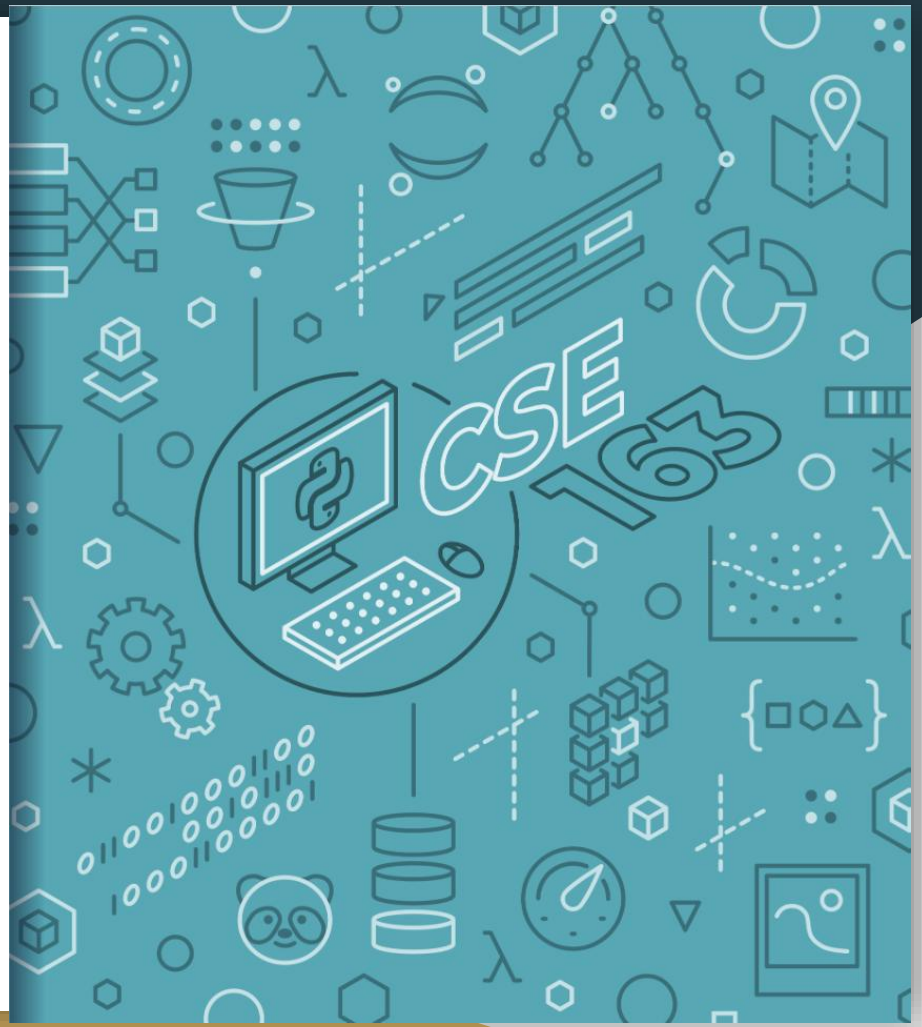
CSE 163
Convolutions

Adrian Salguero
Spring 2026

💭 Icebreaker (discuss with neighbors):
What is your favorite movie?
Add to our Slido!

slido.com
#cse163

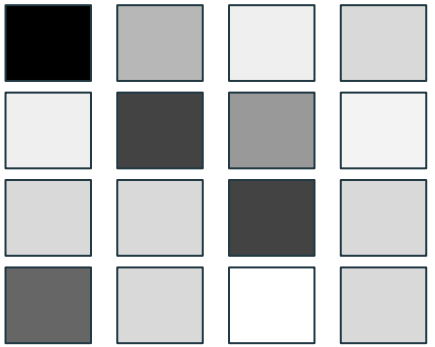


Announcements

- **Take Home Assessment 5: Mapping** due Tuesday May 26th at 11:59pm
- **Reading Assignment 4** is due on Gradescope tonight at 11:59pm!
  - Just include your two annotations/comments in the Gradescope submission to receive credit for this assignment
- **Checkpoint 4** due Monday, May 18th at 11:59pm!
- **Lesson 20 Canvas Quiz** due tonight at 11:59pm!
- **Project Part 3** will be released on Monday!
  - Final Project deliverables!
- **Resubmission Cycle 5** will be released later today, due Tuesday May 19th at 11:59pm
  - Eligible assignments: THA 3 and THA 4
  - Only technical component is eligible for resubmission

Images as Matrices

Grey-scale images can be represented as matrices.



