CS 365: Artificial Intelligence

Galaxy Zoo Challenge

Project Report

Rohit Kumar Jha
11615
rkjha@cse.iitk.ac.in

Irfan Hudda
11319
irfanh@cse.iitk.ac.in

April 24, 2014

Advisor: Amitabha Mukerjee
amit@cse.iitk.ac.in

Contents

List of Figures

Abstract:

Galaxy Zoo is a project with aim of classifying the images from *Sloan Digital Sky Survey* and *Hubble's CANDELS*. The Galaxy Zoo Challenge was a competition with aim of selecting best approaches for this problem. In this report we attempt to explain various approaches we tried and also compare them with each other.

1 Introduction

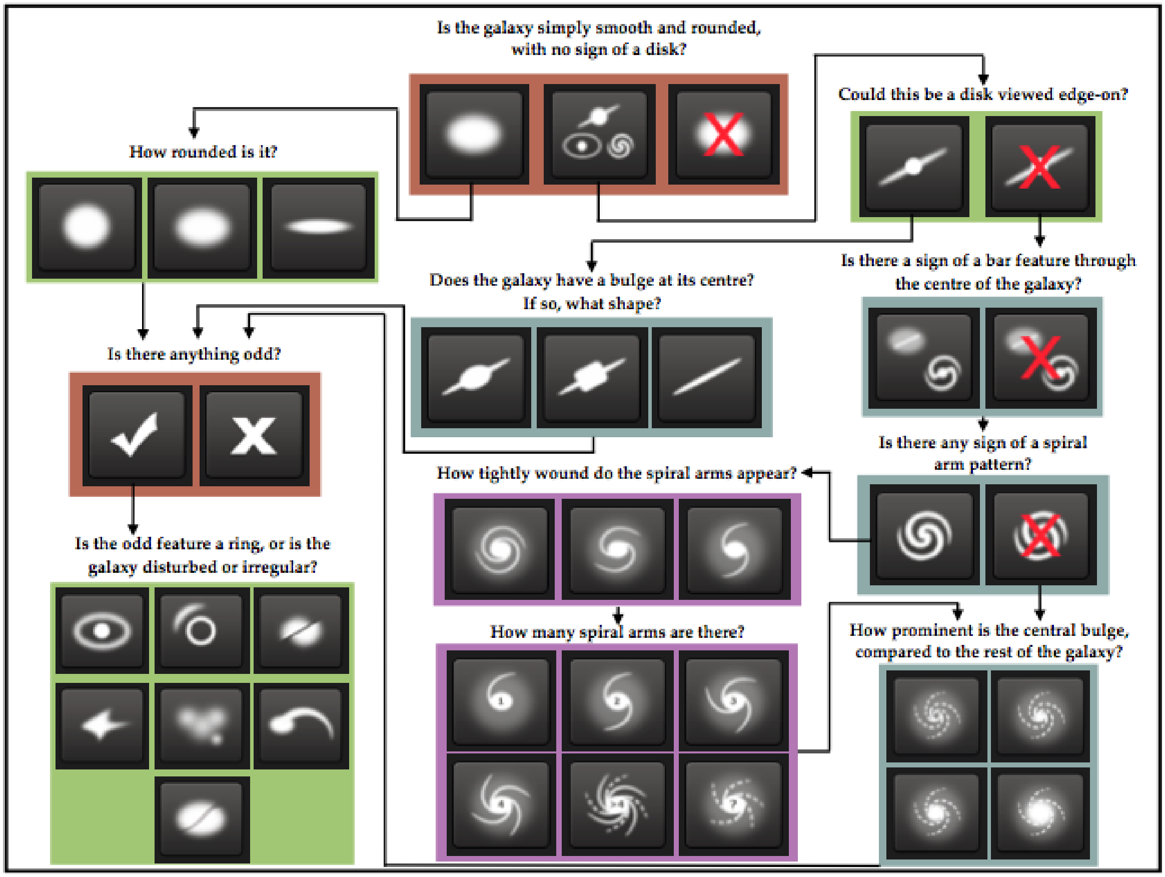
Is the galaxy simply smooth and rounded, with no sign of a disk?
Could this be a disk viewed edge-on?
How rounded is it?
Does the galaxy have a bulge at its centre? If so, what shape?
Is there a sign of a bar feature through the centre of the galaxy?
Is there anything odd?
Is there any sign of a spiral arm pattern?
How tightly wound do the spiral arms appear?
Is the odd feature a ring, or is the galaxy disturbed or irregular?
How many spiral arms are there?
How prominent is the central bulge, compared to the rest of the galaxy?

Figure 1. Flowchart of the classification tasks for GZ2, beginning at the top centre. Tasks are colour-coded by their relative depths in the decision tree. Tasks outlined in brown are asked of every galaxy. Tasks outlined in green, blue, and purple are (respectively) one, two or three steps below branching points in the decision tree. Table 2 describes the responses that correspond to the icons in this diagram.

