Morphologies for the next generation of surveys

Requirements, challenges and opportunities for morphological classification with machine learning in the era of LSST

Garreth Martin (University of Arizona)

Sexten CFA, 04 Feb 2020

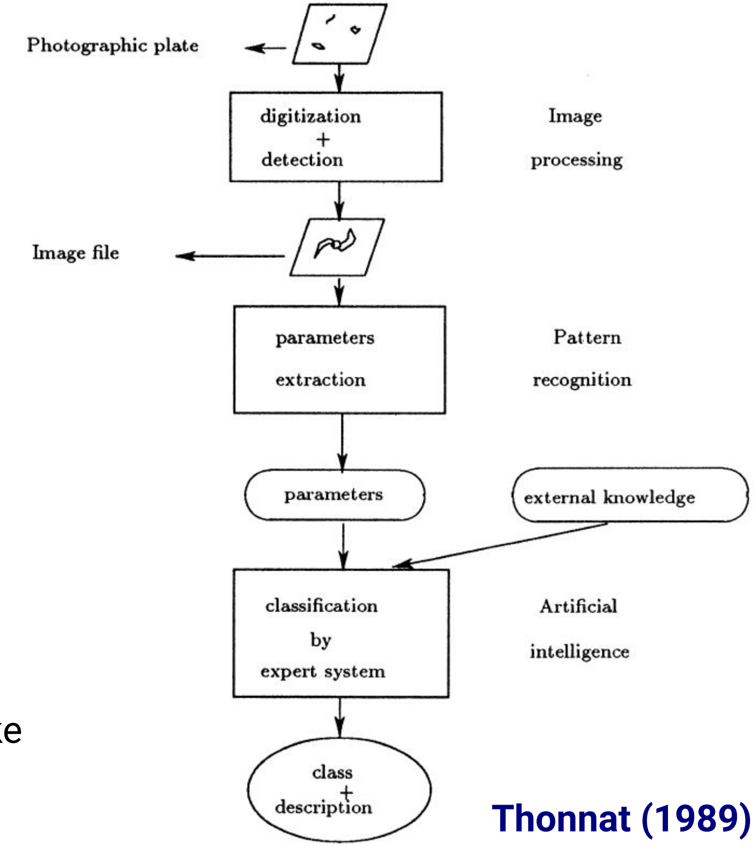
A brief history of ML techniques in astronomy

- Machine learning techniques have been developed for morphological classification since CCDs/plate digitization became widespread in astronomy beginning in the **1980s** e.g. **Kodaira+Watanabe (1984), Thonnat (1989)**.

- At this time, a lack of computer power or sophistication of technique meant that these solutions were **unable to process even the relatively modest data-volumes** seen at this time (< several GB)

- Perhaps the first truly **successful** application of ML to galaxy classification was by **Lahav et al. (1995)**, who were able to efficiently classify ~14000 objects with **similar accuracy to an expert human classifier**

- However, **ML techniques still did not become widespread**, perhaps because did not offer any particular advantage over **expert human classifications** or later **citizen science** efforts like **Galaxy Zoo (Lintott et al., 2008)** , which offer high quality classifications for large numbers of galaxies

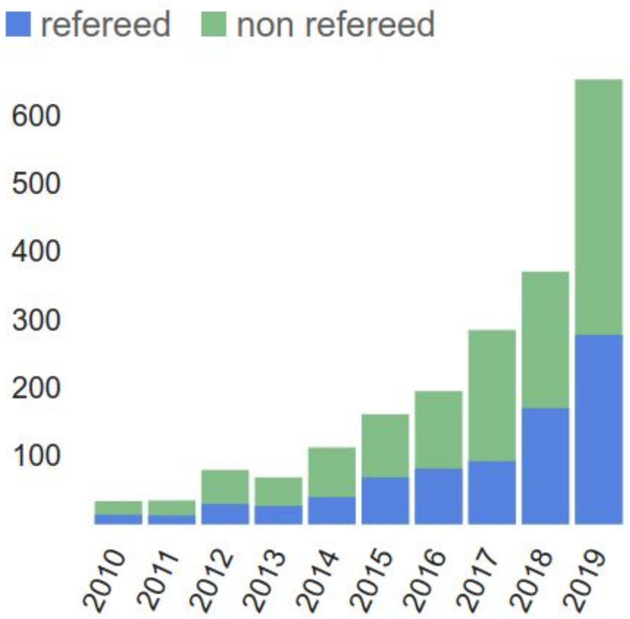


Thonnat (1989)

