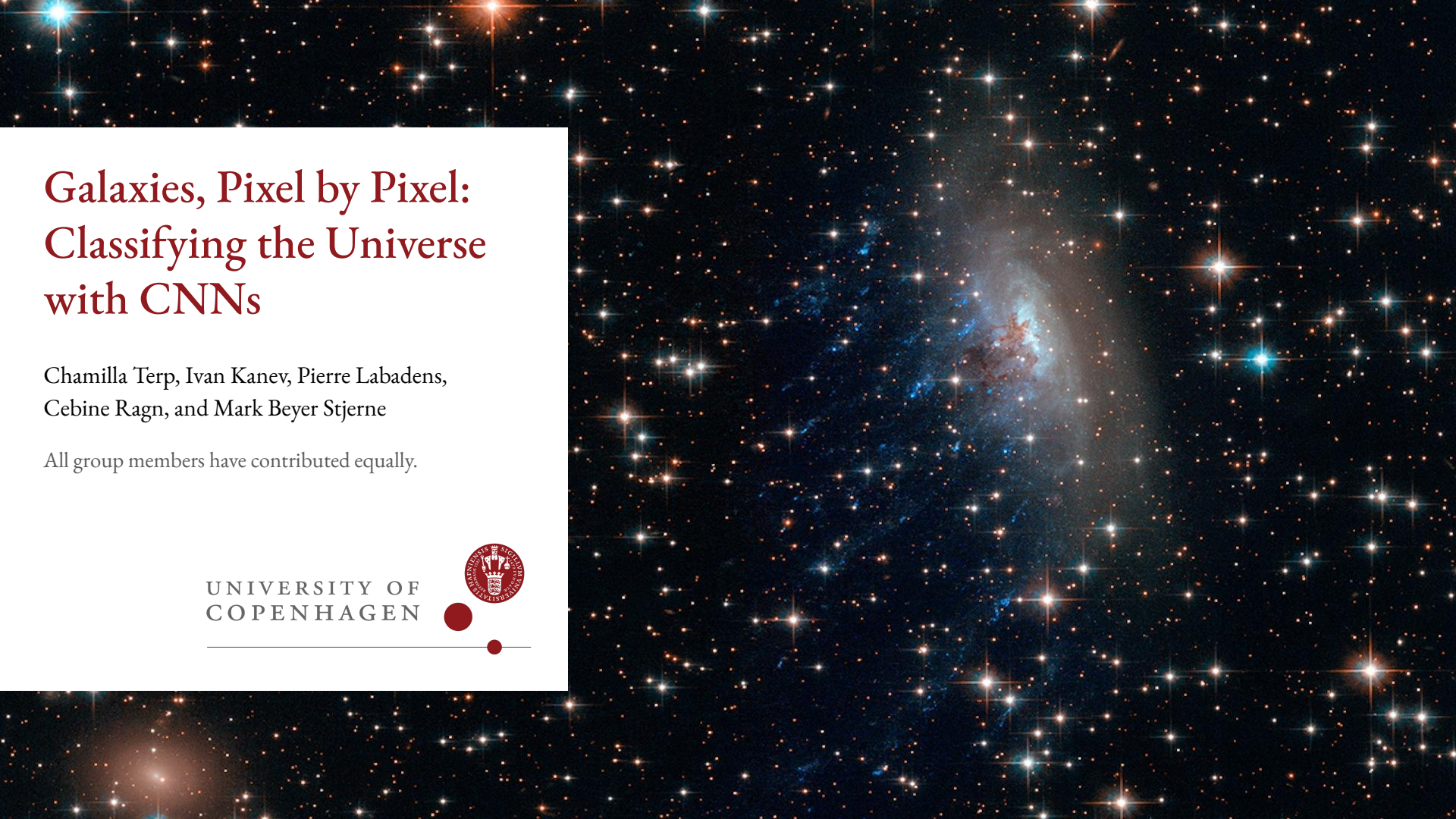
Galaxies, Pixel by Pixel: Classifying the Universe with CNNs

Chamilla Terp, Ivan Kanev, Pierre Labadens, Cebine Ragn, and Mark Beyer Stjerne

All group members have contributed equally.

UNIVERSITY OF COPENHAGEN

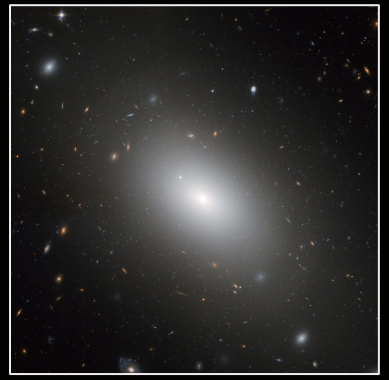


Background

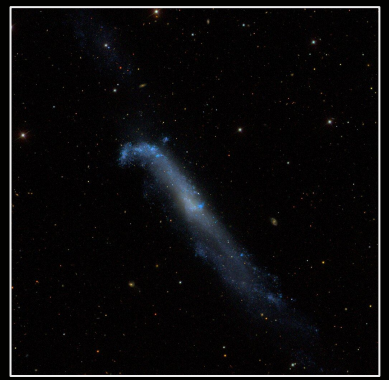
- Galaxies have complicated, individual morphologies
- CNNs offer a promising approach to image-based pattern recognition
- Galaxy surveys offer a wealth of image data, and citizen science projects allow us to label these galaxies and their features

Question: Can we train a model that can **classify** or **identify features** from survey images of galaxies?

