Deblending Galaxies Using Generative Adversarial Networks

Authors:
Muhammad Usaid
Syed Ammar Ahmed
Hiba Jamal

Department of Computer Science
Habib University
Karachi, Gulistan-e-Jauhar, B-18

mu04032@st.habib.edu.pk
sa04050@st.habib.edu.pk
hj04033@st.habib.edu.pk

Abstract

Blended galaxies or galaxy superpositions are captured in many galaxy surveys. From a blended galaxy image, it is difficult to retain the original galaxies and their shapes. This is due to many factors, shape distortions because of light bending, and color distortions in image pixels intensities are to name a few. Hence, conventional segmentation techniques would fail in a scenario like this and result in missing information in the segmented galaxy images. To overcome this issue we deal with the problem at hand using generative adversarial networks [1]. The generative nature of the network will help in providing an image that overcomes the shape, size and color distortions that are lost if conventional object segmentation algorithms are applied.

1 Introduction

Large scale galaxy surveys produce huge amounts of astronomical data. Most of the time, astronomical problems are related to understand the shape of galaxies. They have to study the properties of isolated celestial bodies. Galaxies that are merged together are known as blended galaxies. Blended galaxies are a problem when it comes to studying isolated celestial objects. The image processing techniques to solve this problems have their limitations and large scale surveys produce contain very small areas of these blended galaxies. So it is not possible to devise any such image processing algorithm to solve this problem.

Current surveys such as DES (Dark Energy Survey) (Abbott et al. 2018), KiDs (Kilo-Degree Survey) (de Jong et al. 2013) and HSC (Hyper Suprime-Cam) (Miyazaki et al. 2018) as well as future surveys such as LSST (Large Synoptic Survey Telescope) (LSST Science Collaboration et al. 2009), Euclid (Amiaux et al. 2012) and WFIRST (Wide-Field Infrared Survey Telescope) (Gehrels et al. 2015) will generate immense quantities of imaging data within the next decade.

LSST, for example. Beginning full operation in 2023, LSST is expected to retrieve 15 PB of image data over its scheduled 10-year operation with a limiting magnitude of i 27. In the densest regions of the sky (100 galaxies per square arc-minute), up to half of all images are blended with center-to-center distance of 3 arc-seconds, with one-tenth of all galaxy images blended with center-to-center distance of 1 arc-second. Even considering the most typical regions of sky (37 galaxies per square arc-minute), simulation estimates suggest around 20 percent of galaxies will be superimposed at 3 arc-second separation and up to 5 percent at 1 arc-second separation. (Chang et al. 2013).

Here we made an algorithm to first blend and then de-blend galaxies. Then we made a deep neural network and train it on those specific images that we used. GANs (Generative Adversarial Networks) offers a solution for this. It is able to generate images by learning a probability distribution on random noise.

2 Methodology

2.1 Data

We used 3000 images out of the 141,553 images provided by the Galaxy Zoo via the Kaggle Galaxy Zoo classification challenge [2]. These images are from the 6^{th} data release of the Sloan Digital Sky Survey (SDSS). All the galaxy images from this dataset follow a specific standard, i.e. each image has dimensions $424 \times 424 \times 3$ and has a single galaxy positioned exactly on the center.

2.2 Image Pre-processing

The images from the Kaggle dataset will be provided as ground truth to the GAN. However, for our purposes we we need to make sure that once our network is trained on real images, and that it can in fact generate images of those real galaxies from the distorted segmented versions of those galaxies. To do this, we employed an algorithm that blends galaxy images in a fixed manner according to pixel intensities. Since we know that one galaxy is centered in an image, we can superimpose another one by positioning it 50 pixels right to the center of the x-axis, and using the highest pixel intensity as the blending mechanism. Shown in Figure 1.
At this point, around half of an original galaxy's shape and color has been distorted or lost in the

Figure 1: Blended image created using maximum pixel intensity.

blend. This is when we will segment the galaxies from the blended image using the inverse of the same approach we used to blend it. The result will be half a distorted galaxy image. Adding further to the distortion in the image, random noise is also added to create more obstacles for the GAN to successfully identify a segmented galaxy. Shown in Figure. 2.

