

# Galaxy Zoo Morphology Classifications by CNN

Colin Leach

# Why?

## **Rapid growth in survey data, static number of astronomers**

- Citizen Science helped fill the gap over the last 20 years
- Sufficient for SDSS, will struggle with DES, Rubin, etc
- Need to use machines for a first-pass analysis
- Save humans for analyzing the difficult cases

## **Make machine learning more efficient**

- Galaxy Zoo has already classified >300k galaxy images
- This work created pre-trained CNN models for these
- A better starting point than ImageNet for new surveys!
- Hope to train a new survey on far fewer human classifications

# Papers

MNRAS **000**, 1–23 (2019)

Preprint 7 October 2019

Compiled using MNRAS L<sup>A</sup>T<sub>E</sub>X style file v3.0

## Galaxy Zoo: Probabilistic Morphology through Bayesian CNNs and Active Learning

Mike Walmsley<sup>1\*</sup>, Lewis Smith<sup>2</sup>, Chris Lintott<sup>1</sup>, Yarin Gal<sup>2</sup>, Steven Bamford<sup>3</sup>, Hugh Dickinson<sup>4,8</sup>, Lucy Fortson<sup>4,8</sup>, Sandor Kruk<sup>5</sup>, Karen Masters<sup>6,7</sup>, Claudia Scarlata<sup>4,8</sup>, Brooke Simmons<sup>9,10</sup>, Rebecca Smethurst<sup>1</sup>, Darryl Wright<sup>4,8</sup>

MNRAS **491**, 1554 (2020)

(W+20)

MNRAS **000**, 1–20 (2021)

Preprint 4 January 2022

Compiled using MNRAS L<sup>A</sup>T<sub>E</sub>X style file v3.0

## Galaxy Zoo DECaLS: Detailed Visual Morphology Measurements from Volunteers and Deep Learning for 314,000 Galaxies

Mike Walmsley<sup>1\*</sup>, Chris Lintott<sup>1</sup>, Tobias Géron<sup>1</sup>, Sandor Kruk<sup>2</sup>, Coleman Krawczyk<sup>3</sup>, Kyle W. Willett<sup>4</sup>, Steven Bamford<sup>5</sup>, Lee S. Kelvin<sup>6</sup>, Lucy Fortson<sup>7</sup>, Yarin Gal<sup>8</sup>, William Keel<sup>9</sup>, Karen L. Masters<sup>10</sup>, Vihang Mehta<sup>9</sup>, Brooke D. Simmons<sup>11</sup>, Rebecca Smethurst<sup>1</sup>, Lewis Smith<sup>8</sup>, Elisabeth M. Baeten<sup>12</sup>, Christine Macmillan<sup>12</sup>

MNRAS **509**, 3966 (2021)

(W+21)

*Better methods, better code,  
much better documentation*

# Links

## Papers

- <https://ui.adsabs.harvard.edu/abs/2020MNRAS.491.1554W/abstract> (W+20)
- <https://ui.adsabs.harvard.edu/abs/2022MNRAS.509.3966W/abstract> (W+21)

## Mike Walmsley repos

- For W+20: <https://github.com/mwalmsley/galaxy-zoo-bayesian-cnn> (static)
- For W+21: <https://github.com/mwalmsley/zoobot> (active development)

