Task 2. Filtering and metrics

Mandatory part of the task

A program should be developed that implements the main image filtering algorithms and metrics:

- Median Image Filtering
- Gaussian filter
- Bilateral filtration
- MSE metric
- PSNR metric
- SSIM Metric

Additional part of the task

- Determining whether the second image is a shifted and rotated version of the first

Directions

Particular attention should be paid to processing at the edges of the image, where filtering algorithms need to access pixels that lie outside the original image. In this case, you should continue the image by duplicating the nearest pixel. For example, when accessing a pixel with coordinates (3, -4) , you should return the value of the pixel (3, 0).

When calculating metric values and bilateral filtering, note that the range of image pixel values is [0, 255] . If you are working with images in the [0, 1] range , then you need to recalculate the values of the metrics and parameters accordingly.

Bilateral filtration is applied according to the formulas:

$$J(i,j) = \frac{\sum_{k,l} I(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|I(i,j) - I(k,l)\|^2}{2\sigma_r^2} \right)$$

The kernels of Gaussian filters and bilateral filtering have an unlimited carrier, which makes it difficult to apply filters "on the forehead", therefore, in practice, calculations are limited to some neighborhood of the processed pixel. For the Gaussian filter and bilateral filtering, the optimal radius is $3\sigma_d$. Using a larger filter does not make sense, since the contribution of the cutoff coefficients does not exceed the pixel intensity gradation. And the use of a smaller filter already leads to distortion due to the loss of coefficients.

An additional part of the task is proposed to be done using the properties of the Fourier transform:

- Calculate the Fourier transform, take the amplitude.
- Convert spectrum image to polar coordinates. Previously, you can apply blur to reduce aliasing.
- Calculate the Fourier transform again, take the amplitude. It suffices to apply the Fourier transform only to the axis corresponding to the polar angle.
- Compare the received images according to one of the metrics (you can determine the threshold yourself).

Command line parameter format

The program must support launching from the command line with a strictly defined command format:

```
python main.py (command) (parameters...)
```

Command List:

`mse (input_file_1) (input_file_2)`	Calculate the value of the MSE metric and print it to the console
`psnr (input_file_1) (input_file_2)`	Calculate the value of the PSNR metric and print it to the console
`ssim (input_file_1) (input_file_2)`	Calculate the value of the SSIM metric and print it to the console
`median (rad) (input_file) (output_file)`	Median filtering with a window of size (2rad+1) × (2rad+1)
`gauss (sigma_d) (input_file) (output_file)`	Gaussian filter with parameter σ_d
`bilateral (sigma_d) (sigma_r) (input_file) (output_file)`	Bilateral filtration with parameters σ_d and σ_r
`compare (input_file_1) (input_file_2)`	Determining whether the second image is a shifted and rotated version of the first, and outputting this information to the console as a single digit: 0 (not) or 1 (is).

Value `rad` is a positive integer, values`sigma_d` и `sigma_r` – вещественные положительные. Гарантируется, что размеры изображений `input_file_1` и `input_file_2` совпадают.

Remarks

There should be no extraneous screen output other than what is required in the job. When calculating metric values, all the program should output is a single number.

The time limit for running the program for a 400×300 image is 1 minute.

All metrics must be calculated for a single band grayscale image. If you are working with an image in color, then just take the first component. Explanation on SSIM: it is not necessary to beat the image into blocks, it is enough to process the image as one large block.