G-11: Galaxy Image Classification: A comparison between ML and DL techniques

Ishan Bhatt (ivbhatt), Meghana Kota (mkota), Pragna Bollam (pbollam), Shilpa Kancharla (skancha)
Department of Computer Science, North Carolina State University,
Raleigh, NC, USA
e-mail: [ivbhatt, mkota, pbollam, skancha] @ncsu.edu

Code: https://github.ncsu.edu/ivbhatt/ALDA-G11-GalaxyClassification

1. Problem Statement

SDSS telescopes have captured over 40TB worth of galaxy images and classification of these images is the first step towards obtaining a deeper understanding of physical processes within them, star formation, and the nature of the universe. Since we couldn't find an easily-accessible dataset for Galaxy-classification, we believe that **compiling a dataset for Galaxy classification and providing benchmarks with some of the common learning-algorithms would help in automating the Galaxy-classification which until recently had to be performed by-hand by expert astronomers.** We are classifying the images of galaxies into four classes- spiral, elliptical, irregular, and invalid.

We aim **to build a dataset for this domain**. We also want to benchmark the dataset using some of the most common learning (ML/DL) algorithms. To apply ML algorithms, we will apply **PCA** on generated dataset to reduce the dimensionality and extract relevant features. Once we have features ,We will be applying some of the major **ML techniques (SVM, MLP and Random Forest). We plan to train the CNN** directly using the images (without PCA).

2. Related works

[1]Nour Eldeen M. Khalifa, Mohamed Hamed N. Taha, Aboul Ella Hassanien , I. M. Selim; Deep Galaxy: Classification of Galaxies based on Deep Convolutional Neural Networks

For galaxy classification, a deep convolutional neural network architecture with 8 layers, and one main convolutional layer for features extraction with 96 filters, and two principal fully connected layers is applied. Based on the features, classification is done into three categories - spiral, elliptical, irregular.

[2]Siddhartha Kasivajhula, Naren Raghavan, Hemal Shah; Morphological Galaxy Classification Using Machine Learning

Set of Morphic features generated from Image analysis and direct image pixel data compressed through PCA are used to apply ML algorithms like Support Vector Machines (SVM), Random Forests (RF), and Naïve Bayes (NB) to classify the images. Finally, the performance of these algorithms on the data is compared on both morphic and PCA features.

[3]Jorge De La Calleja, Olac Fuentes; Machine learning and image analysis for morphological galaxy classification

For morphological galaxy classification, a neural network, a locally weighted regression method, and homogeneous ensembles of classifiers are used. Data is augmented and PCA is applied to reduce dimensionality and get relevant features. A 10 fold Cross Validation is applied on homogeneous ensemble regression methods to classify images into three categories - spiral, elliptical and irregular.

3. Approach

We apply **Principal Components Analysis (PCA) to our images in order to reduce the dimensionality of the data for ML approaches.** Since image data requires a lot of storage, we compress the images in order to preserve the most important features of the image data. **We then apply ML classification techniques to the compressed data**.

The first classification technique applied is **Support Vector Machines (SVM)**. This classifies the images into four classes by creating boundary hyperplanes between classes. It is more robust i.e. due to optimal margin gap between separating hyper-planes, it could do predictions better with test data. **SVMs are simple and efficient, and less likely to overfit.**

Moreover, we apply the **Random Forests** algorithm. We understand that random forests are bagged decision trees that are split on a random subset of features. Due to the fact that this technique **splits on a random subset of features, it reduces the variance and is robust to outliers.** Therefore, we have the

potential to produce a robust galaxy classification model.

4. Rationale

We use **PCA for feature selection**, as our data set is big, extracting more relevant features is helpful in efficient use of computational resources. **ML approaches we are using cannot handle datasets with very high dimensionality, therefore we need a way to reduce the dimensionality** by selecting the best features. PCA skips less significant components. Another approach we could consider for dimensionality reduction was **t-distributed stochastic neighbor embedding (t-SNE)**, which requires hyperparameter tuning and relies on a probabilistic distribution. We opted to use **PCA because it is a more straightforward and reliable approach.**

The **SVM approach can be used for relatively high-dimensional data, such as image data**. SVM clearly classifies the data based on kernel equations. An alternative approach to SVM we considered was **logistic regression**. However, **logistic regression would not provide us with clear margins that separate data,** but rather just probabilistic boundaries.

Random forest algorithm considers random selection of features at each node to determine the split. We decided to use random forests instead of **decision trees** because a **random forest is more robust than a single decision tree.** Compared to a single decision tree, random forests reduce bias by aggregating multiple decision trees and can produce more accurate results.

