Probabilistic Graphical Models

Project 1 - Markov Random Fields

Mohammadjavad Matinkia - 96131043

1 Preliminaries

In this project we intend to use *Markov Random Fields* or *MRF*s to segment an image. In the first part of the project we use a synthetic grayscale image for a *Naive Bayes* classifier and investigate the effect of noise on the performance of the Naive Bayes classifier on segmentation of the image. We then use a MRF model to segment the very image, and we compare the results of a MRF model and a Naive Bayes model on segmentation. In the second part we use the trained MRF model to segment a real image into three parts: road, sky, and the road side. At last we build a new MRF model which utilizes more features to segment the full-color image. In each section, for better understanding the performance of the models, we create an image using the predicted labels by the models.

2 Part 1

In this part, our target image is a synthetic image illusterated in Figure(1).

2.1 Question 1

In this part we need to add a Gaussian noise to the image and use a Naive Bayes classifier to segment the image. Note that the only feature considered for segmentation of the image, is the intensity of each pixel, that is we only need to calculate the values of $p(intensity|label)$ for different labels. Here we add a noise with a variance equal to 1000. In Figure(2), you can see the original image on the left pane, the noisy image in the middle and the segmented image on the right.
As you can see, the Gaussian Naive Bayes classifier has a promising results for segmenting a noisy image. To use the Naive Bayes classifier we use the *GaussianNB* in *scikit-learn* library. The procedure of segmentation is as follows: first using our prior knowledge, we create a label matrix where each element represents the label of the corresponding image. Here we assume that the intensity 0 has the label 0, intensity 127, has the label 1, and the intensity 255, has the label 2. Then we contaminate the original picture using this noise, and then we pass the noisy image to a Naive Bayes classifier to segment the image.

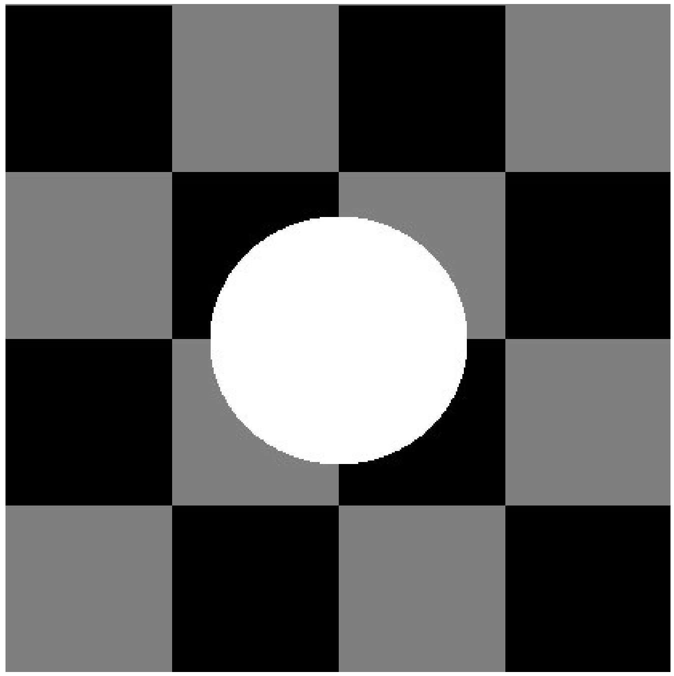

Figure 1: The target synthetic image

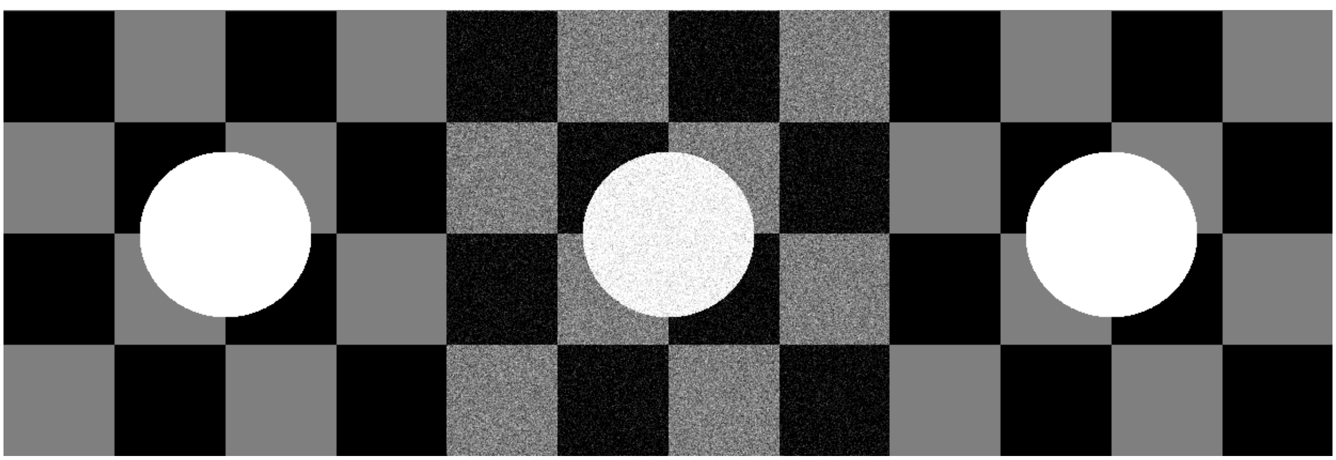

Figure 2: The original image - Noisy image - Reconstructed image

2.2 Question 2

Now we use different values of σ for Gaussian noise to see the effect of the noise power on the performance of the Naive Bayes classifier. We choose seven different values of σ^2: 1, 10, 100, 1000, 10000, 100000, 1000000. The procedure of training and using the Naive Bayes classifier is same as the previous question. In Figure(2) to Figure(8) you can see the results of the Naive Bayes classifier on segmentation.