2.3 Training the GAN

Now that we have a better idea of the sort of input and data that will be given to our network, it is important to understand the steps leading up to training process itself.

2.3.1 Architecture of GAN

There are two networks that are trained individually but who's results are codependent on each other while training. They are called the generator and discriminator. The generator learns to generate a fake image. It takes a random normal distribution as input, and transforms it to a fake image closer to the distribution it has to learn, this image is sent to the discriminator. The discriminator identifies

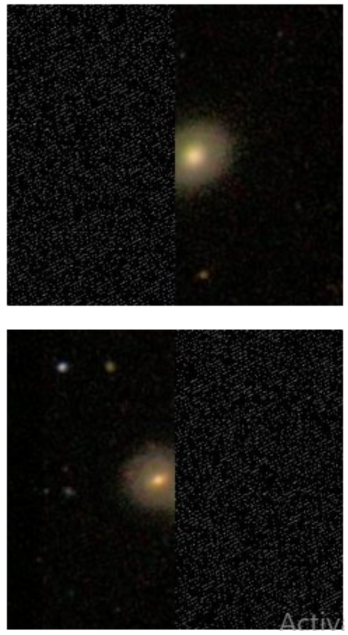

Figure 2: Segmented images attained with distortion and noise.

how close the fake image is to the ground truth. If the generated image is defined as implausible by the discriminator, then through backpropagation, the discriminator's loss signals the generator to update its weights accordingly.

2.3.2 Loss Function

The loss used in our implementation is the Binary Cross-Entropy Sigmoid Loss. It is a sigmoid activation plus a Cross-Entropy loss [cite]. Described as follows: here s_i and t_i represent the score and corresponding ground truth for every ith example respectively, while the f represents the sigmoid function:

$$= \frac{1}{N} \sum_{n=1}^{N} -t_i log(f(s_i)) - (1 - t_i) log(1 - f(s_i))$$

In the context of our galaxy images this is a good choice for a loss function because the use of Sigmoid function here instead of something like softmax loss ensures that the loss calculated for one output vector is independent of the others being calculated. As the word 'binary' suggests, it is easy to understand that a generated image will be classified as either real or fake.

Since the case here is that the GAN has two different networks, the generator and discriminator, the loss is applied differently to both. The aim of the discriminator is to minimize the loss, i.e. that it cannot see the difference between the real and generated image, whereas the aim of the generator is to maximize the loss, i.e. that is putting in more and more detail to create a fake image as its parameters are updated.

2.3.3 Training

There were 3 hyperparamteres that we had to experiment with while training.

1. Batch size
2. Learning rate
3. Number of epochs

Figures 3, 4 and 5, show some of the experiments that were conducted amongst many. But a trend that can be observed is that the generator's loss increases as number of epochs increase and the batch size is reduced. Keeping these observations in hindsight, we decided the hyperparameters to train

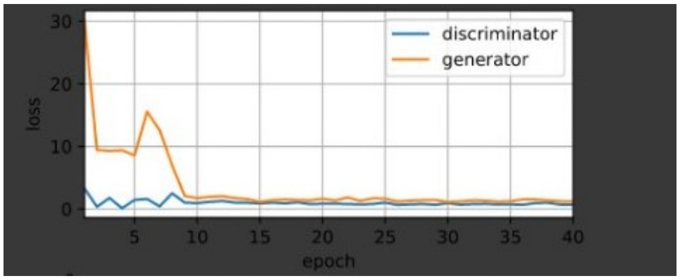

Figure 3: Loss plotted against number of epochs with learning rate as 0.003, 40 epochs and batch size 100.

