

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 (GZ) project 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 handle 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 Github¹ ([Walmsley 2019](https://github.com/mwalmesley/galaxy-zoo-bayesian-cnn)), claiming to be an exact copy of that used for W+20.

Perhaps more interesting is the zoobot repo², 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 Github³. 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 to me 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 affordable and compatible with the existing motherboard and PSU) allowed some progress. This still proved limiting for batch size as discussed below, but was useful for debugging before moving to Colab. Making predictions from a pre-trained model was less demanding and worked well on local hardware.

For training runs, the free tier of Colab had few advantages. Reaching the target batch size of 128 used a 16GB V100 GPU for long periods (> 12h), with Colab upgraded to the Pro and later Pro+ tiers.

Only one attempt was made to use a Colab TPU, unsuccessfully. Getting this working will require copying large files to the TPU before each run because (unlike a GPU) this cannot be linked to Google Drive storage.

5. TERM PROJECT STRATEGU

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.

¹ <https://github.com/mwalmesley/galaxy-zoo-bayesian-cnn>

² <https://github.com/mwalmesley/zoobot>

³ <https://github.com/colinleach/proj502>

5. Rewrite using other languages and frameworks, for my education: Julia with Flux; maybe F#/ML.NET.

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

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.

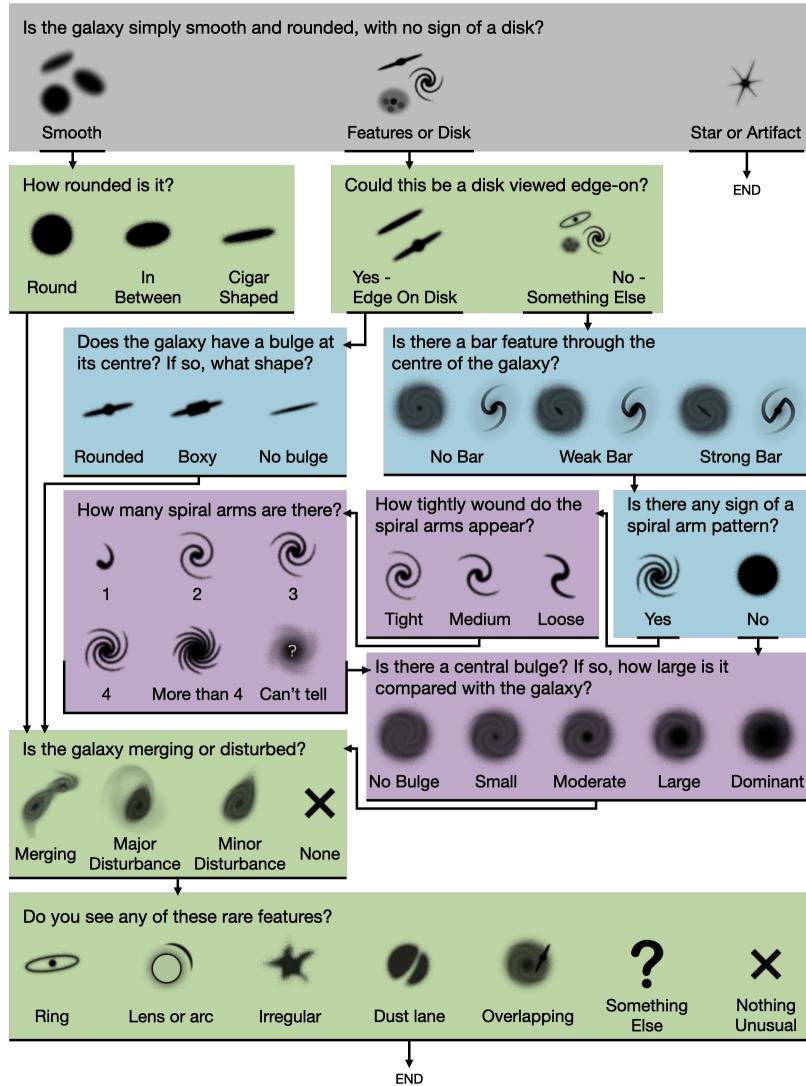


Figure 1. The GZ decision tree used for DECaLS DR5

In the Python code this is represented by two dictionaries: for questions/answers and for dependencies. The Q&A version for DECaLS 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.

```

1  decals_pairs = {
2      'smooth-or-featured': ['_smooth', '_featured-or-disk', '_artifact'],
3      'disk-edge-on': ['_yes', '_no'],
4      'has-spiral-arms': ['_yes', '_no'],
5      'bar': ['_strong', '_weak', '_no'],
6      'bulge-size': ['_dominant', '_large', '_moderate', '_small', '_none'],
7      'how-rounded': ['_round', '_in-between', '_cigar-shaped'],
8      'edge-on-bulge': ['_boxy', '_none', '_rounded'],
9      'spiral-winding': ['_tight', '_medium', '_loose'],
10     'spiral-arm-count': ['_1', '_2', '_3', '_4', '_more-than-4', '_cant-tell'],
11     'merging': ['_none', '_minor-disturbance', '_major-disturbance', '_merger']
12 }
```

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, in principle, produces a probability for each of the possible answers. This is discussed in more detail in section 8.

6.2. *ML model*

This evolved during the development of Zoobot. For W+20 and the mwalmesley/galaxy-zoo-bayesian-cnn repo, the architecture was a cut-down version of VGG16 (Simonyan & Zisserman 2014). For W+21 and mwalmesley/zoobot it had been updated to EfficientNet-B0 (Tan & Le 2019). The latter was used in the current work.

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

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, shown in more detail in Figure 2, followed by pooling and dropout. The final dense layer gives a 34-component output, corresponding to the possible answers from the volunteers.

6.3. *Loss Function*

Stage <i>i</i>	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBCConv1, k3x3	112×112	16	1
3	MBCConv6, k3x3	112×112	24	2
4	MBCConv6, k5x5	56×56	40	2
5	MBCConv6, k3x3	28×28	80	3
6	MBCConv6, k5x5	14×14	112	3
7	MBCConv6, k5x5	14×14	192	4
8	MBCConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 2. EfficientNet-B0, used as published

In W+20 volunteer responses were modeled as binomially distributed

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

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

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

The parameters are now vectors with one element per answer.

6.4. Output

Results are saved in a Tensorflow binary format, suitable for use as a starting point for making predictions. They are also used by the TensorBoard utility to make training plots as in Figure 4.

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_decalvolunteers_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 PostgreSQL.

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 3).