## My repo

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

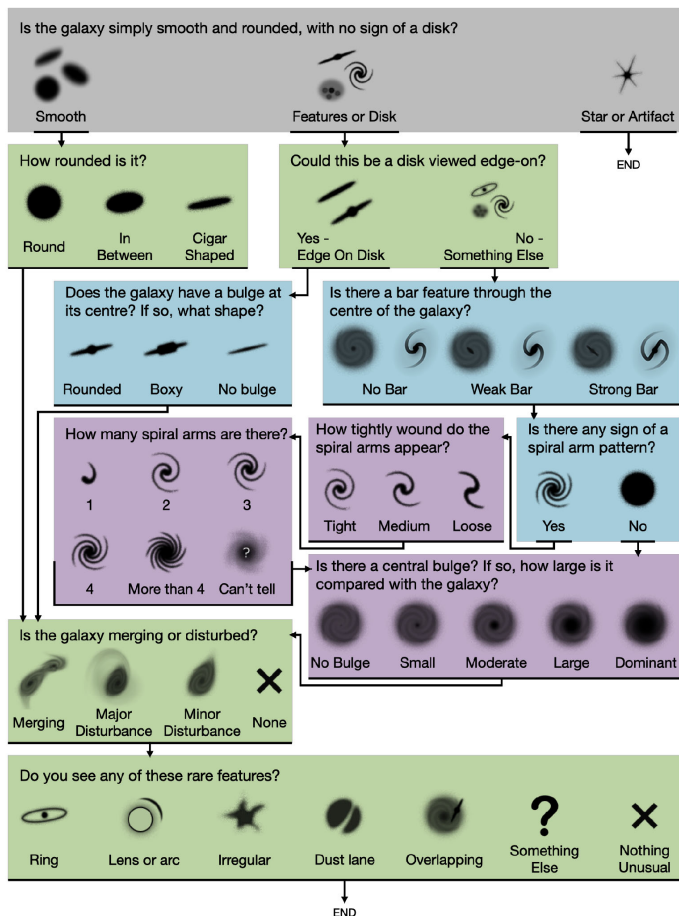
# Objectives (original)

- Get the published code running on my local machine, using whatever cut-down training set proves viable.
- Deploy the code on either AWS or Google.
- Extend the model to other data such as Hubble, CANDELS, DECaLS, for which there is already some GZ classification.
- Think about newer CNN algorithms (*W+20 used VGG16*)
- Rewrite using other frameworks, for my education:
  - PyTorch
  - Julia with Flux
  - (*F# with ML.NET, but don't hold your breath waiting for that!*)

# Objectives (modified)

- Get the published code running on my local machine, using whatever cut-down training set proves viable.
- Deploy the code on either AWS or Google.
- Use GZ2 and DECaLS data initially, perhaps others later
- ~~Think about newer CNN algorithms~~ Stick with EfficientNet B0
- Repeat with PyTorch (*code from branch of mwalmsley/zoobot*)
- **Later**, rewrite using other frameworks, for my education:
  - Julia with Flux
  - (*F# with ML.NET, but don't hold your breath waiting for that!*)

# Galaxy Morphology Classification



```
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']
}
```

- **10 Questions**
- **34 possible answers**
- Model is trying to predict probability of getting each answer (*Bayesian posterior*)
- **Output is 34 (arrays of) floating-point numbers between 0 - 100**

# Hardware and Runtime Environments

- Any 2GB GPU (GTX 1050, MX 230)
  - Immediate out-of-memory error (for *virtually anything*)
- 6GB GTX 1660
  - \$450 purchase, old design but compatible with my Linux PC
  - Limited capability for training
  - Good for debugging, and predictions with pre-trained model
- Colab Pro+ with 16GB Tesla V100
  - Can train at recommended batch size (*just, barely*)
- AWS EC2? Colab TPU?
  - Will try these later



# Galaxy Images and Catalogues

- DECaLS : Everything is easily available on Zenodo

- <https://zenodo.org/record/4573248>

- GZ2 : Catalog is online

- <https://data.galaxyzoo.org>

- GZ2 images are not made public, need to get them from SDSS

```
sdss_url = http://skyserver.sdss.org/dr14/SkyServerWS/ImgCutout/getjpeg  
img_url = sdss_url + f"?ra={ra:.5f}&dec={dec:.5f}&width={jpeg_size}&height={jpeg_size}"  
img_data = requests.get(img_url).content
```

- This isn't the end of the story...

