FERMILAB-PUB-15-141-AE

arXiv:1504.02936v2 [astro-ph.IM] 14 Apr 2015

Automated Transient Identification in the Dark Energy Survey

D. A. Goldstein,[1,2] C. B. D'Andrea,[3] J. A. Fischer,[4] R. J. Foley,[5,6] R. R. Gupta,[7] R. Kessler,[8,9] A. G. Kim,[2] R. C. Nichol,[3] P. Nugent,[2] A. Papadopoulos,[3] M. Sako,[4] M. Smith,[10] M. Sullivan,[10] R. C. Thomas,[2] W. Wester,[11] R.C. Wolf,[4] F. B. Abdalla,[12] M. Banerji,[13,14] A. Benoit-Lévy,[12] E. Bertin,[15] D. Brooks,[12] A. Carnero Rosell,[16,17] F. J. Castander,[18] L. N. da Costa,[16,17] R. Covarrubias,[19] D. L. DePoy,[20] S. Desai,[21] H. T. Diehl,[11] P. Doel,[12] T. F. Eifler,[4,22] A. Fausti Neto,[16] D. A. Finley,[11] B. Flaugher,[11] P. Fosalba,[18] J. Frieman,[8,11] D. Gerdes,[23] D. Gruen,[24,25] R. A. Gruendl,[5,19] D. James,[26] K. Kuehn,[27] N. Kuropatkin,[11] O. Lahav,[12] T. S. Li,[20] M. A. G. Maia,[16,17] M. Makler,[28] M. March,[4] J. L. Marshall,[20] P. Martini,[29,30] K. W. Merritt,[11] R. Miquel,[31,32] B. Nord,[11] R. Ogando,[16,17] A. A. Plazas,[22,33] A. K. Romer,[34] A. Roodman,[35,36] E. Sanchez,[37] V. Scarpine,[11] M. Schubnell,[23] I. Sevilla-Noarbe,[5,37] R. C. Smith,[26] M. Soares-Santos,[11] F. Sobreira,[11,16] E. Suchyta,[29,38] M. E. C. Swanson,[19] G. Tarle,[23] J. Thaler,[6] A. R. Walker[26]

ABSTRACT

We describe an algorithm for identifying point-source transients and moving objects on reference-subtracted optical images containing artifacts of processing and instrumentation. The algorithm makes use of the supervised machine learning technique known as Random Forest. We present results from its use in the Dark Energy Survey Supernova program (DES-SN), where it was trained using a sample of 898,963 signal and background events generated by the transient detection pipeline. After reprocessing the data collected during the first DES-SN observing season (Sep. 2013 through Feb. 2014) using the algorithm, the number of transient candidates eligible for human scanning decreased by a factor of 13.4, while only 1 percent of the artificial Type Ia supernovae (SNe) injected into search images to monitor survey efficiency were lost, most of which were very faint events. Here we characterize the algorithm's performance in detail, and we discuss how it can inform pipeline design decisions for future time-domain imaging surveys, such as the Large Synoptic Survey Telescope and the Zwicky Transient Facility.

Subject headings: transients – discovery, algorithms – statistical, random forest, machine learning.

[1]Department of Astronomy, University of California, Berkeley, 501 Campbell Hall #3411, Berkeley, CA 94720

[2]Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA

[3]Institute of Cosmology and Gravitation, University of Portsmouth, Dennis Sciama Building, Burnaby Road, Portsmouth, PO1 3FX, UK

[4]Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA

[5]Astronomy Department, University of Illinois at Urbana-Champaign, 1002 W. Green Street, Urbana, IL 61801, USA

[6]Department of Physics, University of Illinois at Urbana-Champaign, 1110 W. Green Street, Urbana, IL 61801, USA

[7]Argonne National Laboratory, 9700 South Cass Avenue, Lemont, IL 60439, USA

[8]Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, USA

[9]Department of Astronomy and Astrophysics, University of Chicago, 5640 South Ellis Avenue, Chicago, IL 60637, USA

[10]School of Physics and Astronomy, University of Southampton, Highfield, Southampton, SO17 1BJ, UK

[11]Fermi National Accelerator Laboratory, P. O. Box 500, Batavia, IL 60510, USA

[12]Department of Physics & Astronomy, University College London, Gower Street, London, WC1E 6BT, UK

[13]Kavli Institute for Cosmology, University of Cam-

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

bridge, Madingley Road, Cambridge CB3 0HA, UK

[14]Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK

[15]Institut d'Astrophysique de Paris, Univ. Pierre et Marie Curie & CNRS UMR7095, F-75014 Paris, France

[16]Laboratório Interinstitucional de e-Astronomia - LIneA, Rua Gal. José Cristino 77, Rio de Janeiro, RJ - 20921-400, Brazil

[17]Observatório Nacional, Rua Gal. José Cristino 77, Rio de Janeiro, RJ - 20921-400, Brazil

[18]Institut de Ciències de l'Espai, IEEC-CSIC, Campus UAB, Facultat de Ciències, Torre C5 par-2, 08193 Bellaterra, Barcelona, Spain

[19]National Center for Supercomputing Applications, 1205 West Clark St., Urbana, IL 61801, USA

[20]George P. and Cynthia Woods Mitchell Institute for Fundamental Physics and Astronomy, and Department of Physics and Astronomy, Texas A&M University, College Station, TX 77843, USA

[21]Department of Physics, Ludwig-Maximilians-Universitaet, Scheinerstr. 1, 81679 Muenchen, Germany

[22]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109, USA

[23]Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA

[24]Max Planck Institute for Extraterrestrial Physics, Giessenbachstrasse, 85748 Garching, Germany

[25]University Observatory Munich, Scheinerstrasse 1, 81679 Munich, Germany

[26]Cerro Tololo Inter-American Observatory, National Optical Astronomy Observatory, Casilla 603, Colina El Pino S/N, La Serena, Chile

[27]Australian Astronomical Observatory, North Ryde, NSW 2113, Australia

[28]ICRA, Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud 150, CEP 22290-180, Rio de Janeiro, RJ, Brazil

[29]Center for Cosmology and Astro-Particle Physics, The Ohio State University, Columbus, OH 43210, USA

[30]Department of Astronomy, The Ohio State University, Columbus, OH 43210, USA

[31]Institut de Física d'Altes Energies, Universitat Autònoma de Barcelona, E-08193 Bellaterra, Barcelona, Spain

[32]Institució Catalana de Recerca i Estudis Avançats, E-08010 Barcelona, Spain

[33]Brookhaven National Laboratory, Bldg 510, Upton, NY 11973, USA

[34]Astronomy Centre, University of Sussex, Falmer, Brighton, BN1 9QH, UK

[35]Kavli Institute for Particle Astrophysics & Cosmology, P. O. Box 2450, Stanford University, Stanford, CA 94305, USA

[36]SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

1. Introduction

To identify scientifically valuable transients or moving objects on the sky, imaging surveys have historically adopted a manual approach, employing humans to visually inspect images for signatures of the events (e.g., Zwicky 1964; Hamuy et al. 1993; Perlmutter et al. 1997; Schmidt et al. 1998; Filippenko et al. 2001; Kaiser et al. 2002; Strolger et al. 2004; Blanc et al. 2004; Astier et al. 2006; Sako et al. 2008; Mainzer et al. 2011; Waszczak et al. 2013). But recent advances in the capabilities of telescopes, detectors, and supercomputers have fueled a dramatic rise in the data production rates of such surveys, straining the ability of their teams to quickly and comprehensively look at images to perform discovery.

For surveys that search for objects on difference images—CCD images that reveal changes in the appearance of a region of the sky between two points in time—this problem of data volume is compounded by the problem of data purity. Difference images are produced by subtracting reference images from single-epoch images in a process that involves point-spread function (PSF) matching and image distortion (see, e.g., Alard & Lupton 1998). In addition to legitimate detections of astrophysical variability, they can contain artifacts of the differencing process, such as poorly subtracted galaxies, and artifacts of the single-epoch images, such as cosmic rays, optical ghosts, star halos, defective pixels, near-field objects, and CCD edge effects. Some examples are presented in Figure 1. These artifacts can vastly outnumber the signatures of scientifically valuable sources on the images, forcing object detection thresholds to be considerably higher than what is to be expected from Gaussian fluctuations.

For time-domain imaging surveys with a spectroscopic follow-up program, these issues of data volume and purity are compounded by time-pressure to produce lists of the most promising targets for follow-up observations before they become too faint to observe or fall outside a window of scientific utility. Ongoing searches for Type Ia supernovae (SNe Ia) out to $z \sim 1$, such as the

