

EQ: How do we perform 2D convolution?

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



Lesson 02 - Convolution Practice

Do Now

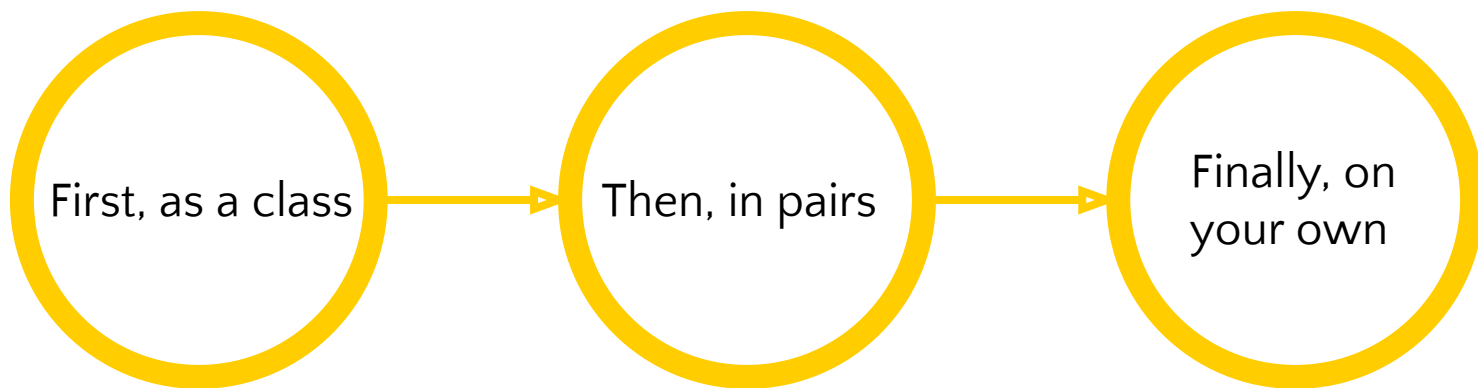
Ask and answer: Ask your partner a question you have about convolution and answer your partner's question. If both of you can't answer a question, ask another pair!





Today's agenda: lots of **convolution** practice

Three start-to-finish convolution problems today:





Review: convolution near borders

There are four different approaches when it comes to convolution near borders:

1. **Zero pad** – values are set to 0
2. **Circular** – values are copied from the opposite end of the list
3. **Replicate** – values are duplicates of the last row/column
4. **Symmetric** – the image is reflected and values are taken as their mirror image



Zero Pad



Circular



Replicate



Symmetric



Class convolution practice

Evaluate the following convolution using *circular padding* for the borders:

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

?	?	?
?	?	?
?	?	?



Class convolution practice: Pixel (0, 0)

First, let's pad the input with necessary values to use circular padding for the border values:

81	233	184
43	210	116
25	123	138

Add symmetric
padding

138	25	123	138	25
184	81	233	184	81
116	43	210	116	43
138	25	123	138	25
184	81	233	184	81



Class convolution practice: Pixel (0, 0)

Then, let's overlay the kernel on the target pixel and do the math

138	25	123	138	25
184	81	233	184	81
116	43	210	116	43
138	25	123	138	25
184	81	233	184	81

$$(138 + 25 + 123 + 184 + 81 + 233 + 116 + 43 + 210) (1/9) = 128$$



Class convolution practice: Pixel (0, 0)

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

128	?	?
?	?	?
?	?	?



Class convolution practice: Pixel (0, 1)

138	25	123	138	25
184	81	233	184	81
116	43	210	116	43
138	25	123	138	25
184	81	233	184	81

$$(25 + 123 + 138 + 81 + 233 + 184 + 43 + 210 + 116) (1/9) = 128$$



Class convolution practice: Pixel (0, 1)

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

128	128	?
?	?	?
?	?	?



Class convolution practice: Pixel (0, 2)

138	25	123	138	25
184	81	233	184	81
116	43	210	116	43
138	25	123	138	25
184	81	233	184	81

$$(123 + 138 + 25 + 233 + 184 + 81 + 210 + 116 + 43) (1/9) = 128$$



Class convolution practice: Pixel (0, 2)

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

128	128	128
?	?	?
?	?	?



Class convolution answer

Input

81	233	184
43	210	116
25	123	138

*

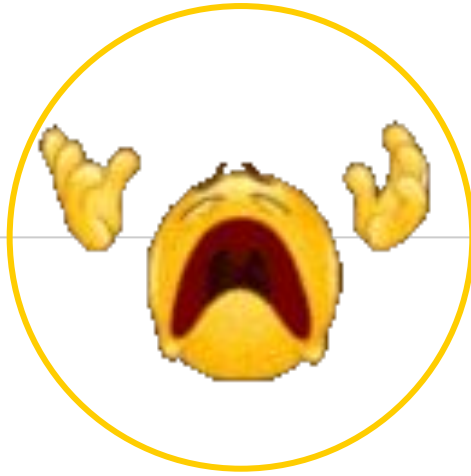
Kernel

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

=

Output

128	128	128
128	128	128
128	128	128



Don't fall victim to **false patterns!**

Make sure to be careful and thorough in your calculations.
One mistake can mess up the whole convolution...

Roll for **confidence!**





Pair convolution practice

Evaluate the following convolution using *zero padding* for the borders.

Protip: You and your partner can split the work and start at opposite ends

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

?	?	?
?	?	?
?	?	?



Pair convolution answer

Evaluate the following convolution using *zero padding* for the borders.

Protip: You and your partner can split the work and start at opposite ends

Input

81	233	184
43	210	116
25	123	138

*

Kernel

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

=

Output

63	96	83
79	128	112
45	73	65



Solo convolution practice

Evaluate the following convolution using *zero padding* for the borders.

Input

207	42	217
30	86	160
170	238	0

*

Kernel

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

=

Output

?	?	?
?	?	?
?	?	?



Solo convolution answer

Evaluate the following convolution using *zero padding* for the borders.

Input

207	42	217
30	86	160
170	238	0

*

Kernel

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

=

Output

50	116	70
125	195	117
74	101	74

Roll for **confidence!**





Homework

02_homework on **Google Classroom** ([link](#))

Solve one 2D convolution problems similar to the ones we did in class.
Don't forget to flip the kernel.