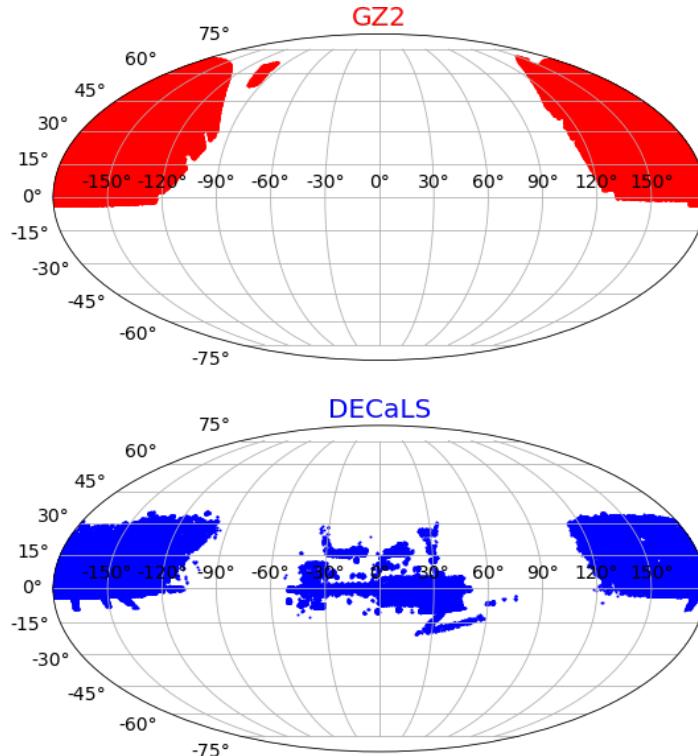


Figure 3. Sky locations of images used for each survey

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 Locally

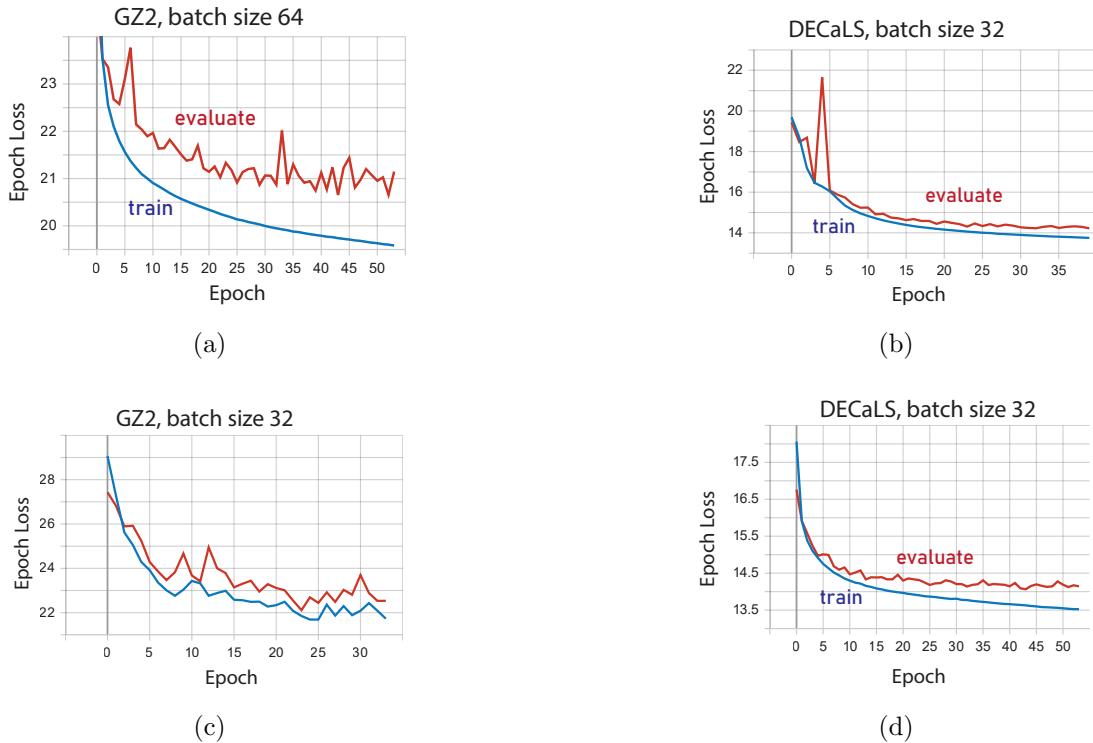


Figure 4. Training runs for GZ2 (left) and DECaLS (right)

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

GPU out-of-memory error. Dropping to batches of 64 was more successful, as in Figure 4(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 (10 epochs without improvement) after 54 epochs, 8.3 hours. However, the validation loss is noisy and suggests 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 4(b). After some initial spikes, the validation loss tracks closely with the training loss. This run failed to reach stopping criteria within the epoch limit (40 epochs, nearly 19 hours). A longer second run, shown in Figure 4(d), terminated successfully. Differences between the two runs mainly illustrate the stochastic variation in this method.

