

Machine Learning on classifying post-merger galaxies

Names: Kavya Mukundan, Mason Footh, Wenhao Li

Team #: m

Introduction: Why study post-merger galaxies?

Theoretically, it is expected that most galaxies host a supermassive black hole (SMBH) in their centers and that a SMBH can be triggered as an Active Galactic Nuclei (AGN) by accreting material through galaxy mergers (Gebhardt+2000; Hickox+ 2009; McConnell+ 2012; Kormendy & Ho 2013). Unfortunately, previous studies on whether mergers trigger AGN have found discrepant results. The likely reasons for these discrepant results are as follows: Firstly, luminous AGN have short lifetimes (\sim 10 million years) compared to the timescale of merger signatures (\sim 1 billion year, Martini 2004; Hopkins+2009). The sporadic nature of the luminous AGN will lead to an underestimate in the fraction of mergers which have experienced a luminous AGN phase. Secondly, AGN in mergers are in a low-luminosity phase for a longer duration (\sim 1 billion year, Hopkins+ 2008) compared to the luminous AGN phase (Conselice+ 2006). Therefore, studies that do not probe low-luminosity AGN cannot provide a complete census of the merger triggered AGN population. In addition, many previous works have relied on incomplete or impure samples of mergers at a range of merger stages.

Post-merger galaxies are the end products of major mergers, observed during coalescence, before or after the ULIRG/quasar phase and are expected to host low-luminosity AGN (Hickox+ 2009, Hopkins+ 2009). Post mergers are highly disturbed and characterized by tidal structures (see the figure) with lifetimes (\sim 1 billion year, Conselice+ 2006) comparable to the lifetimes of low-luminosity AGN (\sim 1 billion year, Hopkins+ 2008). Hence, post-merger galaxies are the ideal targets to look for low-luminosity AGN and to investigate the merger-AGN relation. As post mergers are tidal-disturbed systems, their morphologies are clearly different from those of galaxy pairs, galaxies with double nuclei and normal non-interacting galaxies. Galaxy classification is crucial and also difficult in astronomy. Visual classification is the most powerful method in galaxy classifications which however, needs a lot of professional knowledge and also takes time. We propose to use machine learning methods to classify post mergers based on their optical images.



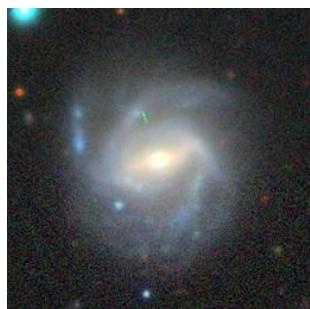
Machine learning methods have been widely used in galaxy classifications. Specifically, convolutional neural networks (CNNs) have achieved great success in galaxy morphology classifications based on the galaxy images (e.g., Huertas-Company+ 2015, Ackermann+ 2018, Dominguez+ 2018, Walmsley+ 2019). Bottrell+ 2019 have used CNNs to classify galaxies into specific merger stages. The accuracy of the CNN-based models in previous works are up to 80 ~ 90%. A recent work has used the Bayesian CNN to classify galaxy morphology and has obtained a high accuracy of 99% (Walmsley+ 2021).

Our Approach

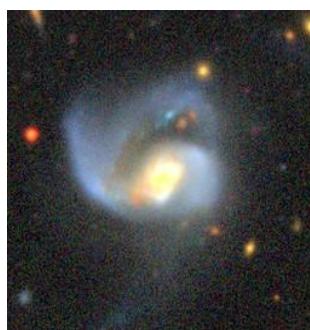
In our project, we propose to train a binary image classifier such that it can learn to accurately identify post merger galaxies from normal undisturbed galaxies. We have a sample of 30,000 normal galaxies and nearly 1,200 post-merger galaxies visually classified and labeled by Dr.Preethi Nair, astronomer, The University of Alabama. Since the sample is highly skewed/imbalance, we wish to adopt two different approaches for this problem. In experiment1, we use a convolutional neural network and in experiment 2, we implement anomaly detection using autoencoders. In the following sections, we describe our approaches in detail.

Data

The JPEG images of all galaxies from our sample were downloaded from the DESI survey (Dey et al. 2018) website (<https://datalab.noao.edu/ls/dataAccess.php>). The images have an input size of (256, 256) with 3 channels (“RGB”).



Normal Galaxies



Post Merger Galaxies

Experiment 1 : Supervised learning : Convolutional Neural Network

The data was first split into train, validation and test sets. The training dataset consists of 2500 normal galaxies and 500 post merger galaxies. The validation and test data sets consist of 200 and 100 images of normal and post merger galaxies respectively. The number of instances in the post merger class are very few compared to the majority class of normal galaxies in the training dataset. In order to resolve the imbalance, we adopt class weights and data augmentation of minority class.

Input

All input images are scaled (1/255) before being fed into the model. In order to increase the size of the training sample, random image augmentations (rotate, zoom, width shift, height shift) are performed on the training data.

Some examples of augmented training data:



Model architecture

After experimenting with multiple CNN architectures, we finalized the following model:

The model consists of three blocks of convolutional and max-pooling layers, where the convolutional layers consist of 128, 64 and 32 filters with a kernel size of (5,5). The convolutional layers employ a ReLU activation function along with the HeUniform kernel initializer. The max-pooling layers also have a pool size of (5,5).

The convolutional and pooling layers are followed by two dense layers of size 64 and 32 and two dropout layers with a dropout fraction of 0.5. Finally, the predictions are done by an output layer with sigmoid activation. The model is compiled using an Adam optimizer with initial learning rate 0.001 along with a Sigmoid Focal Loss Cross Entropy loss function.

Focal loss is a cross-entropy loss that weighs the contribution of each sample in the loss function depending on its classification error. The contribution of a correctly classified sample decreases and while that of incorrectly classified sample increases.

Focal Loss :

$$FL = - \sum_{i=1}^{C=2} (1 - s_i)^\gamma t_i \log(s_i)$$

, where the focussing parameter $\gamma \geq 0$, modulates the influence of the correctly classified sample in the loss.

