Kaggle Galaxy Zoo Image Data Clustering using Deep Embedding Clustering

Anonymous Authors[1]

Abstract

Galaxies are one of the most fundamental entity of the universe. They come in all shapes, sizes and colors and in order to understand how the different shapes (or morphologies) of these galaxies relate to the physics that create them it is important to group similar galaxies based on their structure. Considering the number of galaxy images collected through numerous telescopes this project tries to evaluate the performance of a proposed deep neural network architecture using Convolutional Neural Networks as feature extractor for clustering based of the algorithm: **"Deep Embedding Clustering"** (Xie et al., 2016) over a simple DEC based clustering network architecture on the Kaggle Galaxy Zoo data. The performance of the algorithm is evaluated based on the separation of *"Elliptical"* and *"Spiral"* galaxy images into distinct clusters.

1. Introduction

Understanding how and why we are here is one of the fundamental questions for the human race. In the quest of understand our origin we as human race also realized how important it is to understand the origin of our universe and all of its entities. According to many scientists galaxies, the most fundamental entity of the universe, might just hold the answer about the origin and evolution of the universe. Galaxies are of very different forms, size and shapes which has led scienties to believe that the inherent structure of galaxies might hold the clue to understanding the laws of physics that govern our universe. This quest for answers is what led to space exploration and development of thousands of telescopes to capture snapshots of galaxies in an effort to understand the structure of the galaxies and how it is critical to the origin and evolution of the universe itself. However in each passing moment, telescopes are capturing more and more images of galaxies far far away. Few decades ago the scientific community depended on massive citizen science crowdsourcing projects ((INSERT CITATION)) to help identofy the type of galaxies in each of the images captured. However because of the recent explosion in the number of images around million or even billions of images, it is virtually impossible for the community to depend on the citizen science projects to label each image individually.

To combat this problem the community pivoted to well establised Supervised Deep Learning techniques using Convolutional Neural Networks (CNN) for classifying each galaxy image and help in labelling the huge dataset. However, supervised learning deep learning techniqes are very much like a black box that takes in an input and returns an output, which in this case is just an image and a class label respectivesl. Using these models thus does not help understand what features define these galaxies and how do the inherent structure help in identifying the type of galaxy the image belongs to. Also the perfomance of some of these models decrease when tested on new images being captured. Because of this problem the community has to contnuously train these models which require volunteer labels for new images and thus leads to the same problem.

To address this problem, **Zooniverse** ((INSERT CITATION)) another citizen science crowdsourcing project has developed techniques which rely on clusters of images to efficiently collect labels for many images in much less time. Also using a clustering based approach helps in understanding what features of the galaxy are most influential and how do the structure relate to the galaxy type. Apart from that using clustering technique helps in extracting information about the most representative structure of the galaxies along with information about the similarity among galaxies of the same type. This experiment is part of their project to develop a more sophisticated technique which is able to cluster similar type of galaxy images together. The new galaxy clustering algorithm is based off a proposed architecture that uses a pretrained Convolutional Neural Network as a feature extractor along with the **"Deep Embedding Clustering (DEC)"** (Xie et al., 2016) architecture that helps to cluster the galaxy images. The motivation for this architec-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. **AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ture comes from the recent advances in the development of CNN based image extractors and their proven performance in image classification tasks (Russakovsky et al., 2015), along with the results of the DEC architecture in succesfully clustering the MNIST (LeCun et al., 1998) and REUTERS (Lewis et al., 2004) dataset. In this experiment we evaluate the performance of the proposed architecture over the performance of a standalone DEC architecture over the publicly available Kaggle Galaxy-Zoo dataset. For this experiment we are just evaluating the performance of the architecture on 2 classes of the galaxies which are: "Elliptical" and "Spiral". The size of the images in dataset is 424 by 424. An example of each of these galaxy images is provided in: Figure: 1.

(a) Spiral Galaxy (b) Elliptical Galaxy

Figure 1. Example Images from the Kaggle Galaxy Zoo dataset

Considering this experiment works with using Deep Learning architectures, Kaggle Galaxy Zoo dataset is one of the best datasets available because of the vast size of galaxies. The number of images for each type of galaxy in the Kaggle Galaxy Zoo dataset is provided in Table: 1. Also one of the best advantages of this dataset is that the number of images are not highly skewed based on the class labels and are pretty balanced out.