A brief history of ML techniques in astronomy

As they have become **increasingly necessary due to large data volumes**, a wide range of machine learning solutions have now been applied successfully to problems in astronomy:

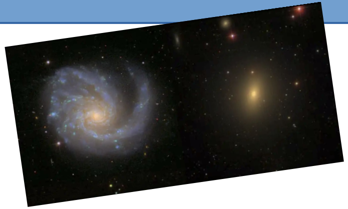
- **Huertas-Company et al. (2015)** convolutional neural networks
- **Ostrovski et al. (2017)** supervised Gaussian mixture models
- **Schawinski et al. (2017)** generative adversarial networks
- **Goulding et al. (2018)** random forest classifier
- **Siudek et al. (2018)** unsupervised Fisher expectation-maximisation
- **Roussi in prep** Siamese networks



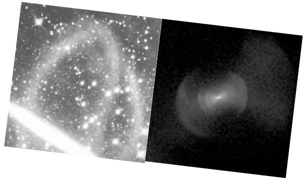
ADS astronomy abstracts referencing "machine learning" by year

The problem

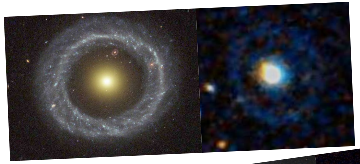
- Morphological classification -- what do our algorithms need to do?:
  - Separation of objects into **Hubble type**



  - Identification of objects of that share **specific features** (e.g. tidal tails, rings, shells and other LSB features)

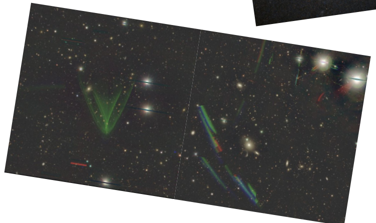


  - Identification of **rare objects, outliers** or objects **that don’t fit into established morphological types** and for which there are no large existing samples (e.g. ring galaxies, certain LSB galaxies)



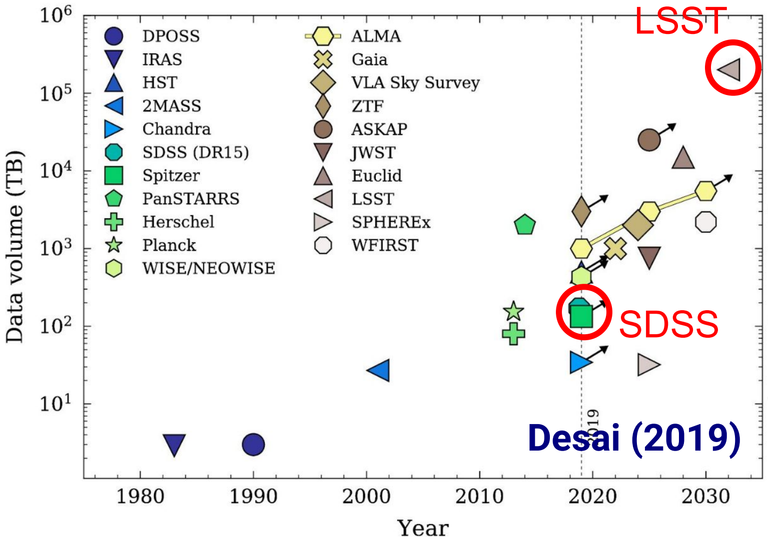
  - Separation of **arbitrary morphologies** and **recovery of blended objects** for which it would not be possible to construct training sets (e.g. low surface brightness objects overlapping other objects)

  - **Identification of ‘junk’** not removed by the pipeline (e.g. satellite trails, ghosts etc), **star / galaxy separation**



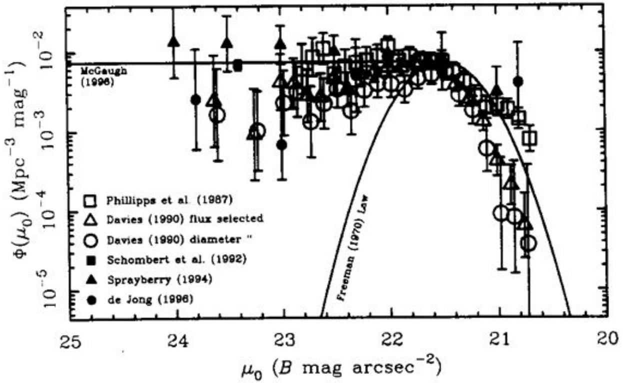
Challenges

- Data volumes **continue to grow** due to the increasing area, depth, resolution and cadence possible with modern survey instruments:
    - Rapidly changing datasets mean we may need to **classify data multiple times**
    - Deep imaging makes **looking for specific types of object laborious**, preventing us from assembling comprehensive samples
    - Large area and higher resolution means **more pixels need to be processed**