[37]Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain

[38]Department of Physics, The Ohio State University, Columbus, OH 43210, USA

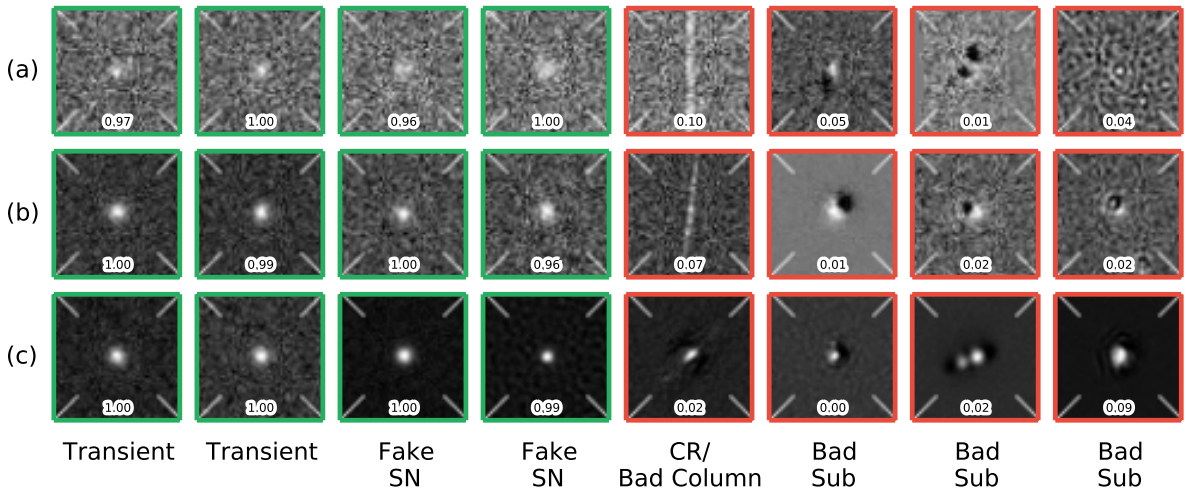

Fig. 1.— Cutouts of DES difference images, roughly 14 arcsec on a side, centered on legitimate (green boxes; left four columns of figure) and spurious (red boxes; right four columns of figure) objects, at a variety of signal-to-noise ratios: (a) S/N $\leq$ 10, (b) 10 $<$ S/N $\leq$ 30, (c) 30 $<$ S/N $\leq$ 100. The cutouts are subclassed to illustrate both the visual diversity of spurious objects and the homogeneity of authentic ones. Objects in the "Transient" columns are real astrophysical transients that subtracted cleanly. Objects in the "Fake SN" columns are fake SNe Ia injected into transient search images to monitor survey efficiency. The column labeled "CR/Bad Column" shows detections of cosmic rays (rows b and c) and a bad column on the CCD detector (row a). The columns labeled "Bad Sub" show non-varying astrophysical sources that did not subtract cleanly; this can result from poor astrometric solutions, shallow templates, or bad observing conditions. The numbers at the bottom of each cutout indicate the score that each detection received from the machine learning algorithm introduced in §3; a score of 1.0 indicates the algorithm is perfectly confident that the detection is not an artifact, while a score of 0.0 indicates the opposite.

Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Kaiser et al. 2002) and the Dark Energy Survey (DES; Flaugher 2005), face all three of these challenges. The DES supernova program (DES-SN; Bernstein et al. 2012), for example, produces up to 170 gigabytes of raw imaging data on a nightly basis. Visual examination of sources extracted from the resulting difference images using `SExtractor` (Bertin & Arnouts 1996) revealed that ∼93 percent are artifacts, even after selection cuts (Kessler et al. 2015, in preparation). Additionally, the survey has a science-critical spectroscopic follow-up program for which it must routinely select the ∼10 most promising transient candidates from hundreds of possibilities, most of which are artifacts. This program is crucial to survey science as it allows DES to confirm transient candidates as SNe, train and optimize its photometric SN typing algorithms (e.g., `PSNID`; Sako et al. 2011, `NNN`; Karpenka, Feroz, & Hobson 2013), and investigate interesting non-SN transients. To prepare a list of objects eligible for consideration for spectroscopic follow-up observations, members of DES-SN scanned nearly 1 million objects extracted from difference images during the survey's first observing season, the numerical equivalent of nearly a week of uninterrupted scanning time, assuming scanning one object takes half a second.

For DES to meet its discovery goals, more efficient techniques for artifact rejection on difference images are needed. Efforts to "crowdsource" similar large-scale classification problems have been successful at scaling with growing data rates; websites such as `Zooniverse.org` have accumulated over one million users to tackle a variety of astrophysical classification problems, including the classification of transient candidates from the Palomar Transient Factory (PTF; Smith et al. 2011). However, for DES to optimize classification accuracy and generate reproducible classification decisions, automated techniques are required.

To reduce the number of spurious candidates considered for spectroscopic follow-up, many surveys impose selection requirements on quantities that can be directly and automatically computed from the raw imaging data. Making hard selection cuts of this kind has been shown to be a suboptimal technique for artifact rejection in difference imaging. Although such cuts are automatic and easy to interpret, they do not naturally handle correlations between features, and they are an inefficient way to select a subset of the high-dimensional feature space as the number of dimensions grows large (Bailey et al. 2007).

In contrast to selection cuts, machine learning (ML) classification techniques provide a flexible solution to the problem of artifact rejection in difference imaging. In general, these techniques attempt to infer a precise mapping between numeric features that describe characteristics of observed data, and the classes or labels assigned to those data, using a training set of feature-class pairs. ML classification algorithms that generate decision rules using labeled data—data whose class membership has already been definitively established—are called "supervised" algorithms. After generating a decision rule, supervised ML classifiers can be used to predict the classes of unlabeled data instances. For a review of supervised ML classification in astronomy, see, e.g. Ivezić et al. (2013). For an introduction to the statistical underpinnings of supervised ML classification techniques, see Willsky, Wornell, & Shapiro (2003).

Such classifiers address many of the shortcomings of scanning and selection cuts. ML algorithms' decisions are automatic, reproducible, and fast enough to process streaming data in real-time. Their biases can be systematically and quantitatively studied, and, most importantly, given adequate computing resources, they remain fast and consistent in the face of increasing data production rates. As more data are collected, ML methods can continue to refine their knowledge about a data set (see §5.1), thereby improving their predictive performance on future data. Supervised ML classification techniques are currently used in a variety of astronomical contexts, including time-series analysis, such as the classification of variable stars (Richards et al. 2011) and SNe (Karpenka, Feroz, & Hobson 2013) from light curves, and image analysis, such as the typing of galaxies (Banerji et al. 2010), and discovery of trans-Neptunian objects (Gerdes et al. 2015, in preparation) on images. Although their input data types differ, light curve shape and image-based ML classification frameworks are quite similar: both operate on tabular numeric classifica-

tion features computed from raw input data (see §3.2.2).

The use of supervised machine learning classification techniques for artifact rejection in difference imaging was pioneered by Bailey et al. (2007) for the Nearby Supernova Factory (Aldering et al. 2002) using imaging data from the Near-Earth Asteroid Tracking program[1] and the Palomar-QUEST Consortium, using the 112-CCD QUEST-II camera (Baltay et al. 2007). They compared the performance of three supervised classification techniques—a Support Vector Machine, a Random Forest, and an ensemble of boosted decision trees—in separating a combination of real and fake detections of SNe from background events. They found that boosted decision trees constructed from a library of astrophysical domain features (magnitude, FWHM, distance to the nearest object in the reference co-add, measures of roundness, etc.) provided the best overall performance.

Bloom et al. (2012) built on the methodology of Bailey et al. (2007) by developing a highly accurate Random Forest framework for classifying detections of variability extracted from PTF difference images. Brink et al. (2013) made improvements to the classifier of Bloom et al. (2012), setting an unbroken benchmark for best overall performance on the PTF data set, using the technique of recursive feature elimination to optimize their classifier. Recently, du Buisson et al. (2014) published a systematic comparison of several classification algorithms using features based on Principal Component Analysis (PCA) extracted from Sloan Digital Sky Survey-II SN survey difference images.

In this article, we describe `autoScan`, a computer program developed for this purpose in DES-SN. Our main objective is to report the methodology that DES-SN adopted to construct an effective supervised classifier, with an eye toward informing the design of similar frameworks for future time domain surveys such as the Large Synoptic Survey Telescope (LSST; LSST Science Collaboration 2009) and the Zwicky Transient Facility (ZTF; Smith et al. 2014). We extend the work of previous authors to a newer, larger data set, showing how greater selection efficiency can be achieved by increasing training set size, using generative models for training data, and implementing new classification features.

The structure of the paper is as follows. In §2, we provide an overview of DES and the DES-SN transient detection pipeline. In §3, we describe the development of `autoScan`. In §4, we present metrics for evaluating the code's performance and review its performance on a realistic classification task. In §5, we discuss lessons learned and areas of future development that can inform the design of similar frameworks for future surveys.

2. The Dark Energy Survey and Transient Detection Pipeline

In this section, we introduce DES and the DES-SN transient detection pipeline ("`DiffImg`"; Kessler et al. 2015, in preparation), which produced the data used to train and validate `autoScan`. DES is a Stage III ground-based dark energy experiment designed to provide the tightest constraints to date on the dark energy equation of state parameter using observations of the four most powerful probes of dark energy suggested by the Dark Energy Task Force (DETF; Albrecht et al. 2006): SNe Ia, galaxy clusters, baryon acoustic oscillations, and weak gravitational lensing. DES consists of two interleaved imaging surveys: a wide-area survey that covers 5,000 deg^2 of the south Galactic cap in 5 filters ($grizY$), and DES-SN, a time-domain transient survey that covers 10 (8 "shallow" and 2 "deep") 3 deg^2 fields in the XMM-LSS, ELAIS-S, CDFS, and Stripe-82 regions of the sky, in four filters ($griz$). The survey's main instrument, the Dark Energy Camera (DECam; Diehl et al. 2012; Flaugher et al. 2012; Flaugher et al. 2015, submitted), is a 570-megapixel 3 deg^2 imager with 62 fully depleted, red-sensitive CCDs. It is mounted at the prime focus of the Victor M. Blanco 4m telescope at the Cerro Tololo Inter-American Observatory (CTIO). DES conducted "science verification" (SV) commissioning observations from November 2012 until February 2013, and it began science operations in August 2013 that will continue until at least 2018 (Diehl et al. 2014). The data used in this article are from the first season of DES science operations ("Y1"; Aug. 2013—Feb. 2014).

A schematic of the pipeline that DES-SN employs to discover transients is presented in Figure

[1] `http://neat.jpl.nasa.gov`.

2. Transient survey "science images" are single-epoch CCD images from the DES-SN fields. After the image subtraction step, sources are extracted using `SExtractor`. Sources that pass the cuts described in the Object section of Table 1 are referred to as "detections." A "raw candidate" is defined when two or more detections match to within 1". A raw candidate is promoted to a "science candidate" when it passes the `NUMEPOCHS` requirement in Table 1. This selection requirement was imposed to reject Solar System objects, such as main belt asteroids and Kuiper belt objects, which move substantially on images from night to night. Science candidates are eligible for visual examination and spectroscopic follow-up observations. During the observing season, science candidates are routinely photometered, fit with multi-band SN light curve models, visually inspected, and slated for spectroscopic follow-up.

