EQ: How can we edit images with convolution?

Mr. Mina
12th Grade Computer Vision
9-12.CT.2, 9-12.CT.5

Lesson 04 - Convolution as Image Editing

Do Now

This week, we've worked with two main convolution kernels (right). With your partner, discuss:

- Why are the weights different for the kernels? What impact might this have in our convolution?
- Can we have convolution weights greater than or equal to 1? What are the effects of this on our image?

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Class practice problem kernel

(L3, S3)

1/5	1/7	1/5
1/7	1/9	1/7
1/5	1/7	1/5

Solo practice problem kernel

(L3, S5)

Do Now Discussion

Different Weights

Used to give different levels of importance to surrounding pixels during convolution

1/5	1/7	1/5
1/7	1/9	1/7
1/5	1/7	1/5

Weights >= 1

We **can** have weights >= 1, but remember our pixel values are bounded (0-255) so then we must normalize afterward.

- What does normalize mean?

Pixel normalization

- **normalize**: to mathematically map values from one numerical range to another
  - Mapping the values from 0-100 to 0-10 by dividing by 10
- Given 5 values (`old_min, old_max, new_min, new_max, value`), the formula is:

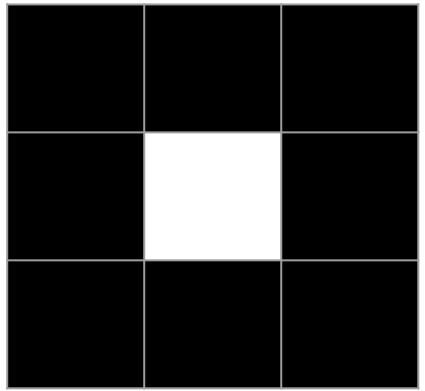
$$f(x) = \frac{value - old\_min}{old\_max - old\_min} * (new\_max - new\_min) + new\_min$$

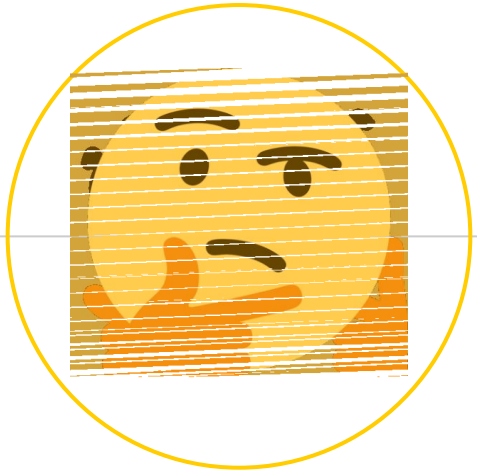
Kernels as images

- Since kernels are 2D arrays, we can also represent them as images. In kernel images, the brightness of a pixel represents its weight in the convolution.
- Black pixels to have a weight of 0 and white pixels to have a weight of 1. Gray pixels are in between.

0	0	0
0	1	0
0	0	0

in image form





Kernels can have negative numbers

Sometimes we want to give negative weight for certain pixels, to emphasize the difference

Kernels with negative numbers

- Since kernels are just 2D arrays, we can include negative numbers. To represent kernels with negative numbers:

Black: weight = –1

Dark gray: –1 < weight < 0

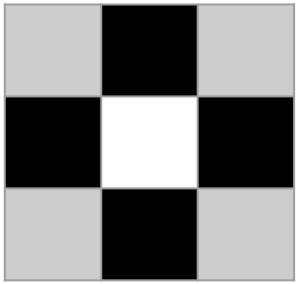
Gray: weight = 0

Light gray: 0 < weight < 1

White: weight = 1

0	-1	0
-1	5	-1
0	-1	0

=



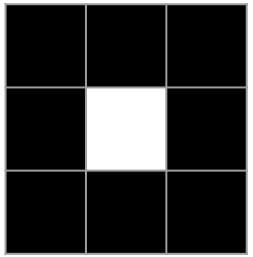
Roll for confidence!

?

