Smoothing and Blurring

Tushar B. Kute,
http://tusharkute.com

Kernel

- In computer vision, a kernel, also known as a convolution matrix or mask, is a small matrix used to manipulate images in various ways.
- Think of it as a tiny window that "slides" across the image, performing calculations at each pixel based on the surrounding pixels and the kernel itself.
- These calculations can achieve various effects, making kernels incredibly versatile tools.

Kernel : Functions

- Smoothing: Blurs images by averaging surrounding pixel values (e.g., average kernel).
- Sharpening: Enhances edges by emphasizing differences between neighboring pixels (e.g., Laplacian kernel).
- Edge detection: Highlights edges by identifying large differences in pixel values (e.g., Sobel kernels).
- Embossing: Creates a 3D effect by highlighting edges with shadows and highlights (e.g., embossing kernel).
- Feature extraction: Extracting specific features from the image like lines, textures, or corners (e.g., custom kernels).

Kernel : Properties

- Properties:
  - Size: Typically small matrices, like 3x3 or 5x5, though larger kernels exist for specific tasks.
  - Values: Each element in the kernel represents a weight applied to the corresponding pixel during calculation.
  - Convolution: The process of sliding the kernel across the image and performing element-wise multiplication with the underlying image pixels.

Kernel : Applications

- Image processing:
  - Preprocessing images for tasks like object detection, image recognition, and segmentation.
- Deep learning:
  - Convolutional neural networks rely heavily on kernels for feature extraction and image classification.

Kernel : Example

- An image kernel is a small matrix used to apply effects like the ones you might find in Photoshop or Gimp, such as blurring, sharpening, outlining or embossing.
- They're also used in machine learning for 'feature extraction', a technique for determining the most important portions of an image.

Kernel : Example

- To see how they work, let's start by inspecting a black and white image.
- The matrix on the left contains numbers, between 0 and 255, which each correspond to the brightness of one pixel in a picture of a face.
- The large, granulated picture has been blown up to make it easier to see; the last image is the "real" size.