Table 1. Number of Images in the Kaggle Galaxy Zoo Dataset

GALAXY CLASS	NUMBER OF IMAGES
SPIRAL	34105
ELLIPTICAL	25868

The evaluation metric for this experiment is the accuracy of the cluster assignments of each galaxy image where the cluster assignments are calculated based upon the dominant class label in the various clusters formed.

2. Related Work

Classification of the galaxy images is one of the most common research topic that is widely studied for solving the problem of understanding the underlying structures in galaxies and trying to solve the problem of label collection for new galaxy images collected by the telescopes every minute. The task of galaxy image classification has been studies for quite some time and traditionally has relied on sophisticated machine learning techniques which depend on extracting useful features from the images and their metadata based on the physical feature information for the galaxy collected by the telescopes.

2.1. Feature Selection and Extraction

One of the earliest research for galaxy classification by Storrie-Lombardi 'et al.' (Storrie-Lombardi et al., 1992) found out 13 most important features based on the datasets and used these features as the input features to different architectures of a simple Artificial Neural Network (ANN). Another team used similarly extracted features from the galaxy dataset but implemented a Decision Tree algorithm (Owens et al., 1996) for classifying the galaxies. Considering the high dimensionality of pixel level image data and the performance of machine learning algorithms through extracted/learned/selected features a lot of research was done in the field of learning better and lower dimensional features from the image data that are better able to classify the galaxy data.

Past work has been done which uses principal component analysis (PCA) to project the dataset to a lower dimensionsional space and use the features from the lower dimensional space as the inputs to ANN and locally weighted regression (De La Calleja & Fuentes, 2004). Another research study (Naim et al., 1995) employs a similar technique where the extract some latent features in an automated matter and use them in conjunction with PCA learned features as the input features to an ANN for the classification purpose. Other work has also been done in using ensemble methods with ANNs, Decision trees and Naive Bayes Classifiers over the PCA learned feature space (Bazell & Aha, 2001). Apart from using PCA learned features several sophisticated features like Gini Coefficients (Abraham et al., 2003), optical spectrum of the light rays collected through the telescopes (Madgwick, 2003) and fourier transforms of the image features (Odewahn et al., 2002) were developed as input feature vectors to ANN for galaxy classification. Apart from focusing on feature engineering from the galaxy images research was also done on using the image pixels as direct inputs to deeper and larger ANN architecture (Goderya & Lolling, 2002). The research explored how well do ANNs learn distinguishing features from the image pixel values.

It is important to note that almost all of these methods focused on feature learning and selection for finding the input vector for machine learning algorithms or ANNs. This was the primary concern for all of these research groups because of the lack of better algorithms as image processing deep learning architectures, i.e. CNN hadnt been developed yet. Also it was not easy to train very large and deep ANNs

because of the lack of especially designed hardware.

2.2. Convolutional Neural Networks

Convolutional Neural Networks (CNN) were designed specifically for the purpose of processing visual data like images. In today's time, CNN based architectures have been widely used in fields like Image Classification (Russakovsky et al., 2015) and Object Detection. The motivating idea behind CNNs is that visual data has an inherent local structure such that values which are closer to each other tend to influence each other (Krizhevsky et al., 2012). To understand this, we can take the example of an image of a scenery. We know pixels representing the grass in the image tend to be similar to each other (mostly green). CNNs are very powerful tool because they automatically learn different latent visual features also termed as "feature maps" from the input image. This property makes CNN's very different from other traditional Machine Learning Algorithms, where the features have to be already specified. Considering that CNNs learn these high dimensional feature maps from the given input research has been conducted on using these deep CNN networks for directly classifying the galaxy image data (Khan et al., 2018). In their work, a pretrained CNN network: **"Xception Network"** (Chollet, 2017) was trained using greedy layerwise training for galaxy classification over the Sloan Digital Sky Survey (SDSS) and Dark Energy Survey (DES) dataset. They have shown very high accuracy is achieved by using such network over this task for both training and test sets.

2.3. Galaxy Zoo Data Classification - Kaggle

The Kaggle Galaxy Zoo Dataset is the official dataset of Kaggle's: **Galaxy Zoo - The Galaxy Challenge**. This competition focused on minimizing the classification loss over all the possible classes for galaxies in the test dataset. The online competition drew a total of 427 competitors from 326 teams totaling around 3159 entries. The winning entry by **Sander Dieleman** used a CNN architecture with heavy data augmentation as part of the training and testing process. The architectur of the CNN was modified a little bit to exploit the rotation invariance of the images and increase parameter sharing. Considering that CNNs are very privy to overfitting the data, regularization for the parameters was implemented in the model by using Dropout layers (Srivastava et al., 2014)

Some work has been done in clustering the galaxy image data using CNN (Khan et al., 2018), however the technique used relies on features output by a CNN which is trained to minimize the classiication loss over the galaxy image dataset, and then applying t-SNE algorithm (Maaten & Hinton, 2008) to cluster the features in a lower dimensional space. This is different from a basic clustering approach because the features were extracted after training was done, which was supervised, while clustering is supposed to be completely unsupervised.

