Astronomy and Computing 25 (2018) 103–109

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Galaxy detection and identification using deep learning and data augmentation☆

R.E. González [a,b,*], R.P. Muñoz [a], C.A. Hernández [b]

[a] Centro I+D MetricArts, Santiago, Chile
[b] Centro de Astro-Ingeniería, Pontificia Universidad Católica, Av. Vicuña Mackenna 4860, Santiago, Chile

ARTICLE INFO

Article history:
Received 25 March 2018
Accepted 4 September 2018
Available online 15 September 2018

Keywords:
Galaxies
General
Techniques
Image processing
Computing methodologies
Machine learning

ABSTRACT

We present a method for automatic detection and classification of galaxies which includes a novel data-augmentation procedure to make trained models more robust against the data taken from different instruments and contrast-stretching functions. This method is shown as part of AstroCV, a growing open source computer vision repository for processing and analyzing big astronomical datasets, including high performance Python and C++ algorithms used in the areas of image processing and computer vision.

The underlying models were trained using convolutional neural networks and deep learning techniques, which provide better results than methods based on manual feature engineering and SVMs in most of the cases where training datasets are large. The detection and classification methods were trained end-to-end using public datasets such as the Sloan Digital Sky Survey (SDSS), the Galaxy Zoo, and private datasets such as the Next Generation Virgo (NGVS) and Fornax (NGFS) surveys.

Training results are strongly bound to the conversion method from raw FITS data for each band into a 3-channel color image. Therefore, we propose data augmentation for the training using 5 conversion methods. This greatly improves the overall galaxy detection and classification for images produced from different instruments, bands and data reduction procedures.

The detection and classification methods were trained using the deep learning framework DARKNET and the real-time object detection system YOLO. These methods are implemented in C language and CUDA platform, and makes intensive use of graphical processing units (GPU). Using a single high-end Nvidia GPU card, it can process a SDSS image in 50 ms and a DECam image in less than 3 s.

We provide the open source code, documentation, pre-trained networks, python tutorials, and how to train your own datasets, which can be found in the AstroCV repository. https://github.com/astroCV/astroCV.

1. Introduction

Astronomical datasets are constantly increasing in size and complexity. The modern generation of integral field units (IFUs) are generating about 60 GB of data per night while imaging instruments are generating 300 GB per night. The Large Synoptic Survey Telescope (LSST; Ivezic et al., 2008) is under construction in Chile and it is expected to start full operations in 2022. With a wide 9.6 square degree field of view 3.2 Gigapixel camera, LSST will generate about 20 TB of data per night and will detect more than 20 million of galaxies.

Machine learning techniques have been increasingly employed in data-rich areas of science. They have been used in genomics, high-energy physics and astronomy. Some examples in astronomy are the detection of weird galaxies using Random Forests on Sloan data (Baron and Poznanski, 2017), Gravity Spy (Zevin et al., 2017) for LIGO detections and using convolutional neural network (CNN) in identifying strong lenses in imaging data (Jacobs et al., 2017).

Computer Vision is an interdisciplinary field that focuses on how machines can emulate the way in which human's brains and eyes work together to visually process the world around them. For many years, the detection of objects was computed using manual feature engineering and descriptors such as SIFT and HOG (Dalal and Triggs, 2005). Thanks to the advent of large annotated datasets and gains in computing power, deep learning methods have become the favorite for doing detection and classification of objects.

The classification of optical galaxy morphologies is based on a few simple rules that make them suitable for machine learning and computer vision techniques. The Kaggle Galaxy Zoo (Willett et al., 2013) was a competition based on a citizen science project where the aim was to predict the probability distribution of people's

☆ This code is registered at the ASCL with the code entry 1804.004.

* Corresponding author at: Centro de Astro-Ingeniería, Pontificia Universidad Católica, Av. Vicuña Mackenna 4860, Santiago, Chile.
E-mail address: regonzar@astro.puc.cl (R.E. González).

https://doi.org/10.1016/j.ascom.2018.09.004
2213-1337/

Table 1
Annotations in different subsamples.

Dataset	Elliptical	Spiral	Edge-on	DK	Merge	Total	Number images
Training S1	10 366	4535	4598	223	381	20 103	6 458
Validation S1	1 261	714	723	27	45	2 770	921
Training S2	18 030	7828	7910	350	648	34 766	11 010
Validation S2	2 119	856	873	36	82	3 966	1 161
Custom	705	401	462	474	135	2 177	87

responses about the morphology of a galaxy using optical image data, and the winning solution used CNNs (Dieleman et al., 2015).

We present a method for galaxy classification and identification with a novel data augmentation procedure which is part of AstroCV, a computer vision library for processing and analyzing big astronomical datasets. The goal of AstroCV is to provide a community repository for fast Python and C++ implementations of common tools and routines used in the areas of image processing and computer vision. In particular, it is focused on the task of object detection, segmentation and classification applied to astronomical sources.

