DeepShadows: Separating Low Surface Brightness Galaxies from Artifacts using Deep Learning

Dimitrios Tanoglidis[a,b,*], Aleksandra Ćiprijanović[c], Alex Drlica-Wagner[c,b,a]

[a] *Department of Astronomy and Astrophysics, University of Chicago, Chicago, IL 60637, USA*
[b] *Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, USA*
[c] *Fermi National Accelerator Laboratory, P. O. Box 500, Batavia, IL 60510, USA*

Abstract

Searches for low-surface-brightness galaxies (LSBGs) in galaxy surveys are plagued by the presence of a large number of artifacts (e.g., objects blended in the diffuse light from stars and galaxies, Galactic cirrus, star-forming regions in the arms of spiral galaxies, etc.) that have to be rejected through time consuming visual inspection. In future surveys, which are expected to collect hundreds of petabytes of data and detect billions of objects, such an approach will not be feasible. We investigate the use of convolutional neural networks (CNNs) for the problem of separating LSBGs from artifacts in survey images. We take advantage of the fact that we have available a large number of labeled LSBGs and artifacts from the Dark Energy Survey, that we use to train, validate, and test a CNN model. That model, which we call *DeepShadows*, achieves a test accuracy of 92.0%, a significant improvement relative to feature-based machine learning models. We also study the ability to use transfer learning to adapt this model to classify objects from the deeper Hyper-Suprime-Cam survey, and we show that after the model is retrained on a very small sample from the new survey, it can reach an accuracy of 87.6%. These results demonstrate that CNNs offer a very promising path in the quest to study the low-surface-brightness universe.

Keywords: Low Surface Brightness Galaxies, Galaxy Surveys, Deep Learning, Convolutional Neural Networks

1. Introduction

Our understanding of galaxy formation, evolution, and the relationship between galaxies and the dark matter halos that they inhabit (e.g., the "galaxy–halo connection"; Wechsler and Tinker, 2018) is constrained by our ability to detect faint galaxies (e.g., Kaviraj, 2020). Low-surface-brightness galaxies (LSBGs) are conventionally defined as galaxies with a central surface brightness fainter than the night sky ($\mu(g) \gtrsim 22$ mag/arcsec^2). Thus, by definition, they are very difficult to detect and characterize. While contributing only a small fraction to the observed luminosity of the local universe, theoretical (e.g., Martin et al., 2019) and observational (e.g., Dalcanton et al., 1997) arguments suggest that LSBGs account for the majority of galaxies, which thus remains relatively unexplored.

Most of the searches for LSBGs to date have targeted small regions of the sky and have revealed LSBG populations in massive galaxy clusters such as Virgo (e.g., Sabatini et al., 2005; Mihos et al., 2015, 2017), Coma (e.g., Adami et al., 2006; van Dokkum et al., 2015) and Fornax (e.g., Hilker et al., 1999; Muñoz et al., 2015; Venhola et al., 2017), as well as faint satellites around the nearby galaxies (e.g., McConnachie, 2012; Martin et al., 2013; Merritt et al., 2016; Danieli et al., 2017; Cohen et al., 2018). To better understand and test galaxy formation models in the low-surface-brightness regime, it is imperative to study LSBGs over a wide sky area and across different environments (inside galaxy clusters vs. field). Wide-field galaxy surveys have already started to reveal a large number of LSBGs. For example, the Hyper Suprime-Cam Sub-

*Corresponding authoread
Email address: dtanoglidis@uchicago.edu (Dimitrios Tanoglidis)

aru Strategic Program (HSC SSP)[1] discovered 781 radially extended (half-light radius $r_{1/2} > 2.5''$) LSBGs with $\bar{\mu}_{\rm eff}(g) > 24.3$ mag/arcsec^2 in an analysis of the first $\sim$ 200 deg^2 of its Wide layer (Greco et al., 2018). More recently, an analysis of the first three years of data from the Dark Energy Survey (DES)[2], covering $\sim$ 5,000 deg^2 on the southern sky, brought to light a population of >20,000 LSBGs with similar size and surface brightness limits (Tanoglidis et al., 2021).

Searches for LSBGs in survey data are plagued by the presence of a large number of low-surface-brightness artifacts in astronomical images. Note that we define the artifact class to consist of any object that passes the selection criteria outlined above but is not an LSBG. This includes imaging artifacts as well as:

- Faint, compact objects blended in the diffuse light from nearby bright stars or giant elliptical galaxies;
- Bright regions of Galactic cirrus;
- Knots and star-forming regions in the arms of large spiral galaxies;
- Tidal ejecta connected to high-surface-brightness host galaxies.

Such objects often dominate the sample of candidate LSBGs. For example, in DES there were 413,000 LSBG candidates, with only $\sim$5% of them being genuine LSBGs. Even after a feature-based machine learning (ML) classification step that reduced the sample by approximately an order of magnitude, a large number of false-positives remained ($\sim$50% of objects classified as LSBGs). These false-positives had to be manually rejected through visual inspection. Similarly, the authors of the HSC SSP study had to go through a visual inspection step, since their pipeline produced a sample that also had a $\sim$50% contamination rate from artifacts (Greco et al., 2018).

Visual inspection is time consuming and difficult to perform systematically. Upcoming galaxy surveys, such as the Legacy Survey of Space and Time (LSST)[3] on the Vera C. Rubin Observatory[4] and Euclid[5] are expected to produce massive volumes of data. LSST will observe $\sim$ 20,000 deg^2 of sky, produce 20TB of data per night, and observe $\sim$10 billion galaxies over its 10 years of observations.[6] With such volumes of data, rejecting artifacts via visual inspection will be impossible. Clearly, the process has to be automated.

Machine learning, and in more recent years deep learning, have started to revolutionize astronomy as the sizes of astronomical datasets grow (for reviews see e.g., Ball and Brunner, 2010; Baron, 2019). Classification tasks are one of the classical examples where machine learning techniques can be applied. In cases dealing with high-dimensional feature spaces where large training sets are available, deep learning usually outperforms other machine learning algorithms and reaches human-level performance (LeCun et al., 2015).

Convolutional neural networks (CNNs; LeCun et al., 1998) constitute a specific class of deep learning algorithms, inspired by the visual cortex and optimized for computer vision tasks. For that reason they are a promising tool for analyzing astronomical images. Furthermore, working directly at the image level (with the pixels as inputs) eliminates the need for deriving and selecting parameters (sizes, magnitudes, colors etc.) as features to be used for the classification task, which can be subjective and non-optimal.

CNNs were first introduced in astronomy by Dieleman et al. (2015) to perform automatic morphological classification of galaxies and have since found a number of applications. For example, other authors further explored their use in classifying galaxy morphologies (e.g., Dai and Tong, 2018; Domínguez Sánchez et al., 2018; Cheng et al., 2020), separating stars from galaxies (e.g., Kim and Brunner, 2017), identifying strong lenses (e.g., Lanusse et al., 2018; Jacobs et al., 2019; Davies et al., 2019; Bom et al., 2019), eliminating polarimetric artifacts (Paranjpye et al., 2020), evaluating flare statistics in young stars (Feinstein et al., 2020), classifying galaxy mergers (Ćiprijanović et al., 2020b), reconstructing lensing of the Cosmic Microwave Background (Caldeira et al., 2019), setting constraints on the cosmological parameters from weak lensing (e.g., Ribli et al., 2019), and many other applications.

[1] https://hsc.mtk.nao.ac.jp/ssp/
[2] https://www.darkenergysurvey.org/
[3] https://www.lsst.org/
[4] https://www.vro.org/
[5] https://www.euclid-ec.org/
[6] https://www.lsst.org/scientists/keynumbers

In this paper we present an application of CNNs in classifying LSBGs in astronomical images and we demonstrate that they can significantly help in automating this process. We take advantage of the fact that we have available a large sample of LSBGs and an equally large number of labeled artifacts from visual inspection (Tanoglidis et al., 2021). These large labeled training sets are necessary to successfully train CNN models. We compare the performance of our CNN architecture to conventional machine learning models (support vector machines and random forests) trained on features extracted from the same objects. We also study how well the model trained on the DES images can classify images from the HSC SSP, thus demonstrating promise for using a similar technique in upcoming surveys, such as LSST.

This paper is organized as follows: In Sec. 2 we describe the datasets we use for the classification problem. In Sec. 3 we briefly summarize the theory and formalism of neural networks and present the specific architecture we use to tackle the problem at hand, which we call *DeepShadows*. In Sec. 4 we present the classification results. In Sec. 5 we use transfer learning to classify objects from the HSC SSP for which we have labels. In Sec. 6 we study the uncertainties present in our study and their impact on the metrics used to assess the classification performance of the model. We discuss our results, propose paths for future investigation, and conclude in Sec. 7.

The code and data related to this work are publicly available at the GitHub page of this project: `https://github.com/dtanoglidis/DeepShadows`.

2. Data

In this section we describe the datasets used for training and evaluating the performance of the CNN and other machine learning models. We briefly describe the astronomical surveys and selection procedures used to obtain these data.

2.1. The Dark Energy Survey

Our primary dataset comes from the first three years (Y3) of observations by DES. DES is an optical/near infrared imaging survey that covers $\sim 5,000$ deg^2 of the southern Galactic cap in five photometric filters, $grizY$, to a depth of $i \sim 24$ over the course of a six-year observational program with the 570-megapixel Dark Energy Camera (DECam) on the 4m Blanco Telescope at the Cerro Tololo Inter-American Observatory (CTIO) in Chile. The DECam field-of-view covers 3 deg^2 with a central pixel scale of $0.263''$ (Flaugher et al., 2015).

Objects were detected in astronomical images using `SourceExtractor` (Bertin and Arnouts, 1996), which provides a catalog of photometric parameters for each object, such as magnitudes, flux radii, mean and central surface brightnesses (adaptive aperture measurements) etc. For a more detailed description of the DES data, see DES Collaboration et al. (2018).

2.2. LSBGs and artifacts in DES

We train, validate and test the performance of our models on LSBGs and artifacts detected by DES, as described in Tanoglidis et al. (2021). Here, we briefly outline the main steps followed in that paper for the LSBG catalog construction:

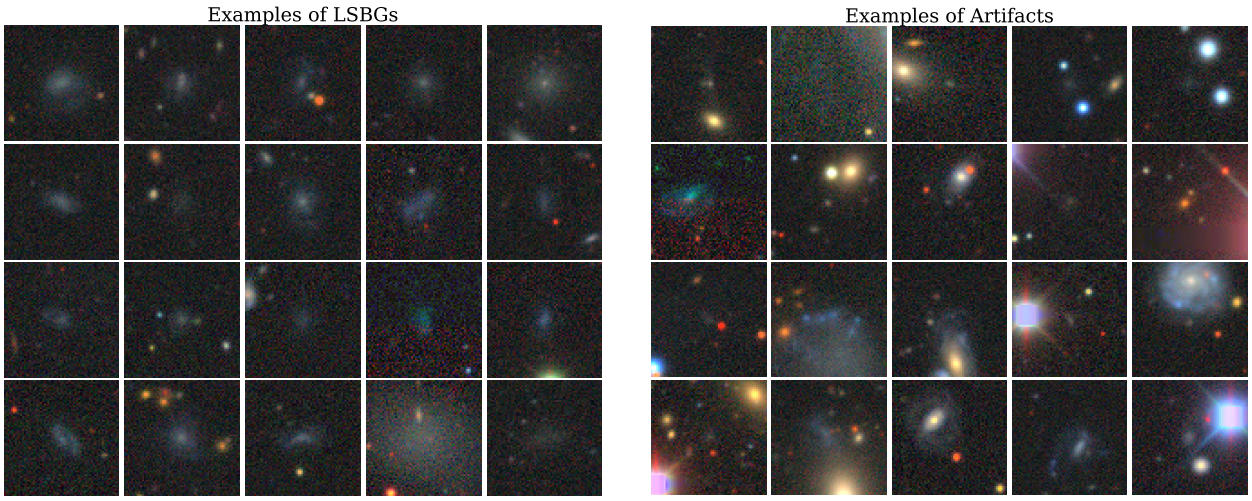
1. Selection cuts were performed using the `SourceExtractor` parameters from the full DES catalog. The most important cuts are on the angular size (half-light radius in the g-band, $r_{1/2} > 2.5''$) and on the mean surface brightness within the effective radius ($\bar{\mu}_{\rm eff}(g) > 24.3$ mag/arcsec^2).[7] The resulting candidate sample consists of ~ 0.5 million objects.