Grey-scale: 255

Grey-scale: 0

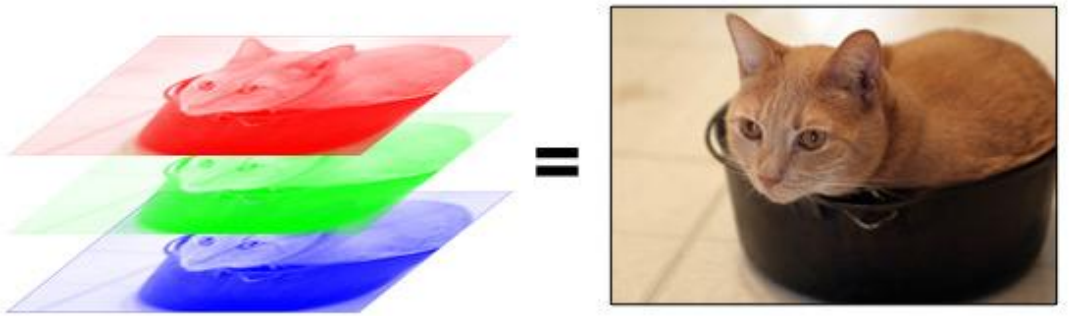
```
data = iio.imread('...')

data[rows, columns] = #
```

Color Images

When you overlap each color channel, it creates a picture we are used to seeing.

- Pixels on your monitor let out specified R/G/B light



```
data[rows, columns, channels] = #
```

Convolution

- When wanting to use “local” information, we commonly use a sliding window approach (i.e. a **convolution**)
- Move the sliding window across the image, and compute the sum of the element wise product of the window (kernel) and image