3. Classifier Development

In this section, we describe the development of `autoScan`. We present the classifier's training data set (§3.1), its classification feature set (§3.2), and the selection (§3.3), properties (§3.4), and optimization (§3.5) of its core classification algorithm.

3.1. Training Data

To make probabilistic statements about the class membership of new data, supervised ML classifiers must be trained or fit to existing data whose true class labels are already known. Each data instance is described by numeric classification "features" (see §3.2.2); an effective training data set must approximate the joint feature distributions of all classes considered. Objects extracted from difference images can belong to one of two classes: "Artifacts," or "Non-Artifacts." Examples of each class must be present in the training set. Failing to include data from certain regions of feature space can corrode the predictive performance of the classifier in those regions, introducing bias into the search that can systematically degrade survey efficiency (Richards et al. 2012). Because the training set compilation described here took place during the beginning of Y1, it was complicated by a lack of available visually scanned "non-artifact" sources.

Fortunately, labeling data does not necessarily require humans to visually inspect images. Bloom et al. (2012) discuss a variety of methods for labeling detections of variability produced by difference imaging pipelines, including scanning alternatives such as artificial source construction and spectroscopic follow-up. Scanning, spectroscopy, and using fake data each have their respective merits and drawbacks. Scanning is laborious and potentially inaccurate, especially if each data instance is only examined by one scanner, or if scanners are not well trained. However, a large group of scanners can quickly label a number of detections sufficient to create a training set for a machine classifier, and Brink et al. (2013) have shown that the supervised classification algorithm Random Forest, which was ultimately selected for `autoScan`, is insensitive to mislabeled training data up to a contamination level of 10 percent.

Photometric typing (e.g., Sako et al. 2011) can also be useful for labeling detections of transients. However, robust photometric typing requires well-sampled light curves, which in turn require high-cadence photometry of difference image objects over timescales of weeks or months. This requirement is prohibitive for imaging surveys in their early stages. Further, because photometric typing is an integral part of the spectroscopic target selection process, by extension new imaging surveys also have too few detections of spectroscopically confirmed SNe, AGN, or variable stars. Native spectroscopic training samples are therefore impractical sources of training data for new surveys.

Artificial source construction is the fastest method for generating native detections of non-artifact sources in the early stages of a survey. Large numbers of artificial transients ("fakes") can be injected into survey science images, and by construction their associated detections are true positives. Difficulties can arise when the joint feature distributions of fakes selected for the training set do not approximate the joint feature distributions of observed transients in production. In DES-SN, SN Ia fluxes from fake SN Ia light curves are overlaid on images near real galaxies. The fake SN Ia light curves are generated by the `SNANA` simulation (Kessler et al. 2009), and they include true parent populations of stretch and color, a realistic model of intrinsic scatter, a redshift range from 0.1 to 1.4, and a galaxy location proportional to surface brightness. On difference images, detections of

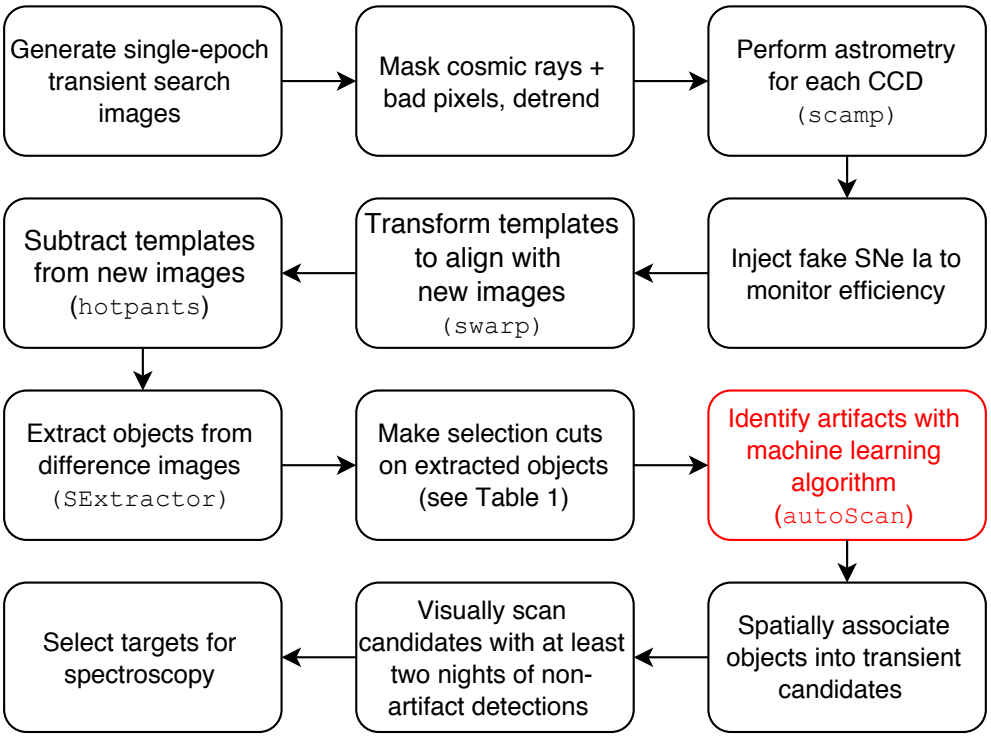

Fig. 2.— Schematic of the DES-SN transient detection pipeline. The magnitudes of fake SNe Ia used to monitor survey efficiency are calibrated using the zero point of the images into which they are injected and generated according to the procedure described in §3.1. The `autoScan` step (red box) occurs after selection cuts are applied to objects extracted from difference images and before objects are spatially associated into raw transient candidates. Codes used at specific steps are indicated in parenthesis.

Table 1

DES-SN object and candidate selection requirements.

Set	Feature	Lower Limit	Upper Limit	Description
Object	MAG	$\cdots$	30.0	Magnitude from SExtractor.
	A_IMAGE	$\cdots$	1.5 pix.	Length of semi-major axis from SExtractor.
	SPREAD_MODEL	$\cdots$	$3\sigma_S + 1.0$	Star-galaxy separation output parameter from SExtractor. σ_S is the estimated SPREAD_MODEL uncertainty.
	CHISQ	$\cdots$	10^4	χ^2 from PSF-fit to 35×35 pixel cutout around object in difference image.
	SNR	3.5	$\cdots$	Flux from a PSF-model fit to a 35×35 pixel cutout around the object divided by the uncertainty from the fit.
	VETOMAG[a]	21.0	$\cdots$	Magnitude from SExtractor for use in veto catalog check.
	VETOTOL[a]	Magnitude-dependent	$\cdots$	Separation from nearest object in veto catalog of bright stars.
	DIPOLE6	$\cdots$	2	N_{pix} in 35×35 pixel object-centered cutout at least 6σ below 0.
	DIPOLE4	$\cdots$	20	N_{pix} in 35×35 pixel object-centered cutout at least 4σ below 0.
	DIPOLE2	$\cdots$	200	N_{pix} in 35×35 pixel object-centered cutout at least 2σ below 0.
Candidate	NUMEPOCHS	2	$\cdots$	Number of distinct nights that the candidate is detected.

[a]The difference imaging pipeline is expected to produce false positives near bright or variable stars, thus all difference image objects are checked against a "veto" catalog of known bright and variable stars and are rejected if they are brighter than 21st magnitude and within a magnitude-dependent radius of a veto catalog source. Thus only one of VETOMAG and VETOTOL must be satisfied for an object to be selected.

overlaid fakes are visually indistinguishable from real point-source transients and Solar System objects moving slowly enough not to streak. All fake SN Ia light curves are generated and stored prior to the start of the survey. The overlay procedure is part of the difference imaging pipeline, where the SN Ia flux added to the image is scaled by the zero point, spread over nearby pixels using a model of the PSF, and fluctuated by random Poisson noise. These fakes are used to monitor the single-epoch transient detection efficiency, as well as the candidate efficiency in which detections on two distinct nights are required. On average, six detections of fake SNe are overlaid on each single-epoch CCD-image.

The final `autoScan` training set contained detections of visually scanned artifacts and artificial sources only. We did not include detections of photometrically typed transients to minimize the contamination of the "Non-Artifact" class with false positives. Bailey et al. (2007) also used a training set in which the "Non-Artifact" class consisted largely of artificial sources.

With 898,963 training instances in total, the `autoScan` training set is the largest used for difference image artifact rejection in production. It was split roughly evenly between "real" and "artifact" labeled instances—454,092 were simulated SNe Ia injected onto host galaxies, while the remaining 444,871 detections were human-scanned artifacts. Compiling a set of artifacts to train `autoScan` was accomplished by taking a random sample of the objects that had been scanned as artifacts by humans during an early processing of DES Y1 data with a pared-down version of the difference imaging pipeline presented in Figure 2.

3.2. Features and Processing

The supervised learning algorithms we consider in this analysis are nonlinear functions that map points representing individual detections in feature space to points in a space of object classes or class probabilities. The second design choice in developing `autoScan` is therefore to define a suitable feature space in which to represent the data instances we wish to use for training, validation, and prediction. In this section, we describe the classification features that we computed from the raw output of the difference imaging pipeline, as well as the steps used to pre- and post-process these features.

3.2.1. Data Preprocessing

The primary data sources for `autoScan` features are 51×51 pixel object-centered search, template, and difference image cutouts. The template and difference image cutouts are sky-subtracted. The search image cutout is sky-subtracted if and only if it does not originate from a coadded exposure, though this is irrelevant for what follows as no features are directly computed from search image pixel values. Photometric measurements, `SExtractor` output parameters, and other data sources are also used. Each cutout associated with a detection is compressed to 25×25 pixels. The seeing for each search image is usually no less than 1 arcsec, while the DECam pixel scale lies between 0.262 and 0.264 arcsec depending on the location on the focal plane, so little information is lost during compression. Although some artifacts are sharper than the seeing, we found that using compressed cutouts to compute some features resulted in better performance.

