

[1] RMS boundaries for sample & test images (test images released!)

We announce baseline, advanced boundaries for sample images we provided.

Also, **test images and its RMS boundaries are released**, which will be used for grading your outputs.

For testing, we will first center crop output images and then compare RMS using cropped images. We also release calculate_rms_cropped function which will be used for grading. Please check utils.py in attachment which contains newly released calculate_rms_cropped function.

RMS Boundaries are set slightly higher than the outputs of our code so that you can easily get inside of boundaries if your code is implemented properly.

Below are RMS boundaries for sample & test images. RMS are measured with calculate_rms_cropped function.

sample

name	RMS baseline	RMS advanced
sample1	8.70	8.61
sample2	6.15	6.10
sample3	8.12	8.07
sample4	3.18	0.79

test (Will be used for grading)

name	RMS baseline	RMS advanced
test1	8.22	7.84
test2	6.79	6.15
test3	3.45	3.21
test4	6.42	0.98
test5	5.23	0.31

[2] Restrictions

- 1) **kernel_size = 1 is not allowed.** Also, at least one convolution should be conducted. (There will be minus point if your output image is identical to src_img)
- 2) Maximum of kernel_size is 15, and it is odd number. It means, kernel_size variable should satisfy following condition.
 $3 \leq \text{kernel_size} \leq 15, \text{kernel_size} \% 2 == 1$
- 3) **sigma_s = 75, sigma_r = 75 for bilateral filter.** You can use different sigma value if you want, but we recommend you to use these values, since our baseline, advanced code used these.
- 4) For border, you may use **zero padding** to make output image size same as input image size. You can also use border reflection and it will not affect to your score, since we will calculate RMS with cropped images.
- 5) For bilateral filter, we recommend you to use gaussian function below.

$$G(x) = e^{-\frac{1}{2}(\frac{x}{\sigma})^2}, \sigma = \text{sigma value (sigma_s, sigma_r)}$$

For space gaussian function, use Euclidian distance, and for range gaussian function, use L1 distance.

[3] Grading Policy Details

- For apply_average_filter, apply_median_filter functions, it will be graded by test cases. 5 test cases will be used for grading, 1 point for each test case. (5 point for each function)
- For apply_bilateral_filter function, it will be also graded by test cases. 5 test cases will be used for grading, 2 point for each test case. (10 point)
- For task1 function, it will be graded by using 5 test images we provided. 2 point for each test image, if your output is in our RMS baseline boundary. (10 point)
- For extra point, same 5 test images will be used for grading. 2 point for each test image, if your output is in our RMS advanced boundary. (extra 10 point)