In order to resolve the imbalance in the two classes, we also use class weights. The class weights are calculated using the class_weight function from sklearn.utils. The minority class is given a higher weight than the majority class. Thus the loss function learns to provide higher importance to the minority class than it would otherwise.

The class weights for our particular case are :

Post Merger galaxies : 3.0

Normal Galaxies : 0.6

The final model is trained for 20 epochs along with the implementation of learning rate scheduler and early stopping to avoid overfitting. The trained model is finally used to predict on the test data.

Model summary

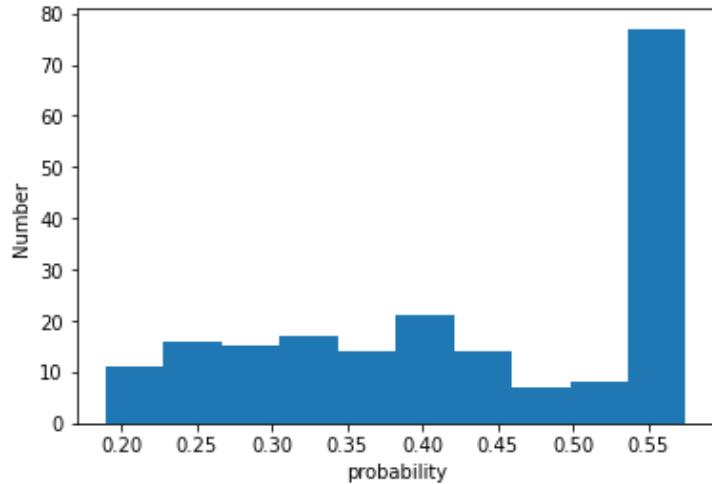
Layer (type)	Output Shape	Param #
<hr/>		
conv2d_15 (Conv2D)	(None, 256, 256, 128)	9728
max_pooling2d_15 (MaxPooling)	(None, 51, 51, 128)	0
conv2d_16 (Conv2D)	(None, 51, 51, 64)	204864
max_pooling2d_16 (MaxPooling)	(None, 10, 10, 64)	0
conv2d_17 (Conv2D)	(None, 10, 10, 32)	51232
max_pooling2d_17 (MaxPooling)	(None, 2, 2, 32)	0
flatten_5 (Flatten)	(None, 128)	0
dense_15 (Dense)	(None, 64)	8256
dropout_10 (Dropout)	(None, 64)	0
dense_16 (Dense)	(None, 32)	2080
dropout_11 (Dropout)	(None, 32)	0
dense_17 (Dense)	(None, 1)	33
<hr/>		
Total params: 276,193		
Trainable params: 276,193		
Non-trainable params: 0		

Results

The following plot shows the learning curve of the model for the training and validation data sets. The loss for both the datasets are small and close to each other. Thus the model is not overfitting the data.

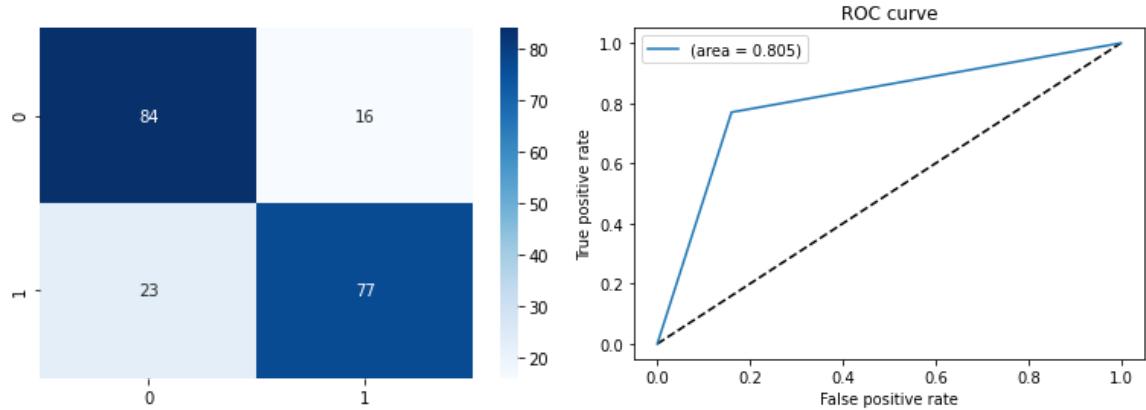


The model predicted the following probabilities when evaluated on the test data:



The probabilities are not well calibrated between 0 and 1. Though we are not aware of the exact reason for this, the usage of the focal loss function seems to be the reason for it. The predicted probabilities are between 0 and 1 when a simple binary cross entropy loss function is used. We visually choose 0.45 as the threshold to classify galaxies into post mergers and normal galaxies. Galaxies with a prediction less than 0.45 belong to class 0 (post merger galaxies) and galaxies with a prediction greater than 0.45 belong to class 1 (normal galaxies).

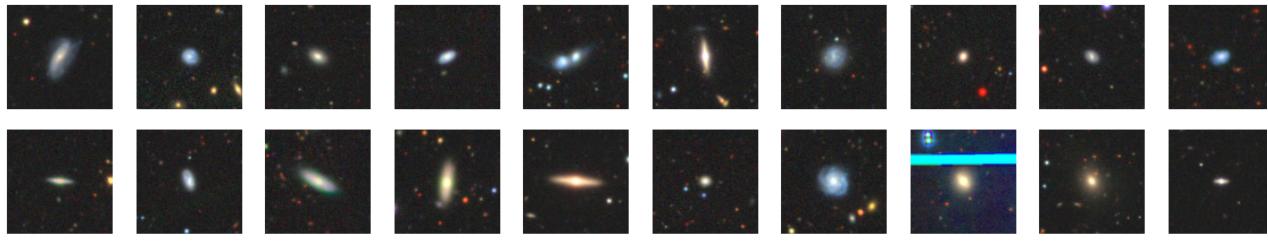
Using this classification threshold, we obtain the following confusion matrix and ROC_AUC plots:



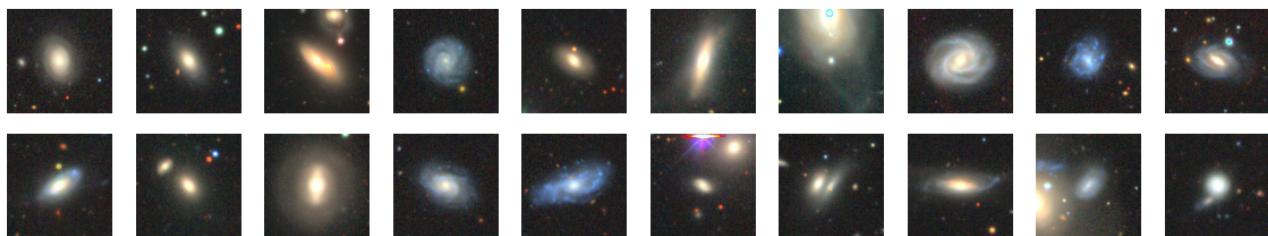
The `roc_auc` score is 0.805

In order to understand whether the model is indeed able to distinguish post merger galaxies from normal galaxies, we also look at the images of correctly and incorrectly classified galaxies.

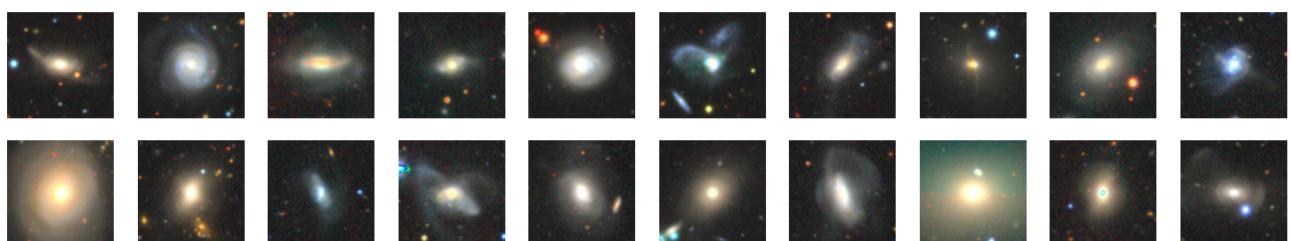
Normal galaxies correctly classified:



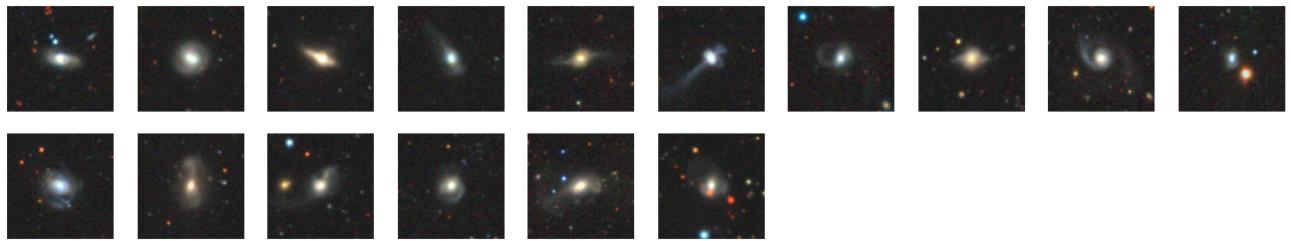
Normal galaxies incorrectly classified:



Post merger galaxies classified correctly:



Post merger galaxies classified incorrectly:



From the above images, it appears like elongated and edge-on galaxies are typically classified as normal galaxies while moderately face-on galaxies and galaxies with clearly visible features such as spiral arms are classified as post merger galaxies. Thus, there is a possibility that the model is misinterpreting the features of a galaxy with the distortions in a galaxy. A simple experiment to check this hypothesis is to feed the model nearly face-on normal galaxies and post merger galaxies and compare its performance on the new and old data.

Improvement on low inclination galaxies

We used low inclination normal galaxies and post mergers for training, validation and testing. The training set contains 2500 normal galaxies and 500 post mergers, while the validation set has 200 for each type and the testing set has 100 for each. The images were scaled and image augmentation was applied.

New model architecture

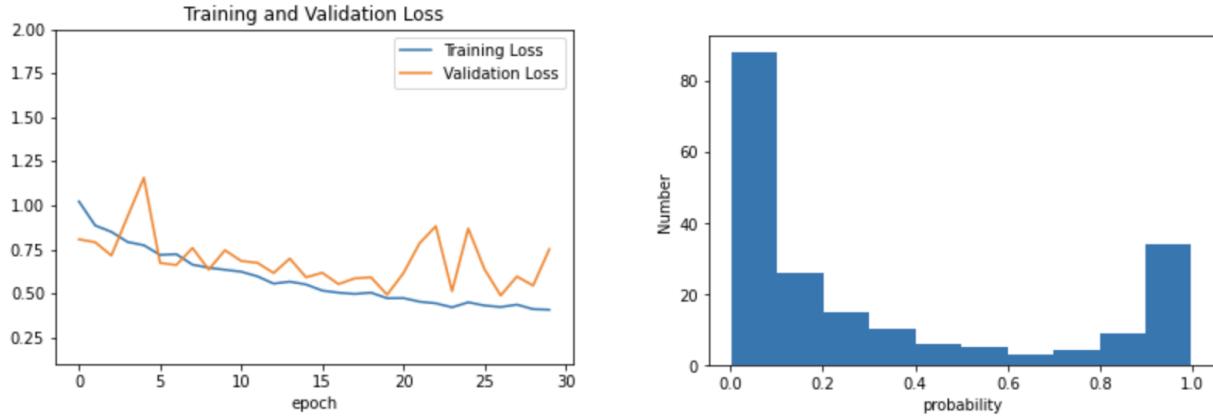
As the previous model did not perform well for these face-on images, we tuned the model by adding convolutional and dense layers. In total, we used 4 convolutional layers with 128, 128, 64 and 64 neurons, and 3 dense layers with 256, 128 and 64 neurons. Max pooling layers with kernel sizes 9x9 and 3x3 and batch normalization layers were also used with each convolutional layer to prevent overfitting. The activation function and kernel initializer functions we used are ReLU and He normalization. A dropout of 0.5 is also used with each dense layer. We used the Adam optimizer with an initial learning rate of 0.001 along with a binary cross entropy loss function. The new model architecture is shown below:

Model Summary

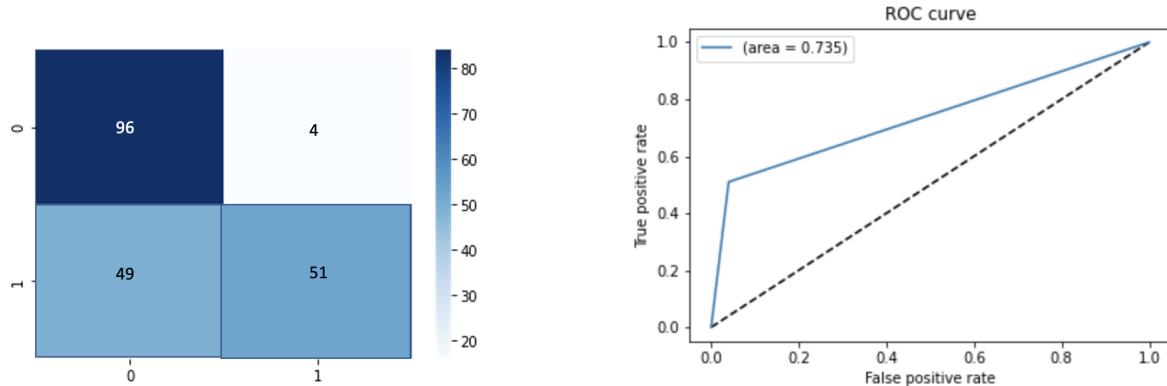
Layer (type)	Output Shape	Param #
=====		
conv2d_53 (Conv2D)	(None, 256, 256, 128)	9728
batch_normalization_81 (BatchNormalization)	(None, 256, 256, 128)	512
max_pooling2d_53 (MaxPooling)	(None, 28, 28, 128)	0
conv2d_54 (Conv2D)	(None, 28, 28, 128)	409728
batch_normalization_82 (BatchNormalization)	(None, 28, 28, 128)	512
max_pooling2d_54 (MaxPooling)	(None, 9, 9, 128)	0
conv2d_55 (Conv2D)	(None, 9, 9, 64)	204864
batch_normalization_83 (BatchNormalization)	(None, 9, 9, 64)	256
max_pooling2d_55 (MaxPooling)	(None, 3, 3, 64)	0
conv2d_56 (Conv2D)	(None, 3, 3, 64)	102464
batch_normalization_84 (BatchNormalization)	(None, 3, 3, 64)	256
max_pooling2d_56 (MaxPooling)	(None, 1, 1, 64)	0
flatten_9 (Flatten)	(None, 64)	0
dense_37 (Dense)	(None, 256)	16640
batch_normalization_85 (BatchNormalization)	(None, 256)	1024
dropout_28 (Dropout)	(None, 256)	0
dense_38 (Dense)	(None, 128)	32896
batch_normalization_86 (BatchNormalization)	(None, 128)	512
dropout_29 (Dropout)	(None, 128)	0
dense_39 (Dense)	(None, 64)	8256
batch_normalization_87 (BatchNormalization)	(None, 64)	256
dropout_30 (Dropout)	(None, 64)	0
dense_40 (Dense)	(None, 1)	65
=====		
Total params: 787,969		
Trainable params: 786,305		
Non-trainable params: 1,664		

Results

The training loss and validation loss for the best model is shown below. Though it is the best model, the validation loss is still slightly over the training loss. It performs well for the first 20 epochs and becomes slightly overfitting after that.

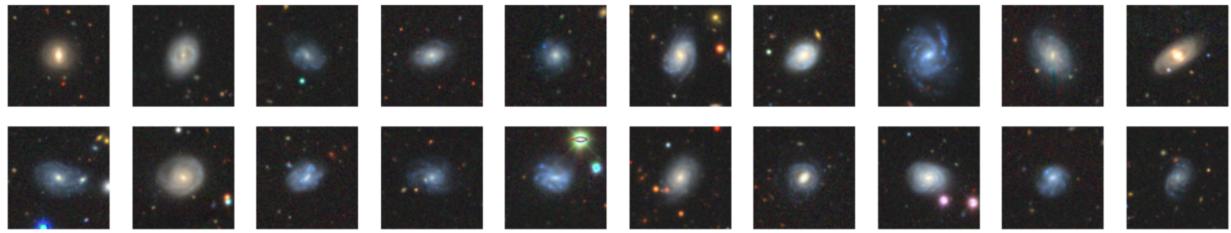


The probabilities are distributed well between 0 to 1, when tested on the test data. We chose 0.5 as a threshold to classify galaxies into the two classes. Using this classification threshold we obtain the following confusion matrix and ROC curve:

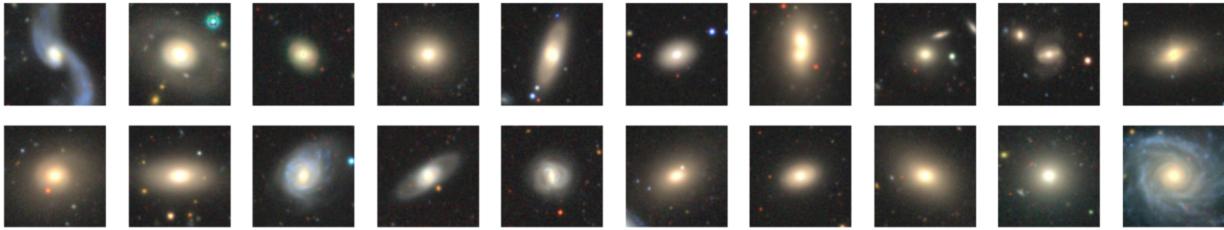


The ROC AUC score is 0.735. Though the model architecture is more complex than the previous one, it didn't perform better on the face-on galaxies. It turns out that classifying post mergers from face-on normal galaxies is more difficult than galaxies with all inclinations. However, the model can classify post mergers well, as only 4 out of 100 post mergers are classified incorrectly. A look at the galaxy images classified by the model clearly shows that the model is learning galaxy features, instead of simply relying on galaxy orientation as before. However, the performance of the model can be improved using larger training data and improved model architecture.