3. Approach

For this experiment we propose a more complex architecture that connects a pretrained CNN network to the DEC architecture. The main idea behind this network architecture is that CNNs are better suited for extracting features from images as compared to simple ANN model. We are using a pretrained networks because this way we are able to avoid training very large networks which takes a lot of time, computation power and training data. Also pretrained models have been proven very effective in extracting useful features from the images which can be fine tuned for the current problem using transfer learning (Yosinski et al., 2014).

3.1. Xception Architecture

For our network architecture we are primarily focusing on using the Xception (Chollet, 2017) network pretrained on the ILSVRC Challenge (Russakovsky et al., 2015). The Xception Network can be referred to in Figure: 2

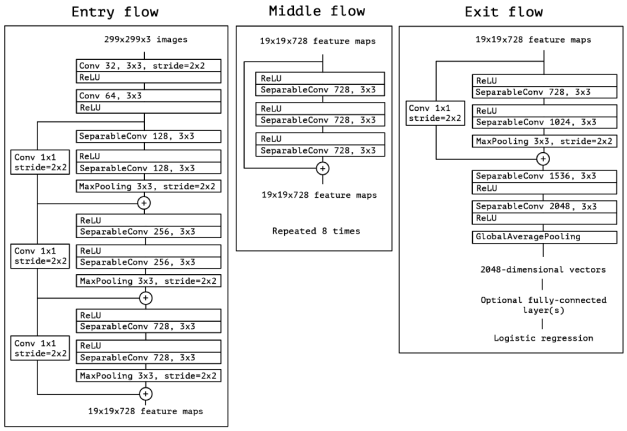

Figure 2. Xception Network Architecture

Xception architecture was proposed by Google as a modification of the original Inception network, such that it includes Depthwise Separable Convolutions. Deptwise Separable Convolutions help in reducing the number of opertaions it takes for a convolutional filter to move from an input feature space to an output feature space. It is able to achieve this by dividing a convolution into 2 sequential steps: Depthwise Convolution and Pointwise Convolution. Depthwise convolution (figure: 3) as the name suggests transforms the input image tensor into an intermediate output tensor that has the desired spatial dimensions but the same depth (number of channels). After the depthwise convolution, pointwise

convolution (figure: 4) transforms this intermediate image tensor and change its number of channels to match the final output image tensor shape using a 1X1 convolution which help preserve the spatial dimensions of the intermediate image tensor (same as output image tensor).

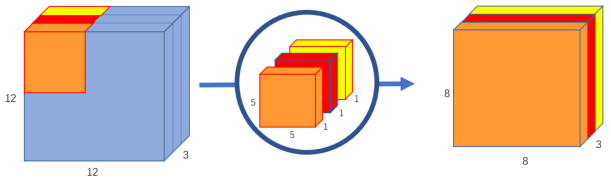

Figure 3. Depthwise Convolution

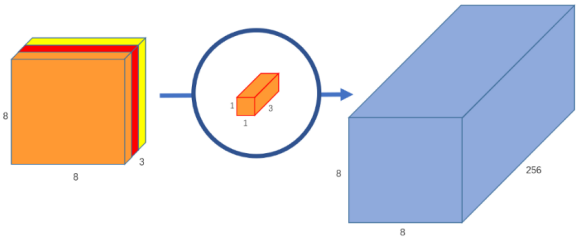

Figure 4. Pointwise Convolution

By employing these techniques the number of parameters decrease in the network, because of smaller size of the convolutions and lower number of convolution filters, without any deviation from the original transformation. Because of the decrease in number of parameters, the number of operations it takes for the transformation also decreases thus leading to a more efficient network. It has been proven that Depthwise Separable Convolutions is able to achieve similar or sometimes even better accuracy on the ILSVRC dataset over the preexisting networks because of the efficient use of the model parameters (Chollet, 2017).

3.2. Deep Embedding Clustering

Because Zooniverse, uses a clustering algorithm to speed up the label collection for the galaxies, this experiment uses Deep Embedding Clustering (Xie et al., 2016): an ANN based architecture that performs clustering on the input feature space as the clustering algorithm attached to the Xception CNN network. The DEC architecture can be visualized in Figure: 5.

