CS 736 Assignment 1

Adish Shah, Prathamesh Sachin Pilkhane, Padakanti Akshay Kiran

Jan 31st, 2022

Contents

1 Problem 1 **2**
1.1 Optimised RRMSE values 2
1.1.1 Quadratic Prior 2
1.1.2 Huber Prior 2
1.1.3 Discontinuity Adaptive Prior 2
1.2 Images 2
1.3 Objective function plots 4

2 Problem 2 **6**
2.1 Optimised RRMSE values 6
2.1.1 Quadratic Prior 6
2.1.2 Huber Prior 6
2.1.3 Discontinuity Adaptive Prior 6
2.2 Images 6
2.3 Objective function plots 8

3 Problem 3 **9**
3.1 Proposed MRF Prior: 9
3.2 Noise Model 10
3.3 Bayesian Denoising Formulation 10

The algorithm for the first two problems is in a single file `ImageDenoising.py`. The function for finding optimal values of alpha and gamma is in `optimise.py`. The main execution for getting the results is in `q1.py` and `q2.py` for Problem 1 and 2 respectively. The noise model used is i.i.d Gaussian for both Problem 1 and 2.

1 Problem 1

The initial RRMSE between the Noiseless image and Noisy image is 0.29857915712437444

1.1 Optimised RRMSE values

1.1.1 Quadratic Prior

a* (optimal value of alpha) = 0.102
RRMSE at a* 0.2812164997489185
RRMSE at 0.8a* 0.2816553431662804
RRMSE at 1.2a* 0.2816231520729511

1.1.2 Huber Prior

a*(optimal value of alpha) = 0.887
b*(optimal value of gamma) = 0.009
RRMSE(a*,b*) = 0.23693671217697682
RRMSE(1.2a*,b*) = 0.35807784791948755
RRMSE(0.8a*,b*) = 0.25576678170981265
RRMSE(a*,1.2b*) = 0.2376064461451403
RRMSE(a*,0.8b*) = 0.23735784219937606

1.1.3 Discontinuity Adaptive Prior

a*(optimal value of alpha) = 0.9
b*(optimal value of gamma) = 0.008
RRMSE(a*,b*) = 0.23974814986765483
RRMSE(1.2a*,b*) = 0.3239945185731995
RRMSE(0.8a*,b*) = 0.26215999590483124
RRMSE(a*,1.2b*) = 0.23990924168646097
RRMSE(a*,0.8b*) = 0.24060204916476619

1.2 Images

The 5 images obtained are as follows :