# CNN Model

- W+20 used a cut-down VGG16 (*cheaper to run*)
- W+21 and **this work** use EfficientNet-B0
  - The PyTorch branch recently added a ResNet50 option (less good, more familiar)

**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

EfficientNet-B0

Stage $i$	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	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

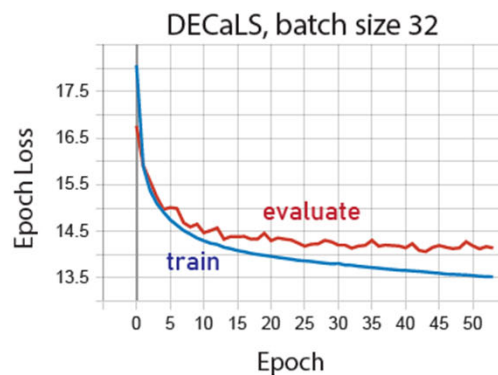
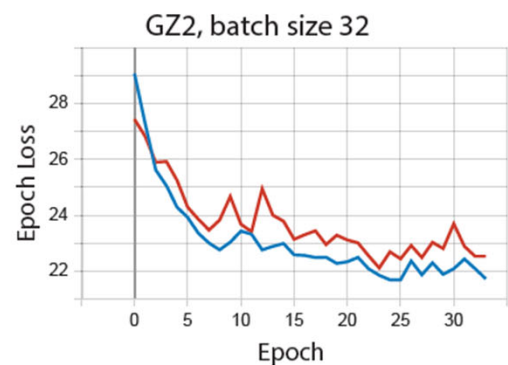
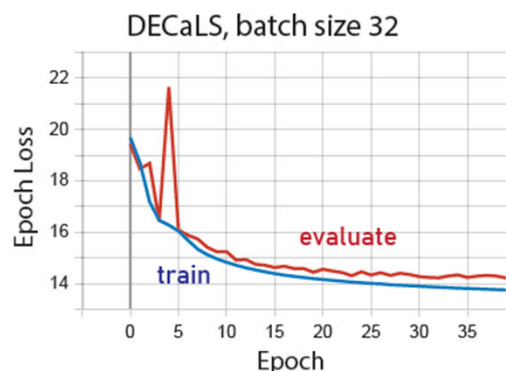
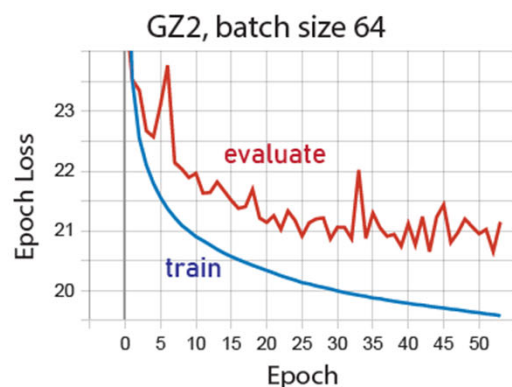
Custom Loss function:  $\mathcal{L} = \int \text{Multi}(\vec{k}|\vec{\rho}, N) \text{Dirichlet}(\vec{\rho}|\vec{\alpha}) d\vec{\alpha}$

Adam optimizer

**No good accuracy function available**

<https://arxiv.org/abs/1905.11946v5>

# Training with Keras/TensorFlow (local)



- Started with batch size 128, kept halving until training ran without OOM errors
- Worked consistently better with DECaLS than GZ2
- Matching batch size only makes GZ2 worse
- Time to take another look at image quality?

# Images, the Sequel

## Why is GZ2 training so poor compared with DECaLS?

- Different training methods (batch size, image size)?
- Different image quality?