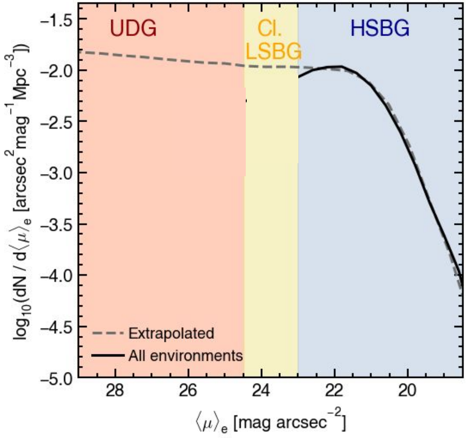


Challenges

This is also regime where we expect a **continuous tail of (resolved) low surface-brightness objects**



(Bothun et al., 1997)



(Martin et al., 2019)

As we probe lower and lower surface brightnesses, there is no indication from <u>observations</u> or <u>simulations</u> that the number of objects will begin to drop

Challenges / requirements

- **Customisable** / general purpose
- And also **efficient and scalable** to large datasets
    - i.e. makes morphological classification feasible and fast for the any given use-case for individual researchers
- Allows for outlier detection
- Ideally applicable to **arbitrary morphologies** without the need for pre-labelled training data
- **Not (too) reliant on human effort**, which can be a significant bottleneck

Solutions

- Human classification will become less and less viable as datasets grow
  - e.g. **billions of individual classifications** required will make it intractable for LSST

- **Machine learning techniques** will soon be the only realistic solution, but face challenges of their own:
  - Repeated construction of **unbiased training sets** for high cadence (rapidly changing) data will be difficult

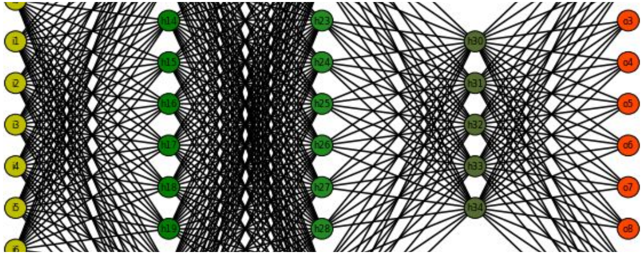
  - The large areas combined with deep imaging will allow the construction of samples of **rare/faint types of object,** but these object **will not have robust training sets** available

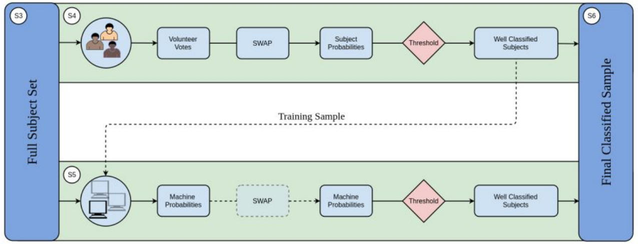
- One solution is to **combine citizen science with machine learning** in order to **continually improve training sets** e.g. **Beck et al. (2018)**, but very large data volumes will continue to be a challenge for any citizen science efforts

THE ZOONIVERSE WORKS

250,469,827

CLASSIFICATIONS SO FAR BY
1,944,912 REGISTERED VOLUNTEERS





Beck et al. (2018)