Figure 1: Galaxy Zoo Decision Tree[8]

We are given 61578 images of galaxies with probabilities corresponding to each of the 37 classes shown in above figure. The test set contains 79975 images for which we need to predict the probabilities corresponding to each class.

2 Motivation

In past few decades, the desire of humans to know more about other galaxies has increased and so has their efforts. We want to know the most fundamental questions about our existence, how and why. A part of the answer to our question lies in how the galaxies originated and evolved over time. Different galaxies have varying shapes, sizes, colors and features. And to solve the puzzle of formation and evolution of galaxies, we need to understand how can we infer the distribution, location and type of galaxies on the basis of their shapes, size and color. This, in turns, requires us to classify the galaxy images based on their shapes, sizes and other features.[2]

In an earlier successful citizen science crowd-sourcing project, hundreds of thousands of volunteers helped classify shapes of some millions of these images by eye. But with growing data, it became difficult to do this manually any more. So, an initiative was launched to find good automated metrics that could potentially be used to analyse the images of the galaxies and answer these questions.[2] [3]

3 Approach

This challenge is really interesting in its own senses. The biggest challenge one has to face is the large data size that needs to be handled. The most trivial, natural and promising approach for this problem is to use Convolutional Neural Nets. But the presence of large training/test data (2 GB) turns out to be a big problem for CNN approach. With few layers ($3-4$), it easily takes around days to train the CNN on this data. And to get good results one needs to train at least on $3-4$ layers. As it turns out, more than 80% of the contestants in top 50 used CNN and the ones on the top trained it for $7-9$ layers. Clearly, this approach was not feasible for us. And training the data on CNN with very few layers won't give good results. So, we chose a different approach altogether. The aim was not to get just good results but to see what alternate approaches one can adopt when similar situations arise.

We had to choose some approach that we could, potentially, run or at least test on our systems. So, we started out with the very simplest of approaches.

3.1 Central Pixel Approach

Central Pixel approach[1] is a very simple approach and works really well in practice. In this approach, we first crop a $10x10$ image at the center of each training set image. We then average the RGB values for these pixels and obtain the average intensity as follows:

$$avgIntensity = int((RGB[0] + RGB[1] + RGB[2])/3) \tag{3.1}$$

We then obtain a value as follows, using the value of *avgIntensity* calculated in 3.1:

$$RGB[0]/ = avgIntensity$$
$$RGB[1]/ = avgIntensity$$
$$RGB[2]/ = avgIntensity$$

$$hashValue = RGB[0] * factor * factor + RGB[1] * factor + RGB[2] \tag{3.2}$$

The value of factor was determined experimentally determined to be 10.
We perform this for all training images and the hashed value is used to create clusters with similar colors. For each cluster of colors, we average the 37 probability values for each galaxy. Now, for each of the image in test set, we find the color of the central $10x10$ image and hash it to obtain a number. We then find the matching cluster and assign the Class values for that cluster to the test image. This approach gives a *RMSE* of 0.16 when trained on the entire training data and tested on the given test data. This approach is really fast compared to any other approach that doesn't use random guessing.

3.2 Classifier Focused Approach

All the images in the test and the training dataset were cropped to $150x150$, while retaining the color information. The idea to retain the color was that color has a significant impact on the way we visualize objects, and their shapes in particular. Since, we are asked to predict the way a human would do, we retained the color information. The images were resized to $50x50$. Each of the pixel was used as a feature and we then used a random forest regressor to train and predict the expected values for each of 37 class. It gave a *RMSE* of 0.14 when we trained it on one-half of training data and tested on another half of the training data.[1]

Over time our experiences have taught us that there is always one Classifier/Regressor suited best for a particular problem. And we definitely didn't know if it was a Random Forest Regressor in this case. So, we stacked Random Forest Regressor, Extra Trees Regressor, Ada Boost Regressor and Gradient Boosting Regressor. The advantage of using stacking is that it performs at least as well as the best Regressor in the ensemble. For each of the regressor, we had $3 - 4$ entries obtained by varying the number of parameters. It has been observed in practice that this performs really well in contrast to any single Regressor.[2]

3.3 Exploring Features

It is pretty much clear in case of images that choice of features is significantly more important than the choice of classifier(s). As it turns out, CNN perform the best for feature extraction in case of images. It was finally evident from the results of Galaxy Zoo Challenge where top three participants ended up using CNN. With CNN being out of option for us, we rather decided to explore the features that we can possibly use and the contribution that each feature has.

3.3.1 Basic Features

Basic features for the image are

[1] As Kaggle doesn't provide the solutions for the test data, we had to check on the training data itself.

[2] This particular method required even more memory and computation power and running it was beyond the capability of our systems.

- Entropy feature of image
- Kurtosis and Skewness Coefficients for Image Histograms
- Average of RGB values for center of image.
- Gini Index.
- Maximum Brightness.

We also tried fitting and ellipse to the image as shown in Figure 3.3.1.