5. Dataset

We choose to use the full catalog of the Galaxy Zoo 1 dataset as our input. This dataset provides us with an object ID, coordinates to where a celestial object is, and a one-hot encoding of the category of the galaxy (*Elliptical, Spiral*) as well as an attribute that reflects the quality of the classification. The Sloan Digital Sky Survey provides an API from where we can fetch images of galaxies given their coordinates. Out of the 600K+ images in the original dataset, we pick only about 10K+ high-quality data points. The API provides functionality to specify size of the image. **We used this API to get a 1000 images each of *Elliptical* and *Spiral* galaxies.** Based on initial analysis, we believed that 512x512 is a reasonable input size. However, at the time of applying PCA to this dataset, **we had to reduce the dimensions further to 128x128** as the RAM was not sufficient while running PCA on the dataset.

Additionally, we **web-scraped about 200 images of *Irregular* galaxies;** removed duplicate (faulty) images and spiral or elliptical images from them manually. Later, **data augmentation is applied on them to get about 1000 images of *Irregular* category. We also scraped 828 non-celestial object images and added them to our dataset and labelled them as *Invalid*.** Finally our generated dataset consists of **991 elliptical, 1001 spiral, 1000 irregular galaxies and 828 invalid images adding up to a total of 3820 images**. Each image is to be classified in one of the following four categories: *Elliptical, Spiral, Irregular,* and *Invalid.*

6. Hypotheses

Hypothesis 0.1: Even after implementing Z-score normalization, we expect to have total-variance in the dataset to be equal to 1, in reality, **there should always be a small difference between the variance expected and the variance obtained.**

Hypothesis 0.2: Even though it is recommended, it is not required by ML algorithms that all classes are represented exactly equally in the dataset. **We believe the learning techniques should still be able to achieve respectable classification accuracy with slightly imbalanced datasets.**

Hypothesis 1: If about 95% variance of the original dataset is preserved in the Principal Components obtained after PCA, then **we should be able to reconstruct the original images using PCs.**

Hypothesis 2: It would be interesting to compare **performance between SVMs and RandomForests.**

7. Experimental design

Using the API provided by the Sloan Digital Sky Survey, we scraped 2000 images (100 each of elliptical and spiral galaxies).

We scraped about 800 images of non-celestial objects (class:invalid) to help our models learn more robust features & learn to differentiate between galaxies and other objects.

Finding images of irregular galaxies was a bit of a challenge. We managed to scrape about 200 images of unique irregular galaxies and then used data-augmentation to generate 1000 images from them.

For data augmentation (irregular galaxies ONLY), we took web-scraped images and removed duplicate images manually. Then, we applied the following transformations:

- ChannelShuffle (RGB to any other order) with a probability of 0.35
- Zoom-in; Zoom-out upto 10% on all images.
- Horizontal Flip or Vertical Flip with a probability of 0.5
- Rotation by 90, 180, 270 or 360 degrees with a probability of 0.5

We generated exactly 1000 images of irregular galaxies using the above transformations.

Figure 1.1: Irregular images before applying Data Augmentation

Figure 1.2: Irregular images after applying Data Augmentation

After augmentation, we resize images of all classes to 128x128 and combine them together. Our final dataset consists of:

- 1001 *Spiral* images
- 1000 *Irregular* images
- 991 *Elliptical* images
- 828 *Invalid* images

On these images, PCA was applied to reduce dimensionality. For PCA on images, the following transformations are applied:

- Flatten Images
- Z-Score Normalization

After the above transformations, each image is a vector of size 128*128*3 = 49152 and so after Z-score normalization we expect the total variance to be equal to 49152. However, we find that observed total variance is about 49116 which is slightly less than the theoretical expectation. Hence, **hypothesis 0.1** is confirmed.

After PCA, we would like to preserve at least about 95% of the variance present in the images. We would like to use the least number of PCs to do so. We start with 1 and go on doubling the number of PCs until we reach a point where 95% of the variance is preserved. It is observed that for number_PCs = 256, a variance of 46784.8 is preserved which is about 95% of variance of total dataset. And so, we will use 256 PCs to reduce the dimension. The below plot gives the variance explained by principal components on the dataset.

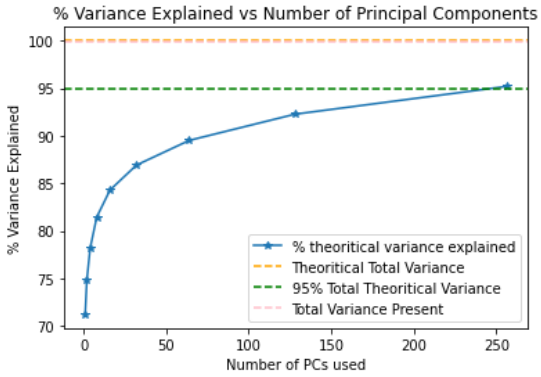

Figure 1.3: Percent of variances explained by number of Principal Components

Now, we reconstructed the images on the dataset using the PCs (as a sanity check) and it is observed that even after reducing the dimensions, the images preserved important features. This provides evidence for **Hypothesis 1**, thus showing that we are able to reconstruct the image to some degree using the PCs.

