utilize the truth labels during the training process, and only the input attributes are used.

Physically based template fitting methods have also been used for the star-galaxy classification problem (Robin et al., 2007; Fadely et al., 2012). Template fitting approaches infer a source's properties by finding the best match between the measured set of magnitudes (or colors) and the synthetic set of magnitudes (or colors) computed from a set of spectral templates. Although it is not necessary to obtain a high-quality spectroscopic training sample, these techniques do require a representative sample of theoretical or empirical templates that span the possible spectral energy distributions (SEDs) of stars and galaxies. Furthermore, they are not exempt from uncertainties due to measurement errors on the filter response curves, or from mismatches between the observed magnitudes and the template SEDs.

In this chapter, we present a novel star-galaxy classification framework that combines and fully exploits different classification techniques to produce a more robust classification. In particular, we show that the combination of a morphological separation method, a template fitting technique, a supervised machine learning method, and an unsupervised ma-

chine learning algorithm can improve the overall performance over any individual method. In Section 2.2, we describe each of the star-galaxy classification methods. In Section 2.3, we describe different classification combination techniques. In Section 2.4, we describe the Canada-France Hawaii Telescope Lensing Survey (CFHTLenS) data set with which we test the algorithms. In Section 2.5, we compare the performance of our combination techniques to the performance of the individual classification techniques. Finally, we outline our conclusions in Section 2.6.

2.2 Classification Methods

In this section, we present four distinct star-galaxy classification techniques. The first method is a morphological separation method, which uses a hard cut in the half-light radius vs. magnitude plane. The second method is a supervised machine learning technique named TPC (Trees for Probabilistic Classification), which uses prediction trees and a random forest (Carrasco Kind and Brunner, 2013). The third method is an unsupervised machine learning technique named SOMc, which uses self-organizing maps (SOMs) and a random atlas to provide a classification (Carrasco Kind and Brunner, 2014b). The fourth method is a Hierarchical Bayesian (HB) template fitting technique based on the work by Fadely et al. (2012), which fits SED templates from star and galaxy libraries to an observed set of measured flux values.

Collectively, these four methods represent the majority of all standard star-galaxy classification approaches published in the literature. It is very likely that any new classification technique would be functionally similar to one of these four methods. Therefore, any of these four methods could in principle be replaced by a similar method.

2.2.1 Morphological Separation

The simplest and perhaps the most widely used approach to star-galaxy classification is to make a hard cut in the space of photometric attributes. As a first-order morphological selection of point sources, we adopt a technique that is popular among the weak lensing community (Kaiser, Squires, and Broadhurst, 1995). As Figure 2.1 shows, there is a distinct locus produced by point sources in the half-light radius (estimated by SExtractor's FLUX_RADIUS parameter) vs. the i-band magnitude plane. A rectangular cut in this size-magnitude plane separates point sources, which are presumed to be stars, from resolved sources, which are presumed to be galaxies. The boundaries of the selection box are determined by manually inspecting the size-magnitude diagram.

One of the disadvantages of such cut-based methods is that it classifies every source with absolute certainty. It is difficult to justify such a decisive classification near a survey's magnitude limits, where measurement uncertainties generally increase. A more informative approach is to provide probabilistic classifications. Although a recent work by Henrion et al. (2011) implemented a probabilistic classification using a Bayesian approach on the morphological measurements alone, here we use a cut-based morphological separation to demonstrate the advantages of our combination techniques. In particular, we later show that the binary outputs (i.e., 0 or 1) of cut-based methods can be transformed into probability estimates by combining them with the probability outputs from other probabilistic classification techniques, such as TPC, SOMc, and HB.

2.2.2 Supervised Machine Learning: TPC

TPC is a parallel, supervised machine learning algorithm that uses prediction trees and random forest techniques (Breiman et al., 1984; Breiman, 2001) to produce a star-galaxy classification. TPC is a part of a publicly available software package called MLZ (Machine Learning for Photo-z). The full software package includes: TPZ, a supervised photomet-

ric redshift (photo-z) estimation technique (regression mode; Carrasco Kind and Brunner, 2013); TPC, a supervised star-galaxy classification technique (classification mode); SOMz, an unsupervised photo-z technique (regression mode; Carrasco Kind and Brunner, 2014b); and SOMc, an unsupervised star-galaxy classification technique (classification mode).

TPC uses classification trees, a type of prediction trees that are designed to provide a classification or predict a discrete category. Prediction trees are built by asking a sequence of questions that recursively split the data into branches until a terminal leaf is created that meets a stopping criterion (e.g., a minimum leaf size). The optimal split dimension is decided by choosing the attribute that maximizes the *Information Gain* (I_G), which is defined as

$$I_G\left(D_{\rm node}, X\right) = I_d\left(D_{\rm node}\right) - \sum_{x \in {\rm values}(X)} \frac{|D_{{\rm node},x}|}{|D_{\rm node}|} I_d\left(D_{{\rm node},x}\right), \tag{2.1}$$

where $D_{\rm node}$ is the training data in a given node, X is one of the possible dimensions (e.g., magnitudes or colors) along which the node is split, and x are the possible values of a specific dimension X. $|D_{\rm node}|$ and $|D_{{\rm node},x}|$ are the size of the total training data and the number of objects in a given subset x within the current node, respectively. I_d is the impurity degree index, and TPC can calculate I_d from any of the three standard different impurity indices: *information entropy*, *Gini impurity*, and *classification error*. In this work, we use the information entropy, which is defined similarly to the thermodynamic entropy:

$$I_d\left(D\right) = -f_g \log_2 f_g - \left(1 - f_g\right) \log_2\left(1 - f_g\right), \tag{2.2}$$

where f_g is the fraction of galaxies in the training data. At each node in our tree, we scan all dimensions to identify the split point that maximizes the information gain as defined by Equation 2.1, and select the attribute that maximizes the impurity index overall.

In a technique called random forest, we create bootstrap samples (i.e., N randomly selected objects with replacement) from the input training data by sampling repeatedly

from the magnitudes and colors using their measurement errors. We use these bootstrap samples to construct multiple, uncorrelated prediction trees whose individual predictions are aggregated to produce a star-galaxy classification for each source.

We also use a cross-validation technique called Out-of-Bag (OOB; Breiman et al., 1984; Carrasco Kind and Brunner, 2013). When a tree (or a map) is built in TPC (or SOMc), a fraction of the training data, usually one-third, is left out and not used in training the trees (or maps). After a tree is constructed using two-thirds of the training data, the final tree is applied to the remaining one-third to make a classification. This process is repeated for every tree, and the predictions from each tree are aggregated for each object to make the final star-galaxy classification. We emphasize that if an object is used for training a given tree, it is never used for subsequent prediction by that tree. Thus, the OOB data is an unbiased estimation of the errors and can be used as cross-validation data as long as the OOB data remain similar to the final test data set. The OOB technique can also provide extra information such as a ranking of the relative importance of the input attributes used in the prediction. The OOB technique can prove extremely valuable when calibrating the algorithm, when deciding which attributes to incorporate in the construction of the trees, and when combining this approach with other techniques.

2.2.3 Unsupervised Machine Learning: SOMc

A self-organizing map (Kohonen, 1990, 2001) is an unsupervised, artificial neural network algorithm that is capable of projecting high-dimensional input data onto a low-dimensional map through a process of competitive learning. In astronomical applications, the high-dimensional input data can be magnitudes, colors, or some other photometric attributes. The output map is usually chosen to be two-dimensional so that the resulting map can be used for visualizing various properties of the input data. The differences between a SOM and typical neural network algorithms are that a SOM is unsupervised, there are no hidden layers and therefore no extra parameters, and it produces a direct mapping between the training

set and the output network. In fact, a SOM can be viewed as a non-linear generalization of a principal component analysis (PCA) algorithm (Yin, 2008).

The key characteristic of SOM is that it retains the topology of the input training set, revealing correlations between input data that are not obvious. The method is unsupervised: the user is not required to specify the desired output during the creation of the lower-dimensional map, and the mapping of the components from the input vectors is a natural outcome of the competitive learning process.

During the construction of a SOM, each node on the two-dimensional map is represented by weight vectors of the same dimension as the number of attributes used to create the map itself. In an iterative process, each object in the input sample is individually used to correct these weight vectors. This correction is determined so that the specific neuron (or node), which at a given moment best represents the input source, is modified along with the weight vectors of that node's neighboring neurons. As a result, this sector within the map becomes a better representation of the current input object. This process is repeated for every object in the training data, and the entire process is repeated for several iterations. Eventually, the SOM converges to its final form where the training data is separated into groups of similar features. Although the spectroscopic labels are not used at all in the learning process, they are used (only after the map has been constructed) to generate predictions for each cell in the resulting two-dimensional map.

In a similar approach to random forest in TPZ and TPC, SOMz uses a technique called *random atlas* to provide photo-z estimation (Carrasco Kind and Brunner, 2014b). In random atlas, the prediction trees of random forest are replaced by maps, and each map is constructed from different bootstrap samples of the training data. Furthermore, we create random realizations of the training data by perturbing the magnitudes and colors by their measurement errors. For each map, we can either use all available attributes, or randomly select a subsample of the attribute space. This SOM implementation can also be applied to the classification problem, and we refer to it as SOMc in order to differentiate it from the

photo-z estimation problem (regression mode). We also use the random atlas approach in some of the classification combination approaches as discussed in Section 2.3.

One of the most important parameter in SOMc is the topology of the two-dimensional SOM, which can be rectangular, hexagonal, or spherical. In our SOM implementation, it is also possible to use periodic boundary conditions for the non-spherical cases. The spherical topology is by definition periodic and is constructed by using HEALPIX (Górski et al., 2005). Similar to TPC, we use the OOB technique to make an unbiased estimation of errors. We determine the optimal parameters by performing a grid search in the parameter space of different toplogies, as well as other SOM parameters, for the OOB data. We find that the spherical topology gives the best performance for the CFHTLenS data, likely due to its natural periodicity. Thus, we use a spherical topology to classify stars and galaxies in the CFHTLenS data. For a complete description of the SOM implementation and its application to the estimation of photo-z probability density functions (photo-z PDFs), we refer the reader to Carrasco Kind and Brunner (2014b).

2.2.4 Template Fitting: Hierarchical Bayesian

One of the most common methods to classify a source based on its observed magnitudes is template fitting. Template fitting algorithms do not require a spectroscopic training sample; there is no need for additional knowledge outside the observed data and the template SEDs. However, any incompleteness in our knowledge of the template SEDs that fully span the possible SEDs of observed sources may lead to misclassification of sources.

Bayesian algorithms use Bayesian inference to quantify the relative probability that each template matches the input photometry and determine a probability estimate by computing the posterior that a source is a star or a galaxy. In this work, we have modified and parallelized a publicly available Hierarchical Bayesian (HB) template fitting algorithm by Fadely et al. (2012). In this section, we provide a brief description of the HB template fitting technique; for the details of the underlying HB approach, we refer the reader to Fadely et al.

(2012).

We write the posterior probability that a source is a star as

$$P\left(S|\mathbf{x},\theta\right) = P\left(\mathbf{x}|S,\theta\right) P\left(S|\theta\right), \tag{2.3}$$

where $\mathbf{x}$ represents a given set of observed magnitudes,. We have also introduced the *hyperparameter* θ, a nuisance parameter that characterizes our uncertainty in the prior distribution. To compute the likelihood that a source is a star, we marginalize over all star and galaxy templates $\mathbf{T}$. In a template-fitting approach, we marginalize by summing up the likelihood that a source has the set of magnitudes $\mathbf{x}$ for a given star template as well as the likelihood for a given galaxy template:

$$P\left(\mathbf{x}|S,\theta\right) = \sum_{t\in\mathbf{T}} P\left(\mathbf{x}|S,t,\theta\right) P\left(t|S,\theta\right). \tag{2.4}$$

The likelihood of each template $P\left(\mathbf{x}|S,\theta\right)$ is itself marginalized over the uncertainty in the template-fitting coefficient. Furthermore, for galaxy templates, we introduce another step that marginalizes the likelihood by redshifting a given galaxy template by a factor of $1+z$.

Marginalization in Equation 2.4 requires that we specify the prior probability $P\left(t|S,\theta\right)$ that a source has a spectral template t (at a given redshift). Thus, the probability that a source is a star (or a galaxy) is either the posterior probability itself if a prior is used, or the likelihood itself if an uninformative prior is used. In a Bayesian analysis, it is preferable to use a prior, which can be directly computed either from physical assumptions, or from an empirical function calibrated by using a spectroscopic training sample. In an HB approach, the entire sample of sources is used to infer the prior probabilities for each individual source.

Since the templates are discrete in both SED shape and physical properties, we parametrize the prior probability of each template as a discrete set of weights such that

$$\sum_{t \in \mathbf{T}} P\left(t|S,\theta\right) = 1. \tag{2.5}$$

Similarly, we also parametrize the overall prior probability, $(S|\theta)$, in Equation 2.3, as a weight. These weights correspond to the hyperparameters, which can be inferred by sampling the posterior probability distribution in the hyperparameter space. For the sampling, we use EMCEE, a Python implementation of the affine-invariant Markov Chain Monte Carlo (MCMC) ensemble sampler (Foreman-Mackey et al., 2013).

As the goal of template fitting methods is to minimize the difference between observed and theoretical magnitudes, this approach heavily relies on both the use of SED templates and the accuracy of the transmission functions for the filters used for particular survey. For our stellar templates, we use the empirical SED library from Pickles (1998). The Pickles library consists of 131 stellar templates, which span all normal spectral types and luminosity classes at solar abundance, as well as metal-poor and metal-rich F–K dwarf and G–K giant and supergiant stars. We supplement the stellar library with 100 SEDs from Chabrier et al. (2000), which include low mass stars and brown dwarfs with different $T_{\rm eff}$ and surface gravities. We also include four white dwarf templates of Bohlin, Colina, and Finley (1995), for a total of 235 templates in our final stellar library. For our galaxy templates, we use four CWW spectra from Coleman, Wu, and Weedman (1980), which include an Elliptical, an Sba, an Sbb, and an Irregular galaxy template. When extending an analysis to higher redshifts, the CWW library is often augmented with two star bursting galaxy templates from Kinney et al. (1996). From the six original CWW and Kinney spectra, intermediate templates are created by interpolation, for a total of 51 SEDs in our final galaxy library.

All of the above templates are convolved with the filter response curves to generate model magnitudes. These response curves consist of u, g, r, i, z filter transmission functions for the observations taken by the Canada-France Hawaii Telescope (CFHT).

2.3 Classification Combination Methods

Building on the work in the field of ensemble learning, we combine the predictions from individual star-galaxy classification techniques using four combination techniques. The main idea behind ensemble learning is to weight the predictions from individual models and combine them to obtain a prediction that outperforms every one of them individually (Rokach, 2010).

2.3.1 Unsupervised Binning

Given the variety of star-galaxy classification methods we are using, we fully expect the relative performance of the individual techniques to vary across the parameter space spanned by the data. For example, it is reasonable to expect supervised techniques to outperform other techniques in areas of parameter space that are well-populated with training data. Similarly, we can expect unsupervised approaches such as SOM or template fitting approaches to generally perform better when a training sample is either sparse or unavailable.

We therefore adopt a binning strategy similar to Carrasco Kind and Brunner (2014a). In this binning strategy, we allow different classifier combinations in different parts of parameter space by creating two-dimensional SOM representations of the full nine-dimensional magnitude-color space: u, g, r, i, z, $u-g$, $g-r$, $r-i$, and $i-z$. A SOM representation can be rectangular, hexagonal, or spherical; here we choose a 10×10 rectangular topology to facilitate visualization as shown in Figure 2.2. We note that this choice is mainly for convenience and that the optimal topology and map size would likely depend on a number of factors, such as the number of objects and attributes. For all combination methods, we use only the OOB (cross-validation) data contained in each cell to compute the relative weights for the base classifiers. The weights within individual cells are then applied to the blind test data set to make the prediction.

Furthermore, we construct a collection of SOM representations and subsequently combine

the predictions from each map into a meta-prediction. Given a training sample of N sources, we generate N_R random realizations of training data by perturbing the attributes with the measured uncertainty for each attribute. The uncertainties are assumed to be normally distributed. In this manner, we reduce the bias towards the data and introduce randomness in a systematic manner. For each random realization of a training sample, we create N_M bootstrap samples of size N to generate N_M different maps.

After all maps are built, we have a total of $N_R \times N_M$ probabilistic outputs for each of the N sources. To produce a single probability estimate for each source, we could take the mean, the median, or some other simple statistic. With a sufficient number of maps, we find that there is usually negligible difference between taking the mean and taking the median, and use the median in the following sections. We note that it is also possible to establish confidence intervals using the distribution of the probability estimates.

2.3.2 Weighted Average

The simplest approach to combine different combination techniques is to simply add the individual classifications from the base classifiers and renormalize the sum. In this case, the final probability is given by

$$P\left(S|\mathbf{x},\mathbf{M}\right) = \sum_i P\left(S|\mathbf{x},M_i\right), \tag{2.6}$$

where $\mathbf{M}$ is the set of models (TPC, SOMc, HB, and morphological separation in our work). We improve on this simple approach by using the binning strategy to calculate the weighted average of objects in each SOM cell separately for each map, and then combine the predictions from each map into a final prediction.

2.3.3 Bucket of Models (BoM)

After the multi-dimensional input data have been binned, we can use the cross-validation data to choose the best model within each bin, and use only that model within that specific bin to make predictions for the test data. We use the mean squared error (MSE; also known as Brier score (Brier, 1950)) as a classification error metric. We define MSE as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N-1}\left(y_i - \hat{y}_i\right)^2, \tag{2.7}$$

where $\hat{y}_i$ is the actual truth value (e.g., 0 or 1) of the i^{th} data, and y_i is the probability prediction made by the models. Thus, a model with the minimum MSE is chosen in each bin, and is assigned a weight of one, and zero for all other models. However, the chosen model is allowed to vary between different bins.

2.3.4 Stacking

Instead of selecting a single model that performs best within each bin, we can train a learning algorithm to combine the output values of several other base classifiers in each bin. An ensemble learning method of using a meta-classifier to combine lower-level classifiers is known as *stacking* or *stacked generalization* (Wolpert, 1992). Although any arbitrary algorithm can theoretically be used as a meta-classifier, a logistic regression or a linear regression is often used in practice. In our work, we use a single-layer multi-response linear regression algorithm, which often shows the best performance (Breiman, 1996; Ting and Witten, 1999). This algorithm is a variant of the least-square regression algorithm, where a linear regression model is constructed for each class.

2.3.5 Bayesian Model Combination

We also use a model combination technique known as Bayesian Model Combination (BMC; Monteith et al., 2011), which uses Bayesian principles to generate an ensemble combination

of different classifiers. The posterior probability that a source is a star is given by

$$P\left(S|\mathbf{x},\mathbf{D},\mathbf{M},\mathbf{E}\right)=\sum_{e\in\mathbf{E}}P\left(S|\mathbf{x},\mathbf{M},e\right)P\left(e|\mathbf{D}\right), \tag{2.8}$$

where $\mathbf{D}$ is the data set, and e is an element in the ensemble space $\mathbf{E}$ of possible model combinations. By Bayes' Theorem, the posterior probability of e given $\mathbf{D}$ is given by

$$P\left(e|\mathbf{D}\right)=\frac{P\left(e\right)}{P\left(\mathbf{D}\right)}\prod_{d\in\mathbf{D}}P\left(d|e\right)\propto P\left(e\right)\prod_{d\in\mathbf{D}}P\left(d|e\right). \tag{2.9}$$

Here, $P\left(e\right)$ is the prior probability of e, which we assume to be uniform. The product of $P\left(d|e\right)$ is over all individual data d in the training data $\mathbf{D}$, and $P\left(\mathbf{D}\right)$ is merely a normalization factor and not important.

For binary classifiers whose output is either zero or one (e.g., a cut-based morphological separation), we assume that each example is corrupted with an average error rate ϵ. This means that $P\left(d|e\right)=1-\epsilon$ if the combination e correctly predicts class $\hat{y}_i$ for the i^{th} object, and $P\left(d|e\right)=\epsilon$ if it predicts an incorrect class. The average rate ϵ can be estimated by the fraction $\left(M_g+M_s\right)/N$, where M_g is the number of true galaxies classified as stars, M_s is the number of true stars classified as galaxies, and N is the total number of sources. Equation 2.9 then becomes

$$P\left(e|\mathbf{D}\right)\propto P\left(e\right)\left(1-\epsilon\right)^{N-M_s-M_g}\left(\epsilon\right)^{M_s+M_g}. \tag{2.10}$$

For probabilistic classifiers, we can directly use the probabilistic predictions and write Equation 2.9 as

$$P\left(e|\mathbf{D}\right)\propto P\left(e\right)\prod_{i=0}^{N-1}\hat{y}_i y_i+\left(1-\hat{y}_i\right)\left(1-y_i\right). \tag{2.11}$$

Although the space $\mathbf{E}$ of potential model combinations is in principle infinite, we can produce a reasonable finite set of potential model combinations by using sampling techniques.

In our implementation, the weights of each combination of the base classifiers is obtained by sampling from a Dirichlet distribution. We first set all alpha values of a Dirichlet distribution to unity. We then sample this distribution q times to obtain q sets of weights. For each combination, we assume a uniform prior and calculate $P(e|\mathbf{D})$ using Equation 2.10 or 2.11. We select the combination with the highest $P(e|\mathbf{D})$, and update the alpha values by adding the weights of the most probable combination to the current alpha values. The next q sets of weights are drawn using the updated alpha values.

We continue the sampling process until we reach a predefined number of combinations, and finally use Equation 2.8 to compute the posterior probability that a source is a star (or a galaxy). In this work, we use a q value of three, and 1,000 model combinations are considered.

We also use a binned version of the BMC technique, where we use a SOM representation to apply different model combinations for different regions of the parameter space. We however note that introducing randomness though the construction of $N_R \times N_M$ different SOM representations does not show significant improvement over using only one single SOM representation. This similarity is likely due to the randomness that has already been introduced by sampling from the Dirichlet distribution. Thus, our BMC technique uses one SOM, while other base models (WA, BoM, and stacking) generate N_R random realizations of N_M maps.

2.4 Data

We use photometric data from the Canada-France-Hawaii Telescope Lensing Survey (CFHTLenS; Heymans et al., 2012; Erben et al., 2013; Hildebrandt et al., 2012). This catalog consists of more than twenty five million objects with a limiting magnitude of $i_{\rm AB} \approx 25.5$. It covers a total of 154 square degrees in the four fields (named W1, W2, W3, and W4) of CFHT Legacy Survey (CFHTLS; Gwyn, 2012) observed in the five photometric bands: u, g, r, i, and z.

We have cross-matched reliable spectroscopic galaxies from the Deep Extragalactic Evolutionary Probe Phase 2 (DEEP2; Davis et al., 2003; Newman et al., 2013), the Sloan Digital Sky Survey Data Release 10 (Ahn et al., 2014, SDSS-DR10), the VIsible imaging Multi-Object Spectrograph (VIMOS) Very Large Telescope (VLT) Deep Survey (VVDS; Le Fèvre et al., 2005; Garilli et al., 2008), and the VIMOS Public Extragalactic Redshift Survey (VIPERS; Garilli et al., 2014). We have selected only sources with very secure redshifts and no bad flags (quality flags -1, 3, and 4 for DEEP2; quality flag 0 for SDSS; quality flags 3, 4, 23, and 24 for VIPERS and VVDS). In the end, we have 8,545 stars and 57,843 galaxies available for the training and testing processes. We randomly select 13,278 objects for the blind testing set, and use the remainder for training and cross-validation. While HB uses only the magnitudes in the five bands, u, g, r, i, and z, TPC and SOMc are trained with a total of 9 attributes: the five magnitudes and their corresponding colors, $u-g$, $g-r$, $r-i$, and $i-z$. The morphological separation method uses SExtractor's FLUX_RADIUS parameter provided by the CFHTLenS catalog.

Our goal here is not to obtain the best classifier performance; for this we would have fine tuned individual base classifiers and chosen sophisticated models best suited to the particular properties of the CFHTLenS data. For example, Hildebrandt et al. (2012) suggest that all objects with $i > 23$ in the CFHTLenS data set may be classified as galaxies without significant incompleteness and contamination in the galaxy sample. Although this approach works because the high Galactic latitude fields of the CFHTLS contain relatively few stars, it is very unlikely that such an approach will meet the science requirements for the quality of star-galaxy classification in lower-latitude, star-crowded fields. Rather, our goal for the CFHTLenS data set is to demonstrate the usefulness of combining different classifiers even when the base classifiers may be poor or trained on partial data. We also note that the relatively few number of stars in the CFHTLS fields might paint too positive a picture of completeness and purity, especially for the stars. Thus, we caution the reader that the specific completeness and purity values will likely vary in other surveys that observe large

portions of the sky, and we emphasize once again that our aim is to highlight that there is a relative improvement in performance when we combine multiple star-galaxy classification techniques to generate a meta-classification.

2.5 Results and Discussion

In this section, we present the classification performance of the four different combination techniques, as well as the individual star-galaxy classification techniques on the CFHTLenS test data.

2.5.1 Classification Metrics

Probabilistic classification models can be considered as functions that output a probability estimate of each source to be in one of the classes (e.g., a star or a galaxy). Although the probability estimate can be used as a weight in subsequent analyses to improve or enhance a particular measurement (Ross et al., 2011), it can also be converted into a class label by using a threshold (a probability cut). The simplest way to choose the threshold is to set it to a fixed value, e.g., $p_{\rm cut} = 0.5$. This is, in fact, what is often done (e.g., Henrion et al., 2011; Fadely et al., 2012). However, choosing 0.5 as a threshold is not the best choice for an unbalanced data set, where galaxies outnumber stars. Furthermore, setting a fixed threshold ignores the operating condition (e.g., science requirements, stellar distribution, misclassification costs) where the model will be applied.

Receiver Operating Characteristic Curve

When we have no information about the operating condition when evaluating the performance of classifiers, there are effective tools such as the Receiver Operating Characteristic (ROC) curve (Swets, Dawes, and Monahan, 2000). An ROC curve is a graphical plot that illustrates the true positive rate versus the false positive rate of a binary classifier as its

classification threshold is varied. The Area Under the Curve (AUC) summarizes the curve information in a single number, and can be used as an assessment of the overall performance.

Completeness and Purity

In astronomical applications, the operating condition usually translates to the completeness and purity requirements of the star or galaxy sample. We define the galaxy *completeness* c_g (also known as recall or sensitivity) as the fraction of the number of true galaxies classified as galaxies out of the total number of true galaxies,

$$c_g = \frac{N_g}{N_g + M_g}, \tag{2.12}$$

where N_g is the number of true galaxies classified as galaxies, and M_g is the number of true galaxies classified as stars. We define the galaxy *purity* p_g (also known as precision or positive predictive value) as the fraction of the number of true galaxies classified as galaxies out of the total number of objects classified as galaxies,

$$p_g = \frac{N_g}{N_g + M_s}, \tag{2.13}$$

where M_s is the number of true stars classified as galaxies. Star completeness and purity are defined in a similar manner.

One of the advantages of a probabilistic classification is that the threshold can be adjusted to produce a more complete but less pure sample, or a less complete but more pure one. To compare the performance of probabilistic classification techniques with that of morphological separation, which has a fixed completeness ($c_g = 0.9964$, $c_s = 0.7145$) at a certain purity ($p_g = 0.9597$, $p_s = 0.9666$), we adjust the threshold of probabilistic classifiers until the galaxy completeness c_g matches that of morphological separation to compute the galaxy purity p_g at $c_g = 0.9964$. Similarly, the star purity p_s at $c_s = 0.7145$ is computed by adjusting the threshold until the star completeness of each classifier is equal to that of morphological

separation.

We can also compare the performance of different classification techniques by assuming an arbitrary operating condition. For example, weak lensing science measurements of the DES require $c_g > 0.960$ and $p_g > 0.778$ to control both the statistical and systematic errors on the cosmological parameters, and $c_s > 0.250$ and $p_s > 0.970$ for stellar Point Spread Function (PSF) calibration (Soumagnac et al., 2015). Although these values will likely be different for the science cases of the CFHTLenS data, we adopt these values to compare the classification performance at a reasonable operating condition. Thus, we compute p_g at $c_g = 0.960$ and p_s at $c_s = 0.250$. We also use the MSE defined in Equation 3.10 as a classification error metric.

2.5.2 Classifier Combination

We present in Table 2.2 the classification performance obtained by applying the four different combination techniques, as well as the individual star-galaxy classification techniques, on the CFHTLenS test data. The bold entries highlight the best technique for any particular metric. The first four rows show the performance of four individual star-galaxy classification techniques. Given a high-quality training data, it is not surprising that our supervised machine learning technique TPC outperforms other unsupervised techniques. TPC is thus shown in the first row as the benchmark.

The simplest of the combination techniques, WA and BoM, generally do not perform better than TPC. It is also interesting that, even with binning the parameter space and selecting the best model within each bin, BoM almost always chooses TPC as the best model in all bins, and therefore gives the same performance as TPC in the end. However, our BMC and stacking techniques have a similar performance and often outperform TPC. Although TPC shows the best performance as measured by the AUC, BMC shows the best performance in all other metrics.

In Figure 2.2, we show in the top left panel the mean CFHTLenS i-band magnitude in

each cell, and in the top right panel the fraction of stars in each cell. The bottom two panels show the mean $u-g$ and $g-r$ colors in each cell. These two-dimensional maps clearly show the ability of the SOM to preserve relationships between sources when it projects the full nine-dimensional space to the two-dimensional map. We note that these SOM maps should only be used to provide guidance, as the SOM mapping is a non-linear representation of all magnitudes and colors.

We can also use the same SOM from Figure 2.2 to determine the relative weights for the four individual classification methods in each cell. We present the four weight maps for the BMC technique in Figure 2.3. In these maps, a darker color indicates a higher weight, or equivalently that the corresponding classifier performs better in that region. These weight maps demonstrate the variation in the performance of the individual techniques across the two-dimensional parameter space defined by the SOM. Furthermore, since the maps in Figure 2.2 and 2.3 are constructed using the same SOM, we can determine the region in the parameter space where each individual technique performs better or worse. Not surprisingly, the morphological separation performs best in the top left corner of the weight map in Figure 2.3, which corresponds to the brightest CFHTLenS magnitudes $i \lesssim 20$ in the i-band magnitude map of Figure 2.2. It is also clear that the SOM cells where the morphological separation performs best have higher stellar fraction than the other cells. On the other hand, TPC seems to perform best in the region that corresponds to intermediate magnitudes $20 \lesssim i \lesssim 22.5$ and $1.5 \lesssim u-g \lesssim 3.0$. Our unsupervised learning method SOMc performs relatively better at fainter magnitudes $i \gtrsim 21.5$ with $0 \lesssim u-g \lesssim 0.5$ and $0 \lesssim g-r \lesssim 0.5$. Although HB shows the worst performance when there exists a high-quality training data set, BMC still utilizes information from HB, especially at intermediate magnitudes $20 \lesssim i \lesssim 22$. Another interesting pattern is that the four techniques seem complementary, and they are weighted most strongly in different regions of the SOM representation.

In Figure 2.4, we compare the star and galaxy purity values for BMC, TPC, and morphological separation as functions of i-band magnitude. We use the kernel density estimation

(KDE; Silverman, 1986) with the Gaussian kernel to smooth the fluctuations in the distribution. Although morphological separation shows a slightly better performance in galaxy purity at bright magnitudes $i \lesssim 20$, BMC outperforms both TPC and morphological separation at faint magnitudes $i \gtrsim 21$. As the top panel shows, the number count distribution peaks at $i \sim 22$, and BMC therefore outperforms both TPC and morphological separation for the majority of objects. It is also clear that BMC outperforms TPC over all magnitudes. BMC can presumably accomplish this by combining information from all base classifiers, e.g., giving more weight to the morphological separation method at bright magnitudes. The bottom panel shows that the star purity of morphological separation drops to $p_s < 0.8$ at fainter magnitudes $i > 21$. This is expected, as our crude morphological separation classifies every object as a galaxy beyond $i > 21$, and purity measures the number of true stars classified as stars. It is again clear that BMC outperforms both TPC and morphological separation in star purity values over all magnitudes.

In Figure 2.5, we show the cumulative galaxy and star purity values as functions of magnitude. Although morphological separation performs better than TPC at bright magnitudes, its purity values decrease as the magnitudes become fainter, and TPC eventually outperforms morphological separation by 1–2% at $i > 21$. BMC clearly outperforms both TPC and morphological separation, and it maintains the overall galaxy purity of 0.980 up to $i \sim 24.5$.

We also show the star and galaxy purity values as functions of photometric redshift estimate in Figure 2.6. Photo-z is estimated with the BPZ algorithm (Benítez, 2000) and provided with the CFHTLenS photometric redshift catalogue (Hildebrandt et al., 2012). The advantage of BMC over TPC and morphological separation is now more pronounced in Figure 2.6. Although the morphological separation method outperforms BMC at bright magnitudes in Figure 2.4, it is clear that BMC outperforms both TPC and morphological separation over all redshifts. We also present in Figure 2.7 how the star and galaxy purity values vary as a function of $g - r$ color. It is again clear that BMC outperforms both TPC

and morphological separation over all $g - r$ colors.

In Figure 2.8, we show the distribution of $P(S)$, the posterior probability that an object is a star, for BMC, TPC, and morphological separation. It is interesting that the BMC technique assigns a posterior star probability $P(S) \lesssim 0.3$ to significantly more true galaxies than TPC, and a probability $P(S) \gtrsim 0.8$ to significantly fewer true galaxies. By utilizing information from different types of classification techniques in different parts of the parameter space, BMC becomes more certain that an object is a star or a galaxy, resulting in improvement of overall performance.

2.5.3 Heterogeneous Training

It is very costly in terms of telescope time to obtain a large sample of spectroscopic observations down to the limiting magnitude of a photometric sample. Thus, we investigate the impact of training set quality by considering a more realistic case where the training data set is available only for a small number of objects with bright magnitudes. To emulate this scenario, we only use objects that have spectroscopic labels from the VVDS 0226-04 field (which is located within the CFHTLS W1 field) and impose a magnitude cut of $i < 22.0$ in the training data, leaving us a training set with only 1,365 objects. We apply the same four star-galaxy classification techniques and four combination methods, and measure the performance of each technique on the same test data set from Section 2.5.2. As the top two panels of Figures 2.11, 2.13, and 2.14 show, the demographics of objects in the training set are different from the distribution of sources in the test set. Thus, this also serves as a test of the efficacy of heterogeneous training.

We present in Table 2.3 the same six metrics for each method, and highlight the best method for each metric. Overall, the results obtained for the reduced data set are remarkable. With a smaller training set, our training based methods, TPC and SOMc, suffer a significant decrease in performance. The performance of morphological separation and HB is essentially unchanged from Table 2.2 as they do not depend on the training data. Without

sufficient training data, the advantage of combining the predictions of different classifiers is more obvious. Even WA, the simplest of combination techniques, outperforms all individual classification techniques in four metrics, AUC, p_s at $c_s = 0.7145$, p_g at $c_g = 0.9600$, and p_s at $c_s = 0.2500$. Although BoM always chooses TPC as the best model when we have a high-quality training set, it now chooses various methods in different bins and outperforms all base classifiers. While the performance of the stacking technique is only slightly worse than that of BMC when we have a high-quality training set, stacking now fails to outperform morphological separation. BMC shows an impressive performance and outperforms all other classification techniques in all six metrics. Overall, the improvements are small but still significant since these metrics are averaged over the full test data.