Image

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Kernel

0	1	2
2	2	0
0	1	2

Convolution Example

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Image

9	8	7
6	5	4
3	2	1

Kernel

2	1

Image

9	8	7
6	5	4
3	2	1

Kernel

2	1

Output

Image

9 2	8 1	7
6	5	4
3	2	1

Kernel

2	1

Output

Image

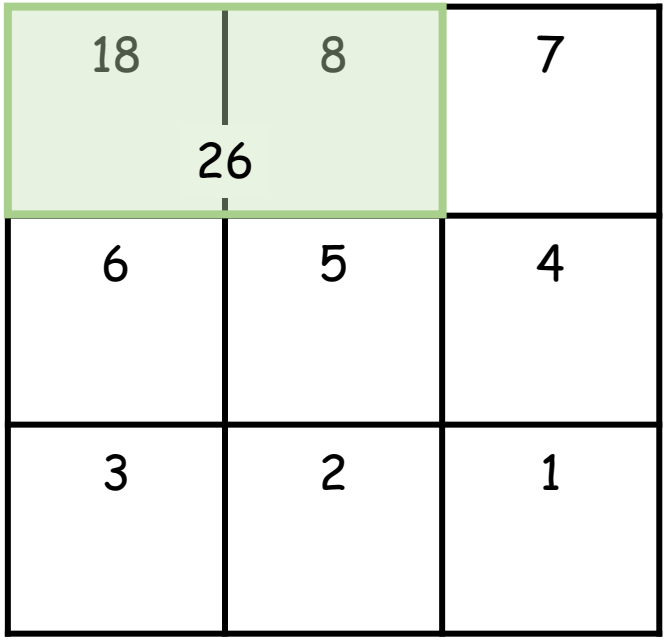
18	8	7
6	5	4
3	2	1

Kernel

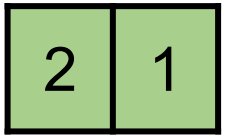
2	1

Output

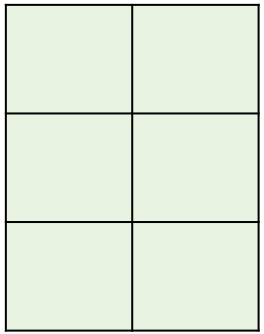
Image



Kernel



Output



Image

18	8	7
6	5	4
3	2	1

Kernel

2	1

Output

26	

Image

9	8 2	7 1
6	5	4
3	2	1

Kernel

2	1

Output

26	

Image

9	16	7
6	5	4
3	2	1

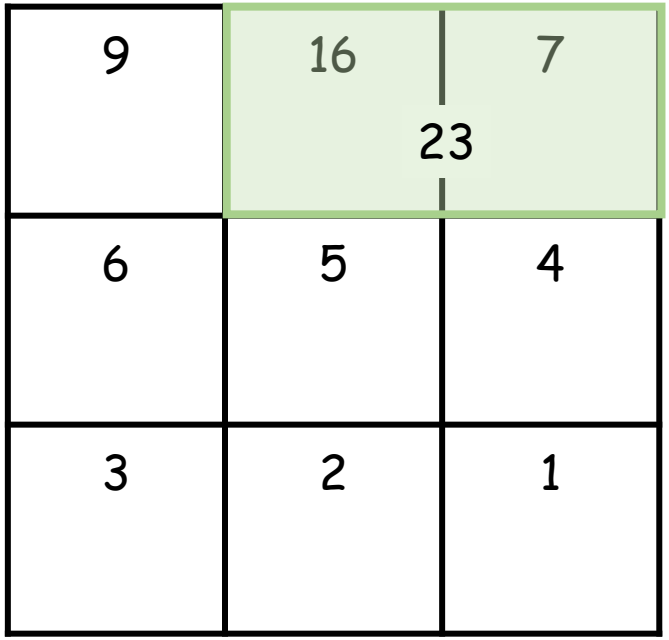
Kernel

2	1

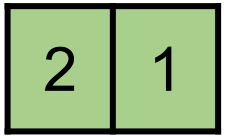
Output

26	

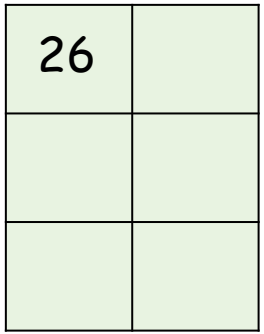
Image



Kernel



Output



Image

9	16	7
6	5	4
3	2	1

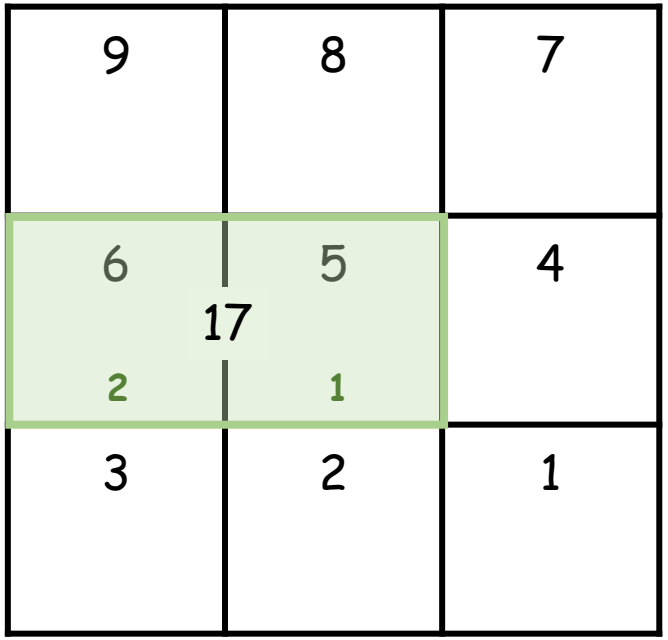
Kernel

2	1

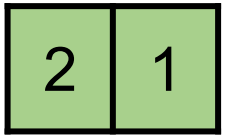
