Draft version April 12, 2022
Typeset using LaTeX **preprint** style in AASTeX631

Term Project
Galaxy Zoo: Probabilistic Morphology through Bayesian CNNs and Active Learning

Colin Leach

ABSTRACT

Astronomical survey data has expanded impressively since the era when professional astronomers could keep up with it by themselves. As an early enhancement, Galaxy Zoo used large numbers of amateur volunteers for classification of SDSS results, more recently extended to HST, CANDELS and DECaLS images. To scale further for the Rubin/Euclid era, that approach needs to be supplemented with ML techniques to use the volunteers more efficiently. Walmsley et al. (2020a) attempts to develop such a hybrid human/ML system. The current term project attempts to reproduce and (perhaps) extend this work.

1. INTRODUCTION

The Galaxy Zoo started as an attempt to scale manual classification of SDSS images by recruiting citizen scientists (Lintott et al. 2008; Lintott 2019). This succeeded beyond expectations, but is struggling to keep up with new data sources: DES, Rubin, Euclid, etc. Volunteer input is increasingly regarded as a finite and valuable resource, which needs to be used more efficiently (Dickinson et al. 2020).

Sorting galaxies by color has been done for decades (blue spirals, red ellipticals), though this has been criticized as inaccurate (Smethurst et al. 2022). Other approaches include radial brightness curves, looking for central bulges and bars. Attempts to use neural networks to classify morphology go back at least to a Kaggle challenge in 2014, won by Dieleman et al. (2015). The concept of transfer learning, using older surveys to train models for a newer one, was explored by Domínguez Sánchez et al. (2019) and later by Walmsley et al. (2020a) (hereafter W+20), discussed in more detail in Walmsley et al. (2021) (hereafter W+21). These all focus on visual images (or their equivalents redshifted to IR), but Fielding et al. (2021) discusses an exchange of techniques with radio astronomy. A broader review of ML in astronomy is given in Fluke & Jacobs (2020).

THe GZ2 catalog (Willett et al. 2013; Hart et al. 2016) is based on SDSS DR7. Later catalogs include Galaxy Zoo: Hubble (Willett et al. 2017), CANDELS (Simmons et al. 2017) and DECaLS (W+21).

2. AIMS

In W+20, an attempt is described to develop a human-machine hybrid strategy for galaxy morphology:

- Use the large Galaxy Zoo 2 (GZ2) catalog to train a CNN that can classify SDSS images.

- Use this model as a starting point to classify new data sources and formats, using only modest amounts of labeling from human volunteers to fine-tune the model.

3. CODE

3.1. *Zoobot Code*

Python/Tensorflow code is on Github[1] (Walmsley 2019), claiming to be an exact copy of that used for W+20.

Perhaps more interesting is the zoobot repo[2], a fork which is still under active development. This extends the project to DECaLS (Dark Energy Camera Legacy Survey) data, as described in W+21. It also has much better documentation than the earlier code.

3.2. *Code for Term Project*

Python code and documentation associated with ASTR 502 is available on Github[3]. This aims to cover both GZ2, as in W+20, and DECaLS, as in W+21.

4. COMPUTATION

W+20 reports that GZ2 training was carried out on a p2.xlarge EC2 instance with K80 GPU, taking about 8 hours. For DECaLS, the GPU was upgraded to a V100.

Experiments with the GPUs available at the start of this project rapidly proved that 2GB of GPU memory is wholly inadequate for training a CNN. Upgrading to a 6GB GTX 1660 (far from state of the art, but only $450 and compatible with the existing motherboard and PSU) allowed some progress. However, this still proved limiting for batch size as discussed below.

TODO Colab and AWS

5. GOALS

My time is less valuable than for faculty or grad students, so goals are open-ended depending on energy, enthusiasm and (hopefully) competence. Roughly:

1. Get the published Keras code running on my local machine, using whatever cut-down training sets (GZ2 and DECaLS) prove viable.
2. Deploy the code on either AWS or Google.
3. Repeat for PyTorch code
4. Extend the model to other data such as Hubble or CANDELS, for which there is already some GZ classification.
5. Rewrite using other languages and frameworks, for my education: Julia with Flux; maybe F#/ML.NET.

Not all of this will be done before the end of the semester (an understatement).

