

---

# Computer Vision: Project Report

---

**Prithvi Krishna Gattamaneni**  
**Adithya Parthasarathy**  
**Vishal Motwani**  
pkg238@nyu.edu  
ap4608@nyu.edu  
vpm238@nyu.edu

## 1 Introduction

We attempted to work on the Galaxy zoo classification problem which was a Kaggle problem a few years back. The dataset was downloaded from Kaggle. The problem at a high level is a galaxy classification but the nature of the dataset makes it a regression problem. The dataset was created by having users answer a series of questions on a given image. The sequence of questions that a user answers for a given image is not fixed, and every question depends on the previous questions' answer, the first question being fixed. There are a total of 11 questions and each question has 3 to 7 responses depending on the question. For each image, we have the Conditional probability distributions for each class of questions. Thus we simply have a vector of 37 numbers for each image that we have to predict. The accuracy of the solution is measured by the root mean square error, and so this is the same metric that our model is optimised on.

## 2 Data Augmentation

Images of galaxies are rotation invariant: there is no up or down in space. They are also scale invariant and translation invariant to a limited extent. All of these invariances could be exploited to do data augmentation: creating new training data by perturbing the existing data points.

Each training example was perturbed before presenting it to the network by randomly scaling it, rotating it, translating it and optionally flipping it. We used the following parameter ranges: We implemented a series of data augmentation steps as mentioned here.

**Cropping, Zooming and downsampling** The data consisted of 424x424 colour JPEG images, along with 37 weighted probabilities that have to be predicted for each image.

For almost all of the images, the interesting part was in the center. The void of space around the galaxies was not very discriminative, so we randomly cropped all images to equal height and width between 207/1.3 and 1.3\*207 (log-uniform), then downsampled them 3x to 69x69, to keep the input size of the network manageable.

**Rotation** Images were rotated at random with angle between 0 and 360 degrees(uniform)

**Flip** Images were flipped with a probability of 0.5

**Translation** Images were translated randomly in the x and y direction, by an amount between -4 to 4 pixels at random (relative to original image)

### **3 Models Attempted**

#### **3 (a) Soumith model**

This section is to have a brief description of the model. If possible a picture of the layers would be nice.

Talk about data Normalisation. This is something that was being done in his code originally and we didn't change it i guess. He is normalising the data before the network trains it and is then unnormalising it after getting output from the test network i guess.

#### **3 (b) Blog model**

This section is to have a brief description of the model. If possible a picture of the layers would be nice.

### **4 Results**

This section is to have results that we got. Plots of convergence should be included.