Output

26	23

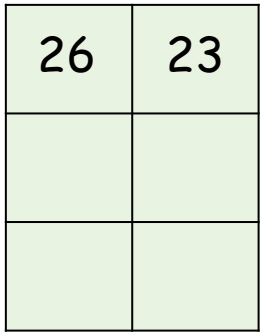
Image



Kernel



Output



Image

9	8	7
6 2	5 1	4
3	2	1

Kernel

2	1

Output

26	23
17	

Image

9	8	7
6	5 4 14 2 1	
3	2	1

Kernel

2	1

Output

26	23
17	

Image

9	8	7
6	5 2	4 1
3	2	1

Kernel

2	1

Output

26	23
17	14

Image

9	8	7
6	5	4
3 2 8 2 1		1

Kernel

2	1

Output

26	23
17	14

Image

9	8	7
6	5	4
3 2	2 1	1

Kernel

2	1

Output

26	23
17	14
8	

Image

9	8	7
6	5	4
3	2 1 5 2 1	

Kernel

2	1

Output

26	23
17	14
8	

Image

9	8	7
6	5	4
3	2 2	1 1

Kernel

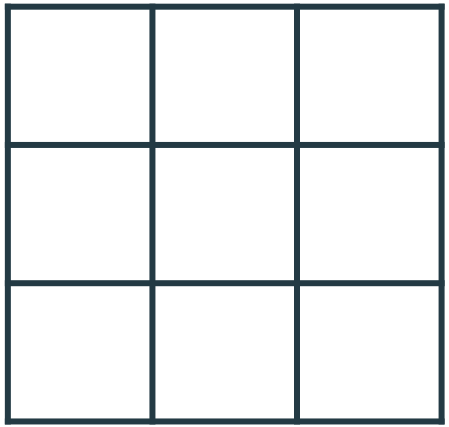
2	1

Output

26	23
17	14
8	5

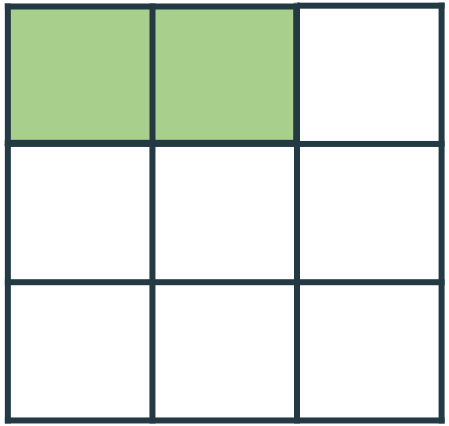
How many times can we fit the kernel?

Image

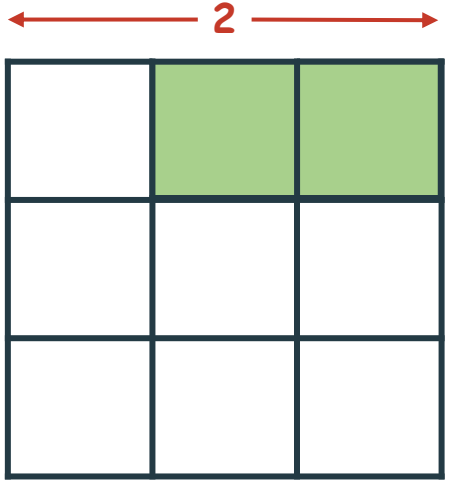


Kernel

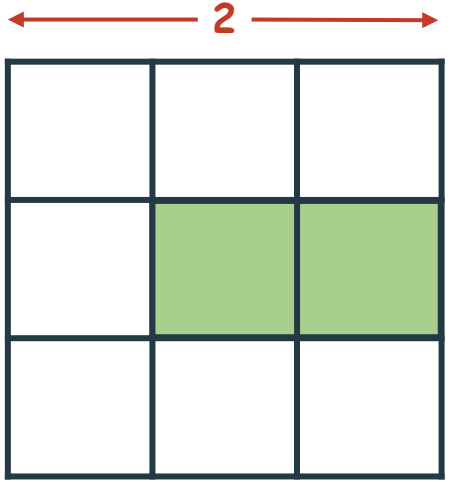
How many times can we fit the kernel?



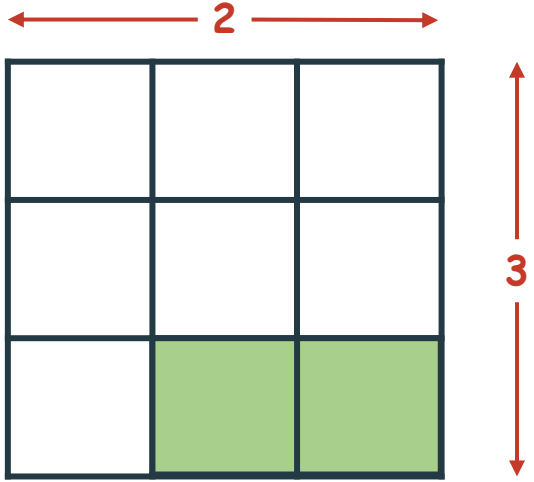
How many times can we fit the kernel?



How many times can we fit the kernel?