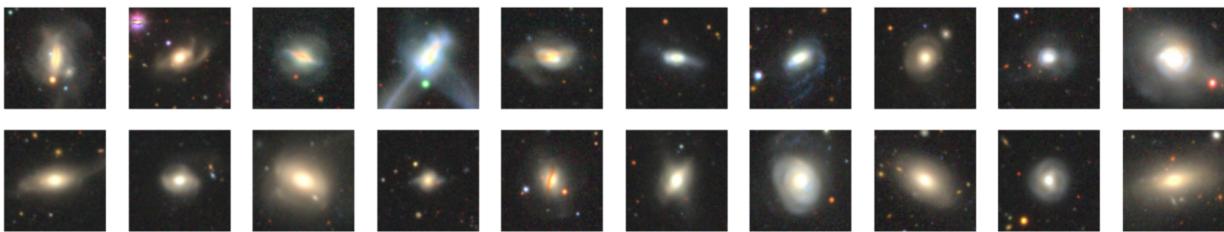
Normal galaxies classified correctly:



Normal galaxies classified incorrectly:



Post merger galaxies classified correctly:



Post merger galaxies classified incorrectly:



Experiment 2 : Unsupervised learning (Anomaly detection)

Introduction and motivation

There has been an abundance of work in the past few years using or proposing unsupervised methods such as autoencoders for anomaly detection in astronomy, from supernovae detection (Villar+ 2020), to the study of cosmic baryons in the form of diffuse plasma (Ichinohe & Yamada 2019). Some have found success using variational and/or convolutional autoencoders to both classify galaxy morphology and generate synthetic images (Spindler+ 2021), evaluate photometric redshifts (Abul Hayat+ 2020), or other similar applications (Ralph et al. 2019). However, to our knowledge, utilizing anomaly detection for classifying post-merger galaxies with a convolutional autoencoder has never been attempted.

The skyrocketing volume of data generated by modern astronomical surveys practically requires machine learning techniques for classification and analysis. By not requiring labelled training data, unsupervised learning algorithms allow for easier integration into the data classification and analysis pipeline of astronomers working with these large surveys. Furthermore, the convolutional autoencoder lends itself well to feature reduction and transfer learning for solving similar problems along the classification and analysis pipeline. Finally, it gives a framework which can be reset and applied to unrelated classification problems, or even completely agnostic anomaly detection searches.

Description of our autoencoder and the training process:

We use a convolutional autoencoder with two independent anomaly metrics. We trained the autoencoder on 5980 normal galaxies, with a validation set of 505 normal galaxies and an anomaly set of 1121 post merger galaxies. Our initial approach was to train it to reconstruct the image as well as possible and use the reconstruction error as a similarity metric to discern whether a galaxy is normal or a post-merger.

However, that approach failed spectacularly in a number of ways. First, the normal galaxies have a high amount of variance, so that the difference between the mean and outlier normal galaxies can be larger than the difference between the mean and true anomalies. In the context of this experiment, all of the post-merger galaxies were within 2 standard deviations from the mean normal galaxy reconstruction error. Even worse, the latent space was monstrously high dimensional, and some of the convolutional layers had so many filters that attempting to utilize the latent space often resulted in Colab running out of memory, even while using the GPU. We experimented with literally hundreds of potential model architectures, and a large number of potential loss functions, optimizers, and other hyperparameters before settling on our solution.

The first part of the solution was to crop the images by 64 pixels on either axis. Because the images are almost all centered on the target galaxy, this works to increase the clearly horrendous signal to noise ratio, in terms of target galactic morphology vs everything else in the image. The second part of the solution was to restructure the network to ensure the latent space was sufficiently manageable and that we could wrestle with large batch sizes unimpeded by the limitations of google's charity.