[1] https://github.com/mwalmsley/galaxy-zoo-bayesian-cnn
[2] https://github.com/mwalmsley/zoobot
[3] https://github.com/colinleach/proj502

6. ALGORITHMS

6.1. *What are we trying to predict?*

Galaxy Zoo catalogs are not just a simple classification, such as elliptical vs spiral. The questions posed to volunteers have evolved over the years, though all follow a decision tree which depends on the answer to previous questions. The version for DECaLS DR5 is shown in Figure 1; GZ2 is similar but slightly simpler.

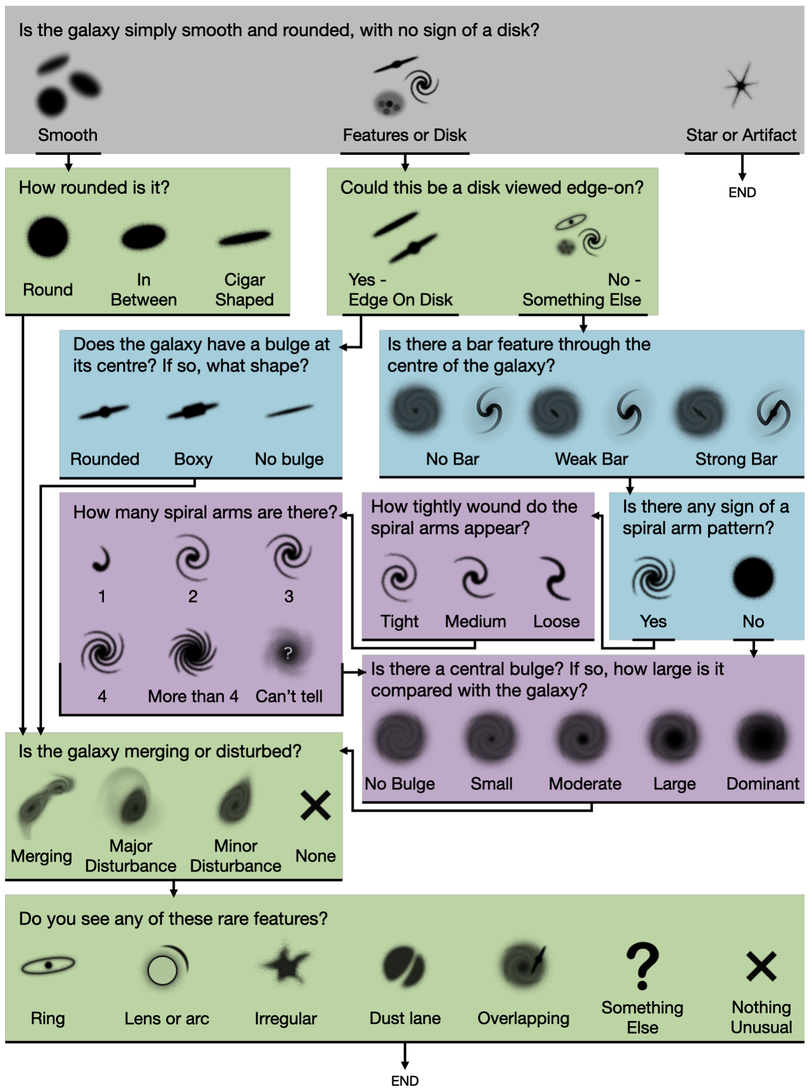


Figure 1. The GZ decision tree used for DECaLS DR5

In the Python code this is represented by two dictionaries: for questions/answers and for dependencies. The Q&A version is shown below: keys are questions, values are lists of allowed answers (as a suffix which will be appended to the question). The dependency dictionary lists previous questions that would allow the current question to be reached.

```
decals_pairs = {
  'smooth-or-featured': ['_smooth', '_featured-or-disk', '_artifact'],
  'disk-edge-on': ['_yes', '_no'],
  'has-spiral-arms': ['_yes', '_no'],
  'bar': ['_strong', '_weak', '_no'],
  'bulge-size': ['_dominant', '_large', '_moderate', '_small', '_none'],
  'how-rounded': ['_round', '_in-between', '_cigar-shaped'],
  'edge-on-bulge': ['_boxy', '_none', '_rounded'],
  'spiral-winding': ['_tight', '_medium', '_loose'],
  'spiral-arm-count': ['_1', '_2', '_3', '_4', '_more-than-4', '_cant-tell'],
  'merging': ['_none', '_minor-disturbance', '_major-disturbance', '_merger']
}
```

Thus there are 10 possible questions (not all of which will be asked in each case), and 34 possible answers. Each answer has its own field in the input to the model (in addition to an identifier and the image), and the training output includes a weighting for each. The prediction step then takes a new galaxy and produces a probability for each of the possible answers.