In this paper we will focus on the automatic detection and classification of galaxies. The detection and classification methods were trained end-to-end using public datasets from the Sloan Digital Sky Survey (SDSS), (Alam et al., 2015), and Galaxy Zoo (Lintott et al., 2008, 2011) explained in Section 2.1. We use YOLO method, (Redmon et al., 2015), for object detection which is explained in Section 2.2. Training process is described in Section 2.3, and results are shown in Section 3.

The open source code, training datasets, documentation and python notebooks of AstroCV are freely available in a Github repository.[1]

2. Data and training

2.1. Dataset

Galaxy Zoo[2] (Lintott et al., 2008, 2011) is the most successful citizen project in Astronomy. It consists of a web platform for doing visual inspection of astronomical images and morphological classification of galaxies. Hundreds of thousands of volunteers classified images of nearly 900,000 galaxies drawn from the SDSS survey. The Galaxy Zoo classification consists of six categories: elliptical, clockwise spiral, anticlockwise spiral, edge-on, star/do not know, or merger.

We extracted the galaxy classification for a sub-sample of 38,732 galaxies and downloaded their respective gri-band images from the SDSS fields. Sub-sample S1 is produced from 20 000 field images and sub-sample S2 is produced from 32 000 field images.

For each field image, we select the galaxies with a size larger than 22 pixels box side. This galaxy size is computed as 2.1 times the r-band petrosian radius. After this size filter we stay with two samples of 7397 images and 12 171 images. Then, we split each of these two samples into training and validation sub-sets, resulting in S1 and S2 datasets. In addition, we include a small custom dataset (Custom hereafter) with manually annotated galaxies from Hubble Deep Field image gallery,[3] CFHT,[4] and others images randomly taken from public databases. See Table 1 with details of the different samples.

2.2. YOLO

You only look once (YOLO) method (Redmon et al., 2015; Redmon and Farhadi, 2016), is a Single Shot Detector (SSD), it means it computes in a single network the region proposal and classifier. This method runs the image on a Convolutional Neural Network (CNN) model and gets the detection on a single pass. The network is composed of 23 convolution layers and 5 maxpool layers shown in Fig. 1, and it is programmed on Darknet, an open source neural network framework in C and CUDA. It is very fast and takes full advantage of graphical processing units (GPU). This method formerly developed for person/object detection, is configured for the training and detection of galaxies.

Table 2
Training sets.

Name	Dataset	Filters	Images
T1	S1	L	6 458
T2	S1	LH	6 458
T3	S2	L	11 010
T4	S2	L+LH+S+SH+Q	55 050
T5	S2	LH+SH	22 020
T6	S2	L+LH+S+SH+Q	32 290

(L = Lupton, LH = Lupton high contrast, S = sinh, SH = sinh high contrast, Q = sqrt; C = custom Hubble sample.)

2.3. Training and data augmentation

YOLO method is designed to work on 3 channel color images, usually RGB color scale. In astronomy images are taken for each filter in FITS format with raw CCD data for each pixel. Data conversion from FITS to RGB images (or contrast stretching) is not unique and depends on the telescope's camera, band filters, reduction schema, and most important, it depends on the conversion method used to scale photon counts to color scale.

There are several conversion methods, however to emphasize galaxies with strong luminosity gradients, linear scaling is not suitable, i.e. a spiral galaxy radial luminosity profile can be modeled as a power law (de Vaucouleurs profile) for the bulge plus an exponential for the disk. In those cases, the scaling methods commonly used are *sinh*, *asinh*, *sqrt* functions. SDSS uses (Lupton et al., 2004) as standard conversion method from FITS in *igr* bands to RGB an image (Lupton method hereafter).

In general, to train neural networks using images, the data augmentation is fundamental, it means increasing the training dataset with different transformations of the same data (scaling, rotations, crop, warp, etc.); In YOLO this data augmentation is already implemented in the training procedure, however we need to produce a color-scale/filter conversion augmentation as well (filter hereafter), to build a training robust against RGB images coming from different filters, bands and instruments.

In the top performance deep learning methods for object detection, we have also Faster R-CNN, Single shot detectors (SSD), Deconvolutional DSSD, Focal Loss, Feature Pyramid Networks (FPN). In Lin et al. (2017) there is a complete review and comparison on current methods. Most of these methods present similar mean average precision when compared to YOLO, however we stay with the latter since it is the fastest and implemented in C with CUDA acceleration.

In Table 2 we show 5 different training sets with RGB images produced to explore dataset size and filter augmentation, we use

[1] https://github.com/astroCV.
[2] https://www.galaxyzoo.org/.
[3] http://www.spacetelescope.org/science/deep_fields.
[4] http://www.cfht.hawaii.edu.