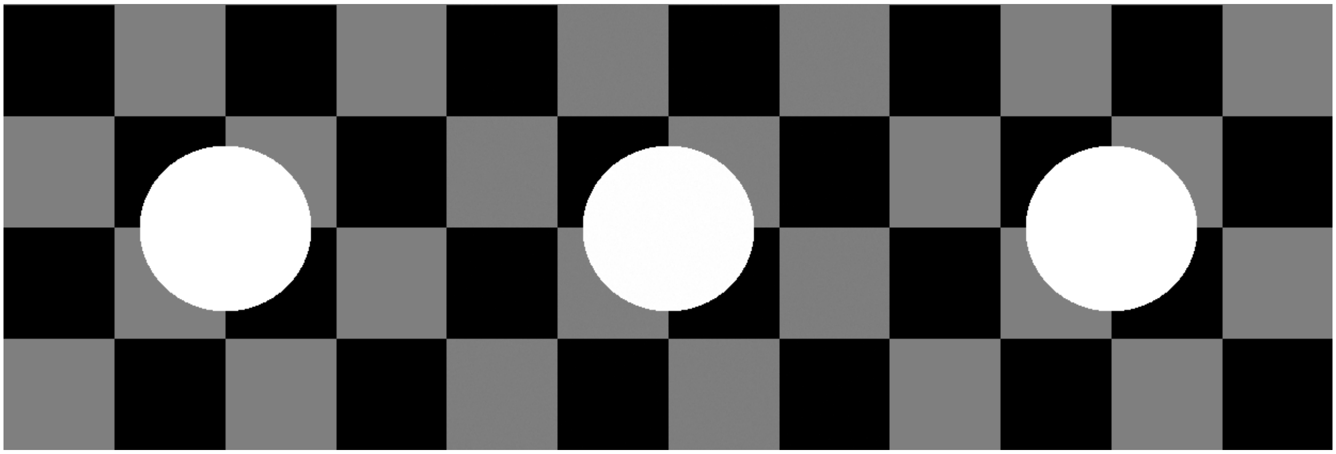

Figure 3: $\sigma^2 = 1$:The original image - Noisy image - Reconstructed image

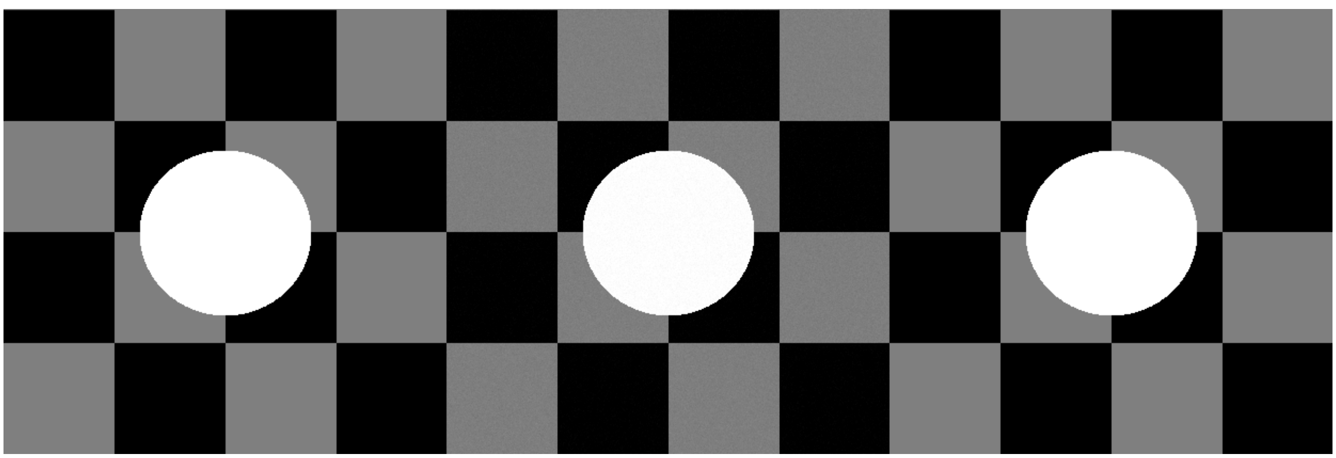

Figure 4: $\sigma^2 = 10$:The original image - Noisy image - Reconstructed image

As you can see for different values of σ^2, the Naive Bayes classifier still has a promising performance on the segmentation of the image, although it can not segment the *white* part correctly for variance values larger than 1000. This fact implies that knowing only the intensities of the image pixels gives us enough information about different segments of the image, and also assuming that knowing the label of the pixels, makes their intensities independent is a plausible presumption.