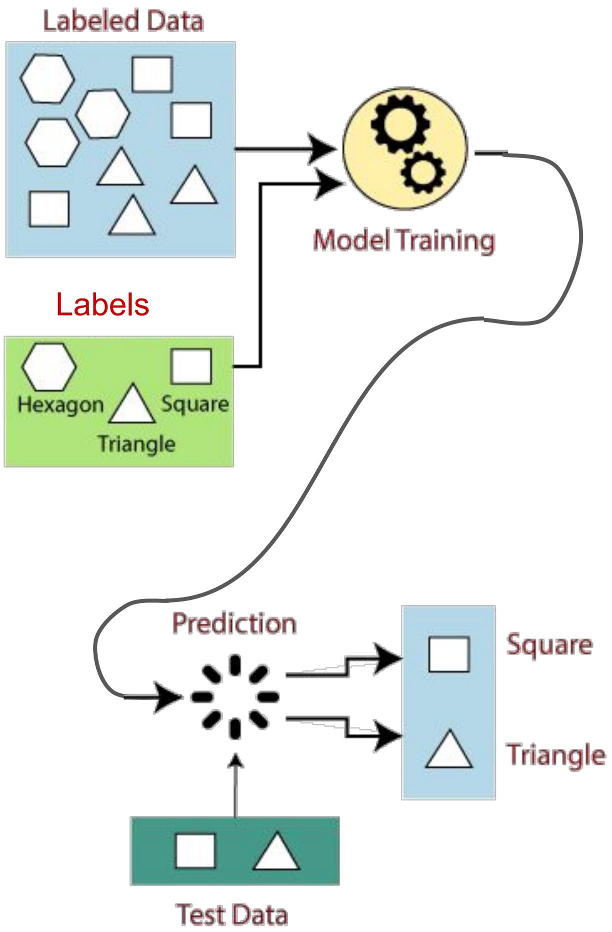
Solutions

- **Supervised machine learning** uses labelled training sets to find a mapping between input and output (e.g. an image of a galaxy and a morphological type).

- Such techniques are **accurate for focussed tasks** (e.g. yes/no classifications, small number of morphological types), but rely on **labelled training data**

- This won't work if we can't **assemble a large enough training set**, which is difficult where very **fine classification** is desired or we are interested in **rare types of object**
  - It is impossible to identify objects for which no training set has been provided

- Since the **assembly of training** sets is now **one of the most significant bottlenecks** for all but the most basic classification tasks, we would ideally want to design a method that requires **minimal human intervention** and can efficiently reduce populations of objects into **arbitrarily fine groups**

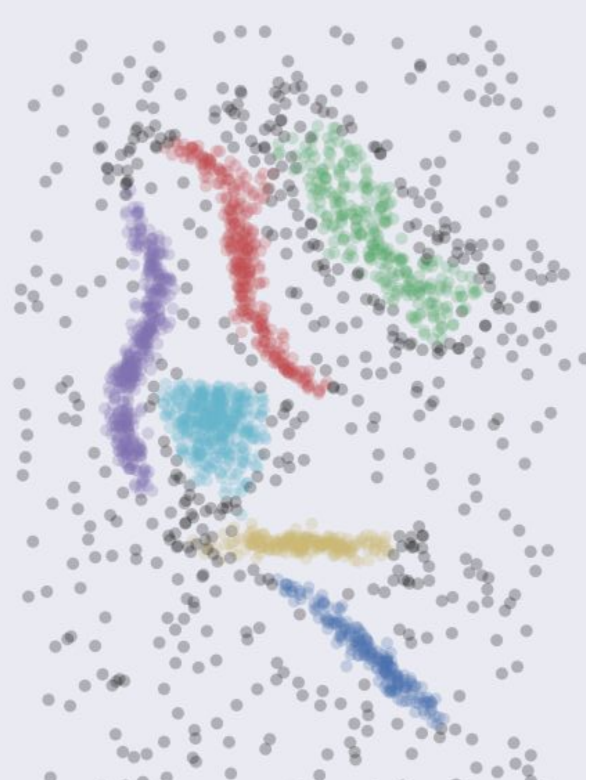


Solutions

- **Deep learning** **(e.g. Barchi et al., 2019)** and **unsupervised** techniques **(e.g. Hocking et al., 2018, Martin et al., 2020)** that can work directly on unlabelled data (**without labelled training sets**) can overcome some of the shortcomings of traditional supervised ML techniques

- Instead of optimising a network to recover provided labels, we try to **find groups of similar objects** within some parameter space

- Importantly, these techniques can be made to be more **general purpose** as they are **not limited to finding only objects in a training set**. They can produce instead **data representations that can be manipulated and used in different ways**



McInnes et al.

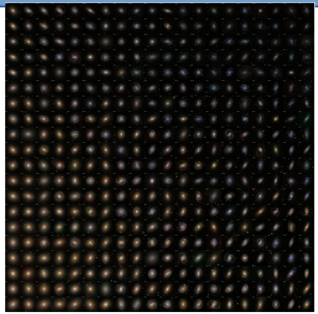
Examples

Labelled

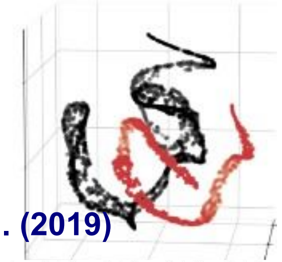
- **Polsterer, Gieseke & Kramer (2012)** -- support vector machines (self organising map method) without feature extraction with limited training sets
- **Dai & Tong (2018)** -- deep convolutional neural networks -- rely on large amounts of training data from galaxy zoo, limited to categories provided by galaxy zoo
- **Kahn et al. (2019)** -- deep learning applied to overlapping SDSS and DES data