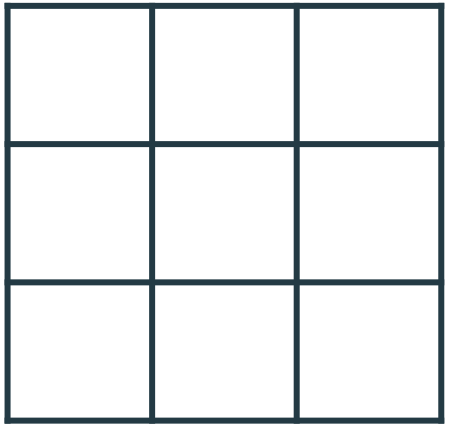


How many times can we fit the kernel?

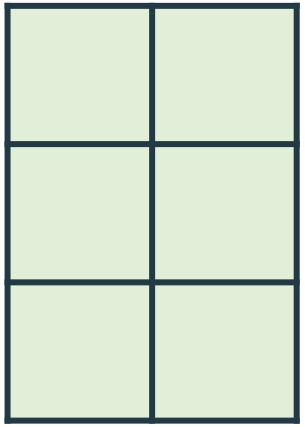


How many times can we fit the kernel?

Image

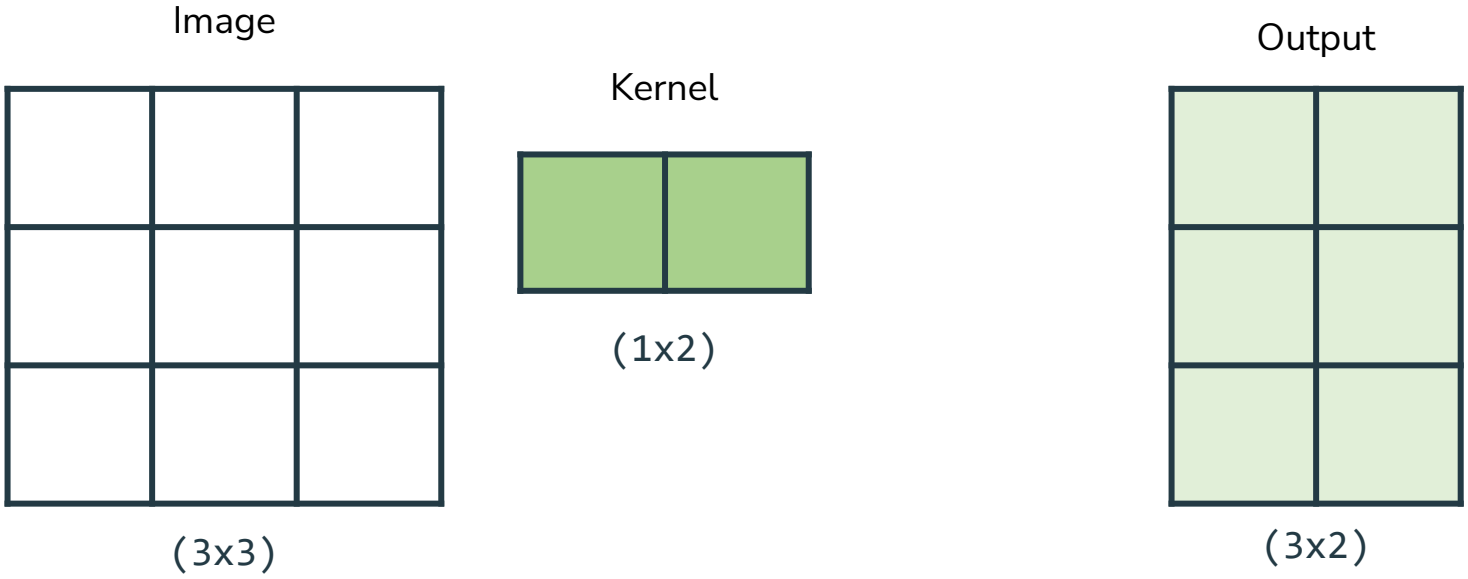


Kernel



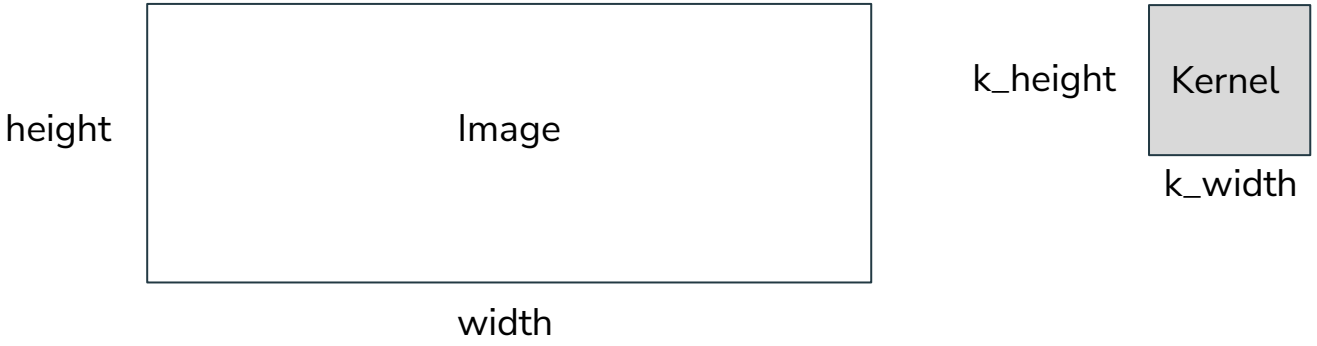
3 x 2

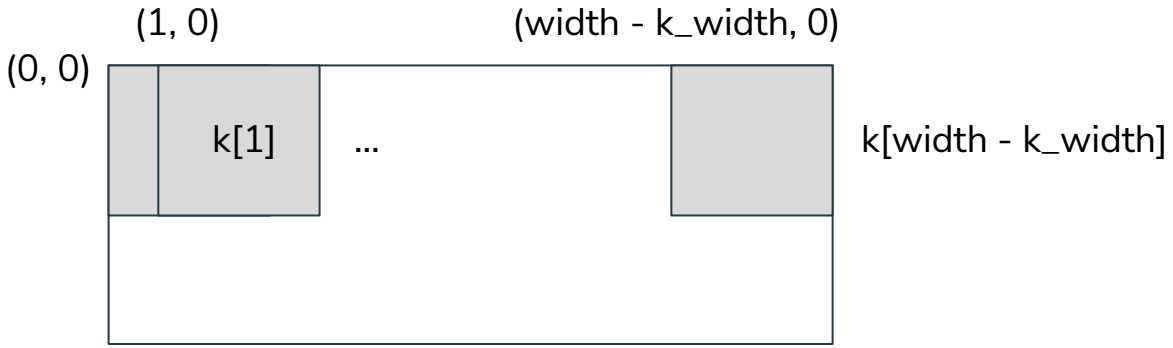
How many times can we fit the kernel?




```
(image_height - k_height + 1, image_width - k_width + 1)
```

Generic Formula for Result Size





Total of (width - k_width + 1) kernels because of zero based indexing!

So the shape of the result will be **(height - k_height + 1, width - k_width + 1)**

Common Kernels

Identity

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Edge Detection