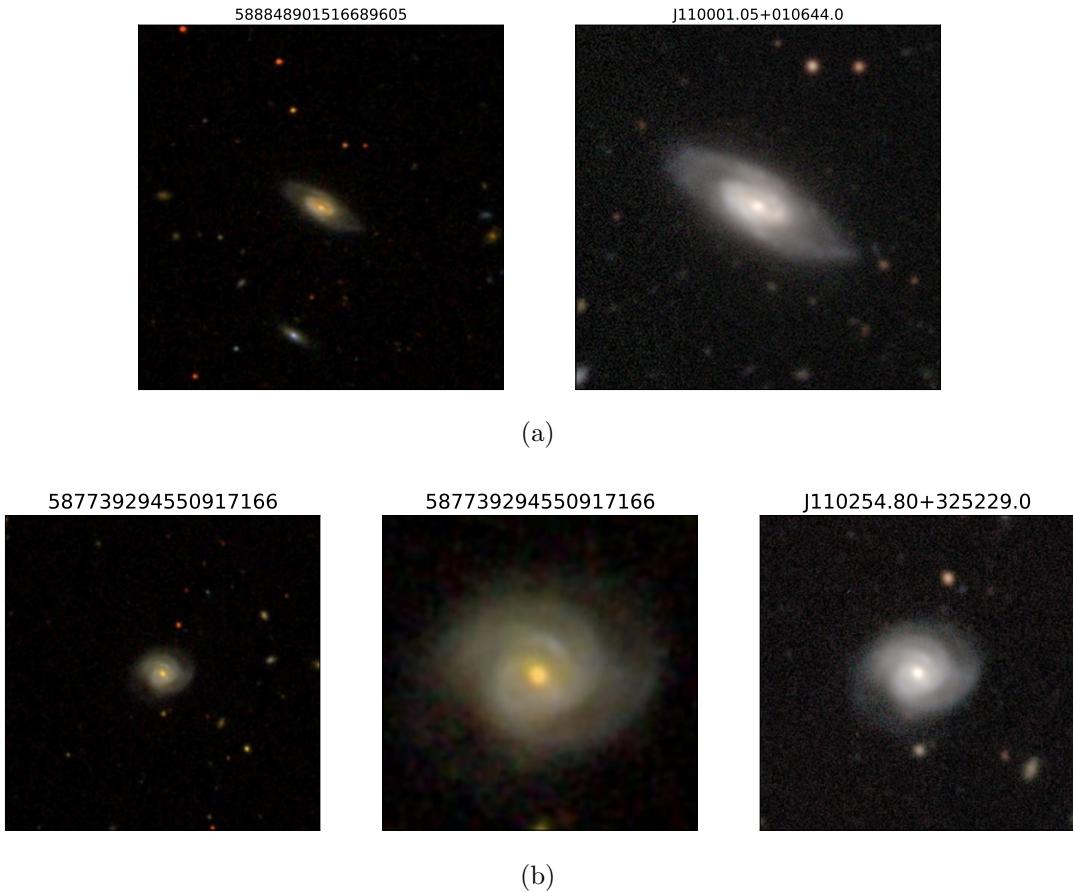


Figure 5. Raw images used for GZ2 at default resolution (left) and zoomed (center); DECaLS (right)

It was not immediately clear why the DECaLS run looks better than the GZ2 run. Preparation of data shards used 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 was easiest to test, so GZ2 training was repeated with batches of 32 as in Figure 4(c). This was not encouraging: training now looks worse without validation looking better.

Images obtained from DECaLS are inherently higher resolution and deeper than those from SDSS used in GZ2 (bigger telescope, newer camera). Indeed, ambiguous galaxies are consistently reported

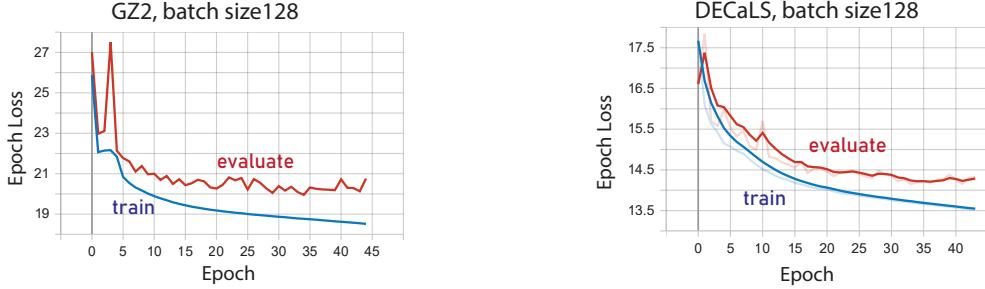


Figure 6. Colab training runs for GZ2 (left) and DECaLS (right)

by volunteers as more featured in GZ DECaLS than GZ2 (W+21, Figure 5). Training was also carried out on differently sized images: 300×300 for DECaLS and 256×256 for GZ2.

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 5. Clearly there is a major difference in quality.

By default, the SDSS cutout service supplies images at a scale of 0.4 arcsec/pixel. As a first attempt to do better, new images were downloaded at 0.1 arcsec/pixel, with the SDSS server handling any necessary pixel interpolation.

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 this is not an urgent priority for an ASTR 502 term paper. From this point only the DECaLS survey is used.