Unlabelled

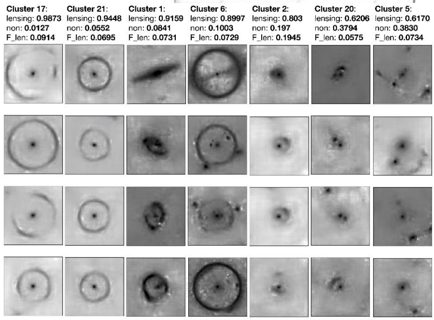
- **Hocking et al. (2018), Martin et al. (2020)** -- Self organising map based unsupervised method
- **Cheng et al. (2019)** -- unsupervised method using convolutional autoencoder for feature extraction rather than engineered features



Polsterer et al. (2012)



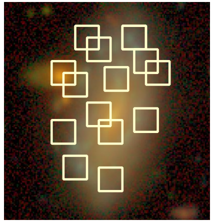
Kahn et al. (2019)



Cheng et al. (2019)

Hocking+ (2018): MNRAS, 473, 1108
Martin+ (2020): MNRAS, 491,1408
(arXiv:1909.10537)

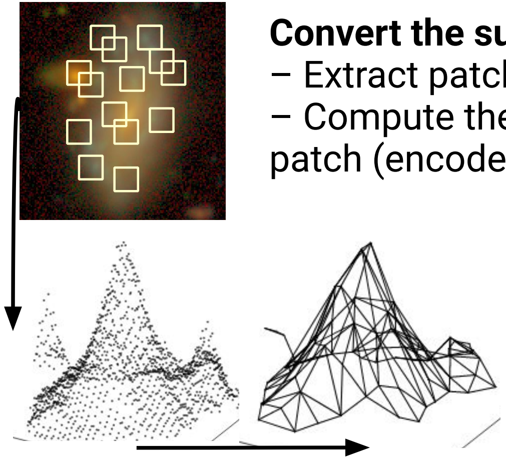
Examples



Convert the survey images into a data matrix
– Extract patches at each non-zero pixel in a multi-band image
– Compute the radial power spectrum to produce rotationally invariant representations of each patch (encodes **intensity**, **colour** and **‘texture’**)

Hocking+ (2018): MNRAS, 473, 1108
Martin+ (2020): MNRAS, 491,1408
(arXiv:1909.10537)

Examples

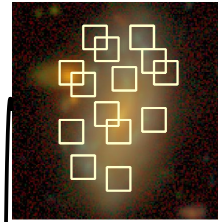


Convert the survey images into a data matrix
– Extract patches at each non-zero pixel in a multi-band image
– Compute the radial power spectrum to produce rotationally invariant representations of each patch (encodes **intensity**, **colour** and **'texture'**)

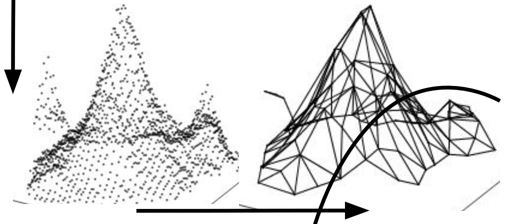
Use GNG and HC to produce a condensed version of the original data set
– Using the output patches, iteratively fit the data using growing neural gas to produce a topological map of sample vectors
– Each vector represents a group of similar patches
– By applying hierarchical clustering, we can further reduce the number of groups by reducing them to similar 'types' of patches

Hocking+ (2018): MNRAS, 473, 1108
Martin+ (2020): MNRAS, 491,1408
(arXiv:1909.10537)

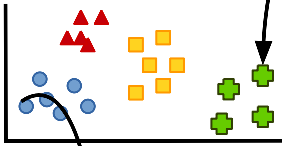
Examples



Convert the survey images into a data matrix
– Extract patches at each non-zero pixel in a multi-band image
– Compute the radial power spectrum to produce rotationally invariant representations of each patch (encodes **intensity**, **colour** and **‘texture’**)