2. Classification was performed using Support Vector Machines (SVMs) trained on `SourceExtractor` output parameters (features) and a manually annotated set of $\sim 8{,}000$ objects, out of which 640 were LSBGs. This step reduces the candidate sample by roughly an order of magnitude. However, the resulting sample still includes $\sim 50\%$ non-LSBG artifacts.

3. Visual inspection was used to reject false positives from more than 40,000 objects positively classified in the previous step.

4. Sérsic model fitting and interstellar extinction correction was applied to objects passing the visual inspection, and new selection cuts were performed on the updated parameters.

[7] An updated version of Tanoglidis et al. (2021) uses a brighter selection, $\bar{\mu}_{\rm eff}(g) > 24.2$ mag/arcsec^2. However, we keep the older definition, since the LSBG/artifact separation is more challenging in the fainter regime.



(a) (b)

Figure 1: Example images of (a) LSBGs and (b) artifacts in our dataset. Each cutout corresponds to a $30'' \times 30''$ angular region on the sky. We remind the reader that our artifact class consists of any object that passes the low surface brightness selection criteria but is not an LSBG (see example categories in Sec. 1).

Note that this dataset contains detection and selection biases that are difficult to estimate due to the use of human visual inspection as described above.

For the current classification study, we randomly select 20,000 LSBGs from those visually verified in Step 3 to be used as the positive class. In the main body of our paper we use as the negative sample 20,000 objects from those visually rejected in Step 3. These are the most challenging artifacts to be separated from LSBGs, since they passed the feature-based classification step in Step 2. We consider a three-class classification problem in Appendix A, where we add another class of artifacts, 20,000 randomly selected objects from those rejected in Step 2.

2.3. Generation of datasets

For the LSBGs and artifacts we consider two datasets: parameters from `SourceExtractor` to be used as features for the classical machine learning models (SVMs and random forests) and images to be used in our *DeepShadows* CNN model.

For the classical machine learning models, we select the following features:

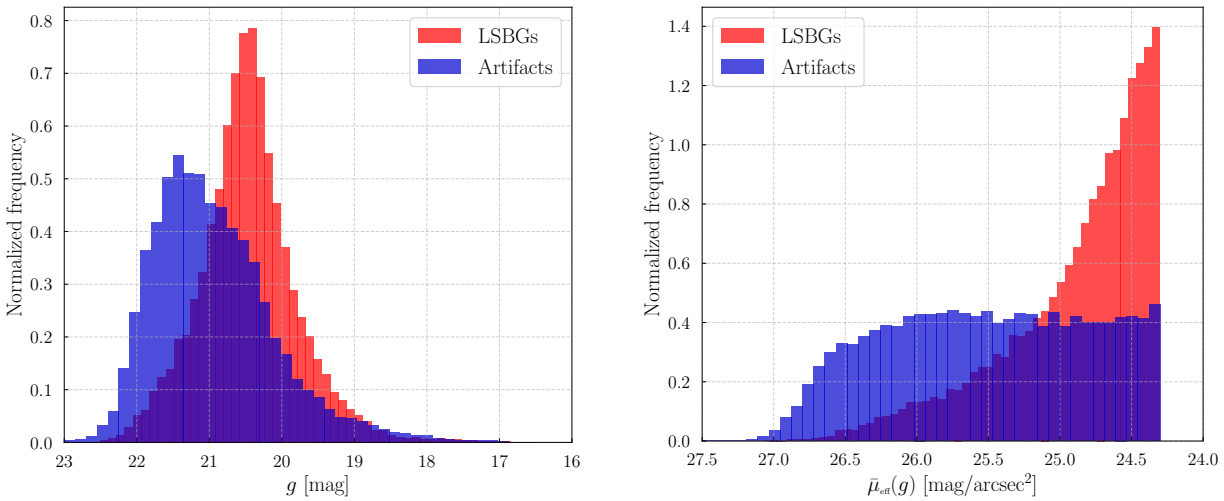
- Ellipticity of the detected objects.
- `MAG_AUTO` magnitudes in the three bands, g, r, i.
- Colors $g - i$, $g - r$, $r - i$.
- Mean, central and effective surface brightnesses in the three bands, g, r and i.

Each of these properties is derived from the `SourceExtractor` output provided by DES Data Release 1 (DR1, Abbott et al. 2018).

For the *DeepShadows* CNN model, we generate the image cutouts using the DESI Legacy Imaging Surveys Sky Viewer (Dey et al., 2019)[8] to access the DES DR1 images. Each image corresponds to a $30'' \times 30''$ region on the sky and is centered at the coordinates of the candidate object (LSBG or artifact). The initial size of each image is 256×256 pixels that we resize to 64×64 pixels to reduce the dataset size and the memory needs for its processing. The images also have inputs in the three RGB channels (which correspond to g, r, z astronomical bands), so their size is finally $64 \times 64 \times 3$ (we follow a "channels last" format). The code we used to generate the cutouts can be found in the GitHub page of the project. In Fig. 1 we show examples of cutouts of LSBGs and artifacts.

Before training, we split our full sample of 40,000 objects into a training set of 30,000 examples, a validation set of 5,000 examples and a test set of 5,000 examples. The selection of objects to be included in each set is random and each set contains an equal number of positive (LSBG) and neg-

[8] `http://legacysurvey.org/`



(a) (b)

Figure 2: The g-band (a) magnitude and (b) surface brightness distributions of the LSBGs (red) and artifacts (blue) in our sample.

ative (artifact) examples. We split the datasets of `SourceExtractor` parameters and images in the same way (same objects in each set). In Fig. 2 we present the g-band (a) magnitude and (b) surface brightness distributions for the LSBGs (red) and the artifacts (blue) in our (full) sample.

2.4. HSC SSP dataset

We also consider a dataset of 640 LSBGs and 640 artifacts discovered in the HSC SSP as described in Greco et al. (2018). This is an independent set from a survey with different specifications and different human biases (in the labeling of LSBGs/artifacts) that can be used to test the ability of our model trained on the DES images to classify LSBG candidates in other surveys. We generate image cutouts for HSC SSP, using the Sky Viewer used for the DES images. We split the full dataset of 1,280 objects into a small set of 320 objects to be used for re-training of the classifier (transfer learning) and one of 960 objects for testing the performance.

3. Methods

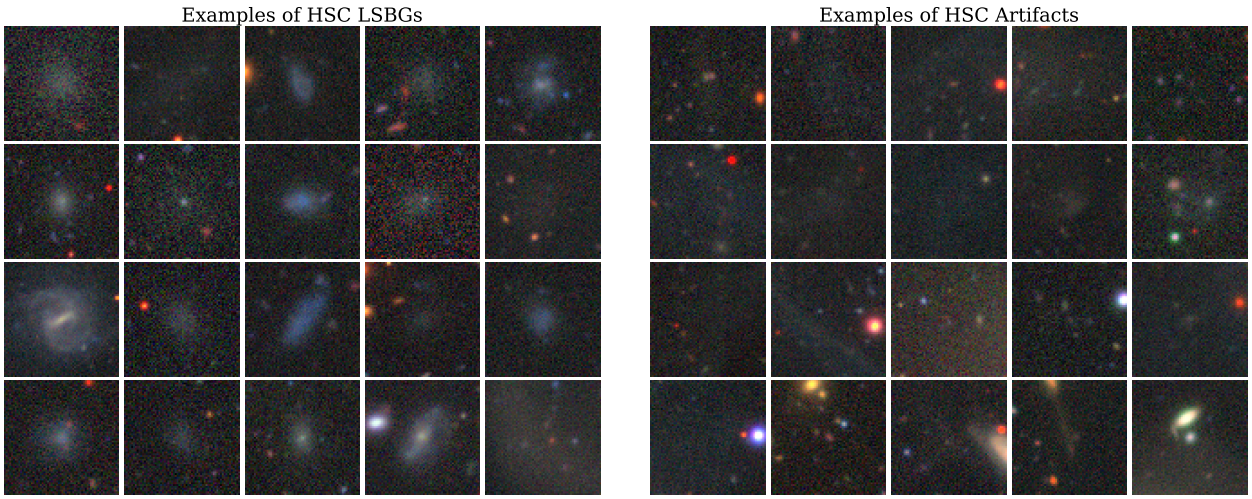
In this section we introduce the notation and briefly describe the machine learning models, the neural networks, and the specific CNN architecture that we use. Our discussion is parsimonious. For a more detailed discussion we suggest the classic book by Hastie et al. (2001) as well as that by Ivezic et al. (2014) that discusses machine learning with a focus on astrophysical applications. The book by Goodfellow et al. (2016) has become a standard reference for deep learning.

3.1. Machine Learning

We consider two machine learning classification algorithms that have been proven to be powerful in a number of astrophysical problems (see the review papers mentioned in the introduction and references therein), namely SVMs and random forests. These algorithms perform best with *structured* data (i.e., features/properties of the objects under study) and we apply them to the `SourceExtractor` output properties.

Consider a number of examples, N, each described by a feature vector $\mathbf{x}_i$, $i = 1, \ldots, N$ and with labels y^i (they usually are denoted as $\{1, -1\}$ or $\{1, 0\}$ in two-class problems).

The SVM classifier (Cortes and Vapnik, 1995) seeks to find a hyperplane that separates the two classes, of the form $\mathbf{w} \cdot \mathbf{x} - b = 0$, with the weights $\mathbf{w}$ and the bias b selected to maximize the margin (distance) between this hyperplane and the training samples that are closest to it (support vectors). In practice some misclassification is allowed (soft margin SVM), controlled by a tunable hyperparameter. Furthermore, in many cases, a non-linear transformation is performed on the data that maps them into a space where they are more easily linearly separable.



(a) (b)

Figure 3: Examples of (a) LSBGs and (b) artifacts from the HSC SSP survey. As in Fig. 1, each cutout corresponds to a $30'' \times 30''$ angular region on the sky.

Random forests is another powerful classification method. Random forests (Tin Kam Ho, 1995; Breiman, 2001) are an ensemble classifier, in the sense that use a collection of simple classifiers, known as decision trees (DTs). Specifically, it considers a set of n DTs and for each one uses a random selection of $\sqrt{m}$ features (m is the total number of features) to construct a DT and train it on a randomly selected subset of the training examples. The output of the random forest is the majority class derived from the DTs.

The values of the hyperparameters are selected (tuned) by exploring a grid of possible values, and obtaining those that give the best results either by evaluating on the validation set, or using $k-$fold cross validation, where the training set is split in k parts, with $k-1$ used for training and the other for validation, and then repeating this process k times.

3.2. Deep Learning

The standard deep learning architecture is that of a multi-layer neural network, with the outputs of the previous layer being the inputs to the following one. For example, for the n-th layer:

$$\mathbf{x}_n = g(\mathbf{W}_n \mathbf{x}_{n-1} + \mathbf{b}_n), \tag{1}$$

where $\mathbf{W}_n$ a matrix containing the weights of all the neurons of that layer and $\mathbf{b}_n$ a vector of biases. The *activation function*, g, and its purpose is to introduce non-linearities in the network. We use the rectified linear unit (ReLU), $g(x) = \max(0, x)$, activation function except in the final layer, where the activation function takes a sigmoid form (for a binary problem) to allow the output to be interpreted as a score that the example belongs to the positive class.

CNNs are modified versions of the network described in Eq. 1, with each neuron in a convolutional layer connected only to neurons within a small rectangle in the previous layer, usually 3×3 to 5×5 pixels in size. The output of such a layer is a predefined number of *feature maps*, generated by convolving the feature maps of each previous layer with different *filters* (or *kernels*), whose trainable weights can capture abstract visual features.

If we have $k = 1, \ldots, K$ input feature maps and $\ell = 1, \ldots, L$ output feature maps, in analogy to Eq. (1) we write the ℓ-th output map of the n-th layer as:

$$\mathbf{x}_n^\ell = g\left(\sum_{k=1}^{K} \mathbf{W}_n^{k,\ell} * \mathbf{x}_{n-1}^k + b_n^\ell\right), \tag{2}$$

where $*$ represents the convolution operation.