7.6. Training on Colab

All relevant files were copied to Google Drive **TODO** size, from where they could be mounted in a Colab notebook. Training with batch size 128 used around 95% of the 16GB memory. At around 15 min/epoch, both GZ2 and DECaLS took approximately 12 h to converge. Results, shown in Figure 6, are broadly similar to those obtained locally.

8. PREDICTIONS

When preparing shards as in Section 7.4, the input catalog was divided into train, validate and test sets, then each was encoded as binary tfrecord shards including the graphical image. This helped speed training but is largely pointless for predictions on the test set. Instead, this step used a CSV (or alternatively feather) file as input which included full paths to the relevant images.

For DECaLS, the test set contains around 43k galaxies. Using the model pretrained on Colab in the previous step, predictions were generated for all of these in around 11 min. Output was to an HDF5 file; CSV is also possible but less convenient.

The raw output proved surprising. For each galaxy, in addition to a full path to the image the software documentation implied there would be an array of 34 probabilities, corresponding the set of possible Q&A pairs in the GZ survey. Instead, there was a 34×5 array. From W+21, it appears that the final dropout step of the model (Figure 6.2) runs multiple times and an output is stored for each of them, giving an assessment of reproducibility. In practice, the five values generally had a small standard deviation, so all further analyses used the mean.

It is also somewhat unclear what the numbers represent. Probabilities were expected, with the options for each question adding up to 1 (or 100%). Instead the sum is highly variable and the distribution centers around roughly 70. Taking a view that the relative values are most important, everything was simply normalized to sum to 1 for clarity.

8.1. *Reviewing predictions: single galaxy*

To get a first feel for the results, a galaxy was chosen at random (only ensuring that it was one with a reasonably clear image). The prediction output is shown in Table 2. Answers with at least an 80% confidence are bolded. In all cases these agree with the consensus of the GZ volunteers, as does “spiral-winding”.

Table 2. Predictions for a single random galaxy, in descending order of probability

Question	Predicted answers (probability)
<i>smooth-or-featured</i>	featured-or-disk (0.89), smooth (0.08), artifact (0.03)
<i>disk-edge-on</i>	no (0.93), yes (0.07)
<i>has-spiral-arms</i>	yes (0.95), no (0.05)
<i>bar</i>	weak (0.44), no (0.41), strong (0.16)
<i>bulge-size</i>	moderate (0.62), small (0.27), large (0.08), none (0.02), dominant (0.01)
<i>how-rounded</i>	in-between (0.57), cigar-shaped (0.39), round (0.05)
<i>edge-on-bulge</i>	rounded (0.82), none (0.14), boxy (0.04)
<i>spiral-winding</i>	medium (0.57), loose (0.23), tight (0.20)
<i>spiral-arm-count</i>	2 (0.91), cant-tell (0.04), 1 (0.02), 3 (0.01), 4 (0.01), more-than-4 (0.01)
<i>merging</i>	none (0.83), minor-disturbance (0.12), major-disturbance (0.03), merger (0.02)

Of the other questions, the volunteers by small majorities favored no bar and small bulge, the second choices of the predictor. The remaining two show the limitations of this crude analysis: “how-rounded” is not relevant to disks and “edge-on-bulge” is only relevant if the disk is edge on. Most volunteers would therefore have bypassed these questions, as shown in the decision tree of Figure 1.

8.2. *Reviewing predictions: all galaxies*

Given the complexities of inter-dependent questions, a simple starting point would be to focus on question 1, “smooth-or-featured”, which all volunteers have to answer. An astronomer might prefer “elliptical or disk”, but the chosen wording works for a wider public⁴. Table 3 shows the confusion matrix, with agreement between volunteers and the model for 88% of the images.

Similarly, all volunteers are asked about possible mergers, with the results in Table 4. In this case there is 87% agreement. Off-diagonal values are asymmetric, with the model tending to favor some disturbance when volunteers prefer none. Note that this only uses the top-ranking choice and ignores confidence levels.

⁴ Award-winning postdoc and podcaster Dr Becky Smethurst, a co-author on several of these papers, likes the name “boring blobby things” for ellipticals. It remains to be seen whether this will help her get a faculty position.