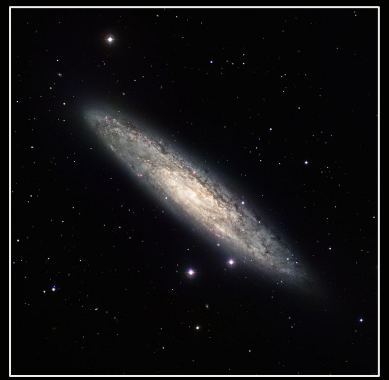
Spiral



Elliptical



Irregular



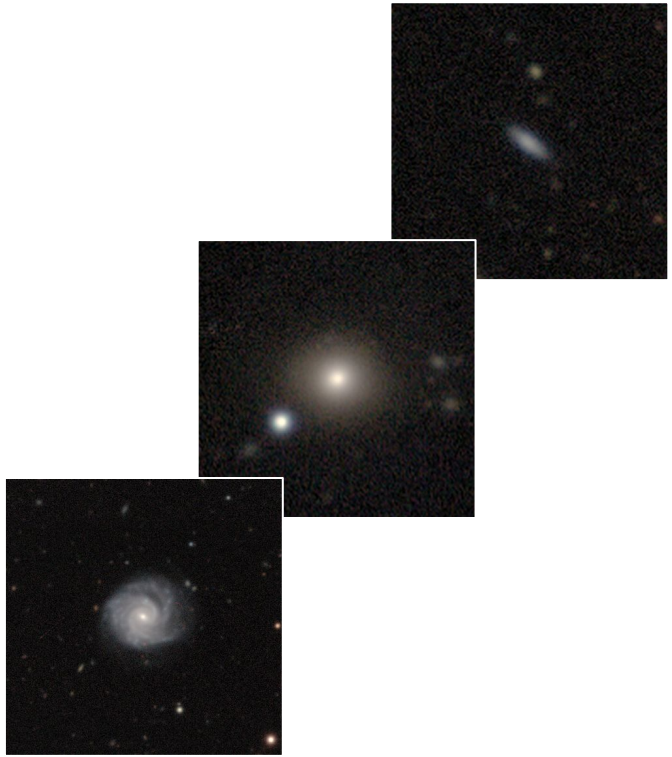
Lenticular/Disc

Data: DECaLS

- **D**ark **E**nergy **Ca**mera **L**egacy **S**urvey
  - Data Release 5 (DR5)
- **Dataset**: 253,286 survey images of galaxies
  - Image size: 424 x 424 x 3 (RGB)
  - Pre-cropped and centered on objects

Challenges regarding the data:

- Varying image quality and noise levels
- Small class imbalance
  - Elliptical and spiral galaxies seemingly more common
- Some images may contain nearby stars or multiple objects



🔗 legacysurvey.org/decamls/

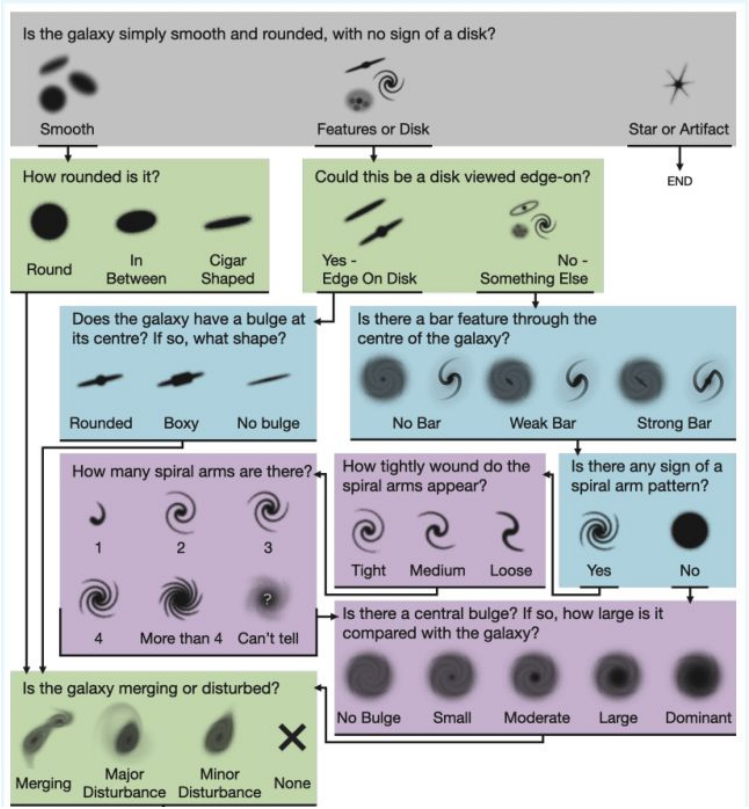
Data: Galaxy Zoo

What is *Galaxy Zoo*?

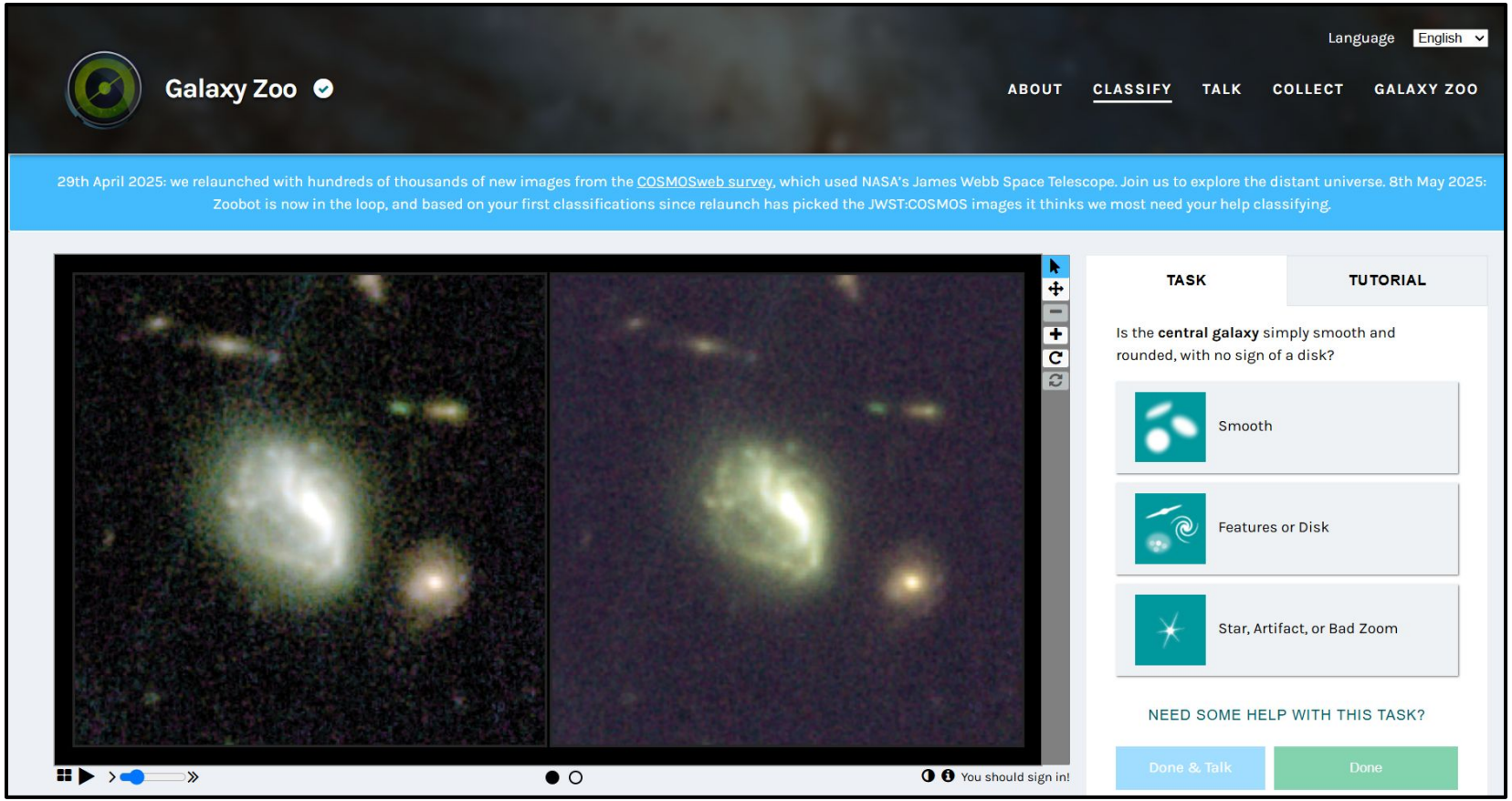
- Citizen science project that enlists online volunteers to visually classify galaxies
- Galaxy Zoo DECaLS 5 (GZD-5) campaign
  - Provides effective 'labels' for DECaLS image data