Consider a search, template, or difference image cutout associated with a single detection. Let the matrix element $\mathbf{I}_{x,y}$ of the 51×51 matrix $\mathbf{I}$ represent the flux-value of the pixel at location x, y on the cutout. We adopt the convention of zero-based indexing and the convention that element (0, 0) corresponds to the pixel at the top left-hand corner of the cutout. Let the matrix element $C_{x,y}$ of the 25×25 matrix C represent the flux-value of the pixel at location x, y on the compressed cutout. Then C is defined element-wise from $\mathbf{I}$ via

$$C_{x,y} = \frac{1}{N_u} \sum_{i=0}^{1} \sum_{j=0}^{1} \mathbf{I}_{2x+i,2y+j}, \qquad (1)$$

where N_u is the number of unmasked pixels in the sum. Masked pixels are excluded from the sum. Only when all four terms in the sum represent masked pixels is the corresponding pixel masked in C. Note that matrix elements from the right-hand column and last row of $\mathbf{I}$ never appear in Equation 1.

To ensure that the pixel flux-values across cutouts are comparable, we rescale the pixel values

of each compressed cutout via "';

$$R_{x,y} = \frac{C_{x,y} - \mathrm{med}(C)}{\hat{\sigma}}, \tag{2}$$

where the matrix element $R_{x,y}$ of the 25×25 matrix R represents the flux-value of the pixel at location x, y on the compressed, rescaled cutout, and $\hat{\sigma}$ is a consistent estimator of the standard deviation of C. We take the median absolute deviation as a consistent estimator of the standard deviation, according to

$$\hat{\sigma} = \frac{\mathrm{med}(|C - \mathrm{med}(C)|)}{\Phi^{-1}\left(\frac{3}{4}\right)} \tag{3}$$

where $1/\Phi^{-1}(3/4) \approx 1.4826$ is the reciprocal of the inverse cumulative distribution for the standard normal distribution evaluated at 3/4. This is done to ensure that the effects of defective pixels and cosmic rays nearly perpendicular to the focal plane are suppressed. We therefore have the following closed-form expression for the matrix element $R_{x,y}$,

$$R_{x,y} \approx \frac{1}{1.4826}\left[\frac{C_{x,y} - \mathrm{med}(C)}{\mathrm{med}(|C - \mathrm{med}(C)|)}\right]. \tag{4}$$

The rescaling expresses the value of each pixel on the compressed cutout as the number of standard deviations above the median. Masked pixels are excluded from the computation of the median in Equation 4.

Finally, an additional rescaling from Brink et al. (2013) is defined according to

$$\mathbf{B}_{x,y} = \frac{\mathbf{I}_{x,y} - \mathrm{med}(\mathbf{I})}{\max(|\mathbf{I}|)} \tag{5}$$

The size of $\mathbf{B}$ is 51×51. We found that using $\mathbf{B}$ instead of R or $\mathbf{I}$ to compute certain features resulted in better classifier performance. Masked pixels are excluded from the computation of the median in Equation 5.

3.2.2. Feature Library

Two feature libraries were investigated. The first was primarily "pixel-based." For a given object, each matrix element of the rescaled, compressed search, template, and difference cutouts was used as a feature. The CCD ID number of each detection was also used, as DECam has 62 CCDs with specific artifacts (such as bad columns and hot pixels) as well as effects that are reproducible on the same CCD depending on which field is observed (such as bright stars). The signal-to-noise ratio of each detection was also used as a feature. The merits of this feature space include relatively straightforward implementation and computational efficiency. A production version of this pixel-based classifier was implemented in the DES-SN transient detection pipeline at the beginning of Y1. In production, it became apparent that the 1,877-dimensional[2] feature space was dominated by uninformative features, and that better false positive control could be achieved with a more compact feature set.

We pursued an alternative feature space going forward, instead using 38 high-level metrics to characterize detections of variability. A subset of the features are based on analogs from Bloom et al. (2012) and Brink et al. (2013). In this section, we describe the features that are new. We present an at-a-glance view of the entire `autoScan` feature library in Table 2. Histograms and contours for the three most important features in the final `autoScan` model (see §3.4) appear in Figure 3.

[2] 625 pixels on a 25×25 pixel cutout $\times$ 3 cutouts per detection + 2 non-pixel features (SNR, CCD) = 1,877.

Table 2

autoScan's feature library.

Feature Name	Importance	Source	Description
r_aper_psf	0.148	New	The flux in a 5-pixel circular aperture centered on the object on the $\mathbf{I}^d$ cutout plus the flux from a 35×35-pixel PSF model-fit to the object on the $\mathbf{I}^d$ cutout, all divided by the PSF model-fit flux.
magdiff	0.094	B12	If a source is found within 5" of the location of the object in the galaxy coadd catalog, the difference between mag and the magnitude of the nearby source. Else, the difference between mag and the limiting magnitude of the parent image from which the $\mathbf{I}^d$ cutout was generated.
spread_model	0.066	New	SPREAD_MODEL output parameter from SExtractor on $\mathbf{I}^d$.
n2sig5	0.055	B12	Number of matrix elements in a 7×7 element block centered on the detection on R^d with values less than -2.
n3sig5	0.053	B12	Number of matrix elements in a 7×7 element block centered on the detection on R^d with values less than -3.
n2sig3	0.047	B12	Number of matrix elements in a 5×5 element block centered on the detection on R^d with values less than -2.
flux_ratio	0.037	B12	Ratio of the flux in a 5-pixel circular aperture centered on the location of the detection on $\mathbf{I}^d$ to the absolute value of the flux in a 5-pixel circular at the same location on $\mathbf{I}^t$.
n3sig3	0.034	B12	Number of matrix elements in a 5×5 element block centered on the detection on R^d with values less than -3.
mag_ref_err	0.030	B12	Uncertainty on mag_ref, if it exists. Else imputed.
snr	0.029	B12	The flux from a 35×35-pixel PSF model-fit to the object on $\mathbf{I}^d$ divided by the uncertainty from the fit.
colmeds	0.028	New	The maximum of the median pixel values of each column on $\mathbf{B}^d$.
nn_dist_renorm	0.027	B12	The distance from the detection to the nearest source in the galaxy coadd catalog, if one exists within 5". Else imputed.
ellipticity	0.027	B12	The ellipticity of the detection on $\mathbf{I}^d$ using a_image and b_image from SExtractor.
amp	0.027	B13	Amplitude of fit that produced gauss.
scale	0.024	B13	Scale parameter of fit that produced gauss.
b_image	0.024	B12	Semi-minor axis of object from SExtractor on $\mathbf{I}^d$.
mag_ref	0.022	B12	The magnitude of the nearest source in the galaxy coadd catalog, if one exists within 5" of the detection on $\mathbf{I}^d$. Else imputed.
diffsum	0.021	New	The sum of the matrix elements in a 5×5 element box centered on the detection location on R^d.
mag	0.020	B12	The magnitude of the object from SExtractor on $\mathbf{I}^d$.
a_ref	0.019	B12	Semi-major axis of the nearest source in the galaxy coadd catalog, if one exists within 5". Else imputed.

3.2.3. New Features

In this section we present new features developed for `autoScan`. Let the superscripts s, t, and d on matrices defined in the previous section denote search, template, and difference images, respectively. The feature `r_aper_psf` is designed to identify objects whose pixel intensity distributions deviate from the PSF using photometry on difference images. Let $F_{aper,I}$ be the flux from a five-pixel circular aperture centered on the location of a detection on the uncompressed difference image. Let $F_{PSF,I}$ be the flux computed by fitting a PSF-model to a 35×35 pixel cutout centered on the location of the detection on the uncompressed difference image. Then `r_aper_psf` is given by

$$\texttt{r_aper_psf} = \frac{F_{aper,I} + F_{PSF,I}}{F_{PSF,I}}. \tag{6}$$

Let $a \in \{2, 3\}$, $b \in \{3, 5\}$. The four features `n`a`sig`b`shift` represent the difference between the number of pixels with flux values greater than or equal to a in $(b+2)\times(b+2)$ element blocks centered on the detection position in R^d and R^t. These features coarsely describe changes in the morphology of the source between the template and search images.

The feature `diffsum` is the sum of the matrix elements in a 5×5 element ($2.8 \times 2.8\,\mathrm{arcsec}^2$) box centered on the detection location in R^d. It is given by

$$\texttt{diffsum} = \sum_{i=-2}^{2} \sum_{j=-2}^{2} R^d_{x_c+i, y_c+j}, \tag{7}$$

where x_c, y_c is the location of the central element on R^d. It gives a coarse measurement of the significance of the detection.

`bandnum` is a numeric representation of the filter in which the object was detected on the search image. This feature enables `autoScan` to identify band-specific patterns.

`numneg` is intended to assess object-smoothness by returning the number of negative elements in a 7×7 pixel box centered on the object in R^d, exposing objects riddled with negative pixels or objects that have a significant number of pixels below $\mathrm{med}(R^d)$. Used in concert with the S/N, `numneg` can help identify high-S/N objects with spatial pixel intensity distributions that do not vary smoothly, useful in rejecting hot pixels and cosmic rays.