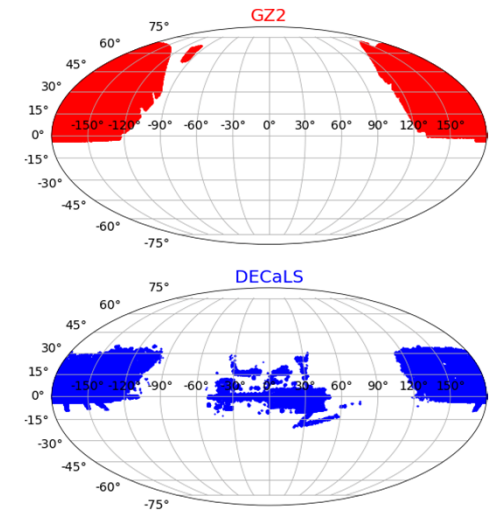
## Need examples of the same galaxy in both surveys

- No common ID field
- Matched on RA and Dec values (in PostgreSQL)
- Found 132,732 matches
- Biggest difference is scaling (ignore color)

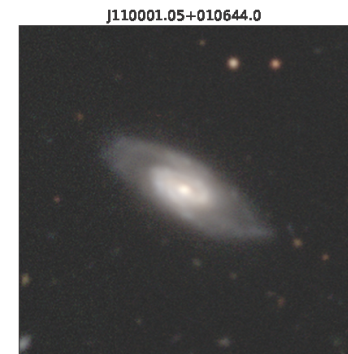
After reading the documentation (!!), a new set of SDSS images were downloaded with 4x zoom-in

- 0.1 vs default 0.4 arcsec/pixel

**Garbage in, Garbage out – a *very* old idea**

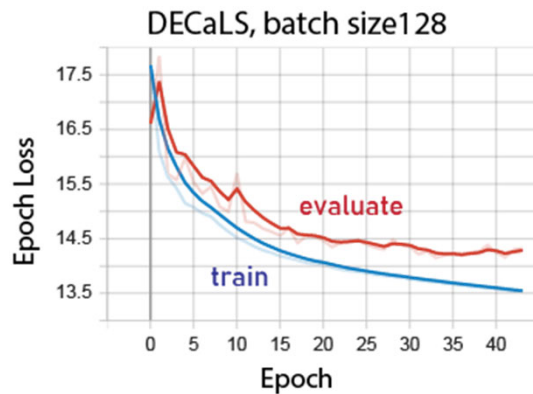
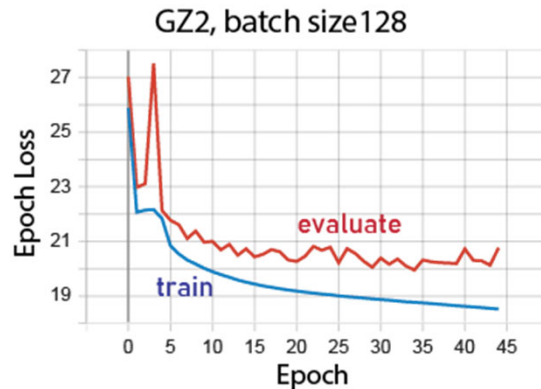


GZ2



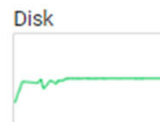
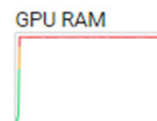
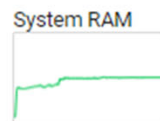
DECaLS


# Training on Colab



- All files stored on Drive
- Training takes >12h on a V100, so eventually upgraded to Colab Pro+ (\$50/month)
- Even a 16GB GPU nearly ran out of memory
- Results are a bit better than on my home PC
  - not as much as hoped for