Classifications

- Completed by volunteers using a decision tree question schema
- **Dataset**: Number of votes for each question and choices for DECaLS galaxies
- Total votes per galaxy cluster around 5 or 40*
  - Some images with high ML potential 'promoted'



*See Appendix A.



🔗 zooniverse.org/projects/zookeeper/galaxy-zoo/

Data Preprocessing

1. Filtering by votes for reliability
   - Removing galaxy entries that received very few volunteer votes, which could make their classification unreliable, improving **label quality**
2. Selecting relevant classification labels
   - Keeps only the subset of label columns needed for particular approach
3. Removing NaN values
   - Ensures that only galaxies with **complete and usable labels** are included → avoids training on incomplete or ambiguous label data
4. Matching image files to labels
   - Drops galaxies with missing image files. Important since the data contains images from all Galaxy Zoo campaigns
5. Resizing images to 224 x 224 pixels
   - Reduces array size and decreases training time

Goals

Two Approaches

1. **Multi-class classification:** Classify galaxies into broad morphological types, e.g. *spiral*, *elliptical*, *irregular*, *lenticular*

2. **Multi-label classification:** Predict detailed multi-label features such as bar strength and disk orientation

Unlike multi-class classification, where each image belongs to **only one class**, multi-label classification allows **multiple labels** per image.

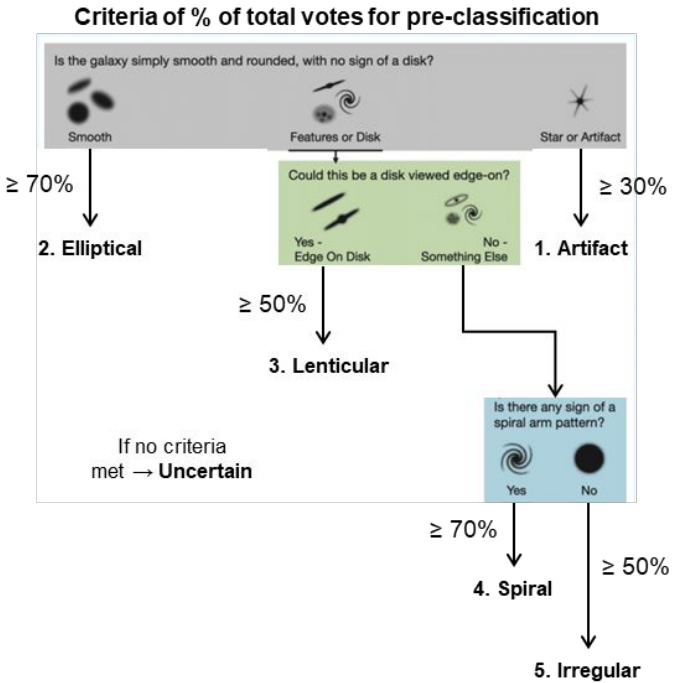
Multi-class Classification

Multi-class Classification, Classic CNN

- What if we only fed images to the CNN that have a **confident** volunteer consensus?
- Goal: Classify images into one of **four** classes*

Further preprocessing

1. Mapping GZ5-D votes to hard class labels based on custom confidence thresholds (to the right)
   - This was based off seeing at what threshold, that a batch of sample images would appear homogeneous
2. Images classified as uncertain/artifact removed from training set → Avoids training on an *ambiguous* class
3. Large class imbalance → Data augmentation to the rescue! (rotating and flipping until balanced)

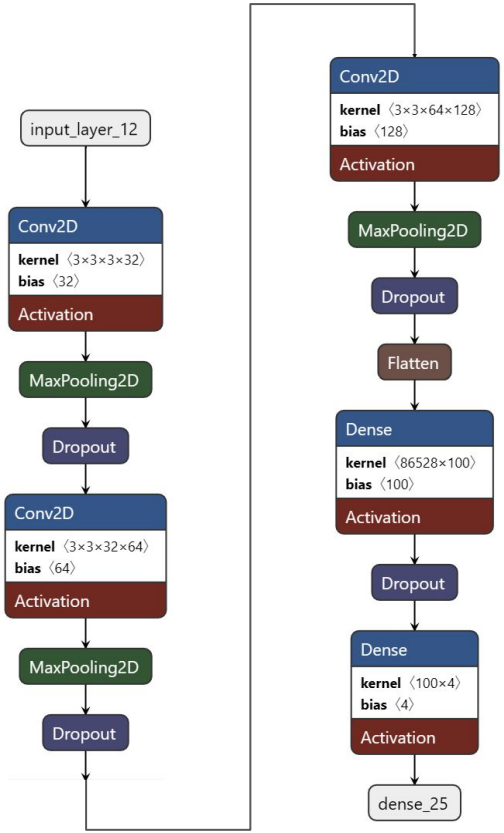


*Provided that the maximum class prediction probability exceeded a threshold.

Multi-class Classification, Classic CNN

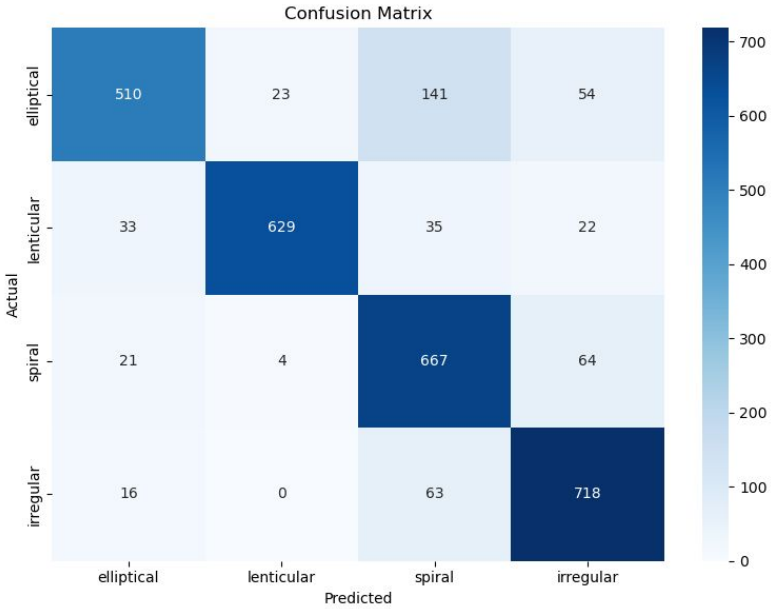
Design & Training

- Self-built CNN using Keras Sequential stacking
  - 3 × 2D Convolutional layers
  - Dropout layers (25%)
- Adam optimiser, learning rate of 0.0001
- Run for 20 epochs
- Class-balanced training sample of 9000 galaxies and a 25% testing set (3000 galaxies)