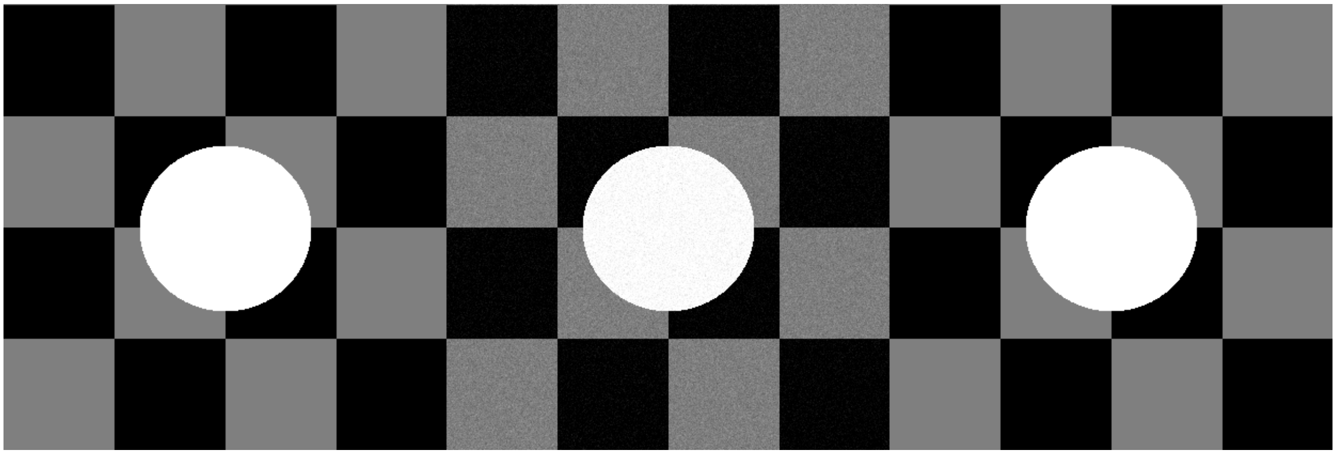

Figure 5: $\sigma^2 = 100$:The original image - Noisy image - Reconstructed image

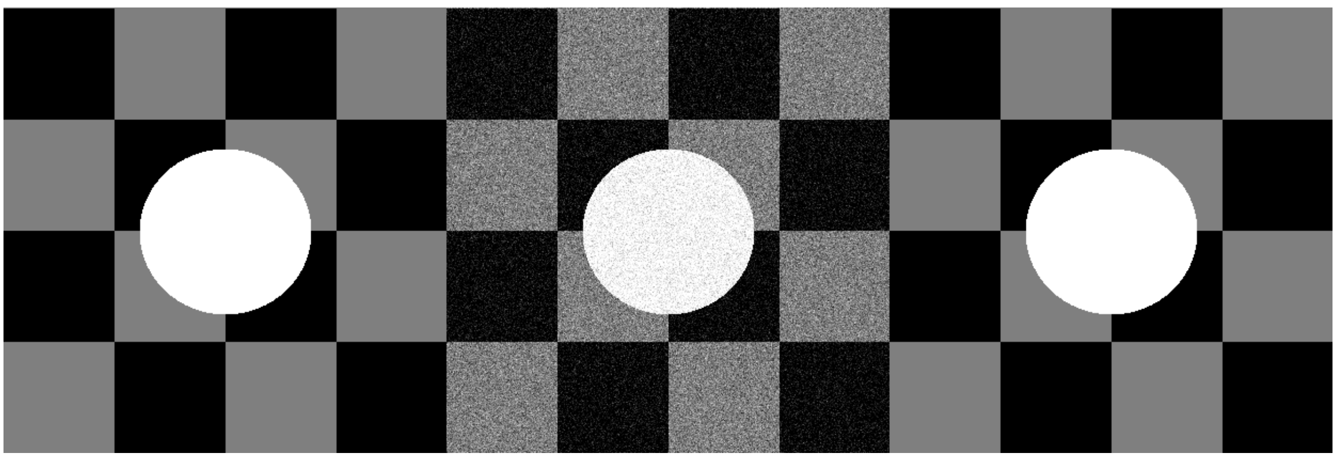

Figure 6: $\sigma^2 = 1000$:The original image - Noisy image - Reconstructed image

2.3 Question 3

We can use a MRF model instead of Naive Bayes model to segment the image of Figure(1). To do so we need to design a *Markov Network* which models our desired features, which are in our case, the intensities of the pixels. We use the term *Markov Random Field* because we need to extract the information from a feild of neighboring pixels. Initially we assume that each pixel has a 4-connect neighborhood relationship with its surronding pixels(see Figure(9)). If we assume such a structure we can utilize two kinds of factors for the Markov network: single factors which are set to log pdf of the pixels and binary factors which are set according to the similarity of labels between two neighboring pixels.