In Figure 2.10, we again show the 10×10 two-dimensional weight map defined by the SOM. When the quality of training data is relatively poor, the performance of training based algorithms will decrease, while the performance of template fitting algorithms or morphological separation methods is independent of training data. Thus, when the weight maps of Figure 2.3 and Figure 2.10 are visually compared, it is clear that the BMC algorithm now uses more information from morphological separation and HB, while it uses considerably less information from our training based algorithms, TPC and SOMc. Not surprisingly, the morphological separation method performs best at bright magnitudes, and BMC assigns more weight to HB at fainter magnitudes.

We present the star and galaxy purity values as functions of i-band magnitude in Figure 2.11. The normalized density distribution as a function of magnitude in the top panel and the stellar distribution in the second panel clearly show that the demographics of the training set and that of the test set are different. Since the training set is cut at $i < 22$, the density distribution falls off sharply around $i \sim 22$ and has a higher fraction of stars than the test set. Compared to the purity values in Figure 2.4, TPC now suffers a significant decrease in star and galaxy purity. However, the purity of BMC does not show such a significant drop and decreases by only 2–5%. As suggested by the weight maps in Figure 2.10, BMC can

accomplish this by shifting the relative weights assigned to each base classifier in different SOM cells. As the quality of training set worsens, BMC assigns less weight to training based methods and more weight to HB and morphological separation.

In Figure 2.12, we show the cumulative galaxy and star purity values as functions of magnitude. Compared to Figure 2.5, the drop in the performance of TPC is clear. However, even when some classifiers have been trained on a significantly reduced training set, BMC maintains a galaxy purity of 0.970 and a star purity of 1.0 up to $i \sim 24.5$, and it sill outperforms morphological separation at fainter magnitudes $i \gtrsim 21$.

We also show the star and galaxy purity values as functions of photo-z in Figure 2.13 and as functions of $g - r$ in Figure 2.14. Compared to Figure 2.6 and 2.7, the performance of BMC becomes worse in some photo-z and $g - r$ bins. However, this drop in performance seems to be confined to only a small number of objects in particular regions of the parameter space, and BMC still outperforms both TPC and morphological separation for the majority of objects.

Compared to Figure 2.8, the difference between the posterior star probability distribution of TPC and that of BMC is now more pronounced in Figure 2.15. The $P(S)$ distribution of BMC for true galaxies falls off sharply at $P(S) \approx 0.95$, and BMC does not assign a star probability $P(S) \gtrsim 0.95$ to any true galaxies, On the other hand, both TPC and morphological separation classify some true galaxies as stars with absolute certainty.

2.5.4 The Quality of Training Data

The combination techniques that we have demonstrated so far use two training based algorithms as base classifiers. Ideally, the training data should mirror the entire parameter space occupied by the data to be classified. Yet we have seen in Section 2.5.3 that the BMC technique does reliably extrapolate past the limits of the training data, even when some base classifiers are trained on a low-quality training data set. In this section, we further investigate if and where BMC begins to break down by imposing various magnitude, photo-z, and

color cuts to change the size and composition of the training set.

In Figure 2.16, we present a visual comparison between different classification techniques, when various magnitude cuts are applied on the training data, and the performance is measured on the same test set from Section 2.5.2 and 2.5.3. It is not surprising that the performance of TPC decreases as we decrease the size of training set by imposing more restrictive magnitude cuts, while the performance of HB and morphological separation is essentially unchanged. However, the effect of change in size and composition of the training set is significantly mitigated by the use of the BMC technique. BMC outperforms both HB and TPC in all four metrics, even when the training set is restricted to $i < 20.0$. BMC also consistently outperforms morphological separation until we impose a magnitude cut of $i < 20.0$ on the training data, beyond which point BMC finally performs worse than morphological separation. It is remarkable that BMC is able to reliably extrapolate past the training data to $i \sim 24.5$, the limiting magnitude of the test set, and outperform HB, TPC, and morphological separation in all performance metrics, even the demographics of training set do not accurately sample the data to be classified.

Similarly, we impose various spectroscopic redshift cuts on the training data in Figure 2.17. Since all stars have $z_{\rm spec}$ values close to zero, we are effectively changing the demographics of training set by keeping all stars and gradually removing galaxies with high redshifts. BMC begins to perform worse than morphological separation when a conservative cut of $z_{\rm spec} < 0.6$ is imposed. However, it is again clear that BMC is able to utilize information from HB and morphological separation to mitigate the drop in the performance of TPC.

In Figure 2.18, we decrease the size of training set by keeping red objects and gradually removing blue objects. A color cut seems to have a more pronounced effect on the performance of TPC and BMC, which perform worse than morphological separation when the training set is restricted to $g - r > 0.4$. The performance depends more strongly on the color distribution, because a significant fraction of blue objects consists of stars, while

objects with fainter magnitudes and higher redshifts are mostly galaxies. We can verify this in Figure 2.2, where the darker (higher stellar fraction) cells in the upper middle region of the stellar fraction map (top right panel) have bright magnitudes $i \lesssim 20$ in the i-band magnitude map (top left panel) and blue colors $g - r \lesssim 0.5$ in the $g - r$ color map (bottom right panel). On the other hand, the darker (fainter magnitude) cells in the right-hand side of the i-band magnitude map have almost no stars in them and are represented by bright (low stellar fraction) cells in the stellar fraction map. Thus, these results indicate that the performance of training based methods depends more strongly on the composition of training data than on the size, and it is necessary to have a sufficient number of the minority class in the training data set to ensure optimal performance.

2.6 Conclusions

We have presented and analyzed a novel star-galaxy classification framework for combining star-galaxy classifiers using the CFHTLenS data. In particular, we use four independent classification techniques: a morphological separation method; TPC, a supervised machine learning technique based on prediction trees and a random forest; SOMc, an unsupervised machine learning approach based on self-organizing maps and a random atlas; and HB, a Hierarchical Bayesian template-fitting method that we have modified and parallelized. Both TPC and SOMc algorithms are currently available within a software package named MLZ. Our implementation of HB and BMC, as well as IPython notebooks that have been used to produce the results in this work, are available at `https://github.com/EdwardJKim/astroclass`.

Given the variety of star-galaxy classification methods we are using, we fully expect the relative performance of the individual techniques to vary across the parameter space spanned by the data. We therefore adopt the binning strategy, where we allow different classifier combinations in different parts of parameter space by creating two-dimensional

self-organizing maps of the full multi-dimensional magnitude-color space. We apply different star-galaxy classification techniques within each cell of this map, and find that the four techniques are weighted most strongly in different regions of the map.

Using data from the CFHTLenS survey, we have considered different scenarios: when an excellent training set is available with spectroscopic labels from DEEP2, SDSS, VIPERS, and VVDS, and when the demographics of sources in a low-quality training set do not match the demographics of objects in the test data set. We demonstrate that the Bayesian Model Combination (BMC) technique improves the overall performance over any individual classification method in both cases. We note that Carrasco Kind and Brunner (2014a) analyzed different techniques for combining photometric redshift probability density functions (photo-z PDFs) and also found that BMC is in general the best photo-z PDF combination technique.

The problem of star-galaxy classification is a rich area for future research. It is unclear if sufficient training data will be available in future ground-based surveys. Furthermore, in large sky surveys such as DES and LSST, photometric quality is not uniform across the sky, and a purely morphological classifier alone will not be sufficient, especially at faint magnitudes. Given the efficacy of our approach, classifier combination strategies are likely the optimal approach for currently ongoing and forthcoming photometric surveys. Future studies could apply the combination technique described in this chapter to other surveys such as the DES. Our approach can also be extended more broadly to classify objects that are neither stars nor galaxies (e.g., quasars). Finally, future studies could explore the use of multi-epoch data, which would be particularly useful for the next generation of synoptic surveys.

2.7 Figures and Tables

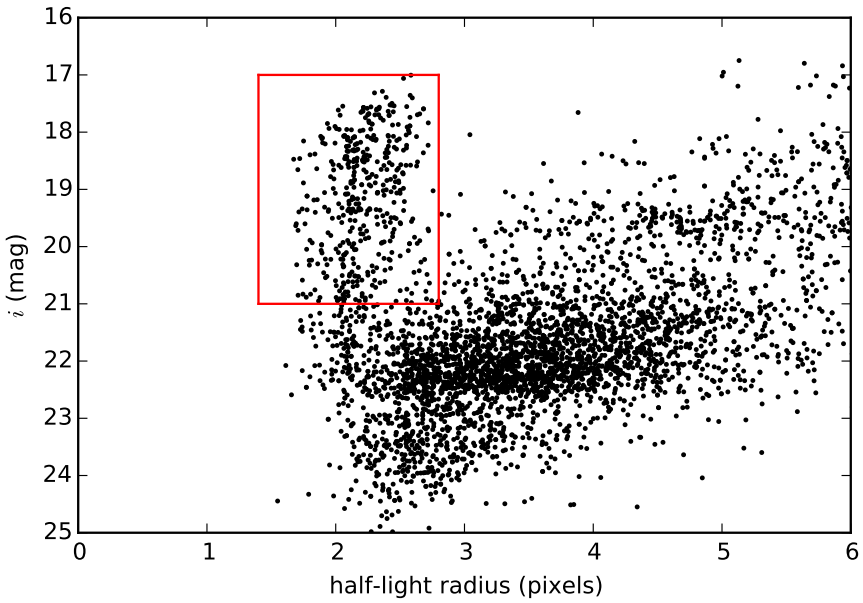


Figure 2.1: Half-light radius vs. magnitude.

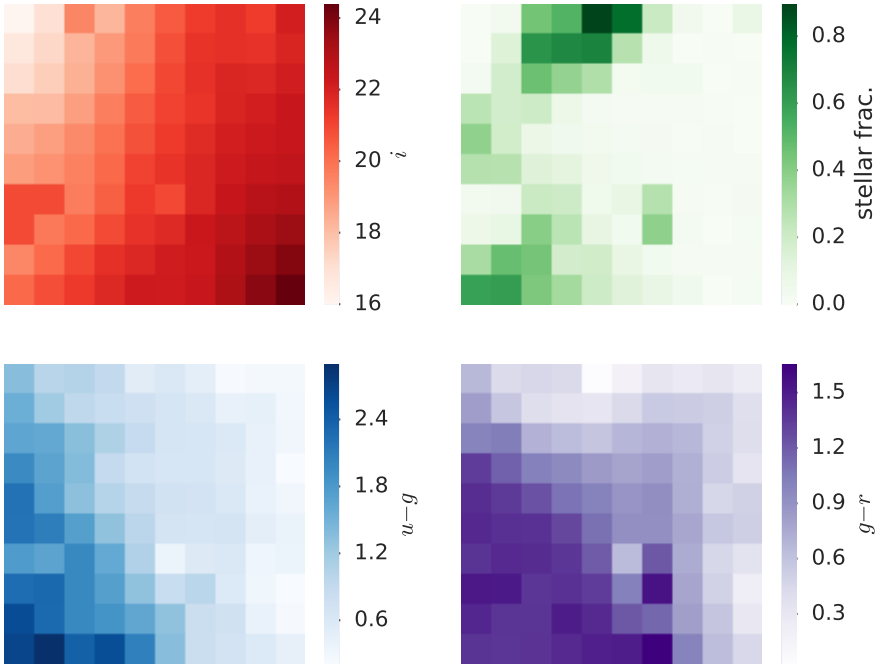


Figure 2.2: A two-dimensional 10×10 SOM representation showing the mean i-band magnitude (top left), the fraction of true stars in each cell (top right), and the mean values of $u-g$ (bottom left) and $g-r$ (bottom right) for the cross-validation data.

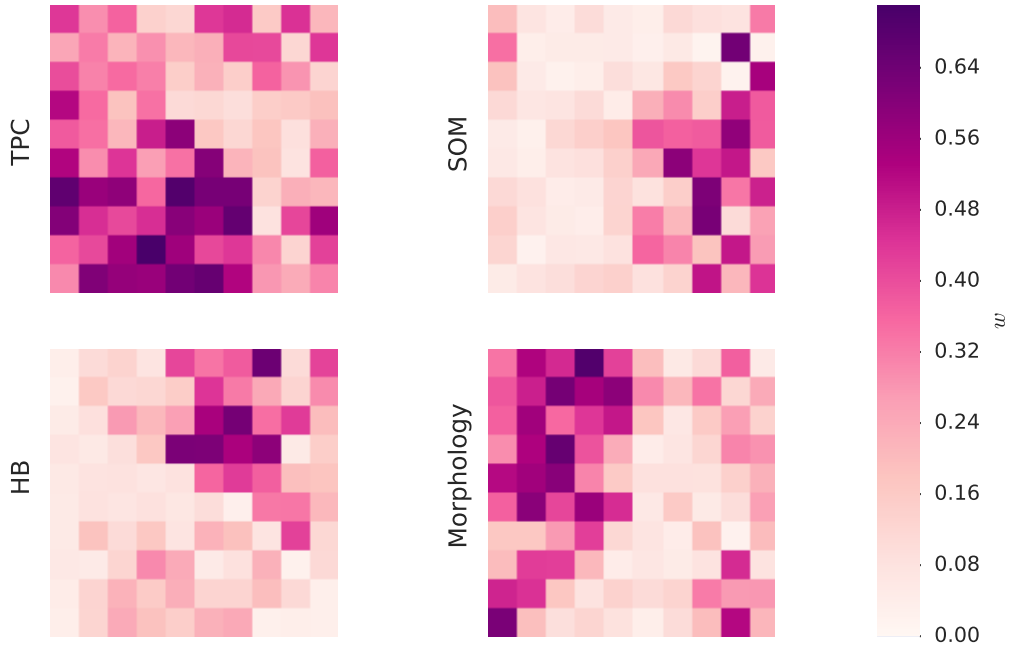


Figure 2.3: A two-dimensional 10×10 SOM representation showing the relative weights for the BMC combination technique applied to the four individual methods for the CFHTLenS data.

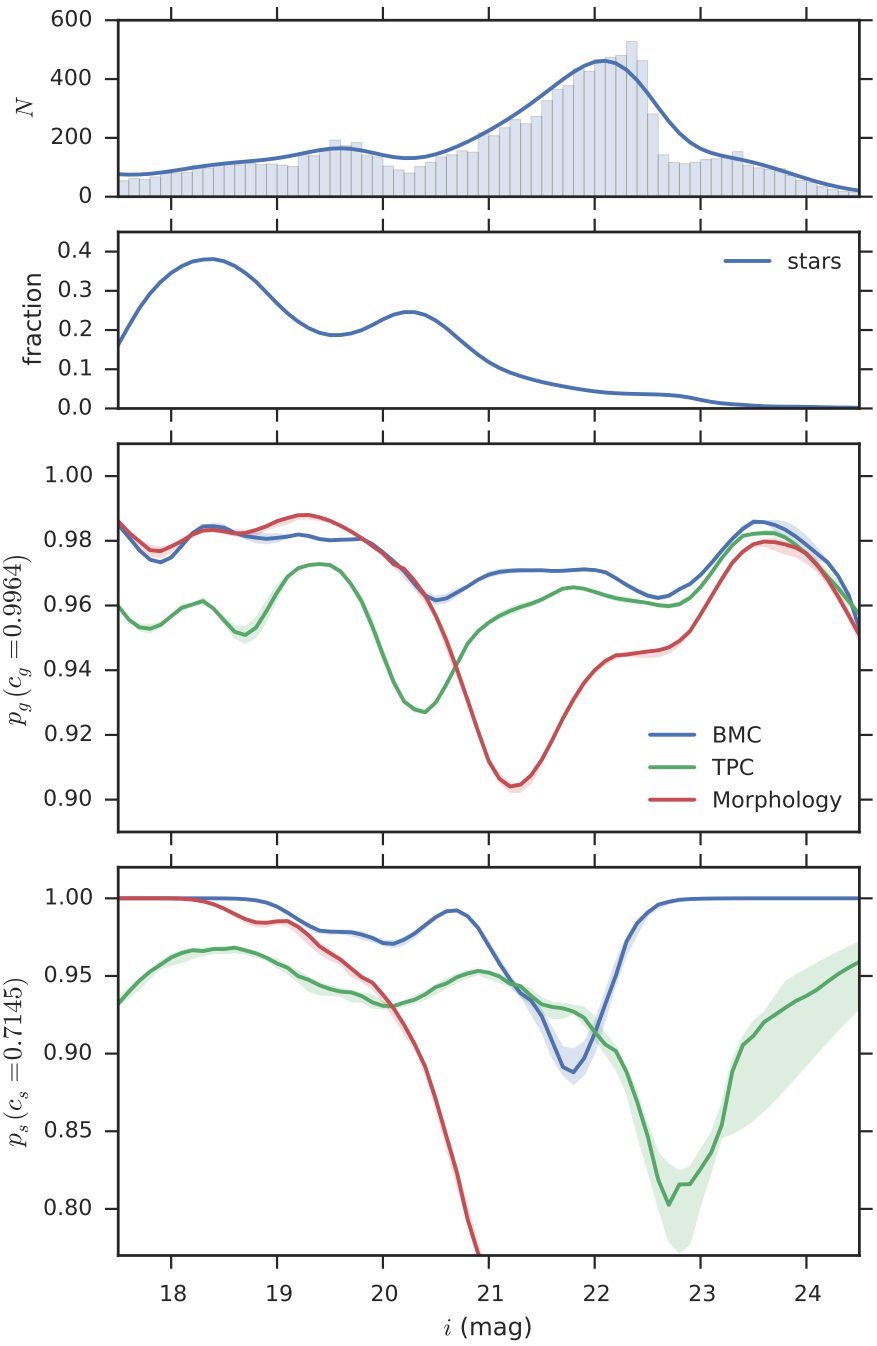


Figure 2.4: Purity as a function of the i-band magnitude as estimated by the kernel density estimation (KDE) method. The top panel shows the histogram with a bin size of 0.1 mag and the KDE for objects in the test set. The second panel shows the fraction of stars estimated by KDE as a function of magnitude. The bottom two panels compare the galaxy and star purity values for BMC, TPC, and morphological separation as functions of magnitude. Results for BMC, TPC, and morphological separation are in blue, green, and red, respectively. The 1σ confidence bands are estimated by bootstrap sampling.

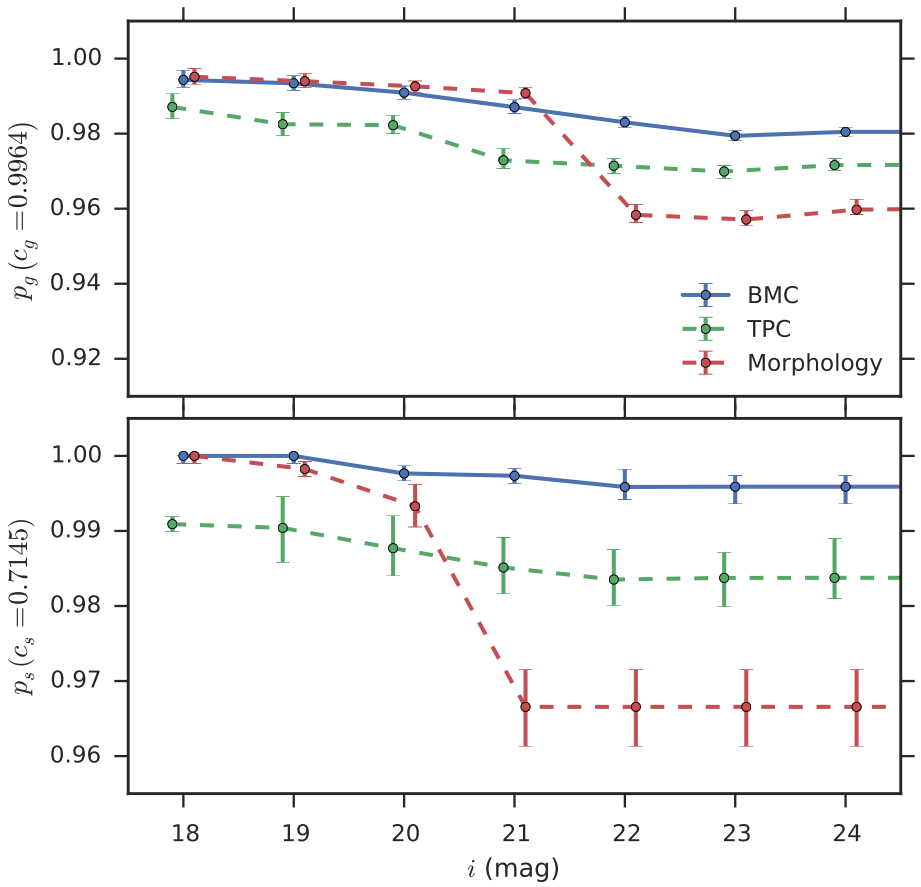


Figure 2.5: Cumulative purity as a function of the i-band magnitude. The upper panel compares the galaxy purity values for BMC (blue solid line), TPC (green dashed line), and morphological separation (red dashed line). The lower panel compares the star purity. The 1σ error bars are computed following the method of Paterno, M (2003) to avoid the unphysical errors of binomial or Poisson statistics.

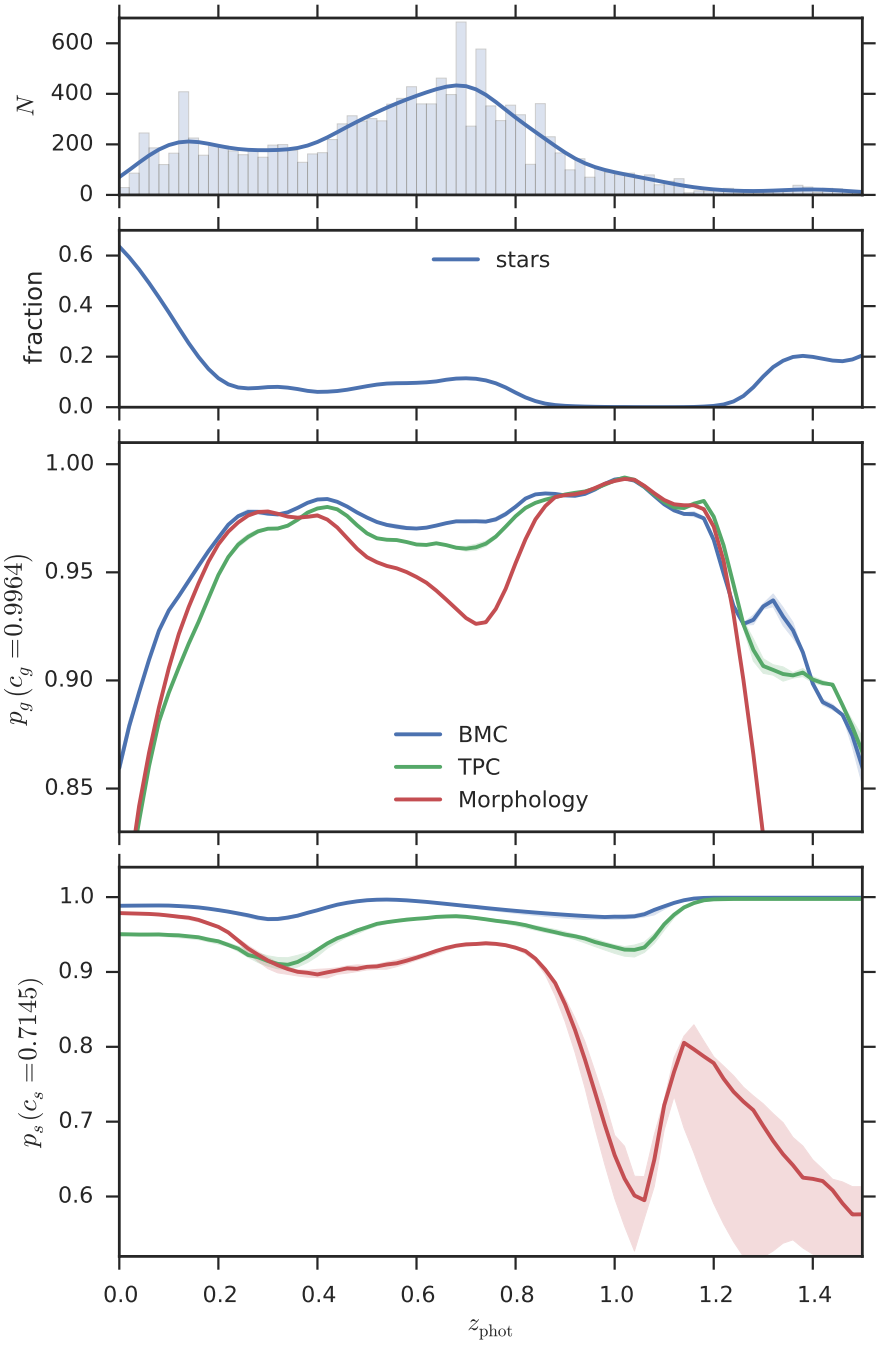


Figure 2.6: Similar to Figure 2.4 but as a function of photo-z. The bin size of histogram in the top panel is 0.02.

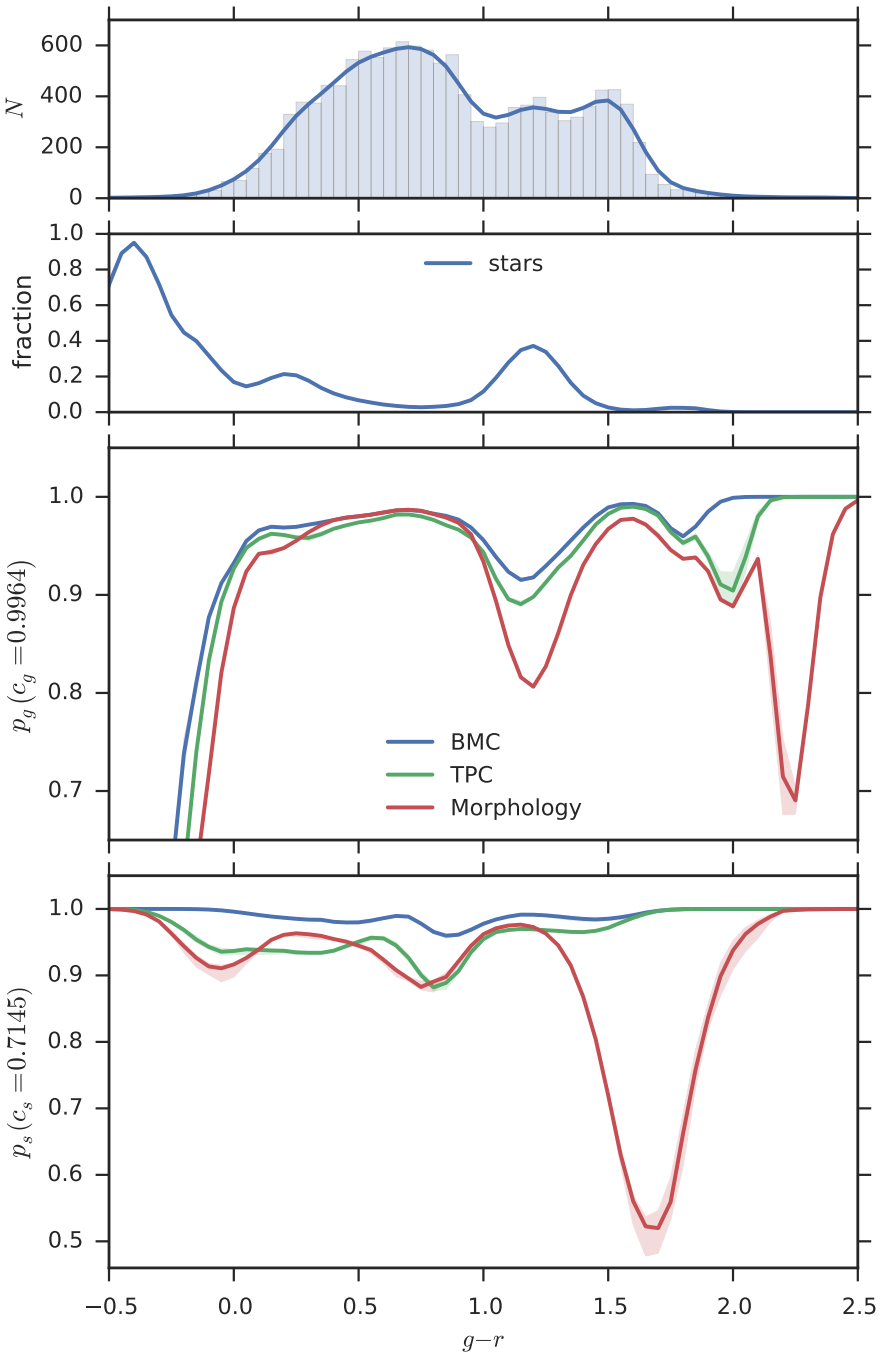


Figure 2.7: Similar to Figure 2.4 but as a function of $g-r$ color. The bin size of histogram in the top panel is 0.05.

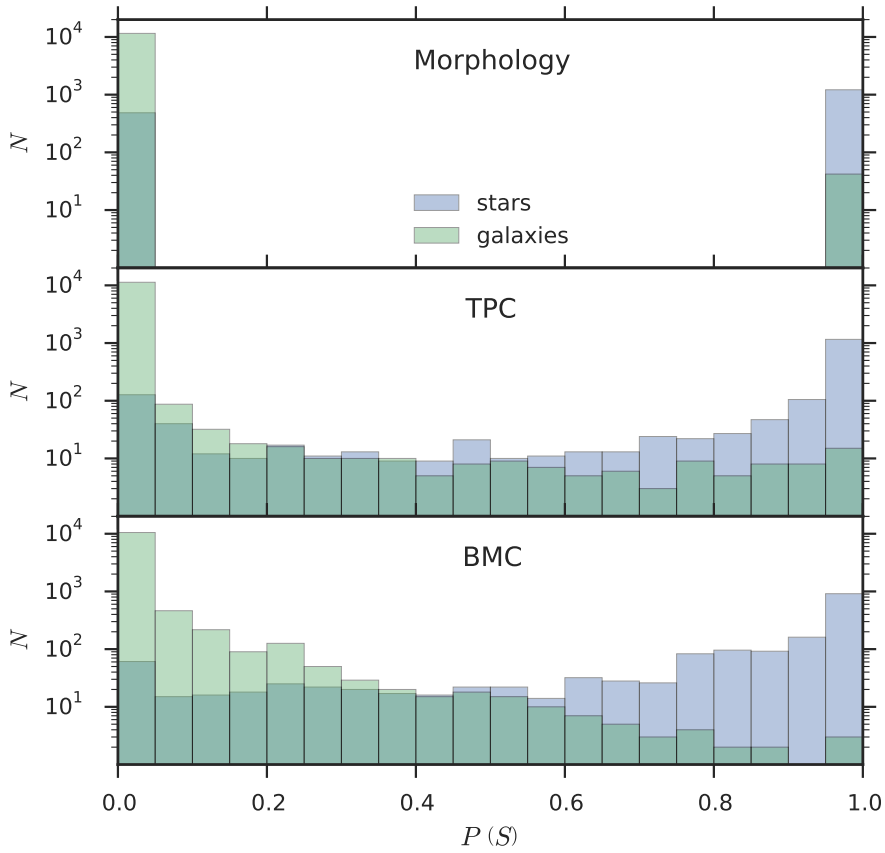


Figure 2.8: Histogram of the posterior probability that a source is a star for morphological separation (top), TPC (middle), and BMC (bottom) for a high-quality training data set. The true galaxies are in green, and true stars are in blue. The bin size is 0.05.

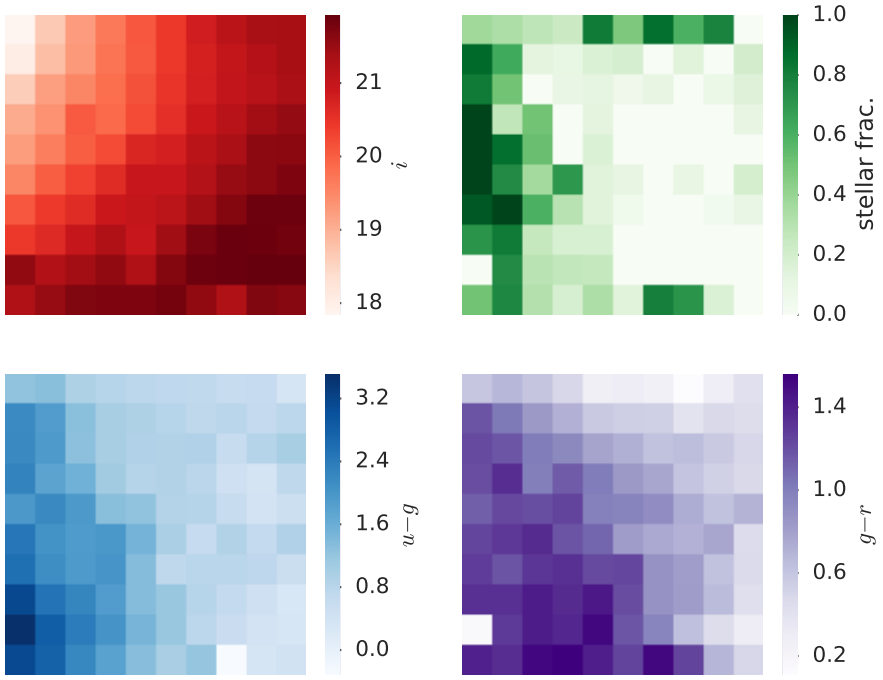


Figure 2.9: Similar to Figure 2.2 but for the reduced training data set.

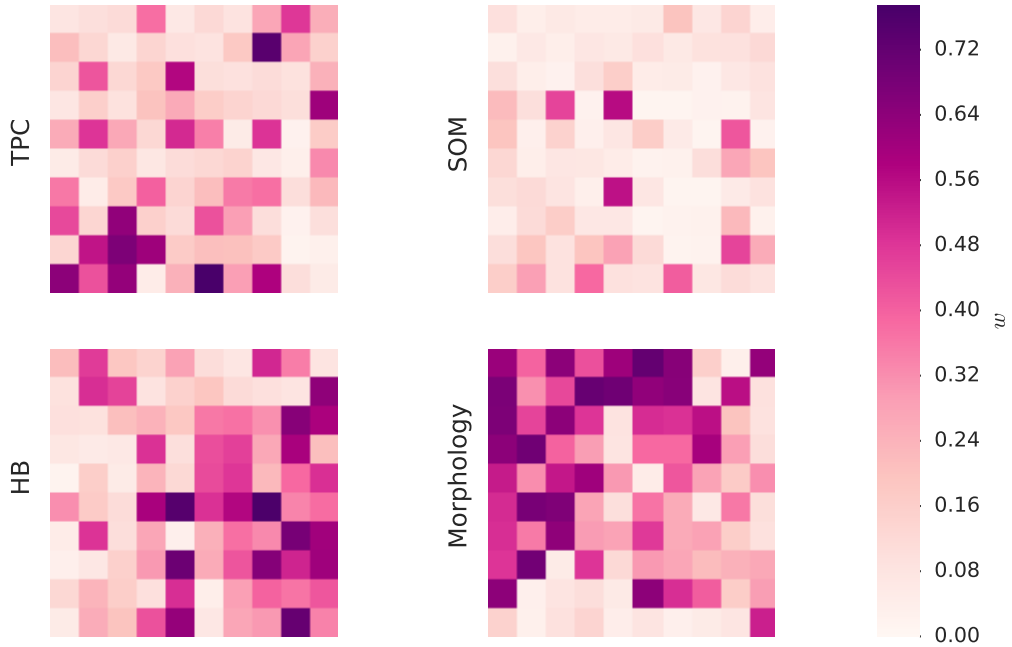


Figure 2.10: Similar to Figure 2.3 but for the reduced training data set.