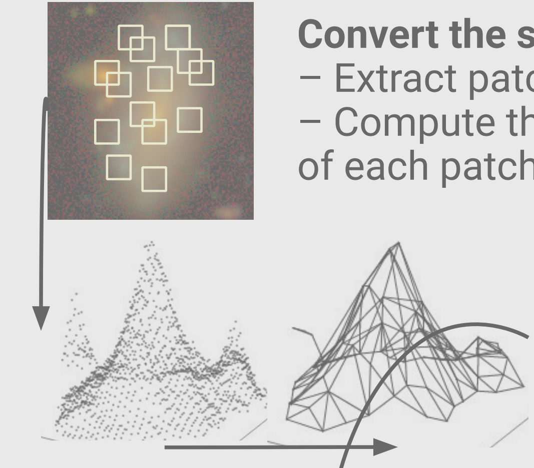
Use GNG and HC to produce a condensed version of the original data set
– Using the output patches, iteratively fit the data using growing neural gas to produce a topological map of sample vectors
– Each vector represents a group of similar patches
– By applying hierarchical clustering, we can further reduce the number of groups by reducing them to similar ‘types’ of patches



Create object sample vectors corresponding to patch ‘types’
– Identify objects using connected component labelling
– Create a sample vector for each object, represented by a histogram of the different ‘types’ of patches they are formed from
– Grouping similar sample vectors allows us to find visually similar objects

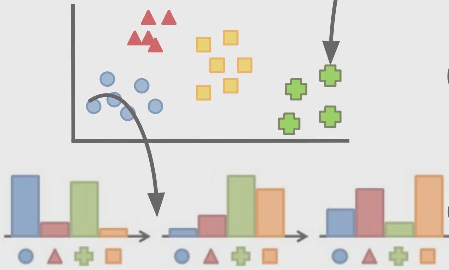
Hocking+ (2018): MNRAS, 473, 1108
Martin+ (2020): MNRAS, 491,1408
(arXiv:1909.10537)

Examples



Convert the survey images into a data matrix
– Extract patche
– Compute the r s
of each patch

r



Create c
– Identif
– Create
different 'types' of patches they are formed from
– Sample vectors are weighted by tf*idf (term frequency-inverse document

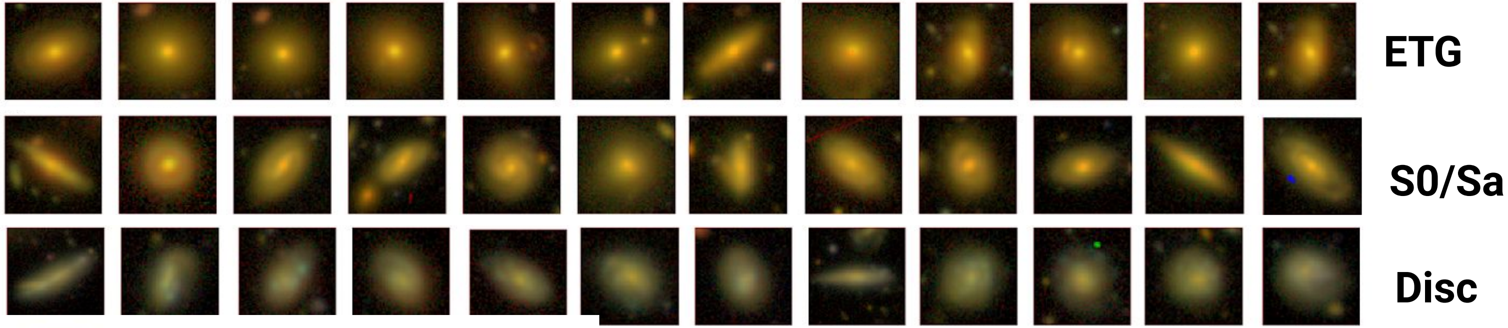
Condensed version:

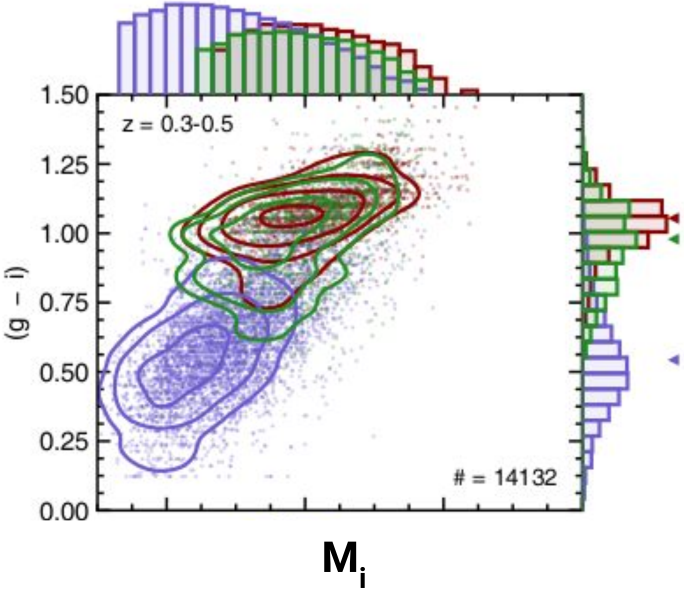
Use clustering techniques (**growing neural gas & hierarchical clustering**) to create a library of pixel 'types' based on colour, intensity and 'texture'