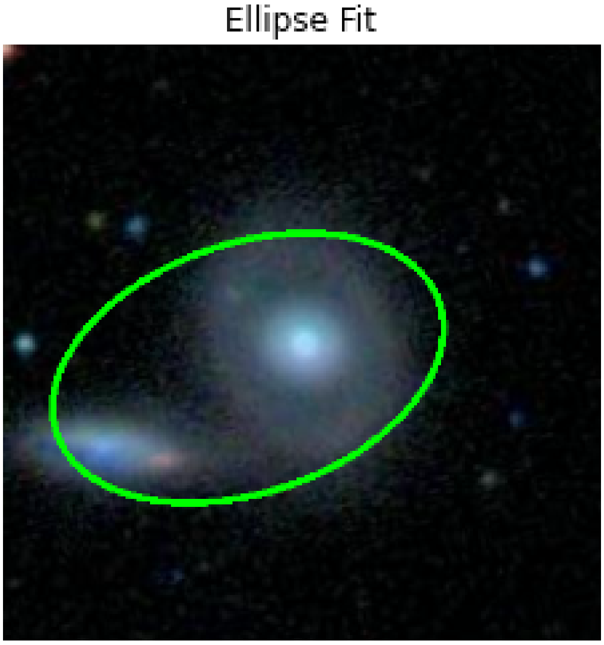

Figure 2: Ellipse fit for the Galaxy Image

From this ellipse fit we used following as features.

- Center of Ellipse.
- Eccentricity.
- Length of major axis.

3.3.2 Contour Based Features

We used number of contours in image at various threshold values. Contours are shown in figures below

For the image the number of contours vary with threshold value. We used this as feature. More precisely the features were

- Contours at 20% threshold.
- Contours at 50% threshold.

This feature could be useful for the question “Is there anything odd?”.

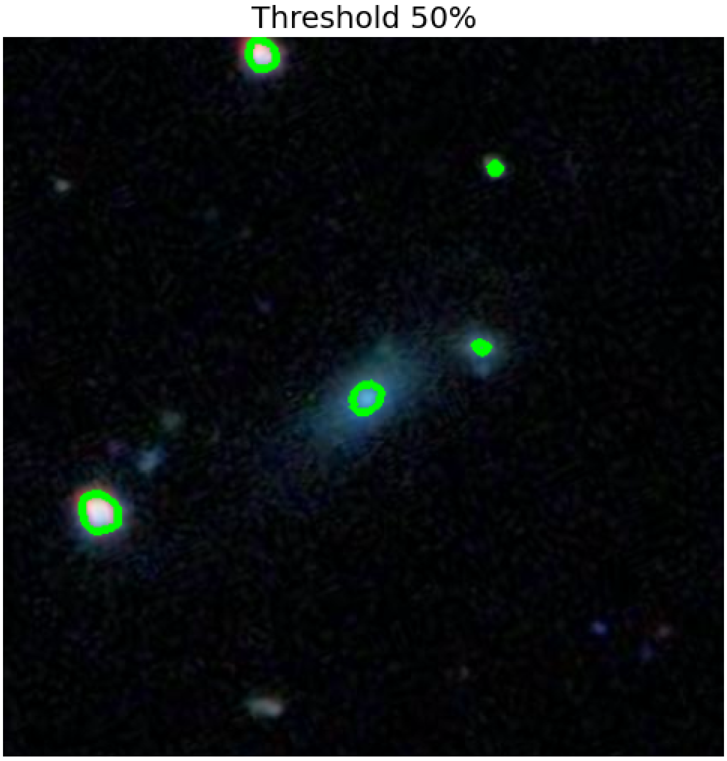

Figure 3: Contours at 50% Threshold

3.3.3 Light Based Features

For galaxy images we could use Light based features using the radial profile of the image from the center found using ellipse fit for galaxy described earlier. Example of the radial profile is the Figure 3.3.3.

The Figure 3.3.3 show the comparison of two galaxies with different radial profile characteristics.

We could also take the cumulative sum of the radial profile and use it to find the radii with 20%, 40%, 60% brightness. This is shown in Figure 3.3.3.

We used the ratios

- Radius with 20% : Radius with 40% intensity.
- Radius with 40% : Radius with 60% intensity.

3.3.4 Texture Features

The features used here were from `skimage.feature.greycoprops`. This calculates the texture properties of Grey Level Co-occurrence Matrix (GLCM).[11] The various features extracted from the image are[5]