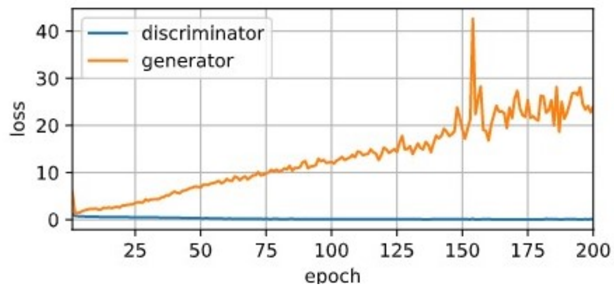

Figure 4: Loss plotted against number of epochs with learning rate as 0.005, 200 epochs and batch size 10.

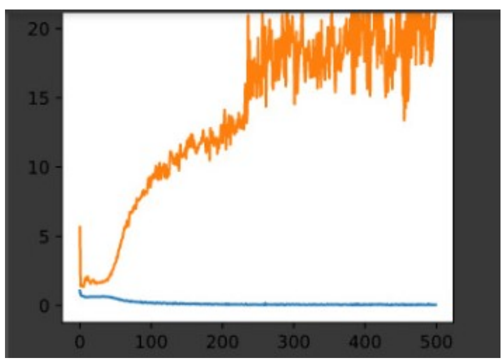

Figure 5: Loss plotted against number of epochs with learning rate as 0.003, 500 epochs and batch size 10.

our model for the last time to accept its results. We had to do so keeping in mind GPU usage and time limitations. The hyperparameters we decided upon were as follows: learning rate as 0.006, batch size as 10, and number of epochs as 1000. The training took around 7 hours to complete, and resulted in the generator's loss being 28.872.

3 Results

To get a sense of evaluating our model, we compare images generated by the model during training and the results it produces on test data. This is very important to note that the images that comprise the test data have not been used for training purposes which means the model has not seen these underlying distribution of galaxies previously.

Figure 6: Images generated during training for 1000 images and 500 epochs.

We then train the model with updated hyper-parameters and increased number of images in training dataset.

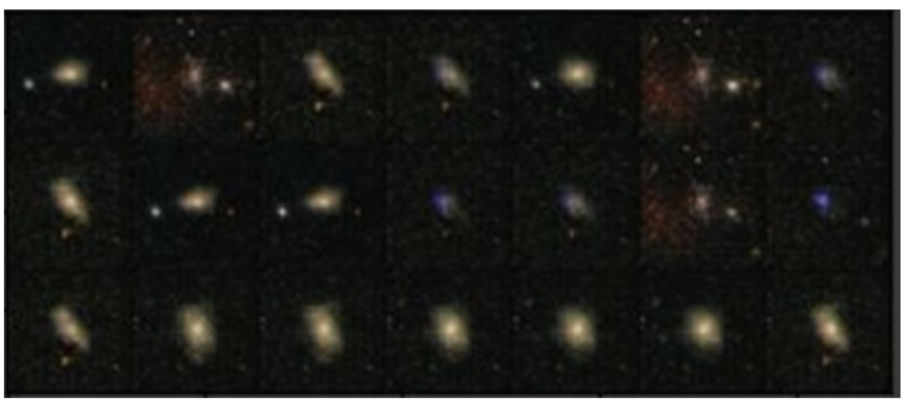

Figure 7: Images generated during training for 3000 images and 1000 epochs.

We now test our model on random noise. We provide it with samples from normal distribution with mean and standard deviation of 0 and 1, respectively on random and observe the results.