The DEC network training for clustering can be divided into two steps as follows:

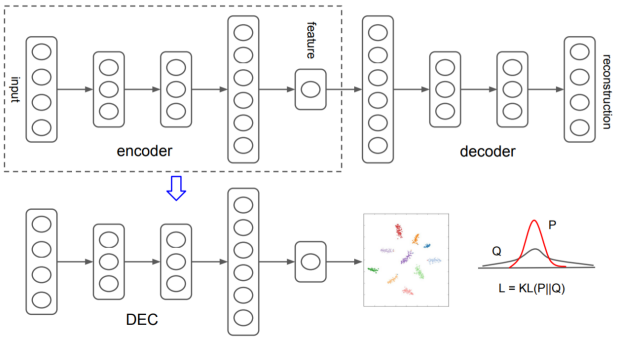

Figure 5. Deep Embedding Clustering Network Architecture

3.2.1. Feature Extraction

The initial step of DEC is to convert the input feature space into a lower dimensional feature space using an ANN stacked autoencoder network. The stacked autoencoder network is constructed using layer wise denoising autoencoder and training them greedily for each denoising autoencoder to reduce the reconstruction loss on the output of the previous denoising autoencoder network. A denoising autoencoder is a simple 2 layer neural network defined using equations (1)

$$\begin{aligned} \tilde{x} &\sim \text{Dropout}(x) \\ h &= g_1\left(W_1\tilde{x} + b_1\right) \\ \tilde{h} &\sim \text{Dropout}(h) \\ y &= g_2\left(W_2\tilde{h} + b_2\right) \end{aligned} \tag{1}$$

Here x is the features learned from the previous autoencoder, $Dropout(.)$ (Srivastava et al., 2014) is a stochastic mapping that randomly sets a portion of its input dimensions to 0, g_1 and g_2 are activation functions for encoding and decoding layer respectively, and $\theta = \{W_1, b_1, W_2, b_2\}$ are model parameters. Training is performed by minimizing the Reconstruction loss which is given by equation (2).

$$\|x - y\|_2^2 \tag{2}$$

By training the stacked autoencoder network, we are able to learn the weights for the encoder which is able to succesfully reduce the dimensionality of the input feature space and project it to a lower dimensional space which is still capable of reconstructing the original features.

3.2.2. Clustering

After the initial step of learning the weights of the encoder to project the features to a lower dimensional space, the architecture uses the K-Means Clustering algorithm to cluster

the encoded features and find the initial cluster center over the single pass of the whole dataset through the encoder. After that the network's weights and the cluster centers are updated using a gradient descent method which tries to minimize the K-L divergence (Kullback & Leibler, 1951). The weights update rule for the encoder features is given in equation (3), and the update rule for the clusters center is given in equation (4).

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha} \sum_j \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{3}$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \sum_i \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{4}$$

Here z_i are the encoder features, q_{ij} is the probability of z_i belonging to μ_j, and p_{ij} is auxiliary distribution based probability of z_i belonging to μ_j. The gradients $\partial L/\partial z_i$ are then passed down to the encoder network and used in standard backpropagation to compute the parameter gradient $\partial L/\partial \theta$. This step is repeated until the number of points that change their cluster assignments is below a certain user defined threshold.

4. Experiment

Before evaluating the performance of the architecture on the task of clustering the galaxies, we also evaluate the proposed architecture on the classification task of the image using a fully supervised training regime. This experiment is conducted to evaluate the performance of the architecture as compared to other past architectures. Apart from supervised training, several experiments were conducted for evaluating the clustering performance of the architecture. Experiments included using different image preprocessing techniques, different architectures of the DEC stacked autoencoder, using different clustering algorithms for the cluster initialization, manually initializing the clusters using labels from the dataset, semi supervised learning to help converge the model and obtain better clustering accuracy. For all the experiments except where discussed, we use the default stacked autoencoder architecture dimensions to 2048-500-500-2000-10 (Xie et al., 2016), along with the number of clusters as 2 which is equal to the number of target classes. The first layer of the network architecture is a 2048 dimensional because it corresponds to the size of the features extracted from the Xception network.

All of the experiments were conducted using code written in python with backend libaries like: tensorflow, keras, numpy, pandas, cv2, matplotlib, seaborn etc. We used ipython notebooks to conduct the experiments because of their ease of access for data analysis. The experiments for this project can be broken into 6 categories, which are discussed in detail.