- *contrast* $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$

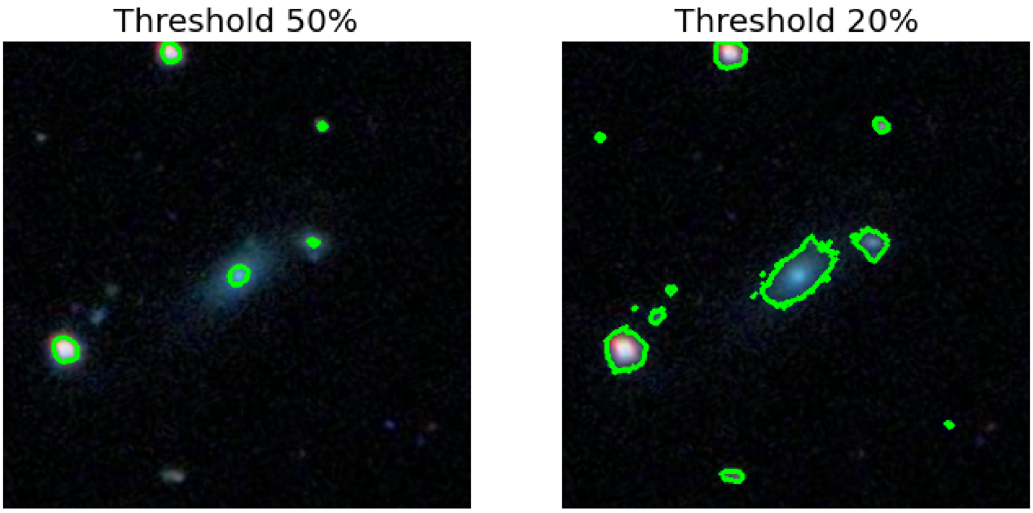

Figure 4: Comparison of Contours at 50% and 20% Threshold

- *dissimilarity* $\sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$
- *homogenity* $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- *ASM* $\sum_{i,j=0}^{levels-1} P_{i,j}^2$
- *energy* $\sqrt{ASM}$
- *correlation* $\sum_{i,j=0}^{levels-1} P_{i,j}\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2\sigma_j^2}}$

3.3.5 Color Based Features

Although it might appear that color of the galaxy image does not effect morphology of galaxy but it has been found that color effect the the classification be humans.[9]

From the paper[9] we also used following features

- $R-B$.
- $R-B$.
- $G-B$.

In addition to the above we used the properties of color histogram also as the feature. An example of color histogram is given in the Figure 3.3.5. This color histogram distribution was used to extract mean, standard deviation and also used them as feature.

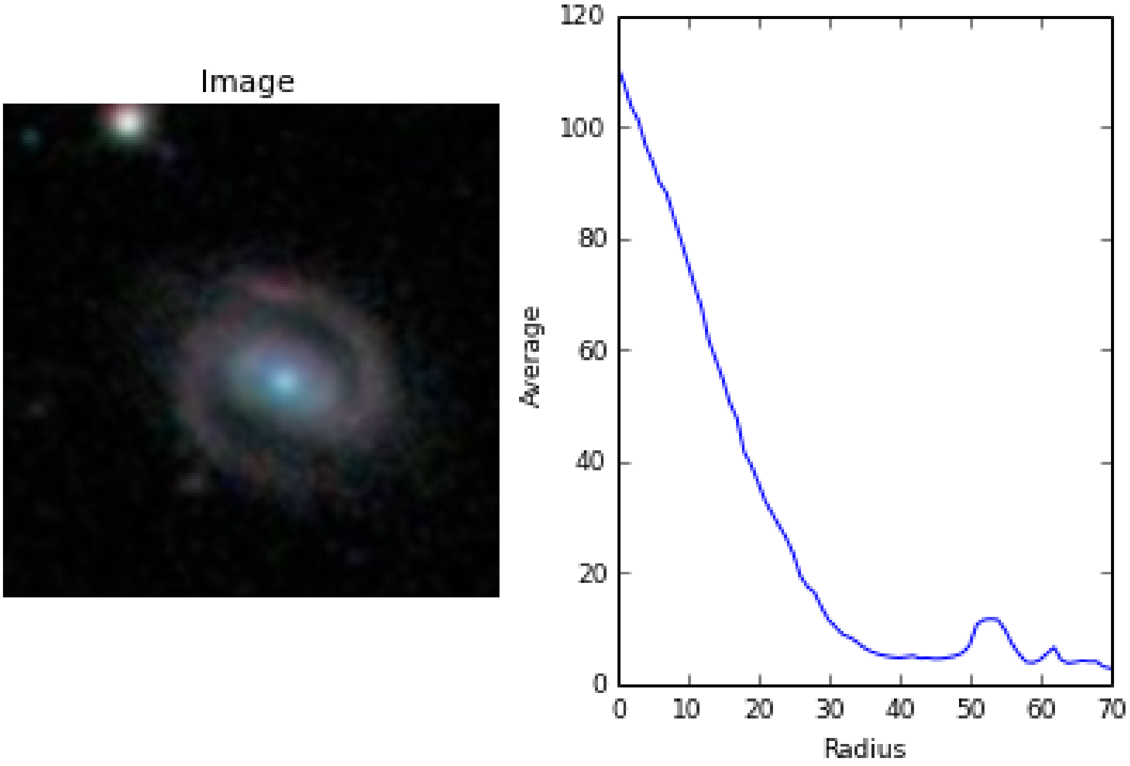

Figure 5: Radial Profile of Galaxy Image

3.3.6 Eigenfaces

Corresponding to each question we took images with probability above a threshold(> 0.95). And used these images to get eigenfaces.