TODO Dirichlet?

6.2. *ML model*

This evolved during the development of Zoobot. For W+20 and the mwalmsley/galaxy-zoo-bayesian-cnn repo, the architecture was a cut-down version of VGG16 (Simonyan & Zisserman 2014). For W+21 and mwalmsley/zoobot it had been updated to EfficientNetB0 (Tan & Le 2019). The latter was used in the current work.

The model summary reported by Keras for DECaLS training is shown in Table 1. The first three layers are fairly standard image preprocessing steps (data augmentation). The EfficientNet component is all in the "sequential 1" layer, followed by pooling and dropout. The final dense layer gives a 34-component output, corresponding to the possible answers from the volunteers.

TODO Details

Table 1. Output from TensorFlow model.summary()

Layer	Output Shape	Param #
random rotation	(None, 300, 300, 1)	0
random flip	(None, 300, 300, 1)	0
random crop	(None, 224, 224, 1)	0
sequential 1	(None, 7, 7, 1280)	4048988
global avg pooling 2d	(None, 1280)	0
top dropout	(None, 1280)	0
dense	(None, 34)	43554

6.3. *Loss Function*

In W+20 volunteer responses were modeled as binomially distributed

$$\mathcal{L} = \int \text{Bin}(k|\rho, N)\, \text{Beta}(\rho|\alpha, \beta)\, d\alpha\, d\beta \tag{1}$$

To address some limitations, W+21 modified this to use the multinomial equavalent of each function, replacing Binomial($k|\rho, N$) with Multinomial($\vec{k}|\vec{\rho}, N$) and Beta($\rho|\alpha, \beta$) with Dirichlet($\vec{\rho}|\vec{\alpha}$):

$$\mathcal{L} = \int \text{Multi}(\vec{k}|\vec{\rho}, N)\, \text{Dirichlet}(\vec{\rho}|\vec{\alpha})\, d\vec{\alpha} \tag{2}$$

The parameters are now vectors with one element per answer.

6.4. *Output*

TODO

7. WORKFLOW

In outline, these are the steps required:

1. Get the Galaxy Zoo catalog data for each survey of interest.
2. Get the image files (JPG or PNG), one per galaxy in the classification.
3. Make a combined catalog, including a path to the image on disk plus the data fields relevant to the model.
4. Split the galaxies into train, evaluate and test sets. For each, prepare a binary-format tensor (tfrecord) file containing image and classification data.
5. Train the model on the train and evaluate sets.
6. Predict results with the test set and compare with the GZ classification.

The following subsections address each of these in more detail.

7.1. *GZ data*

GZ2 catalog files were downloaded from the Galaxy Zoo website. There are a total of 243,500 rows in the table. For better consistency, only those marked 'original' in the sample field were used in subsequent analyses, a set of 211,922.

Extensive DECaLS data is available from Zenodo (Walmsley et al. 2020b). For this study the file 'gz_decals_volunteers_5.parquet' was used, a total of 253,286 rows.

For maximum flexibility (and because old habits die hard), all this data was stored in a PostgreSQL database, running locally.

7.2. *Images*

The GZ team do not make their images library publicly available, so the RA and Dec fields in the GZ2 dataset were used to fetch 424×424 JPG cutouts from the SDSS SkyServer. Because Zoobot is currently configured to use PNG images, the Python code converted each file with PIL. The PNG files totaled around 33 GB, much more than the corresponding JPG files.

DECaLS DR5 images were downloaded from Zenodo (Walmsley et al. 2020b) as 4 ZIP files, unpacked to 272,725 424×424 PNG files totaling 83 GB.

File paths and some metadata was stored in PostgeSQL.

Although the survey telescopes are at different latitudes (SDSS at Apache Point, NM; DECaLS at Cerro Tololo, Chile) there is significant overlap in coverage (Figure 2).