`lacosmic` was designed to identify cosmic rays and other objects with spatial pixel intensity distributions that do not vary smoothly, and is based loosely on the methodology that van Dokkum (2001) uses to identify cosmic rays on arbitrary sky survey images. Derive the "fine structure" image $\mathbf{F}$ from $\mathbf{B}^d$ according to

$$\mathbf{F} = (M_3 * \mathbf{B}^d) - ([M_3 * \mathbf{B}^d] * M_7), \tag{8}$$

where M_n is an $n \times n$ median filter. Then

$$\texttt{lacosmic} = \max(\mathbf{B}^d)/\max(\mathbf{F}). \tag{9}$$

Relatively speaking, this statistic should be large for objects that do not vary smoothly, and small for objects that approximate a PSF. The reader is referred to Figure 3 of van Dokkum (2001) for visual examples.

Bad columns and CCD edge effects that appear as fuzzy vertical streaks near highly masked regions of difference images are common types of artifacts. Because they share a number of visual similarities, we designed a single feature, `colmeds`, to identify them:

$$\begin{aligned}\texttt{colmeds} = \max(\{&\mathrm{med}(\mathrm{transpose}(\mathbf{B}^d)_i);\\ &i \in \{0 \ldots N_{col} - 1\}\}),\end{aligned} \tag{10}$$

where N_{col} is the number of columns in $\mathbf{B}^d$. This feature operates on the principle that the median of a column in $\mathbf{B}^d$ should be comparable to the background if the cutout is centered on a PSF, because, in general, even the column in which the PSF is at its greatest spatial extent in $\mathbf{B}^d$ should still contain more background pixels than source pixels. However, for vertically oriented artifacts that occupy entire columns on $\mathbf{B}^d$, this does not necessarily hold. Since these artifacts frequently appear near masked regions of images, we define `maskfrac` as the percentage of $\mathbf{I}^d$ that is masked.

The feature `spread_model` (Desai et al. 2012; Bouy et al. 2013) is a `SExtractor` star/galaxy separation output parameter computed on the $\mathbf{I}^d$ cutout. It is a normalized simplified linear discriminant between the best fitting local PSF model and a slightly more extended model made from the same PSF convolved with a circular exponential disk model.

Table 2—*Continued*

Feature Name	Importance	Source	Description
`n3sig3shift`	0.019	New	The number of matrix elements with values greater than or equal to 3 in the central 5×5 element block of R^d minus the number of matrix elements with values greater than or equal to 3 in the central 5×5 element block of R^t.
`n3sig5shift`	0.018	New	The number of matrix elements with values greater than or equal to 3 in the central 7×7 element block of R^d minus the number of matrix elements with values greater than or equal to 3 in the central 7×7 element block of R^t
`n2sig3shift`	0.014	New	The number of matrix elements with values greater than or equal to 2 in the central 5×5 element block of R^d minus the number of matrix elements with values greater than or equal to 2 in the central 5×5 element block of R^t.
`b_ref`	0.012	B12	Semi-minor axis of the nearest source in the galaxy coadd catalog, if one exists within 5". Else imputed.
`gauss`	0.012	B13	χ^2 from fitting a spherical, 2D Gaussian to a 15×15 pixel cutout around the detection on $\mathbf{B}^d$.
`n2sig5shift`	0.012	New	The number of matrix elements with values greater than or equal to 2 in the central 7×7 element block of R^d minus the number of matrix elements with values greater than or equal to 2 in the central 7×7 element block of R^t.
`mag_from_limit`	0.010	B12	Limiting magnitude of the parent image from which the I^d cutout was generated minus `mag`.
`a_image`	0.009	B12	Semi-major axis of object on $\mathbf{I}^d$ from `SExtractor`.
`min_dist_to_edge`	0.009	B12	Distance in pixels to the nearest edge of the detector array on the parent image from which the I^d cutout was generated.
`ccdid`	0.008	B13	The numerical ID of the CCD on which the detection was registered.
`flags`	0.008	B12	Numerical representation of `SExtractor` extraction flags on $\mathbf{I}^d$.
`numneg`	0.007	New	The number of negative matrix elements in a 7×7 element box centered on the detection in R^d.
`l1`	0.006	B13	$\mathrm{sign}(\sum \mathbf{B}^d) \times \sum |\mathbf{B}^d| / |\sum \mathbf{B}^d|$
`lacosmic`	0.006	New	$\max(\mathbf{B}^d)/\max(F)$, where F is the `LACosmic` (van Dokkum 2001) "fine structure" image computed on $\mathbf{B}^d$.
`spreaderr_model`	0.006	New	Uncertainty on `spread_model`.
`maglim`	0.005	B12	True if there is no nearby galaxy coadd source, false otherwise.
`bandnum`	0.004	New	Numerical representation of image filter.
`maskfrac`	0.003	New	The fraction of $\mathbf{I}^d$ that is masked.

Note.—Source column indicates the reference in which the feature was first published. B13 indicates the feature first appeared in Brink et al. (2013); B12 indicates the feature first appeared in Bloom et al. (2012), and New indicates the feature is new in this work. See §3.3 for an explanation of how feature importances are computed. Imputation refers to the procedure described in §3.2.4.

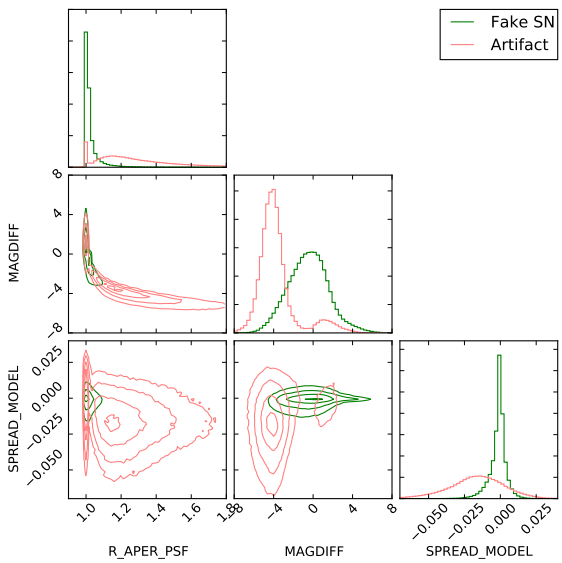

Fig. 3.— Contours for the three most important features in the `autoScan` Random Forest model, computed using the feature importance evaluation scheme described in §3.4. The importances of `r_aper_psf`, `magdiff`, and `spread_model` were 0.148, 0.094, and 0.066, respectively. The contours show that the relationships between the features are highly nonlinear and better suited to machine learning techniques than hard selection cuts.

3.2.4. Data Postprocessing

When there is not a source in the galaxy coadd catalog within 5 arcsec of an object detected on a difference image, certain classification features cannot be computed for the object (see Table 2). If the feature of an object cannot be computed, it is assigned the mean value of that feature from the training set.

3.3. Classification Algorithm Selection

After we settled on an initial library of classification features, we compared three well-known ML classification algorithms: a Random Forest (Breiman 2001), a Support Vector Machine (SVM; Vapnik 1995), and an AdaBoost decision tree classifier (Zhu et al. 2009). We used `scikit-learn` (Pedregosa et al. 2012), an open source Python package for machine learning, to instantiate examples of each model with standard settings. We performed a three-fold cross-validated comparison using a randomly selected 100,000-detection subset of the training set described in §3.1. The subset was used to avoid long training times for the SVM. For a description of cross validation and the metrics used to evaluate each model, see §4 and §4.2. The results appear in Figure 4. We found that the performance of all three models was comparable, but that the Random Forest outperformed the other models by a small margin. We incorporated the Random Forest model into `autoScan`.

Random Forests are collections of decision trees, or cascading sequences of feature-space unit tests, that are constructed from labeled training data. For an introduction to decision trees, see Breiman et al. (1984). A simple schematic of a decision tree appears in Figure 5. Random Forests can be used for predictive classification or regression. During the construction of a supervised Random Forest classifier, trees in the forest are trained individually. To construct a single tree, the training algorithm first chooses a bootstrapped sample of the training data. The algorithm then attempts to recursively define a series of binary splits on the features of the training data that optimally separate the training data into their constituent classes. During the construction of each node, a random subsample of features with a user-specified size is selected with replacement. A fine grid of splits on each feature

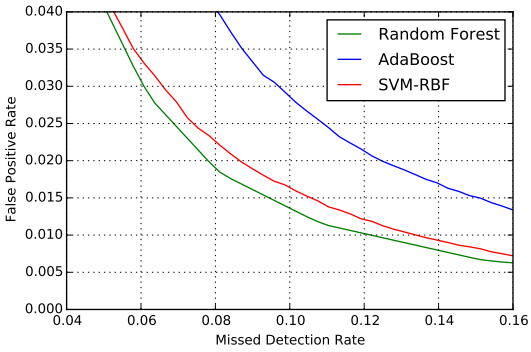

Fig. 4.— Initial comparison of the performance of a Random Forest, a Support Vector Machine with a radial basis function kernel, and an AdaBoost Decision Tree classifier on the DES-SN artifact/non-artifact classification task. Each classifier was trained on a randomly selected 67% of the detections from a 100,000-detection subset of the training set, then tested on the remaining 33%. This process was repeated three times until every detection in the subset was used in the testing set once. The curves above represent the mean of each iteration. The closer a curve is to the origin, the better the classifier. The unoptimized Random Forest outperformed the other two methods, and was selected.