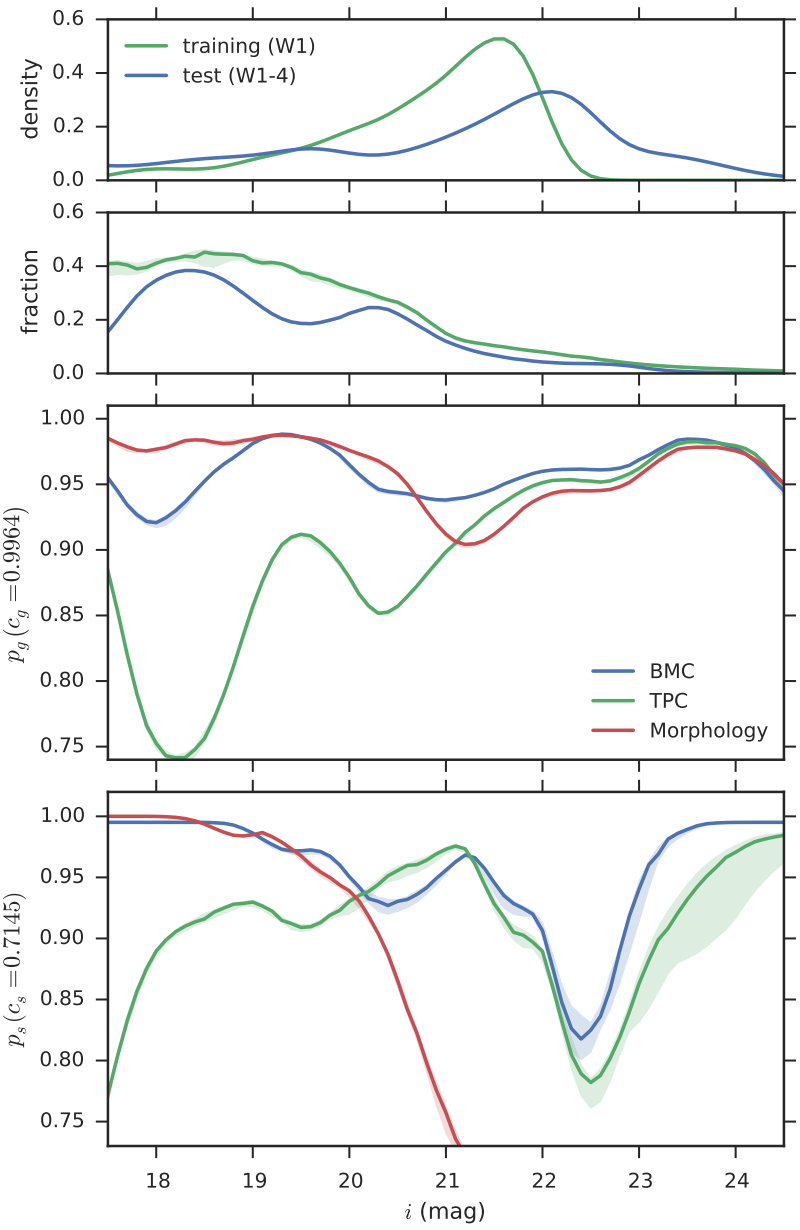


Figure 2.11: Purity as a function of the i-band magnitude for the reduced training data set. Top panel shows the histograms and KDEs for the number count distribution for the training (blue) and test (green) data set. The second panel shows the fraction of stars in the training and test data set in blue and green, respectively. The bottom two panels compare the galaxy and star purity values for BMC, TPC, and morphological separation as functions of i-band magnitude.

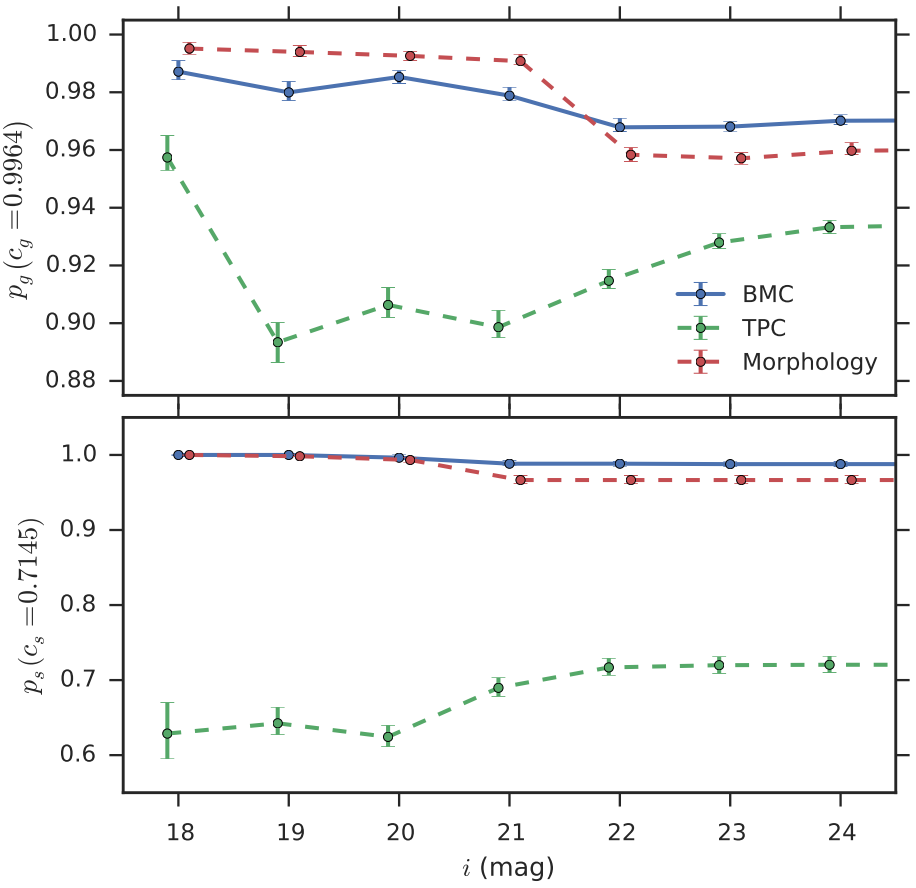


Figure 2.12: Similar to Figure 2.5 but for the reduced training data set.

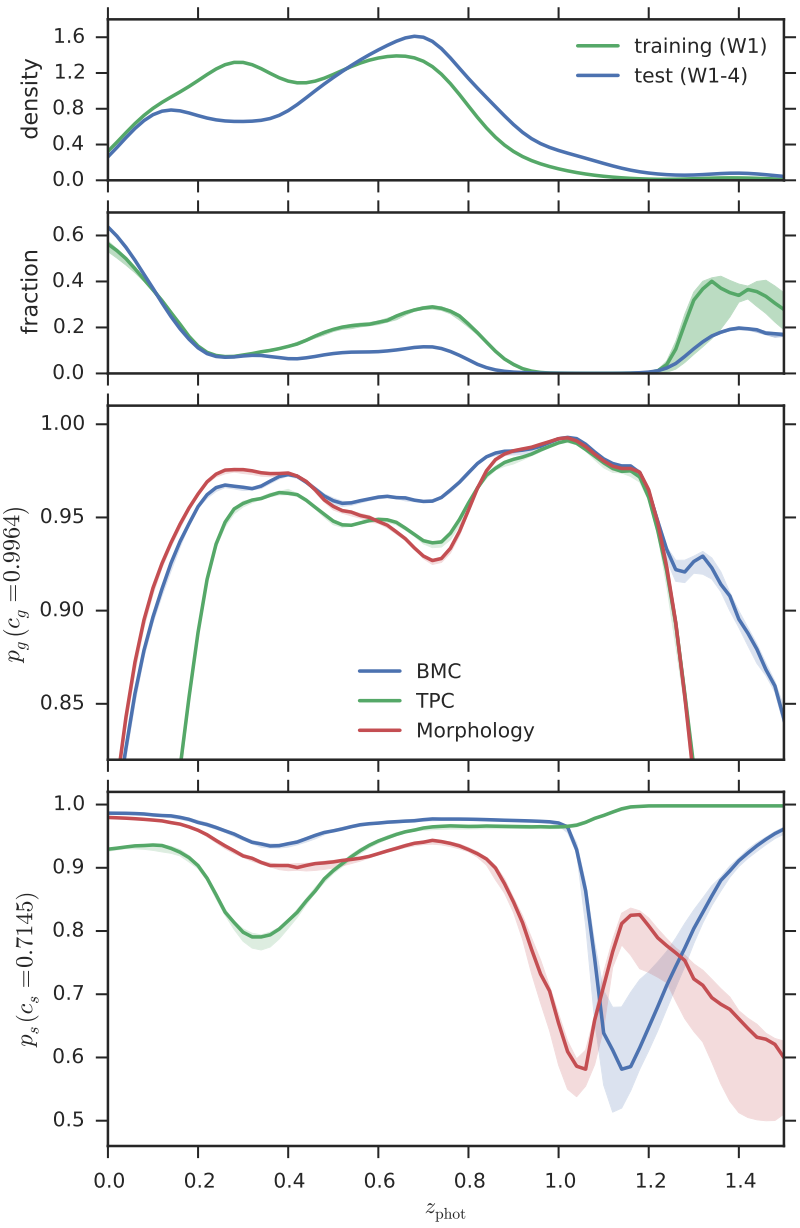


Figure 2.13: Similar to Figure 2.11 but as a function of photo-z.

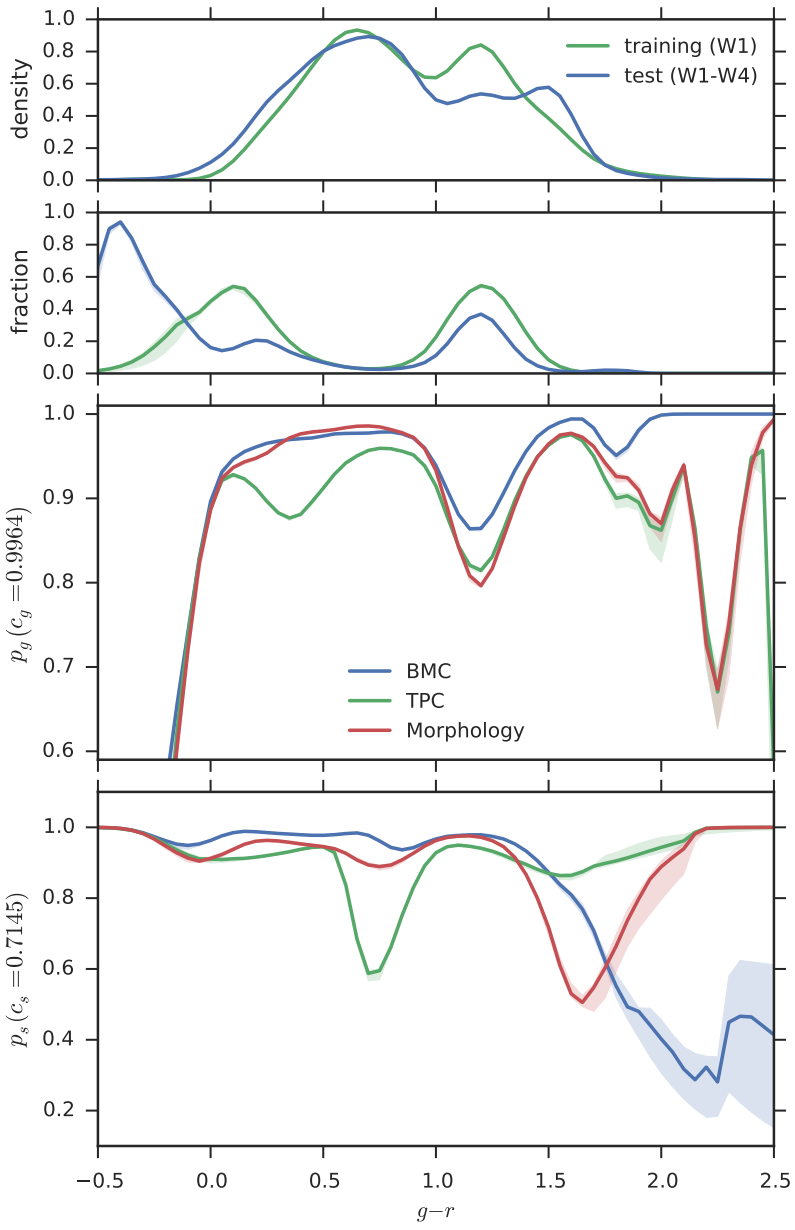


Figure 2.14: Similar to Figure 2.11 but as a function of $g - r$ color.

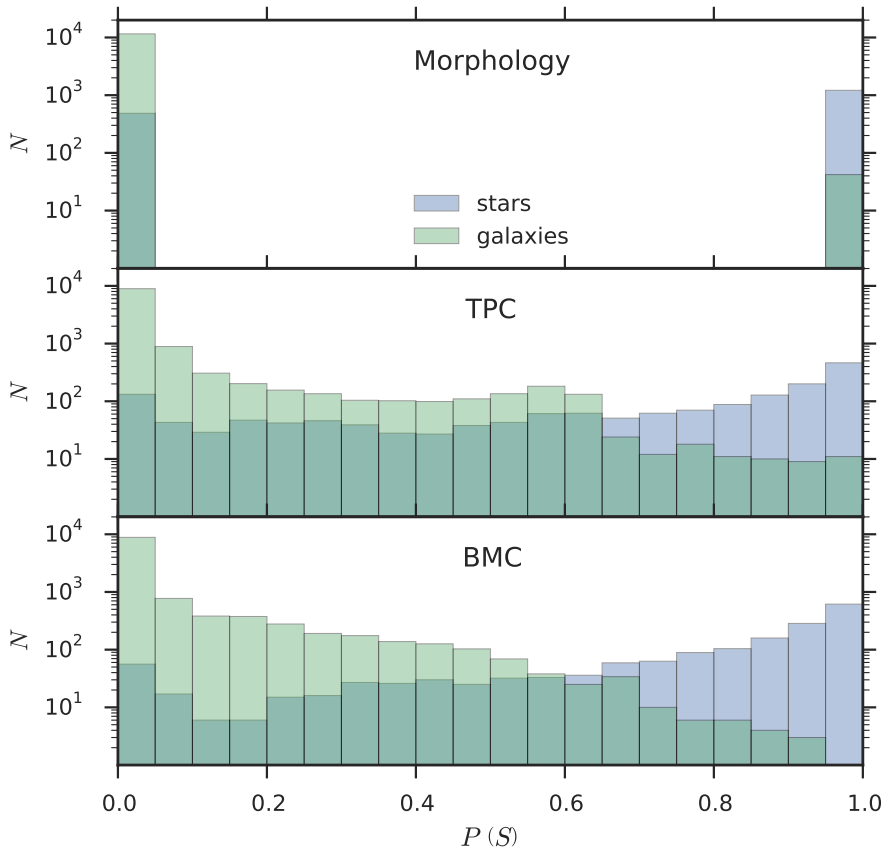


Figure 2.15: Similar to Figure 2.8 but for the reduced training data set.

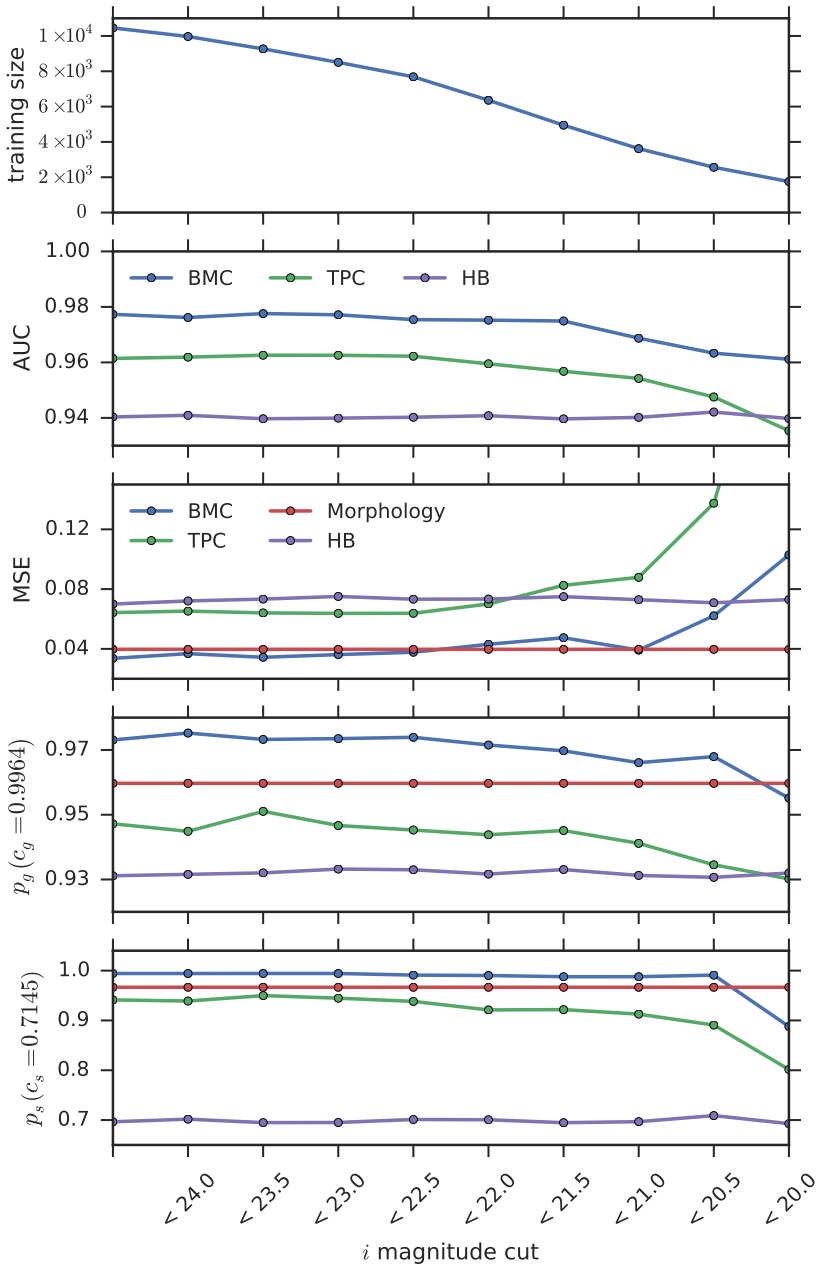


Figure 2.16: The classification performance metrics for BMC (blue), TPC (green), morphology (red), and HB (purple) as applied to the CFHTLenS data in the VVDS field with various magnitude cuts. The top panel shows the number of sources in the training set at corresponding magnitude cuts. We show only one of the four combination methods, BMC, which has the best overall performance.

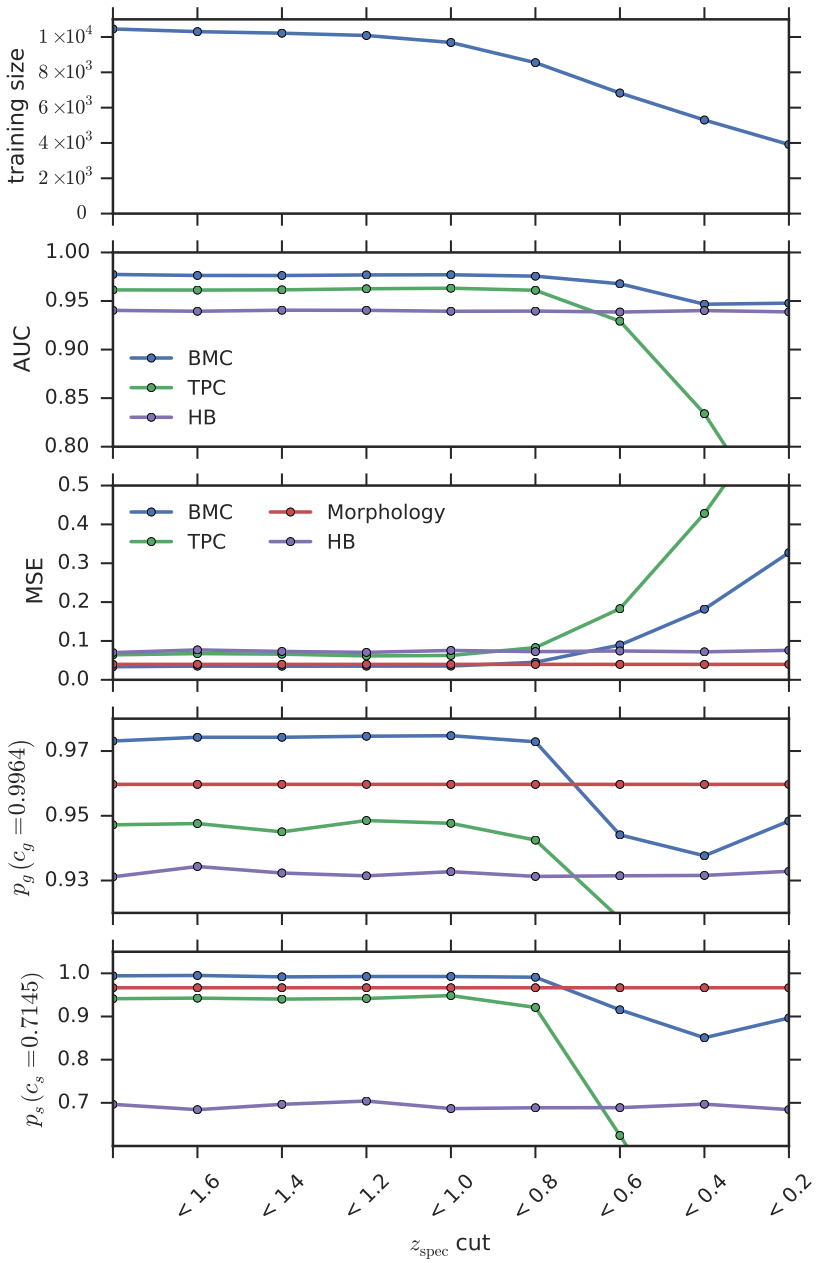


Figure 2.17: Similar to Figure 2.16 but using $z_{\rm spec}$ cuts.

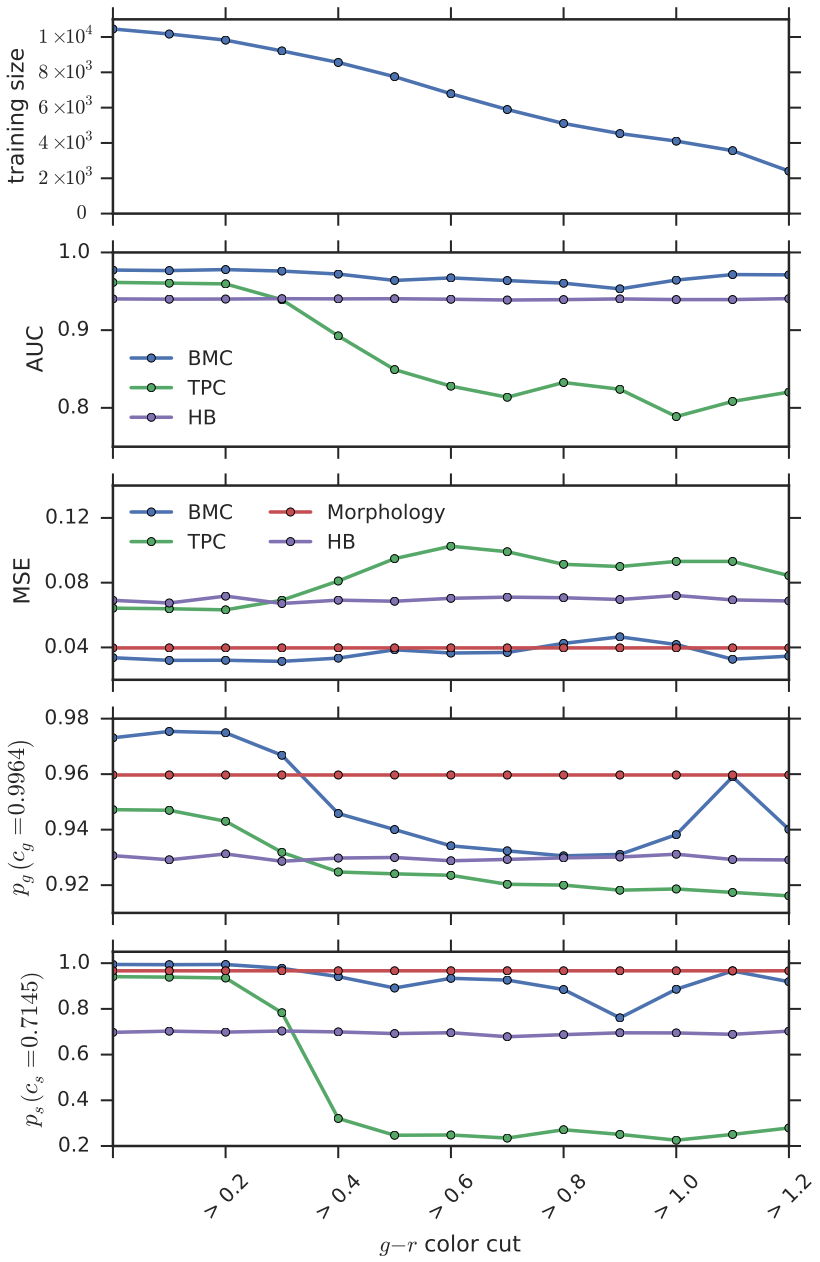


Figure 2.18: Similar to Figure 2.16 but using $g - r$ color cuts.

Metric	Meaning
AUC	Area under the Receiver Operating Curve
MSE	Mean squared error
c_g	Galaxy completeness
p_g	Galaxy purity
c_s	Star completeness
p_s	Star purity
$p_g(c_g = x)$	Galaxy purity at x galaxy completeness
$p_s(c_s = x)$	Star purity at x star completeness

Table 2.1: The definition of the classification performance metrics.

Classifier	AUC	MSE	$p_g\,(c_g = 0.9964)$	$p_s\,(c_s = 0.7145)$	$p_g\,(c_g = 0.9600)$	$p_s\,(c_s = 0.2500)$
TPC	**0.9870**	0.0208	0.9714	0.9838	0.9918	0.9977
SOMc	0.9683	0.0452	0.9125	0.8454	0.9788	0.9551
HB	0.9403	0.0705	0.9219	0.7017	0.9471	0.6963
Morphology	-	0.0397	0.9597	0.9666	-	-
WA	0.9806	0.0266	0.9755	0.9926	0.9872	0.9977
BoM	0.9870	0.0208	0.9714	0.9838	0.9918	0.9977
Stacking	0.9842	0.0194	0.9752	0.9902	0.9918	**1.0000**
BMC	0.9852	**0.0174**	**0.9800**	**0.9959**	**0.9924**	**1.0000**

Table 2.2: A summary of the classification performance metrics for the four individual methods and the four different classification combination methods as applied to the CFHTLenS data, with no cut applied to the training data set. The definition of the metrics is summarized in Table 3.2. The bold entries highlight the best performance values within each column. Note that some objects in the test set have bad or missing values (e.g., -99 or 99) in one or more attributes, which are included here (but are omitted, for example, in Figure 2.5 when the corrsponding attribute is not available.)

Classifier	AUC	MSE	$p_g\,(c_g = 0.9964)$	$p_s\,(c_s = 0.7145)$	$p_g\,(c_g = 0.9600)$	$p_s\,(c_s = 0.2500)$
TPC	0.9399	0.0511	0.9350	0.7060	0.9570	0.9747
SOMc	0.8861	0.0989	0.8843	0.4316	0.9165	0.6263
HB	0.9386	0.0760	0.9325	0.6911	0.9424	0.6918
Morphology	-	0.0397	0.9597	0.9666	-	-
WA	0.9600	0.0536	0.9208	0.8818	0.9757	0.9815
BoM	0.9587	0.1511	0.9658	0.9862	0.9790	0.9977
Stacking	0.9442	0.1847	0.9561	0.9309	0.9664	0.9983
BMC	**0.9738**	**0.0291**	**0.9696**	**0.9862**	**0.9856**	**1.0000**

Table 2.3: A summary of the classification performance metrics for the four individual methods and the four different classification combination methods when the training data set consists of only the sources that are in CFHTLS W1 field, has spectroscopic labels available from VVDS, and has $i < 22$. The definition of the metrics is summarized in Table 3.2. The bold entries highlight the best performance values within each column. Note that some objects in the test set have bad or missing values (e.g., -99 or 99) in one or more attributes, which are included here (but are omitted, for example, in Figure 2.12 when the corrsponding attribute is not available.)

2.8 References

C. P. Ahn et al. The Tenth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Apache Point Observatory Galactic Evolution Experiment. *ApJS*, 211:17, April 2014.

N. M. Ball, R. J. Brunner, A. D. Myers, and D. Tcheng. Robust Machine Learning Applied to Astronomical Data Sets. I. Star-Galaxy Classification of the Sloan Digital Sky Survey DR3 Using Decision Trees. *ApJ*, 650:497–509, October 2006.

N. Benítez. Bayesian Photometric Redshift Estimation. *ApJ*, 536:571–583, June 2000.

E. Bertin and S. Arnouts. Sextractor: Software for source extraction. *A&AS*, 117:393–404, June 1996.

R. C. Bohlin, L. Colina, and D. S. Finley. White Dwarf Standard Stars: G191-B2B, GD 71, GD 153, HZ 43. *AJ*, 110:1316, September 1995.

Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

M. Carrasco Kind and R. J. Brunner. TPZ: photometric redshift PDFs and ancillary information by using prediction trees and random forests. *MNRAS*, 432:1483–1501, June 2013.

M. Carrasco Kind and R. J. Brunner. Exhausting the information: novel bayesian combination of photometric redshift pdfs. *MNRAS*, 442:3380–3399, August 2014a.

M. Carrasco Kind and R. J. Brunner. SOMz: photometric redshift PDFs with self-organizing maps and random atlas. *MNRAS*, 438:3409–3421, March 2014b.

G. Chabrier, I. Baraffe, F. Allard, and P. Hauschildt. Evolutionary Models for Very Low-Mass Stars and Brown Dwarfs with Dusty Atmospheres. *ApJ*, 542:464–472, October 2000.

G. D. Coleman, C.-C. Wu, and D. W. Weedman. Colors and magnitudes predicted for high redshift galaxies. *ApJS*, 43:393–416, July 1980.

M. Davis et al. Science objectives and early results of the deep2 redshift survey. *Astronomical Telescopes and Instrumentation*, pages 161–172, 2003.

T. Erben et al. Cfhtlens: the canada–france–hawaii telescope lensing survey–imaging data and catalogue products. *MNRAS*, page stt928, 2013.

R. Fadely, D. W. Hogg, and B. Willman. Star-Galaxy Classification in Multi-band Optical Imaging. *ApJ*, 760:15, November 2012.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC Hammer. *PASP*, 125:306–312, March 2013.

B. Garilli et al. The vimos vlt deep survey-global properties of 20 000 galaxies in the iab ¡ 22.5 wide survey. *A&A*, 486(3):683–695, 2008.

B. Garilli et al. The vimos public extragalactic survey (vipers)-first data release of 57 204 spectroscopic measurements. *A&A*, 562:A23, 2014.

K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *ApJ*, 622:759–771, April 2005.

Stephen DJ Gwyn. The canada-france-hawaii telescope legacy survey: Stacked images and catalogs. *AJ*, 143(2):38, 2012.

M. Henrion, D. J. Mortlock, D. J. Hand, and A. Gandy. A Bayesian approach to star-galaxy classification. *MNRAS*, 412:2286–2302, April 2011.

C. Heymans et al. Cfhtlens: the canada–france–hawaii telescope lensing survey. *MNRAS*, 427(1):146–166, 2012.

H. Hildebrandt et al. Cfhtlens: improving the quality of photometric redshifts with precision photometry. *MNRAS*, 421(3):2355–2367, 2012.

N. Kaiser, G. Squires, and T. Broadhurst. A Method for Weak Lensing Observations. *ApJ*, 449:460, August 1995.

E. J. Kim, R. J. Brunner, and M. Carrasco Kind. A hybrid ensemble learning approach to star-galaxy classification. *MNRAS*, 453(1):507–521, 2015.

A. L. Kinney, D. Calzetti, R. C. Bohlin, K. McQuade, T. Storchi-Bergmann, and H. R. Schmitt. Template Ultraviolet to Near-Infrared Spectra of Star-forming Galaxies and Their Application to K-Corrections. *ApJ*, 467:38, August 1996.

Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Teuvo Kohonen. *Self-organizing maps*, volume 30 of *Springer*. Springer, 2001.

R. G. Kron. Photometry of a complete sample of faint galaxies. *ApJS*, 43:305–325, June 1980.

O. Le Fèvre et al. The vimos vlt deep survey-first epoch vvds-deep survey: 11 564 spectra with 17.5 ¡= iab ¡= 24, and the redshift distribution over 0 ¡ z ¡= 5. *A&A*, 439(3):845–862, 2005.

C. Messier. Catalogue des nébuleuses & des amas d'Étoiles. *Connoissance des Temps for 1784*, pages 227–267, 1781.

K. Monteith, J. L. Carroll, K. Seppi, and T. Martinez. Turning bayesian model averaging into bayesian model combination. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2657–2663. IEEE, 2011.

J. A. Newman et al. The deep2 galaxy redshift survey: design, observations, data reduction, and redshifts. *ApJS*, 208(1):5, 2013.

S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys, and W. A. Zumach. Automated star/galaxy discrimination with neural networks. *AJ*, 103:318–331, January 1992.

Paterno, M. Calculating efficiencies and their uncertainties. `http://home.fnal.gov/~paterno/images/effic.pdf`, May 2003.

A. J. Pickles. A Stellar Spectral Flux Library: 1150-25000 Å. *PASP*, 110:863–878, July 1998.

A. C. Robin et al. The Stellar Content of the COSMOS Field as Derived from Morphological and SED-based Star/Galaxy Separation. *ApJS*, 172:545–559, September 2007.

Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

AJ Ross et al. Ameliorating systematic uncertainties in the angular clustering of galaxies: a study using the sdss-iii. *MNRAS*, 417(2):1350–1373, 2011.

W. L. Sebok. Optimal classification of images into stars or galaxies - A Bayesian approach. *AJ*, 84:1526–1536, October 1979.

Ignacio Sevilla-Noarbe and Penélope Etayo-Sotos. Effect of training characteristics on object classification: an application using boosted decision trees. *Astronomy and Computing, in press (arXiv:1504.06776)*, 2015. doi: 10.1016/j.ascom.2015.03.010. URL `http://dx.doi.org/10.1016/j.ascom.2015.03.010`.

Bernard W Silverman. Density estimation for statistics and data analysis. *CRC press*, 26, 1986.

M. T. Soumagnac et al. Star/galaxy separation at faint magnitudes: Application to a simulated dark energy survey. *MNRAS*, 450:666–680, jun 2015.

A. A. Suchkov, R. J. Hanisch, and B. Margon. A Census of Object Types and Redshift Estimates in the SDSS Photometric Catalog from a Trained Decision Tree Classifier. *AJ*, 130:2439–2452, December 2005.