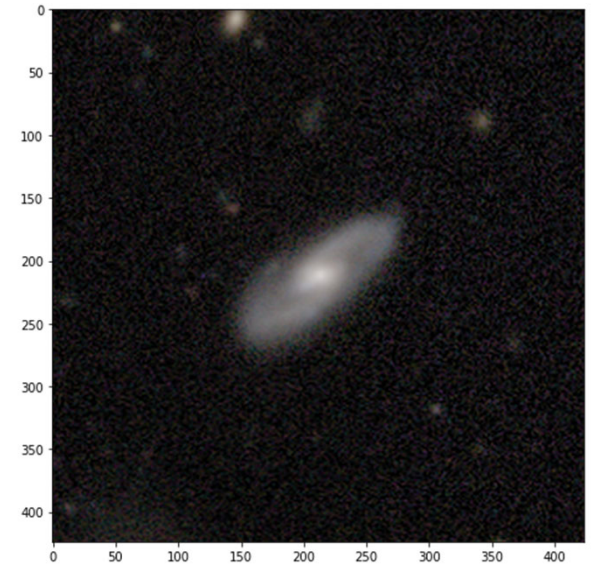
Python 3 Google Compute Engine backend (GPU)  
Showing resources since 10:51 AM



Title		Last execution	RAM used	GPU used	
	TrainGZ2.ipynb Current session, Background execution	GPU	0 minutes ago	5.29 GB	15.14 GB

# Predictions

- Relatively fast, not hardware-intensive
- Expected each result shape to be (34,), got (34, 5)!
- It *seems* to store results from different dropouts in penultimate model layer
- StdDev is small so I just worked with the mean
- Numbers still a bit odd?
- Take highest-scoring response to each question, normalize probability



	question	pred	pred_confidence	obs	obs_confidence
✓	smooth-or-featured	featured-or-disk	0.89	featured-or-disk	1.0
✓	disk-edge-on	no	0.93	no	1.0
✓	has-spiral-arms	yes	0.95	yes	1.0
✗	bar	weak	0.44	no	0.6
✗	bulge-size	moderate	0.62	small	0.8
✗	how-rounded	in-between	0.57	round	NaN
✗	edge-on-bulge	rounded	0.82	boxy	NaN
✓	spiral-winding	medium	0.57	medium	0.6
✓	spiral-arm-count	2	0.91	2	1.0
✓	merging	none	0.83	none	0.8

# Prediction analysis

Confusion matrix (one of 10)

smooth-or-featured	predicted		
	smooth	-or-disk Featured	artifact
	smooth	27411	2183
	featured-or-disk	2365	10356
	artifact	329	54
volunteers			
321			

Accuracy by question  
*(very rough first-pass analysis)*

Question	Agree
smooth-or-featured	88%
disk-edge-on	72%
has-spiral-arms	58%
bar	52%
bulge-size	42%
how-rounded	80%
edge-on-bulge	25%
spiral-winding	76%
spiral-arm-count	25%
merging	87%



# Transfer Learning and Fine-Tuning

- Extend the model to new tasks
  - Other surveys (e.g. Rubin, Euclid)
  - Other features: rings, dust lanes, *etc*
- Start from published weights
- Freeze most of the model layers
- Train the last few layers on limited new data, order  $10^2$  not  $10^5$

<https://arxiv.org/abs/2110.12735>



(a) Random selection from the top 200 DECaLS galaxies identified as mergers with our method



(b) Random selection from the top 200 DECaLS galaxies identified as rings with our method



(c) Random selection from the top 200 DECaLS galaxies identified as irregulars with our method



## Still To Do (next few weeks)...

- Transfer Learning
- Switch to PyTorch and see what already works
  - Probably need to write some new code

### Housekeeping:

- Tidy up GitHub repo
- Submit some small PRs upstream
- Finish report

*But I'm not looking for a grade*

# Summary

- Galaxy Zoo has published >300k detailed classifications
- GZ team members have published multiple CNN models, both Keras and PyTorch
- Using these as a starting point for new surveys and searches should be efficient
- I have reproduced some of the published methods and added notebook-based code
- It's a public resource, please use it if appropriate

# Summary

- Galaxy Zoo has published >300k detailed classifications
- GZ team members have published multiple CNN models, both Keras and PyTorch
- Using these as a starting point for new surveys and searches should be efficient
- I have reproduced some of the published methods and added notebook-based code
- It's a public resource, please use it if appropriate