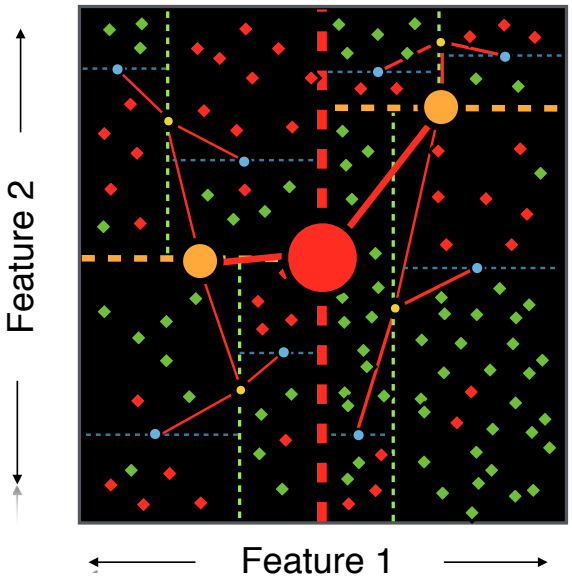

Fig. 5.— Schematic diagram of a two-class classification tree in a two-dimensional feature space. Tree nodes are represented by circles, unit tests by dashed lines, training data by diamonds, classes by diamond colors, and leaf nodes by rectangular regions enclosed by dashed lines. Unseen data instances begin at the root node, then go along red branches to progressively bluer tree nodes until they reach a leaf node. The predicted probability that an unseen data instance is a member of class i is given by the number of instances of class i at the leaf node, divided by the total number of instances at the leaf node.

is then defined, and the split that maximizes the increase in the purity of the incident training data is chosen for the node.

Two popular metrics for sample-purity are the Gini coefficient (Gini 1921) and the Shannon entropy (Shannon 1948). Define the purity of a sample of difference image objects to be[3]

$$P = \frac{N_{NA}}{N_A + N_{NA}}, \tag{11}$$

where N_{NA} is the number of non-artifact objects in the sample, and N_A is the number of artifacts in the sample. Note that $P = 1$ for a sample composed entirely of artifacts, $P = 0$ for a sample composed entirely of non-artifacts, and $P(1-P) = 0$ for a sample composed entirely of either artifacts or non-artifacts. Then the Gini coefficient is

$$\text{Gini} = P(1-P)(N_A + N_{NA}). \tag{12}$$

A tree with a Gini objective function seeks at each node to minimize the quantity

$$\text{Gini}_{\text{lc}} + \text{Gini}_{\text{rc}}, \tag{13}$$

where Gini_{lc} is the Gini coefficient of the data incident on the node's left child, and Gini_{rc} is the Gini coefficient of the data incident on the node's right child. If $\text{Gini}_{\text{lc}} + \text{Gini}_{\text{rc}} > \text{Gini}$, then no split is performed and the node is declared a terminal node. The process proceeds identically if another metric is used, such as the Shannon entropy, the most common alternative. The Shannon entropy S of a sample of difference image objects is given by

$$S = -p_{NA}\log_2(p_{NA}) - p_A\log_2(p_A), \tag{14}$$

where p_{NA} is the proportion of non-artifact objects in the sample, and p_A is the proportion of artifacts in the sample.

Nodes are generated in this fashion until a maximum depth or a user-specified measure of node purity is achieved. The number of trees to grow in the forest is left as a free parameter to be set by the user. Training a single Random Forest using the entire ~900,000 object training sample with the hyperparameters selected from the grid search described in Table 3 took ~4.5 minutes when the construction of the trees was distributed across 60 1.6GHz AMD Opteron 6262 HE processors.

Random Forests treat the classes of unseen objects as unknown parameters that are described probabilistically. An object to be classified descends each tree in the forest, beginning at the root nodes. Once a data point arrives at a terminal node, the tree returns the fraction of the training instances that reached that node that were labeled "non-artifact." The output of the trained `autoScan` Random Forest model on a single input data instance is the average of the outputs of each tree, representing the probability that the object is not an artifact, henceforth the "`autoScan` score" or "ML score." Ultimately, a score of 0.5 was adopted as the cut τ to separate real detections of astrophysical variability from artifacts in the DES-SN data; see §4.4 for details. Class prediction for 200,000 unseen data instances took 9.5s on a single 1.6GHz AMD Opteron 6262 HE processor.

3.4. Feature Importances

Numeric importances can be assigned to the features in a trained forest based on the amount of information they provided during training (Breiman et al. 1984). For each tree T in the forest, a tree-specific importance for feature i is computed according to

$$\zeta_{i,T} = \sum_{n \in T} N(n) B_i(n) \left[m(n) - m_{ch}(n) \right], \tag{15}$$

where n is an index over nodes in T, $N(n)$ is the number of training data points incident on node n, $B_i(n)$ is 1 if node n splits on feature i and 0 otherwise, $m(n)$ is the value of the objective function (usually the Gini coefficient or the Shannon entropy, see §3.3) applied to the the training data incident on node n, and $m_{ch}(n)$ is the sum of the values of the objective function applied to the node's left and right children. The global importance of feature i is the average of the tree-specific importances:

$$I_i = \frac{1}{N_T} \sum_T \zeta_{i,T}, \tag{16}$$

where N_T is the number of trees in the forest. In this article, importances are normalized to sum to

[3]Some authors define $P = \frac{\sum_{NA} w_i}{\sum_{NA} w_i + \sum_A w_i}$, where w_i is the weight of instance i, $\sum_A$ is a sum over artifact events, and $\sum_{NA}$ is a sum over non-artifact events. This renders the definition of the Gini coefficient in Equation 12 as $\text{Gini} = P(1-P)\sum_i w_i$.

unity.

3.5. Optimization

The construction of a Random Forest is governed by a number of free parameters called hyperparameters. The hyperparameters of the Random Forest implementation used in this work are `n_estimators`, the number of decision trees in the forest, `criterion`, the function that measures the quality of a proposed split at a given tree node, `max_features`, the number of features to randomly select when looking for the best split at a given tree node, `max_depth`, the maximum depth of a tree, and `min_samples_split`, the minimum number of samples required to split an internal node.

We performed a 3-fold cross-validated (see §4.2) grid search over the space of Random Forest hyperparameters described in Table 3. A total of 1,884 trainings were performed. The best classifier had 100 trees, used the Shannon entropy objective function, chose 6 features for each split, required at least 3 samples to split a node, and had unlimited depth, and it was incorporated into the code. Recursive feature elimination (Brink et al. 2013) was explored to improve the performance of the classifier, but we found that it provided no statistically significant performance improvement.

4. Performance

In this section, we describe performance of `autoScan` on a realistic classification task and the effect of the code on the DES-SN transient candidate scanning load. Performance statistics for the classification task were measured using production Y1 data, whereas candidate-level effects were measured using a complete reprocessing of Y1 data using an updated difference imaging pipeline. The reprocessed detection pool differed significantly from its production counterpart, providing a out-of-sample data set for benchmarking the effects of the code on the scanning load.[4]

[4] Although the re-processing of data through the difference imaging pipeline from the raw images is not useful for getting spectra of live transients, it is quite useful for acquiring host-galaxy targets for previously missed transients and is therefore performed regularly as pipeline improvements are made.

4.1. Performance Metrics

The performance of a classifier on an n-class task is completely summarized by the corresponding $n \times n$ confusion matrix $\mathbb{E}$, also known as a contingency table or error matrix. The matrix element $\mathbb{E}_{ij}$ represents the number of instances from the task's validation set with ground truth class label j that were predicted to be members of class i. A schematic 2×2 confusion matrix for the `autoScan` classification task is shown in Figure 6.

From the confusion matrix, several classifier performance metrics can be computed. Two that frequently appear in the literature are the False Positive Rate (FPR) and the Missed Detection Rate (MDR; also known as the False Negative Rate or False Omission Rate). Using the notation from Figure 6, the FPR is defined by:

$$FPR = \frac{F_p}{F_p + T_n}, \tag{17}$$

and the missed detection rate by

$$MDR = \frac{F_n}{T_p + F_n}. \tag{18}$$

For `autoScan`, the FPR represents the fraction of artifacts in the validation set that are predicted to be legitimate detections of astrophysical variability. The MDR represents the fraction of non-artifacts in the task's validation set that are predicted to be artifacts. Another useful metric is the efficiency or True Positive Rate (TPR),

$$\epsilon = \frac{T_p}{T_p + F_n}, \tag{19}$$

which represents the fraction of non-artifacts in the sample that are classified correctly. For the remainder of this study, we often refer to the candidate-level efficiency measured on fake SNe Ia, ϵ_F (see §4.4).

Finally, the receiver operating characteristic (ROC) is a graphical tool for visualizing the performance of a classifier. It displays FPR as a function of MDR, both of which are parametric functions of τ, the `autoScan` score that one chooses to delineate the boundary between "non-artifacts" and "artifacts." One can use the ROC to determine the location at which the trade-off between the FPR and MDR is optimal for the survey at

Table 3
Grid search results for autoScan hyperparameters.

Hyperparameter	Values
n_estimators	10, 50, **100**, 300
criterion	gini, **entropy**
max_features	5, **6**
min_samples_split	2, **3**, 4, 10, 20, 50
max_depth	**Unlimited**, 100, 30, 15, 5

Note.—A 3-fold cross-validated search over the grid of Random Forest hyperparameters tabulated above was performed to characterize the the performance of the machine classifier. The hyperparameters of the best-performing classifier appear in bold.

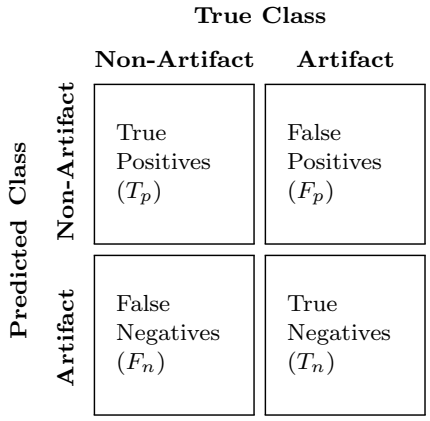

Fig. 6.— Schematic confusion matrix for the autoScan classification task. Each matrix element $\mathbb{E}_{ij}$ represents the number of instances from the task's validation set with ground truth class label j that were predicted to be members of class i.

hand, a function of both the scanning load and the potential bias introduced by the classifier, then solve for the corresponding τ. By benchmarking the performance of the classifier using the the ROC, one can paint a complete picture of its performance that can also serve as a statistical guarantee on performance in production, assuming a validation set and a production data set that are identically distributed in feature space, and that detections are scanned individually in production (see §4.4).