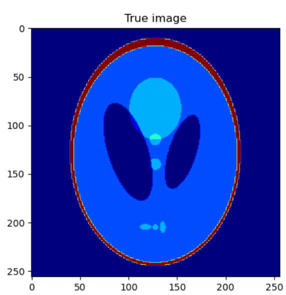

Figure 1: Noiseless image

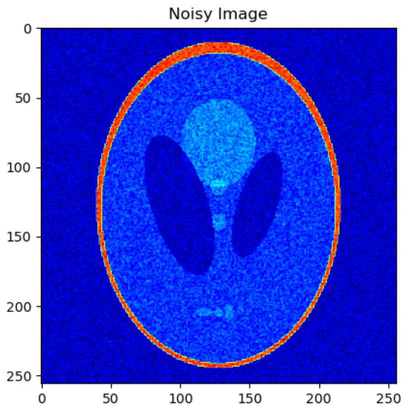

Figure 2: Noisy image

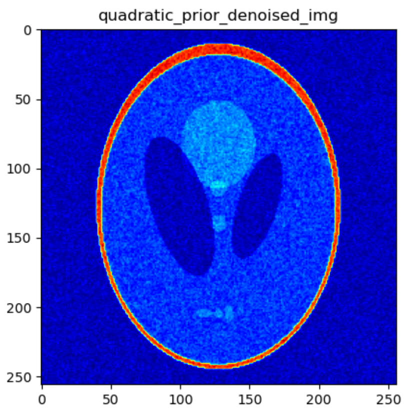

Figure 3: Image denoised using quadratic prior and optimal parameter tuning

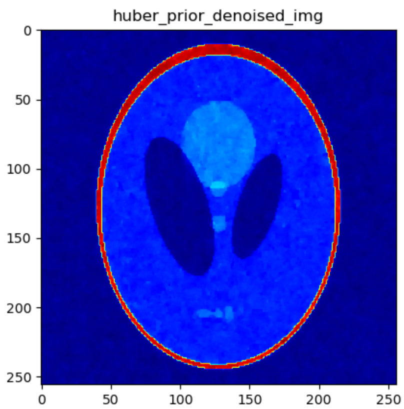

Figure 4: Image denoised using Huber prior and optimal parameter tuning

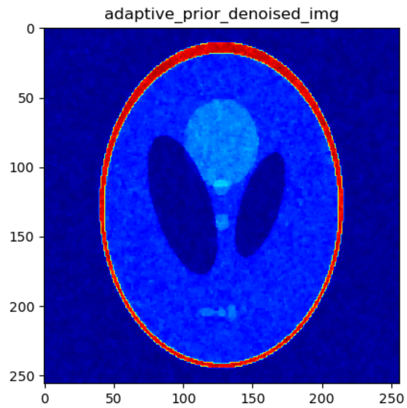

Figure 5: Image denoised using discontinuity-adaptive prior and optimal parameter tuning

1.3 Objective function plots

The 3 plots obtained are as follows :

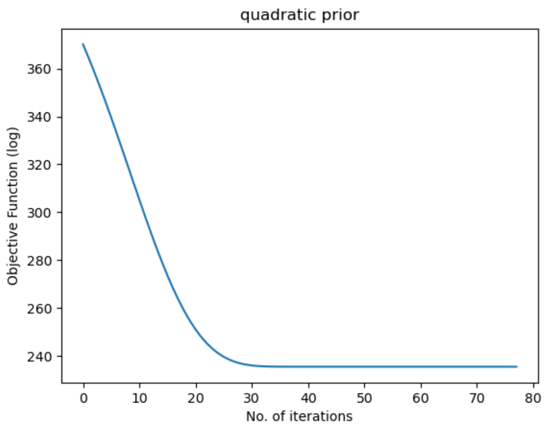

Figure 6: Quadratic prior

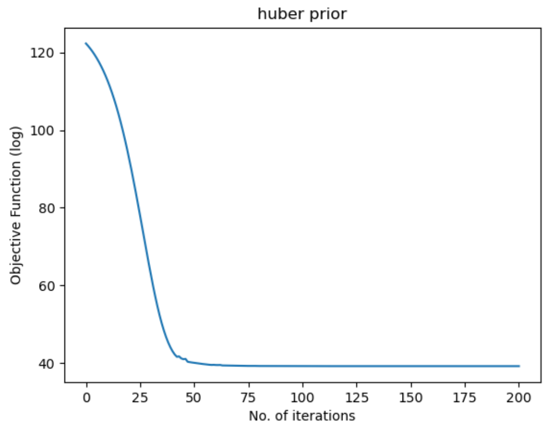

Figure 7: Huber prior

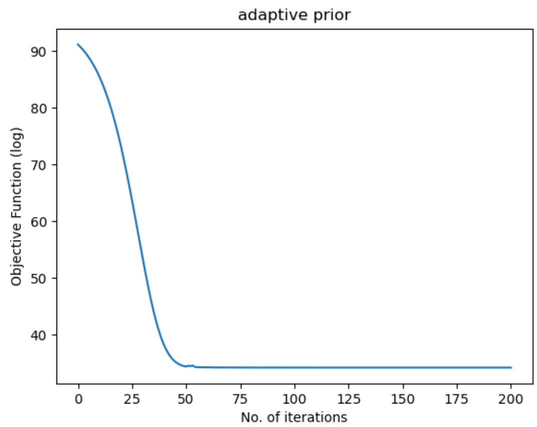

Figure 8: Discontinuity adaptive prior

2 Problem 2

The initial RRMSE between the Noiseless image and Noisy image is 0.14242647963234373

2.1 Optimised RRMSE values

2.1.1 Quadratic Prior

a* (optimal value of alpha) = 0.133
RRMSE at a* 0.12232447114496871
RRMSE at 0.8a* 0.12285527436077741
RRMSE at 1.2a* 0.1227640861019232