John A Swets, Robyn M Dawes, and John Monahan. Better decisions through. *Scientific American*, page 83, 2000.

Kai Ming Ting and Ian H Witten. Issues in stacked generalization. *J. Artif. Intell. Res.(JAIR)*, 10:271–289, 1999.

F. Valdes. Resolution classifier. In *Instrumentation in Astronomy IV*, volume 331 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 465–472, October 1982.

E. C. Vasconcellos, R. R. de Carvalho, R. R. Gal, F. L. LaBarbera, H. V. Capelato, H. Frago Campos Velho, M. Trevisan, and R. S. R. Ruiz. Decision Tree Classifiers for Star/Galaxy Separation. *AJ*, 141:189, June 2011.

N. Weir, U. M. Fayyad, and S. Djorgovski. Automated Star/Galaxy Classification for Digitized Poss-II. *AJ*, 109:2401, June 1995.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

H. K. C. Yee. A faint-galaxy photometry and image-analysis system. *PASP*, 103:396–411, April 1991.

Hujun Yin. The self-organizing maps: background, theories, extensions and applications. *Computational intelligence: a compendium*, pages 715–762, 2008.

Chapter 3

Star-galaxy Classification Using Deep Convolutional Neural Networks

3.1 Introduction

Currently ongoing and forthcoming large-scale photometric surveys, such as the Dark Energy Survey (DES) and the Large Synoptic Survey Telescope (LSST), aim to collect photometric data for hundreds of millions to billions of stars and galaxies. Due to the sheer volume of data, it is not possible for human experts to manually classify them, and the separation of photometric catalogs into stars and galaxies has to be automated. Furthermore, any classification approach must be probabilistic in nature. A fully probabilistic classifier enables a user to adopt probability cuts to obtain a pure sample for population studies, or to optimize the allocation of observing time by selecting objects for follow-up. Ideally, however, the probability estimates themselves would be retained for all sources and used in subsequent analyses to improve or enhance a particular measurement (Ross et al., 2011; Seo et al., 2012).

With machine learning, we can use algorithms to automatically create accurate source catalogs with well-calibrated posterior probabilities. Machine learning techniques have been a popular tool in many areas of astronomy (Ball et al., 2008; Banerji et al., 2010; Carrasco Kind and Brunner, 2013; Ivezić et al., 2014; Kamdar et al., 2016). Artificial neural

This chapter contains material from the following previously published article:

- E. J. Kim and R. J. Brunner. Star–galaxy classification using deep convolutional neural networks. *MNRAS*, 464(4):4463–4475, 2017

networks were first applied to the problem of star-galaxy classification in the work of Odewahn et al. (1992), and they have become a core part of the astronomical image processing software `SExtractor` (Bertin and Arnouts, 1996). Other successfully implemented examples of applying machine learning to the star-galaxy classification problem include decision trees (Weir et al., 1995; Suchkov et al., 2005; Ball et al., 2006; Vasconcellos et al., 2011; Sevilla-Noarbe and Etayo-Sotos, 2015), Support Vector Machines (Fadely et al., 2012), and classifier combination strategies (Kim, Brunner, and Carrasco Kind, 2015).

Almost all star-galaxy classifiers published in the literature use the reduced summary information available from astronomical catalogs. Constructing catalogs requires careful engineering and considerable domain expertise to transform the reduced, calibrated pixel values that comprise an image into suitable features, such as magnitudes or shape information of an object. In a branch of machine learning called *deep learning* (LeCun et al., 2015), features are not designed by human experts; they are learned directly from data by deep neural networks. Deep learning methods learn multiple levels of features by transforming the feature at one level into a more abstract feature at a higher level. For example, when an array of pixel values is used as input to a deep learning method, the features in the first layer might represent locations and orientations of edges. The second layer could assemble particular arrangements of edges into more complex shapes, and subsequent layers would detect objects as combinations of low-level features. These multiple layers of abstraction progressively amplify aspects of the input that are important for classification tasks. Deep learning has been applied successfully to galaxy morphological classification in Sloan Digital Sky Survey (SDSS; Dieleman et al., 2015a) and Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (CANDELS; Huertas-Company et al., 2015) and to photometric redshift estimation (Hoyle, 2015), but it has not yet been applied to the problem of source classification.

In this chapter, we present a star-galaxy classification framework that uses a convolutional neural network (ConvNet) model directly on the images from the SDSS and the Canada-

France-Hawaii Telescope Lensing Survey (CFHTLenS). We compare its performance with a standard machine learning technique that uses the reduced summary information from catalogs, and we demonstrate that our ConvNet model is able to produce accurate and well-calibrated probabilistic classifications with very little feature engineering by hand. In Section 4.2, we describe the data sets used in this work and the pre-processing steps for preparing the image data sets. We provide a brief overview of deep learning and ConvNets in Section 3.3, and discuss our strategy for preventing overfitting in Section 3.4. In Section 3.5, we describe a state-of-the-art tree-based machine learning algorithm, to which the performance of our ConvNet model is compared. We present the main results of our ConvNet model in Section 3.6, and we outline our conclusions in Section 4.5.

3.2 Data

To demonstrate the performance of our ConvNet model, we use photometric and spectroscopic data sets with different characteristics and compositions. In this section, we briefly describe these data sets and the image pre-processing steps for retrieving cutout images.

3.2.1 Sloan Digital Sky Survey

The Sloan Digital Sky Survey (SDSS; York et al., 2000) phases I–III obtained photometric data in five bands, u, g, r, i, and z, covering 14,555 square degrees, more than one-third of the entire sky. The resulting catalog contains photometry of over 300 million stars and galaxies with a limiting magnitude of $r \approx 22$, making the SDSS one of the largest sky surveys ever undertaken. The SDSS also conducted an expansive spectroscopic follow-up of more than three million stars and galaxies (Eisenstein et al., 2011). In this work, we use a subset of the photometric and spectroscopic data contained within the Data Release 12 (DR12; Alam et al., 2015), which is publicly available through the online CasJobs server (Li and Thakar, 2008).

Using the CasJobs server, we randomly select a total of 65,000 sources, which are either stars or galaxies. In this work, we exclude objects that clearly are neither stars nor galaxies. Most of the excluded objects are QSOs or quasars. Quasars appear as point sources, rather than resolved sources similar to galaxies, and many of them have one or more saturated pixels in the images. However, unlike any known stars, their spectra show strong and broad emission lines. Quasars are also different from galaxies because of their intrinsic variability on a wide range of time scales, which may be due to variation in the accretion rate or instabilities of the accretion disk around the black hole (Popović et al., 2012). Thus, many studies exclude quasars in the binary star-galaxy classification scheme (e.g., Vasconcellos et al., 2011; Fadely et al., 2012). Expanding the historical star-galaxy classification problem to include additional classes, e.g., *nsng* (neither star nor galaxy), may have advantages (Ball et al., 2006).

We also exclude some bad photometric observations as follows. We consider only objects with no warning flags in the spectroscopic measurement (`zWarning = 0`); the half-light radius in the r band is less than 30 arc seconds as measured by the exponential and de Vaucouleurs light profiles; the error on the spectroscopic redshift measurement is less than 0.1; and the spectroscopic redshift is less than 2.

To create training images, we obtain the image FITS files for SDSS fields containing these objects in five photometric bands: u, g, r, i, and z. We use the astrometry information in the FITS headers in the `Montage` software to align each image to the reference (r-band) image. We then use `SExtractor` to find the pixel positions of the 65,000 objects we have selected, and to center each object in the cutout image. Magnitudes in the SDSS photometric catalog are expressed as inverse hyperbolic sine magnitudes (also known as luptitudes; Lupton et al., 1999), and we follow the SDSS convention and convert all flux values to luptitudes. Finally, in order to account for the effect of Galactic dust, extinction corrections in magnitudes are applied following Schlegel et al. (1998). In the end, we have cutout images of size 48×48 pixels with luptitude values in each pixel. We note that we have experimented with increasing

the pixel dimensions to 60×60 and 72×72 pixels, but do not find noticeable improvement in the performance of our model.

In the end, we have 17,344 stars and 47,656 galaxies available for the training and testing processes. The apparent magnitudes range from $10.7 < r < 23.1$, and the galaxies in this sample have a mean redshift of $z \sim 0.36$. We randomly split the objects into training, held-out validation, and blind test sets of size 40,000, 10,000, and 15,000, respectively. We note that cross-validation is often avoided in deep learning in favor of hold-out validation, since cross-validation is computationally expensive. We also note that we perform a blind test, and the test set is not used in any way to train or calibrate the algorithms. The first two panels of Figure 3.8 show the number of objects and the fraction of stars in the test set as functions of r-band magnitude. Similarly, Figure 3.10 shows the number of objects and the fraction of stars in the test set as functions of $g - r$ color. The normalized kernel density estimate distributions for the training and validation sets are almost identical to those of the test set, and they are nearly indistinguishable when overlapped. We do not show the distributions for the training and validation sets in Figures 3.8 and 3.10 to avoid cluttering the plots.

3.2.2 Canada-France-Hawaii Telescope Lensing Survey

We also use photometric data from the Canada-France-Hawaii Telescope Lensing Survey (CFHTLenS; Heymans et al., 2012; Erben et al., 2013; Hildebrandt et al., 2012). This catalog consists of more than twenty five million objects with a limiting magnitude of $i_{\rm AB} \approx 25.5$. It covers a total of 154 square degrees in the four fields (named W1, W2, W3, and W4) of the CFHT Legacy Survey (CFHTLS; Gwyn, 2012) observed in the five photometric bands: u, g, r, i, and z.

We have cross-matched reliable spectroscopic galaxies from the Deep Extragalactic Evolutionary Probe Phase 2 (DEEP2; Davis et al., 2003; Newman et al., 2013), the Sloan Digital Sky Survey Data Release 10 (Alam et al., 2015, SDSS-DR10), the VIsible imaging

Multi-Object Spectrograph (VIMOS) Very Large Telescope (VLT) Deep Survey (VVDS; Le Fèvre et al., 2005; Garilli et al., 2008), and the VIMOS Public Extragalactic Redshift Survey (VIPERS; Garilli et al., 2014). We have selected only sources with very secure redshifts and no bad flags (quality flags -1, 3, and 4 for DEEP2; quality flag 0 for SDSS; quality flags 3, 4, 23, and 24 for VIPERS and VVDS).

We obtain FITS images for each 1 square degree CFHTLenS pointing that contains objects with spectroscopic labels. We create cutout images of size 96×96 pixels by using a similar method to that described in Section 3.2.1. Finally, images are downscaled to 48×48 pixels to reduce the computational cost.

In the end, we have 8,545 stars and 57,843 galaxies available for the training and testing processes. The apparent magnitudes range from $13.9 < r < 25.6$, and the galaxies in this sample have a mean redshift of $z \sim 0.59$. We randomly split the objects into training, held-out validation, and blind test sets of size 40,000, 10,000, and 13,278, respectively. Figures 3.2 and 3.4 show the distribution of objects in the test set as functions of i-band magnitude and $g - r$ color. We do not show the distributions for the training and validation sets, since the normalized kernel density estimate distributions for the training and validation sets are almost identical to those of the test set.

3.3 Deep Learning

Neural networks have many hyperparameters, including those that specify the network itself (e.g., the size and non-linearity of each layer) and those that specify how the network is trained (e.g., the mini-batch size or the learning rate). Furthermore, the architecture of a neural network can have a significant impact on its performance. In this section, we provide a brief description of key hyperparameters in our ConvNet model, and also present the network architecture.

3.3.1 Neural Networks

An artificial neuron in most artificial neural networks is represented as a mathematical function that models a biological neural structure (Aggarwal, 2014). A schematic representation is shown in Figure 3.1a. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a vector of inputs to a given neuron, $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ be a vector of weights, and b be the bias. Then, the output of the neuron is (Rosenblatt, 1961)

$$y = \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right), \tag{3.1}$$

where σ is the activation function (or *non-linearity*). The most popular non-linearity at present is the rectified linear unit (ReLU; Nair and Hinton, 2010), $\sigma(x) = \max(0, x)$. ReLUs generally allow much faster training of deep neural networks with many layers. However, ReLU units can sometimes result in dead neurons whose output is always zero. To mitigate this problem, we use leaky ReLUs (Maas et al., 2013) that have a small, non-zero slope in the negative region,

$$\sigma(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0.01x & \text{if } x < 0. \end{cases} \tag{3.2}$$

Many deep learning models use feedforward neural network architectures with multiple layers, where each neuron in one layer is connected to the neurons of the subsequent layer (LeCun et al., 2015). A schematic representation is shown in Figure 3.1b. All layers except the input and output layers are conveniently called hidden layers.

We find a set of weights and biases such that, given N samples, the output from the network $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ approximates the desired output $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$ as closely as possible for all input $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. We can formulate this as the minimization of a loss function $L(\mathbf{y}, \hat{\mathbf{y}})$ over the training data. In this work, we use *cross-entropy* (also called log loss; Murphy, 2012) as the loss function. For binary classification, the cross-entropy per sample is given by

$$L(y_j, \hat{y}_j) = -\hat{y}_j \log_2 y_j - (1 - \hat{y}_j) \log_2 (1 - y_j), \tag{3.3}$$

where $\hat{y}_j$ is the actual truth value (e.g., 0 or 1) of the j-th data, and y_j is the probability prediction made by the model. We compute the loss function by taking the average of all cross-entropies in the sample. Thus, the loss function becomes

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{j=1}^{N} \hat{y}_j \log_2 y_j + (1 - \hat{y}_j) \log_2 (1 - \hat{y}_j). \tag{3.4}$$

To find the weights $\mathbf{w}$ and biases $\mathbf{b}$ which minimize the loss, we use a technique called *gradient descent*, where we use the following rules to update the parameters in each layer l:

$$\begin{aligned} \mathbf{w}_l \to \mathbf{w}_l' &= \mathbf{w}_l - \eta \frac{\partial L}{\partial \mathbf{w}_l} \\ \mathbf{b}_l \to \mathbf{b}_l' &= \mathbf{b}_l - \eta \frac{\partial L}{\partial \mathbf{b}_l}, \end{aligned} \tag{3.5}$$

where η is a small, positive number known as the *learning rate*. The gradients can be computed using the backpropagation procedure (Rumelhart et al., 1988). A common approach to speed up training is to split the training data into mini-batches (LeCun et al., 1998b). In mini-batch gradient descent, instead of computing the gradients in Equation 3.5 for the entire training data, we only compute the gradient of randomly chosen training examples at each step. As training examples are usually correlated, the gradient computed from each mini-batch is a good approximation of the overall gradient (Bottou, 1998). As a result, mini-batch gradient descent results in much faster convergence. However, there is a trade-off: the lower the batch size is, the lower the convergence rate will be; the higher the batch size is, the longer it will take to compute the gradient at each step (Bousquet and Bottou, 2008). Thus, a moderate batch size, combined with a decaying learning rate, is generally used in practice. We use a batch size of 128 in this work.

We define an *epoch* as a single, complete pass through the training data, and full training usually requires many epochs. At the end of each epoch, we evaluate the loss function on the validation set, and the model that minimizes the validation loss is chosen as the best

model.

3.3.2 Convolutional Neural Networks

The convolutional neural network (ConvNet; Fukushima, 1980; LeCun et al., 1998a) is a type of deep, feedforward neural network that has recently become a popular approach in the computer vision community. In a typical ConvNet, the first few stages are composed of two types of layers: convolutional layers and pooling layers.

The input to a convolutional layer is an image, and the output channels of each layer are called *feature maps*. To produce output feature maps, we convolve each feature map with a set of weights called *filters*, and apply a non-linearity such as ReLU to the weighted sum of these convolutions. Different feature maps use different sets of filters, but all neurons in a feature map share the same set of filters. Mathematically, we replace the dot product in Equation 3.1 with a sum of convolutions. Thus, the k-th feature map is given by

$$y^k = \sigma\left(\sum_m \mathbf{w}_m^k * \mathbf{x}_m + b^k\right), \tag{3.6}$$

where we sum over the set of input feature maps, $*$ is the convolution operator, and $\mathbf{w}_m^k$ represent the filters.

Typically, a pooling layer computes the maximum of a local 2×2 patch in a feature map (Krizhevsky et al., 2012). Since the pooling layer aggregates the activations of neighboring units from the previous layer, it reduces the dimensionality of the feature maps and makes the model invariant to small shifts and distortions (Boureau et al., 2010) . Two or more layers of convolution and pooling are stacked, followed by more convolutional and fully-connected layers.

3.3.3 Neural Network Architecture

We present the overall architecture of our ConvNet model in Table 3.1. The network consists of eleven trainable layers. The first convolutional layer filters the $5 \times 44 \times 44$ input image (i.e., 44×44 images in five bands *ugriz*) with 32 square filters of size $5 \times 5 \times 5$. We have also experimented with using only three bands *gri* (for three channels of RGB) and four bands *ugri* and *griz* (corresponding to RGBA), and using only colors, e.g., $u - g$, $g - r$, $r - i$, and/or $i - z$, but we find that using magnitudes in all five bands *ugriz* yields the best performance.

The leaky ReLU non-linearity is applied to the output of the first convolutional layer (and all subsequent layers), and the second convolutional layer filters it with 32 filters of $32 \times 3 \times 3$. In the second convolutional layer (and all subsequent convolutional layers), we pad the input with zeros spatially on the border (i.e., the zero-padding is 1 pixel for 3×3 convolutional layers) such that the spatial resolution is preserved after convolution. Max-pooling with filters of size 2×2 follows the second convolutional layer. A stack of six additional convolutional layers, all with filters of size 3×3, is followed by three fully-connected layers. The first two fully-connected layers have 2048 channels each, and the third performs binary classification.

The output of the final fully-connected layer is fed to a *softmax* function. The softmax function is given by

$$P(G \mid \mathbf{x}) = \frac{e^{\mathbf{x} \cdot \mathbf{w}_G}}{\sum_i e^{\mathbf{x} \cdot \mathbf{w}_i}}, \tag{3.7}$$

where we sum over the different possible values of the class label (i.e., star or galaxy), and interpret its output as the posterior probability that an object is a galaxy (or a star). We note that we could also try to solve a regression problem, e.g., by normalizing the output values that the network produces for each class. However, we find that solving a regression problem instead of using the softmax function results in significantly worse performance.

We have performed a manual search to explore more than 200 combinations of different

architectures and hyperparameters to find an architecture that minimizes the loss function (Equation 3.4) on the validation set of the SDSS data. The architecture described in this section provides the best performance on the SDSS validation set. To test how this model performs across different, related data, we use the same architecture on the CFHTLenS data set.

The architecture of Krizhevsky et al. (2012) uses relatively large receptive fields (11×11) in the first convolutional layers. Zeiler and Fergus (2014) and Dieleman et al. (2015a) also use large receptive fields of 7×7 and 6×6 in the first convolution layer, respectively. However, we find that using a receptive field larger than 5×5 in the first convolutional layer leads to worse performance. This result is in agreement with the network of Simonyan and Zisserman (2014), which has become known as "VGGNet". VGGNet features an extremely homogeneous architecture that only performs 3×3 convolutions. Using a large receptive field instead of a stack of multiple 3×3 convolutions leads to a shallower network, and it is often preferable to increase the depth by using smaller receptive fields. However, we find that replacing the first layer with a stack of two 3×3 convolutional layers increases the validation error, and thus use a 5×5 convolution in the first layer.

In the remaining layers, we still follow VGGNet and add many 3×3 convolutions (with zero-padding of size 1 pixel). Note that with the padding of 1 pixel for 3×3 convolutional layers, the spatial resolution will be preserved after convolution. Such preservation of spatial resolution allows us to build relatively deep networks, as shown in Table 3.1. The main contribution of VGGNet is in showing that the depth plays an important role in good performance. In our case, we start with four convolutional layers and progressively add more layers, while monitoring the validation loss; we stop at eight convolutional layers after we find that adding more layers leads to worse performance. We conjecture that a greater depth and hence larger number of parameters lead to overfitting in our case.

The choice of momentum, learning rate, and initial weights is crucial for achieving high predictive performance and speeding up the learning process (Sutskever et al., 2013). To

train our models, we use mini-batch gradient descent with a batch size of 128 and Nesterov momentum (Bengio et al., 2013) of 0.9. We initialize the learning rate η at 0.003 for all layers and decrease it linearly with the number of epochs from 0.003 to 0.0001 over 750 epochs. We also initialize the weights in each layer with random orthogonal initial conditions (Saxe et al., 2013). We use slightly positive values ($b = 0.01$ or 0.1) for all biases. We find initializing biases to a small constant value helps eliminate dead neurons by ensuring that all ReLU neurons fire in the beginning.

To implement our model, we use Python and the Lasagne library (Dieleman et al., 2015b), which is built on top of Theano (Theano Development Team, 2016). The Theano library simplifies the use of GPU for computation, and using the GPU allows about an order of magnitude faster training than using just the CPU. We note that training our network takes about forty hours on an NVIDIA Tesla K40 GPU. In the interest of scientific reproducibility, we make all our code available at `https://github.com/EdwardJKim/dl4astro`.

3.4 Reducing Overfitting

Our convolution neural network has 11×10^6 learnable parameters, while there are only 4×10^4 images in the training set. As a result, the model is very likely to *overfit* without regularization. In this section, we describe the techniques we used to minimize overfitting.

3.4.1 Data Augmentation

One common method to combat overfitting is to artificially increase the number of training data by using label-preserving transformations (Krizhevsky et al., 2012; Dieleman et al., 2015a, 2016). Each image is transformed as follows:

- Rotation: Rotating an image does not change whether the object is a star or a galaxy. We exploit this rotational symmetry and randomly rotate each image by a multiple of 90°.

- Reflection: We flip each image horizontally with a probability of 0.5 to exploit mirror symmetry.

- Translation: We also have translational symmetry in the images. Given an image of size 48×48 pixels, we extract a random contiguous crop of size 44×44. Each cropping is equivalent to randomly shifting a 44×44 image by up to 4 pixels vertically and/or horizontally.

- Gaussian noise: We introduce random Gaussian noise to each pixel values by using a similar method to Krizhevsky et al. (2012).

In addition to artificially increasing the size of the data set, these data augmentation schemes make the resulting model more invariant to rotation, reflection, translation, and small noise in the pixel values. We also note that the data augmentation steps add almost no computational cost, as they are performed on the CPU while the GPU is training the ConvNets on images.

3.4.2 Dropout

We use a regularization technique called dropout (Hinton et al., 2012) in the fully-connected layers. Dropout consists of randomly setting to zero the output of each hidden neuron of the previous layer with probability 0.5. The weights of the remaining neurons are multiplied by 0.5 to preserve the scale of input values to the next layer. Since a neuron can be removed at any time, it cannot rely on the presence of other neurons in the same layer and is forced to learn more robust features.

3.4.3 Model Combination

To make final classifications, we use our ConvNet model to make 64 sets of predictions for 64 transformations of the input images: 4 rotations, 4 horizontal translations, and 4 vertical

translations (with random horizontal reflections). Although we use an identical network architecture for all transformations, we consider each set of predictions as separate results from different models. Finally, we use a model combination technique known as Bayesian Model Combination (BMC; Monteith et al., 2011), which uses Bayesian principles to generate an ensemble combination of different models. Although the data augmentation step in Section 3.4.1 should make our ConvNet model invariant to these types of transformations, we find that applying BMC still results in a significant increase in performance. For a thorough discussion of BMC, we refer the reader to Monteith et al. (2011) (See also Carrasco Kind and Brunner (2014) for application of BMC to photometric redshift estimation, and Kim et al. (2015) for combining star-galaxy classifiers).

3.5 Trees for Probabilistic Classifications

To compare the performance of ConvNets with machine learning algorithms that use standard photometric features, we use a machine learning framework called Trees for Probabilistic Classifications (TPC). TPC is a parallel, supervised machine learning algorithm that uses prediction trees and random forest techniques (Breiman et al., 1984; Breiman, 2001) to produce a star-galaxy classification. For a more detailed description of TPC, see Section 3.5. While other random forest implementations exist, we have chosen TPC, because it is tailored specifically for astronomical use (Carrasco Kind and Brunner, 2013); it has been tested for astronomical use cases, including photometric redshift estimation (Sánchez et al., 2014) and star-galaxy classification (Kim et al., 2015); and it uses parallelism to handle large data sets on distributed memory systems.

We train two TPC models on the SDSS data set by using different sets of attributes. The first model, which we denote $\mathrm{TPC}_{\mathrm{phot}}$, is trained with only nine photometric attributes: the extinction-corrected model magnitudes in five bands (u, g, r, i, z) and their corresponding colors ($u-g$, $g-r$, $r-i$, $i-z$). The second model, which we denote $\mathrm{TPC}_{\mathrm{morph}}$, is

trained with the concentration parameter in each band in addition to the magnitudes and colors, for a total of fourteen dimensions. The concentration is defined as the difference between the PSF magnitude (psfMag) and the composite model magnitude (cModelMag), i.e., concentration $\equiv$ psfMag $-$ cModelMag. The SDSS pipeline uses a parametric method based on the concentration, an object is classified as a galaxy if concentration > 0.145. We find that the concentration is an excellent morphological feature for star-galaxy separation, and including more morphological features does not show noticeable improvement in performance. The concentration is a good example of carefully handcrafted feature extraction; we show in Section 3.6 that ConvNets do not require such feature engineering.

We also train two models on the CFHTLenS data set. TPC_{phot} is trained with the five magnitudes and their corresponding colors: u, g, r, i, z, $u-g$, $g-r$, $r-i$, and $i-z$. Since the CFHTLenS catalog does not provide the concentration parameter, $\text{TPC}_{\text{morph}}$ uses `SExtractor`'s `FLUX_RADIUS` (the half-light radius), `A_WORLD` (the semi-major axis), and `B_WORLD` (the semi-minor axis) for morphological features, in addition to the five magnitudes and their corresponding colors, for a total of twelve dimensions.

3.6 Results and Discussion

In this section, we first describe the performance metrics that were used for evaluating the models. We then present the classification performance of our ConvNet model on the CFHTLenS and SDSS data sets, and compare it with the performance of TPC.

3.6.1 Classification Metrics

Probabilistic classifiers, rather than only assigning discrete labels to each source, produce a continuous probability distribution of whether each source is a star or a galaxy. To evaluate the performance of probabilistic classifiers, many studies (e.g., Henrion et al., 2011; Fadely et al., 2012) convert probability estimates into class labels by choosing a probability

threshold, e.g., a source is classified as a star if $P_{\text{class}} < 0.5$, and a galaxy if $P_{\text{class}} > 0.5$. However, using a fixed threshold ignores the model's operating conditions, such as science requirements, misclassification costs, and stellar distribution. Furthermore, the probability threshold of 0.5 is not necessarily optimal for an unbalanced data set, where galaxies outnumber stars.

Following Kim et al. (2015), we use performance metrics that are well-suited for probabilistic classifiers: Area Under the Curve (AUC) for the Receiver Operating Characteristic (ROC) curve, completeness and purity, and the Mean Squared Error (MSE). A good probabilistic classifier should also provide well-calibrated posterior probabilities. Thus, to evaluate calibration performance, we also use the calibration error and the absolute error in the estimation of number of galaxies.

Receiver Operating Characteristic Curve

A Receiver Operating Characteristic (ROC) curve is the most commonly used method for evaluating the overall performance of a binary classifier (Swets, Dawes, and Monahan, 2000). In an ROC curve, we plot the true positive rate as a function of the false positive rate by varying the classification threshold. The Area Under the Curve (AUC) quantifies the overall performance in a single number.

Completeness and Purity

Let N_g be the number of true galaxies classified as galaxies, and M_g the number of true galaxies classified as stars. Then the galaxy *completeness* c_g (also called recall or sensitivity) is given by

$$c_g = \frac{N_g}{N_g + M_g}. \tag{3.8}$$

Let M_s be the number of true stars classified as galaxies. Then the galaxy *purity* p_g (also called precision or positive predictive value) is given by

$$p_g = \frac{N_g}{N_g + M_s}. \tag{3.9}$$

We define the star completeness and purity in a similar way. As discussed in our previous work (Kim et al., 2015), we adopt weak lensing science requirements of the DES (Soumagnac et al., 2015), and compute p_g at $c_g = 0.960$ and c_s at $p_s = 0.970$.

Mean Squared Error

We also use the mean squared error (MSE; also known as the Brier score (Brier, 1950)) as a performance metric. We define MSE as

$$\mathrm{MSE} = \frac{1}{N} \sum_{j=1}^{N} \left(y_j - \hat{y}_j\right)^2, \tag{3.10}$$

The MSE can be considered as both a score function that quantifies how well a set of probabilistic predictions is calibrated, or a loss function.

Calibration Error

A fully probabilistic classifier predicts not only the class label, but also its confidence level on the prediction. In a well-calibrated classifier, the posterior class probability estimates should coincide with the proportion of objects that truly belong to a certain class. Probability *calibration curves* (or reliability curves; DeGroot and Fienberg, 1983) are often used to display this relationship, where we bin the probability estimates and plot the fraction of positive examples versus the predicted probability in each bin (see Figures 3.5 and 3.11).

The problem with a binning approach is either too few or too many bins can distort the evaluation of calibration performance. Thus, we adopt a calibration measure based on

overlapping binning (Caruana and Niculescu-Mizil, 2004). We order the predicted values $P_{\rm class}$ and put the first 1,000 elements in the first bin. We calculate the true probability $P_{\rm gal}$ by counting the true galaxies in this bin. The calibration error for this bin is $\mid P_{\rm gal} - P_{\rm class} \mid$. We then repeat this for the second bin (2 to 1,001), the third bin (3 to 1,002), and so on, and average the binned calibration errors. Thus, the overall calibration error is given by

$$\mathrm{CAL} = \frac{1}{N-s} \sum_{b=1}^{N-s} \sum_{j=b}^{b+s-1} \left| P_{\mathrm{class},j} - \frac{\sum_{j=b}^{b+s-1} P_{\mathrm{gal},j}}{s} \right|, \tag{3.11}$$

where $s = 1000$ is the bin length, which is chosen approximately equal to the number of objects in the testing set divided by the number of bins used in the calibration curve, i.e., $s \approx N/10$.

Number of galaxies

Ideally, the probabilistic output of a classifier would be used in subsequent scientific analyses. For example, one can weight each object by the probability that it is a galaxy when measuring auto-correlation functions of luminous galaxies (Ross et al., 2011). In other words, given a well-calibrated classifier, instead of counting each galaxy equally, a galaxy could be counted as $P_{\rm class}$, the probability estimate. This should in principle remove the contamination effect of stars. For a perfect classifier, we can count the total number of galaxies in the sample by summing the values of $P_{\rm class}$. Thus, we measure the reliability of classifier output by the absolute error in the estimation of number of galaxies,

$$\frac{\mid \Delta N_g \mid}{N_g} = \frac{\left| N_g - \sum_{j=1}^{N} P_{\mathrm{class},j} \right|}{N_g}. \tag{3.12}$$

3.6.2 CFHTLenS

As described in Section 3.3.2, we train our ConvNet model by monitoring its performance on the validation set. Once we have finished training the model, we evaluate its performance

on the blind test set. We also use the same training and validation sets to train and tune the hyperparameters of $\mathrm{TPC}_{\mathrm{morph}}$ and $\mathrm{TPC}_{\mathrm{phot}}$, and perform classifications on the same test set to compare their performance with that of ConvNet.

We present in Table 3.3 a summary of the results obtained by applying ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$ on the test set of the CFHTLenS data. The bold entries highlight the best technique for any particular metric. ConvNet outperforms $\mathrm{TPC}_{\mathrm{morph}}$ in five metrics (AUC, p_g, CAL, $|\Delta N_g|/N_g$, and log loss), while $\mathrm{TPC}_{\mathrm{morph}}$ performs better in two metrics (MSE and c_g). It is not surprising that $\mathrm{TPC}_{\mathrm{phot}}$, which is trained on only magnitudes and colors, performs significantly worse than both ConvNet and $\mathrm{TPC}_{\mathrm{phot}}$. Thus, magnitudes and colors alone are not sufficient to separate stars from galaxies, and morphology is critical in separating stars from galaxies. ConvNet is able to learn the morphological features automatically from the images, and the performance of ConvNet is therefore comparable to that of $\mathrm{TPC}_{\mathrm{morph}}$, which is trained on both morphological and photometric attributes.

In Figure 3.2, we compare the galaxy purity and star completeness values for ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$, as a function of i-band magnitude for the differential counts. We use kernel density estimation (KDE; Silverman, 1986) with a Gaussian kernel. As the first panel shows, KDE is able to smooth the fluctuations in the distribution without binning. While ConvNet shows a slightly better performance than $\mathrm{TPC}_{\mathrm{morph}}$ in galaxy purity, ConvNet performs slightly worse than $\mathrm{TPC}_{\mathrm{morph}}$ in star completeness. Again, $\mathrm{TPC}_{\mathrm{phot}}$ performs significantly worse than both ConvNet and $\mathrm{TPC}_{\mathrm{morph}}$, and this suggests that ConvNets are able to learn the shape information automatically from the images. We note that, at these operating conditions ($c_g = 0.96$ or $p_s = 0.97$), both ConvNet and $\mathrm{TPC}_{\mathrm{morph}}$ outperform the star-galaxy classification provided by the CFHTLenS pipeline (Hildebrandt et al., 2012) over all magnitudes.

In Figure 3.3, we show the overall galaxy purity and star completeness values as a function of i-band magnitude for the integrated counts. ConvNet is able to maintain a galaxy purity of 0.9972 up to $i \sim 24.5$, while the galaxy purity of $\mathrm{TPC}_{\mathrm{morph}}$ drops to 0.9963. However,

$\text{TPC}_{\text{morph}}$ performs better than ConvNet in terms of star completeness, maintaining a purity of 0.9252 up to $i \sim 24.5$, while ConvNet drops to 0.8966.

We also show the galaxy purity and star completeness values as functions of $g - r$ color in Figure 3.4. $\text{TPC}_{\text{morph}}$ provides slightly better completeness and purity than ConvNet between $0.8lesssimg - rlesssim1.6$ while ConvNet outperforms $\text{TPC}_{\text{morph}}$ in the remaining regions.

Figure 3.5 shows the calibration curves that compare P_{gal}, the fraction of objects that are galaxies (as determined from their spectra), to P_{class}, the probabilistic outputs produced by ConvNet and $\text{TPC}_{\text{morph}}$. The calibration curve for our ConvNet model is nearly diagonal, which implies that ConvNet is well-calibrated and we can treat its probabilistic output as the probability that an object is a galaxy. In contrast, the calibration curve for the probabilistic output of $\text{TPC}_{\text{morph}}$ is apparently not as well-calibrated as ConvNet. These calibration curves visually confirm the results in Table 3.3 that the calibration error of ConvNet is about 20% lower than that of $\text{TPC}_{\text{morph}}$. While probabilistic predictions can be further calibrated by using, e.g., isotonic calibration (Zadrozny and Elkan, 2001), we do not consider additional probability calibration in this work.

It is informative to visualize how an input image activates the neurons in the convolutional layers. Figures 3.6 and 3.7 show the activations of the network when images of a galaxy and a star are fed into the network. The size of feature maps decreases with depth, and layers near the input layer have fewer filters while the later layers have more. The low-level features, e.g., edges or blobs, of the input images are still recognizable in the first convolutional layer. Subsequent layers use these low-level features to detect higher-level features, and the final layer is a classifier that uses these high-level features. Thus, by performing hierarchical abstraction from low-level to high-level features, ConvNets are able to utilize shape information in the classification process.

