2a)

Two solutions immediately come to mind. First, as the simultaneous processing of channels and second, as the separate processing of separate channels.

Second Strategy is selected. We processed each color channel independently , so for a given prior , each channel would have different optimal parameters also (alpha and gamma). Then applied gradient descent on each channel independently and then combined the three channels . This idea is inspired by many image processing algorithms for RGB which process each color channel independently , like jpeg compression algo (compresses individually).

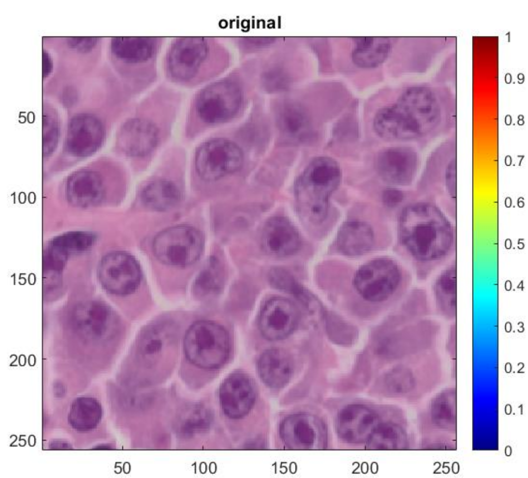
Pros:

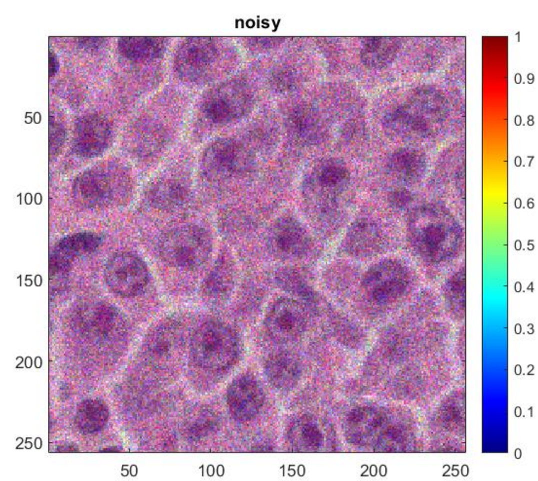
1) Noise may not be uniform across all channels. Several factors (such as lighting) can skew the error in one channel while keeping the other channels error free.
2) It will lead to better RMSE values than the first strategy because the latter strategy only identifies the "average" optimization parameters.

Cons:

1) It can shift original colors of the denoised image. Because each color is denoised separately, different amount of change is added/subtracted to different channels leading to a relative shift between colors.

Weighing the pros and cons, second strategy was adopted. The obtained image did not show detectable color shift, forcing us to adopt the method.





Q2

QUADRATIC $[(\alpha),(\mathbf{0.8}\alpha),(\mathbf{1.2}\alpha)]$ <- rmse_my values in this order

$[\alpha_r\ \alpha_g\ \alpha_b]$ = [0.77 0.625 0.77];

Red Channel:

rmse_my = 0.048472 0.051975 0.063129

rmse_noisy = 0.18775

Green Channel:

rmse_my = 0.072246 0.0784 0.07586

rmse_noisy = 0.23064

Blue Channel:

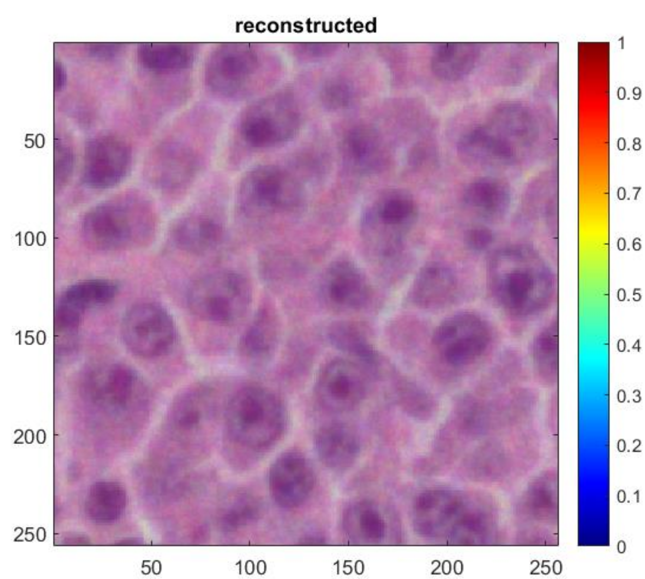
rmse_my = 0.044708 0.051553 0.052603

rmse_noisy = 0.19485

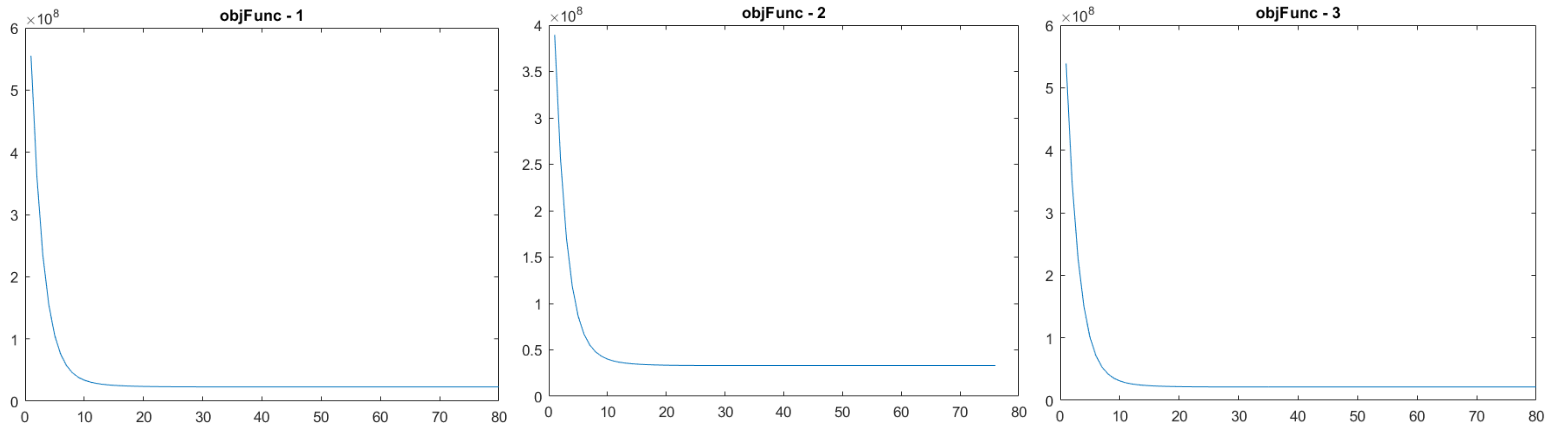
Overall:

rmse_my = 0.043436 0.05262 0.048646

rmse_noisy = 0.19938



Objective Function for R, G, B



Q2

Discontinuity-adaptive Huber function

$[(\alpha,\gamma),(0.8\alpha,\gamma),(1.1\alpha,\gamma),(\alpha,0.8\gamma),(\alpha,1.1\gamma)]$<- rmse_my values in this order

$[\alpha_r\ \alpha_g\ \alpha_b]$ = [0.87 0.84 0.89];
$[\gamma_r\ \gamma_g\ \gamma_b]$ = [8.6 9 7.2];

Red Channel:

rmse_my = 0.047113 0.089177 0.061476 0.048139 0.047321

rmse_noisy = 0.18775

Green Channel:

rmse_my = 0.070526 0.1076 0.089429 0.071573 0.07126

rmse_noisy = 0.23064

Blue Channel:

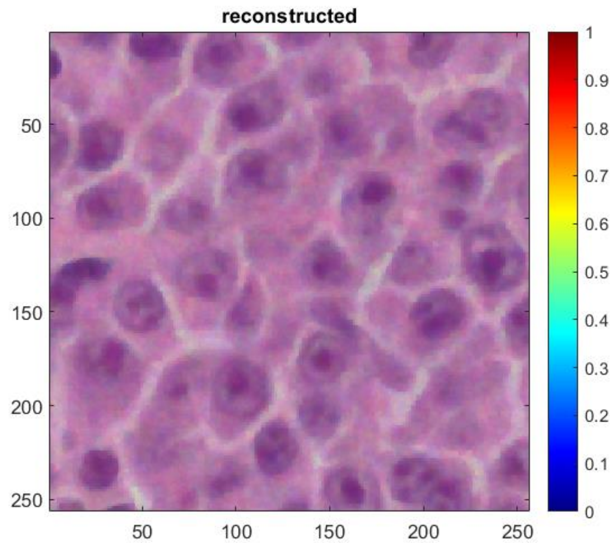
rmse_my = 0.043924 0.09659 0.064346 0.044642 0.0441

rmse_noisy = 0.19485

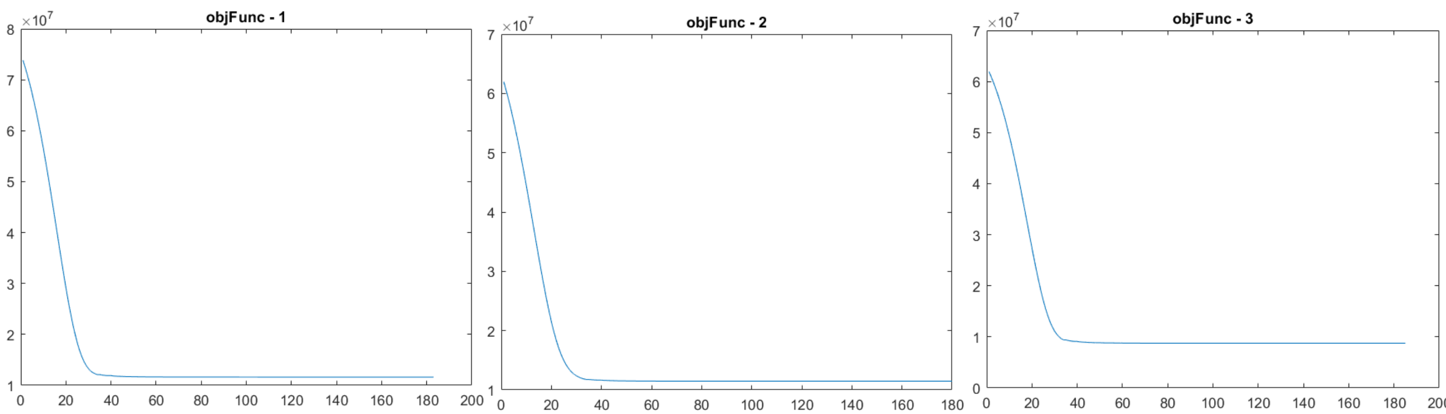
Overall:

rmse_my = 0.051375 0.095813 0.068786 0.052287 0.051709

rmse_noisy = 0.19938



<u>Objective Function for R, G, B</u>



Q2

Discontinuity-adaptive function

$[(\alpha, \gamma), (0.95\alpha, \gamma), (1.05\alpha, \gamma), (\alpha, 0.8\gamma), (\alpha, 1.2\gamma)]$ <- rmse_my values in this order

$[\alpha_r\ \alpha_g\ \alpha_b]$ `= [0.91 0.89 0.91];`
$[\gamma_r\ \gamma_g\ \gamma_b]$ `= [11 9.1 17.73];`

Red Channel:

rmse_my = 0.047349 0.052217 0.054562 0.047558 0.047493

rmse_noisy = 0.18775

Green Channel:

rmse_my = 0.070519 0.075544 0.080574 0.07085 0.071003

rmse_noisy = 0.23064

Blue Channel:

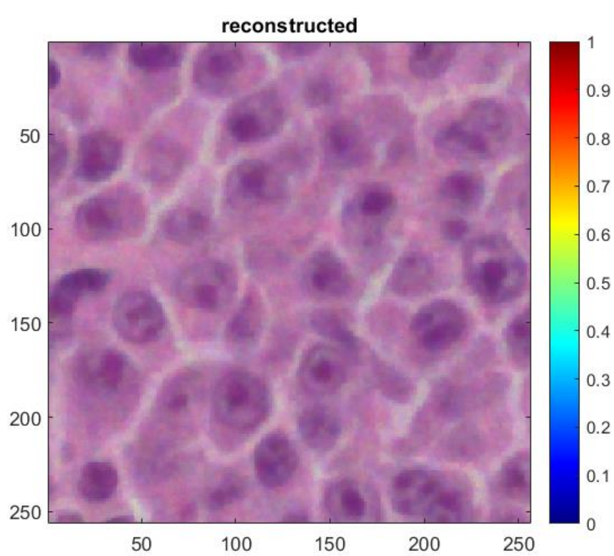
rmse_my = 0.044255 0.047681 0.048873 0.044314 0.044284

rmse_noisy = 0.19485

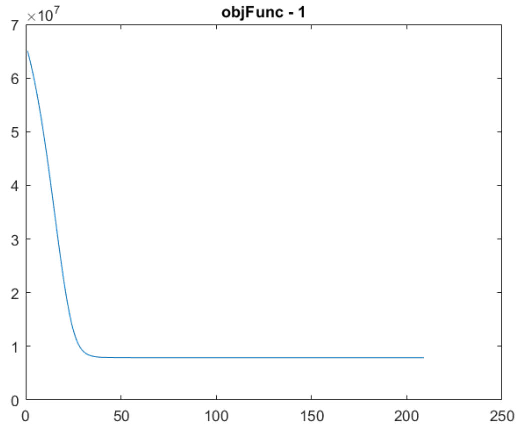
Overall:

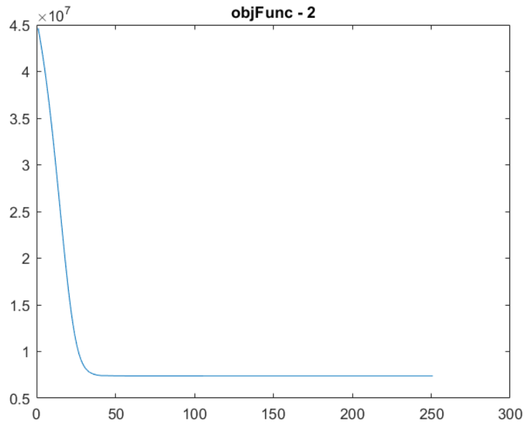
rmse_my = 0.051574 0.055924 0.058559 0.051762 0.051767

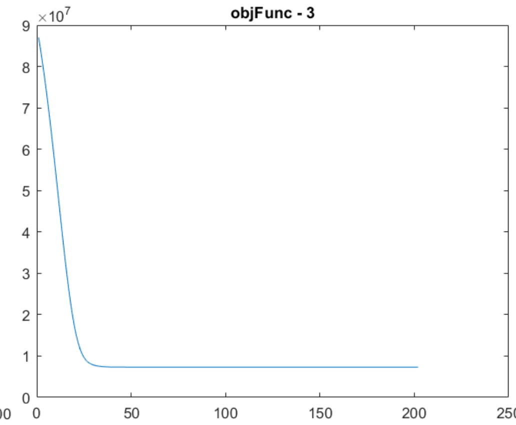
rmse_noisy = 0.19938



Objective Function for R, G, B


objFunc - 1
×10^7
7
6
5
4
3
2
1
0
0
50
100
150
200
250


objFunc - 2
×10^7
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250
300


objFunc - 3
×10^7
9
8
7
6
5
4
3
2
1
0
0
50
100
150
200
250