Convolutional layers are almost always followed by *pooling* layers whose purpose is to subsample the output of the convolutional layer, reducing the number of trainable parameters. They have no weights and instead keep the maximum (max pooling) or the mean (average pooling) within a small window (usually 2×2 pixels) sliding over the input

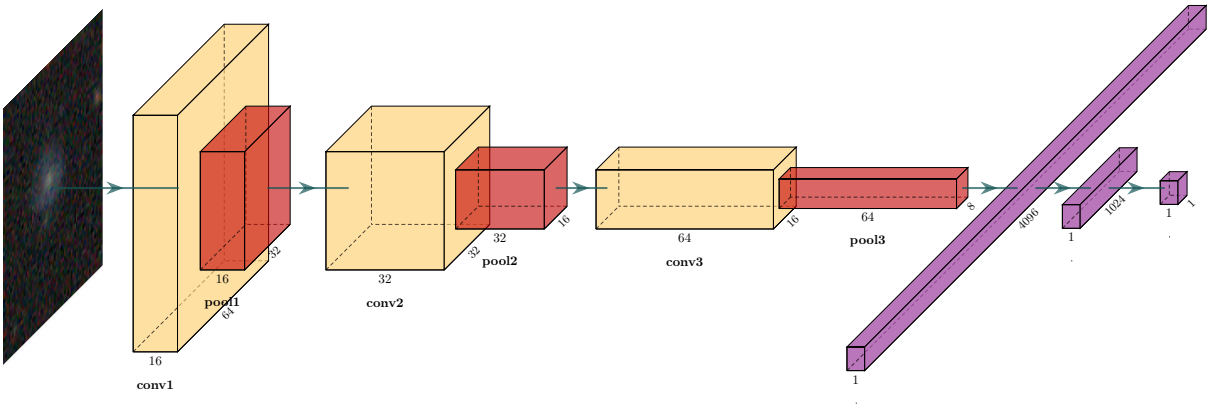


Figure 4: Schematic overview of the *DeepShadows* CNN architecture. There are three convolutional layers (yellow), each followed by a max pooling layer (red). The number of filters are 16, 32 and 64 for each layer, respectively. Two dense layers (purple) follow the last pooling layer after flattening. The output is a probability score that the image contains an LSBG. Figure was created using the PlotNeuralNet code (Iqbal, 2018).

Table 1: Architecture of the *DeepShadows* CNN.

Layers	**Properties**	**Stride**	**Padding**	**Output Shape**	**Parameters**
Input	$64 \times 64 \times 3$[a]	-	-	(64, 64, 3)	0
Convolution (2D)	Filters: 16	1×1	Same	(64, 64, 16)	448
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
	Reg: L2 (0.13)	-	-	-	-
Batch Normalization	-	-	-	(64, 64, 16)	64
MaxPooling	Kernel: 2×2	2×2	Valid	(32, 32, 16)	0
Dropout	Rate: 0.4	-	-	(32, 32, 16)	0
Convolution (2D)	Filters: 32	1×1	Same	(32, 32, 32)	4640
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
	Reg: L2 (0.13)	-	-	-	-
Batch Normalization	-	-	-	(32, 32, 32)	128
MaxPooling	Kernel: 2×2	2×2	Valid	(16, 16, 32)	0
Dropout	Rate: 0.4	-	-	(16, 16, 32)	0
Convolution (2D)	Filters: 64	1×1	Same	(16, 16, 64)	18496
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
	Reg: L2 (0.13)	-	-	-	-
Batch Normalization	-	-	-	(16, 16, 64)	256
MaxPooling	Kernel: 2×2	2×2	Valid	(8, 8, 64)	0
Dropout	Rate: 0.4	-	-	(8, 8, 64)	0
Flatten	-	-	-	(4096)	-
Fully connected	Activation: ReLU	-	-	(1024)	4195328
	Reg: L2 (0.12)	-	-	-	-
Fully connected	Activation: Sigmoid	-	-	(1)	1025

[a]We use "channel last" image data format.

feature map. Here we use max pooling.

In Fig. 4 we present a schematic overview of the CNN architecture we use for LSBG/artifact classification, that we call *DeepShadows*. It is further described in more detail in Table 1. *DeepShadows* is a simple sequential architecture, consisted of three convolutional layers (yellow) alternating with pooling layers (red); after the last pooling layer the array is flattened and followed by two fully-connected layer (purple), the last one being a single neuron that outputs the score (0 to 1) that can be interpreted as the confidence that the input image contains an LSBG. All convolutional layers use kernels of size 3×3, while the pooling layers use kernels of size 2×2. Between each convolutional and pooling layer we perform batch normalization (Ioffe and Szegedy, 2015) to make training faster and more stable.

To tackle overfitting we employ the following methods: first, we use dropout (Srivastava et al., 2014) after each pooling layer. Dropout sets a specific fraction (here we use 0.4) of randomly selected weights equal to zero. We also use L2 (also known as ridge or Tikhonov) regularization (e.g., Hastie, 2020) applied on the weights of the convolutional layers with a penalty term $\lambda = 0.13$ and on the first fully connected layer with penalty $\lambda = 0.12$. We provide more training details for the *DeepShadows* model in Sec. 4.2.

4. Classification Results

4.1. Classification Metrics

To evaluate and compare the performance of the classifiers used in this work we use a number of useful classification metrics, each of which quantifies a different aspect of what a "good" classification is. For binary probabilistic classifiers, we assume (unless otherwise specified) that an example with sigmoid output score (loosely interpreted as probability) $P_{\rm out} > 0.5$ is classified as an LSBG, while an example with $P_{\rm out} < 0.5$ is classified as an artifact. We also refer to the LSBGs as the positive class [1] and artifacts as the negative class [0].

True positives (TP) are the correctly classified positive examples, and we analogously define the true negatives (TN), false positives (FP) and false negatives (FN). All the classification metrics, in a binary setting, can be expressed as combinations of these quantities.

The Receiver Operating Characteristic (ROC) curve is a commonly used graphical way to evaluate the performance of a binary classifier; the true positive rate (TPR = TP/(TP+FN)) is plotted versus the false positive rate (FPR = FP/(FP+TN)) at different output thresholds; A derived metric is the Area Under the ROC Curve (AUC). ROC curves are useful for visual inspection of the performance of different classifiers and of their uncertainties.

One of the most widely used evaluation metrics is the *accuracy*, which measures the fraction of the correct predictions among the total sample examined:

$$accuracy = \frac{\rm TP+TN}{\rm TP+TN+FP+FN}. \quad (3)$$

However, specific problems or applications may have specific requirements that are not fully captured in the overall accuracy. For example, we may be interested in the *completeness* of our classification, in other words, what fraction of the LSBGs were actually classified as such (this metric is also known as "*recall*", "*sensitivity*" or "*True Positive Rate*" in the machine learning literature):

$$completeness = \frac{\rm TP}{\rm TP+FN}. \quad (4)$$

Another useful quantity is the *purity* of the classification: the fraction of objects classified as LSBGs that are true LSBGs (this quantity is also known as "*precision*" or "*Positive Predictive Value*" in the machine learning literature):

$$purity = \frac{\rm TP}{\rm TP+FP}. \quad (5)$$

Finally, we also present the confusion matrix, which includes all four TP, TN, FP, FN values. The confusion matrix can be used to construct a number of other classification metrics.

4.2. Results

We start by considering the classification results from the two machine learning models (SVMs with an RBF kernel and random forests) described in Sec. 3.1. We use these classification algorithms as implemented in the Python library `scikit_learn` (Pedregosa et al., 2011).[9]

We train these models on the dataset of features derived from `SourceExtractor`, as described in Sec. 2.3. The hyperparameters of the models

[9] `https://scikit-learn.org/stable/index.html`

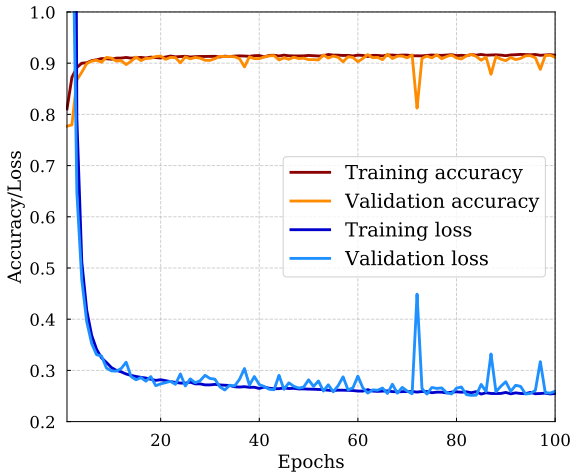


Figure 5: Training and validation accuracy/loss as a function of the training epoch for the *DeepShadows* model. Training was performed on 30,000 images and validation on 5,000. The model reaches a training accuracy of $\sim$ 92% after 100 epochs.

were tuned by searching a grid of values and using five-fold cross validation on the validation set. The best values were found to be $C = 10^4$ (controls how much error is allowed in the soft-margin SVM), $\gamma = 0.001$ (controls the width of the kernel of the non-linear transformation of the data) for the SVM model, while for the random forests model we tune the number of trees in the forest (`n_estimators` = 100) and the number of samples required to split internal nodes (`min_samples_split`=10).

The performance of the models was evaluated on the test set. SVMs reach slightly higher accuracy (81.9% vs 79.7%), higher completeness (or recall, 86.7% vs 80.4%), similar purity (or precision, 79.6% vs 79.7%) and higher AUC (0.894 vs 0.872) compared to the random forest classifier[10]. These results serve as a baseline to compare the performance of our *DeepShadows* CNN model with.

We implement the *DeepShadows* architecture, as described in Sec. 3.2 and Table 1, using the `Keras`[11] framework on a `TensorFlow`[12] backend. We train the model on the training set of 30,000 images described in Sec. 2.3. The weights were updated using Adadelta (Zeiler, 2012), an optimized version of the vanilla stochastic gradient descent algorithm, with a learning rate of $\eta = 0.1$. The loss function used is the binary cross-entropy. The update is performed iteratively in batches of images; we use a batch size of 64. A training *epoch* occurs once every image in the training set has been used to update the network weights. We train our model for 100 epochs; we do not continue for more epochs since the results do not improve more. We validate the process on the validation set of 5,000 images described in Sec. 2.3. In Fig. 5 we present the training history of our model (accuracy/loss as a function of training epoch). We can see that our model converges well and that there are no signs of overfitting or underfitting (the training and validation curves closely follow each other).

Table 2: Comparison of the classification metrics for the three machine learning models presented in Sec. 4.2.

Metric \ Model	SVM	RF	CNN
Accuracy	0.819	0.797	0.920
Completeness	0.867	0.804	0.944
Purity	0.796	0.797	0.903
AUC score	0.894	0.872	0.974

The *DeepShadows* CNN classifier reaches an accuracy of 92.0%, completeness (recall) of 94.4% and purity (precision) of 90.3% and AUC score equal to 0.974, all evaluated on the test set of 5,000 images. These values are significantly higher than those obtained from the SVM and random forest models (see also Table 2 for a direct comparison). Note that these classical machine learning models were trained on a physically motivated set of features that is not guaranteed to be optimal, while *DeepShadows* works directly at the pixel level.

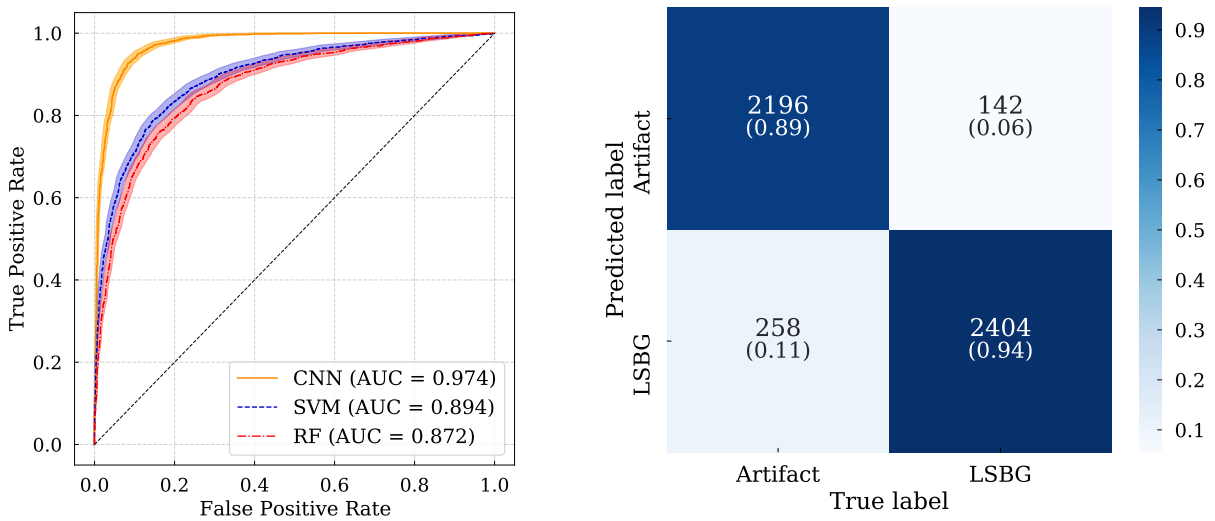
The fact that *DeepShadows* is a more powerful classifier can be visually demonstrated by plotting the ROC curves of the three models (left panel of Fig. 6). In the same figure we also show the AUC scores, as well as the 95% confidence intervals on the ROC curves, estimated using the bootstrap method on the test set (see Sec. 6).