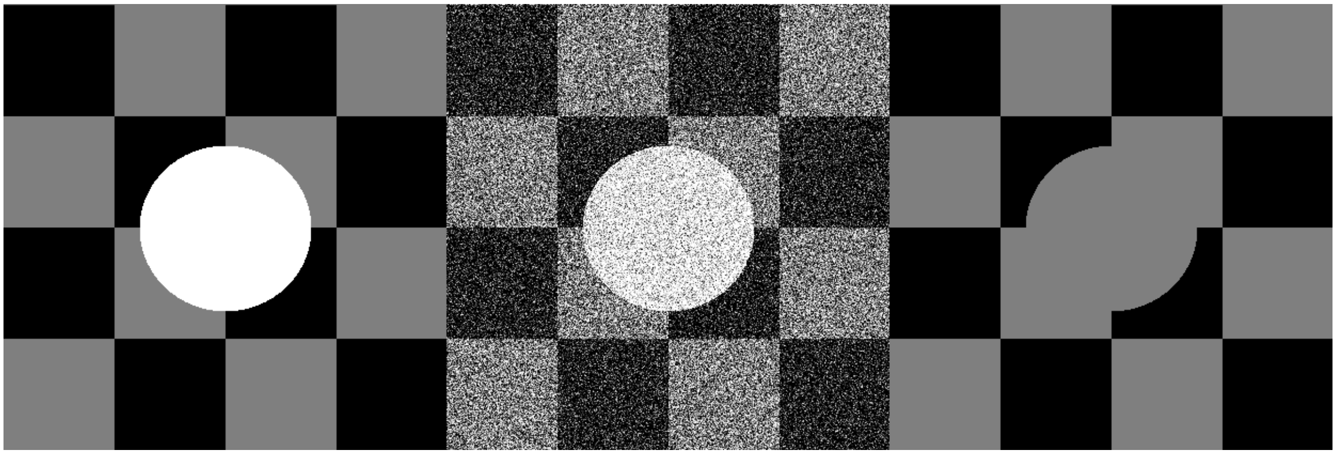

Figure 7: $\sigma^2 = 10000$:The original image - Noisy image - Reconstructed image

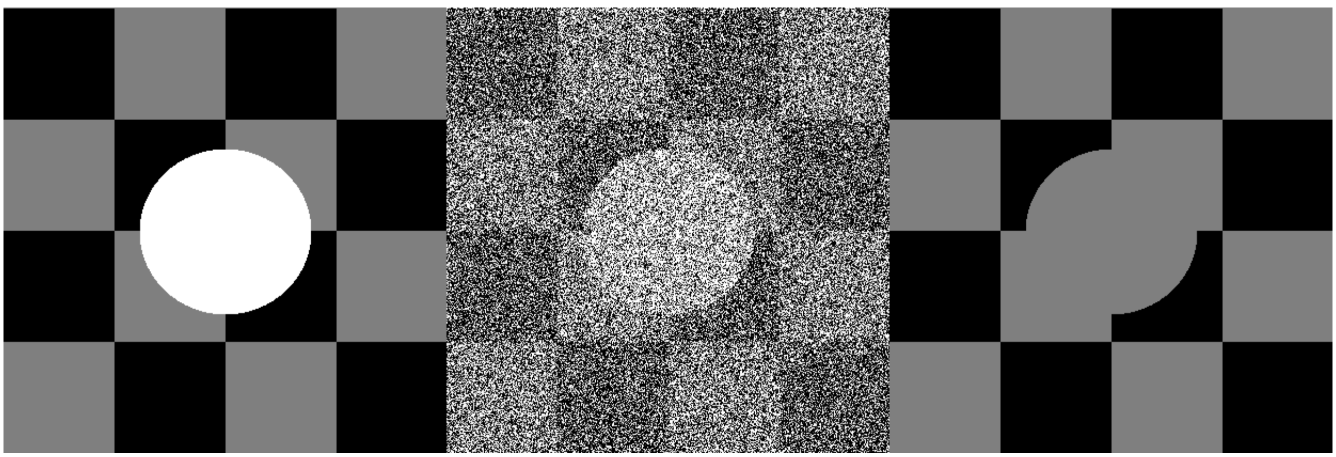

Figure 8: $\sigma^2 = 100000$:The original image - Noisy image - Reconstructed image

So we can define an energy function for the model as Eq.(1). By minimizing this energy function we reach a distribution of the labels which is the most probable distribution.

$$U(\omega) = \sum_s (\log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(i_s - \mu_{\omega_s})^2}{2\sigma^2_{\omega_s}}) + \sum_{s,r} \beta\delta(\omega_s, \omega_r) \tag{1}$$

where, the summation over s is over all the pixels of the image, and the summation over s, r is over all the pixels s and their neighbors r. i_s corresponds to the intensity of the pixel, σ_{ω_s}, and μ_{ω_s} correspond to the variance and the mean of labels which their values are ω_s. Eq.(1) is a non-convex function so we can

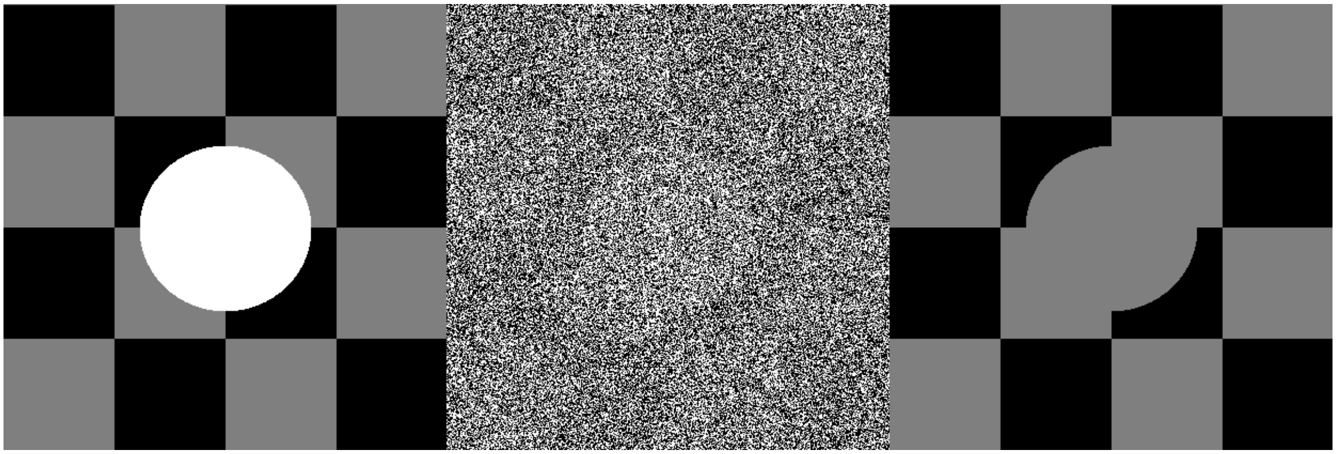

Figure 9: $\sigma^2 = 1000000$:The original image - Noisy image - Reconstructed image