The encoder portion of our autoencoder consists of four convolutional layers, with SELU activation function and Lecun Normal kernel initialization to ensure a self normalizing network, followed by max pooling layers. The decoder mirrors the encoder but in reverse and with up sampling (using bilinear interpolation) instead of max pooling. For a loss function, we used the mean squared error, and we used stochastic gradient descent with momentum and nesterov methods for optimization.

```
encoder.summary()
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 192, 192, 48)	1344
max_pooling2d_4 (MaxPooling2D)	(None, 48, 48, 48)	0
conv2d_10 (Conv2D)	(None, 48, 48, 32)	13856
max_pooling2d_5 (MaxPooling2D)	(None, 12, 12, 32)	0
conv2d_11 (Conv2D)	(None, 12, 12, 12)	3468
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 12)	0
conv2d_12 (Conv2D)	(None, 6, 6, 6)	654
max_pooling2d_7 (MaxPooling2D)	(None, 3, 3, 6)	0
<hr/>		
Total params: 19,322		
Trainable params: 19,322		
Non-trainable params: 0		

The following is an example input and output of our autoencoder:

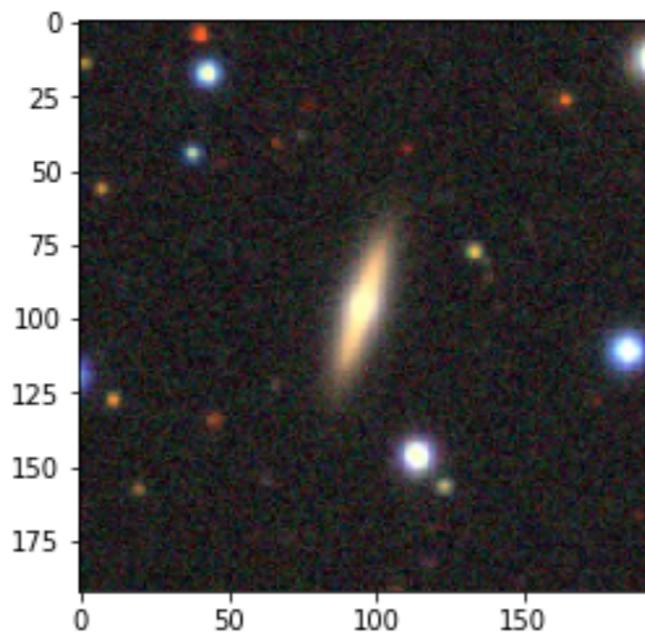


Image: Normal galaxy

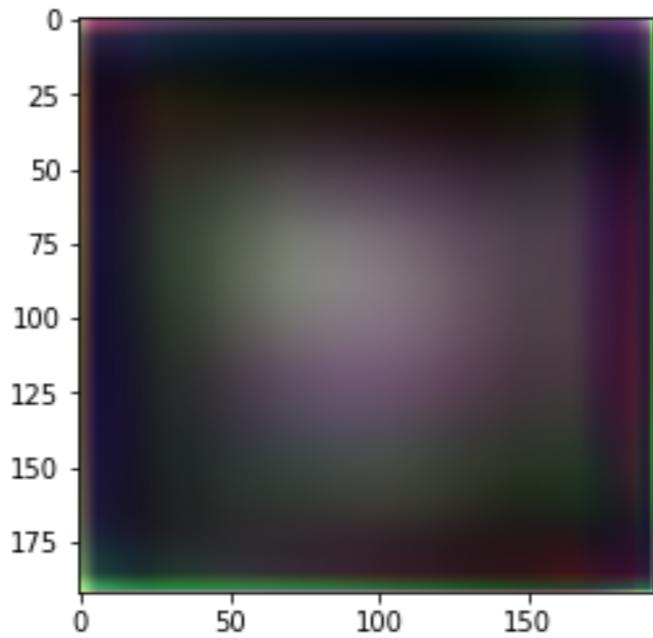


Image: Reconstructed normal galaxy.

Clearly, the output is not a perfect reconstruction of the input, as evidenced by the reconstructed example and the validation loss of 0.0121. Although it is worse, the anomaly reconstruction error is not significantly worse than the validation error, at 0.0137. Thus we turn to our latent space,

using the encoder as an efficient means of dimensionality reduction and kernel density estimation as our means of anomaly detection.

Description of Kernel Density Estimation:

Kernel Density Estimation is a nonparametric way of approximating the underlying distribution some sample of data comes from, and the KDE score is the log of the likelihood density for that data point were it from the same distribution as the data the KDE model is fit on. The 1 dimensional Kernel Density estimator at some point x is given by:

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^N K(d(x, x_i); h)$$

Where h is the bandwidth, $d(x, x_i)$ is a distance metric, and $K(\alpha; h)$ is a non-negative kernel function.

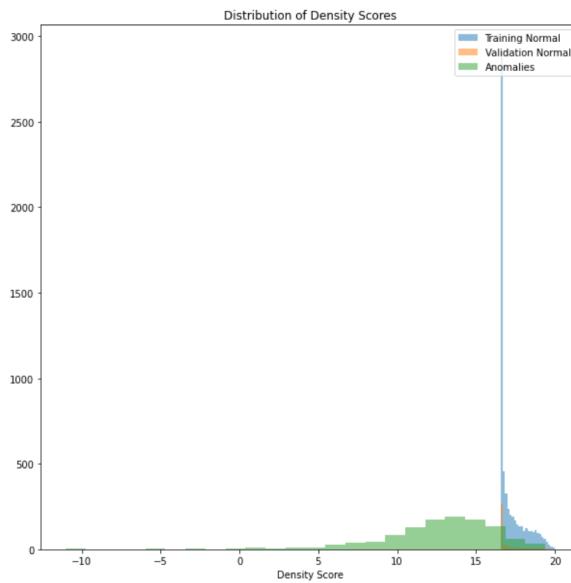
The bandwidth is the only tunable hyperparameter, though there are a number of different kernels and distance metrics of which to choose from. The bandwidth serves as a smoothing factor, with the usual trade off between bias and variance. The kernel and distance metric are chosen based on the desired distribution and what makes the most sense topologically for the data. Since we are already using mean squared error in training we stick with the Euclidean distance metric and since we have a lot of data we expect something akin to a Gaussian distribution thanks to the central limit theorem. Thus our kernel density estimator is:

$$\hat{f}_h(\vec{x}) = \frac{1}{Nh} \sum_{i=1}^N e^{-\frac{(\vec{x}-\vec{x}_i) \cdot (\vec{x}-\vec{x}_i)}{2h^2}}$$