On the right-hand side of Fig. 6 we present the confusion matrix for *DeepShadows* that shows the number of the correctly classified and misclassified objects. Many common classification metrics can

[10] We note again that the SVM model discussed here is different from the one described in the second bullet of Sec. 2.2. The SVM presented here is trained on the same annotated dataset of LSBGs and artifacts that was used to train *DeepShadows* and that had confused the original SVM model (Sec. 2.2).

[11] `https://keras.io/`

[12] `https://www.tensorflow.org/`



(a) (b)

Figure 6: (a) ROC curves for the *DeepShadows* CNN model (orange, solid line), the SVM model (blue dashed line), and the random forest model (red dashed-dotted line) evaluated on the test set. The diagonal dashed line corresponds to the performance of a random classifier. We also show the 95% confidence intervals on the vertical direction (true positive rate). These were estimated using the bootstrap method on the test set of images (see Sec. 6). (b) Confusion matrix of the *DeepShadows* model predictions on the test set. The values in parentheses correspond to the normalized version of the matrix, obtained by dividing the number of objects in each case by the total number of objects in each category (true label).

be derived from the confusion matrix. In parentheses we present the entries of the normalized confusion matrix, which can be obtained by dividing by the total number of objects in each (true label) category. We see that most misclassification cases occur in artifacts classified as LSBGs, something that is also evident from the lower value of purity compared to completeness.

As can be seen in Fig. 2b, the objects in our sample (LSBGs and artifacts) span a wide range in surface brightness ($24.3 < \bar{\mu}_{\rm eff}(g) \lesssim 27.0$ mag/arcsec^2). To investigate whether the performance of our classifier depends on the surface brightness, we split the test sample into three bins according to their g-band surface brightness ($[27-26, 26-25, 25-24.3]$ mag/arcsec^2), and we evaluate the performance of the classifier in each bin independently. Since the samples are not balanced in these bins (i.e., there are more artifacts than LSBGs in the faintest bin, while the opposite is true in the brightest bin), we calculate the *balanced accuracy* in each bin, which is defined as the average of recall obtained for each class.

We present the results of this study in Fig. 7. The balanced accuracies in the three bins (starting from the faintest) are 0.906, 0.926, and 0.874, respectively. This variation in performance is small and comparable to the expected intrinsic uncertainty in the prediction of the classifier (see Sec. 6). Inter-

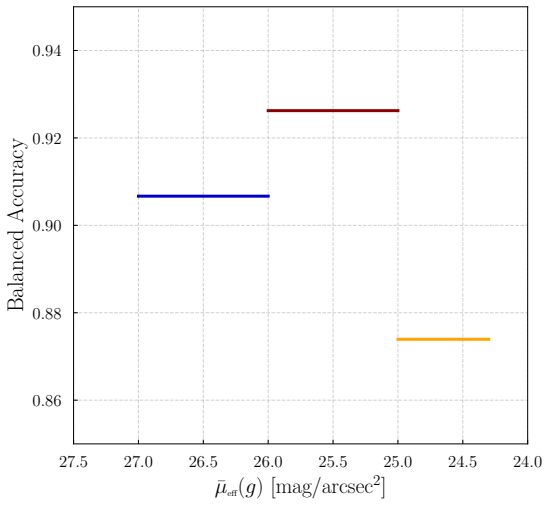


Figure 7: Balanced accuracy score of the *DeepShadows* predictions on the test set, for the objects in three different surface brightness bins.

estingly, we see that the worst performance comes from the brightest bin. We find that a larger fraction of artifacts are misclassified as LSBGs in that bin.

4.3. Interpretation of Results

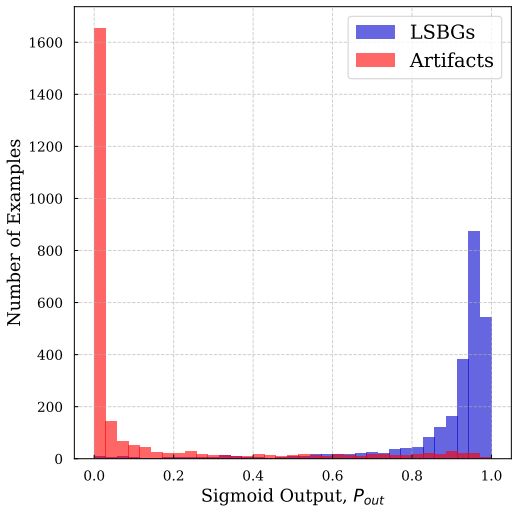


Figure 8: Output probabilities from the *DeepShadows* CNN for the LSBGs and artifacts in the test set.

The *DeepShadows* CNN is a classifier that outputs a score that can be interpreted as approximately the probability that an example is an LSBG. The classification results presented so far assume a threshold $P_{\rm out} = 0.5$ (any image with higher output score is classified as an LSBG, while any image with lower output score is classified an artifact). We can get more insight about the classification outcomes, and try to interpret the results, by examining the predicted output scores for the objects in the test set. We plot these sigmoid output scores in Fig. 8. Output scores of those objects with true label "artifact" are in red and those with true label "LSBG" are in blue. Artifacts are found to be more concentrated towards the $P_{\rm out} = 0$, implying that most can be easily distinguished from LSBGs. However, there is also a long tail in this distribution with some objects that are labeled (true value) as artifacts but have been assigned a high confidence level of being LSBGs. LSBGs, on the other hand, have a wider distribution in output scores (less concentrated towards $P_{\rm out} = 1$) but a less significant tail to very low scores.

To better understand our results it is useful to inspect some of the objects that: (a) were assigned to the wrong class but with high confidence, or (b) were assigned to the correct class with high confidence. To interpret the classification results we employ a recent technique, called Gradient-weighted Class Activation Mapping (Grad-CAM, Selvaraju et al., 2016). Grad-CAM allows us to produce "visual explanations" for the classification results, by highlighting the most important regions in the classification procedure. We provide more technical details about Grad-CAM in Appendix B; here we just note that these images are produced by calculating the gradients of the feature maps of the last convolutional layer with respect to the output score for each class.

In Fig. 9 we present examples and corresponding Grad-CAMs for randomly selected (clockwise): true negatives, false negatives, true positives, false positives. All these examples were classified with high confidence to their assigned categories ($P_{\rm out} > 0.8$ for those classified as LSBGs and $P_{\rm out} < 0.2$ for those classified as artifacts).

The images of the objects classified as negatives (artifacts), both true and false, are characterized by the presence of off-centered light sources (such as stars or components of galaxies). The fact that these are the important regions for the classification problems is also confirmed by Grad-CAM maps. Especially in the case of false negatives, we see that the central LSBGs are shadowed by the presence of other nearby objects that contribute to the decision of *DeepShadows* to classify them as artifacts.

On the other hand, the images of those objects classified as LSBGs (positive class) are dominated by the presence of a central object; this can also be seen in the highlighted regions of the Grad-CAM maps. Interestingly, the high-confidence false positives presented here seem to be real galaxies. These objects were likely rejected out of an abundance of caution when visually selecting LSBGs, since these objects were generally more compact or faint compared to other LSBGs. The neural network classifier is able to "correct" the human labeling.

5. Transfer Learning

In the previous section, both the training and evaluation of the classifiers used data from the same survey, namely DES. These results demonstrate that a CNN classifier trained on a large training sample can be used to separate LSBGs from artifacts with a high accuracy (and purity/completeness). However, a *large* number of

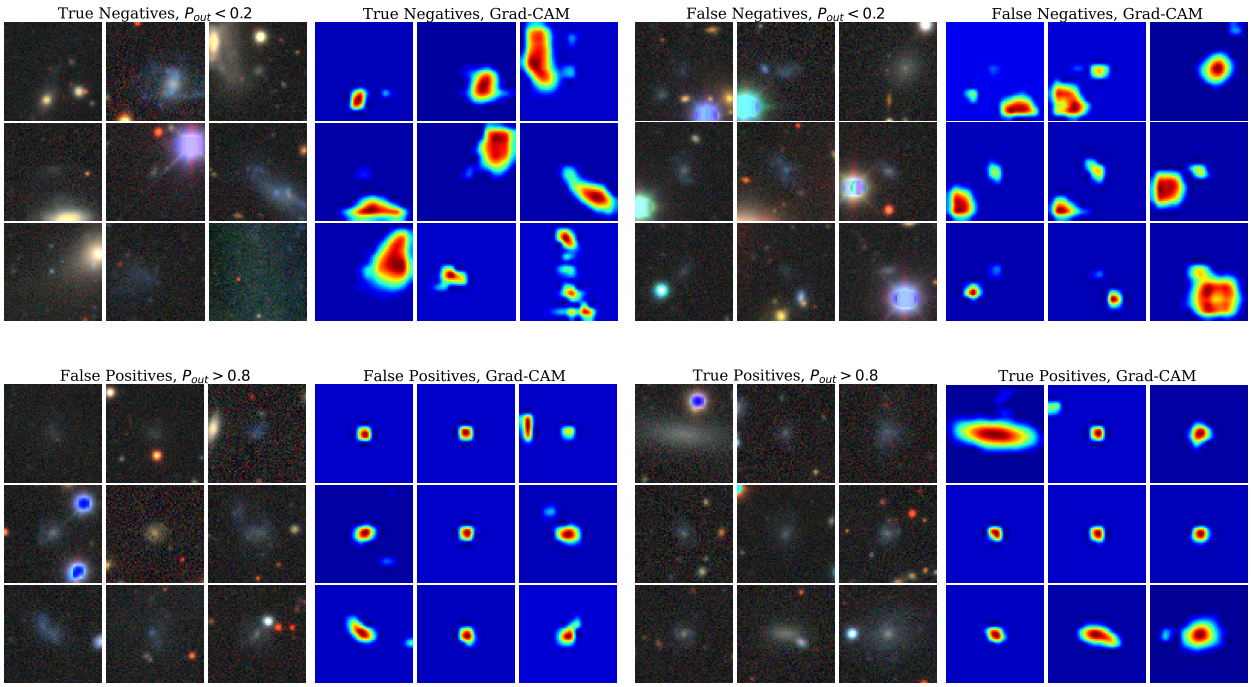


Figure 9: Examples of objects classified with high output score in their respective class and corresponding Grad-CAM visualization maps for the same objects. Clockwise: True Negatives, False Negatives, True Positives, False Positives (same arrangement as the confusion matrix).

labeled training examples is required. In this section we explore whether we can a classifier trained on one survey and apply it to data from another survey. If such an approach is successful, it can significantly reduce the need to generate large training sets via visual inspection in future surveys.

Transfer learning refers to the process of training a machine learning algorithm to perform a task and then using it to perform another related task or perform the same task on a dataset with different specifications (e.g., Weiss et al., 2016). Transfer learning has found many applications, including image recognition problems (e.g., Pan and Yang, 2010; Bengio, 2012; Yosinski et al., 2014; Zhuang et al., 2019). Its power has recently been explored in astronomy (Vilalta, 2018), especially in the context of cross-survey classification, namely in the field of galaxy morphology prediction (Domínguez Sánchez et al., 2019). The use of transfer learning has also been investigated for classification of galaxy mergers (Ackermann et al., 2018), radio galaxy classification (Tang et al., 2019), star-galaxy classification (Wei et al., 2020), even glitch classification of LIGO events (George et al., 2018).

Here we use data from the HSC SSP, for which we have a small visually classified sample of LSBGs (Sec. 2.4), to study how successfully a model trained on one survey can be used to distinguish between LSBGs and artifacts in another survey. Following Domínguez Sánchez et al. (2019) we consider two cases:

a) Apply the *DeepShadows* model, which was trained on DES data, directly to the HSC SSP data (test set) without any further training.

b) Before predicting on the HSC SSP test set, we use a small set of 320 objects from the HSC SSP to perform a *fine-tuning* step. We re-train the whole model using a much smaller learning rate (in order to keep the change of the weights low and avoid overfitting), $\eta = 0.005$, for 30 epochs, using a batch size of 16 (Note that, alternatively, one can re-train only the final, dense layers. We do not consider this case here).