Table 3. Confusion matrix for “smooth or featured”.

	smooth	featured-or-disk	artifact
smooth	27411	2183	159
featured-or-disk	2365	10356	95
artifact	329	54	321

Table 4. Confusion matrix for “merging”.

	none	minor-disturbance	major-disturbance	merger
none	36093	133	230	785
minor-disturbance	2145	159	123	159
major-disturbance	441	31	287	155
merger	1327	14	39	1152

8.3. Using predictions: find top-scoring galaxies

Where the prediction is uncertain between two choices, such as with 0.48 versus 0.45 probabilities, this may be a failure of the model. Perhaps more likely, it reflects genuine ambiguity in the image and a need for humans to take a closer look. Triaging images to use humans more efficiently was, of course, an original aim of this research. Also, there is some data to show that volunteers are often divided about the same ambiguous images, discussed in W+21. The GZ team classifies “confident” galaxies as those where at least 80% of the volunteers agree on a classification (Domínguez Sánchez et al. 2019), and these are just a subset of the total.

An alternative approach may better use the strengths of the model: identify the top- N galaxies confidently predicted to have a particular feature. This proved quick and easy, with a few examples shown in Figure 7. We could argue with the precise nomenclature, but this could be a simple and useful addition to survey data pipelines such as Rubin.⁵

9. FINE-TUNING AND TRANSFER LEARNING

So far, the model is useful for most but not all of the parameters in the DECaLS catalog. The next steps use this as a starting point to train other models, either to answer different questions with DECaLS, or to use other surveys as input.

9.1. Fine-tuning for DECaLS

Although the GZ DECaLS catalog contains fields for rare features such as rings, lenses and dust lanes, these are not included in the existing CNN model. Retraining the model to predict one of these features is a simple example of fine-tuning.

⁵ Other problems are harder. Several years ago I heard an IT engineer from LSST describe their hardware strategy as “delay purchase orders until the last possible moment and hope that Moore’s Law saves us”. We were both drinking beer at the time (but you already guessed that).