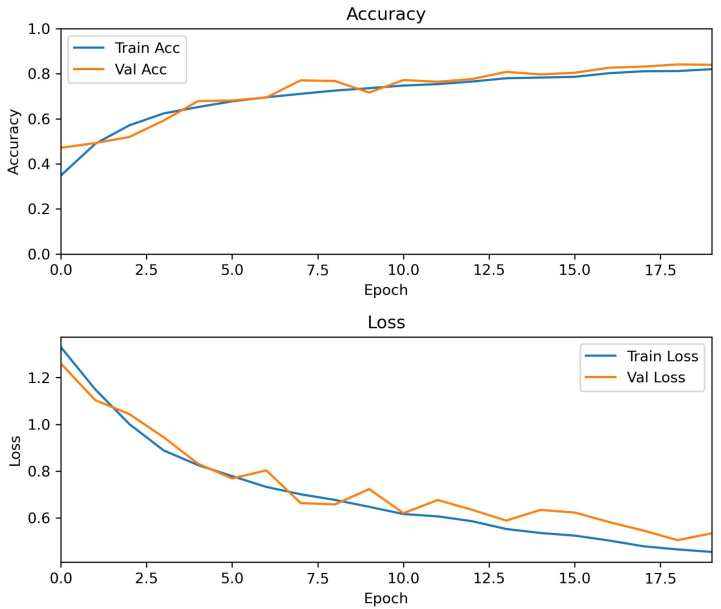


Multi-class Classification, Classic CNN

Performance



Taken from a 20% testing set derived from augmented 4-class data.

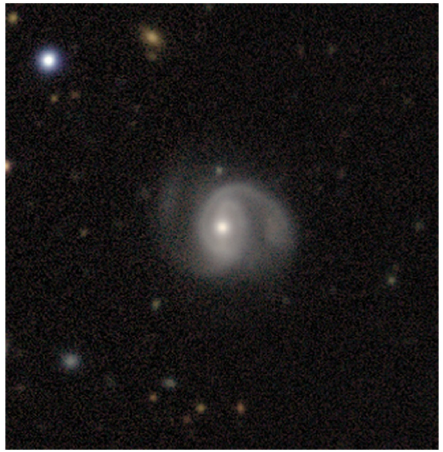


Doesn't appear to be overfitting!

Multi-class Classification, Classic CNN

Output Examples

J004507.95+002116.0



irregular: 0.04 (True vote: 0.00)
spiral : 0.96(Truevote : 0.93)
lenticular: 0.00 (True vote: 0.03)
elliptical: 0.00 (True vote: 0.03)

J024227.38-075931.0

irregular: 0.03 (True vote: 0.10)
spiral: 0.29 (True vote: 0.00)
lenticular: 0.01 (True vote: 0.00)
elliptical : 0.67(Truevote : 0.78)

J015144.90+010544.4

irregular : 0.61(Truevote : 0.23)
spiral: 0.07 (True vote: 0.00)
lenticular: 0.05 (True vote: 0.08)
elliptical: 0.27 (True vote: 0.67)

J100355.04+072645.4

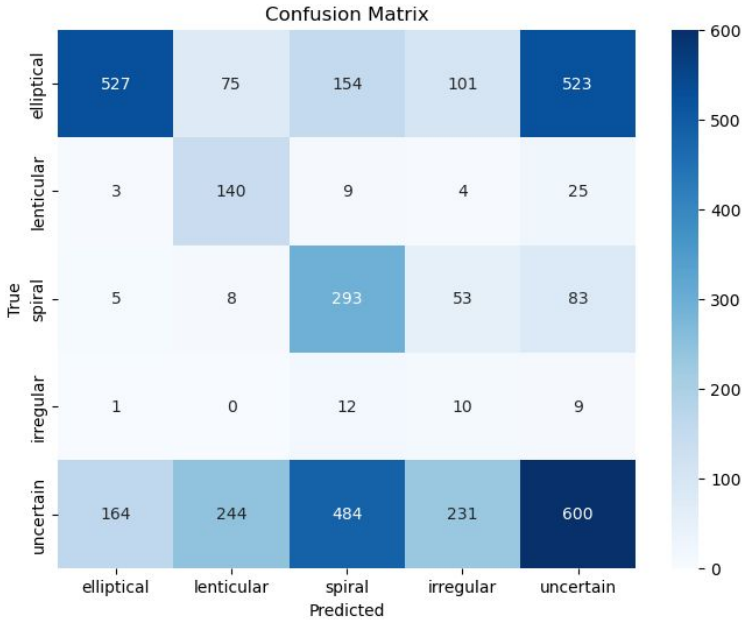
irregular: 0.08 (True vote: 0.05)
spiral : 0.55(Truevote : 0.03)
lenticular: 0.00 (True vote: 0.00)
elliptical: 0.37 (True vote: 0.84)

Results: Very confident on clearly spiral, but some elliptical/irregular images have a drop in confidence.

Multi-class Classification, Classic CNN

Caveats

- Performance on common validation set
  - Defining ‘uncertain’ to be those with not a high confidence in any particular class (<0.6)
  - Model has capabilities to identify morphological features (according to testing set results), *but* the validation set has a high uncertain population
  - These are reflected in the ‘uncertain’ category
- Fault of the encoding of noisy fractional data into hard labels
  - Future model could keep softmax-type input, but lose hard class definition



Multi-label Classification

Multi-label Classification

Further Data Processing

- Resized images to 224 x 224 to standardize input size
- Converted images to **PyTorch tensors** and **normalized**
- Labels selected → binary, debiased
  - Smooth vs. featured
  - Edge-on disk
  - Bar strength
  - Spiral arms

Keep in mind!
Approach ignores the fact that some labels *are* in fact mutually exclusive

Final dataset:

Clean set of galaxy images + 10-label multi-label targets

Multi-label Classification

Model Architecture: Simple Convolutional Neural Network

Model

- 3 convolutional layers
  - 3 x 3 filters, ReLU activation, max pooling
- Dropout layer (50%) to prevent overfitting
- 2 fully connected layers
  - Output layer: 10 neurons for multi-label classification

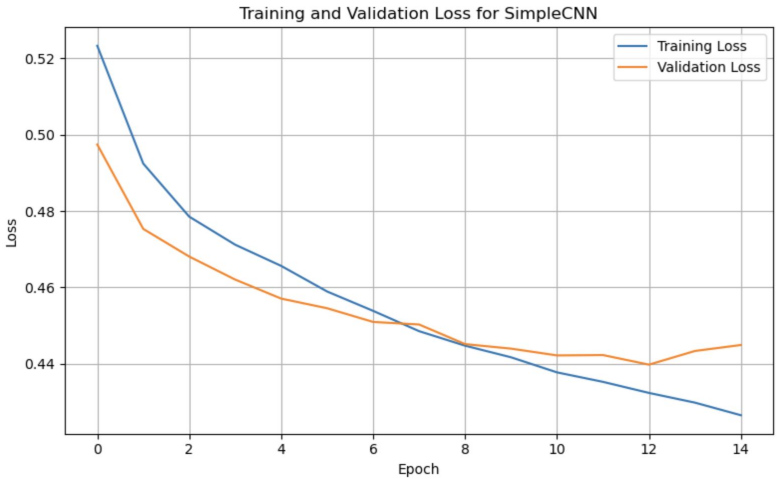
- Encapsulated in a LightningModule
- "True" values are in this model, all debiased fractions over a threshold of 0.5

Activation & Loss Function

- Sigmoid Activation
- BCEWithLogitsLoss

Combines sigmoid + binary cross-entropy, and treats each label as a *separate binary classification*

Multi-label Classification



Loss is averaged across batches per epoch for both training and validation sets.

Training Performance

- Loss decreases steadily over epochs
  - Slight flattening after ~8 epochs
- No clear overfitting
  - However, more epochs seemed to make the validation loss increase

- Model learns meaningful patterns from data
- Stable training behavior

Multi-label Classification

Evaluation Metrics

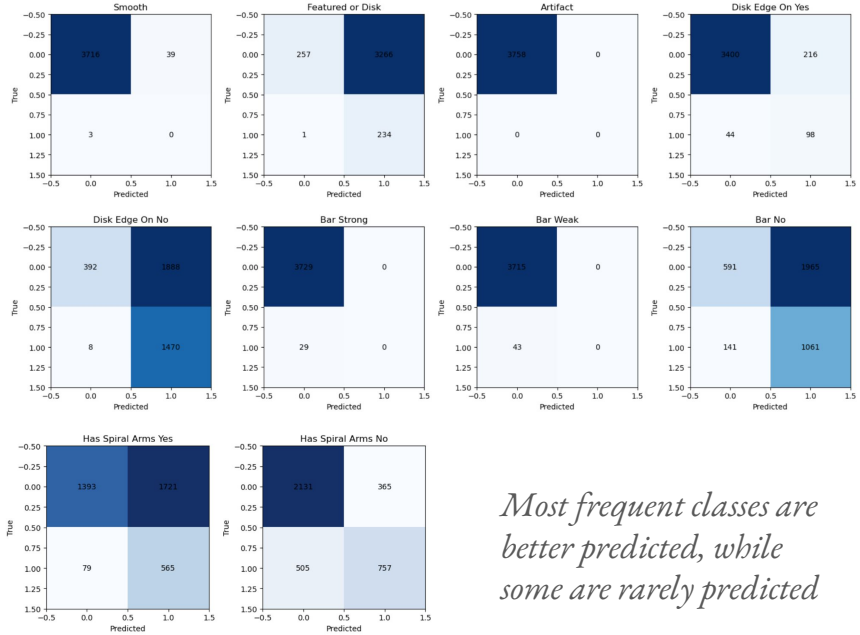
```
🔍 Validation Metrics:
Accuracy: 0.0319
Precision (Macro): 0.2089
Recall (Macro): 0.5040
F1 Score (Macro): 0.2686
F1 Score (Micro): 0.4480
```

⚠️ Very low overall accuracy → common in multi-label setups with imbalanced classes

💡 High recall → catches many true positives

❌ Low precision → often guess labels that aren't there

Confusion Matrices



Most frequent classes are better predicted, while some are rarely predicted

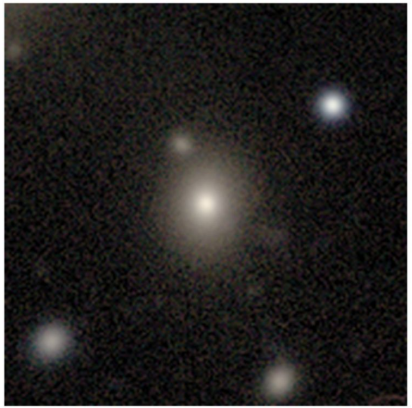
Multi-label Classification

Output Examples

J100355.04+072645.4

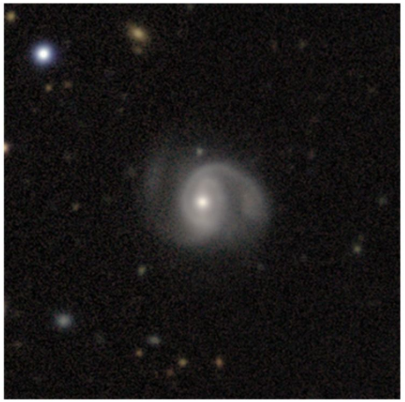
smooth-or-featured_featured-or-disk: Predicted ✅, True ❌
disk-edge-on_no: Predicted ✅, True ❌
bar_weak: Predicted ❌, True 🔷
bar_no: Predicted ✅, True ❌
has-spiral-arms_yes: Predicted ✅, True 🔷

J024227.38-075931.0



smooth-or-featured_featured-or-disk: Predicted ✅, True ❌
disk-edge-on_no: Predicted ✅, True ❌
bar_no: Predicted ✅, True ❌
has-spiral-arms_no: Predicted ✅, True ❌

J004507.95+002116.0



smooth-or-featured_featured-or-disk: Predicted ✅, True ❌
disk-edge-on_no: Predicted ✅, True ❌
has-spiral-arms_yes: Predicted ✅, True ❌

J015144.90+010544.4

smooth-or-featured_featured-or-disk: Predicted ✅, True ❌
smooth-or-featured_artifact: Predicted ❌, True 🔷
disk-edge-on_no: Predicted ✅, True ❌
bar_strong: Predicted ❌, True 🔷
bar_no: Predicted ✅, True ❌
has-spiral-arms_no: Predicted ✅, True ❌