We present the results (confusion matrices and ROC curves) for the two cases in Fig. 10. Before fine tuning, *DeepShadows* has an accuracy of 82.1%, purity (precision) of 84.9% and completeness (re-

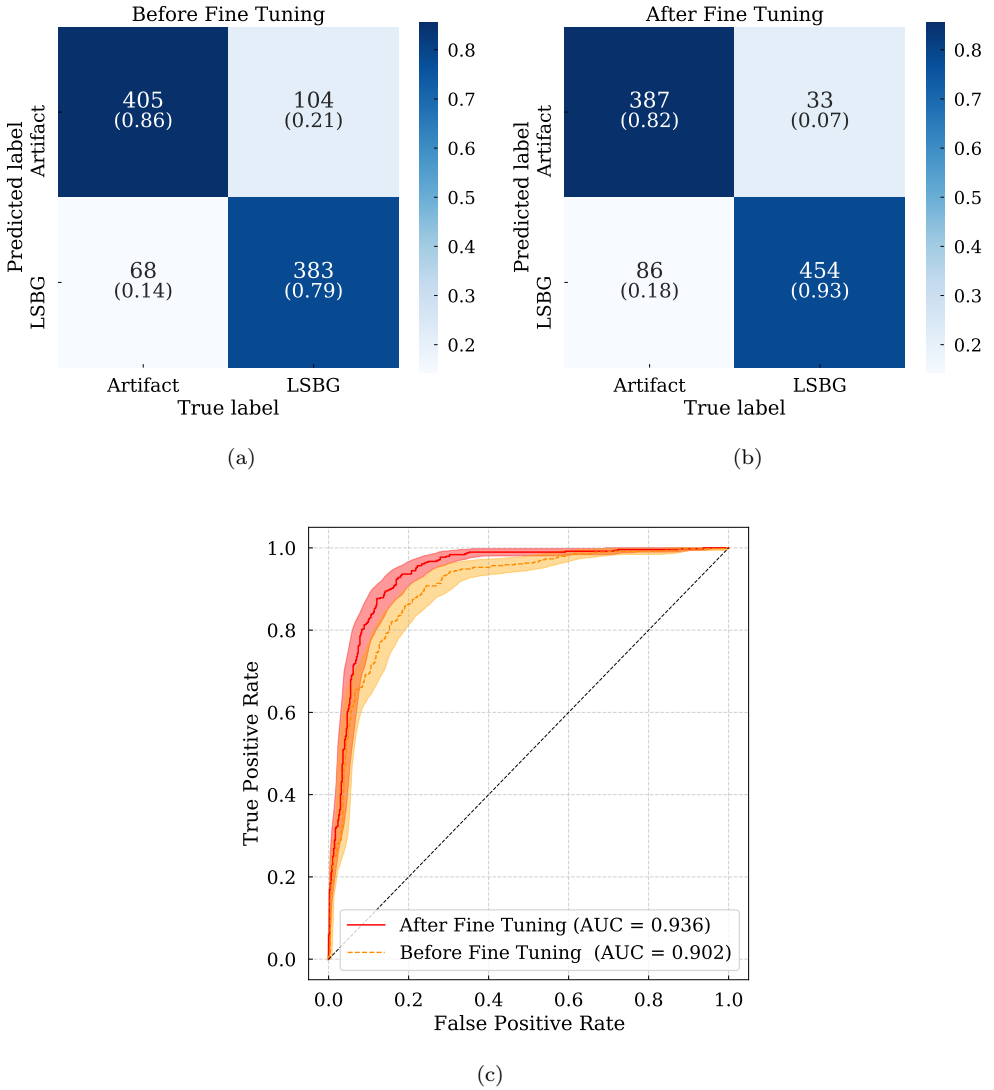


Figure 10: Transfer learning results on the HSC SSP test set. (a) Confusion matrix (raw and normalized values) before fine tuning. (b) Confusion matrix (raw and normalized values) after fine tuning. (c) ROC curves and AUC scores, before (orange, dashed line) and after (red, solid line) fine tuning. The bands correspond to the 95% confidence level intervals.

call) of 78.6%. Classification performance significantly improves with fine tuning, as can be seen by inspecting the two ROC curves (and the corresponding AUC scores) in panel (c) of Fig. 10. The better performance is driven by the increased number of true positives and correspondingly smaller number of false negatives, as can be seen by comparing the confusion matrices in panels (a) and (b) of Fig. 10. The overall accuracy after fine tuning reaches a value of 87.6%, the purity a value of 84.1% and the completeness an impressive 93.2% (almost as good as the application to the one we get from applying to the DES data).

We have demonstrated that we can achieve good performance classifying LSBGs in HSC SSP using transfer learning with a fairly small set for retraining. We would like to quantify the benefit of transfer learning by estimating the size the HSC training set that would be needed to reach comparable accuracy without using transfer learning. Unfortunately, given the small size of the HSC data set, we cannot directly answer this question. However, we can use the DES data to quantify model performance as a function of training set size. We

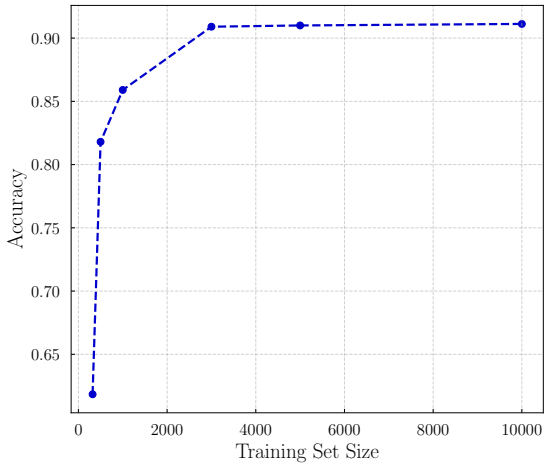


Figure 11: Accuracy, evaluated on the test set, for the *DeepShadows* model trained on progressively larger datasets.

train *DeepShadows* from scratch using progressively larger subsets of the full DES training set. We randomly select 360, 500, 1000, 3000, 5000, and 10000 examples objects in each run; we evaluate the performance (accuracy) on the same test set as the one used in Sec. 4. To prevent overfitting, which is likely to happen when small training sets are used, we employ early stopping.

The results are presented in Fig. 11. The accuracy when the small training set of 360 examples is used is very low, $\sim$ 61.6%; it rapidly increases though reaching $\sim$ 81.8% when 500 examples are used and $\sim$ 85.9% when the training set has a size of 1000. After increasing the training size to 3000 the accuracy almost plateaus with the three larger training sets giving accuracies of 90.9%, 91.0%, and 91.2%, respectively. To achieve an accuracy of 87.6% (the one achieved using transfer learning) a training set of $\sim$ 2000 object is required, about one order of magnitude larger than the sample we used for fine tuning. We note that the accuracy reaches high values (comparable to those reached when the full dataset was used) even with relatively small training sets.

6. Uncertainty Quantification

We have presented performance metrics for our models without discussing potential uncertainties in these estimates. We now consider three potential sources of uncertainty:

a) Random statistical uncertainties when calculating the evaluation metrics on the test set (subsection 6.1).

b) Uncertainties arising from randomly splitting the data into training, validation, and test sets (subsection 6.2).

c) Label noise from the presence of mislabeled examples in our training set (subsection 6.3).

Our analysis is not intended to be exhaustive. For a more complete discussion of the uncertainties present in deep learning models see e.g., Kendall and Gal (2017), Hüllermeier and Waegeman (2019), Caldeira and Nord (2020), and references therein.

6.1. Bootstrap resampling of test set

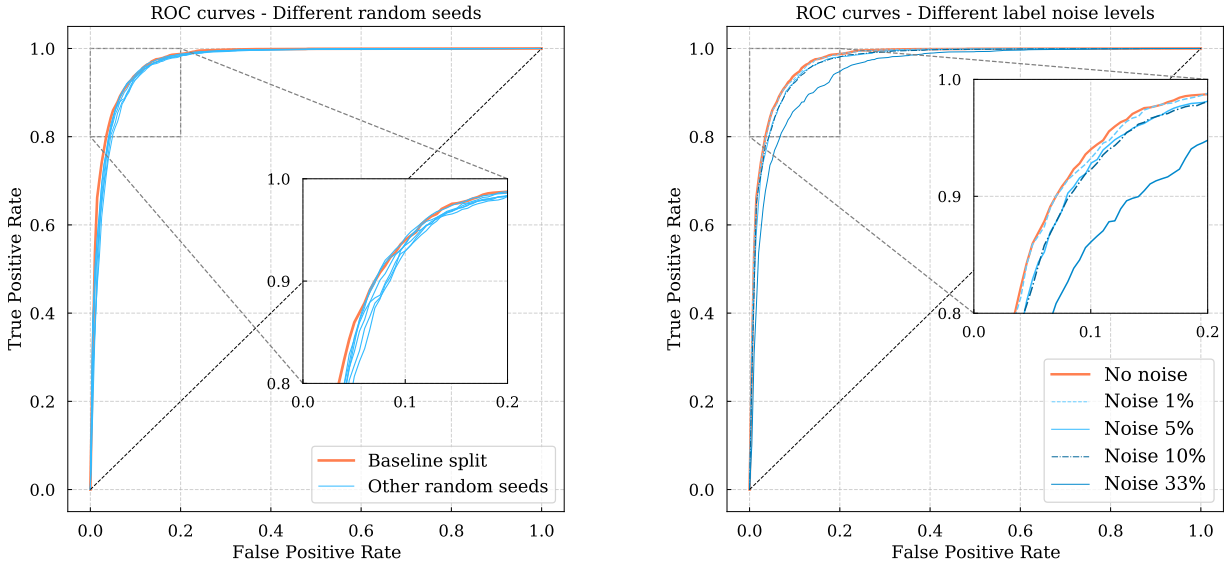
We estimate 95% confidence intervals on the classification metrics by bootstrapping the test set 1000 times with replacement and classifying each realization. We present these confidence intervals in Table 3; the 95% intervals for the true positive rate are also presented in panel (a) of Fig. 6. Note that this approach is not comprehensive. In principle, we could have bootstrapped the training set to evaluate model error in addition to statistical prediction error. However, re-training *DeepShadows* 1000 times was computationally prohibitive.

Table 3: 95% Confidence intervals on the classification metrics from Bootstrap resampling of the test set.

Metric	**95% Confidence Interval**
Accuracy	0.912 – 0.928
Completeness	0.935 – 0.953
Purity	0.891 – 0.914
AUC score	0.970 – 0.974

The typical interval for all parameters is ($\sim$1.5%). This is much smaller than the difference between the performance of *DeepShadows* and the other machine learning models, robustly demonstrating that *DeepShadows* performs better in classifying galaxies and artifacts. The confidence intervals of the SVM and random forest models are of similar width.

Another popular method for uncertainty estimation in deep learning models is Monte Carlo dropout (Gal and Ghahramani, 2015). This method uses dropout at testing time to produce slightly different label predictions on the test set each time we make predictions. We have implemented this method, and we found results comparable to those from bootstrapping.



(a) (b)

Figure 12: (a) ROC curves of the performance on the test set using the baseline split (orange line) between training-validation-test sets and using splits with different random seeds (blue lines). (a) ROC curves of the performance on the test set using different levels of label noise in the training set, starting from no noise (orange line) to 33% random mislabeling.

Table 4: 95% Confidence Intervals on the classification metrics from Bootstrap for the transfer learning task, after fine-tuning, evaluated on the HSC SSP test set.

Metric	**95% Confidence Interval**
Accuracy	0.856 – 0.896
Completeness	0.910 – 0.954
Purity	0.808 – 0.871
AUC score	0.921 – 0.951

We have seen in Fig. 10 that the confidence intervals in the ROC curves (true positive rates) in the case where we explored transfer learning are wider than those of the main section (training and test on the DES data). In Table 4 we present 95% confidence intervals for the transfer learning task, after fine tuning, evaluated on the HSC SSP test set. As we can see the typical range in the evaluation metrics in this case is ∼0.04–0.05, significantly larger than before.

6.2. DES dataset assignment

We have noticed that the model performance is sensitive (at a level similar to the uncertainties described in the previous section) to the random assignment of examples into the training-validation-test sets. We study variations in model performance stemming from random assignment by splitting the whole dataset into training-validation-test using 6 different random seeds, retraining using each new training set, and evaluating on the new test sets. Note that we do not change the sizes of these sets; these are always 30,000 (training), 5,000 (validation), 5,000 (test).