3.6.3 SDSS

We have also trained and tested our ConvNet model on the SDSS data set, and we present in Table 3.4 the same six metrics for ConvNet, $\text{TPC}_{\text{morph}}$, and TPC_{phot}. The bold entries highlight the best technique for any particular metrics. In contrast with the CFHTLenS data set in Section 3.6.2, it is apparent that $\text{TPC}_{\text{morph}}$ outperforms ConvNet in all metrics except CAL and cross-entropy. Both ConvNet and $\text{TPC}_{\text{morph}}$ still outperform TPC_{phot} in all six metrics by a significant amount, as magnitudes and colors alone are not sufficient to separate stars from galaxies. Although ConvNet performs worse than $\text{TPC}_{\text{morph}}$ on the SDSS data, its performance is much closer to $\text{TPC}_{\text{morph}}$, as ConvNet is able to learn the shape information automatically from the images.

In Figure 3.8, we compare the galaxy purity and star completeness values for ConvNet, $\text{TPC}_{\text{morph}}$, and TPC_{phot} as a function of r-band magnitude for the differential counts in the SDSS data. We note that $\text{TPC}_{\text{morph}}$ outperforms the star-galaxy classifier used by the SDSS pipeline (i.e., an object is classified as a galaxy if concentration > 0.145) over all magnitudes. We do not show the SDSS classifications to avoid cluttering the plots. While ConvNet shows a similar but slightly worse performance than $\text{TPC}_{\text{morph}}$, the galaxy purity and star completeness values of ConvNet begin to drop at faint magnitudes $ilesssim21$. Again, TPC_{phot} performs significantly worse than both ConvNet and TPC at bright magnitudes. One reason that ConvNet fails to outperform TPC_{phot}, especially at faint magnitudes, might be its over-reliance on morphological features. Near a survey's limit, the measurement uncertainties generally increase, and morphology is not a reliable metric for star-galaxy classification. Another possibility is that data augmentation has a negative effect at faint magnitudes, as the network may get confused by additional examples of faint galaxies that look like point sources. Data augmentation however is indispensable, since it improves the overall performance greatly.

In Figure 3.9, we show the overall galaxy purity and star completeness values as a function

of magnitude for the integrated counts. ConvNet is able to maintain a galaxy purity of 0.9915 up to $i \sim 22.5$, while $\mathrm{TPC}_{\mathrm{morph}}$ provides a galaxy purity of 0.9977. $\mathrm{TPC}_{\mathrm{morph}}$ also outperforms ConvNet in terms of star completeness, maintaining a purity of 0.9810 up to $i \sim 22.5$, while the star completeness of ConvNet drops to 0.9500.

We also show the galaxy purity and star completeness values as a function of $g-r$ color in Figure 3.10. ConvNet performs slightly better than $\mathrm{TPC}_{\mathrm{morph}}$ in both galaxy completeness and star purity between $0.7lesssimg - rlesssim2.0$, where the stellar fraction is relatively low. On the other hand, both $\mathrm{TPC}_{\mathrm{morph}}$ and $\mathrm{TPC}_{\mathrm{phot}}$ outperform ConvNet in the region $g - rlesssim0.8$ where the stellar fraction is higher.

Figure 3.11 shows the calibration curves of ConvNet and $\mathrm{TPC}_{\mathrm{morph}}$. The calibration curve of ConvNet in Figure 3.11 is not as well-calibrated as the calibration curve in Figure 3.5, where the same ConvNet model was applied to the CFHTLenS data set. However, ConvNet may still be better calibrated than $\mathrm{TPC}_{\mathrm{morph}}$, even when it is applied to the SDSS data set. Although it is not straightforward to compare the two calibration curves by visual inspection, Table 3.4 shows that the CAL metric of ConvNet is lower than that of $\mathrm{TPC}_{\mathrm{morph}}$.

Figures 3.12 and 3.13 show the activations when images of a galaxy and a star are fed into the network. Similarly to Figures 3.6 and 3.7 in Section 3.6.2, the feature maps show hierarchical abstraction from low-level features in the first convolutional layer to high-level features in the subsequent layers. This hierarchical abstraction is what enables ConvNets to learn morphological features automatically from images.

3.7 Conclusions

We have presented a convolutional neural network for classifying stars and galaxies in the SDSS and CFHTLenS photometric images. For the CFHTLenS data set, the network is able to provide a classification that is as accurate as a random forest algorithm (TPC), while the probability estimates of our ConvNet model appear to be better calibrated. When the same

network architecture is applied to the SDSS data set, the network fails to outperform TPC, but the probabilities are still slightly better calibrated. The major advantage of ConvNets is that useful features are learned automatically from images, while traditional machine learning algorithms require feature engineering as a separate process to produce accurate classifications.

ConvNets have recently achieved record-breaking results in many image classification tasks (LeCun et al., 2015) and have been quickly and widely adopted by the computer vision community. One of the main reasons for the success is that ConvNets are general-purpose algorithms that are applicable to a variety of problems without the need for designing a feature extractor. The lack of requirement for feature extraction is a huge advantage, e.g., when the task is to classify 1,000 classes in the ImageNet data set (Russakovsky et al., 2015), as a good feature extractor for identifying images of cats would be of little use for classifying sailboats, and it is impractical to design a separate feature extractor for each class. However, when there already exists a good feature extractor for the problem at hand, e.g., the concentration parameter, the weight-averaged `spread_model` parameter from the Dark Energy Survey (Desai et al., 2012; Crocce et al., 2016), or even the `SExtractor` software, conventional machine learning algorithms that have been shown to be effective, such as TPC (Carrasco Kind and Brunner, 2013; Kim et al., 2015), remain a viable option. As the "no free lunch" theorem (Wolpert, 1996) states, there is no one model that works for every problem. For the CFHTLenS data set, our ConvNet model outperforms TPC. Since the SDSS catalog provides the concentration parameter that is highly optimized for star-galaxy classification, TPC works better for SDSS.

Although we used various techniques to combat overfitting, it is possible that our ConvNet model has overfit the data. Overfitting could explain why our ConvNet model with maximal information fails to significantly outperform a standard machine learning algorithm that uses the reduced summary information from catalogs. The most effective way to prevent overfitting would be to simply collect more training images with spectroscopic follow-up, as

the performance of ConvNets generally improves with more training data. However, spectroscopic observations are expensive and time-consuming, and it is unclear if sufficient training data will be available in future photometric surveys. If enough training data become available in DES or LSST, ConvNets become an attractive option because it can be applied immediately on reduced, calibrated images to produce well-calibrated posterior probabilities. We also note that using more training images will require the use of multi-GPU systems, which was beyond the scope of the present work.

Deep learning is a rapidly developing field, and recent developments include improved network architectures. For future work, we plan to train more ConvNet variants, such as the Inception Module (Szegedy et al., 2015) and Residual Network (He et al., 2015). To improve the predictive performance, we have combined the predictions of different models across multiple transformations of the input images (Section 3.4.3). To further improve the performance, we could also train several networks with different architectures and combine the models. For example, the winning solution of Dieleman et al. (2015a) for the Galaxy Zoo challenge was based on a ConvNet model, and it required averaging many sets of predictions from models with different neural network architectures. Futhermore, future work could compare the performance of other deep learning variants, such as deep belief networks (Hinton et al., 2006), deep Boltzmann machines (Salakhutdinov and Hinton, 2009), or multilayer perceptrons (Wasserman and Schwartz, 1988).

It is also likely that the performance will be improved not only by training multiple network architectures, but also by combining them with different star-galaxy classifiers. In Kim et al. (2015), we combined a purely morphological classifier, a supervised machine learning method (TPC), an unsupervised machine learning method based on self-organizing maps, and a hierarchical Bayesian template fitting method, and demonstrated that our combination technique improves the overall performance over any individual classification method. ConvNets could be included as a different machine learning paradigm in this classifier combination framework to produce further improvements in predictive performance.

Our ConvNet model is a supervised algorithm, and one of the criticisms of supervised techniques is their difficulty in extrapolating past the limits of available spectroscopic training data. Since it is difficult to assess the classification performance without a deeper spectroscopic sample, we evaluated the performance using a test set that is drawn from the same underlying distribution as the spectroscopic sample. However, when our ConvNet model —trained on sources from a spectroscopic sample— is applied to a photometric sample —which is often fainter than our training set— the performance of ConvNet will be less reliable. Combining our ConvNet model with unsupervised methods (e.g., a template fitting method) in the meta-classification framework in Chapter 2 will help improve the efficacy of star-galaxy classification beyond the limits of spectroscopic training data.

3.8 Figures and Tables

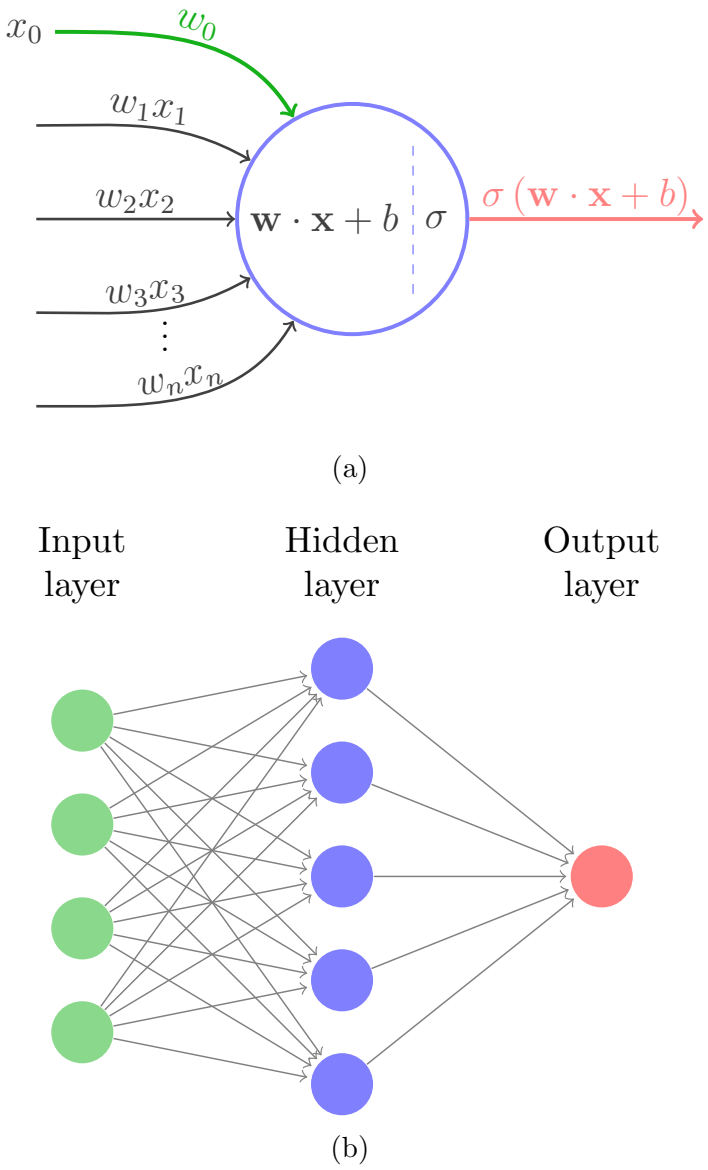


Figure 3.1: (a) A mathematical model of a biological neuron. (b) A schematic diagram of a neural network with one hidden layer.

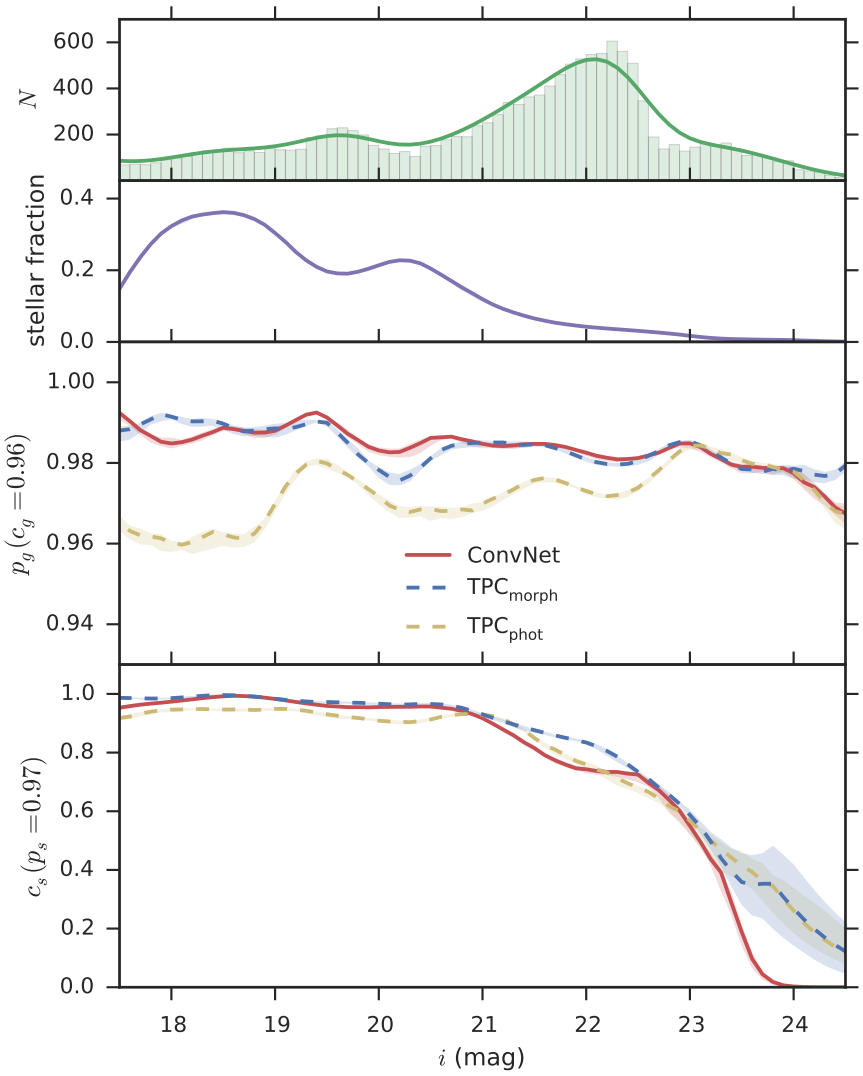


Figure 3.2: Galaxy purity and star completeness values as functions of the i-band magnitude (differential counts) as estimated by kernel density estimation (KDE) in the CFHTLenS data set. The top panel shows the histogram with a bin size of 0.1 mag and the KDE for objects in the test set. The second panel shows the fraction of stars estimated by KDE as a function of magnitude. The bottom two panels compare the galaxy purity and star completeness values for ConvNet (red solid line), TPC$_{\rm morph}$ (blue dashed line), and TPC$_{\rm phot}$ (yellow dashed line) as functions of magnitude. The 1σ confidence bands are estimated by bootstrap sampling.

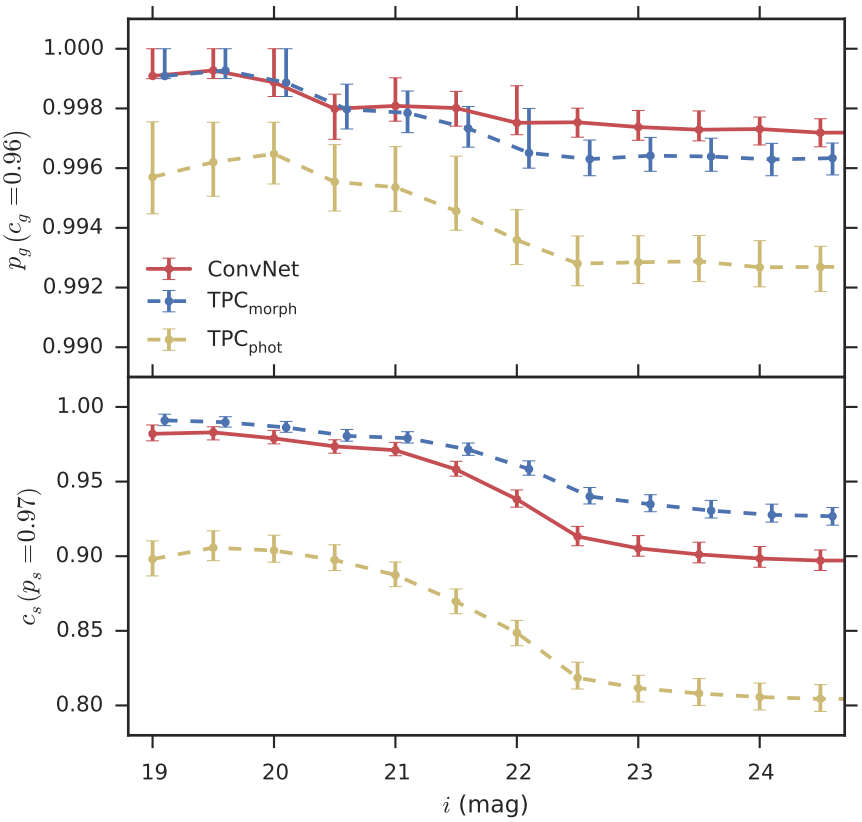


Figure 3.3: Galaxy purity and star completeness as functions of the i-band magnitude (integrated counts) in the CFHTLenS data set. The upper panel compares the galaxy purity values for ConvNet (red solid line), $\mathrm{TPC}_{\mathrm{morph}}$ (blue dashed line), and $\mathrm{TPC}_{\mathrm{phot}}$ (yellow dashed line). The lower panel compares the star completeness values. The 1σ error bars are computed following the method of Paterno, M (2003) to avoid the unphysical errors of binomial or Poisson statistics.

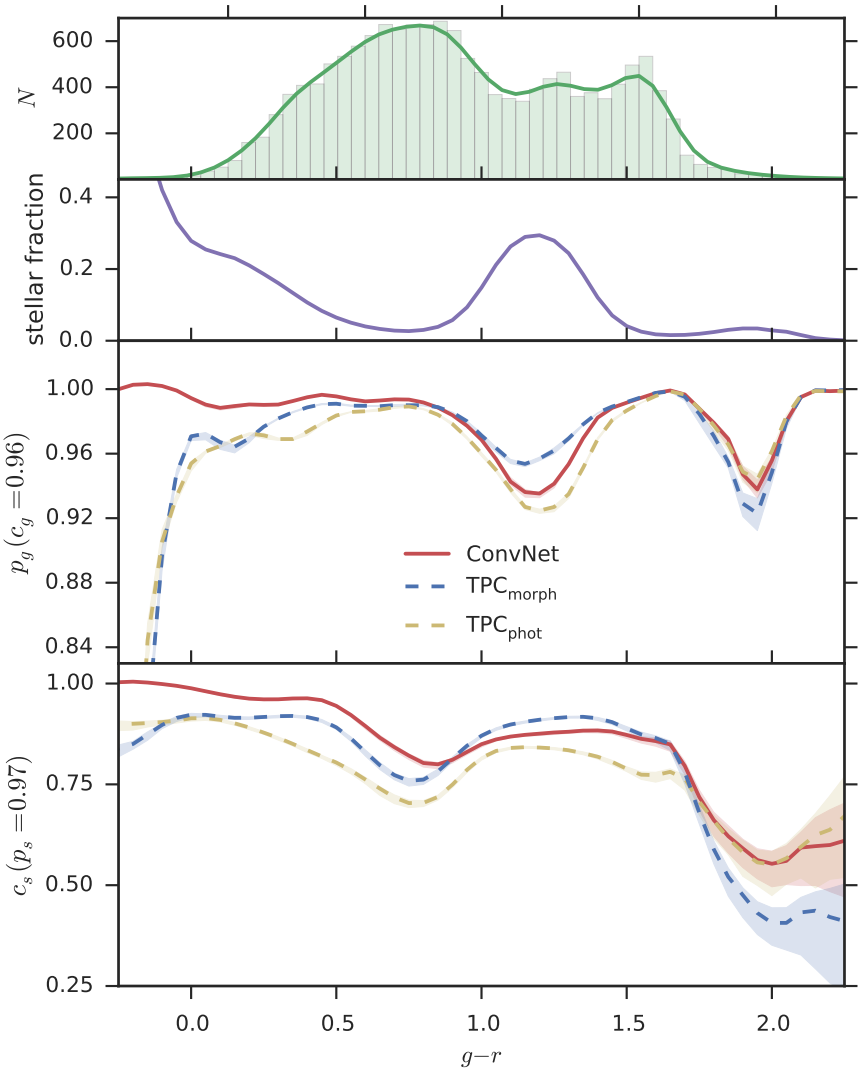


Figure 3.4: Similar to Figure 3.2 but as a function of $g-r$ color. The bin size of histogram in the top panel is 0.05.

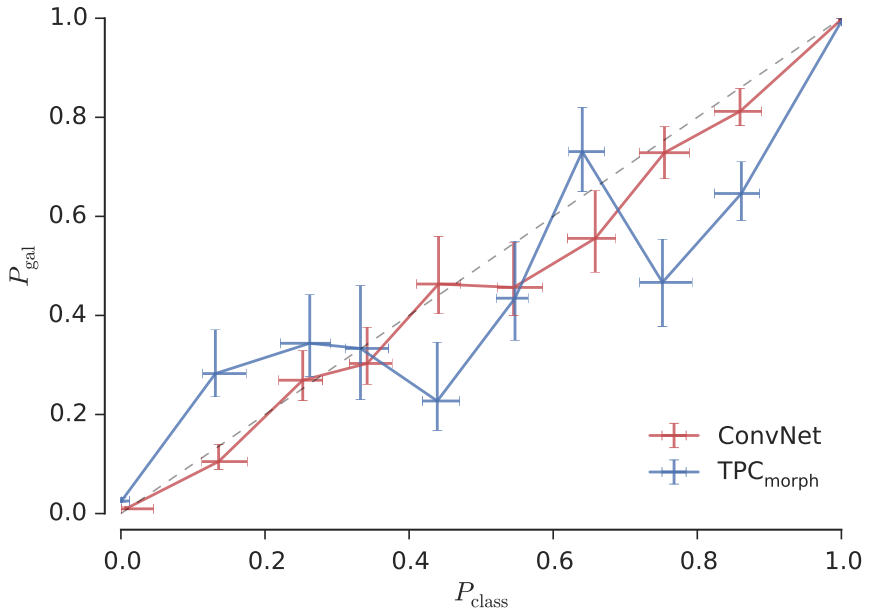
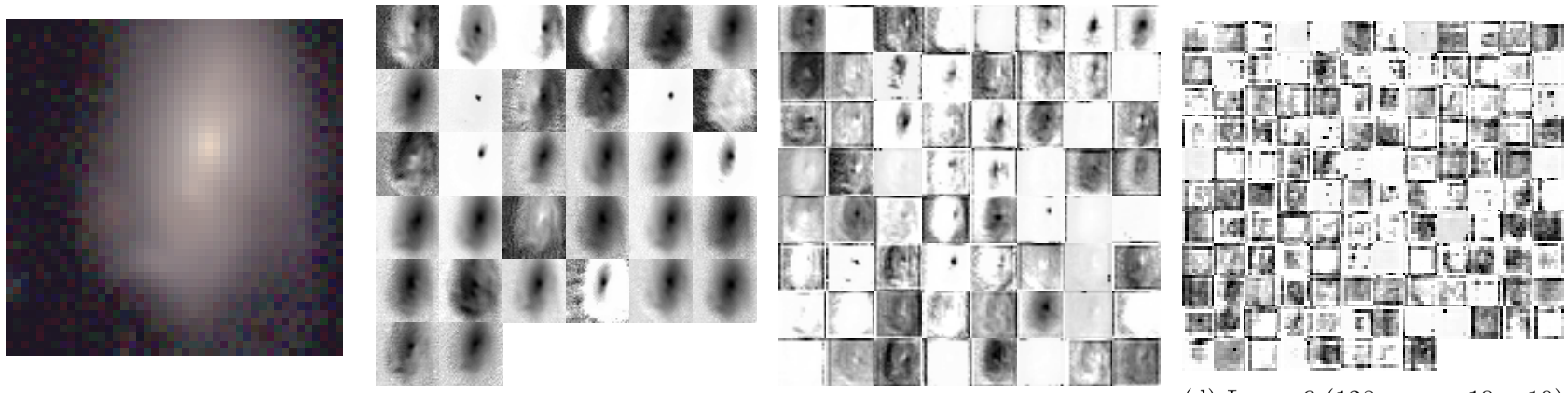
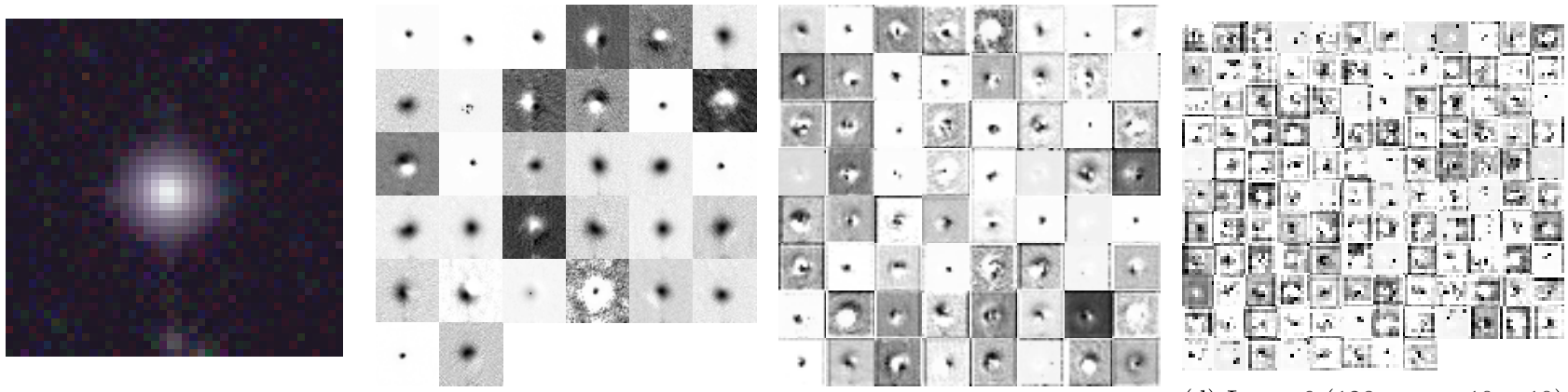


Figure 3.5: The calibration curves for ConvNet (red) and TPC$_{\rm morph}$ (blue) as applied to the CFHTLenS data set. $P_{\rm gal}$ is the fraction of objects that are galaxies, and $P_{\rm class}$ is the probabilistic outputs generated by the classifiers. The dashed line displays the relationship $P_{\rm gal} = P_{\rm conv}$. The 1σ error bars are computed following the method of Paterno, M (2003).



(a) Input (5 bands×44 × 44) (b) Layer 1 (32 maps×40 × 40) (c) Layer 3 (64 maps×20 × 20) (d) Layer 6 (128 maps×10 × 10)

Figure 3.6: (a) A sample 44×44 RGB image of a galaxy in the CFHTLenS data set. The RGB image is created by mapping R $\rightarrow$ i band magnitude, G $\rightarrow$ r band magnitude, and B $\rightarrow$ g band magnitude. (b) Activations on the first convolutional layer when a $5 \times 44 \times 44$ image is fed into the network. (c) Activations on the third convolutional layer. (d) Activations on the sixth convolutional layer. Each image in (b), (c), and (d) is a feature map corresponding to the output for one of the learned features.



(a) Input (5 bands×44 × 44) (b) Layer 1 (32 maps×40 × 40) (c) Layer 3 (64 maps×20 × 20) (d) Layer 6 (128 maps×10 × 10)

Figure 3.7: Similar to Figure 3.6 but for a star in the CFHTLenS data set.

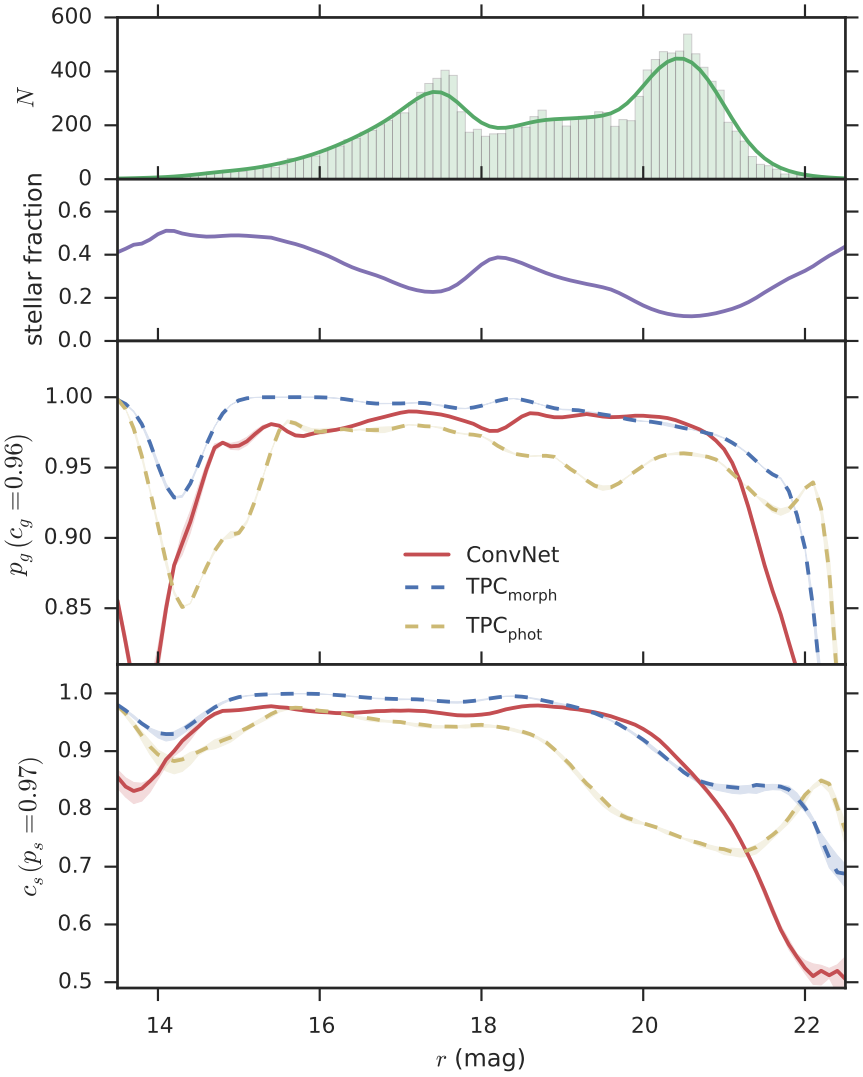


Figure 3.8: Galaxy purity and star completeness as function of the r-band magnitude for the differential counts in the SDSS data set.

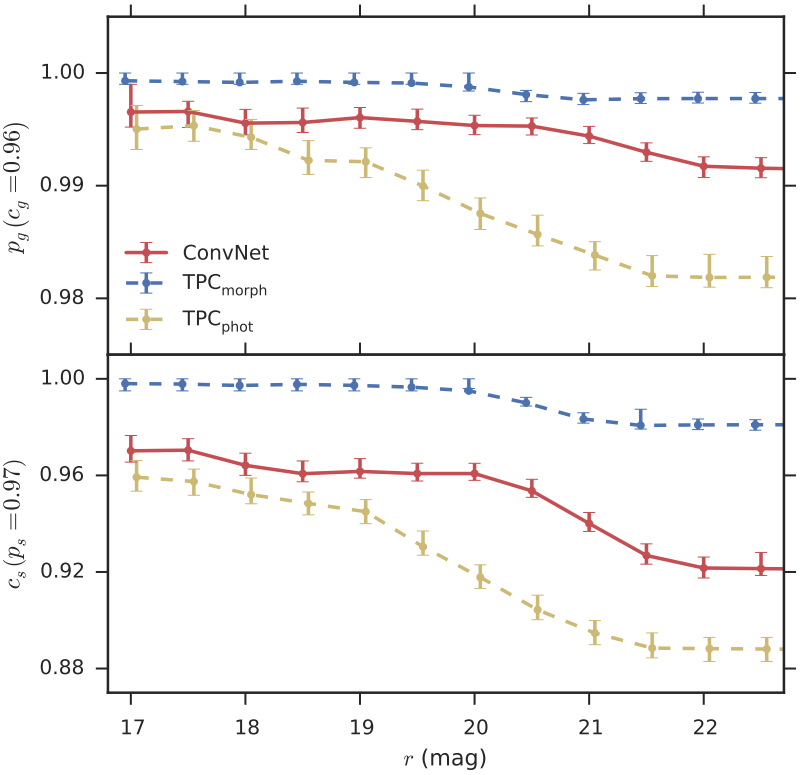


Figure 3.9: Galaxy purity and star completeness as functions of the r-band magnitude for the integrated counts in the SDSS data set.

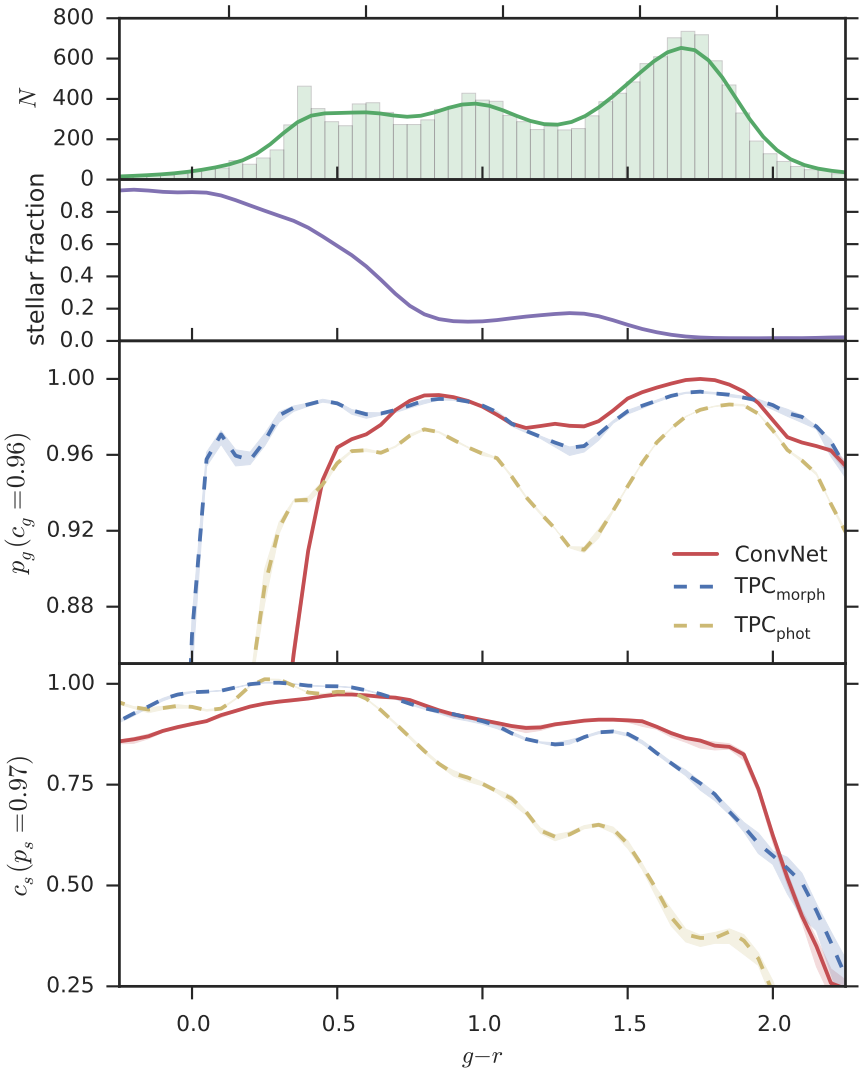


Figure 3.10: Similar to Figure 3.8 but as a function of $g-r$ color. The bin size of histogram in the top panel is 0.05.