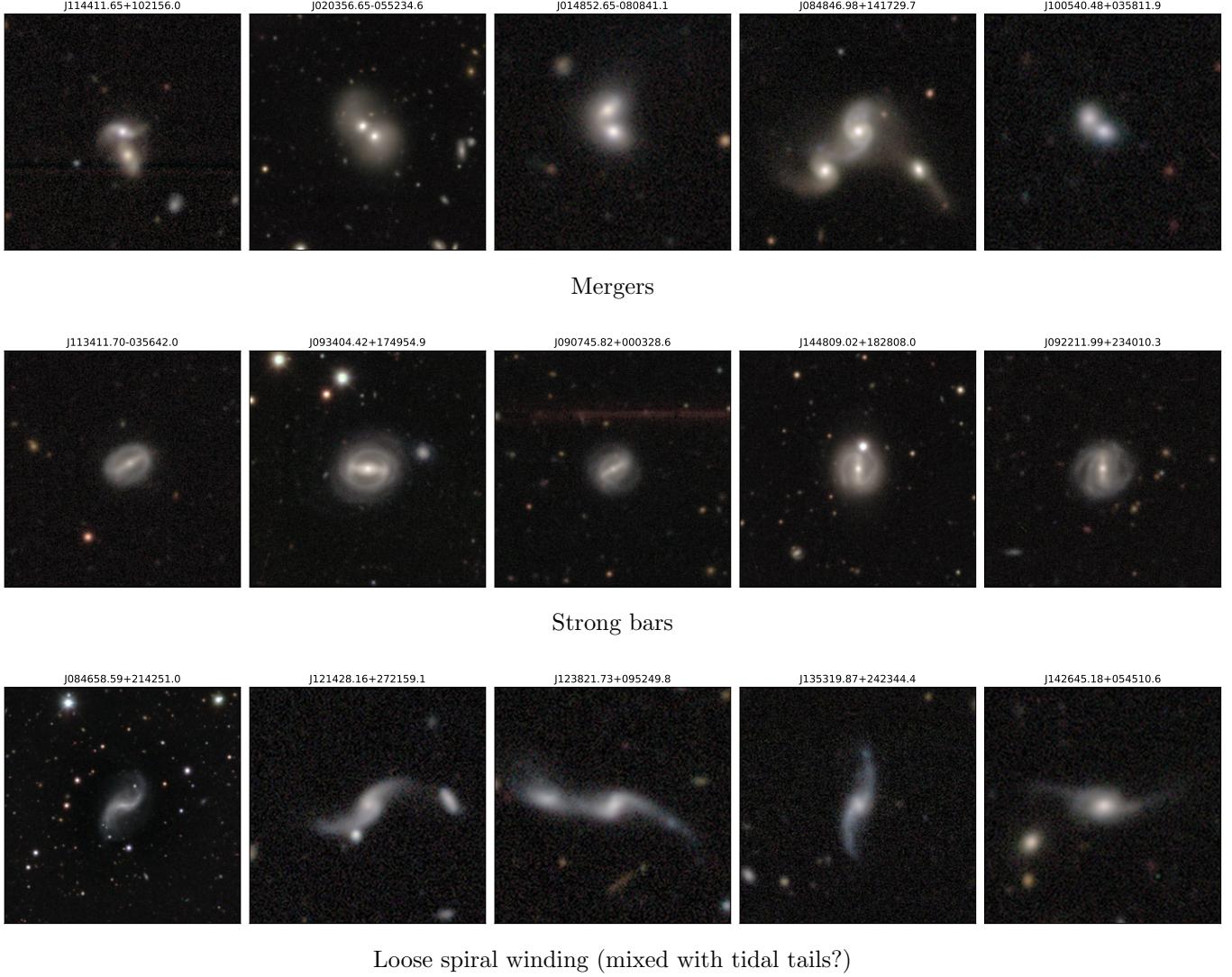


Figure 7. Top-5 prediction results for various parameters

Zoobot includes an example set of 470 images and a CSV catalog matching file locations to a ring/no-ring field, balanced to have roughly equal numbers of each. Generating such a catalog needs some effort from humans, but far less than the big GZ classifications.

These images were split into training and validation sets, the pretrained model from Section 7.6 was loaded with most of the layers frozen, then new final layers were added and the model retrained as a ring predictor. Locally, this ran with batch size 64 in less than 2 minutes, converging after 46 epochs.

Because the output is much simpler in this case, a single sigmoid value for each galaxy, the ring model is easier to understand than the full model. The loss function is a standard binary crossentropy and the accuracy on the class-balanced validation set converges to about 0.77.

The fine-tuned model was used to predict rings for the 272,725 galaxies in the DECaLS catalog: this time unbalanced, so the prior for having a ring is much less than 0.5. Running locally, this took about 15 min.

Unfortunately, the results were disappointing. Prediction confidence was < 0.75 in all cases, and on viewing the top 10 images there were no convincing rings.

Better results were reported in Walmsley et al. (2022), suggesting improvement is possible. Zoobot documentation gives a clue:

Zoobot includes two complete working examples:

`finetune_minimal.py` shows how to partially finetune a model to classify ring galaxies using an example dataset. It is designed to be easy to understand.