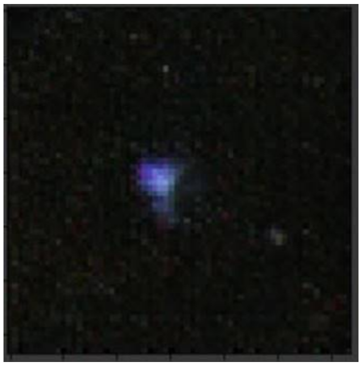

Figure 8:

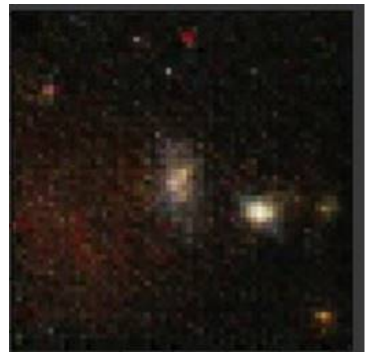

Figure 9:

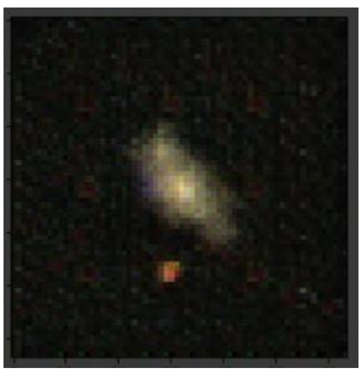

Figure 10:

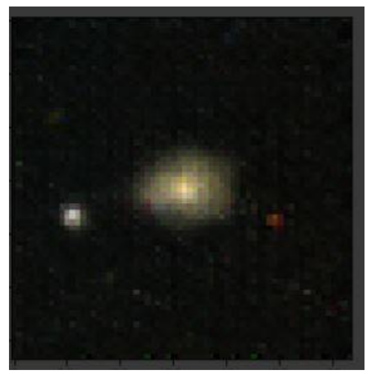

Figure 11:

We now give our model segmented images on which it eventually has to work on.

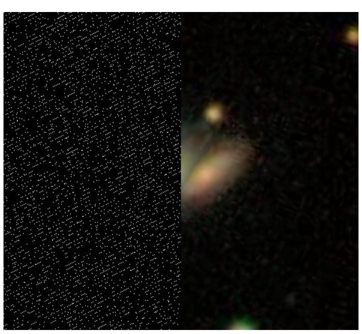

Figure 12:

Figure 13:

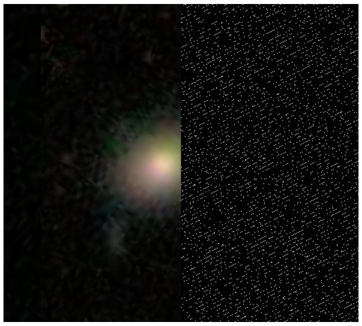

Figure 14:

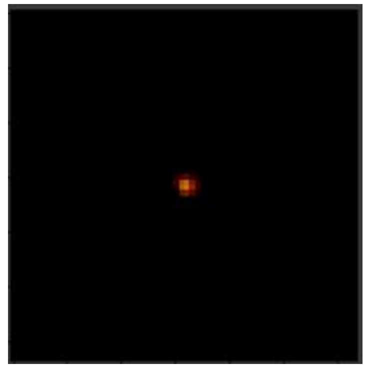

Figure 15:

Note that Figure 12 and Figure 14 are inputs to the model and Figure 13 and Figure 15 are their respective outputs. We will analyze these results in the section below.

4 Discussion

We see that for fewer number of examples in dataset and a lower number of epochs, the images produced by the model (Figure 6) lack the precision of the distribution of galaxy images which gets quite better with more number of examples and epochs as can be seen in Figure 7.

We see that when the model is given random normal noise as input, it gives a very reasonable distribution for a galaxy. We can see through Figure 8 that the model is not a trained in a particular manner. We say this because of the color distribution of this image. Most of the images in the dataset show color distribution very similar to that of Figure 10 and Figure 11, yet the model was able to learn this distribution and mapped the random normal noise to it. Similarly, for Figure 9 we see that there is a cloud of red pixels from left to center. This is because of presence of two such images in the training dataset. The model tried to preserve a feature of an image which was very unique throughout

Figure 16: Original image from dataset

the dataset. Similarly, for Figure 11 we can see that the model not only generated the galaxy but also placed other non-focused objects such as stars. However, the model cannot be praised similarly for Figure 10. Such is so because of the shape of the galaxy it generated. This shape is not a shape of a galaxy but a mixture of two different kinds of galaxies which are as follows (Figure 17 and Figure 18).