2.1.2 Huber Prior

a*(optimal value of alpha) = 0.42
b*(optimal value of gamma) = 0.062
RRMSE(a*,b*) 0.11473764982964398
RRMSE(1.2a*,b*) 0.11590240434492519
RRMSE(0.8a*,b*) 0.11677337892123574
RRMSE(a*,1.2b*) 0.11531265777967496
RRMSE(a*,0.8b*) 0.11500421041901811

2.1.3 Discontinuity Adaptive Prior

a*(optimal value of alpha) = 0.56
b*(optimal value of gamma) = 0.05
RRMSE(a*,b*) 0.11522800015682062
RRMSE(1.2a*,b*) 0.11795880719094377
RRMSE(0.8a*,b*) 0.11809573371356286
RRMSE(a*,1.2b*) 0.11553970589375076
RRMSE(a*,0.8b*) 0.11546636068864137

2.2 Images

The 5 images obtained are as follows :

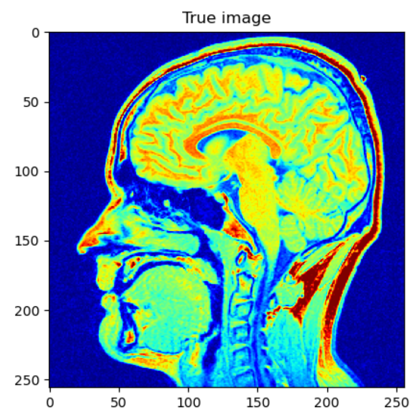

Figure 9: Noiseless image

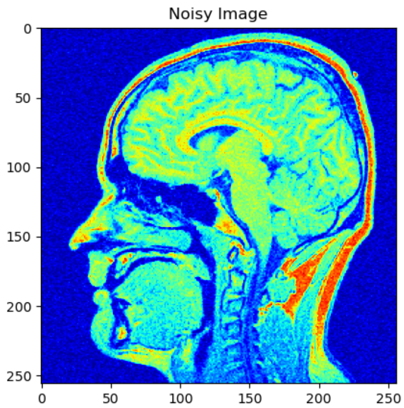

Figure 10: Noisy image

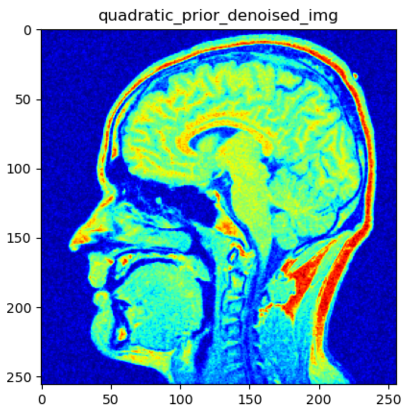

Figure 11: Image denoised using quadratic prior and optimal parameter tuning

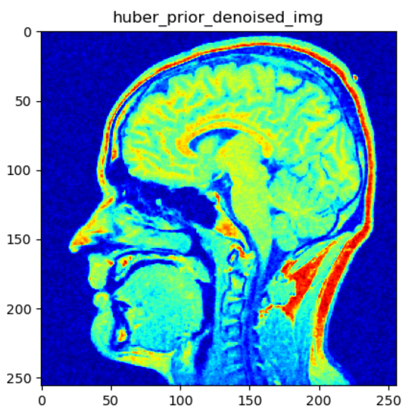

Figure 12: Image denoised using Huber prior and optimal parameter tuning

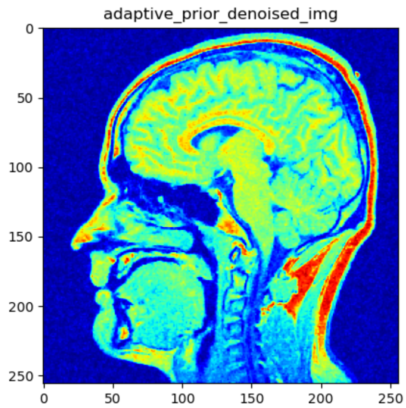

Figure 13: Image denoised using discontinuity-adaptive prior and optimal parameter tuning

2.3 Objective function plots

The 3 plots obtained are as follows :