In Table 5 we present the classification metrics for each one of the runs while in the left-hand side of Fig. 12 we show the ROC curves for each one of these cases. Due to the limited number of runs (constrained by the cost of re-training the model each time) we do not present summary statistics (like 95% confidence intervals); however we can see that for each metric, the values span a range similar to those presented in Table 3. The baseline split, used in Sec. 4 was selected as one of the better performing models after several trials. In the GitHub page of this project we make available the on-sky coordinates (right ascension, declination) of the training, validation and test sets under the baseline split.

6.3. Impact of mislabeling

A final source of uncertainty that we consider is the presence of mislabeled examples in the training set, also known as *label noise*. In Sec. 4.3 we showed

Table 5: Classification metrics for the baseline split of training-validation-test sets and six alternative splits using different random seeds. Because of the small number of runs we present each case individually and not as an interval like in Table 3.

Metric / Seed #	Accuracy	Completeness	Purity	AUC score
Baseline	0.920	0.944	0.903	0.974
1st run	0.912	0.974	0.868	0.967
2nd run	0.915	0.972	0.870	0.971
3rd run	0.917	0.950	0.889	0.968
4th run	0.914	0.936	0.896	0.968
5th run	0.920	0.917	0.924	0.972
6th run	0.915	0.937	0.898	0.971

Table 6: Classification metrics using the baseline split and different levels of random label noise (mislabeling) in the training set.

Metric / Noise level	Accuracy	Completeness	Purity	AUC score
No noise	0.920	0.944	0.903	0.974
Noise 1%	0.918	0.945	0.899	0.974
Noise 5%	0.914	0.935	0.890	0.970
Noise 10%	0.909	0.897	0.923	0.969
Noise 33%	0.883	0.892	0.880	0.950

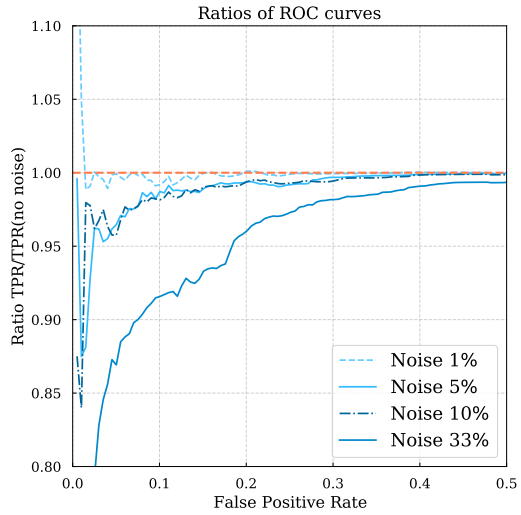


Figure 13: Ratio of the ROC curves with different label noise levels to the one without noise. More specifically, we plot the ratio of the true positive rates, as a function of the false negative rate. The horizontal, orange, dashed, line corresponds to ratio = 1. Notice that the range in the false positive rate plotted is 0–0.5 since for higher values all curves converge to a value equal to one.

that our dataset contains some LSBGs that are labeled as artifacts. Generally, we know that we were conservative when performing the visual inspection, meaning that we favored purity over completeness when assembling the LSBG catalog (if we were unsure if an object was a LSBG we preferred to flag it as an artifact).

Label noise has been extensively studied in the deep learning literature (for overview papers see e.g., Frenay and Verleysen, 2014; Algan and Ulusoy, 2019; Song et al., 2020). The general conclusion is that neural networks are relatively robust to label noise, in some cases proving to perform well even in the presence of large noise (Rolnick et al., 2017). Most of these studies consider well-known annotated datasets (MNIST, CIFAR, etc.) and introduce artificial label noise. However, it is interesting and useful to study potential effects of mislabeling in our dataset.

The best way to do this study would be to identify the mislabeled examples by carefully inspecting and reclassifying them. However, such a detailed study is time-intensive and beyond the scope of this work. Here we consider the impact of label noise in the following way: we randomly select a number of examples (LSBGs and artifacts alike) equal to

1%, 5%, 10% and 33% of images from the training set and we flip their label. We retrain the model each time and we evaluate on the test set.

The classification performance metrics for each noise level (and for the baseline case without noise) can be found in Table 6. The corresponding ROC curves, that allow for a visual comparison of the performance of the models, can be seen in the right-hand panel of Fig. 12. Furthermore, in Fig. 13 we plot the ratios of the ROC curves for the models with different noise levels to that without label noise, for better inspection of the differences. We can see that for small ($\lesssim 10\%$) label noise levels the reduction in performance is minimal. Reduced accuracy only becomes noticeable once the label noise reaches 33%, though even in this case the accuracy reduction is not very large. Our results thus confirm the studies of Rolnick et al. (2017).

So far we have considered a purely random noise, in the sense that we randomly selected an equal number of LSBGs and artifacts and flipped their labels. We repeated the above exercise introducing targeted noise—i.e., we selected only artifacts and changed their label to LSBGs in the first case, and LSBGs that changed their labels to artifacts in the second case. In both cases the results were qualitatively similar to the random noise case. We conclude that the presence of label noise in our sample, either random or biased against one of the two categories, does not significantly change the model performance.

7. Discussion and Conclusions

In this work we presented the application of deep learning to the problem of automatic LSBG/artifact classification in astronomical images of LSBG candidates. This study was enabled by the availability of large samples of both LSBGs and artifacts from DES (Tanoglidis et al., 2021).

We showed that a simple CNN architecture with three convolutional and two dense layers can achieve classification accuracy of 92.0% (completeness 94.4% and purity 90.3%) and significantly improves over conventional machine learning models (SVMs and random forests) trained on `SourceExtractor`-derived features (accuracy 81.9% and 79.7%, respectively). This performance is found to be relatively robust to label noise.

We also demonstrated that knowledge obtained from one survey (training on DES data) can be transferred to another survey (prediction on HSC SSP data, accuracy 82.1%) and the performance of this *transfer learning* can be significantly improved when the model is retrained on a small sample of examples from the new survey (accuracy 87.6%).

These results are promising and impactful for two reasons. First, automating the classification process (or, at least, significantly reducing the need for visual inspection) will be necessary given the data volumes of future surveys such as LSST and Euclid, and even future analyses of current surveys, such as the full 6-year of DES observations and future data releases from HSC SSP. Second, automated classification makes it much easier to characterize, in an unbiased way, the completeness/detection efficiency of future LSBG catalogs. The standard way to characterize detection efficiency is by injecting a large number of mock galaxies with a known parameters (e.g., effective radius, surface brightness, Sérsic index etc.) into the imaging data and then applying the same detection pipeline. The efficiency can be calculated as a function of galaxy parameters by measuring the fraction of mock galaxies that are recovered (e.g., Song et al., 2012; Suchyta et al., 2016; Venhola et al., 2018). To characterize the detection efficiency over the allowed space of galaxy parameters, it is often necessary to simulate more mock galaxies than are observed in the data itself. This makes unbiased human classification very challenging.

We have presented a study of CNNs for LSBG–artifact separation[13]. Our primary goal was to demonstrate the feasibility of such an approach, and we briefly outline possible further investigations. Specifically, improvements in the CNN model, the training data, the use of domain adaptation techniques for transfer learning, and the systematic study of uncertainties would all improve future models for LSBG classification and discovery.

In particular, CNN architectures have a large number of tunable hyperparameters—e.g., number of layers, filters, kernel sizes, dropout levels, regularization parameters. Finding the optimal combination in a manner similar to that used for the SVM and random forest classifiers is computation-

[13]Note that the reason we consider only these two categories is because of the preprocessing steps described in Sec. 2.3; because of them the final candidate sample is forced to contain only those two categories. Without this preprocessing one may consider a multi-class classification, like normal galaxies, stars etc.

ally expensive. We have tested that the architecture presented here is robust to small changes – e.g., the model with 3 convolutional layers performs better compared to one with 2 or 4 layers, etc. However, a grid of hyperparameters should be explored to ensure an architecture that gives the best results. This would likely require parallelizing the model training and evaluation processes to reduce computational time. Furthermore, more complex types of networks, such the Residual Neural Networks (ResNet He et al., 2015) that allow for very long (large number of epochs) training, should be explored.

The quality of data used for training and testing is also very important. In Sec. 6.3 we showed that the presence of label noise does not significantly change performance. However, the selection of objects for which the labels were flipped was totally random; in practice the objects that are more challenging to characterize and thus prone to mislabeling are of a specific type – for example very faint or very compact. It would be useful to reclassify our sample into different confidence categories and check how the performance of the classifier changes when only high-confidence LSBGs and artifacts are used for training. Having a test set without label noise is also important; we have seen that some of the "misclassifications" were actually result of label noise, thus leading to slightly misestimated performance metrics (here we refer to noise introduced by a human labeler, not the artificial label we introduced in Sec. 6.3). Furthermore, *data augmentation* is a commonly used technique that could be explored in the future (e.g., Shorten and Khoshgoftaar, 2019). Data augmentation is a regularization technique used to avoid overfitting, where one increases the number of training examples by adding slightly modified copies of the existing images (rotated, resized etc.).

The topic that likely requires the most detailed further study is that of transfer learning from one survey to another. Here our exploration was minimal, either applying the model trained on DES data directly to HSC SSP data or retraining the whole model on a very small set from HSC SSP. More *domain adaptation* techniques (e.g., Kouw and Loog, 2018; Wang and Deng, 2018) (techniques that allow algorithms trained in one or more "source domains" to be successfully used in a different, but related, "target domain") should be explored before choosing an approach to apply to forthcoming surveys (for domain adaptation applications in astronomy, see e.g., Vilalta et al. 2019; Ćiprijanović et al. 2020a, 2021). These techniques would allow the models to be successfully applied to the new data without the need to retrain the model later and more importantly to manually label new "target" datasets since these techniques often use unlabeled target datasets. This makes the process much faster. Furthermore, the benefit of using larger example sets from the target survey for re-training of the model should also be explored.

Finally, the topic of uncertainty quantification in deep learning is a very active area of research; here a simple error estimation was presented. Future exploration should include bootstrapping the training set and not just the test set (something that would be computationally expensive and should be parallelized), sources of statistical and systematic errors (known as "epistemic" and "aleatoric" in the machine learning community) should be studied separately, as well as potential correlations between the two.

We plan to address some of these questions in future work, but the results presented here are already very promising for the upcoming analysis of the full six years of DES data. However, given the inherent challenges present in classifying very low-surface-brightness objects, the performance of a CNN classifier may never reach human-level accuracy. We argue that as we enter a new, big-data based, era in astronomy, the community should be ready to take a leap of faith in accepting the presence of small and well-characterized classification errors in favor of the great statistical power that comes when assembling large catalogs of objects, using automated deep learning methods, that can illuminate the low-surface-brightness universe.

Acknowledgement

We are grateful to Johnny Greco for providing us a list of visually rejected artifacts from the HSC survey, described in Sec 2.4,to Brian Nord for useful comments and suggestions, and two anonymous reviewers that helped us improve this paper. A. Ćiprijanović is partially supported by the High Velocity Artificial Intelligence grant as part of the Department of Energy High Energy Physics Computational HEP sessions program.

We acknowledge the Deep Skies Lab as a community of multi-domain experts and collaborators who've facilitated an environment of open discussion, idea-generation, and collaboration. This com-

munity was important for the development of this project.

This material is based upon work supported by the National Science Foundation under Grant No. AST-2006340. This work was supported by the University of Chicago and the Department of Energy under section H.44 of Department of Energy Contract No. DE-AC02-07CH11359 awarded to Fermi Research Alliance, LLC. This work was partially funded by Fermilab LDRD 2018-052. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

This project used public archival data from the Dark Energy Survey (DES). Funding for the DES Projects has been provided by the U.S. Department of Energy, the U.S. National Science Foundation, the Ministry of Science and Education of Spain, the Science and Technology FacilitiesCouncil of the United Kingdom, the Higher Education Funding Council for England, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, the Kavli Institute of Cosmological Physics at the University of Chicago, the Center for Cosmology and Astro-Particle Physics at the Ohio State University, the Mitchell Institute for Fundamental Physics and Astronomy at Texas A&M University, Financiadora de Estudos e Projetos, Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro, Conselho Nacional de Desenvolvimento Científico e Tecnológico and the Ministério da Ciência, Tecnologia e Inovação, the Deutsche Forschungsgemeinschaft, and the Collaborating Institutions in the Dark Energy Survey.