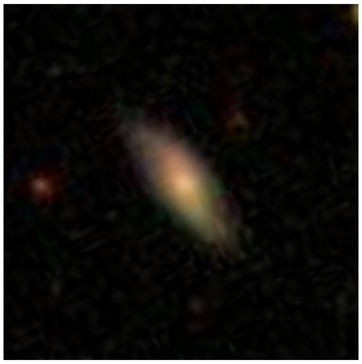

Figure 17:

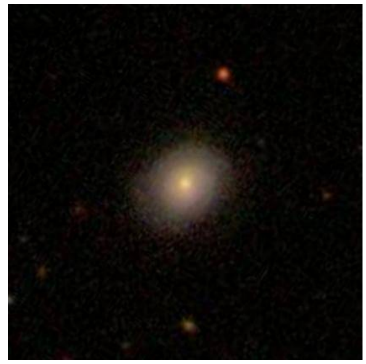

Figure 18:

This shows that the model has not yet clearly learned the difference between these two different distribution of pixels and generated a mixture of these two.

We saw how we encountered our original problem of deblending by first segmenting a blended image into two separate images and then passing them on to our model which will in result completing the missing pixel values from these images. This can be done because of the very nature of GANs as generative adversarial networks are natural solutions for deblending galaxy images because of their ability to fill missing pixel values due to the blend using information learned about the distribution of original images which are provided during training. This pipeline can be depicted as follows.