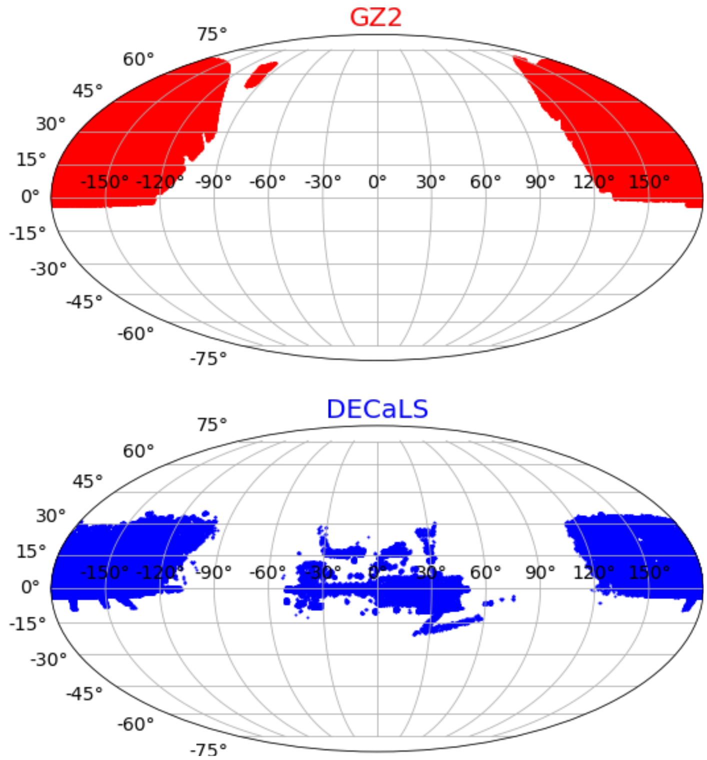


Figure 2. Sky locations of images used for each survey

7.3. *Combined catalog*

Having everything in PostgreSQL makes it easy to join the data and image tables and select the desired columns. Each resulting dataset was converted to a pandas DataFrame and saved as a CSV file. This is quick and produces relatively small files (around 35 MB).

Zoobot requires the columns to have the correct names and appear in the correct order. A galaxy identifier is in 'id_str' and a full path to the PNG is in 'file_loc', then the remaining columns contain total votes cast for each answer in the GZ decision tree.

7.4. *Tensor shards*

Before training, input data needs to be converted to a tfrecord format that TensorFlow can read quickly. The combined catalog is split into train, evaluate and test sets; for this project a 7:2:1 ratio was used. For each set, image files are read and undergo initial cropping and resizing before combining with the GZ votes and written to binary tfrecord files. This took around 1 hour per survey (i9 processor, local SSD storage) and the output files total about 100 GB.

For debugging, a much smaller GZ2 shard set was also created, with fewer records and low-resolution images. This proved valuable in quickly finding some minor bugs in the current Zoobot repo: apparently it was tested mainly with DECaLS data and there are some typos and omissions in the GZ2 code. Accordingly, from this point the project uses my fork of the mwalmsley/zoobot repo. A PR with the corrections will be submitted upstream once everything is working correctly.

7.5. *Training*

As expected, this proved a slow step in the workflow and exposed the limitations of the local (6 GB) GPU. A batch size of 128 was used in the published work. For GZ2, this caused an immediate