4.2. Classification Task

We used stratified 5-fold cross-validation to test the performance of autoScan. Cross validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. In a k-fold cross-validated analysis, a data set is partitioned into k disjoint subsets. k iterations of training and testing are performed. During the ith iteration, subset i is held out as a "validation" set of labeled data instances that are not included in the training sample, and the union of the remaining $k-1$ subsets is passed to the classifier as a training set. The classifier is trained and its predictive performance on the validation set is recorded. In standard k-fold cross-validation, the partitioning of the original data set into disjoint subsets is done by drawing samples at random without replacement from the original

data set. But in a stratified analysis, the drawing is performed subject to the constraint that the distribution of classes in each subset be the same as the distribution of classes in the original data set. Cross-validation is useful because it enables one to characterize how a classifier's performance varies with respect to changes in the composition of training and testing data sets, helping quantify and control "generalization error."

4.3. Results

Figure 7 shows the ROCs that resulted from each round of cross-validation. We report that `autoScan` achieved an average detection-level MDR of 4.0 ± 0.1 percent at a fixed FPR of 2.5 percent with $\tau = 0.5$, which was ultimately adopted in the survey; see §4.4. We found that `autoScan` scores are correlated with detection signal-to-noise ratio (S/N). Figure 8 displays the efficiency of `autoScan` using 138,057 out-of-sample detections of fake SNe. At S/N $\lesssim 10$, the out-of-sample fake efficiency is markedly lower than it is at higher S/N. The efficiency asymptotically approaches unity for S/N $\gtrsim 100$. The effect becomes more pronounced when the class discrimination boundary is raised. This occurs because legitimate detections of astrophysical variability at low S/N are similar to artifacts.

4.4. Effect of `autoScan` on Transient Candidate Scanning Load

As discussed in §2, DES-SN performs target selection and scanning using aggregates of spatially coincident detections from multiple nights and filters ("candidates"). After the implementation of `autoScan`, the `NUMEPOCHS` requirement described in Table 1 was revised to require that a candidate be detected on at least two distinct nights having at least one detection with an ML score greater than τ to become eligible for visual scanning. In this section we describe the effect of this revision on the scanning load for an entire observing season using a full reprocessing of the Y1 data.

We sought to minimize the size of our transient candidate scanning load with no more than a 1 percent loss in ϵ_F. By performing a grid search on τ, we found that we were able to reduce the number of candidates during the first observing season of DES-SN by a factor of 13.4, while maintain-

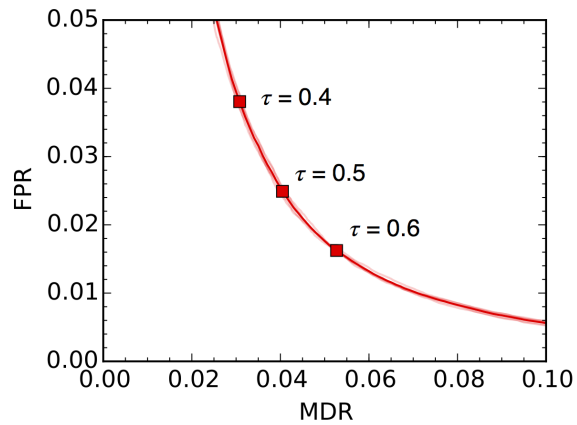

Fig. 7.— 5-fold cross-validated receiver operating characteristics of the best-performing classifier from §3.5. Six visually indistinguishable curves are plotted: one translucent curve for each round of cross-validation, and one opaque curve representing the mean. Points on the mean ROC corresponding to different class discrimination boundaries τ are labeled. $\tau = 0.5$ was adopted in DES-SN.

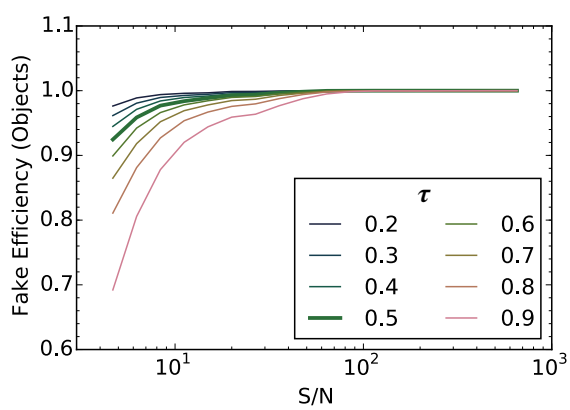

Fig. 8.— Object-level fake efficiency as a function of S/N, at several `autoScan` score cuts. The performance of `autoScan` degrades at low S/N. $\tau = 0.5$ (bold) was adopted in DES-SN.

ing $\epsilon_F > 99.0$ per cent by adopting $\tau = 0.5$. After implementing `autoScan` using this τ, we measured the quantity $\langle N_A/N_{NA} \rangle$, the average ratio of artifact objects to non-artifact detections that a human scanner encountered during a scanning session, using random samples of 3,000 objects drawn from the pool of objects passing the modified and unmodified cuts in Table 1. We found that the ratio decreased by a factor of roughly 40 after the production implementation of `autoScan`. Table 4 summarizes these results.

5. Discussion

With the development of `autoScan` and the use of fake overlays to robustly measure efficiencies, the goal of automating artifact rejection on difference images using supervised ML classification has reached a certain level of maturity. With several historical and ongoing time-domain surveys using ML techniques for candidate selection, it is clear that the approach has been successful in improving astrophysical source selection efficiency on images. However, there are still several ways the process could be improved for large-scale transient searches of the future, especially for ZTF and LSST, whose demands for reliability, consistency, and transparency will eclipse those of contemporary surveys.

5.1. Automating Artifact Rejection in Future Surveys

For surveys like LSST and ZTF, small decreases in MDR are equivalent to the recovery of vast numbers of new and interesting transients. Decreasing the size of the feature set and increasing the importance of each feature is one of the most direct routes to decreasing MDR. However, designing and engineering effective classification features is among the most time-consuming and least intuitive aspects of framework design. Improving MDR by revising feature sets is a matter of trial and error—occasionally, performance improvements can result, but sometimes adding features can degrade the performance of a classifier. Ideally, surveys that will retrain their classifiers periodically will have a rigorous, deterministic procedure to extract the optimal feature set from a given training data set. This is possible with the use of convolutional neural networks (CNNs), a subclass of Artificial Neural Networks, that can take images as input and infer an optimal set of features for a given set of training data. The downside to CNNs is that the resulting features are significantly more abstract than astrophysically motivated features and consequently can be more difficult to interpret, especially in comparison with Random Forests, which assign each feature a relative importance. However, CNNs have achieved high levels of performance for a diverse array of problems. They remain relatively unexplored in the context of astrophysical data processing, and bear examination for use in future surveys.

Next, unless great care is taken to produce a training data set that is drawn from the same multidimensional feature distribution as the testing data, dense regions of testing space might be completely devoid of training data, leading to an unacceptable degradation of classification accuracy in production. Developing a rigorous method for avoiding such sample selection bias is crucial for future surveys, for which small biases in the training set can result in meaningful losses in efficiency. The idea of incorporating active learning techniques into astronomical ML classification frameworks has been advanced as a technique for reducing sample selection bias (Richards et al. 2012).

Given a testing set and a training set which are free to be drawn from different distributions in feature space, in the pool-based active learning for classification framework, an algorithm iteratively selects, out of the entire set of unlabeled data, the object (or set of objects) that would give the maximum performance gains for the classification model, if its true label were known. The algorithm then solicits a user to manually input the class of the object under consideration, and then the object is automatically incorporated into future training sets to improve upon the original classifier. Under this paradigm, human scanners would play the valuable role of helping the classifier learn from its mistakes, and each human hour spent vetting data would immediately carry scientific return. Active learning could produce extremely powerful classifiers over short timescales when used in concert with generative models for training data. Instead of relying on historical data to train artifact rejection algorithms during commissioning phases, experiments like LSST could

Table 4

Effect of autoScan on Reprocessed DES Y1 Transient Candidate Scanning Load.

	No ML	ML ($\tau = 0.5$)	ML / No ML
N_c[a]	100,450	7,489	0.075
$\langle N_A/N_{NA} \rangle$[b]	13	0.34	0.027
ϵ_F[c]	1.0	0.990	0.990

[a]Total number of science candidates discovered.

[b]Average ratio of artifact to non-artifact detections in human scanning pool.

[c]`autoScan` candidate-level efficiency for fake SNe Ia.

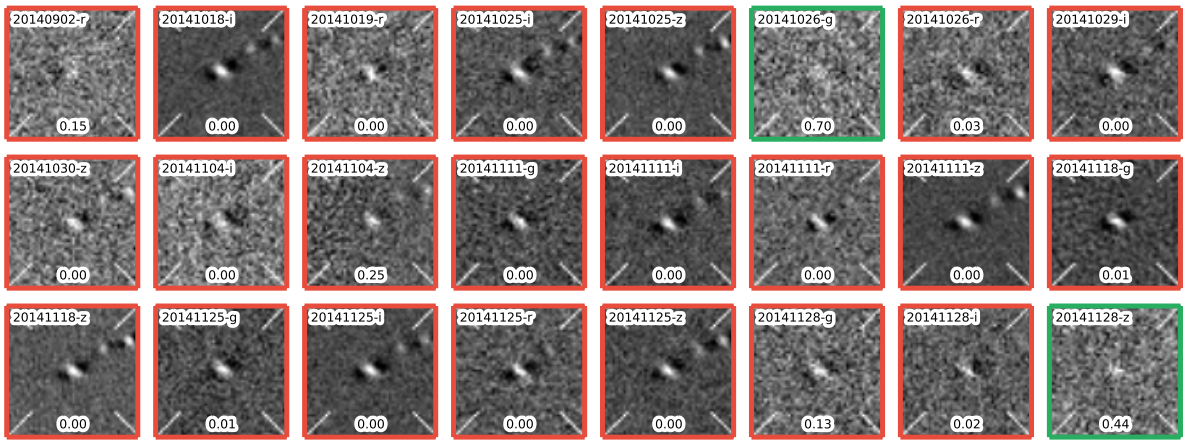

Fig. 9.— 24 consecutively observed difference image cutouts of a poorly subtracted galaxy that was wrongly identified as a transient. The `autoScan` score of each detection appears at the bottom of each cutout. The mis-identification occurred because on two nights the candidate had a detection that received a score above an `autoScan` class discrimination boundary $\tau = 0.4$ used during early code tests (green boxes). Night-to-night variations in observing conditions, data reduction, and image subtraction can cause detections of artifacts to appear real. If a two-night trigger is used, spurious "transients" like this one can can easily accumulate as a season goes on. Consequently, care must be taken when using an artifact rejection framework that scores individual detections to make statements about aggregates of detections. Each image is labeled with the observation date and filter for the image, in the format YYYYMMDD-filter.

use generative models for survey observations to simulate new data sets. After training a classifier using simulated data, in production active learning could be used to automatically fill in gaps in classifier knowledge and augment predictive accuracy.