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Sharpen

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

Box Blur

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Image Classification

- For a really long time, image classification was done by painstakingly crafting these features (like edge detectors), by hand.
- This kind of worked, but we quickly hit our peak using this method.
- Then came the buzz-word… deep learning

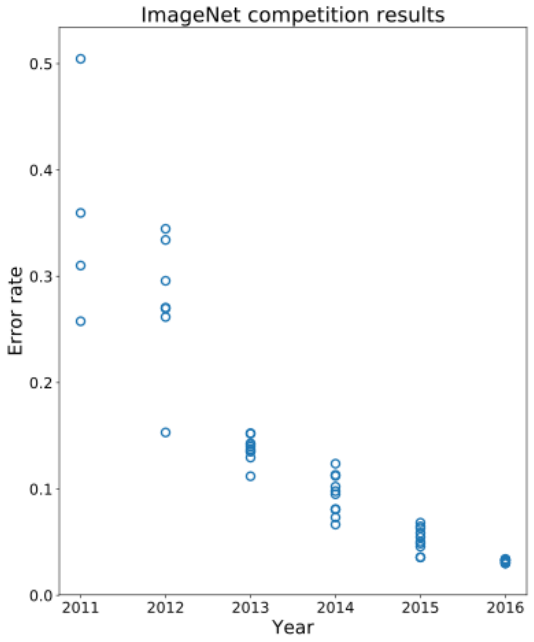


Image Classification

- Is this a solved problem?
  - We get pretty decent error rates on challenges like ImageNet
- What we can't do
  - Sometimes can't generalize to other real-world datasets
  - Adversarial attacks