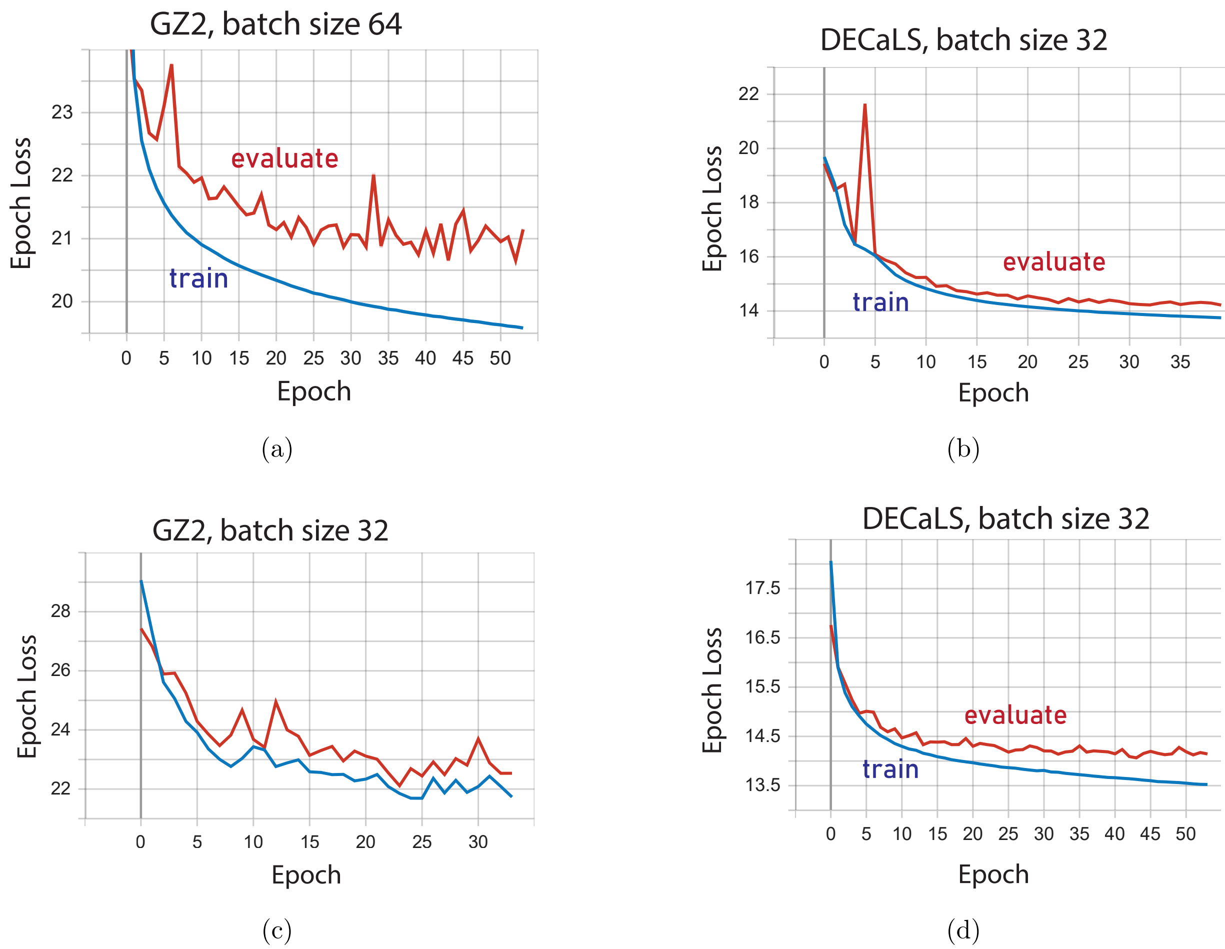


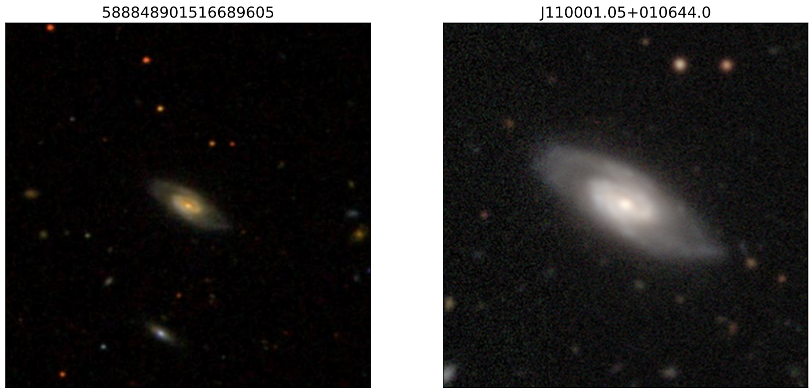
Figure 3. Training runs for GZ2 (left) and DECaLS (right)

GPU out-of-memory error. Dropping to batches of 64 was more successful, as in Figure 3(a), though this used most of the available GPU memory. Progressing at about 10 min/epoch, the training loss drops smoothly and reached stopping criteria (TODO) after 54 epochs, 8.3 hours. However, the evaluation loss (??) is noisy and suggests rather poor generalization.

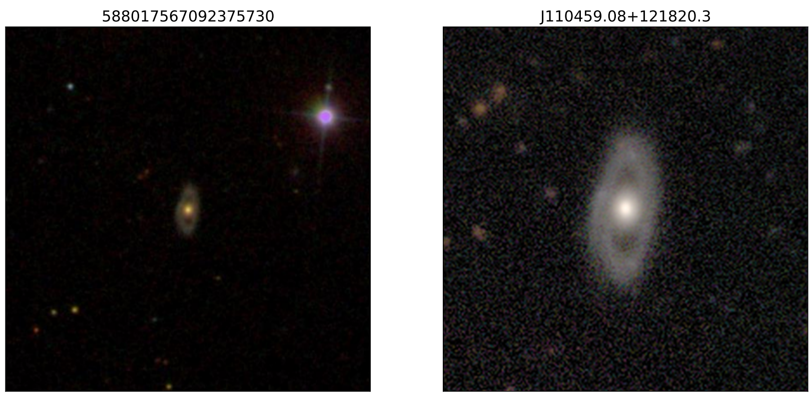
For DECaLS, the batch size needed to be reduced to 32 to fit in GPU memory. Training is slower (about 30 min/epoch) but the results are more encouraging, as shown in Figure 3(b). After some initial spikes, the evaluation loss tracks closely with the training loss. This run failed to reach stopping criteria within the epoch limit (40 epochs, nearly 19 hours) but looks good enough to progress with.