We used Randomized PCA[10] to extract top eigenfaces from the images for each question.[7] It is one of the most important features used. The eigen faces are shown in Figure 3.3.6

3.4 Final Feature Based Approach

This problem was basically a regression problem with values given for each of 37 classes. We converted it into a classification problem. For each of the 11 question, see image above, and each sub-class, we chose the images unambiguously belonging to that sub-class. Then we extracted the below mentioned features. Some of the major features used were:

- Gini, Kurtosis and Skewness Coefficients for Image Histogram Distribution Inequality
- Entropy feature of an image
- Contour feature: Ellipse fitting
- Contour Feature: Bounding box
- *Chirality:* An object is chiral if it is not identical to its mirror image

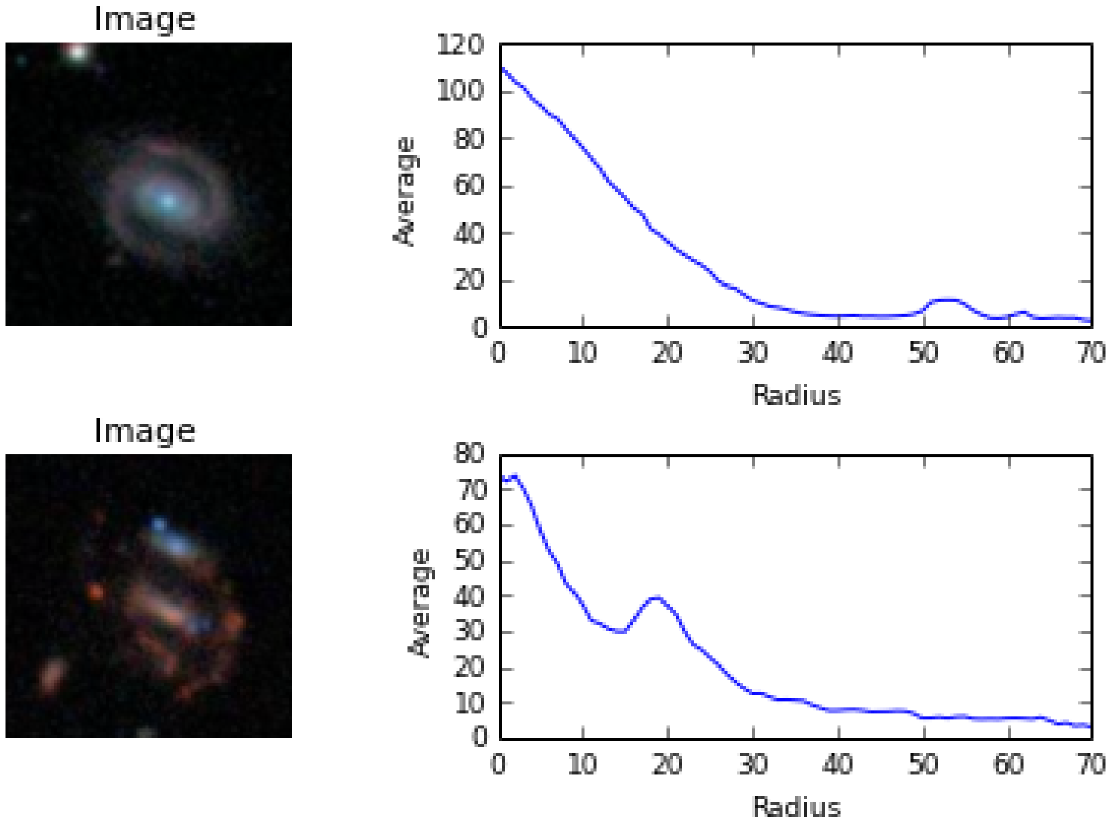

Figure 6: Comparison of Radial Profile

- Central Moments (Contours) which are invariant to size, position and orientation

- Blob features: Blobs provide a complementary description of image structures in terms of regions, as opposed to corners that are more point-like.

A lot of other, apparently insignificant features, were also used which we have described before. Three set of different and contrasting features were selected and then used SVM to get predicted probabilities for each class. These results were used as input to Extra Trees Regressor to get the final values for each of the 37 classes.

It gave a *RMSE* of 0.115 when we trained it on one-half of training data and tested on another half of the training data.[1]

4 Preprocessing

There were few ideas that were really promising but it didn't make sense to implement then as they significantly increased the computation time and cost. We briefly discuss few of them. One of the things we learnt was that the images of galaxies are invariant under rotation. In short, there is no up or down in

[1] As Kaggle doesn't provide the solutions for the test data, we had to check on the training data itself.