The Collaborating Institutions are Argonne National Laboratory, the University of California at Santa Cruz, the University of Cambridge, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas-Madrid, the University of Chicago, University College London, the DES-Brazil Consortium, the University of Edinburgh, the Eidgenössische Technische Hochschule (ETH) Zürich, Fermi National Accelerator Laboratory, the University of Illinois at Urbana-Champaign, the Institut de Ciències de l'Espai (IEEC/CSIC), the Institut de Física d'Altes Energies, Lawrence Berkeley National Laboratory, the Ludwig-Maximilians Universität München and the associated Excellence Cluster Universe, the University of Michigan, the National Optical Astronomy Observatory, the University of Nottingham, The Ohio State University, the OzDES Membership Consortium, the University of Pennsylvania, the University of Portsmouth, SLAC National Accelerator Laboratory, Stanford University, the University of Sussex, and Texas A&M University.

Based in part on observations at Cerro Tololo Inter-American Observatory, National Optical Astronomy Observatory, which is operated by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the National Science Foundation.

Appendix A. Three-class classification

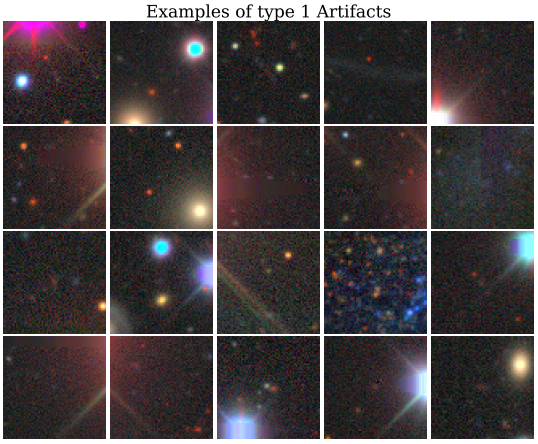
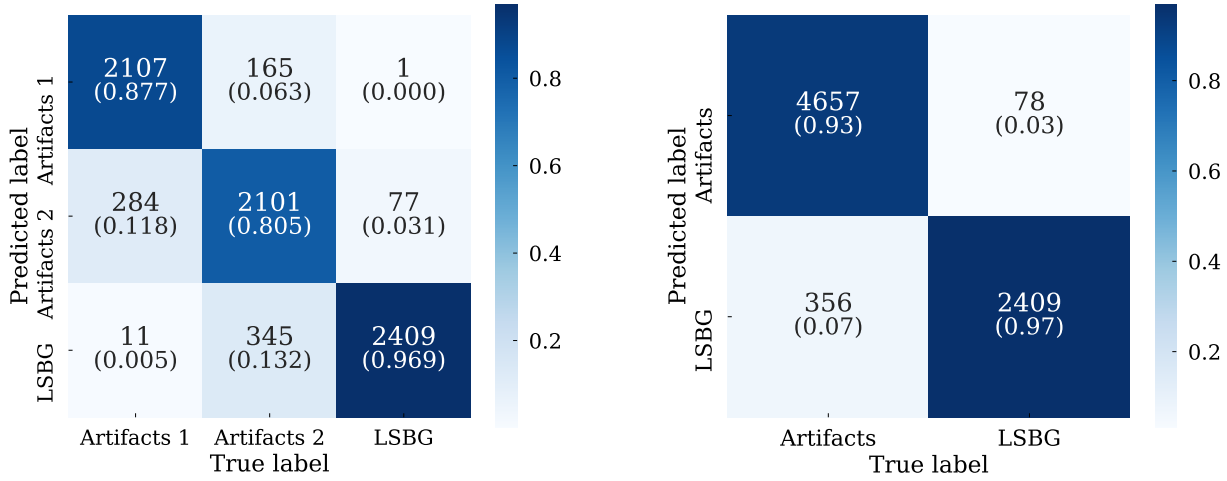


Figure A.14: Randomly selected examples of type 1 artifacts (see text).

In the main body of this paper we considered a two-class classification: one (positive) category of LSBGs and one (negative) category of artifacts – those visually rejected in the third step of the LSBG catalog generation in Tanoglidis et al. (2021), also described in Sec. 2.2. These artifacts were the hardest to classify, since they had been classified as LSBGs by an SVM classifier trained on `SourceExtractor` features. Before the visual inspection the same classifier had rejected a very large number of artifacts (most of them correctly).

We also investigated the performance of *DeepShadows* on a three-class classification problem where two artifact classes are considered, by adding a sample of 20,000 randomly selected artifacts from those rejected by the SVM classifier. We call this class "Artifacts 1", while the artifacts considered in the main body of the paper are called "Artifacts 2". In Fig. A.14 we present a



(a) (b)

Figure A.15: (a) Confusion matrix for the three-class classification problem, where two artifact classes are considered. The numbers in parentheses correspond to the normalized values. (b) "reduced" confusion matrix, where the two artifact classes have been combined.

randomly selected sample of artifacts of the first kind. We see that these artifacts are dominated by the presence of strong diffraction spikes, and they are less confusing (more easily recognized as artifacts) than those presented in Fig. 1.

We keep the same architecture for *DeepShadows*, except that the last dense layer has size of three. We also change the final activation function to softmax and the loss to categorical crossentropy. We train again the model for 100 epochs with a batch size of 64. We split the total dataset of 60,000 object into 45,000 (training), 7,500 (validation), and 7,500 (test) sets.

The resulting three-class confusion matrix of the predictions on the test set can be seen on the left-hand side of Fig. A.15. We can see that there is very small confusion between the "Artifacts 1" class and "LSBGs" categories, confirming our notion that these are artifacts that can be very easily excluded. Interestingly, the classifier is able to distinguish between the two categories of artifacts, too (with some confusion, of course, accuracy $\sim$ 90% when calculating for the submatrix between artifacts only).

In the right-hand panel of Fig. A.15 we combine the two artifact categories, in order to better see the confusion between artifacts and LSBGs. Note that this is now an imbalanced two-class problem, since the artifacts class is twice as large as the LSBGs class. For that reason, accuracy is not a good metric but we can still calculate the completeness and purity, which are more meaningful metrics for the problem at hand. These numbers are 96.8% and 87.1%, respectively and they are comparable (slightly less in purity) to those from the 2-class model discussed in the text.

Our conclusion is that there is not much benefit in using a three-class classification, unless we prefer to eliminate the SVM classification step in future applications.

Appendix B. Grad-CAM details

We present here some technical details of the Grad-CAM technique for highlighting the most important regions for classification in an image. More details can be found in the original article (Selvaraju et al., 2016). We also suggest the following blog post[14] for an explanation of the technique and its application using `Keras`.

We define A^k to be the $k-$th $(k = 1, \ldots, K)$ feature map of the last convolutional layer, that has dimensions $m \times n = Z$ (Pixel values A^k_{ij}, $i = 1, \ldots, m$,

[14] `https://fairyonice.github.io/Grad-CAM-with-keras-vis.html`

$j = 1, \ldots, n$). Let also y^c the output score (probability) for the class c (obviously, here we have only one positive class, thus only one probability score).

If each convolutional kernel captures a specific visual pattern, then each feature map of the final layer will show where this visual pattern exists in the image. We can thus imagine that the classification output depends on a weighted sum of all the feature maps, with weights depending on the importance each feature has for class c. So, the Grad-CAM maps can be written as: $L^c_{\text{Grad-CAM}} \sim \sum_k \alpha^c_k A^k$.

What are the class-dependent weights a^c_k? The idea is that the gradients of the output score with respect to the (i, j) pixel of the $k-$th feature map, $\partial y^c / \partial A^k_{ij}$, measures the effect of that pixel to the classification score. Grad-CAM then proposes to take the average of all pixels (also known as average pooling) as that the weight for map k and class c is:

$$a^c_k = \frac{1}{Z} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial y^c}{\partial A^k_{ij}}. \tag{B.1}$$

Finally, a ReLU function is applied to the weighted sum, to keep the positive regions, so we get:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_{k=1}^{K} a^c_k A^k \right), \tag{B.2}$$

which produces a localization map that retains the spatial information present in the last convolutional layer.

References

Abbott, T.M.C., Abdalla, F.B., Allam, S., Amara, A., et al., 2018. The Dark Energy Survey: Data Release 1. ApJS 239, 18. doi:10.3847/1538-4365/aae9f0, arXiv:1801.03181.

Ackermann, S., Schawinski, K., Zhang, C., et al., 2018. Using transfer learning to detect galaxy mergers. MNRAS 479, 415–425. doi:10.1093/mnras/sty1398, arXiv:1805.10289.

Adami, C., Scheidegger, R., Ulmer, M., et al., 2006. A deep wide survey of faint low surface brightness galaxies in the direction of the Coma cluster of galaxies. A&A 459, 679–692. doi:10.1051/0004-6361:20053758, arXiv:astro-ph/0607559.

Algan, G., Ulusoy, I., 2019. Image Classification with Deep Learning in the Presence of Noisy Labels: A Survey. arXiv e-prints , arXiv:1912.05170 arXiv:1912.05170.

Ball, N.M., Brunner, R.J., 2010. Data Mining and Machine Learning in Astronomy. International Journal of Modern Physics D 19, 1049–1106. doi:10.1142/S0218271810017160, arXiv:0906.2173.

Baron, D., 2019. Machine Learning in Astronomy: a practical overview. arXiv e-prints , arXiv:1904.07248 arXiv:1904.07248.

Bengio, Y., 2012. Deep learning of representations for unsupervised and transfer learning, JMLR Workshop and Conference Proceedings, Bellevue, Washington, USA. pp. 17–36. URL: http://proceedings.mlr.press/v27/bengio12a.html.

Bertin, E., Arnouts, S., 1996. SExtractor: Software for source extraction. A&AS 117, 393–404. doi:10.1051/aas:1996164.

Bom, C., Poh, J., Nord, B., Blanco-Valentin, M., Dias, L., 2019. Deep Learning in Wide-field Surveys: Fast Analysis of Strong Lenses in Ground-based Cosmic Experiments. arXiv e-prints , arXiv:1911.06341 arXiv:1911.06341.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32. URL: http://dx.doi.org/10.1023/A%3A1010933404324, doi:10.1023/A:1010933404324.

Caldeira, J., Nord, B., 2020. Deeply Uncertain: Comparing Methods of Uncertainty Quantification in Deep Learning Algorithms. arXiv e-prints , arXiv:2004.10710 arXiv:2004.10710.

Caldeira, J., Wu, W.L.K., Nord, B., Avestruz, C., Trivedi, S., Story, K.T., 2019. DeepCMB: Lensing reconstruction of the cosmic microwave background with deep neural networks. Astronomy and Computing 28, 100307. doi:10.1016/j.ascom.2019.100307, arXiv:1810.01483.

Cheng, T.Y., Conselice, C.J., Aragón-Salamanca, A., et al., 2020. Optimizing automatic morphological classification of galaxies with machine learning and deep learning using Dark Energy Survey imaging. MNRAS 493, 4209–4228. doi:10.1093/mnras/staa501, arXiv:1908.03610.

Ćiprijanović, A., Kafkes, D., Downey, K., Jenkins, S., Perdue, G.N., Madireddy, S., Johnston, T., Snyder, G.F., Nord, B., 2021. DeepMerge II: Building Robust Deep Learning Algorithms for Merging Galaxy Identification Across Domains. arXiv e-prints , arXiv:2103.01373 arXiv:2103.01373.

Ćiprijanović, A., Kafkes, D., Jenkins, S., et al., 2020a. Domain adaptation techniques for improved cross-domain study of galaxy mergers. arXiv e-prints , arXiv:2011.03591 arXiv:2011.03591.

Ćiprijanović, A., Snyder, G.F., Nord, B., Peek, J.E.G., 2020b. DeepMerge: Classifying high-redshift merging galaxies with deep neural networks. Astronomy and Computing 32, 100390. doi:10.1016/j.ascom.2020.100390, arXiv:2004.11981.

Cohen, Y., van Dokkum, P., Danieli, S., et al., 2018. The Dragonfly Nearby Galaxies Survey. V. HST/ACS Observations of 23 Low Surface Brightness Objects in the Fields of NGC 1052, NGC 1084, M96, and NGC 4258. ApJ 868, 96. doi:10.3847/1538-4357/aae7c8, arXiv:1807.06016.

Cortes, C., Vapnik, V., 1995. Support vector networks. Machine Learning 20, 273–297.

Dai, J.M., Tong, J., 2018. Galaxy Morphology Classification with Deep Convolutional Neural Networks. arXiv e-prints , arXiv:1807.10406 arXiv:1807.10406.

Dalcanton, J.J., Spergel, D.N., Gunn, J.E., Schmidt, M., Schneider, D.P., 1997. The Number Density of Low-Surface Brightness Galaxies with 23 ¡ mu_0 ¡ 25 V Mag/arcsec$\hat{2}$. AJ 114, 635–654. doi:10.1086/118499, arXiv:astro-ph/9705088.