Figure 1.4: Sample Images before applying PCA (high dimension)

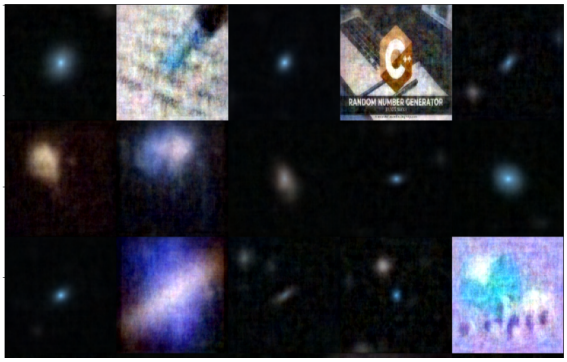

Figure 1.5: Sample images after applying PCA (reduced dimension)

After applying PCA on the dataset, we plan to split our data into training and validation sets in order to create a baseline model. We use 70% of the data for training and 30% for validation.

For training on dataset, both SVM and Random Forest Classifier are used. We compared the performances of different kernels (like rbf, poly, sigmoid) in SVM and we observed that a **linear kernel** gave higher accuracy for the images classification. For the Random Forest classifier, **Gini** is taken as a split criterion and the number of random trees to be generated are taken as **100**. For testing, the predictions of both SVM classifier and Random Forest Classifier are given. We are generating the classification reports with **precision, recall, F1-Score, and accuracy** for both SVM and Random Forest predictions. Also, an ROC curve is plotted to compare the **True Positive rate** and the **False Positive rate**.

8. Partial results

For our experimental run with a linear SVM, we achieved 86.13% accuracy, with the precision, recall, and F1-scores for this model displayed in Table 1.1. Moreover, using the **random forest method, we achieved 92.06% accuracy** with the aforementioned metrics for it displayed in Table 1.2. We found that the accuracy of our random forest run was higher than that of our linear SVM, thus corroborating our **Hypothesis 2**. Furthermore, in relation to **Hypothesis 0.2**, we found that even though the distribution of our data was not equal, we were still able to achieve accuracy above 80% as described above.

The below Figure 1.6 shows the predictions of each image along with the actual class label. The tables 1.1 and 1.2 are the classification reports with precision, recall and F1-score for SVM classifier and Random Forest Classifier respectively.

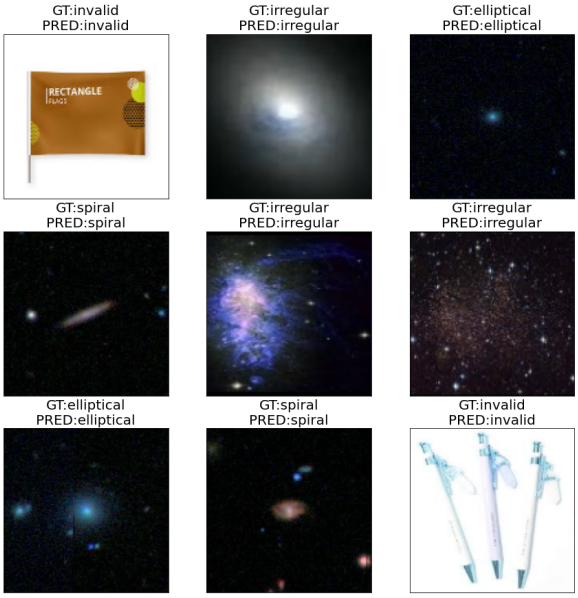

Figure 1.6 Sample images with their actual class and predicted class

Table 1.1: Classification Report of Linear SVM

Class	Precision	Recall	F1-Score
0	0.7936	0.8806	0.8349
1	0.8621	0.8143	0.8375
2	0.8639	0.9414	0.9010
3	0.9745	0.7992	0.8782

Table 1.2: Classification Report of Random Forest

Class	Precision	Recall	F1-Score
0	0.8978	0.9065	0.9021
1	0.9193	0.8534	0.8851
2	0.9123	0.9690	0.9397
3	0.9625	0.9665	0.9645

In Figure 1.7 and Figure 1.8, we see the ROC curve for the linear SVM and random forest classifiers, respectively. The ROC curves further evince that the random forest classifier achieves better performance than the linear SVM. We can see that in Figure 1.8, the true positive rate is very close to 1.0 while the false positive rate is very close to 0.0 for classes 2 and 3 compared to the curves in Figure 1.7 for classes 2 and 3. We can also see that the curves for classes 0

and 1 are closer to the (0, 1) point of the curve in Figure 1.8 than in Figure 1.7.