4.1. Supervised Learning - Classification

We perform an experiment using Supervised Learning for classification of the galaxy images using a simple ANN network and also a CNN network using the Xception network architecture. The experiments for the supervised learning also includes changing the architecture of the ANN (fully connected layers at the end of the network). We perform this experiment to fgure out the best possible performance on the dataset and use this as a reference for the best performance that our unsupervised algorithm can achieve. The fully connected layers in the ANN and CNN network use **ReLU** ((INSERT REFERENCE)) activation (5), except for the output layer which uses **Sigmoid** activation (6) because we are attempting a binary classification problem.

$$f(x) = x^+ = \max(0, x) \tag{5}$$

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

The loss of the network is **binary crossentropy** (7), and the network is trained using the **adam** ((INSERT REFERENCE)) optimizer.

$$BCE = \begin{cases} -log(f(s_1)) & if \quad t_1 = 1 \\ -log(1 - f(s_1)) & if \quad t_1 = 0 \end{cases} \tag{7}$$

where s_1 is the output of the network after applying the sigmoid activation, and t_1 is the true class label.

4.2. Data Preprocessing

For this project we consider 2 different categories of data preprocessing.

4.2.1. Image Preprocessing

Considering that our input data is galaxy images of size 424 by 424 pixels, we use different image preprocessing techniques to reduce the dimensions of the image to avoid curse of high dimensionality. We reduce the size of image by either resizing the images to a lower dimension, or cropping to a smaller part of the image.

4.2.2. Feature Preprocessing

As our inputs are galaxy images, the features to the networks are the RGB values of each pixel from the image. The RGB values are integers falling in the range from 0 to 255. However it is always preferred to preprocess the values for the features into smaller values or values corresponding to some distribution ((INSERT REFERENCE)). To accomplish this we consider two techniqes as follows:

- Normalization to a [0, 1] range
- Global Standardization of the features

4.3. DEC Stacked Autoencoder Architecture

4.4. Clustering Algorithm

4.5. Cluster Initialization

4.6. Semi Supervised Learning

5. Results

6. Analysis

7. Conclusion

References

Abraham, R. G., Van Den Bergh, S., and Nair, P. A new approach to galaxy morphology. i. analysis of the sloan digital sky survey early data release. *The Astrophysical Journal*, 588(1):218, 2003.

Bazell, D. and Aha, D. W. Ensembles of classifiers for morphological galaxy classification. *The Astrophysical Journal*, 548(1):219, 2001.

Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

De La Calleja, J. and Fuentes, O. Machine learning and image analysis for morphological galaxy classification. *Monthly Notices of the Royal Astronomical Society*, 349 (1):87–93, 03 2004. ISSN 0035-8711. doi: 10.1111/j.1365-2966.2004.07442.x. URL https://doi.org/10.1111/j.1365-2966.2004.07442.x.

Goderya, S. N. and Lolling, S. M. Morphological classification of galaxies using computer vision and artificial neural networks: A computational scheme. *Astrophysics and space science*, 279(4):377–387, 2002.

Khan, A., Huerta, E., Wang, S., and Gruendl, R. Unsupervised learning and data clustering for the construction of galaxy catalogs in the dark energy survey. *arXiv preprint arXiv:1812.02183*, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Madgwick, D. S. Correlating galaxy morphologies and spectra in the 2df galaxy redshift survey. *Monthly Notices of the Royal Astronomical Society*, 338(1):197–207, 2003.

Naim, A., Lahav, O., SodrĂ©, L., J., and Storrie-Lombardi, M. C. Automated morphological classification of APM galaxies by supervised artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 275(3):567–590, 08 1995. ISSN 0035-8711. doi: 10.1093/mnras/275.3.567. URL https://doi.org/10.1093/mnras/275.3.567.

Odewahn, S., Cohen, S., Windhorst, R., and Philip, N. S. Automated galaxy morphology: A fourier approach. *The Astrophysical Journal*, 568(2):539, 2002.

Owens, E. A., Griffiths, R. E., and Ratnatunga, K. U. Using oblique decision trees for the morphological classification of galaxies. *Monthly Notices of the Royal Astronomical Society*, 281(1):153–157, 07 1996. ISSN 0035-8711. doi: 10.1093/mnras/281.1.153. URL https://doi.org/10.1093/mnras/281.1.153.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Storrie-Lombardi, M., Lahav, O., Sodre Jr, L., and Storrie-Lombardi, L. Morphological classification of galaxies by

artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 259(1):8P–12P, 1992.

Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487, 2016.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

A. Appendix