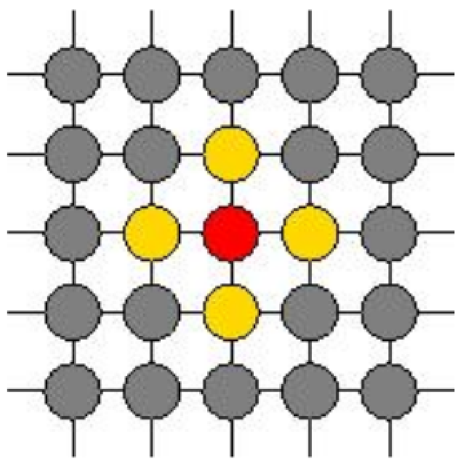

Figure 10: Field of pixels and definition of the 4-connect neighborhood

not use conventional optimization methods to find its minimum. Instead we use *Simulated Annealing* to optimize thid objective function. In the next question we use the this model to segment the noisy image in Question 2.

2.4 Question 4

For training the MRF model decribed above we use a noisy model which is contaminated with a Gaussian noise with variance 10000. Note that in this value if variance, Naive Bayes model failed to distinguish the *white* part of the image, and here we want to check wheather MRF model can segment the image more efficiently. To optimize the energy function using Simulated Annealing, we set the initial temperature of the SA algorithm to 25000, and its minimum value to 0.1. We run the algorithm for 4×10^6 iterations. The result of the algorithm is shown in Figure(11). We also use the variance 1000 to test model and compare it to Naive Bayes model. This result is given in Figure(12).

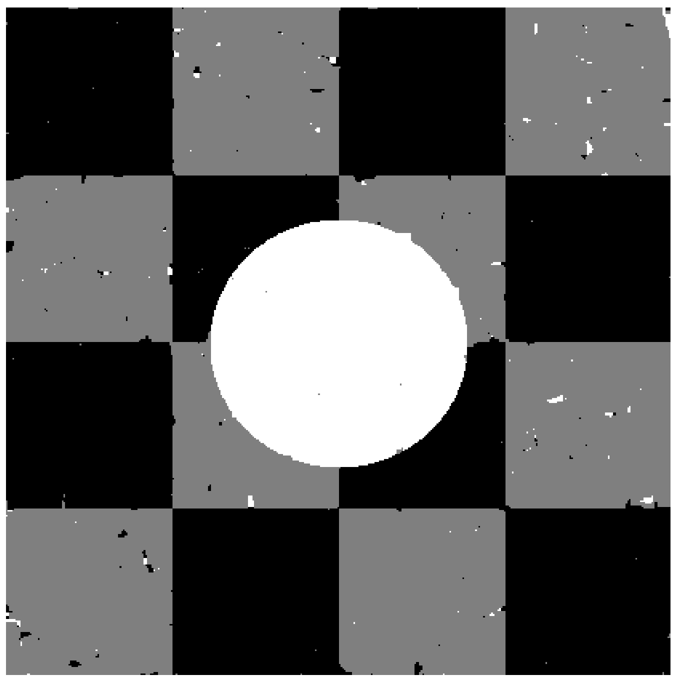

Figure 11: Result of MRF segmentation for a noisy image with $\sigma^2 = 10000$

As you can see the MRF model outperforms the Naive Bayes model in a noisy environment and it can make a precise segmentation of the image. The reason behind this behavior is in fact using the mutual information between neighboring pixels which apears as *doubleton* factors of the Markov network.

2.5 Question 5

The neighborhood architecture considered in last parts was a 4-connect neighborhood. However we can consider a 8-connect neighborhood in which a pixel has four additional neighbors which lie in its diagonal directions. We can simply modify our former MRF model to consider 8-connect neighborhood of a pixel. The result of a 8-connected neighborhood MRF model is given in Figure(13).