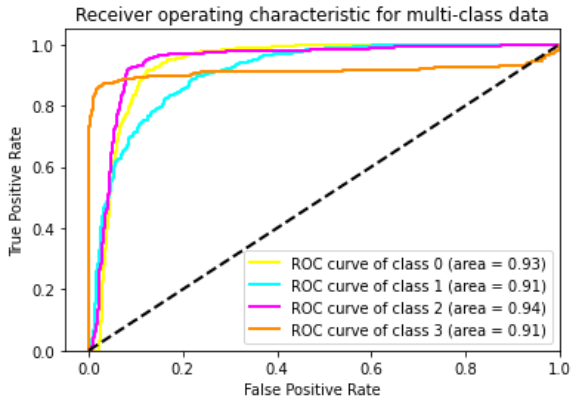

Figure 1.7: ROC curve for the linear SVM.

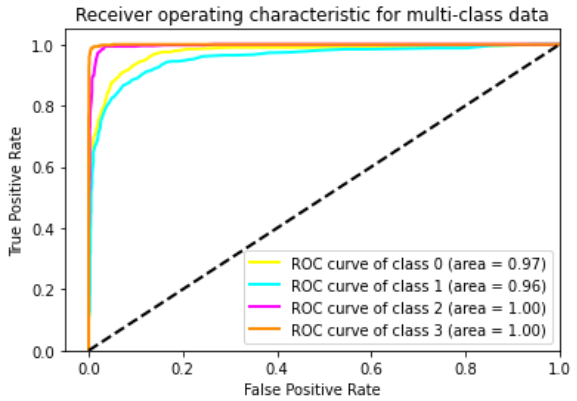

Figure 1.8: ROC curve for the Random Forest Classifier

9. Design Of Future Experiments

We plan to train a Multi-Layer Perceptron on the PCA generated features of images. We plan to train a Convolutional Neural Network (CNN) - on the image data, In order to optimize the model performance we use hyper-parameter grid-search methods on the generated dataset. Final performance of all algorithms and metric-presentation scripts will be given as results.

We plan to implement these algorithms using Python3 and available packages (pandas for data preprocessing; scikit-learn for ML algorithms; keras for CNN; pandas, numpy for data-manipulations; matplotlib for presentation).

10. Plan of activities

- *Member-1* and *Member-2* will implement MLPs.
- *Member-3* and *Member-4* will implement a CNN model.
- All team members will compile and analyze the experimental data.

11. Online meetings

- Meeting 1 on 03/15/2021 for 45 minutes- Discussed about project ideas to implement.
- Meeting 2 on 03/20/2021 for one hour- Focused on this topic and how to implement this project, which algorithms to implement.
- Meeting 3 on 03/22/2021 for one hour- To complete the project proposal document.
- Meeting 4 on 03/28/2021 for one hour- Assigned tasks for each member and briefly talked about the procedures to follow.
- Meeting 5 on 04/05/2021 for 2 hours - Discussed problems concerned with deployment and resolved those issues.
- Meeting 6 on 04/07/2021 for 2 hours- To complete Midway report.

Tentative Schedule:

- Meeting 7 on 04/09/2021 for discussing MLP implementation.
- Meeting 8 on 04/16/2021 for discussing CNN model implementation.
- Meeting 9 on 04/23/2021 for discussing optimization of CNN model.
- Meeting 10 on 04/29/2021 for finalizing all submission software and the report.

References

[1] Nour Eldeen M. Khalifa, Mohamed Hamed N. Taha, Aboul Ella Hassanien , I. M. Selim; Deep Galaxy: Classification of Galaxies based on Deep Convolutional Neural Networks

[2] Siddhartha Kasivajhula, Naren Raghavan, Hemal Shah; Morphological Galaxy Classification Using Machine Learning

[3] Jorge De La Calleja, Olac Fuentes; Machine learning and image analysis for morphological galaxy classification

[4] Chris J. Lintott, Kevin Schawinski, Anze Slosar, Kate Land, Steven Bamford, Daniel Thomas, M. Jordan Raddick, Robert C. Nichol, Alex Szalay, Dan Andreescu, Phil Murray, Jan van den Berg; Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey

[5] Yanxia Zhang , Yongheng Zhao; Astronomy in the Big Data Era