Multi-class Classification v.2

Hierarchical

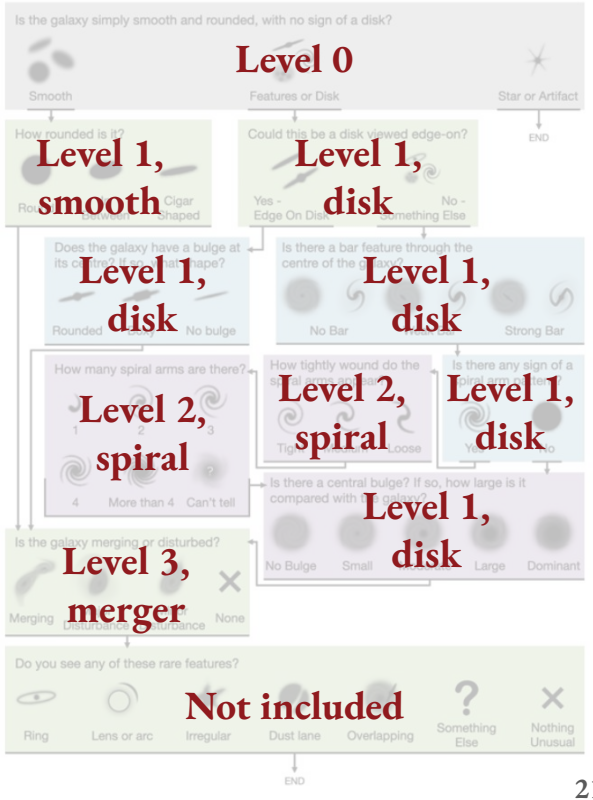
Different Approach: Hierarchical Multi-class Classification

Procedure

- Follow the structure of the decision tree question schema
- Using same simple CNN as before
  - Training it at each level and each "path" → 5 CNN models
- Making hierarchic predictions based on each levels predictions

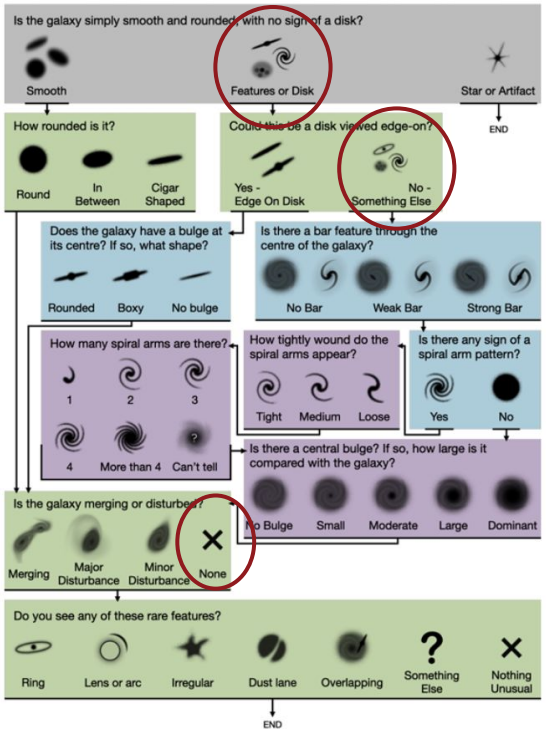
Data Processing

- Filtering the training data based on level, require a minimum of votes, dropping images with missing answers



Different Approach: Hierarchical Multi-class Classification


```
Level 0 prediction: smooth-or-featured_featured-or-disk_debiased
Level 0 probabilities:
  smooth-or-featured_smooth_debiased: 0.657
  smooth-or-featured_featured-or-disk_debiased: 0.763
  smooth-or-featured_artifact_debiased: 0.011
Level 1 branch: disk
Level 1 top predicted class: disk-edge-on_no_debiased (index 1)
Level 1 probability: 0.971
Level 2 branch: merging
Level 2 top predicted class: merging_none_debiased (index 0)
Level 2 probability: 0.870
--------------------------------------------------
```



Different Approach: Hierarchical Multi-class Classification

Output Examples

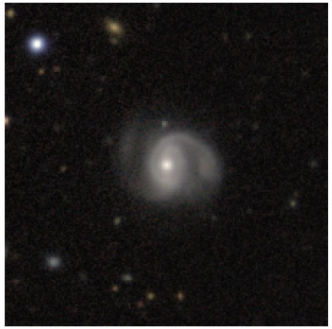
J100355.04+072645.4

Level 0: smooth-or-featured_featured-or-disk_debiased, Conf.: 0.87
Level 1 (disk): disk-edge-on_no_debiased, Conf.: 0.97
Level 2 (merging): merging_none_debiased, Conf.: 0.88

J024227.38-075931.0

Level 0: smooth-or-featured_smooth_debiased, Conf.: 0.75
Level 1 (smooth): how-rounded_round_debiased, Conf.: 0.67
Level 2: None

J004507.95+002116.0



Level 0: smooth-or-featured_featured-or-disk_debiased, Conf.: 0.89
Level 1 (disk): disk-edge-on_no_debiased, Conf.: 0.97
Level 2 (merging): merging_none_debiased, Conf.: 0.87

J015144.90+010544.4

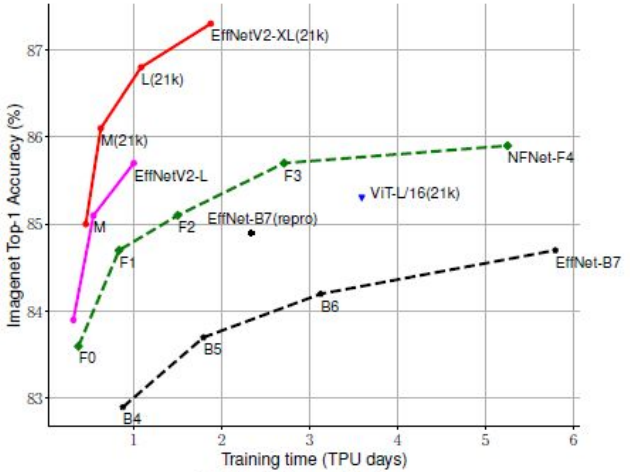
Level 0: smooth-or-featured_smooth_debiased, Conf.: 0.79
Level 1 (smooth): how-rounded_round_debiased, Conf.: 0.75
Level 2: None

Fine-tuning an existing model

EfficientNetV2

EfficientNetV2

- State of the art model built using automated architecture search
  - Achieves 87% accuracy on ImageNet dataset with 1000 classes
  - Much more resource efficient than other approaches, such as older ConvNets or even Vision Transformers.
- EfficientNet is not that much more accurate, but it is *efficient*.
- We replace the classifier head with our own and keep the pre-trained feature layers.
- Fine-tune takes a few hours on an Nvidia GPU.