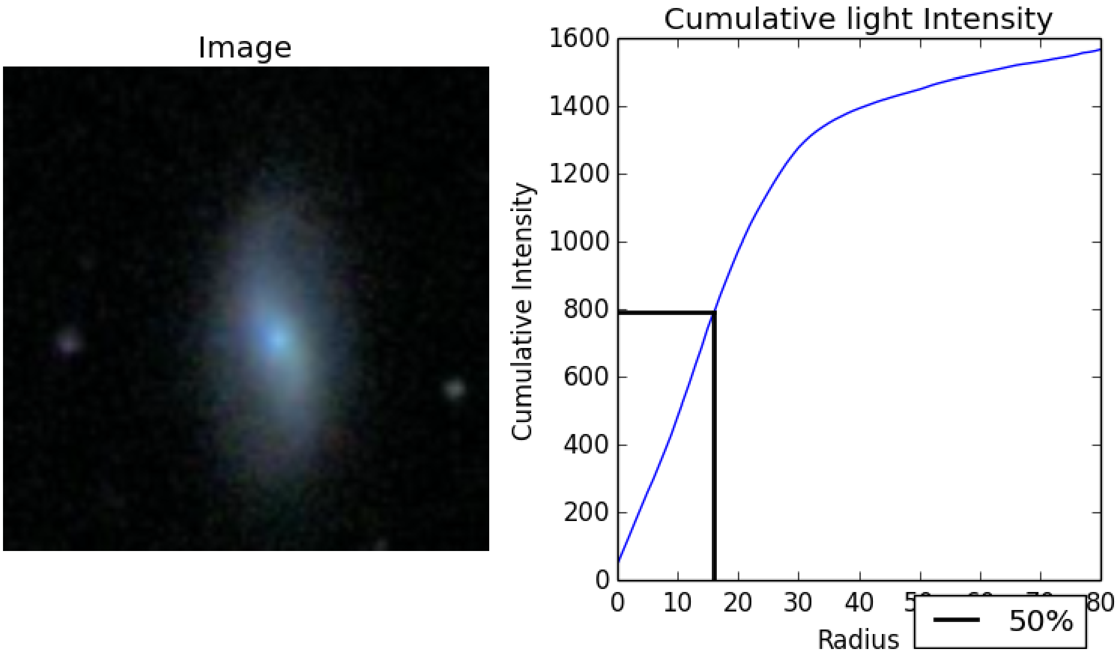

Figure 7: Cumulative light intensity and radius with 50% intensity

space. They are also scale invariant and translation invariant to a limited extent. It was possible to exploit all of these invariances to create new training data by perturbing the existing data points. We could easily do the following transformations to increase the training data size:

- *rotation:* rotating with a random angle in between $0 \deg$ and $360 \deg$
- *translation:* random translation of between -4 to 4 pixels
- *zoom:* zooming with scale factor between 1/1.5 to 1.5
- *flip:* taking mirror image of the original image

This helps avoid over-fitting in case the underlying CNN network is large.
We could also do a second form of data augmentation which involves altering the intensities of the RGB channels in the training galaxy images. Basically, PCA is performed on the set of RGB values of pixels. Multiples of the found principal components are added to each training image with their magnitudes in proportion to the corresponding eigenvalues multiplied by a random number drawn a Gaussian distribution with mean zero and standard deviation 0.1. Then the following quantity can be added to each RGB galaxy image pixel:

$$[p_1, p_2, p_3][a_1.l_1, a_2.l_2, a_3.l_3]^T$$

where p_i are eigenvectors, l_1 are eigenvalues and a_1 are the random variables. It turns out that that this scheme captures an important property of natural images that is the identity of the object is invariant to changes in the intensity and color of the illumination. As quoted by *Alex Krizhevsky* in his paper *ImageNet Classification with Deep Convolutional Neural Networks*:[13]

"This scheme reduces the top-1 error rate by over 1%"

Figure 5.1 shows how an image can be preprocessed to increase data-set size.