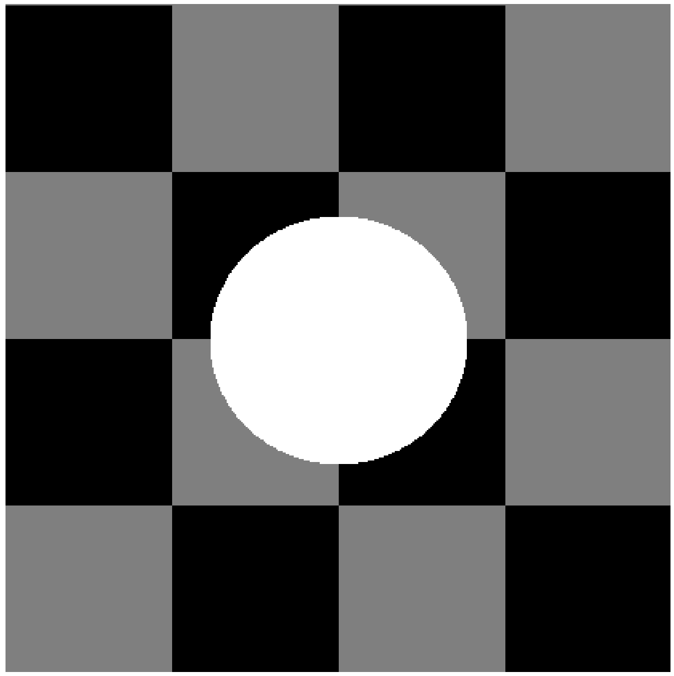

Figure 12: Result of MRF segmentation for a noisy image with $\sigma^2 = 1000$

Although the result of Figure(13) seems nonpromising,the most important benefit of using 8-connected neighborhood is that in same number of iteration we can reach a much more less value of energy function. For instance for 4-connected neighborhood the final energy of the model is in the order of 7×10^5 but the in 8-connected neighborhood case the final energy is in the order of 1×10^5, which means a better segmentation of the image. In Figure(14) you can see the result 8-connected neighborhood MRF model on a noisy image with variance 1000.

2.6 Question 6

In all the previous parts the predefined value for parameter β was 1. Here we need to check that how changing this parameter affects the performance of the model. To do so, we set the β equal to 2,3, and 4, and report the results of segmentation. The results are given in Figure(15) to Figure(17).
Parameter β determines the importance of similarity of neighboring pixels' labels, and based on the results, obtained above , we can see that increasing the value of β can make the model better at first but then make it worse for large values of β . The best reported value for beta is 1, based on the results obtained.

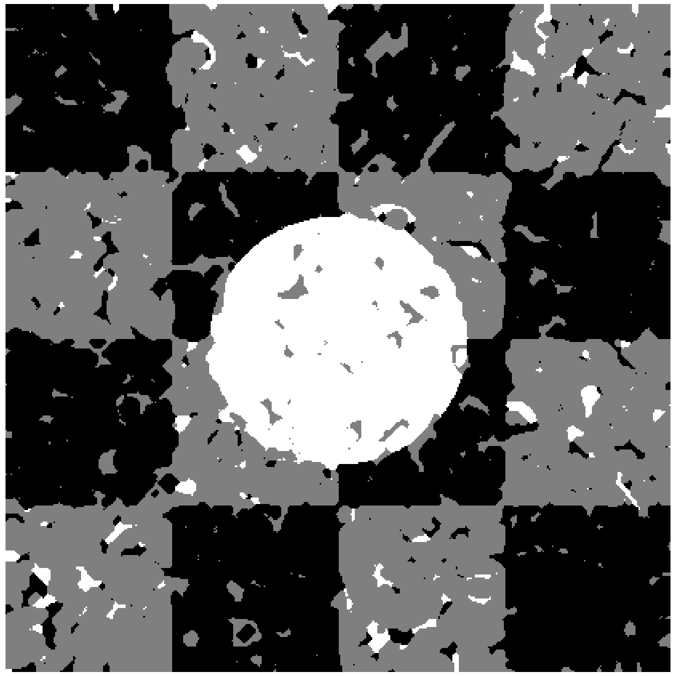

Figure 13: Result of 8-connected neighborhood MRF segmentation for a noisy image with $\sigma^2 = 10000$

2.7 Question 7

In all the previous sections we started the optimization algorithm with a random assignment to labels. However another method is to use a Naive Bayes model to extract an approximate segmentation of the image and use this segmentation as the initial state of the MRF model. This method has this benefit that the MRF model starts from a state which is very nearer to the optimal state, so the required amount of iteration decreases significantly and the quality of the final segmentation improves. But there might be cases in which the MRF model does not improve the quality of the Naive Bayes classifier. In such cases the Naive Bayes classifier obtaines a state which is near a local minimum, and the MRF model get stuch in this local minimum and cannot escape from it. For example in Figure(18) you can see that there are no tangible improvement with respect to Naive Bayes model. In Figure(18) you can see the result of the MRF model initialized with the result of a Naive Bayes model. This result is obtained after 500000 iteration.

2.8 Question 8

In *Simulated Annealing*, the initial amount of temperature must be high enough to allow the model to jump to different state, despite they might be worse than current state. This makes the model to run away from local minima of the objective function. However the temperature must fall down quickly and

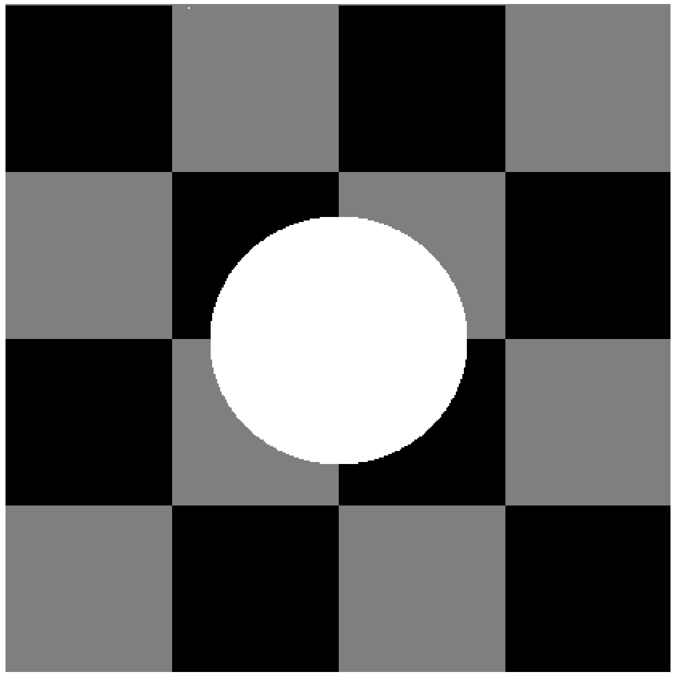

Figure 14: Result of 8-connected neighborhood MRF segmentation for a noisy image with $\sigma^2 = 1000$