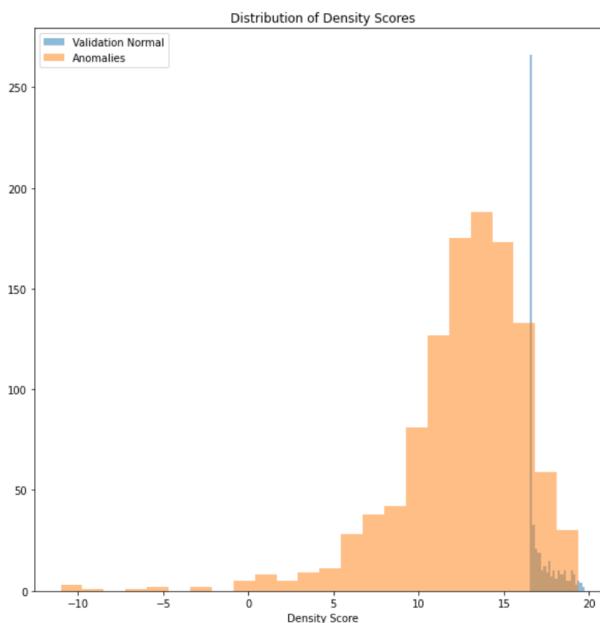
It is worth noting once more that the output scores are the log likelihood densities, and not the probabilities themselves, so we don't necessarily expect them to be less than 1 and care should be taken with their individual interpretation. The transformations which could be performed to extract the likelihood (e.g. integration and the exponential function) are either linear or continuous and increasing so that we can interpret relative distances as having the same meaning as if they were likelihoods, that is a lower KDE score implies lower likelihood, but again, the numeric value is not the likelihood.

Results and Discussion:

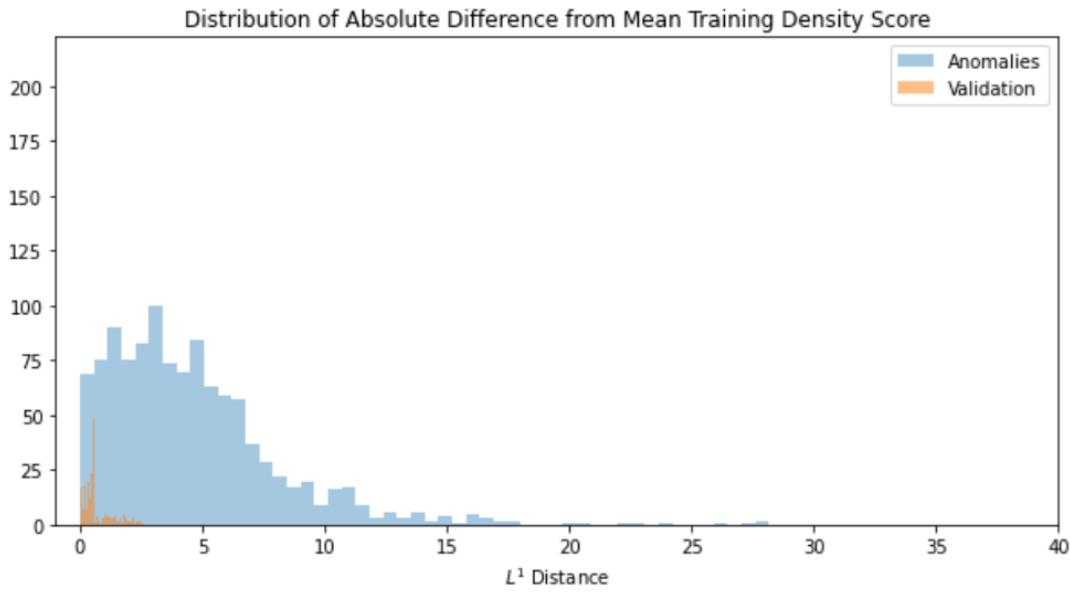
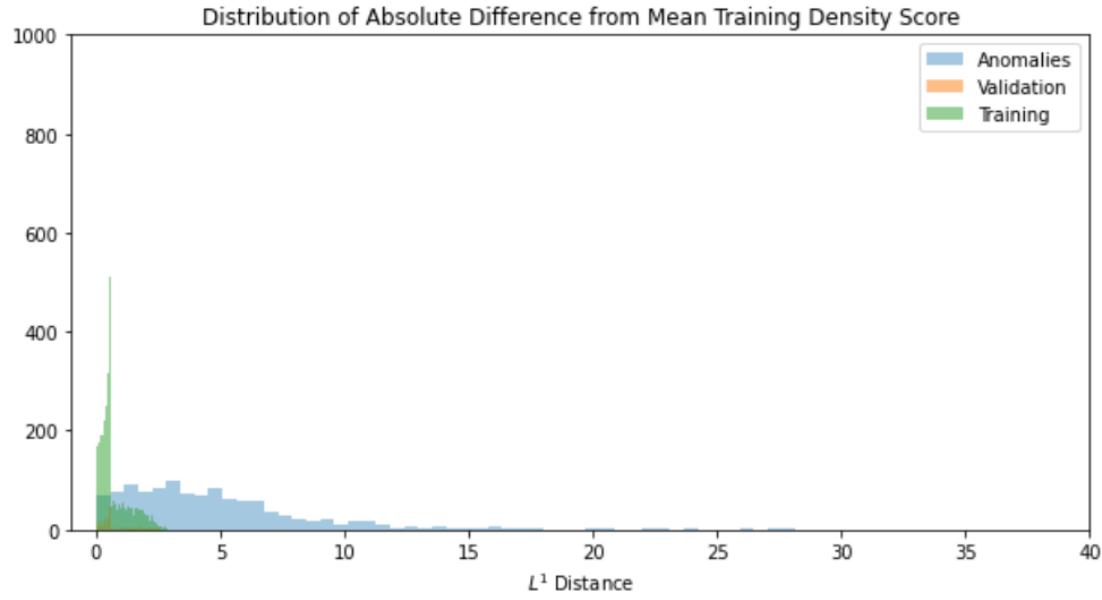
We use our 54-dimensional encoded training data to fit a Kernel Density Estimation model with bandwidth of 0.25, Gaussian kernel, and Euclidean distance metric. The relatively low bandwidth serves to ensure minimal variance while not overfitting. Evaluating the resultant log likelihood density scores of the training, anomaly and validation sets yield the following plots:



This is the overall distribution of KDE scores for the training, validation, and anomaly sets. Since the training set is so large, we also plotted just the validation and anomaly sets:



Note that, despite being fit only on the training set, the validation set appears to have almost the exact same distribution of log likelihood density. To further elucidate this relationship, we plotted the absolute distance of every KDE score from the mean training KDE score.



Again, note the low variance (by design) in the training and validation data, and similarity in distribution between the training and validation data. From these plots we can see that there is a clear break in the normal galaxy and post-merger galaxies at a KDE score of around 16. This suggests using a KDE score threshold as a similarity metric. Essentially, this is hypothesis testing that the galaxies are from the same distribution, but with extra (and necessary) steps.

The other metric used was mean squared reconstruction error. As already noted, the error was not significantly worse in reconstruction of the post-merger galaxies than it was for the normal galaxies. We chose 2 standard deviations away from the mean of the training reconstruction error as a threshold, which, assuming a Gaussian distribution in error, would exclude only the 5% of normal galaxies with the largest error. We then wrote a function to specifically evaluate a set of images using these metrics, and count the number of potential anomalies within the set from either. To evaluate either of these as metrics, we ran the validation data, anomaly data, and a final test set of 305 normal galaxies through our evaluation function. The purpose of the final test set is to assuage any anxiety over potentially overfitting the KDE estimator model.