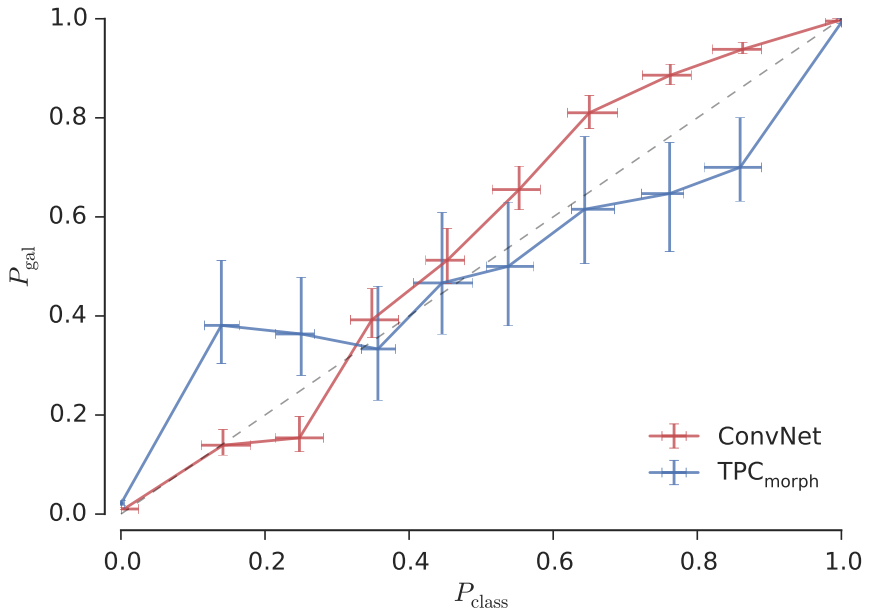
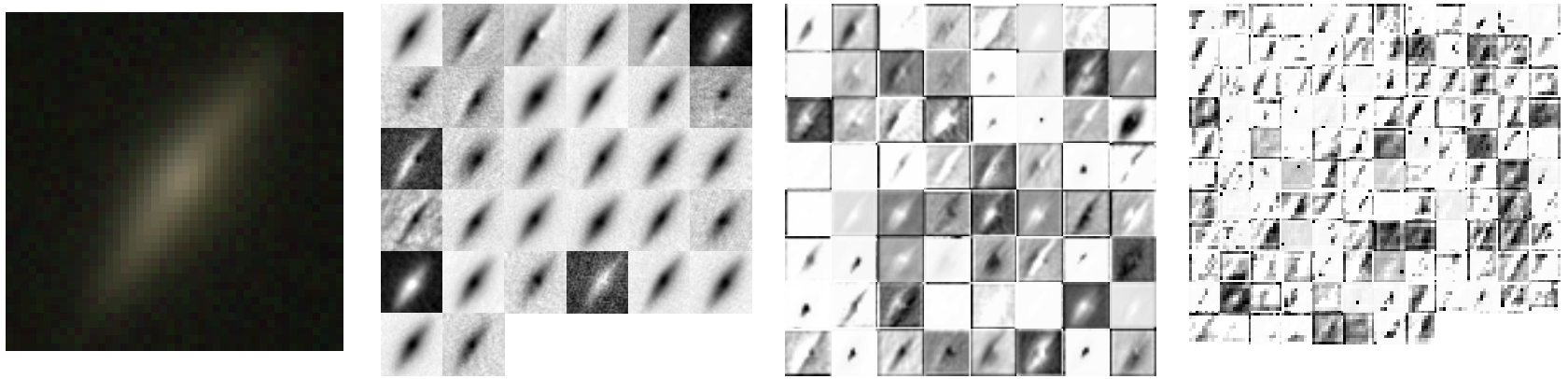
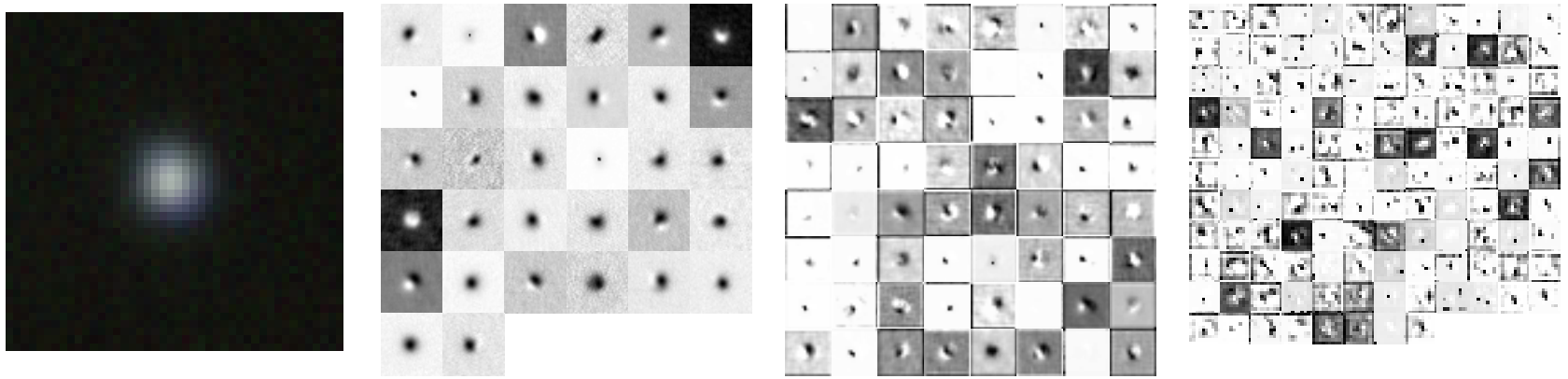


Figure 3.11: Calibration curves for ConvNet (red) and TPC$_{\mathrm{morph}}$ (blue) as applied to the SDSS data set.



(a) Input (5 bands×44 × 44) (b) Layer 1 (32 maps×40 × 40) (c) Layer 3 (64 maps×20 × 20) (d) Layer 6 (128 maps×10 × 10)

Figure 3.12: Similar to Figure 3.6 but for a galaxy in the SDSS data set.



(a) Input (5 bands×44 × 44) (b) Layer 1 (32 maps×40 × 40) (c) Layer 3 (64 maps×20 × 20) (d) Layer 6 (128 maps×10 × 10)

Figure 3.13: Similar to Figure 3.12 but for a star in the SDSS data set.

Table 3.1: Summary of ConvNet architecture and hyperparameters. Note that pooling layers have no learnable parameters.

type	filters	filter size	padding	non-linearity	initial weights	initial biases
convolutional	32	5×5	-	leaky ReLU	orthogonal	0.1
convolutional	32	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	64	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
convolutional	128	3×3	1	leaky ReLU	orthogonal	0.1
pooling	-	2×2	-	-	-	-
fully-connected	2048	-	-	leaky ReLU	orthogonal	0.01
fully-connected	2048	-	-	leaky ReLU	orthogonal	0.01
fully-connected	2	-	-	softmax	orthogonal	0.01

Table 3.2: The definition of the classification performance metrics.

Metric	Meaning
AUC	Area under the Receiver Operating Curve
MSE	Mean squared error
c_g	Galaxy completeness
p_g	Galaxy purity
c_s	Star completeness
p_s	Star purity
$p_g(c_g = x)$	Galaxy purity at x galaxy completeness
$c_s(p_s = x)$	Star completeness at x star purity
CAL	Calibration error with overlapping binning
$\mid \Delta N_g \mid /N_g$	Absolute error in number of galaxies
log loss	Cross-entropy

Table 3.3: A summary of the classification performance metrics as applied to the CFHTLenS data. The definition of the metrics is summarized in Table 3.2. The bold entries highlight the best performance values within each column. Note that $p_g(c_g = 0.96)$ and $c_s(p_s = 0.97)$ require adjusting threshold values (i.e., probability cuts), while other metrics do not. To obtain a galaxy completeness of $c_g = 0.96$, we choose the threshold values 0.9972, 0.9963, and 0.9927 for ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$, respectively; for star purity $p_s = 0.97$, we choose 0.6990, 0.5297, and 0.8570 for ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$, respectively.

classifier	AUC	MSE	$p_g(c_g = 0.96)$	$c_s(p_s = 0.97)$	CAL	$\lvert\Delta N_g\rvert/N_g$	log loss
ConvNet	**0.9948**	0.0112	**0.9972**	0.8971	**0.0197**	**0.0029**	**0.0441**
$\mathrm{TPC}_{\mathrm{morph}}$	0.9924	**0.0109**	0.9963	**0.9268**	0.0245	0.0056	0.0809
$\mathrm{TPC}_{\mathrm{phot}}$	0.9876	0.0189	0.9927	0.8044	0.0266	0.0101	0.1085

Table 3.4: A summary of the classification performance metrics as applied to the SDSS data. To obtain a galaxy completeness of $c_g = 0.96$, we choose the threshold values 0.7558, 0.9989, and 0.9360 for ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$, respectively; for star purity $p_s = 0.97$, we choose 0.6046, 0.0547, and 0.7449 for ConvNet, $\mathrm{TPC}_{\mathrm{morph}}$, and $\mathrm{TPC}_{\mathrm{phot}}$, respectively.

classifier	AUC	MSE	$p_g(c_g = 0.96)$	$c_s(p_s = 0.97)$	CAL	$\lvert\Delta N_g\rvert/N_g$	log loss
ConvNet	0.9952	0.0182	0.9915	0.9500	**0.0243**	0.0157	**0.0731**
$\mathrm{TPC}_{\mathrm{morph}}$	**0.9967**	**0.0099**	**0.9977**	**0.9810**	0.0254	**0.0044**	0.0914
$\mathrm{TPC}_{\mathrm{phot}}$	0.9886	0.0283	0.9819	0.8879	0.0316	0.0160	0.1372

3.9 References

Charu C Aggarwal. *Data classification: algorithms and applications*. CRC Press, 2014.

S. Alam et al. The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. *ApJS*, 219:12, July 2015.

N. M. Ball, R. J. Brunner, A. D. Myers, and D. Tcheng. Robust Machine Learning Applied to Astronomical Data Sets. I. Star-Galaxy Classification of the Sloan Digital Sky Survey DR3 Using Decision Trees. *ApJ*, 650:497–509, October 2006.

N. M. Ball et al. Robust machine learning applied to astronomical data sets. iii. probabilistic photometric redshifts for galaxies and quasars in the sdss and galex. *ApJ*, 683(1):12, 2008.

Manda Banerji, Ofer Lahav, Chris J Lintott, Filipe B Abdalla, Kevin Schawinski, Steven P Bamford, Dan Andreescu, Phil Murray, M Jordan Raddick, Anze Slosar, et al. Galaxy zoo: reproducing galaxy morphologies via machine learning. *MNRAS*, 406(1):342–353, 2010.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8624–8628. IEEE, 2013.

E. Bertin and S. Arnouts. Sextractor: Software for source extraction. *A&AS*, 117:393–404, June 1996.

Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

M. Carrasco Kind and R. J. Brunner. TPZ: photometric redshift PDFs and ancillary information by using prediction trees and random forests. *MNRAS*, 432:1483–1501, June 2013.

M. Carrasco Kind and R. J. Brunner. Exhausting the information: novel bayesian combination of photometric redshift pdfs. *MNRAS*, 442:3380–3399, August 2014.

Rich Caruana and Alexandru Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78. ACM, 2004.

M Crocce, J Carretero, AH Bauer, AJ Ross, I Sevilla-Noarbe, T Giannantonio, F Sobreira, J Sanchez, E Gaztanaga, M Carrasco Kind, et al. Galaxy clustering, photometric redshifts and diagnosis of systematics in the des science verification data. *MNRAS*, 455(4):4301–4324, 2016.

M. Davis et al. Science objectives and early results of the deep2 redshift survey. *Astronomical Telescopes and Instrumentation*, pages 161–172, 2003.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *The statistician*, pages 12–22, 1983.

S Desai, R Armstrong, JJ Mohr, DR Semler, J Liu, E Bertin, SS Allam, WA Barkhouse, G Bazin, EJ Buckley-Geer, et al. The blanco cosmology survey: Data acquisition, processing, calibration, quality diagnostics, and data release. *ApJ*, 757(1):83, 2012.

Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *MNRAS*, 450(2):1441–1459, 2015a.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

Sander Dieleman et al. Lasagne: First release., August 2015b.

Daniel J Eisenstein, David H Weinberg, Eric Agol, Hiroaki Aihara, Carlos Allende Prieto, Scott F Anderson, James A Arns, Éric Aubourg, Stephen Bailey, Eduardo Balbinot, et al. Sdss-iii: Massive spectroscopic surveys of the distant universe, the milky way, and extra-solar planetary systems. *AJ*, 142(3):72, 2011.

T. Erben et al. Cfhtlens: the canada–france–hawaii telescope lensing survey–imaging data and catalogue products. *MNRAS*, page stt928, 2013.

R. Fadely, D. W. Hogg, and B. Willman. Star-Galaxy Classification in Multi-band Optical Imaging. *ApJ*, 760:15, November 2012.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4): 193–202, 1980.

B. Garilli et al. The vimos vlt deep survey-global properties of 20 000 galaxies in the iab ¡ 22.5 wide survey. *A&A*, 486(3):683–695, 2008.

B. Garilli et al. The vimos public extragalactic survey (vipers)-first data release of 57 204 spectroscopic measurements. *A&A*, 562:A23, 2014.

Stephen DJ Gwyn. The canada-france-hawaii telescope legacy survey: Stacked images and catalogs. *AJ*, 143(2):38, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

M. Henrion, D. J. Mortlock, D. J. Hand, and A. Gandy. A Bayesian approach to star-galaxy classification. *MNRAS*, 412:2286–2302, April 2011.

C. Heymans et al. Cfhtlens: the canada–france–hawaii telescope lensing survey. *MNRAS*, 427(1):146–166, 2012.

H. Hildebrandt et al. Cfhtlens: improving the quality of photometric redshifts with precision photometry. *MNRAS*, 421(3):2355–2367, 2012.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Geoffrey E Hinton et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Ben Hoyle. Measuring photometric redshifts using galaxy images and deep neural networks. *arXiv preprint arXiv:1504.07255*, 2015.

M. Huertas-Company, R. Gravet, G. Cabrera-Vives, P. G. Prez-Gonzlez, J. S. Kartaltepe, G. Barro, M. Bernardi, S. Mei, F. Shankar, P. Dimauro, E. F. Bell, D. Kocevski, D. C. Koo, S. M. Faber, and D. H. Mcintosh. A catalog of visual-like morphologies in the 5 candels fields using deep learning. *ApJS*, 221(1):8, 2015.

Željko Ivezić et al. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data*. Princeton University Press, 2014.

Harshil M Kamdar, Matthew J Turk, and Robert J Brunner. Machine learning and cosmological simulations–i. semi-analytical models. *MNRAS*, 455(1):642–658, 2016.

E. J. Kim and R. J. Brunner. Star–galaxy classification using deep convolutional neural networks. *MNRAS*, 464(4):4463–4475, 2017.

E. J. Kim, R. J. Brunner, and M. Carrasco Kind. A hybrid ensemble learning approach to star-galaxy classification. *MNRAS*, 453(1):507–521, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

O. Le Fèvre et al. The vimos vlt deep survey-first epoch vvds-deep survey: 11 564 spectra with 17.5 ¡= iab ¡= 24, and the redshift distribution over 0 ¡ z ¡= 5. *A&A*, 439(3):845–862, 2005.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Yann LeCun et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a.

Yann A LeCun et al. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998b.

Nolan Li and Ani R Thakar. Casjobs and mydb: A batch query workbench. *Computing in Science & Engineering*, 10(1):18–29, 2008.

R. H. Lupton, J. E. Gunn, and A. S. Szalay. A Modified Magnitude System that Produces Well-Behaved Magnitudes, Colors, and Errors Even for Low Signal-to-Noise Ratio Measurements. *AJ*, 118:1406–1410, September 1999.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML*, 30(1), 2013.

K. Monteith, J. L. Carroll, K. Seppi, and T. Martinez. Turning bayesian model averaging into bayesian model combination. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2657–2663. IEEE, 2011.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

J. A. Newman et al. The deep2 galaxy redshift survey: design, observations, data reduction, and redshifts. *ApJS*, 208(1):5, 2013.

S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys, and W. A. Zumach. Automated star/galaxy discrimination with neural networks. *AJ*, 103:318–331, January 1992.

Paterno, M. Calculating efficiencies and their uncertainties. `http://home.fnal.gov/~paterno/images/effic.pdf`, May 2003.

LČ Popović et al. Photocentric variability of quasars caused by variations in their inner structure: consequences for gaia measurements. *A&A*, 538:A107, 2012.

Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document, 1961.

AJ Ross et al. Ameliorating systematic uncertainties in the angular clustering of galaxies: a study using the sdss-iii. *MNRAS*, 417(2):1350–1373, 2011.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *AISTATS*, volume 1, page 3, 2009.

C Sánchez, M Carrasco Kind, Huan Lin, Robi Miquel, Filipe B Abdalla, A Amara, M Banerji, C Bonnett, R Brunner, D Capozzi, et al. Photometric redshift analysis in the dark energy survey science verification data. *MNRAS*, 445(2):1482–1506, 2014.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

D. J. Schlegel, D. P. Finkbeiner, and M. Davis. Maps of Dust Infrared Emission for Use in Estimation of Reddening and Cosmic Microwave Background Radiation Foregrounds. *ApJ*, 500:525–553, June 1998.

Hee-Jong Seo, Shirley Ho, Martin White, Antonio J Cuesta, Ashley J Ross, Shun Saito, Beth Reid, Nikhil Padmanabhan, Will J Percival, Roland de Putter, et al. Acoustic scale from the angular power spectra of sdss-iii dr8 photometric luminous galaxies. *ApJ*, 761 (1):13, 2012.

Ignacio Sevilla-Noarbe and Penélope Etayo-Sotos. Effect of training characteristics on object classification: an application using boosted decision trees. *Astronomy and Computing, in press (arXiv:1504.06776)*, 2015. doi: 10.1016/j.ascom.2015.03.010. URL `http://dx.doi.org/10.1016/j.ascom.2015.03.010`.

Bernard W Silverman. Density estimation for statistics and data analysis. *CRC press*, 26, 1986.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

M. T. Soumagnac et al. Star/galaxy separation at faint magnitudes: Application to a simulated dark energy survey. *MNRAS*, 450:666–680, jun 2015.

A. A. Suchkov, R. J. Hanisch, and B. Margon. A Census of Object Types and Redshift Estimates in the SDSS Photometric Catalog from a Trained Decision Tree Classifier. *AJ*, 130:2439–2452, December 2005.

Ilya Sutskever et al. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1139–1147, 2013.

John A Swets, Robyn M Dawes, and John Monahan. Better decisions through. *Scientific American*, page 83, 2000.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL `http://arxiv.org/abs/1605.02688`.

E. C. Vasconcellos, R. R. de Carvalho, R. R. Gal, F. L. LaBarbera, H. V. Capelato, H. Frago Campos Velho, M. Trevisan, and R. S. R. Ruiz. Decision Tree Classifiers for Star/Galaxy Separation. *AJ*, 141:189, June 2011.

Philip D Wasserman and Tom Schwartz. Neural networks. ii. what are they and why is everybody so interested in them now? *IEEE Expert*, 3(1):10–15, 1988.

N. Weir, U. M. Fayyad, and S. Djorgovski. Automated Star/Galaxy Classification for Digitized Poss-II. *AJ*, 109:2401, June 1995.

David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

Donald G York et al. The sloan digital sky survey: Technical summary. *AJ*, 120(3):1579, 2000.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, volume 1, pages 609–616. Citeseer, 2001.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer, 2014.

Chapter 4

Star-galaxy Classification Using Semi-Supervised Generative Adversarial Networks

4.1 Introduction

The Sloan Digital Sky Survey (SDSS; York et al., 2000), one of the most successful surveys to date, has obtained photometric observations of more than 3×10^8 objects, covering more than one-third of the sky. The SDSS has also conducted spectroscopic follow-up observations of more than 3×10^6 objects (Eisenstein et al., 2011). However, the spectroscopic sample is about one hundred times smaller than the photometric sample, since spectroscopy is considerably more expensive than photometry in terms of telescope time. Due to the difficulty of making spectroscopic measurements, modern, large-scale surveys, such as the Dark Energy Survey (DES), and the Large Synoptic Survey Telescope (LSST), are purely photometric surveys. The scarcity of spectroscopic samples will only be exacerbated in the current and next generation of surveys, as they probe larger cosmological volumes and the majority of photometric observations become too faint for a uniform spectroscopic follow-up.

Machine learning techniques have been a popular method for classifying stars and galaxies in large sky surveys. Odewahn et al. (1992) pioneered the use of artificial neural networks in star-galaxy classification. Some of the more recent examples of applying machine learning

This chapter contains material from the following previously published article:

- E. J. Kim and R. J. Brunner. Star–galaxy classification using semi-supervised generative adversarial neural networks. *MNRAS*, Submitted

techniques to the star-galaxy classification problem include decision trees (e.g., Ball et al., 2006), support vector machines (e.g., Fadely et al., 2012), and convolutional neural networks (Kim and Brunner, 2017). All of these techniques are *supervised* learning algorithms, where the input attributes (e.g., magnitudes or colors) are provided along with the truth labels (e.g., star or galaxy). We must be careful when extrapolating these supervised algorithms beyond the limits of the labeled data, because most machine learning methods commonly assume that the labeled samples are governed by the same or similar underlying data distribution as the target distribution where we apply our model to make predictions.

Semi-supervised learning falls between supervised learning, where training data are completely labeled, and unsupervised learning, where all training data are unlabeled. Semi-supervised techniques make use of a large amount of unlabeled data, in conjunction with a small amount of labeled data, to better capture the underlying data distribution. Semi-supervised learning is of great interest because there will be many orders of magnitude more unlabeled than labeled data available in future ground-based imaging surveys.

Generative adversarial networks (GANs) — a class of state-of-the-art deep learning algorithms commonly used in image-to-image translation applications — are usually unsupervised, but semi-supervised learning variants of GANs have recently been introduced (Springenberg, 2016; Salimans et al., 2016; Odena, 2016). Unsupervised GANs have previously been applied to several astronomical applications, but the main focus in these studies is image generation. For example, using the COSMOS data, Ravanbakhsh et al. (2017) trained Variational Autoencoders—another commonly used generative model—and GANs to generate new galaxy images with the goal of synthesizing calibration sets for weak gravitational lensing. Mustafa et al. (2017) applied a GAN model to the problem of generating cosmological weak lensing convergence maps. Schawinski et al. (2017) showed that GANs are able to recover detailed features such as galaxy morphology from artificially degraded images of low-redshift galaxies.

In this chapter, we study the application of GANs in generating realistic images of not

only galaxies, but also the stars and quasi-stellar objects (QSOs or quasars) in ground-based photometric surveys. More importantly, we demonstrate that semi-supervised GANs are able to produce accurate and well-calibrated classifications using only a small number of labeled examples. The rest of the chapter is organized as follows. In Section 4.2, we describe the SDSS data set and the image pre-processing steps. In Section 4.3, we provide a brief overview of semi-supervised GANs. In Section 4.4, we present the main results of image generation and source classification using our semi-supervised GAN model. Finally, we outline our conclusions in Section 4.5.

4.2 Data

To demonstrate the performance of our approach, we follow a similar approach to Chapter 3. However, we restrict our analysis to data from the SDSS survey in this chapter. We use SDSS because of the the large number of objects and the concurrent spectroscopy. In this section, we briefly describe the SDSS data set and the image pre-processing steps for preparing training examples.

The SDSS survey, one of the largest astronomical surveys that has ever existed, covers 14,555 square degrees in five bands: u, g, r, i, and z. The twelfth data release (Alam et al., 2015), the final data release from SDSS phases I–III, contains photometry of over 3×10^8 objects with a limiting magnitude of $r \approx 22$. More than 3×10^6 objects from the photometric catalog were also targeted for spectroscopy (Eisenstein et al., 2011).

In a semi-supervised learning setting, we typically have a large amount of unlabeled data and a small amount of labeled data. Thus, the data sets we use in this work consists of an *unlabeled* training set, a *labeled* training set, and a blind, *labeled* test set. For the unlabeled training set, we use the `photoObj` view of the publicly available CasJobs server [1] (Li and Thakar, 2008), and randomly select 1×10^6 objects. We exclude objects with bad

[1] `http://skyserver.sdss.org/casjobs/`

photometric observations as follows. We only include objects with magnitudes between 0 and 40; we consider only objects with the half-light radius (as measured by the exponential and de Vaucouleurs light profiles) in the r band less than 30 arc seconds; we also exclude any objects with warning flags in the photometric measurement; and we exclude any images with missing or masked values. After using the CasJobs server to select objects with clean photometry, we download the FITS images for SDSS fields covering these objects. Using the astrometry information in the FITS headers and the `Montage` [2] software, we align each image in u, g, i, and z bands to the r-band image. To generate cutout images of size 32×32 pixels that can be used as input to our GAN framework, we use `SExtractor` to center each object in the cutout image. Furthermore, we convert all flux values in the FITS files to luptitudes (i.e., inverse hyperbolic sine magnitudes; Lupton et al., 1999). Finally, we use the dust reddening map of Schlegel et al. (1998) to account for extinction. Sample postage stamps of typical galaxies, stars, and quasars are shown in Figure 4.1.

The labeled training set and the test set are drawn from objects in the `specObj` view of the DR12 catalog. We randomly select a total of 1×10^6 sources from `SpecObj`, and follow the same image pre-processing steps. In addition to removing bad photometry similarly to the unlabeled data, we exclude some bad spectroscopic observations by only including objects with no warning flags in the spectroscopic measurement (i.e., `zWarning = 0`) and sources with spectroscopic redshift less than 2 or its error less than 0.1. We randomly split the labeled objects into a blind test set of size 2×10^5 and multiple labeled training sets with only a small number of labeled data in each set for running multiple experiments (See Section 4.4.2). To optimize the model, we perform eightfold cross-validation on the labeled training set.

We emphasize that this setup is considerably different from a typical training-test split used in a supervised setting. In a supervised setting, the learning algorithm would be trained on the SDSS spectroscopic sample although it would eventually have to be applied to the

[2] `http://montage.ipac.caltech.edu/`

SDSS photometric sample. As shown in Figure 4.3, the SDSS photometric sample is considerably fainter than the SDSS spectroscopic sample, and they have different distributions due to the target selection process. This can become a major drawback of any supervised algorithms since most machine learning algorithms assume that both training and test sets are drawn from the same parent distribution, However, in our semi-supervised setting, only a small number of labeled data are used for training, and the vast majority of training data are drawn from the SDSS photometric sample. As discussed in the following sections, this could be an advantage for semi-supervised learning because it will not only require a significantly smaller number of labeled examples to obtain a comparable performance but also will extrapolate better beyond the limits of the training data.

4.3 Methods

Generative adversarial networks (GANs; Goodfellow et al., 2014) are a class of algorithms where a generative model is pitted against a discriminative model. In this section, we briefly introduce the standard, unsupervised GAN, and describe how we can perform semi-supervised learning by replacing the discriminator in conventional GANs with a classifier. We also briefly present how uncertainty analysis of model predictions can be performed in deep learning. For more details, interested readers are referred to the references herein.

4.3.1 Semi-Supervised Generative Adversarial Networks

In most GANs, the generator network G takes random noise $p_z(z)$ as input and produces samples x from the data distribution $p(x)$. The discriminator network D is then trained to classify real data and fake samples from the generator G. In the adversarial setting, the generator is trained to fool the discriminator into classifying its fake instances as real. In other words, we train D to maximize $D(x)$, the probability of classifying real training examples as real, and maximize $\log(1 - D(G(z)))$, the probability of classifying samples

from G as fake. In the original GAN framework of Goodfellow et al. (2014), the loss function for training D is

$$L = -\mathbb{E}_{x\sim p} \log D(x) - \mathbb{E}_{z\sim p_z} \log\left(1 - D\left(G\left(z\right)\right)\right), \tag{4.1}$$

and G is trained to minimize $\log\left(1 - D\left(G\left(z\right)\right)\right)$. When the generator and the discriminator are trained alternatively to converge to a fixed point, the fake samples generated by G are realistic enough to fool the discriminator.

Traditionally, the discriminator network in a normal GAN employs a binary classification, but it can also be implemented with any standard classifier that classifies the input x into one of K possible classes (Salimans et al., 2016; Odena, 2016). In this modified setting, we increase the number of output classes in our classifier from K to $K+1$, where real data are classified into the first K classes and samples from the GAN generator G are classified into the new $(K+1)$-th class. We now have a semi-supervised classifier, since we can use unlabeled data to maximize $P_{\mathrm{C}}\left(y \leq K \mid x\right)$, the probability that the classifier C classifies input into one of the K classes. Our loss function for training C becomes

$$L = L_{\text{supervised}} + L_{\text{unsupervised}} \tag{4.2}$$

$$L_{\text{supervised}} = -\mathbb{E}_{x,y\sim p} \log P_{\mathrm{C}}\left(y \mid x, y \leq K\right) \tag{4.3}$$

$$\begin{aligned} L_{\text{unsupervised}} = &-\mathbb{E}_{x\sim p} \log P_{\mathrm{C}}\left(y \leq K \mid x\right) \\ &- \mathbb{E}_{z\sim p_z} \log P_{\mathrm{C}}\left(y = K+1 \mid x = G\left(z\right)\right). \end{aligned} \tag{4.4}$$

Note that we recover Equation 4.1 when we substitute $P_{\mathrm{C}}\left(y = K+1 \mid x\right) = 1 - D(x)$ or $P_{\mathrm{C}}\left(y \leq K \mid x\right) = D(x)$ into Equation 4.4, and our unsupervised loss function $L_{\text{unsupervised}}$ is therefore equivalent to the original GAN objective.

We also use *feature matching*, one of the techniques proposed by Salimans et al. (2016) to address the instability of GANs. A major hurdle in training GANs is mode collapse, where the generator overtrains on the current discriminator, and fake samples from the generator

capture only a few of modes from the data. In feature matching GANs, the objective of the generator is to match the statistics between the generator distribution and the real distribution rather than minimize $\log(1 - D(G(z)))$. Specifically, we train G to minimize $\left\|\mathbb{E}_{x\sim p} f(x) - \mathbb{E}_{z\sim p_z} f(G(z))\right\|_2^2$, where $f(x)$ denote the features (i.e., activations) on the final intermediate layer before the fully-connected layer.

Following Salimans et al. (2016), we use deconvolutional layers, batch normalization, weight normalization (Salimans and Kingma, 2016), and leaky ReLU activation functions in our generator network. For optimization, we use Adam optimizer with exponential decay rate of 0.5 for the first moment estimates. Our discriminator network consists of seven convolutional layers, two network-in-network layers (Lin et al., 2013), one global pooling layer, and a fully connected layer. We also use dropout, weight normalization, leaky ReLU activation functions, and Adam optimizer in our discriminator.

4.3.2 Dropout Sampling

To obtain predictive probabilities, standard deep learning techniques use the softmax function at the end of the pipeline. However, using the softmax output does not adequately address model uncertainty, because most algorithms pass the predictive mean through the softmax rather than the entire probability distribution. Gal and Ghahramani (2016) have recently shown that model uncertainty can be obtained from deep neural networks that use a technique called dropout. Dropout is a technique commonly used to avoid overfitting, and we select a random subset of hidden neurons in the previous layer and set the output of these neurons to zero. Dropout has been a ubiquitous technique in deep learning since Hinton et al. (2012) proposed it as a way to avoid overfitting, but most deep learning practitioners do not utilize the information contained in the dropout layers. To estimate the predictive mean and predictive uncertainty, we simply collect Monte Carlo samples from the networks and then compute the standard deviation over the softmax outputs of the samples. This adds almost no additional computational cost at training time and often improves predictive

performance.

4.4 Results

In this section, we first evaluate the quality of generated images by comparing the magnitude and size distributions between real and generated images. We then present the classification performance of our semi-supervised GAN model on the SDSS data set. We also demonstrate that dropout sampling not only improves the performance but also enables us to obtain the model uncertainty.

4.4.1 Image Statistics

By using GANs for semi-supervised learning, we have an added benefit of being able to synthesize new images of stars, galaxies, and quasars. Samples of generated images for different magnitude ranges are shown in Figure 4.2. Comparing the generated images to Figure 4.1, the bright objects in GAN generated images are slightly blurry and lack fine details, while the faint objects appear similar to the real SDSS images.

Generative models could potentially be an inexpensive alternative to image simulation pipelines, such as the *GalSim* package (Rowe et al., 2015), which require high-quality images with high resolution and signal-to-noise ratio as input to the pipeline. However, it is important to evaluate the goodness-of-fit in order to use generative models in practical applications. The evaluation of generative models is still an active area of research (Theis et al., 2016). In the deep learning community, where the focus is on natural images, many researchers rely on visual inspection to assess the quality of images generated by GANs. In the case of scientific applications of generative models, however, we can also assess the quality of the generated images by studying the characteristic image statistics.

The most relevant image statistics for star-galaxy classification are the magnitude and size (i.e., half-light radius) distributions. Figure 4.4 compares the r-band magnitude distribution

of generated images to that of the SDSS photometric sample. To compare the two probability distributions, we also generate a quantile-quantile plot (Q-Q plot; Wilk and Gnanadesikan, 1968) by plotting their quantiles against each other. From visual inspection of Figure 4.5, it is clear that most points in the Q-Q plot approximately lie on the 45° line $y = x$, although objects generated by GAN are slightly fainter at bright magnitudes $r \lesssim 19.5$. This is also consistent with our visual inspection of Figure 4.1, where bright objects appear blurry but faint objects are similar to real objects. However, since the majority of objects are faint, the overall magnitude distribution of GAN generated images are in good agreement with that of real images, and our GAN model has successfully reproduced the overall data distribution in the original images.

In Figure 4.6, we compare the half-light radius (estimated by `SExtractor`'s `FLUX_RADIUS` parameter) distribution of generated images to that of the test set. The Q-Q plot in Figure 4.7 shows that the size distribution of generated images is more dispersed and has heavier tails than the original size distribution. Although the the samples generated by our GAN model are slightly larger than real objects, it is only for a small number of outliers, and the overall size distribution of the generated images is in good agreement with that of the original data.

4.4.2 Classification Performance

Although our GAN generator is able to learn the original distribution of pixel intensities, our main focus is on semi-supervised classification performance. We perform semi-supervised training with a small random subset of the spectroscopic sample, and the remaining training images are drawn from the photometric sample. We perform eightfold cross-validation on the labeled training data to evaluate the cross-entropy (also called log loss; Murphy, 2012), and the model that minimizes the cross-validation error is chosen as the best model. Here,

the cross-entropy is defined as

$$H = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} y_{i,k}\log\hat{y}_{i,k}, \tag{4.5}$$

where $y_{i,k}$ is the true labels (i.e., $y_{i,k} = 1$ if sample i has label k) and $\hat{y}_{i,k}$ is the probability prediction made by the models. Our final model is then applied to the blind test set, where we compare the model predictions with spectroscopic labels. Figure 4.8 shows the cross-entropy as a function of the number of labeled examples. As expected, we get better performance as we increase the number of labeled data, but most of the improvement in performance comes from the first 1×10^3 labeled examples. Thus, we use 1,024 labeled examples in the following sections.