freeze the model to assure that model will move to the correct direction towards the minimum of the objective function. Here in this project we set the initial value of the temperature to 25000, and the minimum value to 0.1. We also update the temperature according to Eq.(2)

$$T(t) = \frac{T(t-1)}{\ln(t)} \tag{2}$$

where t represents the iteration.

3 Part 2

In this part, we first intend to test the MRF model built in the last part on another image which is given in Figure(19), and then we are going to incorporate more features into MRF model to check the performance of model in presence of the new features. These new features are the RGB values of the color picture.
As you can see in Figure(19), this image can be segmented generally into three different sections: road, road side, and the sky. Our previous MRF model, also could segment an image into three different segments. So it might be tempting to use this model to segment the image of Figure(19).

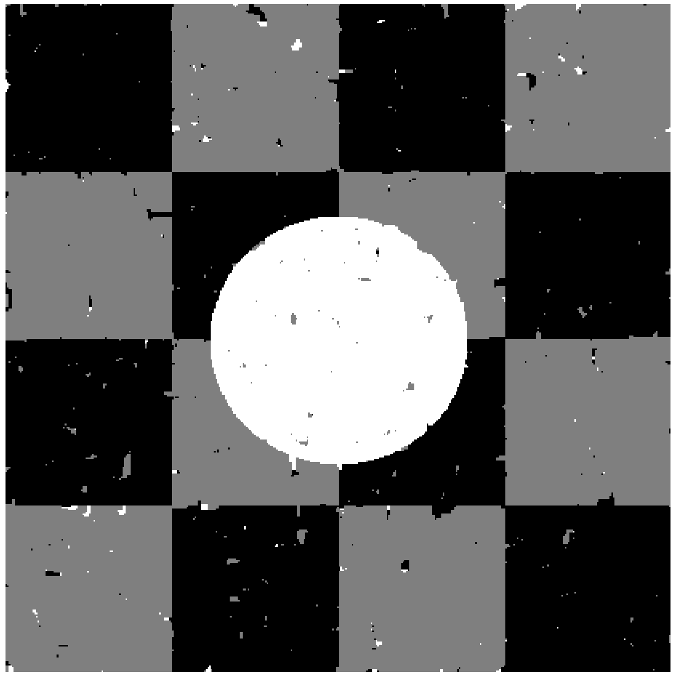

Figure 15: The result of segmentation with $\beta = 2$

3.1 Question 1

We use the MRF model we built in the previous section to segment the grayscale version of Figure(19). There are no other informative points here to mention. The result of segmentation is presented in Figure(20).
As you can see, when we use only the grayscale intensity of the pixels, the model is not strong enough to segment more complex images. This is a hint for us to incorporate more features into model. The result of segmentation on the HSV version of image of Figure(19) is presented in Figure(21).
Still we have non-satisfying results when we use the *Hue* feature of the HSV image. It is still because we only exploit one feature of the image.

3.2 Question 2

In Question 1, we saw that our MRF model can not segment the image properly, because exploiting only one feature does not give enough segmentation power to the model. Now we incorporate the information contained in the other channels of the image, that is the information contained in RGB values of the pixels.
In this case, the only modification made on the energy function is that we have to incorporate the log pdf of intensities in *Red* channel, *Green* channel, and *Blue* channel. So we have aggregate number of

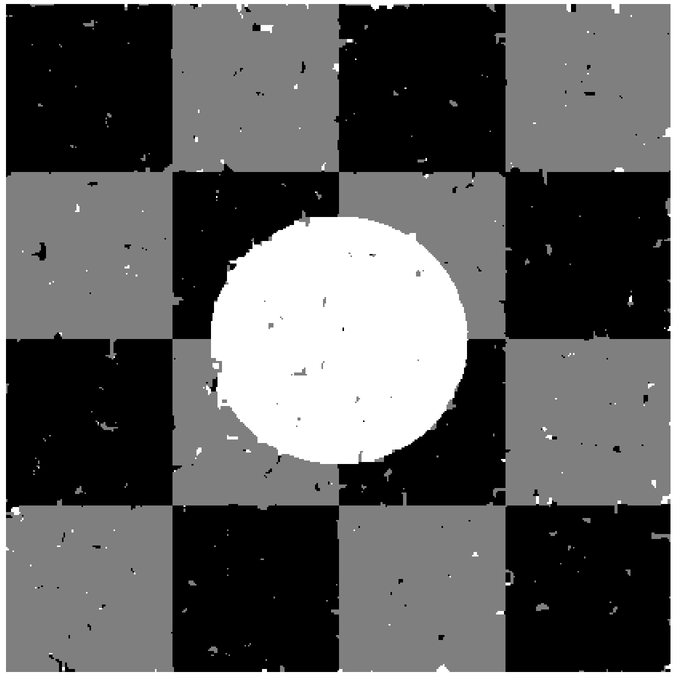

Figure 16: The result of segmentation with $\beta = 3$

four features. Using these four features in MRF model we can have a more satisfying segmentation of Figure(19) which is given in Figure(22). In is worth to note that in order to train the new modified MRF model, we used a synthetic image, which is given in Figure(23). By presenting this synthetic image to the MRF model, the network can extract the information about different colors, and combine it with its former knowledge in grayscale space to provide a more precise segmentation result.