Kernel : Example

206	205	247	245	244	253	247	245	136	151	255	255	255	255	255	255	234	207	231	255	254	254	255	255	254	255	252	255	255	254	255	247
244	161	137	244	254	255	254	255	118	103	209	228	155	153	236	193	74	52	66	173	255	254	254	255	255	255	254	255	254	253	244	184
192	154	75	200	249	255	255	255	110	96	84	61	35	44	89	53	44	45	43	54	140	213	253	255	255	255	255	245	187	186	176	223
90	109	96	143	223	255	255	252	117	75	41	35	31	24	25	36	45	44	44	46	81	118	148	234	252	254	255	248	231	248	255	254
67	69	107	196	236	255	255	255	104	25	34	35	29	20	25	34	32	30	32	34	53	85	100	142	231	242	247	249	255	255	255	255
55	51	45	134	218	251	255	232	51	12	26	33	24	24	46	75	82	78	71	66	58	53	67	90	136	228	208	158	253	246	249	255
79	58	56	75	224	255	255	118	11	27	74	99	91	106	140	162	173	173	173	172	158	137	92	46	78	187	217	206	254	222	233	255
38	43	47	52	147	255	229	56	41	81	129	145	160	169	169	172	178	179	178	179	177	177	172	110	31	82	209	238	255	244	249	255
40	40	33	36	90	245	171	32	65	110	139	145	151	162	171	174	178	179	182	184	187	183	173	162	71	45	167	255	254	255	254	255
37	44	44	31	69	250	158	36	70	129	143	142	153	162	171	175	177	178	182	191	194	188	180	170	120	51	137	255	254	250	254	255
34	45	51	64	116	237	181	53	116	138	140	143	154	164	176	178	174	177	183	186	185	185	183	178	140	66	141	254	252	225	249	255
34	36	52	74	71	188	156	63	131	134	144	155	160	161	173	179	178	179	189	193	190	185	187	182	156	93	148	250	254	214	247	255
32	38	52	54	159	250	126	57	129	138	138	140	151	156	166	168	171	178	180	187	186	185	185	183	180	102	136	242	255	255	254	254
36	32	72	129	212	228	115	65	121	104	102	104	94	103	134	158	170	162	125	106	121	143	155	190	191	104	134	230	253	253	255	251
61	82	116	107	179	247	124	60	101	90	111	119	103	81	94	147	191	178	126	98	123	153	147	161	200	92	100	222	207	167	227	215
144	178	167	231	210	232	170	67	115	88	76	62	83	85	88	139	192	190	135	80	53	99	141	165	201	97	79	192	245	235	248	249
127	145	149	195	204	213	197	95	133	122	117	133	126	108	110	139	191	197	167	129	127	148	147	171	188	110	121	228	233	180	215	212
87	112	100	79	85	82	65	75	142	148	151	153	138	125	120	149	191	190	193	175	174	193	198	190	208	127	163	239	219	149	196	195
63	83	109	134	129	106	39	78	132	142	155	159	139	111	124	164	195	200	186	192	191	195	200	202	200	143	217	253	249	242	238	234
69	78	78	113	97	74	43	106	127	140	152	155	125	97	112	150	185	194	174	183	196	198	202	208	209	166	247	254	255	254	254	254
72	44	63	59	46	52	49	74	127	137	146	149	132	103	78	90	134	141	168	165	199	207	204	203	216	193	236	244	251	242	236	243
55	20	69	73	59	80	46	74	117	127	144	161	148	124	105	120	156	187	193	162	189	206	201	205	214	194	174	185	197	188	183	193
65	49	77	89	50	68	43	61	109	127	141	147	113	100	121	145	148	169	181	176	181	201	201	205	202	174	166	169	178	183	188	184
82	76	92	79	54	58	37	47	90	121	132	116	89	78	111	146	163	149	122	124	180	197	197	198	178	149	146	152	155	157	159	168
104	107	122	123	105	79	27	33	66	111	122	120	114	114	147	175	190	196	163	101	170	200	187	185	156	146	145	139	137	141	140	145
117	124	127	133	135	105	21	28	37	88	115	121	128	128	141	142	168	202	212	153	164	186	180	168	154	146	144	149	151	151	147	144
119	118	118	125	128	111	21	29	28	58	100	118	131	140	151	159	186	201	205	192	180	168	149	166	119	144	147	143	140	141	144	148
117	119	125	130	139	106	18	29	44	58	70	102	133	147	168	197	212	215	210	195	177	152	133	195	57	59	126	151	145	143	142	141
115	123	126	134	145	102	27	54	52	38	45	69	105	135	175	189	193	216	206	166	139	111	164	203	74	5	121	151	142	142	143	146
101	108	123	121	132	105	44	40	31	35	57	44	58	101	147	144	138	163	145	94	90	145	196	187	84	48	185	180	142	144	142	145
98	97	97	96	104	76	34	33	30	48	41	49	51	58	74	53	55	66	63	89	150	188	209	156	62	108	140	149	125	133	131	131
102	102	97	88	73	35	30	23	42	50	65	41	90	60	59	51	57	82	123	157	187	205	169	62	96	151	105	101	154	135	130	129

Kernel : Example

sharpen

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Below, for each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right. Hover over a pixel on either image to see how its value is computed.

input image

$$\left(\begin{array}{ccccc} 165 \times 0 & + & 160 \times -1 & + & 142 \times 0 \\ + \; 140 \times -1 & + & 149 \times 5 & + & 125 \times -1 \\ + \; 105 \times 0 & + & 101 \times -1 & + & 154 \times 0 \end{array} \right)$$

$$= 219$$

kernel:

sharpen

output image

Convolutional Operator

- The convolution operator, often denoted by an asterisk (*), is a powerful mathematical tool used in various fields, including signal processing, image processing, and machine learning.
- It essentially blends two functions together by sliding one of them over the other and multiplying corresponding elements before summing the products.
- This process extracts specific features from the input signals or images.

Convolutional Operator

- Imagine you have two signals:
  - Signal 1: Represents a song's melody played on a piano.
  - Signal 2: Represents a filter that highlights higher frequencies.
- By convolving these signals, you're essentially asking: "How much of the high-frequency content is present in the melody at each point in time?".
- The resulting signal would emphasize the high-pitched notes in the melody, revealing its prominent features.