```
evaluate_model('/content/drive/MyDrive/NormalVal/')
```

```
Found 505 images belonging to 2 classes.  
Reconstruction error anomalies = 20  
Density score anomalies = 4  
Maximum potential anomalies = 20  
Overall detection rate = 0.039603960396039604
```

```
evaluate_model('/content/drive/MyDrive/PostMergers/Train/Data/')
```

```
Found 1001 images belonging to 1 classes.  
Reconstruction error anomalies = 49  
Density score anomalies = 889  
Maximum potential anomalies = 889  
Overall detection rate = 0.8881118881118881
```

```
evaluate_model('/content/drive/MyDrive/PostMergers/Validation/')
```

```
Found 120 images belonging to 2 classes.  
Reconstruction error anomalies = 2  
Density score anomalies = 110  
Maximum potential anomalies = 110  
Overall detection rate = 0.9166666666666666
```

```
evaluate_model('/content/drive/MyDrive/TestNormal/')
```

```
Found 305 images belonging to 1 classes.  
Reconstruction error anomalies = 9  
Density score anomalies = 3  
Maximum potential anomalies = 9  
Overall detection rate = 0.029508196721311476
```

The mean squared error threshold ends up excluding a smaller percent of the post mergers (~4.5%) than it does the training normal galaxies. Though it is worth noting that it excludes more post mergers than the validation and final hold out test sets, so it's not completely feckless. Also worth noting is that it doesn't exclude any anomaly galaxies that aren't also excluded by the

KDE score threshold. The reconstruction error and density score anomalies are stored in arrays as a 1 or 0 for each image, which are then summed over to get their individual counts. The maximum potential anomalies are calculated by summing the max of either array, to avoid double counting. Thus the fact that the maximum potential number of anomalies is always equal to either the reconstruction error anomalies or KDE score anomalies says that they are completely overlapping sets.

The KDE score threshold manages to exclude approximately 89% of post-merger galaxies, by definition zero of the training normal galaxies, and less than 1% of the validation normal galaxies. As previously mentioned, to ensure we are not overfitting the Kernel Density estimator model and KDE score threshold as a metric, we evaluate it on a hold out set of 305 normal galaxies who have been completely separated from any training or evaluation up until this point. We find that the KDE score threshold excludes less than 1% of the hold out normal galaxies, as well.

Finally, we calculate the accuracy and F1 score of the KDE score threshold classification method. We use the anomalies as positive, so that an anomaly correctly classified as an anomaly is a true positive, and the validation and test sets of normal galaxies as negatives so that a normal galaxy above the KDE score threshold is a true negative. From the above data, we calculated these and the subsequent scores by hand. The resultant scores are an accuracy score of ~0.9332 and an F1 score of ~0.9394.

Summary

Post mergers are the end products of galaxy mergers and are ideal targets to look for low-luminosity AGN and to investigate the merger-AGN relation. They are highly disturbed systems that are characterized by tidal tails and shells. Visual inspection is by far the most efficient technique to classify galaxies. However, it is a time intensive process that requires a lot of professional knowledge. In recent years, machine learning has emerged as a potential alternative solution to visual classification. In our project, we applied two different machine learning techniques to identify post merger galaxies.

In experiment 1, we trained a convolutional neural network using 2500 normal galaxy images and 500 post merger images. In order to account for class imbalance, we experimented with class weights, focal loss and data augmentation. The model performed fairly well producing a ROC AUC score of 0.805. However, we realized that the model had unfortunately learnt to classify galaxies based on galaxy orientation instead of the actual galaxy features. We then trained a slightly modified model with normal and post merger galaxies with similar orientations. The second model produced a ROC AUC score of only 0.735, however, it was able to identify most of the post merger galaxies in the sample. The experiment highlights the complex nature of classifying galaxy images, particularly that of post mergers. We believe that performance of both models could be improved if trained with a larger volume of images.

In experiment 2, we used a convolutional autoencoder for anomaly detection with two independent anomaly metrics. We trained the autoencoder on 5980 normal galaxies, with a validation set of 505 normal galaxies and an anomaly set of 1121 post merger galaxies. In training the autoencoder, we focused on creating the best version of our images while retaining a network structure which allowed for a small bottleneck, to ensure that our latent space was as low dimensional as possible while still retaining the requisite information to perform analysis upon. We then fit a kernel density estimator model to the encoded training data, and used the minimum training data log likelihood density score as a threshold below which we classify a galaxy as a post-merger remnant. We evaluate this metric on the validation data, the anomaly data, and a final test set of 305 normal galaxies otherwise excluded from training until this point. This results in an accuracy of about 93% and an F1 score of approximately 0.9394. We believe that, given the time for preprocessing with proper astronomical image software (e.g. SExtractor, the astropy and photutils libraries) to clean the data, a graph neural network could handle this problem significantly more efficiently and elegantly. However, these results are satisfactory for a first pass on a difficult problem.

Reference

- Abul Hayat+ 2020, arXiv:2012.13083;
- Ackermann+ 2018, MNRAS, 479, 415;
- Bottrell+ 2019, MNRAS, 490, 5390;
- Conselice+ 2006, ApJ, 638, 686;
- Dominguez+ 2018, MNRAS, 476, 3661;
- Gebhardt+ 2000, ApJ, 539L, 13;
- Hickox+ 2009, ApJ, 696, 891;
- Hickox+ 2009, ApJ, 696, 891;
- Hopkins+ 2008, ApJS, 175, 356;
- Hopkins+ 2009, ApJ, 698, 1550;
- Huertas-Company+ 2015, ApJS, 221, 8;
- Ichinohe & Yamada 2019, MNRAS, 487, 2874;
- Kormendy & Ho, 2013, ARA&A, 51, 511;

Martini 2004, cbhg.symp, 169;
McConnell+ 2012, ApJ, 756, 179;
Ralph+ 2019, PASP, 131, 108011;
Spindler+ 2021, MNRAS, 502, 985;
Villar+ 2021, arXiv:2010.11194;
Walmsley+ 2019, MNRAS, 483, 2968;
Walmsley+ 2021, arXiv210208414W;