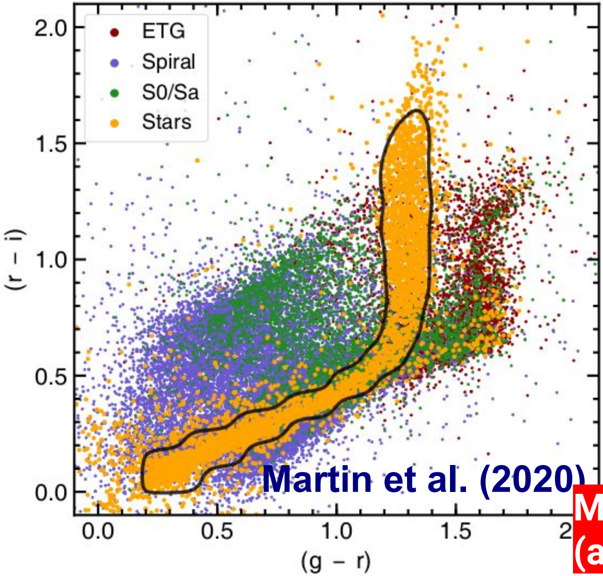
Produce histogram descriptions ('**feature vector**') of objects that describe the frequency of each pixel type in that object

https://github.com/garrethmartin/HSC_UML

Examples - Classification by Hubble type (HSC data)



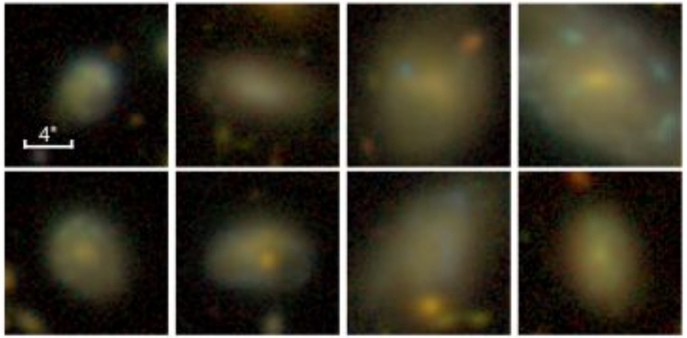




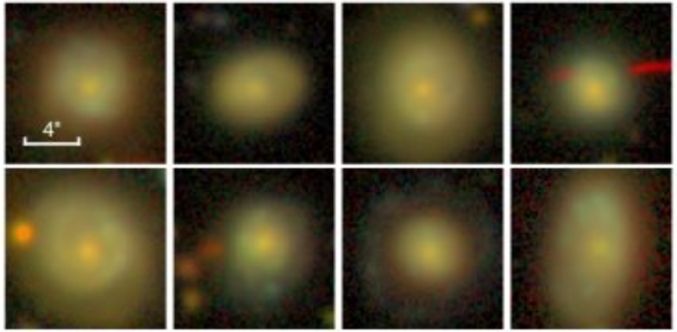
Classifications based on **visual inspection of a small subset** of each group produce expected relations

Martin+ (2020): MNRAS, 491,1408 (arXiv:1909.10537)

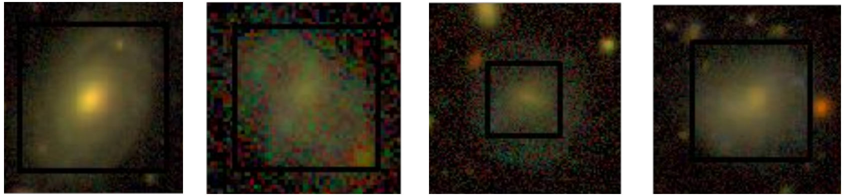
Examples - Arbitrary classification by clustering