It is not immediately clear why the DECaLS run looks better than the GZ2 run. Preparation of data shards uses the same code and no error has yet been found. Other hypotheses include the different batch size and different image size and quality. Batch size is easiest to test, so GZ2 training was repeated with batches of 32 as in Figure 3(c). This is not encouraging: training now looks worse without evaluation looking better.

Images obtained from DECaLS are inherently higher resolution and deeper than those from SDSS used in GZ2 (bigger telescope, newer camera). Training was also carried out on differently sized images: 300×300 for DECaLS and 256×256 for GZ2.



(a)



(b)

Figure 4. Raw images used for GZ2 (left) and DECaLS (right)

There are also significant differences in image preparation. W+20 gives little detail about this for GZ2, so the simple method described in section 7.2 was followed. In contrast, W+21 describes a more complex process, starting from FITS data files at native telescope resolution. Something equivalent may be possible for SDSS, but as this is not an urgent priority for an ASTR 502 term paper it may be better to focus on the DECaLS survey.

The respective catalogs share no common ID field, but a match on RA/Dec coordinates identified 132,722 images which are in both data sets. A few representative examples are shown in Figure 4. Clearly there is a major difference in quality.

TODO Repeat with GZ2 images zoomed in

8. PREDICTIONS

TODO

REFERENCES

Dickinson, H., Fortson, L., Scarlata, C., Beck, M., & Walmsley, M. 2020, in Panchromatic Modelling with Next Generation Facilities, ed. M. Boquien, E. Lusso, C. Gruppioni, & P. Tissera, Vol. 341, 99–103, doi: 10.1017/S1743921319001418

Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441, doi: 10.1093/mnras/stv632

Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., et al. 2019, MNRAS, 484, 93, doi: 10.1093/mnras/sty3497

Fielding, E., Nyirenda, C. N., & Vaccari, M. 2021, arXiv e-prints, arXiv:2111.04353. https://arxiv.org/abs/2111.04353

Fluke, C. J., & Jacobs, C. 2020, WIREs Data Mining and Knowledge Discovery, 10, e1349, doi: 10.1002/widm.1349

Hart, R. E., Bamford, S. P., Willett, K. W., et al. 2016, MNRAS, 461, 3663, doi: 10.1093/mnras/stw1588

Lintott, C. 2019, The crowd & the cosmos: adventures in the zooniverse, first edition edn. (Oxford: Oxford University Press)

Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179, doi: 10.1111/j.1365-2966.2008.13689.x

Simmons, B. D., Lintott, C., Willett, K. W., et al. 2017, MNRAS, 464, 4420, doi: 10.1093/mnras/stw2587

Simonyan, K., & Zisserman, A. 2014, arXiv e-prints, arXiv:1409.1556. https://arxiv.org/abs/1409.1556

Smethurst, R. J., Masters, K. L., Simmons, B. D., et al. 2022, \mnras, 510, 4126, doi: 10.1093/mnras/stab3607

Tan, M., & Le, Q. V. 2019, arXiv e-prints, arXiv:1905.11946. https://arxiv.org/abs/1905.11946

Walmsley, M. 2019, Zenodo, doi: 10.5281/zenodo.2677874

Walmsley, M., Smith, L., Lintott, C., et al. 2020a, \mnras, 491, 1554, doi: 10.1093/mnras/stz2816

Walmsley, M., Lintott, C., Tobias, G., et al. 2020b, Galaxy Zoo DECaLS: Detailed Visual Morphology Measurements from Volunteers and Deep Learning for 314,000 Galaxies, 0.0.2, Zenodo, doi: 10.5281/zenodo.4573248

Walmsley, M., Lintott, C., Géron, T., et al. 2021, Monthly Notices of the Royal Astronomical Society, 509, 3966–3988, doi: 10.1093/mnras/stab2093

Willett, K. W., Lintott, C. J., Bamford, S. P., et al. 2013, Monthly Notices of the Royal Astronomical Society, 435, 2835, doi: 10.1093/mnras/stt1458

Willett, K. W., Galloway, M. A., Bamford, S. P., et al. 2017, MNRAS, 464, 4176, doi: 10.1093/mnras/stw2568