Our GAN classifier is a probabilistic classifier, and its softmax function outputs a multi-class categorical probability distribution. Ideally, the probability distribution would be used in subsequent scientific analyses. For example, we can in principle remove the contamination effect of stars when measuring auto-correlation functions of luminous galaxies by weighting each object by the probability that it is a galaxy (Ross et al., 2011). Thus, the probability estimates for each class should reflect the proportion of objects that actually belong to that class. In Figure 4.9, we show the calibration curve (or reliability curves; DeGroot and Fienberg, 1983), where we bin the probability estimates and plot the true fraction of positive examples versus the probabilities assigned by the classifier. The calibration curve for galaxies is nearly diagonal and well-calibrated, so we can confidently use the classifier output to estimate the probability that an object is a galaxy. The calibration curve for stars is also nearly diagonal and well-calibrated, while the probabilities assigned to quasars are not as accurate as stars or galaxies. This might be due to the relatively small number of quasars in the training set. Furthermore, quasars appear as point sources in photometry, rather than extended sources like galaxies, and photometric images of quasars often have one or more saturated pixels. The GAN classifier's over-reliance on morphological features would worsen

the classification performance for quasars.

Probability estimates can also be converted into discrete class labels by choosing a probability threshold. A naive way is to choose the most probable class label as the final classification. However, this simple approach is not optimal in cases where there is class imbalance, certain science requirements have to be satisfied, or misclassifying one class is more costly than misclassifying the others. Thus, it is ideal to adjust the classification decision threshold by considering the model's operating conditions. For example, the optimal star-galaxy catalog for fast-transient surveys would produce a pristine sample of point sources, and Miller et al. (2017) adjust the threshold to identify as many point sources as possible, while minimizing the number of galaxies misclassified as stars. Following Chapter 3, we adopt the weak lensing science requirements of the DES as a realistic operating condition. The weak lensing science measurements of the DES require $c_g > 0.960$ and $p_g > 0.778$ to control the statistical and systematic errors on the cosmological parameters, and $c_s > 0.250$ and $p_s > 0.970$ for stellar Point Spread Function (PSF) calibration (Soumagnac et al., 2015). Although other surveys, such as SDSS or LSST, will likley have different science requirements, we adopt these values, and compute the galaxy purity at 96% completeness and the star purity at 25% completeness. Here, we define the galaxy completeness c_g as the number of true galaxies classified as galaxies compared to the total number of galaxies in the whole sample:

$$c_g = \frac{N_g}{N_g + M_g}, \tag{4.6}$$

where N_g is the number of true galaxies classified as galaxies, and M_g is the number of true galaxies classified as stars or quasars. The galaxy purity p_g is defined as the fraction of true galaxies among objects that are classified as galaxies:

$$p_g = \frac{N_g}{N_g + M_s + M_q}, \tag{4.7}$$

where M_s is the number of true stars classified as galaxies, and M_q is the number of true

quasars classified as galaxies. We define the completeness and purity for stars and quasars in a similar way.

In Figure 4.10, we show the galaxy completeness and purity values as a function of apparent magnitude in the r band. For galaxies, the overall purity is 98.1% at 96.0% completeness. Comparing these values to the results of our previous work (Kim and Brunner, 2017), where a supervised method using convolutional neural networks was shown to achieve a purity of 99.1% at 96.0% completeness, our semi-supervised approach performs slightly worse than the state-of-the-art supervised algorithm on the spectroscopic sample. However, the supervised method was trained on 5×10^5 labeled data, while our semi-supervised approach uses only 10^3 labeled examples. Furthermore, it is highly likely that supervised algorithms, which are trained on the spectroscopic sample, will show worse performance when they are applied on the photometric sample, since we are extrapolating our models beyond the limits of training data. In contrast, since our semi-supervised algorithm is trained on the photometric sample, we are extrapolating beyond the limits of underlying data distribution when we measure its performance on the spectroscopic sample. Thus, when our semi-supervised approach is applied to the photometric sample, its performance will likely be competitive to that of supervised algorithms, and it may even significantly outperform supervised learning on the SDSS photometric sample.

In Figure 4.11, we show the star purity as a function of r-band magnitude. For stars, the overall purity is 99.9% at 25.0% completeness. To choose the threshold for assigning quasar classifications, we maximize the metric $\sqrt{c_q^2 + p_q^2}$. We show the completeness and purity values for quasars in Figure 4.12. The overall purity of the quasars is rather low at 80.6% but the completeness 90.3%. Figure 4.12 shows that the low purity is due to quasars at $r \gtrsim 20.5$, where counts reach their peak. The low value may also be due to the fact that there are relatively small number of training examples for quasars.

4.4.3 Uncertainty

As mentioned in Section 4.3.2, although many deep neural network architectures are trained with dropout, it is usually not used at testing time. In Figure 4.13, we show the cross-entropy on the test set as a function of the number of forward passes used in dropout sampling. When we use dropout for only a few forward passes, the performance is worse than the deterministic case where all neurons are activated. However, by performing multiple forward passes with dropout and averaging the results, we reduce the cross-entropy by more than one standard deviation after 30 samples. Thus, we can improve our predictive performance significantly by using dropout sampling, although this adds some computational cost at testing time. Furthermore, dropout sampling enables us to estimate model uncertainty, which could be an important source of systematic error if the probability estimates are used in subsequent scientific analyses (Ross et al., 2011). Figure 4.14 shows the standard deviation of 100 different probability estimates from dropout sampling as a function of r-band magnitude. The model uncertainties are relatively small at bright magnitudes $r \lesssim 20$, but our model produces increasingly uncertain probability estimates for faint objects as it becomes difficult to distinguish between noise and faint sources.

4.5 Conclusions

We have presented a semi-supervised generative adversarial network for classifying stars, galaxies, and quasars in the SDSS photometric images. We have demonstrated that the brightness and size distributions of images generated by our generative model are in good agreement with those of the SDSS photometric images. However, unlike most work on GANs, our focus was not solely on the generation of realistic images. By using a small number of labeled images in conjunction with a large amount of unlabeled training data, we have shown that our semi-supervised GAN is able to provide a classification that is comparable to the state-of-the-art supervised methods.

Our goal in this work was not to obtain the best classification performance for the SDSS, but to explore the potential impact of semi-supervised learning in the next generation of photometric surveys, such as DES and LSST. As future surveys probe larger and larger cosmological volumes, it is not clear if we will have sufficient spectroscopic observations for supervised learning algorithms. Even with the scale of expansive spectroscopic coverage of the original SDSS and the Baryon Oscillation Spectroscopic Survey (BOSS; Dawson et al., 2012), the spectroscopic sample will be susceptible to selection bias, and a truly unbiased training set will have only a minimal number of examples. Thus, semi-supervised and unsupervised learning will become increasingly important in future surveys. To emulate this scenario, we have performed most of our analysis using only 10^3 spectroscopic labels, and found that the classification performance is competitive with supervised algorithms.

In this work, we have also demonstrated the use of various scientific tools to validate our deep generative model. In astronomy, we have powerful techniques for characterizing classifications, even in the absence of spectroscopic labels. In contrast, most of the data sets used in the deep learning community are composed of natural images and text corpuses, which lack such statistical techniques, and direct comparison between different generative models is often difficult (Theis et al., 2016). As a result, Astronomy has the potential to provide robust frameworks for evaluation and interpretation of generative models.

In this chapter, we used photometry and spectra from the SDSS. While the SDSS provides a rich data set for deep learning, it is limited to the optical and near-infrared wavelengths. We plan to combine multiple photometry sources by matching the SDSS objects to photometric objects in other surveys, such as GALEX, WISE, or UKIDSS. We are also exploring different strategies to improve the quality of generated images. For example, although we used feature matching in this work to obtain a strong classifier, if the goal is to improve the quality of generated images, an alternative technique called minibatch discrimination will likely work better (Salimans et al., 2016; Dai et al., 2017). Finally, future studies could investigate the application of deep generative models in other settings, such as unsupervised classification,

object segmentation, and redshift estimation.

4.6 Figures and Tables

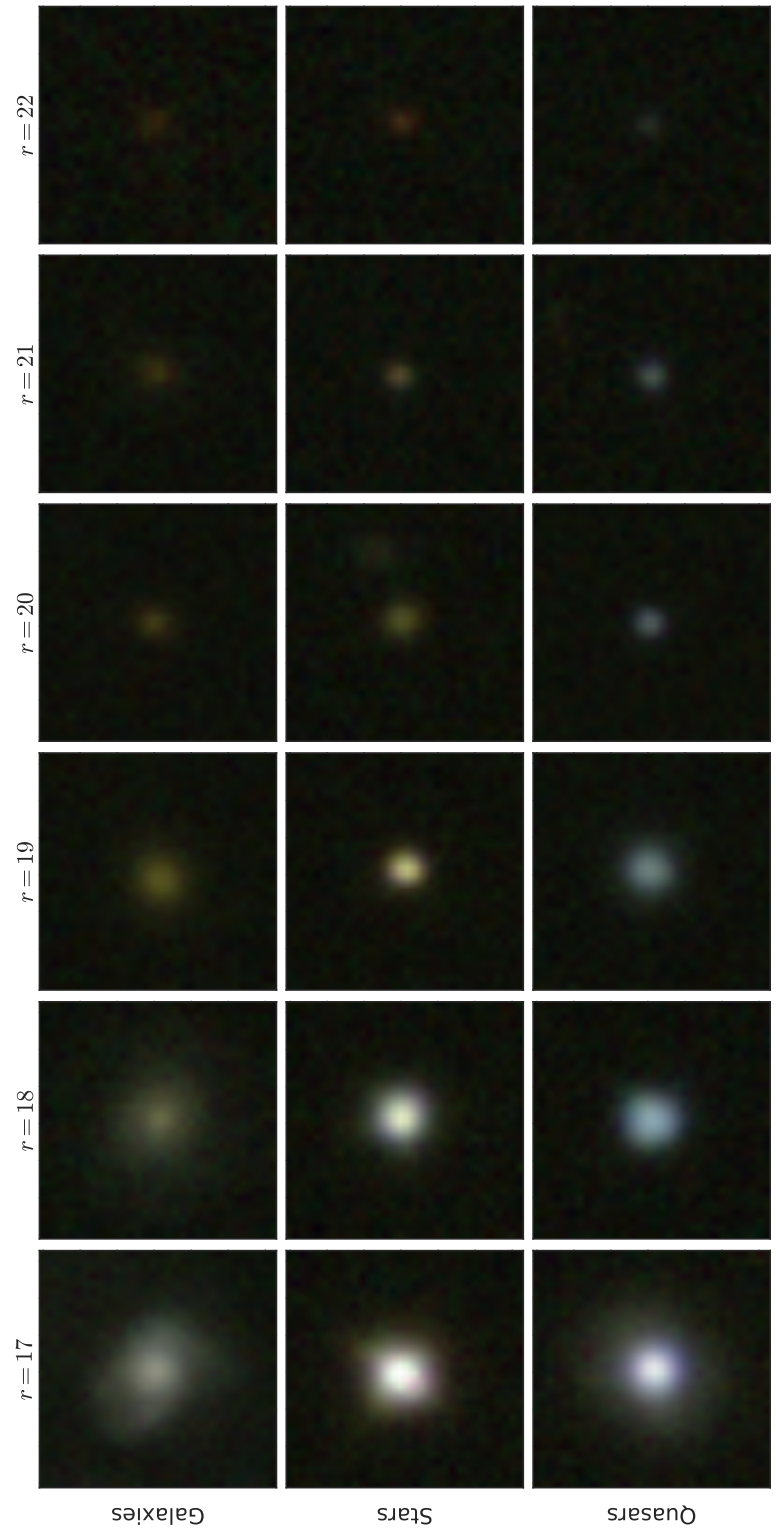


Figure 4.1: Postage stamp images showing typical stars, galaxies, and quasars in SDSS as a function of r-band magnitude. The magnitude corresponds to `SExtractor`'s `MAG_AUTO` (Kron-like elliptical aperture magnitude). Each image is 32×32 pixels and centered on the source of interest. The RGB image is created by mapping the R channel values to the i band magnitude, G channel to r band, and B to g band. Each image is individually normalized for visualization purposes.

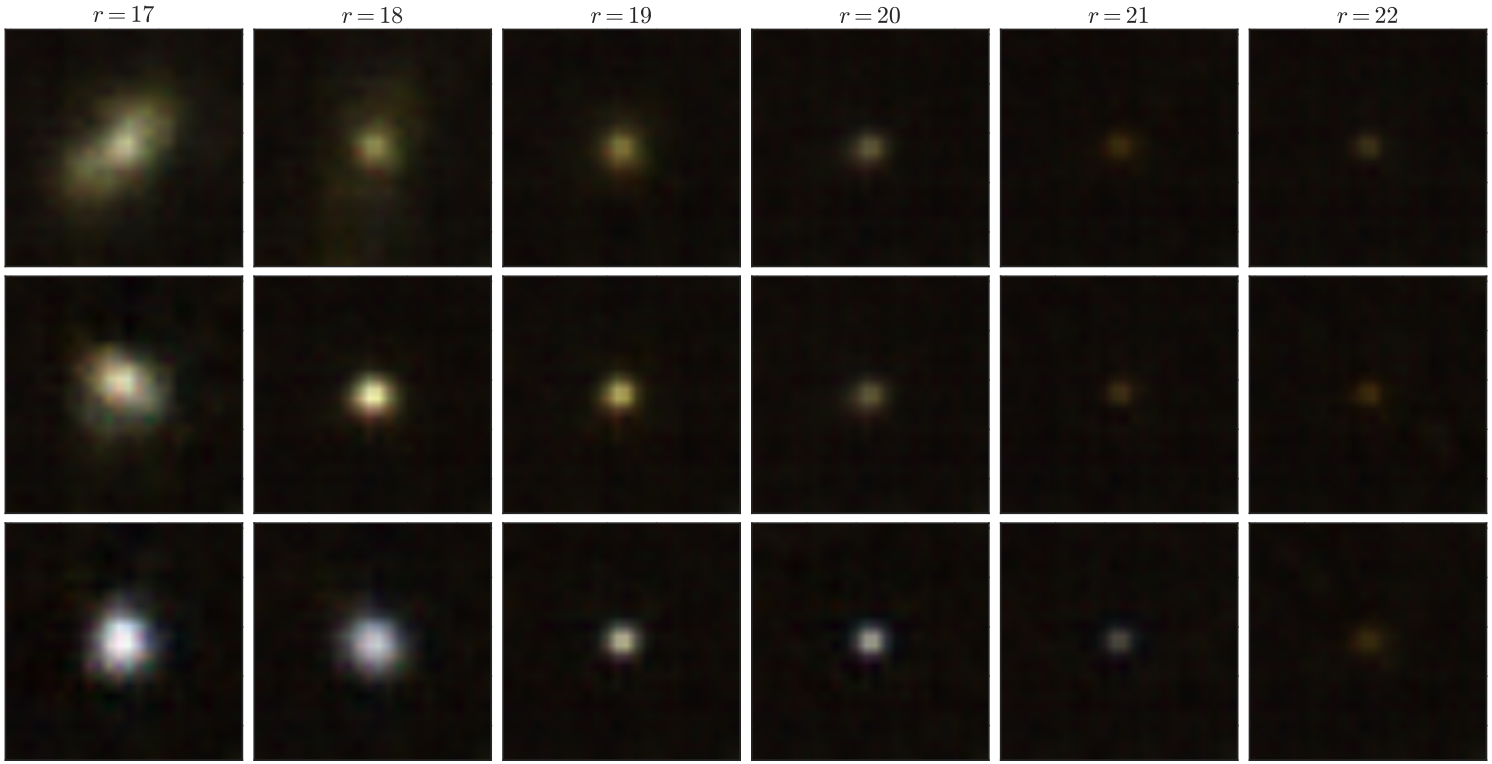


Figure 4.2: Sample 32×32 RGB images generated by our feature-matching GAN model as a function of r band magnitude. Although the generated images for bright objects are slightly blurry and lack some details, the model generated, faint objects appear indistinguishable from real, faint objects. Since most objects are faint, the magnitude and half-light radius statistics of GAN generated images are in good agreement with the SDSS distribution as shown in Section 4.4.1.

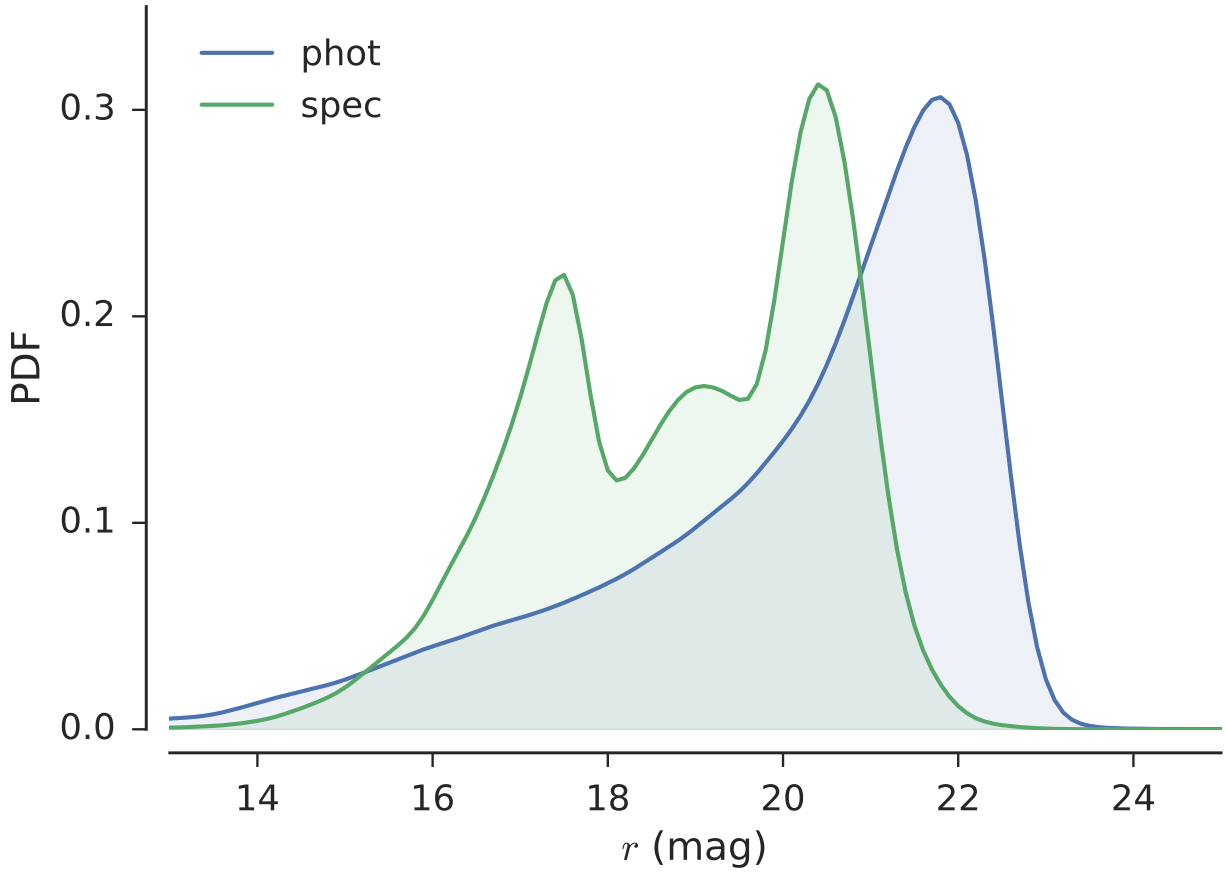


Figure 4.3: Number counts of SDSS objects as a function of the r-band magnitude as estimated by kernel density estimation (KDE). The blue curve shows the KDE of 1×10^6 objects in the unlabeled training set, which are randomly selected from the `PhotoObj` view. The green curve is the KDE of 2×10^5 objects in the labeled test set, which are randomly selected from the `SpecObj` view. We use the SDSS `cModelMag`, the composite model magnitude resulting from the best-fitting linear combination of the best-fit exponential and de Vaucouleurs fits. All KDEs presented here and throughout this chapter use a Guassian kernel and Silverman's rule to determine the kernel bandwidth (Silverman, 1986).

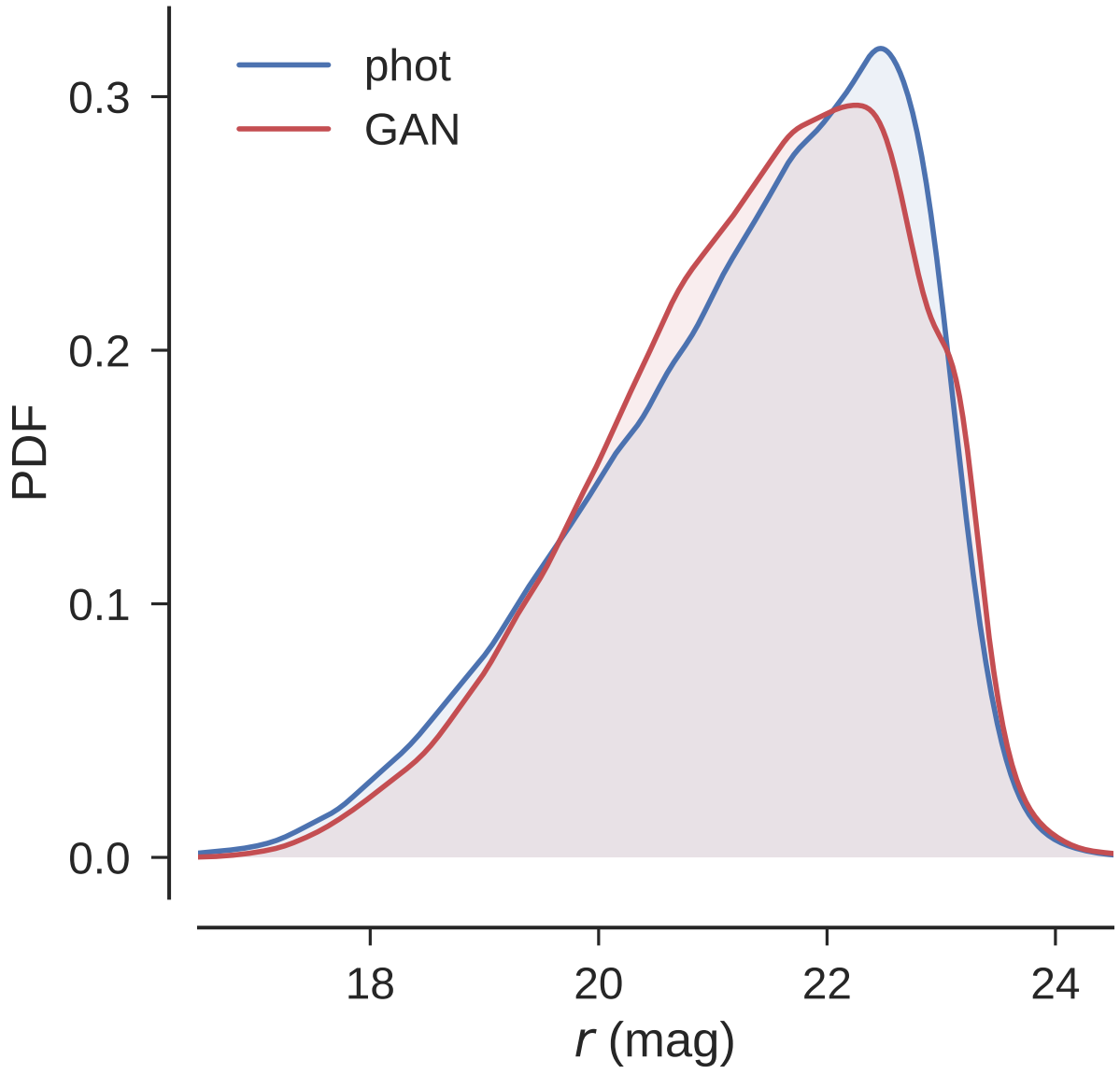


Figure 4.4: Comparison of r-band magnitude distributions between real images in the SDSS photometric sample and GAN generated images. The blue curve shows the KDE of 1×10^5 objects in the unlabeled training set. The red curve shows the KDE of 1×10^5 objects generated by GAN. We use the `SExtractor`'s `MAG_AUTO` values as an approximate total magnitude for each objects.

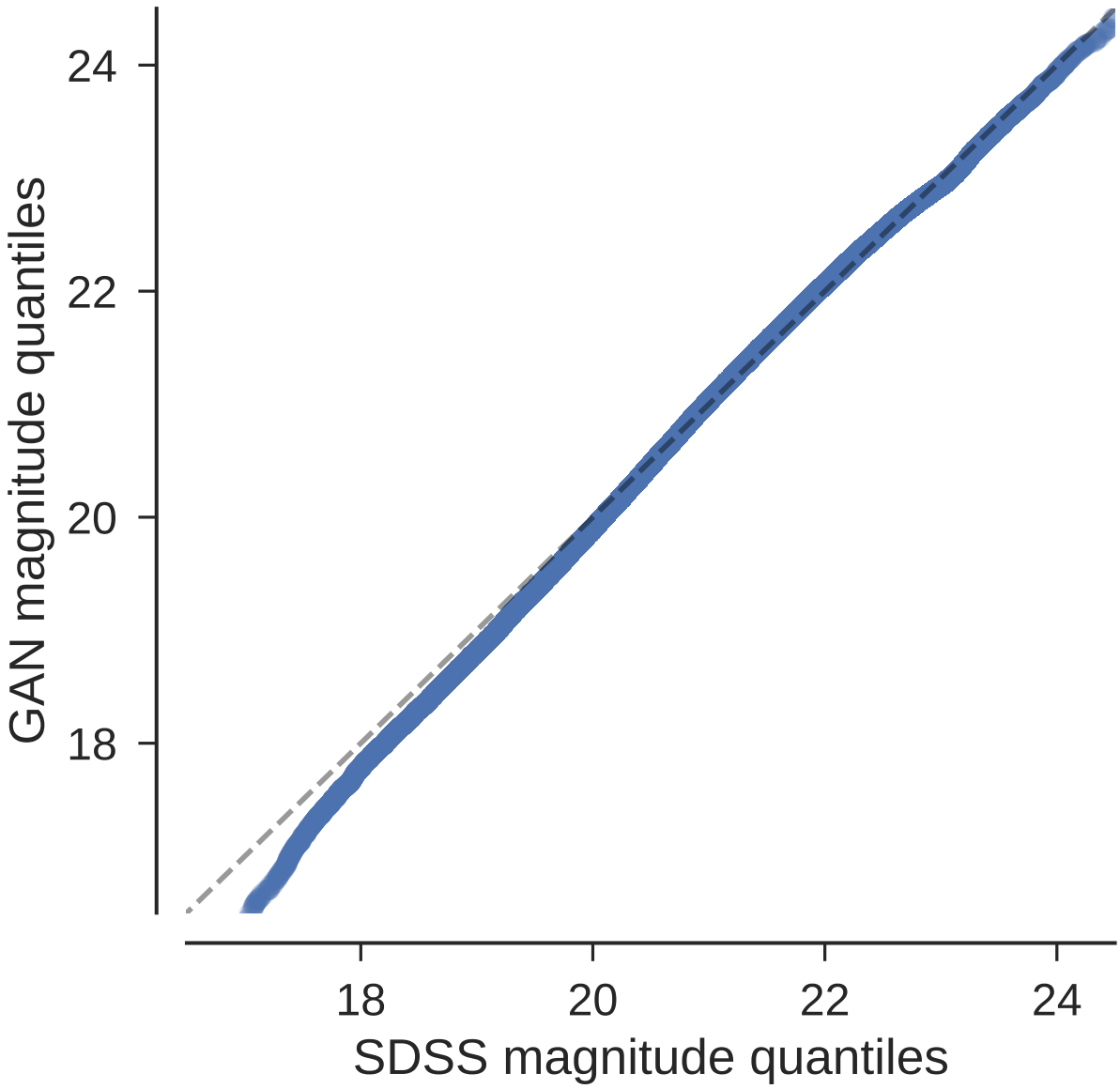


Figure 4.5: Q-Q plot comparing the distributions of r-band magnitudes between real SDSS images and GAN generated images.

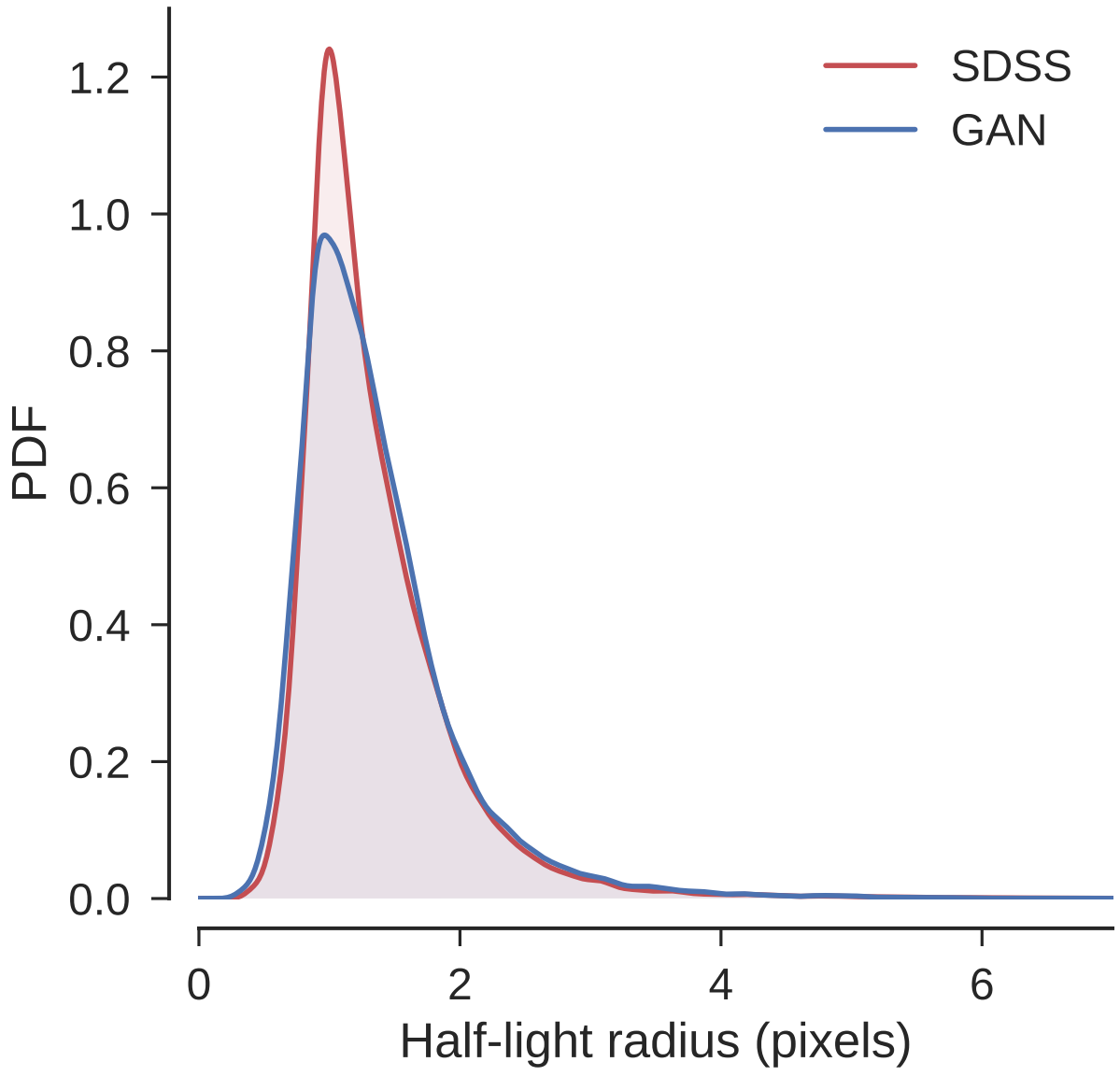


Figure 4.6: Comparison of half-light radius distributions between real images in the SDSS photometric sample and GAN generated images.

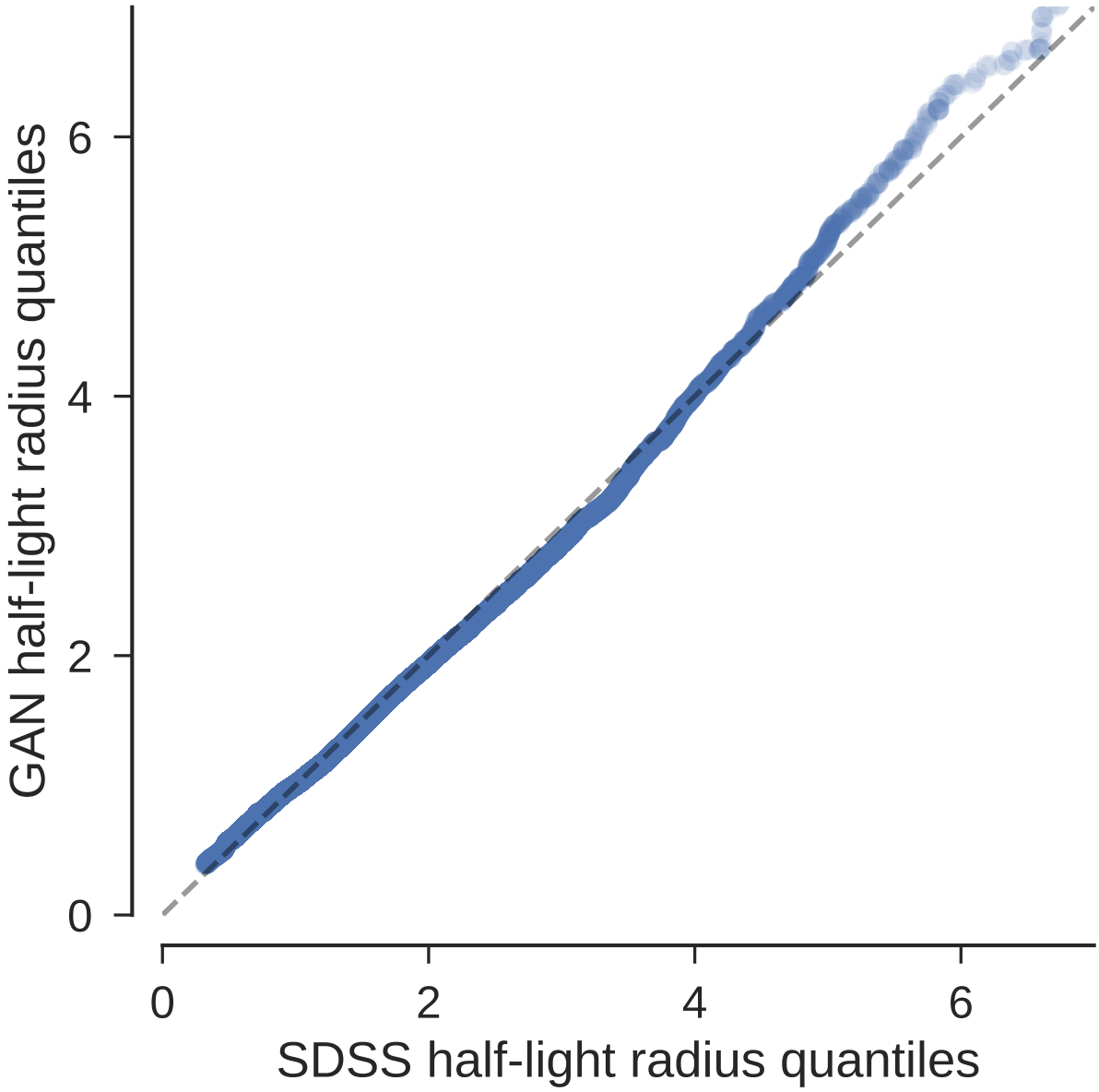


Figure 4.7: Q-Q plot comparing the distributions of half-light radius between real SDSS images and GAN generated images.