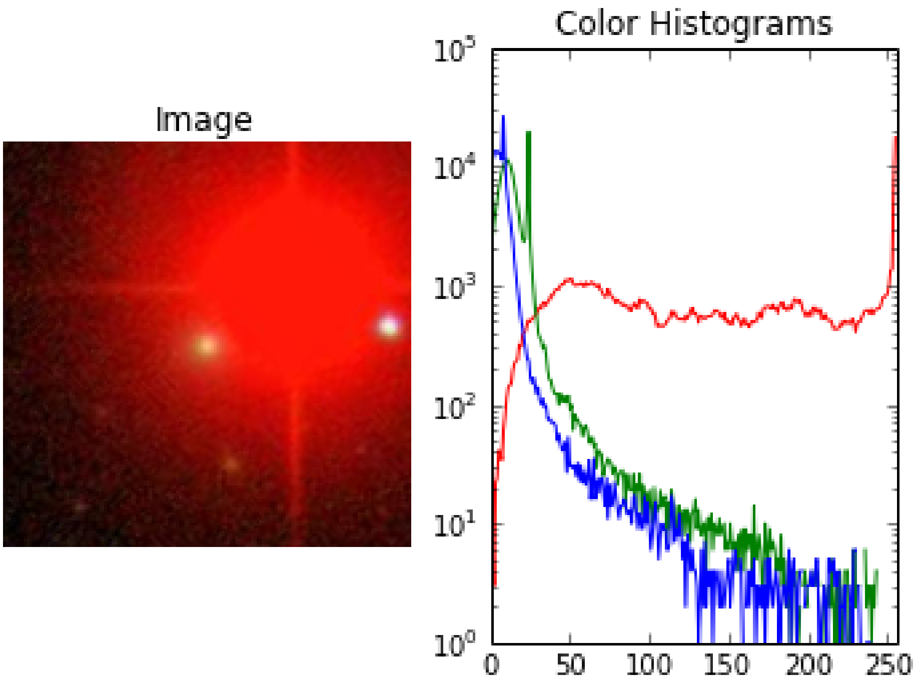

Figure 8: Example of Color Histogram for Galaxy image

5 Postprocessing

The 37 outputs to be predicted are weighted probabilities and they adhere to a number of constraints which we can exploit we get better results. One of the constraints was that the probabilities should sum up to one for the top class. And for each of the sub-classes the probabilities should sum up to the predicted probability of the super-class. So, if the output comes out to be a_1, a_2, a_3, a_4, the actual output can be given by:

$$a_i = a_i * P(super - class)/(a_1 + a_2 + a_3 + a_4)$$

We have seen that creating transformations on training image can help increase the training data size and that significantly improves performance. These transformations can also be used to improve the results further. We can create sufficient number of transformations for each image in the test set, predict the results for each each transformation of the image separately and then give these as input to a regressor which then produces the final output.

6 Data

The data was provided by Kaggle[4]. It consisted of more than $60,000$ training images and around $80,000$ test images. The images had size of 424×424 and were all RGB images of galaxies.

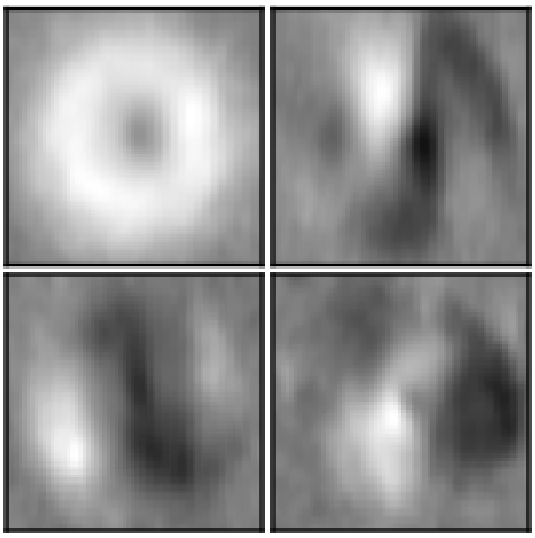

Figure 9: Eigenfaces learnt for “Is There a Spiral Pattern?”

7 Results and Conclusion

Through this challenge, we have realized that we can get good results with handcrafted features and it can get close to the results we obtain with Convolution Neural Nets. But it is impossible to beat a CNN with good architecture using handcrafted features in case of image data. This has been proved to be true in this contest. All the top three contestants have used Convolutional Neural Network. In the end how they actually implemented it and the architecture they used made the final difference.

For evaluating the results Root Mean Squared Error(RMSE) was used. RMSE was calculated for two vectors A and B by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(A_i - B_i)^2}{N}}$$

Our Central Pixel Approach got a *RMSE* value of 0.16 after training on the entire training data. This decent but not good enough.

Our second approach of training RF Regressor, when trained on one-half of the training data and tested on the other half, resulted on *RMSE* of 0.137. This clearly shows that this approach outperforms the Central Pixel based approach by a decent margin.

Our final approach of using hand-crafted features when trained on one-half of the training data and tested on the other half resulted in *RMSE* of 0.115. We can clearly see that it outperforms the other two by a decent margin. So, we can say for sure that out of the three approaches, this one performs the best. The final results have been shown in the table.

However, CNN with good architecture was able to achieve *RMSE* of around 0.8. But there was an associated memory cost (could be run only on 64 GB RAM systems) and took weeks on big computation servers.