Danieli, S., van Dokkum, P., Merritt, A., et al., 2017. The Dragonfly Nearby Galaxies Survey. III. The Luminosity Function of the M101 Group. ApJ 837, 136. doi:10.3847/1538-4357/aa615b, arXiv:1702.04727.

Davies, A., Serjeant, S., Bromley, J.M., 2019. Using convolutional neural networks to identify gravitational lenses

in astronomical images. MNRAS 487, 5263–5271. doi:10.1093/mnras/stz1288, arXiv:1905.04303.

DES Collaboration, Abbott, T.M.C., Abdalla, F.B., Allam, S., Amara, A., et al., 2018. The Dark Energy Survey: Data Release 1. ApJS 239, 18. doi:10.3847/1538-4365/aae9f0, arXiv:1801.03181.

Dey, A., Schlegel, D.J., Lang, D., et al., 2019. Overview of the DESI Legacy Imaging Surveys. AJ 157, 168. doi:10.3847/1538-3881/ab089d, arXiv:1804.08657.

Dieleman, S., Willett, K.W., Dambre, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. MNRAS 450, 1441–1459. doi:10.1093/mnras/stv632, arXiv:1503.07077.

Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., et al., 2019. Transfer learning for galaxy morphology from one survey to another. MNRAS 484, 93–100. doi:10.1093/mnras/sty3497, arXiv:1807.00807.

Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., Tuccillo, D., Fischer, J.L., 2018. Improving galaxy morphologies for SDSS with Deep Learning. MNRAS 476, 3661–3676. doi:10.1093/mnras/sty338, arXiv:1711.05744.

Feinstein, A.D., Montet, B.T., Ansdell, M., et al., 2020. Flare Statistics for Young Stars from a Convolutional Neural Network Analysis of *TESS* Data. arXiv e-prints , arXiv:2005.07710 arXiv:2005.07710.

Flaugher, B., Diehl, H.T., Honscheid, K., et al., 2015. The Dark Energy Camera. AJ 150, 150. doi:10.1088/0004-6256/150/5/150, arXiv:1504.02900.

Frenay, B., Verleysen, M., 2014. Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems 25, 845–869. doi:10.1109/TNNLS.2013.2292894.

Gal, Y., Ghahramani, Z., 2015. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. arXiv e-prints , arXiv:1506.02142 arXiv:1506.02142.

George, D., Shen, H., Huerta, E.A., 2018. Classification and unsupervised clustering of LIGO data with Deep Transfer Learning. Phys. Rev. D 97, 101501. doi:10.1103/PhysRevD.97.101501, arXiv:1706.07446.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. The MIT Press.

Greco, J.P., Greene, J.E., Strauss, M.A., et al., 2018. Illuminating Low Surface Brightness Galaxies with the Hyper Suprime-Cam Survey. ApJ 857, 104. doi:10.3847/1538-4357/aab842, arXiv:1709.04474.

Hastie, T., 2020. Ridge Regularizaton: an Essential Concept in Data Science. arXiv e-prints , arXiv:2006.00371 arXiv:2006.00371.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition. arXiv e-prints , arXiv:1512.03385 arXiv:1512.03385.

Hilker, M., Kissler-Patig, M., Richtler, T., Infante, L., Quintana, H., 1999. The central region of the Fornax cluster. I. A catalog and photometric properties of galaxies in selected CCD fields. A&AS 134, 59–73. doi:10.1051/aas:1999433, arXiv:astro-ph/9807143.

Hüllermeier, E., Waegeman, W., 2019. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. arXiv e-prints , arXiv:1910.09457 arXiv:1910.09457.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv e-prints , arXiv:1502.03167 arXiv:1502.03167.

Iqbal, H., 2018. Harisiqbal88/plotneuralnet v1.0.0. URL: https://doi.org/10.5281/zenodo.2526396, doi:10.5281/zenodo.2526396.

Ivezic, Z., Connolly, A.J., VanderPlas, J.T., Gray, A., 2014. Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data. Princeton University Press, USA.

Jacobs, C., Collett, T., Glazebrook, K., et al., 2019. Finding high-redshift strong lenses in DES using convolutional neural networks. MNRAS 484, 5330–5349. doi:10.1093/mnras/stz272, arXiv:1811.03786.

Kaviraj, S., 2020. The low-surface-brightness Universe: a new frontier in the study of galaxy evolution. arXiv e-prints , arXiv:2001.01728 arXiv:2001.01728.

Kendall, A., Gal, Y., 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? arXiv e-prints , arXiv:1703.04977 arXiv:1703.04977.

Kim, E.J., Brunner, R.J., 2017. Star-galaxy classification using deep convolutional neural networks. MNRAS 464, 4463–4475. doi:10.1093/mnras/stw2672, arXiv:1608.04369.

Kouw, W.M., Loog, M., 2018. An introduction to domain adaptation and transfer learning. arXiv e-prints , arXiv:1812.11806 arXiv:1812.11806.

Lanusse, F., Ma, Q., Li, N., et al., 2018. CMU DeepLens: deep learning for automatic image-based galaxy-galaxy strong lens finding. MNRAS 473, 3895–3906. doi:10.1093/mnras/stx1665, arXiv:1703.02642.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. URL: https://doi.org/10.1038/nature14539, doi:10.1038/nature14539.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, pp. 2278–2324.

Martin, G., Kaviraj, S., Laigle, C., et al., 2019. The formation and evolution of low-surface-brightness galaxies. MNRAS 485, 796–818. doi:10.1093/mnras/stz356, arXiv:1902.04580.

Martin, N.F., Ibata, R.A., McConnachie, A.W., et al., 2013. The PAndAS View of the Andromeda Satellite System. I. A Bayesian Search for Dwarf Galaxies Using Spatial and Color-Magnitude Information. ApJ 776, 80. doi:10.1088/0004-637X/776/2/80, arXiv:1307.7626.

McConnachie, A.W., 2012. The Observed Properties of Dwarf Galaxies in and around the Local Group. AJ 144, 4. doi:10.1088/0004-6256/144/1/4, arXiv:1204.1562.

Merritt, A., van Dokkum, P., Danieli, S., et al., 2016. The Dragonfly Nearby Galaxies Survey. II. Ultra-Diffuse Galaxies near the Elliptical Galaxy NGC 5485. ApJ 833, 168. doi:10.3847/1538-4357/833/2/168, arXiv:1610.01609.

Mihos, J.C., Durrell, P.R., Ferrarese, L., et al., 2015. Galaxies at the Extremes: Ultra-diffuse Galaxies in the Virgo Cluster. ApJ 809, L21. doi:10.1088/2041-8205/809/2/L21, arXiv:1507.02270.

Mihos, J.C., Harding, P., Feldmeier, J.J., et al., 2017. The Burrell Schmidt Deep Virgo Survey: Tidal Debris, Galaxy Halos, and Diffuse Intracluster Light in the Virgo Cluster. ApJ 834, 16. doi:10.3847/1538-4357/834/1/16, arXiv:1611.04435.

Muñoz, R.P., Eigenthaler, P., Puzia, T.H., et al., 2015. Un-

veiling a Rich System of Faint Dwarf Galaxies in the Next Generation Fornax Survey. ApJ 813, L15. doi:10.1088/2041-8205/813/1/L15, arXiv:1510.02475.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359. doi:10.1109/TKDE.2009.191.

Paranjpye, D., Mahabal, A., Ramaprakash, A.N., et al., 2020. Eliminating artefacts in polarimetric images using deep learning. MNRAS 491, 5151–5157. doi:10.1093/mnras/stz3250, arXiv:1911.08327.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Ribli, D., Pataki, B.Á., Zorrilla Matilla, J.M., et al., 2019. Weak lensing cosmology with convolutional neural networks on noisy data. MNRAS 490, 1843–1860. doi:10.1093/mnras/stz2610, arXiv:1902.03663.

Rolnick, D., Veit, A., Belongie, S., Shavit, N., 2017. Deep Learning is Robust to Massive Label Noise. arXiv e-prints , arXiv:1705.10694 arXiv:1705.10694.

Sabatini, S., Davies, J., van Driel, W., et al., 2005. The dwarf low surface brightness galaxy population of the Virgo Cluster - II. Colours and HI line observations. MNRAS 357, 819–833. doi:10.1111/j.1365-2966.2005.08608.x, arXiv:astro-ph/0411401.

Selvaraju, R.R., Cogswell, M., Das, A., et al., 2016. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. arXiv e-prints , arXiv:1610.02391 arXiv:1610.02391.

Shorten, C., Khoshgoftaar, T., 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6, 1–48.

Song, H., Kim, M., Park, D., Lee, J.G., 2020. Learning from Noisy Labels with Deep Neural Networks: A Survey. arXiv e-prints , arXiv:2007.08199 arXiv:2007.08199.

Song, J., Mohr, J.J., Barkhouse, W.A., et al., 2012. A Parameterized Galaxy Catalog Simulator for Testing Cluster Finding, Mass Estimation, and Photometric Redshift Estimation in Optical and Near-infrared Surveys. ApJ 747, 58. doi:10.1088/0004-637X/747/1/58, arXiv:1104.2332.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Suchyta, E., Huff, E.M., Aleksić, J., et al., 2016. No galaxy left behind: accurate measurements with the faintest objects in the Dark Energy Survey. MNRAS 457, 786–808. doi:10.1093/mnras/stv2953, arXiv:1507.08336.

Tang, H., Scaife, A.M.M., Leahy, J.P., 2019. Transfer learning for radio galaxy classification. MNRAS 488, 3358–3375. doi:10.1093/mnras/stz1883, arXiv:1903.11921.

Tanoglidis, D., Drlica-Wagner, A., Wei, K., et al., 2021. Shadows in the Dark: Low-surface-brightness Galaxies Discovered in the Dark Energy Survey. ApJS 252, 18. doi:10.3847/1538-4365/abca89, arXiv:2006.04294.

Tin Kam Ho, 1995. Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, pp. 278–282 vol.1. doi:10.1109/ICDAR.1995.598994.

van Dokkum, P.G., Abraham, R., Merritt, A., et al., 2015. Forty-seven Milky Way-sized, Extremely Diffuse Galaxies in the Coma Cluster. ApJ 798, L45. doi:10.1088/2041-8205/798/2/L45, arXiv:1410.8141.

Venhola, A., Peletier, R., Laurikainen, E., et al., 2017. The Fornax Deep Survey with VST. III. Low surface brightness dwarfs and ultra diffuse galaxies in the center of the Fornax cluster. A&A 608, A142. doi:10.1051/0004-6361/201730696, arXiv:1710.04616.

Venhola, A., Peletier, R., Laurikainen, E., et al., 2018. The Fornax Deep Survey with the VST. IV. A size and magnitude limited catalog of dwarf galaxies in the area of the Fornax cluster. A&A 620, A165. doi:10.1051/0004-6361/201833933, arXiv:1810.00550.

Vilalta, R., 2018. Transfer Learning in Astronomy: A New Machine-Learning Paradigm, in: Journal of Physics Conference Series, p. 052014. doi:10.1088/1742-6596/1085/5/052014, arXiv:1812.10403.

Vilalta, R., Dhar Gupta, K., Boumber, D., Meskhi, M.M., 2019. A General Approach to Domain Adaptation with Applications in Astronomy. PASP 131, 108008. doi:10.1088/1538-3873/aaf1fc, arXiv:1812.08839.

Wang, M., Deng, W., 2018. Deep Visual Domain Adaptation: A Survey. arXiv e-prints , arXiv:1802.03601 arXiv:1802.03601.

Wechsler, R.H., Tinker, J.L., 2018. The Connection Between Galaxies and Their Dark Matter Halos. ARA&A 56, 435–487. doi:10.1146/annurev-astro-081817-051756, arXiv:1804.03097.

Wei, W., Huerta, E.A., Whitmore, B.C., et al., 2020. Deep transfer learning for star cluster classification: I. application to the PHANGS-HST survey. MNRAS 493, 3178–3193. doi:10.1093/mnras/staa325, arXiv:1909.02024.

Weiss, K.R., Khoshgoftaar, T., Wang, D., 2016. A survey of transfer learning. Journal of Big Data 3, 1–40.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? arXiv e-prints , arXiv:1411.1792 arXiv:1411.1792.

Zeiler, M.D., 2012. ADADELTA: An Adaptive Learning Rate Method. arXiv e-prints , arXiv:1212.5701 arXiv:1212.5701.

Zhuang, F., Qi, Z., Duan, K., et al., 2019. A Comprehensive Survey on Transfer Learning. arXiv e-prints , arXiv:1911.02685 arXiv:1911.02685.