Pokémon

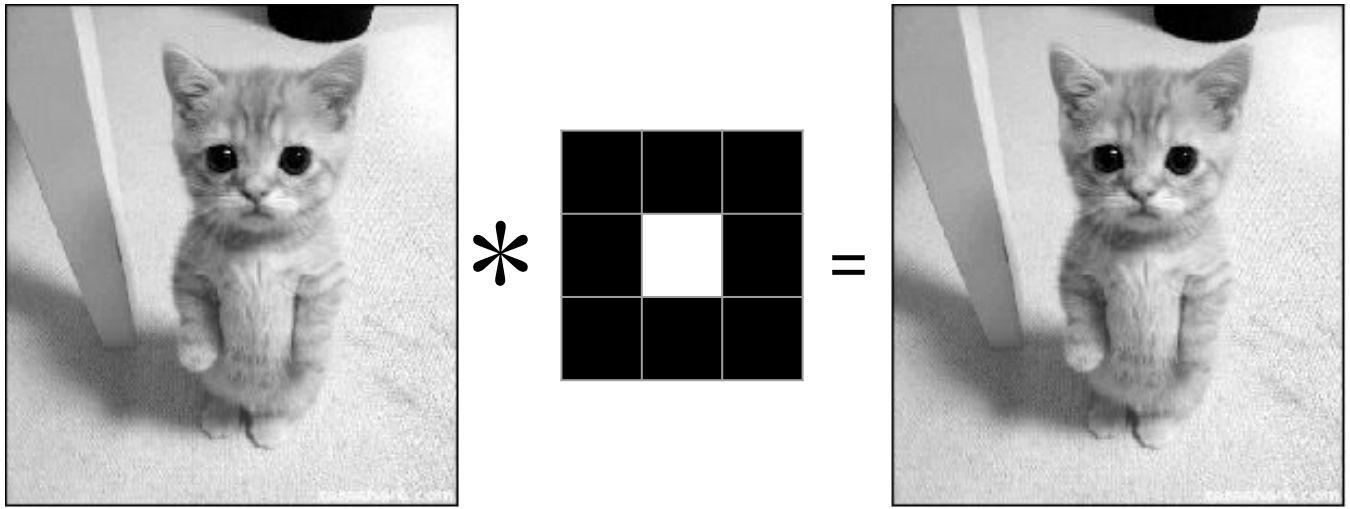
What's that kernel?

$*$



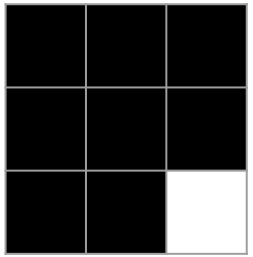
$=$

- **Identity kernel**: outputs the original image



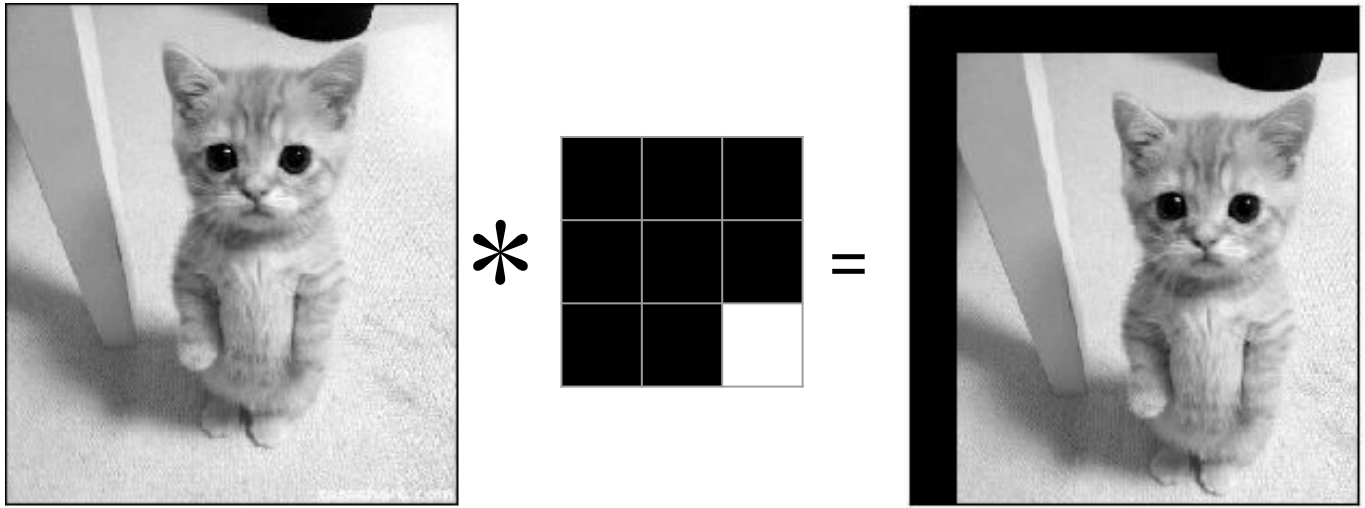
What's that kernel?

$*$



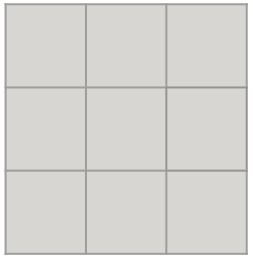
$=$

- **Shift kernel**: shifts the image; in this case, down and right



What's that kernel?