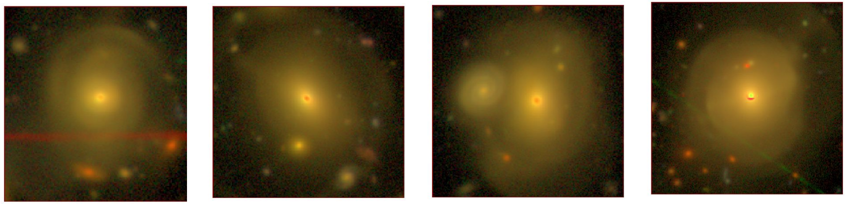
Clumpy discs



Rings / accretion events



LSB discs



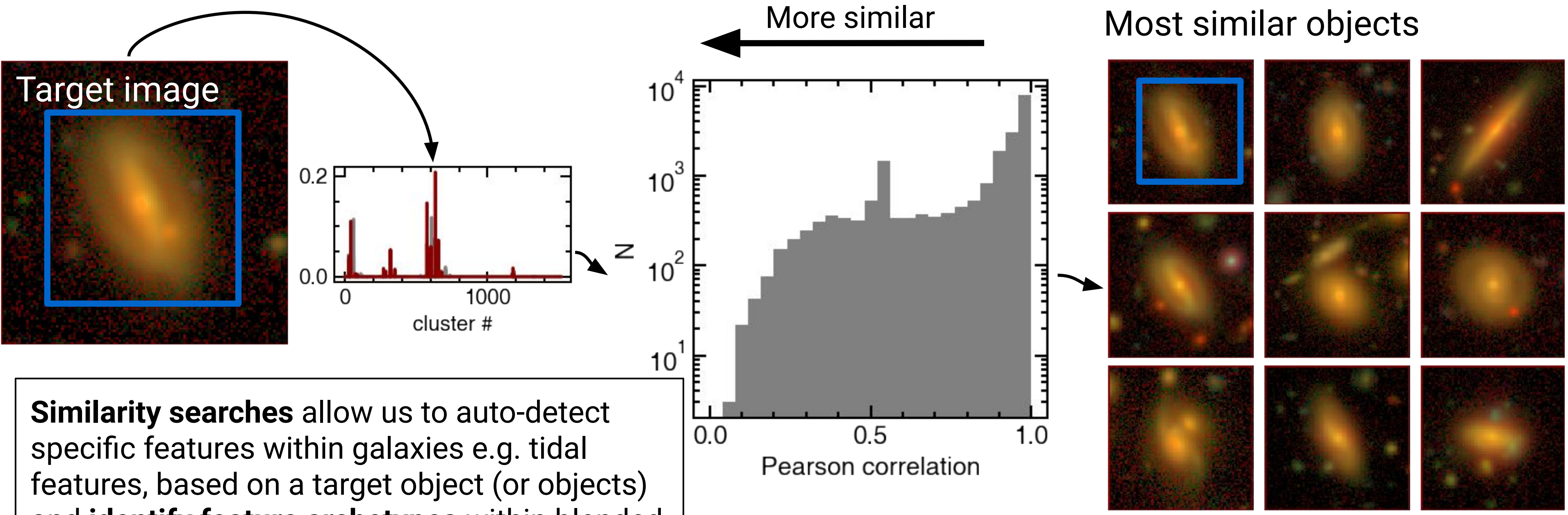
Shells

Some more examples of individual clusters featuring rare/specific types of object

(i.e. groups of object with similar feature vectors)

Martin+ (2020): MNRAS, 491,1408 (arXiv:1909.10537)

Examples - Classification by visual similarity



Similarity searches allow us to auto-detect specific features within galaxies e.g. tidal features, based on a target object (or objects) and **identify feature archetypes** within blended objects

Searching for the nearest feature vectors allows us to produce a library of similar objects

Martin+ in prep.

Summary

- Data volumes **continue to grow** as the area, depth, resolution and cadence of astronomical surveys continues to increase
  - Now becoming **intractable for citizen science** initiatives
- Deep surveys like LSST will allow us to **more finely classify galaxies** than we have been able to before
  - But it will not be possible to produce training sets for every category of object
- For supervised machine learning, the **creation of training sets will be a significant bottleneck**
- Unsupervised machine learning can offer a number of benefits over supervised methods in terms of their **scalability and ability to arbitrarily classify objects** without the need for labelled training data
- Unsupervised techniques do not just produce classifications -- they can also be used to create **usable descriptions** of each object which we can manipulate in various ways **(Martin+ (2020) MNRAS, 491,1408)**