In this work, we used a generative model of SN Ia observations—overlaying fake SNe Ia onto real host galaxies—to produce the "Non-Artifact" component of our training data set. However, the nearly 500,000 artifacts in our training set were human-scanned, implying that future surveys will still need to do a great deal of scanning before being able to get an ML classifier off the ground. A new survey should not intentionally alter the pipeline to produce artifacts during commissioning, as it is crucial that the unseen data be drawn from the same feature distributions as the training data. For surveys with $\langle N_A/N_{NA}\rangle \gtrsim 100$, Brink et al. (2013) showed that a robust artifact library can be prepared by randomly sampling from all detections of variability produced by the difference imaging pipeline. For surveys or pipelines that do not produce as many artifacts, some initial scanning to produce a few 10^4-artifact library from commissioning data should be sufficient to produce an initial training set (Brink et al. 2013; du Buisson et al. 2014).

5.2. Eliminating Spurious Candidates

Using a two-night trigger, some spurious science candidates can be created due to nightly variations in astrometry, observing conditions, and repeatedly imaged source brightnesses that cause night-to-night fluctuations in the appearance of candidates on difference images. These variations lead to a spread of ML scores for a given candidate. As an observing season progress, artifacts can accumulate large numbers of detections via repeated visits. Although for a typical artifact the vast majority of detections fail the ML requirement, the fluctuations in ML scores can cause a small fraction of the detections to satisfy the `autoScan` requirement. Figure 9 shows an example of this effect.

Mitigating the buildup of spurious multi-night candidates could be achieved by implementing a second ML classification framework that takes as input multi-night information, including the detection-level output of `autoScan`, to predict whether a given science candidate represents a bona-fide astrophysical source. Training data compilation could be performed by randomly selecting time-contiguous strings of detections from known candidates. The lengths of the strings could be drawn from a distribution specified during framework development. Candidate-level features could characterize the temporal variation of detection level features, such as the highest and lowest night-to-night shifts in `autoScan` score, magnitude, and astrometric uncertainty.

DAG thanks Josh Bloom for productive conversations that improved the paper. We are grateful for the extraordinary contributions of our CTIO colleagues and the DES Camera, Commissioning and Science Verification teams for achieving excellent instrument and telescope conditions that have made this work possible. The success of this project also relies critically on the expertise and dedication of the DES Data Management organization. Funding for DES projects has been provided by the U.S. Department of Energy, the U.S. National Science Foundation, the Ministry of Science and Education of Spain, the Science and Technology Facilities Council of the United Kingdom, the Higher Education Funding Council for England, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, the Kavli Institute of Cosmological Physics at the University of Chicago, Financiadora de Estudos e Projetos, Fundação Carlos Chagas Filho de Amparo á Pesquisa do Estado do Rio de Janeiro, Conselho Nacional de Desenvolvimento Científico e Tecnológico and the Ministério da Ciência e Tecnologia, the Deutsche Forschungsgemeinschaft and the collaborating institutions in the Dark Energy Survey.

The collaborating institutions are Argonne National Laboratory, the University of California, Santa Cruz, the University of Cambridge, Centro de Investigaciones Energeticas, Medioambientales y Tecnologicas-Madrid, the University of Chicago, University College London, the DES-Brazil Consortium, the Eidgenössische Tecnische Hochschule (ETH) Zürich, Fermi National Accelerator Laboratory, the University of Edinburgh, the University of Illinois at Urbana-Champaign, the Institut de Ciencies de l'Espai (IEEC/CSIC), the Intitut de Fisica d'Altes Energies, Lawrence Berke-

ley National Laboratory, the Ludwig-Maximilians Universität and the associated Excellence Cluster Universe, the University of Michigan, the National Optical Astronomy Observatory, the University of Nottingham, the Ohio State University, the University of Pennsylvania, the University of Portsmouth, SLAC National Acclerator Laboratory, Stanford University, the University of Sussex, and Texas A&M University.

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Figure 3 was generated with a modified version of `triangle.py` (Foreman-Mackey et al. 2014). ACR acknowledges financial support provided by the PAPDRJ CAPES/FAPERJ Fellowship. FS acknowledges financial support provided by CAPES under contract No. 3171-13-2. The DES participants from Spanish institutions are partially supported by MINECO under grants AYA2012-39559, ESP2013-48274, FPA2013-47986, and Centro de Excelencia Severo Ochoa SEV-2012-0234, some of which include ERDF funds from the European Union.

REFERENCES

Alard, C., & Lupton, R. H. 1998, ApJ, 503, 325

Albrecht, A., et al. 2006, arXiv:astro-ph/0609591

Aldering, G., et al. 2002, SPIE, 4836, 61

Aragon, C. R., et al. 2008, JPhCS, 125, 012091

Astier, P., et al. 2006, A&A, 447, 31

Bailey, S., et al. 2007, ApJ, 665, 1246

Baltay, C., et al. 2007, PASP, 119, 1278

Banerji, M., et al. 2010, MNRAS, 406, 342

Bernstein, J. P., et al. 2012, ApJ, 753, 152

Bertin, E. 2006, ASPC, 351, 112

Bertin, E., & Arnouts, S., 1996, A&AS, 117, 393

Blanc G., et al. 2004, A&A, 423, 881

Bloom, J. S., et al. 2012, PASP, 124, 1175

Bloom, J. S., & Richards, J. W. 2012, in Advances in Machine Learning and Data Mining for Astronomy, eds. M. J. Way et al. (London: CRC Press), 89

Bouy, H., et al. 2013, A&A, 554, AA101

Breiman, L. 2001, Machine Learning, 45, 5

Breiman, L., et al. 1984, Classification and Regression Trees (London: CRC Press)

Brink, H., et al. 2013, MNRAS, 435, 1047

du Buisson, L., Sivanandam, N., Bassett, B. A., Smith, M. 2014, arXiv, arXiv:1407.4118

Desai, S., et al. 2012, ApJ, 757, 83

Diehl, H. T. (for the Dark Energy Survey Collaboration) 2012, PhPro, 37, 1332

Diehl, H. T., et al. 2014, SPIE, 9149, 91490V

Filippenko, A. V., et al. 2001, ASPC, 246, 121

Flaugher, B. 2005, IJMPA, 20, 3121

Flaugher, B., et al. 2012, SPIE, 8446, 844611

Flaugher, B., et al. 2015, preprint FERMILAB-PUB-15-121-PPD.

Foreman-Mackey, D., Price-Whelan, A., Ryan, G., et al. 2014, http://dx.doi.org/10.5281/zenodo.11020

Frieman, J. A., et al. 2008, AJ, 135, 338

Gini, C. 1921, Econ. J., 31, 124

Hamuy, M., et al. 1993, AJ, 106, 2392

Holtzman, J. A., et al. 2008, AJ, 136, 2306

Ivezić, Ż., et al. 2013, Statistics, Data Mining, and Machine Learning in Astronomy (Princeton, NJ: Princeton UP)

Kaiser, N., et al. 2002, SPIE, 4836, 154

Karpenka, N. V., Feroz, F., Hobson, M. P. 2013, MNRAS, 429, 1278

Kessler, R., et al. 2009, PASP, 121, 1028

LSST Science Collaboration 2009, arXiv:0912.0201

Mainzer, A., et al. 2011, ApJ, 731, 53

Pedregosa, F., et al. 2012, arXiv:1201.0490

Perlmutter, S., et al. 1997, AAS, 29, 1351

Richards, J. W., et al. 2011, ApJ, 733, 10

Richards, J. W., et al. 2012, ApJ, 744, 192

Sako M., et al. 2008, AJ, 135, 348

Sako M., et al. 2011, ApJ, 738, 162

Sako M., et al. 2014, arXiv:1401.3317

Schmidt, B. P., et al. 1998, ApJ, 507, 46

Shannon, C. E. 1948, BSTJ, 27, 379

Smith, A. M., et al. 2011, MNRAS, 412, 1309

Smith, R. M., et al. 2014, SPIE, 9147,

Strolger, L.-G., et al. 2004, ApJ, 613, 200

Tyson, J. A. 2002, SPIE, 4836, 10

Vapnik, V. 1995, The Nature of Statistical Learning Theory (New York: Springer)

van Dokkum, P. G. 2001, PASP, 113, 1420

Waszczak, A., et al. 2013, MNRAS, 433, 3115

Willsky, A. S., Wornell, G. W., Shapiro, J. H. 2003, Stochastic Processes, Detection and Estimation (Course notes for MIT 6.432)

Zhu, J., et al. 2009, SII, 2.3, 349

Zwicky , F. 1964, AnAp, 27, 300

This 2-column preprint was prepared with the AAS LaTeX macros v5.2.