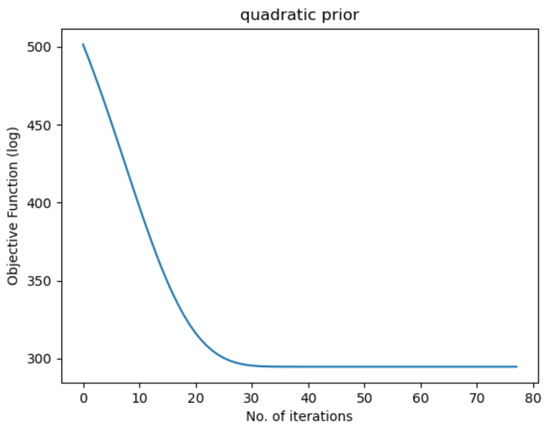

Figure 14: Quadratic prior

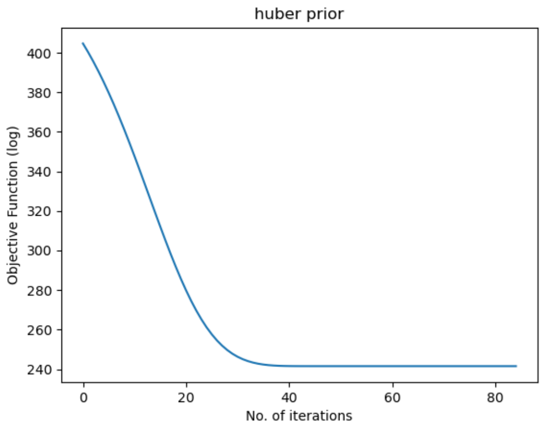

Figure 15: Huber prior

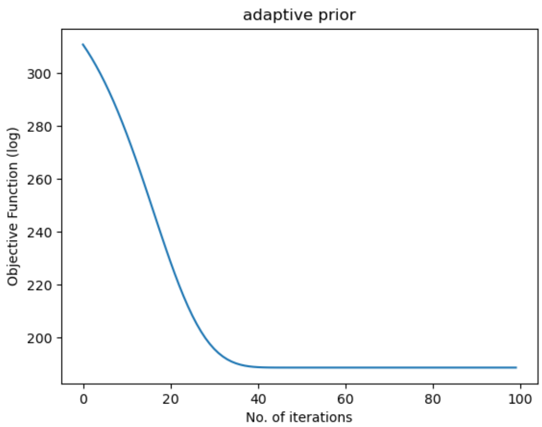

Figure 16: Discontinuity adaptive prior

3 Problem 3

To construct a prior model which depends on the three color channels, which are dependent on each other, we need to transform the RGB image to a system of values where the channels are independent upto some extent. One such system is the Hue-Saturation-Value, **HSV** format. This system upto some extent is independent in the three coordinates.

3.1 Proposed MRF Prior:

Three independent Huber prior for each channel, where a clique is represented by the similar 4 neighbourhood system in each coordinate.For the i^{th} channel we have the following Potential functions as:

$$g_i(u) = \begin{cases} \frac{1}{2}|u|^2, & \text{if } |u| \leq \gamma_i \\ \gamma_i |u| - \frac{\gamma_i^2}{2}, & \text{otherwise} \end{cases}$$

3.2 Noise Model

Three independent Gaussian noise models, with standard deviation 1, and weighting factor α_i for each channel i. Thus at the j^{th}pixel for the i^{th} channel we have :

$$G_{\alpha_i}(y_j|x_j) \propto e^{-\alpha_i|y_j-x_j|^2}$$

3.3 Bayesian Denoising Formulation

For each channel we thus have the negative log likelihood given by:

$$-log(P(x|y) = \alpha_i(|y_j - x_j|^2) + (1-\alpha_i) \sum_{x \in A_j(x)} g_i(x_j - x)$$

Thus for gradient based optimization , we have the derivative at voxel j in the i^{th} channel given by:

$$-\frac{d}{dx_j}log(P(x|y)) = 2\alpha_i|y_j - x_j| +$$

$$(1-\alpha_i) \sum_{x \in A_j(x)} I(|x_j - x| \leq \gamma_i)(x_j - x) + (I(|x_j - x| > \gamma_i))\gamma sgn(x_j - x)$$

where,

$I(x) = 1$ if x is True, else 0

$A_j(x) = \{x_j^{up}, x_j^{down}, x_j^{left}, x_j^{right}\}$