Convolutional Operator

- Image 1: A photograph with various textures and edges.
- Kernel 1: A small matrix with positive values in the center and negative values around it (edge detection kernel).
- Convolving the image with this kernel would enhance the edges and boundaries within the image, making them appear sharper and more prominent.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Averaging Kernel

- In the world of image processing, averaging kernels, also known as mean filters, play a crucial role in smoothing images and reducing noise. They achieve this by blending the intensities of neighboring pixels, resulting in a more uniform and visually pleasing image.
- Understanding the Principle:
  - Imagine an averaging kernel as a tiny window sliding across your image.
  - At each pixel, it multiplies the corresponding pixel intensity with its weight (usually all positive and equal) and then sums the results.
  - This new average value becomes the output pixel in the processed image.

Averaging Kernel: Benefits

- Noise reduction: Blurs out random pixel variations, making the image appear smoother and less "grainy."
- Edge and detail preservation: Unlike some aggressive smoothing filters, averaging kernels can retain important edges and details by not aggressively blurring them.
- Wide range of applications: Useful for pre-processing images for tasks like feature extraction, segmentation, and compression.

Averaging Kernel

- Practical

Gaussian Kernel

- The Gaussian kernel, named after the famous mathematician Carl Friedrich Gauss, takes inspiration from the bell curve to bring elegance and precision to the world of image processing.
- Just like the gentle slope of a Gaussian distribution, this special type of kernel smooths images beautifully, attenuating noise while preserving essential details.

Gaussian Kernel

- The Gaussian kernel, also known as the Gaussian filter, works by applying a weighted average to the pixels in a specific neighborhood around each pixel in an image.
- This weighting is based on the Gaussian distribution, which gives higher weights to closer pixels and lower weights to farther pixels.
- This creates a smooth blurring effect while preserving edges.

Gaussian Kernel: Working

- Gaussian function:
  - Imagine a bell-shaped curve, where the highest point is in the center and the values gradually decrease towards the sides. This curve represents the Gaussian function.
  - The distance from the center determines the weight assigned to each pixel. Closer pixels (closer to the center) have higher weights, contributing more to the final value.

Gaussian Kernel: Working

- Kernel creation:
  - A small square matrix represents the kernel, typically odd-sized (e.g., 3x3, 5x5).
  - Each element in the kernel corresponds to a weight based on its position relative to the center.
  - The center element usually has the highest weight, and weights decrease as you move farther away.

Gaussian Kernel: Working

- Convolution:
  - The kernel is "slided" over the image, one pixel at a time.
  - At each position, the kernel elements are multiplied by the corresponding pixel values in the image.
  - These products are then summed up, giving a weighted average for the target pixel in the filtered image.

Gaussian Kernel: Working

- Blurring effect:
  - Pixels with nearby neighbors of similar values will see smaller changes in their intensity after filtering.
  - Pixels with significantly different neighbors will be more affected, leading to a smoothing effect.
  - Edges, defined by sharp changes in intensity, are partially preserved because the weights drop off quickly for distant pixels.

Gaussian Kernel: Features

- Smooths noise:
  - By averaging pixel values, the Gaussian kernel reduces random variations in intensity.
- Preserves edges:
  - The rapid weight drop-off with distance helps maintain sharp transitions between regions.
- Adjustable blur:
  - The kernel size and sigma (standard deviation) control the blur strength. Larger sizes and higher sigma lead to stronger blurring.

Gaussian Kernel

- Practical

Median Kernel

- The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal.
- Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image).
- Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise, also having applications in signal processing.

Median Filter

- Non-linear filter:
  - Unlike Gaussian filtering, which uses weighted averages, median filtering replaces a pixel's value with the median value of its surrounding pixels within a defined neighborhood (similar to a kernel).
- Noise reduction:
  - It's particularly effective at removing "salt and pepper" noise, characterized by isolated bright or dark pixels.
- Edge preservation:
  - Since it replaces values based on local statistics, it tends to preserve edges better than Gaussian filtering, making it useful for scenarios where edge information is crucial.