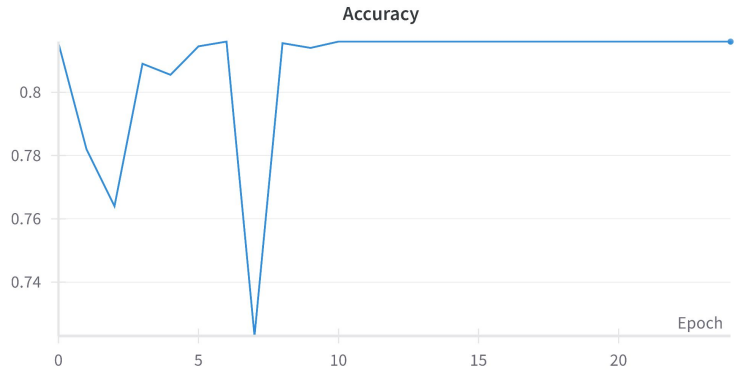
(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Transfer learning

- Fine-tuning a state of the art CNN
  - Pre-trained EfficientNetV2 M model
  - Keeping the feature blocks and replacing the output layers.
- We quickly get to 80% accuracy on a multiclass problem.
  - Accuracy plateau after 10 epochs
  - Limited by our dataset and class definitions
  - Even with 10 000 images, a few epochs are sufficient
- More data does not improve the fit
  - We tried a number of techniques (weight regularization, learning rate scheduling, etc.)



Comparison: Zoobot

- The original GalazyZoo authors also trained a model, Zoobot
  - Use of a hierarchical Dirichlet multinomial loss function
  - Given that the questions are a hierarchical decision tree, we could have done the same
  - A number of different CNN architectures
- In multi-class classification examples, the authors achieve an accuracy of 83% on a simple dataset of galaxies with rings
- The original dataset is about the *characterization* of galaxies
  - To some extent it is more of a regression problem than a classification problem
  - The goal of the original paper might be anomaly detection, by quickly sifting through catalog data to find interesting galaxies

Conclusion

- Training on data with high entropy (unconfident labels, image noise) is a daunting task
- Multiple approaches taken, including a classic and hierarchical CNN, and a pre-trained ImageNet model
- Difficult to improve classification accuracy beyond 80% (possibly a data limitation)
  - Fine-tuning an existing model can be done with few images/epochs, reaching similar accuracy in much shorter time

Future works

- With modifications to the labels
- Work on using the loss function, that does the hierarchical multi-class classification
  - Instead of “brute-forcing”


Thank you!

Appendix

Appendix A. GZD-5 data

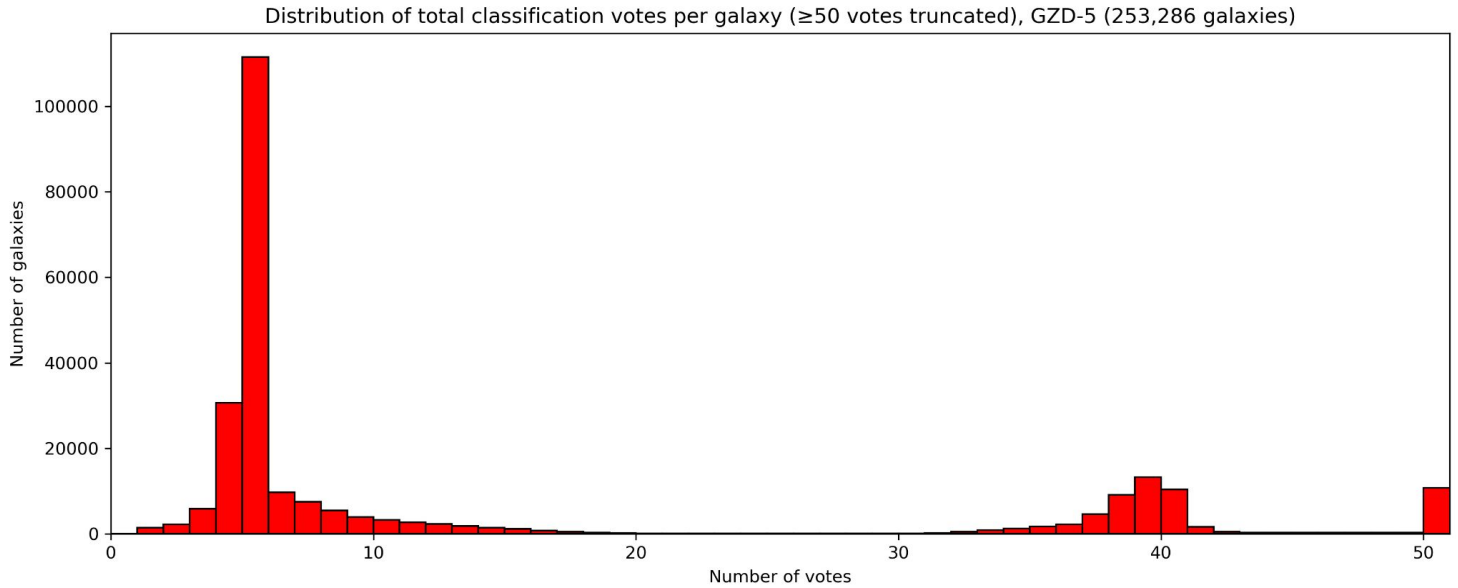
1. Sample of GZD-5 dataset

iauname	J112953.88-000427.4	J104325.29+190335.0	J104629.54+115415.1	J082950.68+125621.8	J122056.00-015022.0
smooth_or_featured_total_votes	84	37	5	8	5
smooth_or_featured_smooth	57	33	1	2	2
smooth_or_featured_smooth_fraction	0.678571	0.891892	0.2	0.25	0.4
smooth_or_featured_smooth_debiased	0.102564	0.857143	NaN	NaN	NaN
smooth_or_featured_featured_or_disk	23	2	4	6	3
smooth_or_featured_featured_or_disk_fraction	0.27381	0.054054	0.8	0.75	0.6
smooth_or_featured_featured_or_disk_debiased	0.916667	0.038462		NaN	NaN
smooth_or_featured_artifact	4	2	0	0	0
smooth_or_featured_artifact_fraction	0.047619	0.054054	0	0	0
smooth_or_featured_artifact_debiased	0.025742	0.022166	NaN	NaN	NaN
disk_edge_on_total_votes	23	2	4	6	3
disk_edge_on_yes	7	0	0	6	0
disk_edge_on_yes_fraction	0.304348	0	0	1	0
disk_edge_on_yes_debiased	0.04878	0	NaN	NaN	NaN
disk_edge_on_no	16	2	4	0	3
disk_edge_on_no_fraction	0.695652	1	1	0	1
disk_edge_on_no_debiased	0.805502	1	NaN	NaN	NaN
has_spiral_arms_total_votes	16	2	4	0	3
has_spiral_arms_yes	1	0	4	0	2
has_spiral_arms_yes_fraction	0.0625	0	1		0.666667
has_spiral_arms_yes_debiased	0.820513	0	NaN	NaN	NaN
has_spiral_arms_no	15	2	0	0	1
has_spiral_arms_no_fraction	0.9375	1	0	NaN	0.333333
has_spiral_arms_no_debiased	0.108171	1	NaN	NaN	NaN

Some metadata rows removed for clarity. **debiased** rows (accounting for source visibility at certain redshift) are not used, as their origin calculation was unclear.