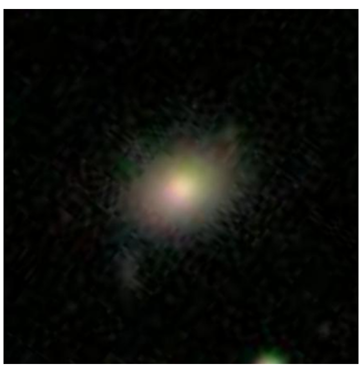

Figure 19: Preblended 1

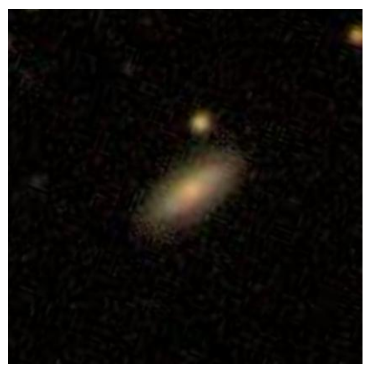

Figure 20: Preblended 2

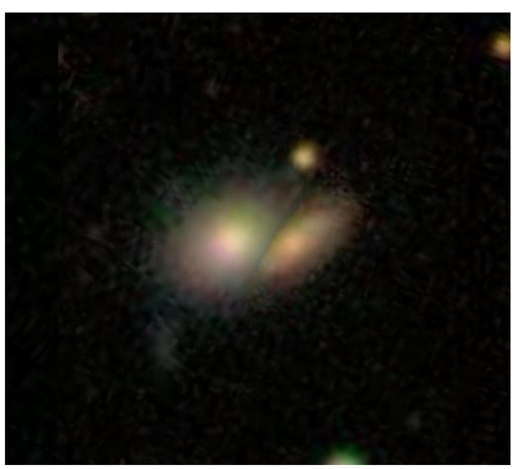

Figure 21: Blended image of Preblended 1 and Preblended 2

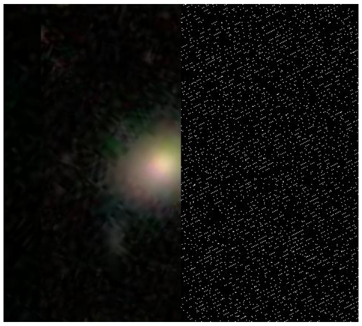

Figure 22: Segmented image of blended 1

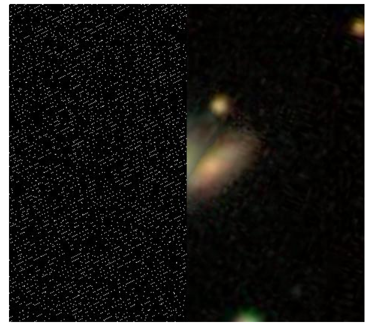

Figure 23: Segmented image of blended 2

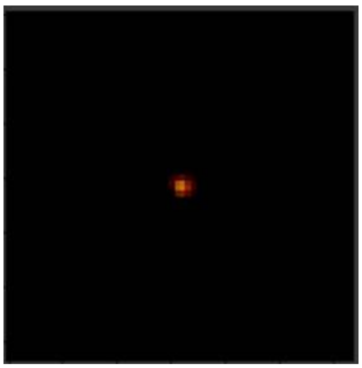

Figure 24: Deblended 1

Figure 25: Deblended 2

For segmented and incomplete galaxy images the model is not showing very promising results. However, we can see that for Figure 22 and Figure 23 it preserved their original shapes. This shows the ability of the model to find and learn such delicate features. The reason for these results on both random normal noise and segmented images is due to insufficient training of model. The model was trained on 3000 images and it required considerable amount of time to do so. But because of the nature of galaxies which have no constant shape, color and pixel distribution, this was not enough to train the model to produce very accurate results yet the model shows great tendency to produce promising results.

5 Future Directions

More work in this field is required. Our current project has some limitations but it can be extended to more general scenarios doing more research on it. Following are the ways to extend this project:-

1. We need to find an algorithm of image segmentation to separate blended galaxies by creating a bounding box on it. The perfect algorithm for this doesn't exist so we will have to train a model ourselves.
2. The limitation of our current model is that it is trained on specific kind of blending. we need to extend it to more generalized blending by experimenting different techniques.
3. We are not considering other than spherical galaxies. The blending of spiral galaxies is much more complicated than spherical galaxies. In the former case we need to take care of specific laws of physics that govern the gravitational interaction of such celestial bodies.
4. There is a possibility of more than two galaxies merging. We need a more efficient and perfect way to deal with such situations. Currently, we only cared about two galaxies but it can be extended to more then one.
5. Our current model is trained on image files. The data generated by large scale surveys is mostly in fits files. We need to work directly on fits file to make our work more efficient.

References

[1] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., and Bengio, Y, "Generative Adversarial Nets.", NIPS.

[2] Galaxy Zoo 1: data release of morphological classifications for nearly 900000 galaxies. 2017. Retrieved from: *https://academic.oup.com/mnras/article/410/1/166/1032478* [3] Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names. (2019). Retrieved 9 December 2019, from https://gombru.github.io/2018/05/23/cross_entropy_loss/

[4] Chang C., et al., 2013, MNRAS, 434, 2121

[5] Abbott T. M. C., et al., 2018, preprint, (arXiv:1801.03181)

[6] de Jong J. T. A., Verdoes Kleijn G. A., Kuijken K. H., Valentijn E. A., 2013, Experimental Astronomy, 35, 25

[7] Miyazaki S., et al., 2018, Publications of the Astronomical Society of Japan, 70

[8] LSST Science Collaboration et al., 2009, preprint, (arXiv:0912.0201)

[9] Amiaux J., et al., 2012, in Space Telescopes and Instrumenta- tion 2012: Optical, Infrared, and Millimeter Wave. p. 84420Z (arXiv:1209.2228), doi:10.1117/12.926513

[10] Gehrels N., Spergel D., SDT W., et al., 2015, in Journal of Physics: Conference Series. p. 012007

[11] Reiman, D., Göhre, B. (2019). Deblending galaxy superpositions with branched generative adversarial networks. Monthly Notices Of The Royal Astronomical Society. doi: 10.1093/mnras/stz575