`finetune_advanced.py` solves the same problem with some additional tips and tricks: filtering the dataset, balancing the classes (rings are rare), and unfreezing the model.

9.2. Transfer Learning for other surveys

TODO add

10. PYTORCH CODE

TODO add

11. CONCLUSIONS

TODO add

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](https://doi.org/10.1017/S1743921319001418)
- Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441, doi: [10.1093/mnras/stv632](https://doi.org/10.1093/mnras/stv632)
- Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., et al. 2019, MNRAS, 484, 93, doi: [10.1093/mnras/sty3497](https://doi.org/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](https://doi.org/10.1002/widm.1349)
- Hart, R. E., Bamford, S. P., Willett, K. W., et al. 2016, MNRAS, 461, 3663, doi: [10.1093/mnras/stw1588](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.5281/zenodo.2677874)
- Walmsley, M., Smith, L., Lintott, C., et al. 2020a, \mnras, 491, 1554, doi: [10.1093/mnras/stz2816](https://doi.org/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](https://doi.org/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](https://doi.org/10.1093/mnras/stab2093)
- . 2022, \mnras, 509, 3966,
doi: [10.1093/mnras/stw2568](https://doi.org/10.1093/mnras/stw2568)
- 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](https://doi.org/10.1093/mnras/stt1458)
- Willett, K. W., Galloway, M. A., Bamford, S. P.,
et al. 2017, MNRAS, 464, 4176,
doi: [10.1093/mnras/stw2568](https://doi.org/10.1093/mnras/stw2568)