Appendix A. GZD-5 data

2. Distribution of total classification votes per galaxy



Appendix B. Precision, recall & F1-score

- **Precision**
  - *Of all labels the model predicted as present, how many were actually correct?*

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall**
  - *Of all the labels that should have been predicted, how many did the model find?*

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1-score**
  - *Balances precision and recall → high only when both are high*
    - Macro: equal weight for all labels
    - Micro: weight by label frequency (more common = more influence)

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

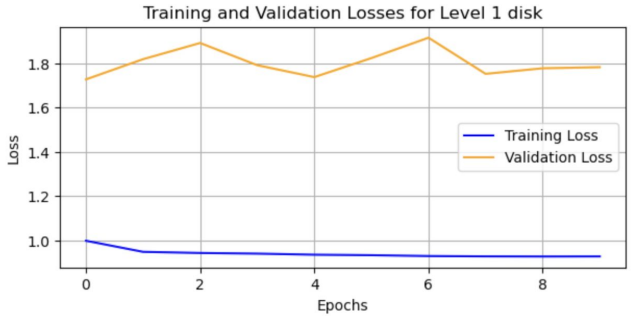
Appendix C. Number of Images for Each Level

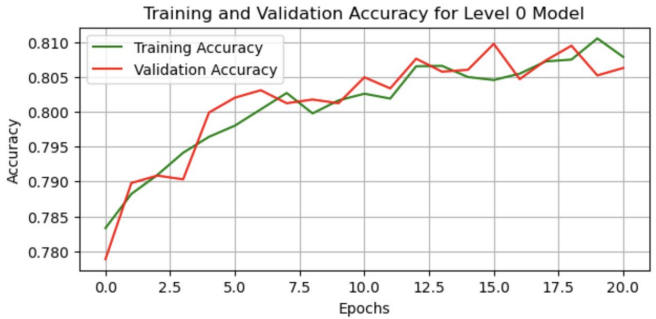
Hierarchical Multi-class training data for each level

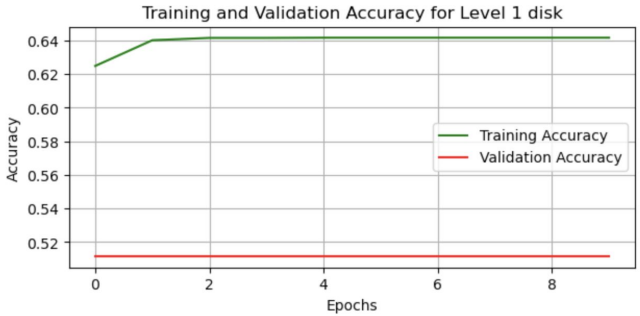
```
Total galaxies/images in dataset after removing validation set: 15030
Total galaxies/images in smooth branch after removing validation set: 2915
Total galaxies/images in disk branch after removing validation set: 6403
Total galaxies/images in spiral branch after removing validation set: 4275
Total galaxies/images in merging branch after removing validation set: 4275

Total number of galaxies/images in the validation set: 3758
```

- Branch sizes vary due to vote thresholds and Galaxy Zoo branching logic - disk branch is most populated, smooth is smallest. For all images where spiral questions were answered, merging questions were also answered - i.e., same number of images.

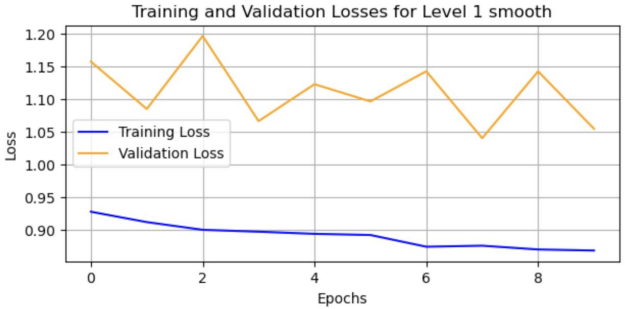
Appendix D1. Train & Validation Loss, Hierarchical Multi-class

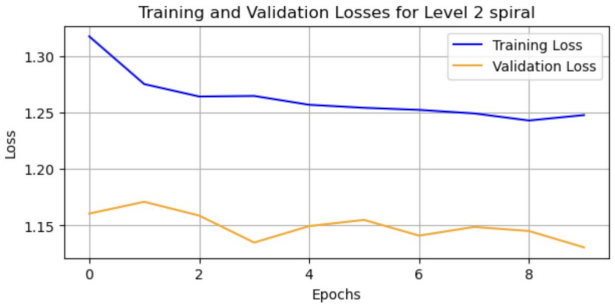


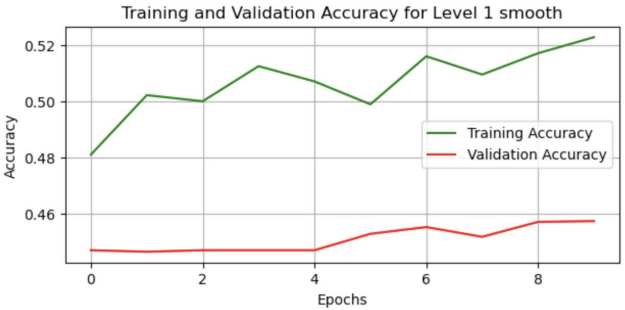


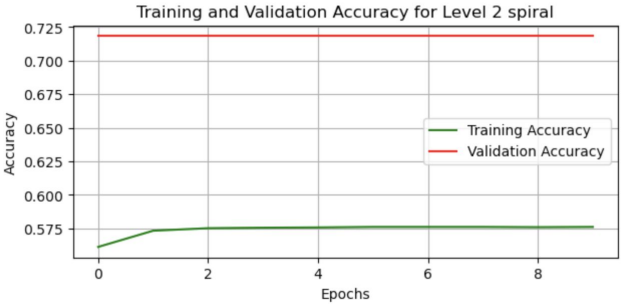


Appendix D2. Train & Validation Loss, Hierarchical Multi-class









Appendix D3. Train & Validation Loss, Hierarchical Multi-class