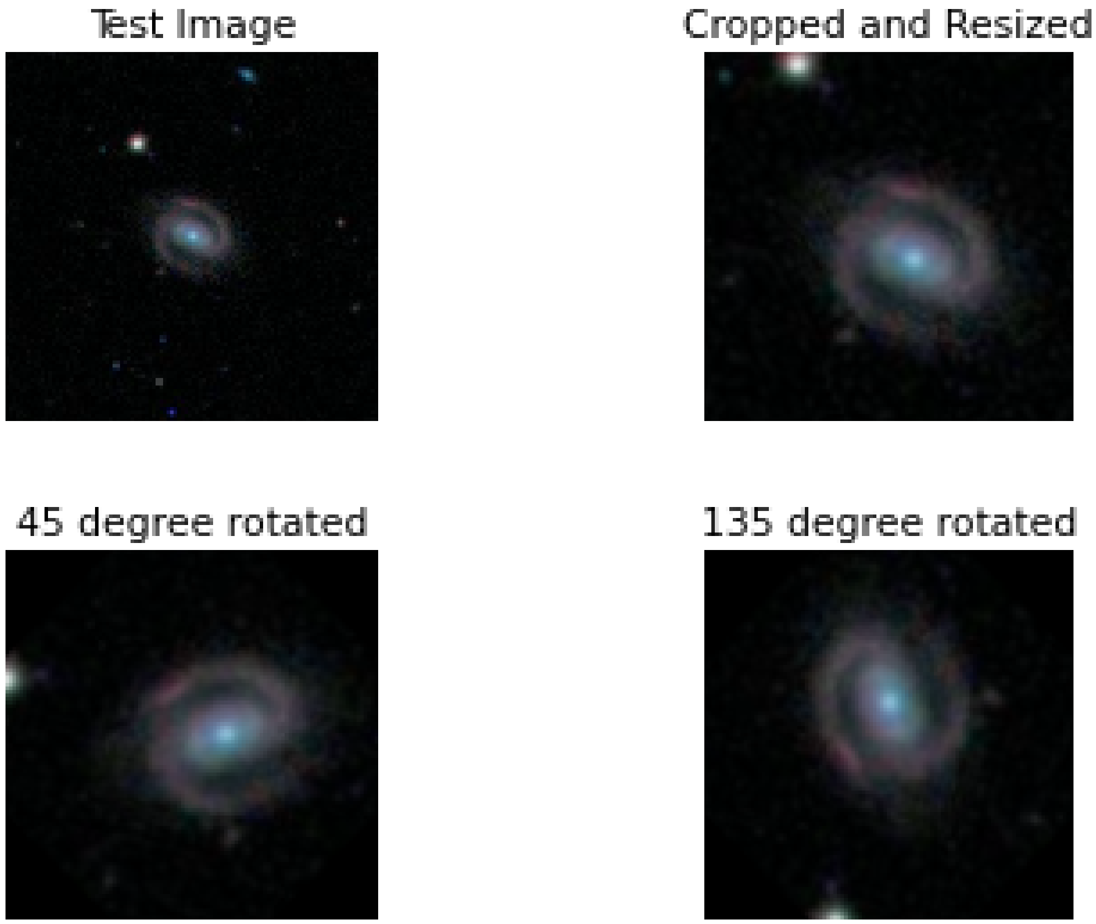

Figure 10: Rotate and Resize image

Method	RMSE
Central Pixel Method	0.1649218
Classifier Based Method	0.1373704
Feature Based Method	0.1177611
CNN (Kaggle Winner)	0.075

Table 1: Results for various methods

So, if we have to choose between the different available approaches, we have to first decide what is more important for us. If we don't care about cost of computation and have a good computation server, then CNN is the approach to go for. But for situation where cost of computation and time taken is also a factor, handcrafted feature-based approach is the approach to be taken.

8 Further Work and Improvements

Although for various reasons we have used handcrafted features as our main approach, using CNN is undoubtedly one of the best approaches available to us for now.

The winners of the contest have been generous enough to share their approaches and source-code, all of which were based on deep learning networks. But their approaches differed when it came to architecture and network.[6][14][15]

Although we can model this problem as a multi-regression problem, the probabilities of different classes aren't independent. This constraint also could be used to further improve the results.

References

[1] *Central Pixel Benchmark Code: Galaxy Zoo* *https://github.com/noahvanhoucke/GalaxyZooChallenge.*

[2] *Galaxy Zoo* *http://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge.*

[3] *Kaggle Galaxy Zoo Challenge* *http://www.galaxyzoo.org/.*

[4] *Kaggle Galaxy Zoo Challenge Data* *http://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/data/.*

[5] *SciKit-Image Documentation* *http://scikit-image.org/docs/dev/api/skimage.feature.html.*

[6] Sander Dieleman. *Galaxy Zoo Challenge :* 1^{st} *place.* https://github.com/benanne/kaggle-galaxies.

[7] Scikit documentation http://scikit-learn.org/stable/auto_examples/applications/face_recognition.html. *Face recognition using eigenfaces and SVM.*

[8] K. W. Willett et al. *Galaxy Zoo 2: detailed morphological classifications.* 2012.

[9] M. Banerji et al. *Galaxy Zoo: Reproducing Galaxy Morphologies Via Machine Learning.* 2009.

[10] M. Turk et al. *Face recognition using eigenfaces.*

[11] P. Mohanaiah et al. *Image Texture Feature Extraction using GLCM.* 2013.

[12] Mryka Hall-Beyer. *GLCM Tuturial* *http://www.fp.ucalgary.ca/mhallbey/tutorial.htm.* 2007.

[13] Alex Krizhevsky. *ImageNet ClassiïňĄcation with Deep Convolutional Neural Networks* *http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf.* 2012.

[14] Maxim Milakov. *Galaxy Zoo Challenge :* 2^{nd} *place.* https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/forums/t/7599/so-what-were-your-approaches/41505#post41505.

[15] Truyen T. *Galaxy Zoo Challenge :* 3^{rd} *place.* https://github.com/tund/kaggle-galaxy-zoo.