Average filter:
Produce a k×k filter kernel with a matrix (all elements are 1/k×k), then make a convolution of the original image
Median Filter:
sort k × k filter kernels with a complexity of O(k^2logk) , then take two noise maps of 450*800 pixels with produce-time 0.277657s（k=3），1.086158s（k=5），2.469530s（k=7）(Basically proportional to k^2lnk) . Filter kernel changes k pixels when each time right shift.
Build two sticks small and large , The numbers of the left half and the right half of the ordered array are saved, and the length of the small is longer than or equal to the length of the large. Traversing the array Nums, if i>=k , Explain that the sliding window already has k elements, and then slide to delete the leftmost value. Find the leftmost value in small and large sticks, and delete it if you have one. Then deal with the situation of increasing the number. It comes to be 2 situations: 1. If length of small one smaller than the length of large one with the large one is empty or the new element is small than the first element in large one, then put this element into small one , or else put the first element of large one into the small one , and put the new element into large one. 2. If length of small one longer than the length of large one with new element larger the last element of the small one, then put the new element into large one, or else put out the last element of small one and put it into large one and put the new element into small one. At last calculate the middle value and put it in the result res.
Bilateral Filter:
The spatial weight sigma takes r/3, r is the filter kernel radius. Use Normal distribution of 3sigma principle, Pixels more than 3sigma(r) have little effect on filter kernel filtering results.
According to the formula of the paper the pixel weight of the search zone center calculated by Nlm Filter must be 1 and the others are less than 1.

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(\mathcal{N}_i)-v(\mathcal{N}_j)\|_{2,a}^2}{h^2}},$$

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{\|v(\mathcal{N}_i)-v(\mathcal{N}_j)\|_{2,a}^2}{h^2}}$$

（from *A non-local algorithm for image denoising*）When the sigma is small, the weights of other pixels are much smaller than 1, so that the gray value after filtering is close to the center pixel. In order to avoid the center pixel as the noise, the weight of the center pixel is set to the maximum value of the weight of the other pixels, and the filtering effect is slightly improved. Trying to apply this trick to the bilateral filter does not work.