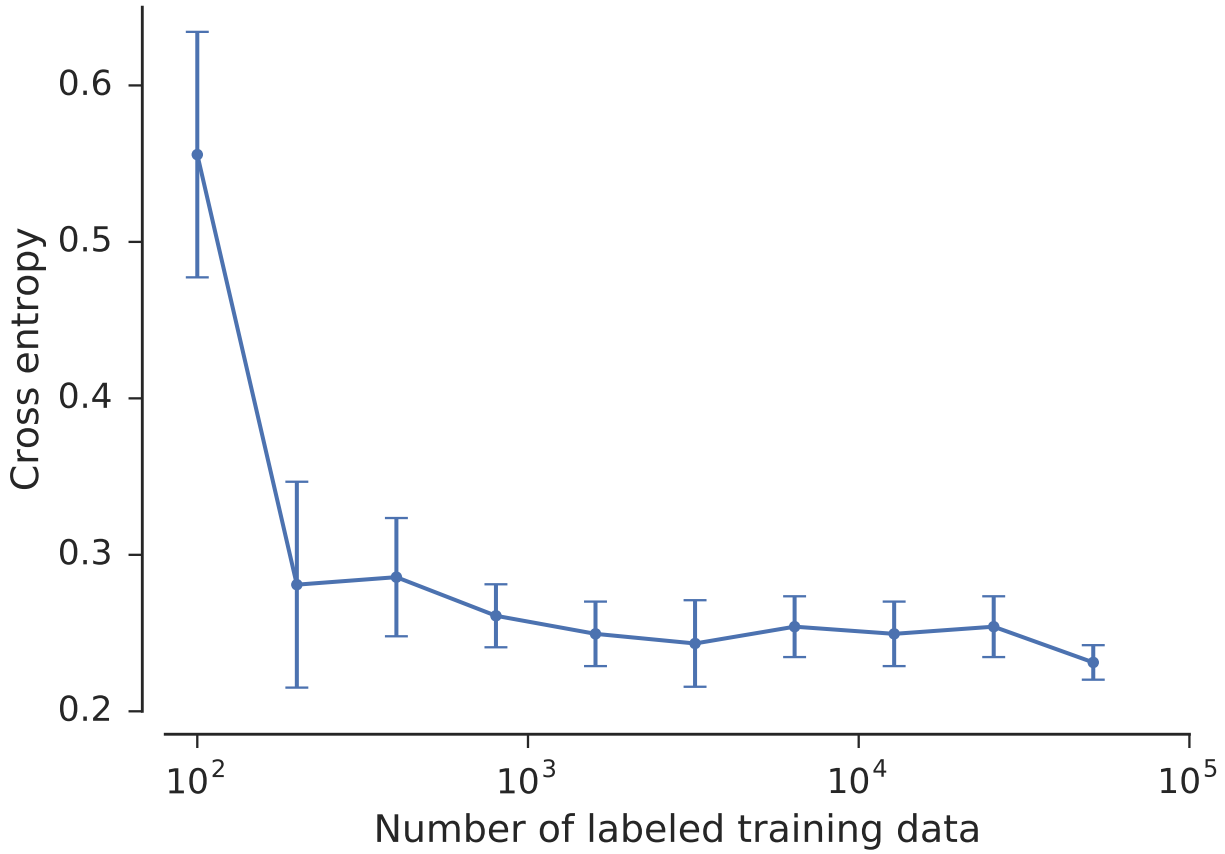


Figure 4.8: Cross-entropy as a function of the number of labeled examples in the training set.

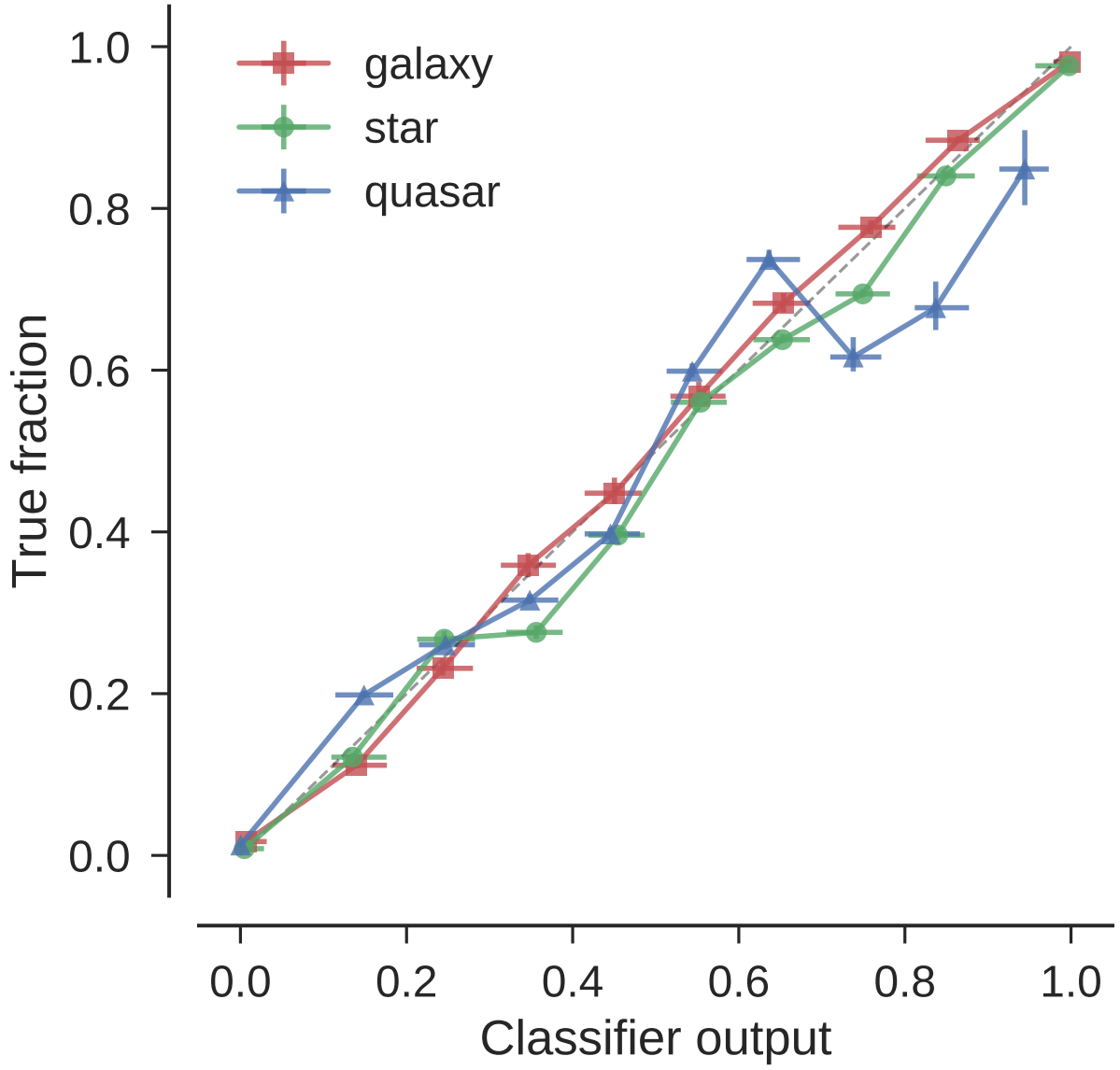


Figure 4.9: Calibrations curve for galaxies (red), stars(green), and quasars (blue). We compare the true fraction to the probabilistic output generated by the classifier for each type of objects. The dashed line displays the ideal relationship. The 1σ error bars are computed following Paterno, M (2003).

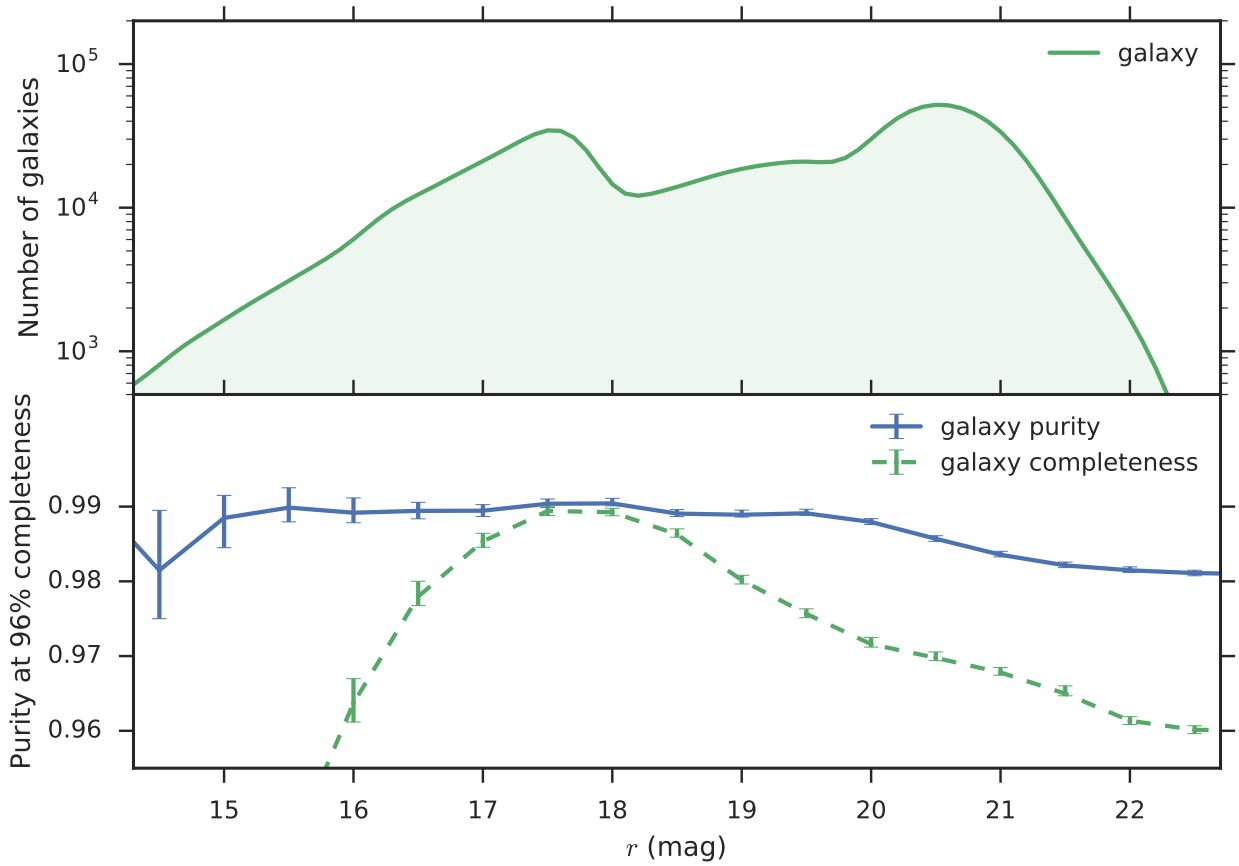


Figure 4.10: Galaxy completeness and purity values as functions of the r-band magnitude. The upper panel shows the differential counts for true galaxies in the test set. The lower panel shows the galaxy completeness and purity for the integrated counts. We use the threshold value of 0.826 to obtain overall completeness of 96%. The 1σ error bars are calculated with a Bayesian method in Paterno, M (2003).

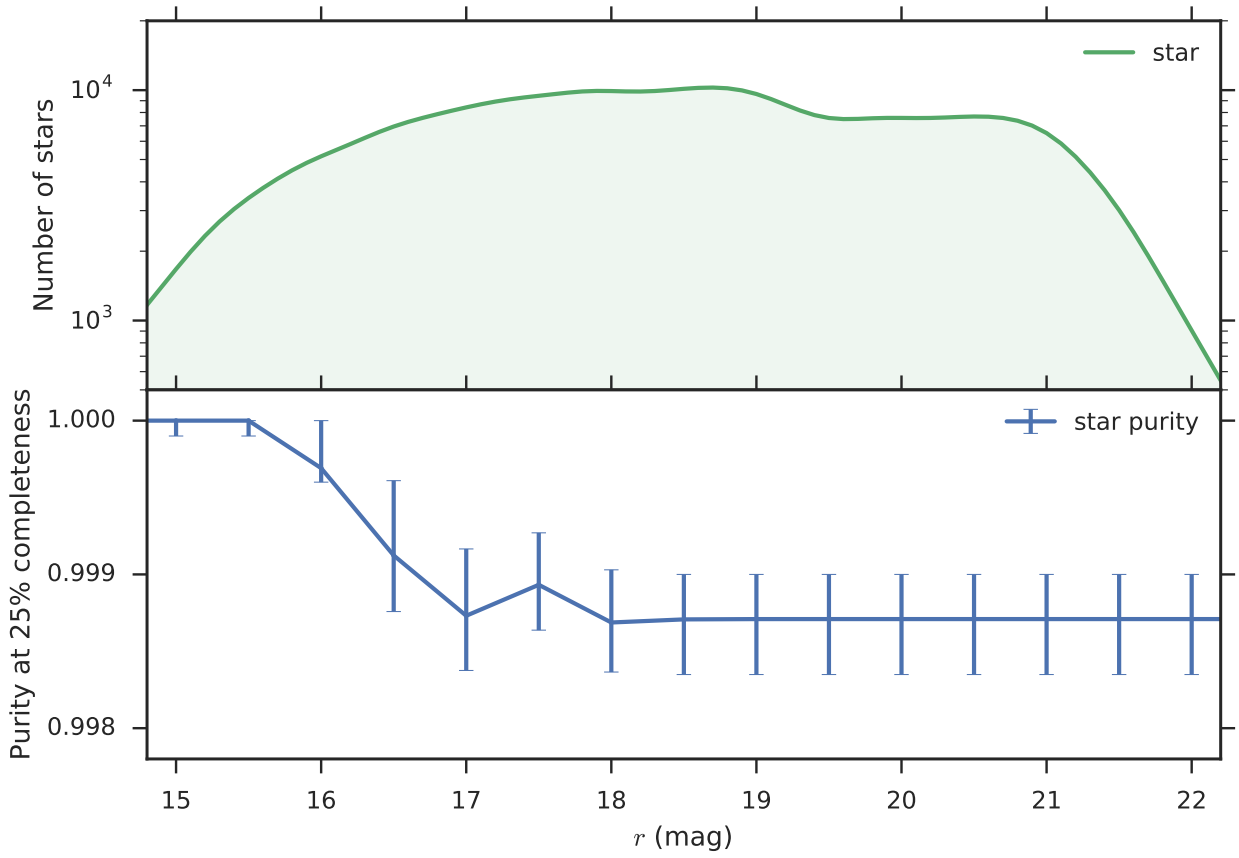


Figure 4.11: Similar to Figure 4.10 but showing completeness for stars. We use the threshold value of 0.977 to obtain overall completeness of 25%.

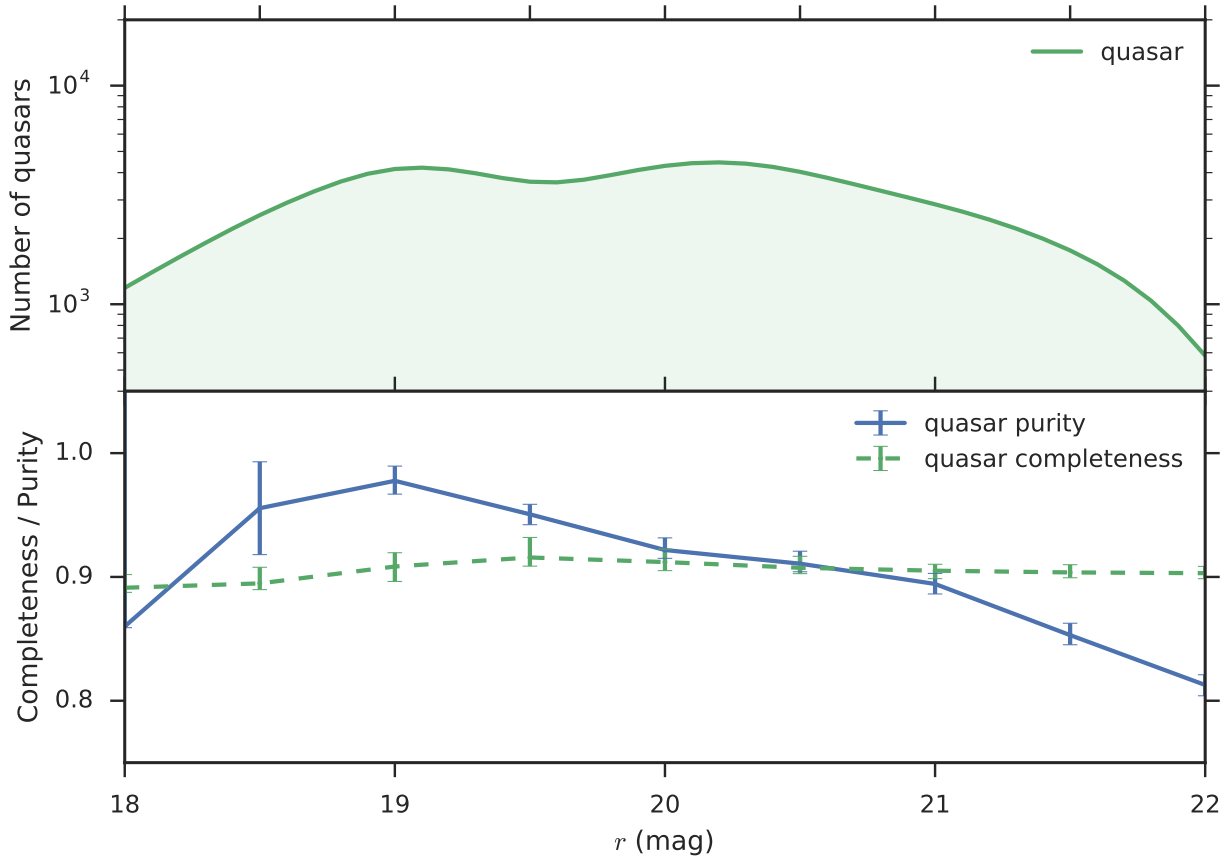


Figure 4.12: Similar to Figure 4.10 but showing completeness and purity for quasars. We use the threshold value of 0.541 to maximize $\sqrt{c_q^2 + p_q^2}$.

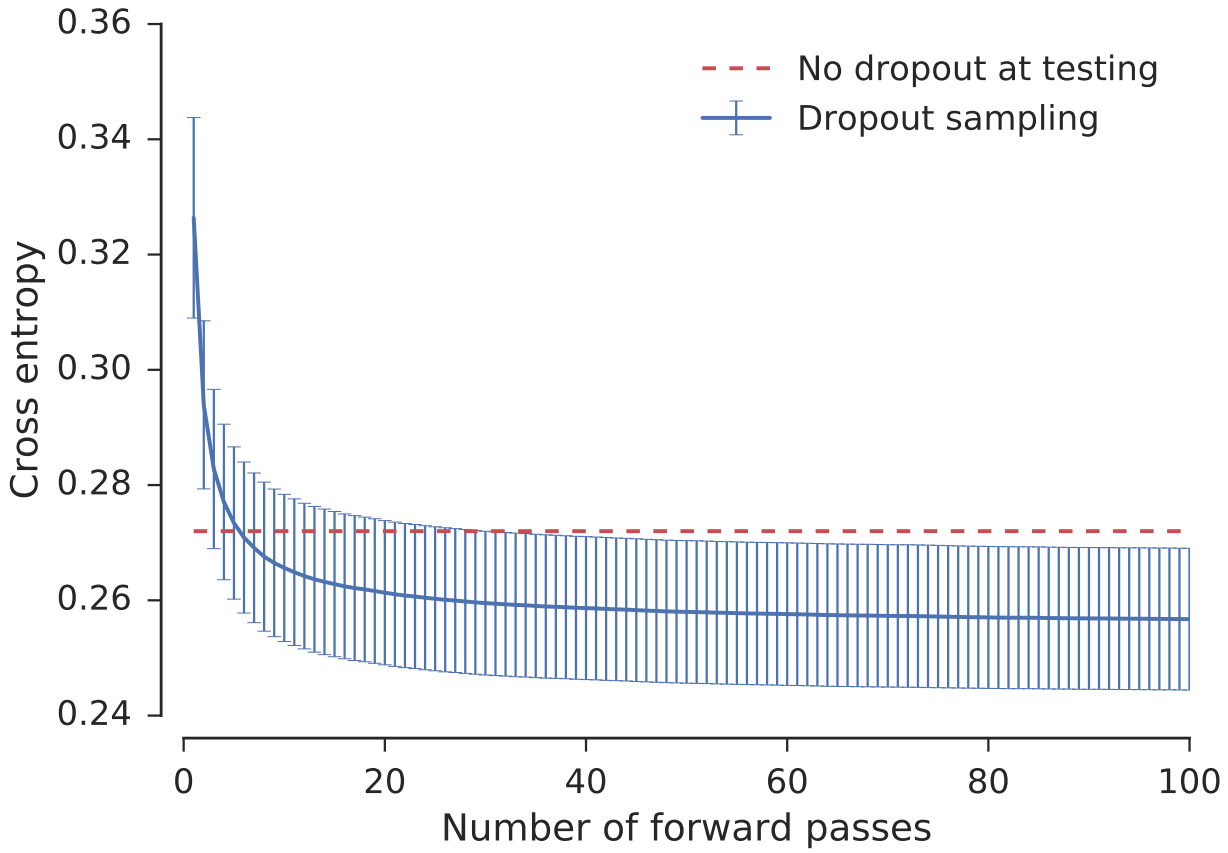


Figure 4.13: Cross-entropy for different numbers of averaged forward passes in dropout sampling. The solid blue line indicates mean cross-entropy of 10 experiments, and the error bars are 1 standard deviation. The red dotted line shows the cross-entropy with no dropout at testing time.

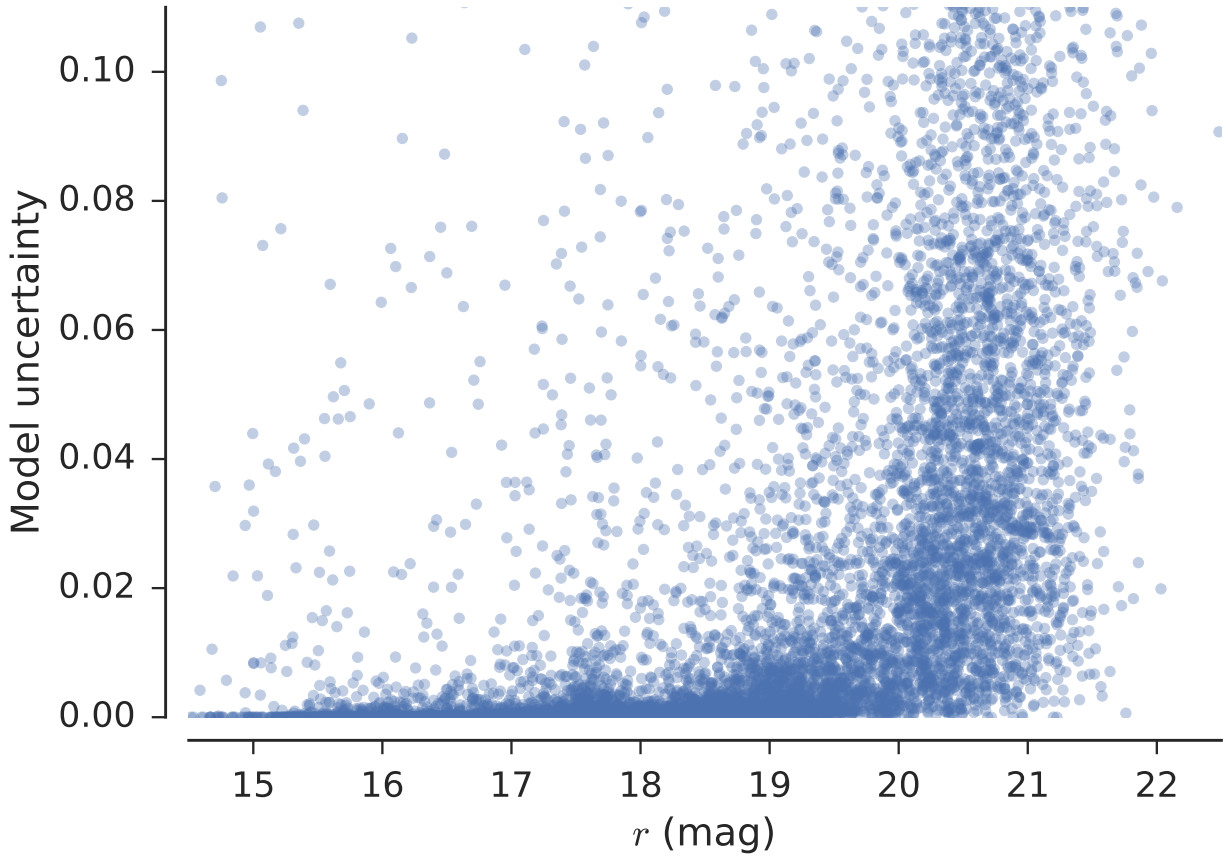


Figure 4.14: Model uncertainty as a function of r-band magnitude for 10,000 randomly selected objects in the test set. Model uncertainty is measured by the standard deviation of 100 forward passes in dropout sampling. The red dotted line indicates the magnitude at which the number count of the SDSS spectroscopic sample peaks.

4.7 References

S. Alam et al. The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. *ApJS*, 219:12, July 2015.

N. M. Ball, R. J. Brunner, A. D. Myers, and D. Tcheng. Robust Machine Learning Applied to Astronomical Data Sets. I. Star-Galaxy Classification of the Sloan Digital Sky Survey DR3 Using Decision Trees. *ApJ*, 650:497–509, October 2006.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*, 2017.

Kyle S Dawson, David J Schlegel, Christopher P Ahn, Scott F Anderson, Éric Aubourg, Stephen Bailey, Robert H Barkhouser, Julian E Bautista, Alessandra Beifiori, Andreas A Berlind, et al. The baryon oscillation spectroscopic survey of sdss-iii. *The Astronomical Journal*, 145(1):10, 2012.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *The statistician*, pages 12–22, 1983.

Daniel J Eisenstein, David H Weinberg, Eric Agol, Hiroaki Aihara, Carlos Allende Prieto, Scott F Anderson, James A Arns, Éric Aubourg, Stephen Bailey, Eduardo Balbinot, et al. Sdss-iii: Massive spectroscopic surveys of the distant universe, the milky way, and extra-solar planetary systems. *AJ*, 142(3):72, 2011.

R. Fadely, D. W. Hogg, and B. Willman. Star-Galaxy Classification in Multi-band Optical Imaging. *ApJ*, 760:15, November 2012.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Geoffrey E Hinton et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

E. J. Kim and R. J. Brunner. Star–galaxy classification using deep convolutional neural networks. *MNRAS*, 464(4):4463–4475, 2017.

E. J. Kim and R. J. Brunner. Star–galaxy classification using semi-supervised generative adversarial neural networks. *MNRAS*, Submitted.

Nolan Li and Ani R Thakar. Casjobs and mydb: A batch query workbench. *Computing in Science & Engineering*, 10(1):18–29, 2008.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *Proceddings of the International Conference on Learning Representations*, 2013.

R. H. Lupton, J. E. Gunn, and A. S. Szalay. A Modified Magnitude System that Produces Well-Behaved Magnitudes, Colors, and Errors Even for Low Signal-to-Noise Ratio Measurements. *AJ*, 118:1406–1410, September 1999.

AA Miller, MK Kulkarni, Y Cao, RR Laher, FJ Masci, and JA Surace. Preparing for advanced ligo: A star–galaxy separation catalog for the palomar transient factory. *AJ*, 153(2):73, 2017.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Rami Al-Rfou, and Zarija Lukić. Creating virtual universes using generative adversarial networks. *arXiv preprint arXiv:1706.02390*, 2017.

Augustus Odena. Semi-supervised learning with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, 2016.

S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys, and W. A. Zumach. Automated star/galaxy discrimination with neural networks. *AJ*, 103:318–331, January 1992.

Paterno, M. Calculating efficiencies and their uncertainties. `http://home.fnal.gov/~paterno/images/effic.pdf`, May 2003.

Siamak Ravanbakhsh, Francois Lanusse, Rachel Mandelbaum, Jeff Schneider, and Barnabas Poczos. Enabling dark energy science with deep generative models of galaxy images. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

AJ Ross et al. Ameliorating systematic uncertainties in the angular clustering of galaxies: a study using the sdss-iii. *MNRAS*, 417(2):1350–1373, 2011.

BTP Rowe, Mike Jarvis, Rachel Mandelbaum, Gary M Bernstein, James Bosch, Melanie Simet, Joshua E Meyers, Tomasz Kacprzak, Reiko Nakajima, Joe Zuntz, et al. Galsim: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150, 2015.

Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016.

K. Schawinski, C. Zhang, H. Zhang, L. Fowler, and G. K. Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *MNRAS Letters*, 467(1):L110–L114, 2017.

D. J. Schlegel, D. P. Finkbeiner, and M. Davis. Maps of Dust Infrared Emission for Use in Estimation of Reddening and Cosmic Microwave Background Radiation Foregrounds. *ApJ*, 500:525–553, June 1998.

Bernard W Silverman. Density estimation for statistics and data analysis. *CRC press*, 26, 1986.

M. T. Soumagnac et al. Star/galaxy separation at faint magnitudes: Application to a simulated dark energy survey. *MNRAS*, 450:666–680, jun 2015.

J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2016.

L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

Martin B Wilk and Ram Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

Donald G York et al. The sloan digital sky survey: Technical summary. *AJ*, 120(3):1579, 2000.

Chapter 5

Conclusions

5.1 Summary and Conclusions

In Chapter 2, we have presented and analyzed a novel star-galaxy classification framework for combining star-galaxy classifiers using the CFHTLenS data. In particular, we use four independent classification techniques: a morphological separation method; TPC, a supervised machine learning technique based on prediction trees and a random forest; SOMc, an unsupervised machine learning approach based on self-organizing maps and a random atlas; and HB, a Hierarchical Bayesian template-fitting method that we have modified and parallelized. Using data from the CFHTLenS survey, we have considered different scenarios: when an excellent training set is available with spectroscopic labels from DEEP2, SDSS, VIPERS, and VVDS, and when the demographics of sources in a low-quality training set do not match the demographics of objects in the test data set. We demonstrate that the Bayesian Model Combination (BMC) technique improves the overall performance over any individual classification method in both cases.

The problem of star-galaxy classification is a rich area for future research. It is unclear if sufficient training data will be available in future ground-based surveys. Furthermore, in large sky surveys such as DES and LSST, photometric quality is not uniform across the sky, and a purely morphological classifier alone will not be sufficient, especially at faint magnitudes. Given the efficacy of our approach, classifier combination strategies are likely the optimal approach for currently ongoing and forthcoming photometric surveys. Future studies could apply the combination technique described in Chapter 2 to other surveys such

as the DES. Our approach can also be extended more broadly to classify objects that are neither stars nor galaxies (e.g., quasars). Finally, future studies could explore the use of multi-epoch data, which would be particularly useful for the next generation of synoptic surveys.

In Chapter 3, we have presented a convolutional neural network for classifying stars and galaxies in the SDSS and CFHTLenS photometric images. For the CFHTLenS data set, the network is able to provide a classification that is as accurate as a random forest algorithm (TPC), while the probability estimates of our ConvNet model appear to be better calibrated. When the same network architecture is applied to the SDSS data set, the network fails to outperform TPC, but the probabilities are still slightly better calibrated. The major advantage of ConvNets is that useful features are learned automatically from images, while traditional machine learning algorithms require feature engineering as a separate process to produce accurate classifications.

Deep learning is a rapidly developing field, and recent developments include improved network architectures. Future work could explore more ConvNet variants, such as the Inception Module (Szegedy et al., 2015) and Residual Network (He et al., 2015). To improve the predictive performance, we have combined the predictions of different models across multiple transformations of the input images (Section 3.4.3). To further improve the performance, we could also train several networks with different architectures and combine the models. For example, the winning solution of Dieleman et al. (2015) for the Galaxy Zoo challenge was based on a ConvNet model, and it required averaging many sets of predictions from models with different neural network architectures. It is also likely that the performance will be improved not only by training multiple network architectures, but also by combining them with different star-galaxy classifiers. ConvNets could be included as a different machine learning paradigm in the classifier combination framework to produce further improvements in predictive performance.

Our ConvNet model is a supervised algorithm, and one of the criticisms of supervised

techniques is their difficulty in extrapolating past the limits of available spectroscopic training data. Since it is difficult to assess the classification performance without a deeper spectroscopic sample, we evaluated the performance using a test set that is drawn from the same underlying distribution as the spectroscopic sample. However, when our ConvNet model—trained on sources from a spectroscopic sample— is applied to a photometric sample—which is often fainter than our training set—the performance of ConvNet will be less reliable. Combining our ConvNet model with unsupervised methods (e.g., a template fitting method) in the aforementioned meta-classification framework will help improve the efficacy of star-galaxy classification beyond the limits of spectroscopic training data.

In Chapter 4, we have presented a semi-supervised generative adversarial network for classifying stars, galaxies, and quasars in the SDSS photometric images. We have demonstrated that the brightness and size distributions of images generated by our generative model are in good agreement with those of the SDSS photometric images. However, unlike most work on GANs, our focus was not solely on the generation of realistic images. By using a small number of labeled images in conjunction with a large amount of unlabeled training data, we have shown that our semi-supervised GAN is able to provide a classification that is comparable to the state-of-the-art supervised methods. we have also demonstrated the use of various scientific tools to validate our deep generative model. In astronomy, we have powerful techniques for characterizing classifications, even in the absence of spectroscopic labels. In contrast, most of the data sets used in the deep learning community are composed of natural images and text corpuses, which lack such statistical techniques, and direct comparison between different generative models is often difficult (Theis et al., 2016). As a result, Astronomy has the potential to provide robust frameworks for evaluation and interpretation of generative models.

We used photometry and spectra from the SDSS. While the SDSS provides a rich data set for deep learning, it is limited to the optical and near-infrared wavelengths. Future studies could explore combining multiple photometry sources by matching the SDSS objects to

photometric objects in other surveys, such as GALEX, WISE, or UKIDSS. Future studies could also explore different strategies to improve the quality of generated images. For example, although we used feature matching in this work to obtain a strong classifier, if the goal is to improve the quality of generated images, an alternative technique called minibatch discrimination will likely work better (Salimans et al., 2016; Dai et al., 2017). Finally, future studies could investigate the application of deep generative models in other settings, such as unsupervised classification, object segmentation, and redshift estimation.

5.2 References

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*, 2017.

Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *MNRAS*, 450(2):1441–1459, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.