Median Filter

Feature	Gaussian Filter	Median Filter
Operation	Weighted average	Median selection
Noise reduction type	Gaussian	Impulsive ("salt and pepper")
Edge preservation	Moderate	Good
Non-linearity	Yes	Yes

Median Filter vs Gaussian

- The choice between Gaussian and median filtering depends on your specific needs:
  - Gaussian filter: Use it for general noise reduction while maintaining some edge detail.
  - Median filter: Use it for removing impulse noise while preserving sharp edges.

Median Filter : Working

- Neighborhood definition:
  - Imagine a small square or rectangular window (similar to a kernel) sliding over the image, one pixel at a time. This window represents the neighborhood used for calculations.
  - The size of the neighborhood is critical. A larger window helps reduce noise but might blur edges, while a smaller window preserves edges but might be less effective at noise reduction.

Median Filter : Working

- Sorting pixel values:
  - Within each neighborhood, the filter collects the intensity values of all pixels.
  - These values are then sorted in ascending or descending order, creating a ranked list.

Median Filter : Working

- Median selection:
  - The median value from the sorted list is chosen. The median represents the "middle" value, unaffected by extreme outliers like noise spikes.
  - This median value replaces the original pixel value in the center of the neighborhood, effectively smoothing out the image.

Median Filter : Working

- Sliding window approach:
  - The process repeats as the window slides across the entire image, pixel by pixel.
  - Each pixel is replaced with the median of its local neighborhood, resulting in a denoised image.

Median Filter

- Practical

Bilateral Filter

- The bilateral filter is a powerful image processing tool that effectively removes noise while preserving edges.
- It achieves this by incorporating both spatial proximity and intensity similarity when considering neighboring pixels.

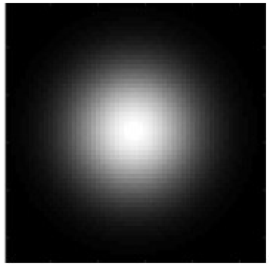
Bilateral Filter

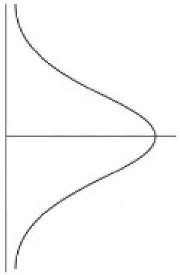
$$BF[I]_p = \boxed{\frac{1}{W_p}} \sum_{q \epsilon S} \boxed{G_{\sigma_s}(||p - q||)} \boxed{G_{\sigma_r}(|I_p - I_q|)} I_q$$

Normalization Factor

Space Weight

Range Weight





Bilateral Filter

- Here, the normalization factor and the range weight are new terms added to the previous equation. \sigma_s denotes the spatial extent of the kernel, i.e. the size of the neighborhood, and \sigma_r denotes the minimum amplitude of an edge.
- It ensures that only those pixels with intensity values similar to that of the central pixel are considered for blurring, while sharp intensity changes are maintained.
- The smaller the value of \sigma_r , the sharper the edge. As \sigma_r tends to infinity, the equation tends to a Gaussian blur.

Bilateral Filter: How it works?

- Combining Factors:
  - The bilateral filter calculates a weighted average of a pixel's neighbors, but the weights depend on two factors:
- Spatial proximity:
  - Similar to other filters, pixels closer to the target pixel receive higher weights. This ensures local smoothing.
- Intensity similarity:
  - Pixels with similar intensity values to the target pixel also receive higher weights. This helps preserve edges and textures.

Bilateral Filter: How it works?

- Gaussian Weights:
  - Both the spatial and intensity similarity are represented by Gaussian functions:
  - Spatial weight: Decreases with distance from the target pixel.
  - Intensity weight: Decreases with the difference in intensity between the neighbor and the target pixel.

Bilateral Filter: How it works?

- Weighted Average:
  - Multiplying these two weights for each neighbor gives the final weight.
  - Summing the weighted intensities of the neighbors provides the filtered pixel value.

Bilateral Filter: How it works?

- Balancing Smoothness and Edge Preservation:
  - The standard deviations of the Gaussian functions control the balance between spatial and intensity similarity.
  - Larger spatial standard deviation leads to more smoothing.
  - Larger intensity standard deviation allows for larger intensity differences, preserving more edges.

Bilateral Filter

- Practical

Thank you

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com