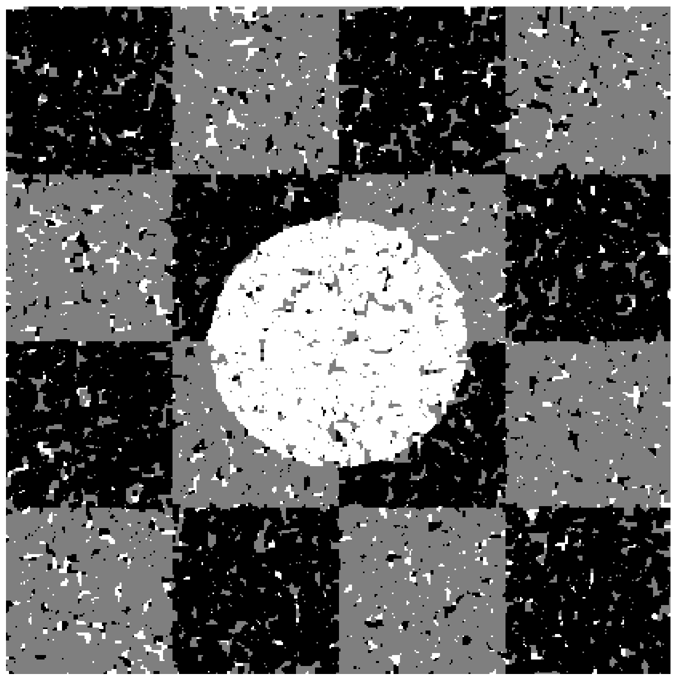

Figure 17: The result of segmentation with $\beta = 4$

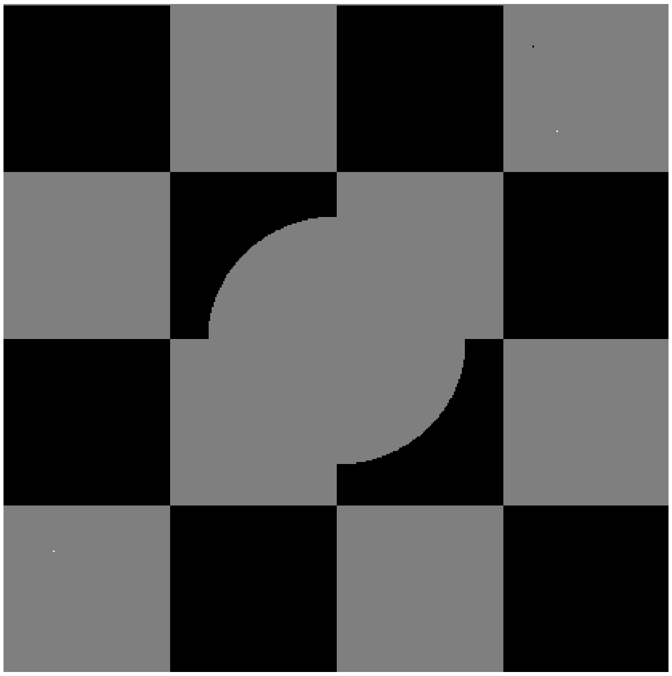

Figure 18: The result of MRF model initialied with Naive Bayes

Figure 19: The original color image

Figure 20: The result of segmentation on grayscale image

Figure 21: The result of segmentation on HSV image

Figure 22: The result of segmentation on using four features

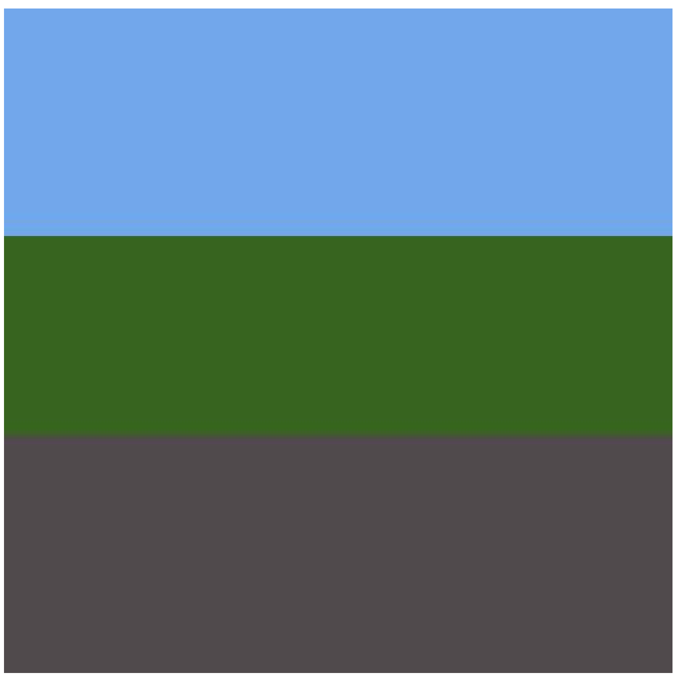

Figure 23: The synthetic image to train the new modified MRF model