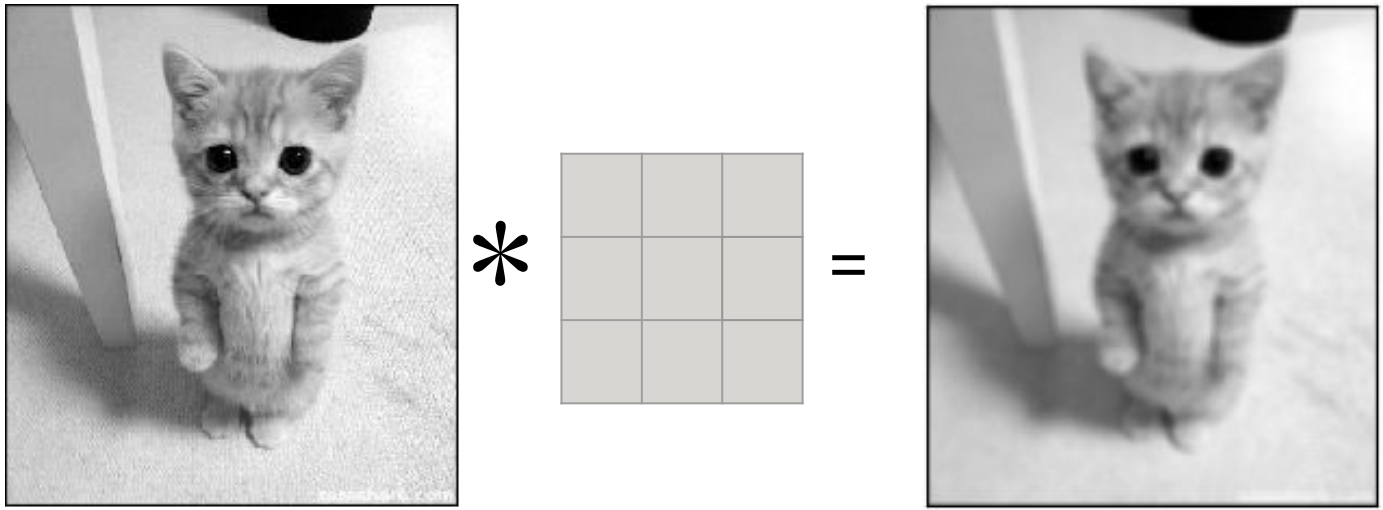
$*$



$=$

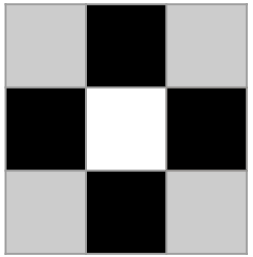
Block blur!

- **Block blur**: blurs the image with a “blocky” effect
  - To prevent over-saturation, the sum of each of the weights should be 1



What's that kernel?

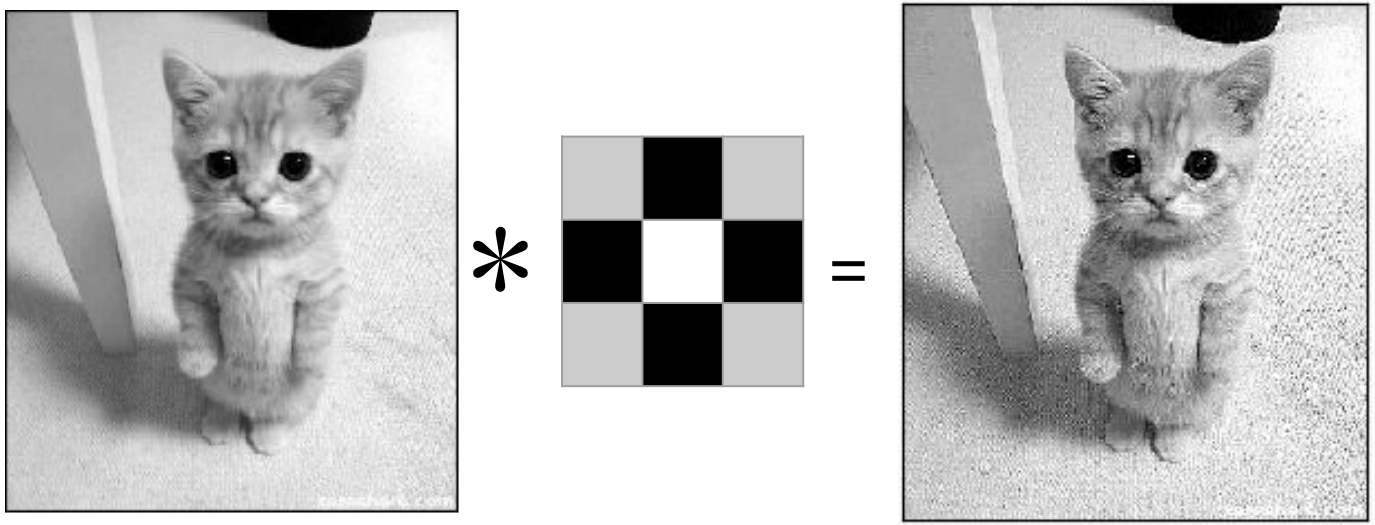
*



=

* this kernel has negative weights!

- **Sharpen kernel**: emphasizes differences in adjacent pixels



Basic kernels

Identity kernel

Outputs the original image unchanged.

0	0	0
0	1	0
0	0	0

Shift kernel

Shifts the image in the direction of the non-zero pixel.

0	0	0
0	0	0
0	0	1

Basic kernels

Block blur kernel

Blurs the image with some blockiness. The sum of weights should equal 1.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Sharpen kernel

Shifts the image in the direction of the non-zero pixel. Result needs to be normalized to 0-255 afterward.

0	-1	0
-1	5	-1
0	-1	0

Homework

04_homework on Google Classroom (link)

Using the starter code, perform image-editing convolutions in code.