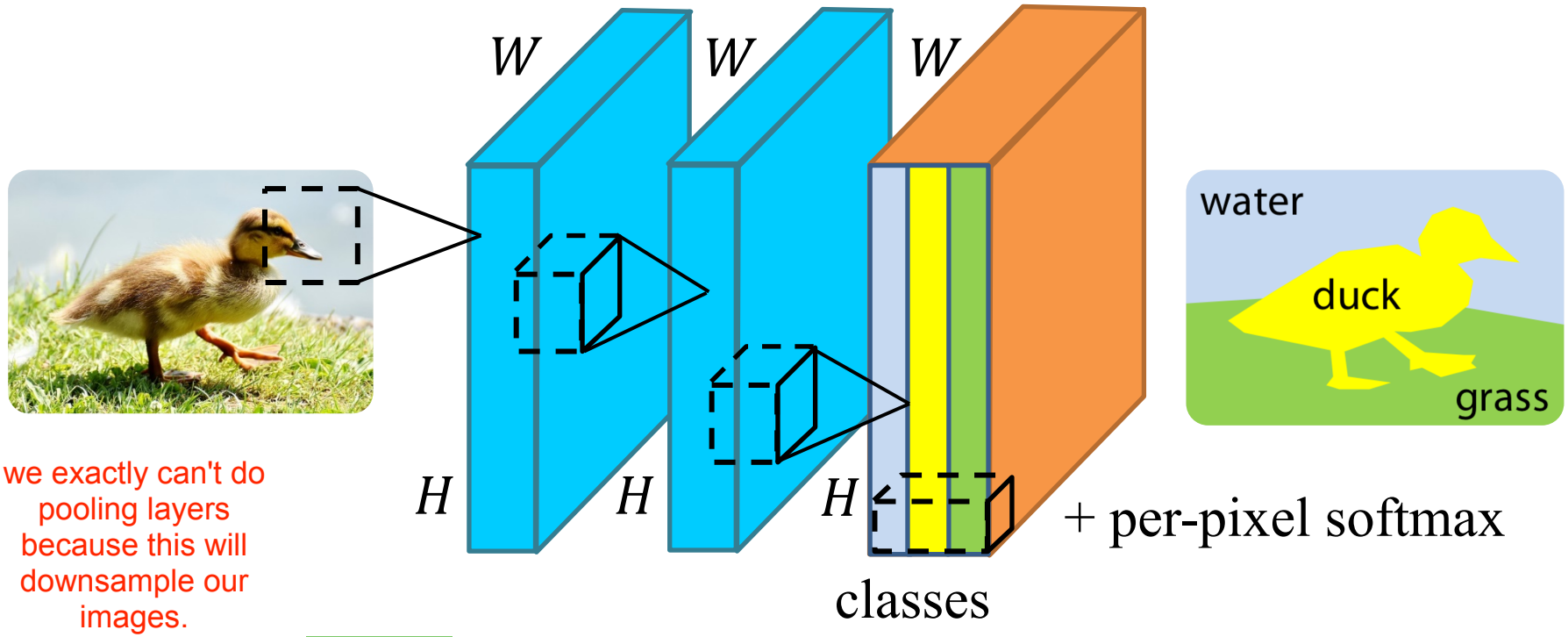
Semantic segmentation

We need to classify each pixel

image classification + localization.
Not only what is the image, but where we see it.

Semantic segmentation

We need to classify each pixel



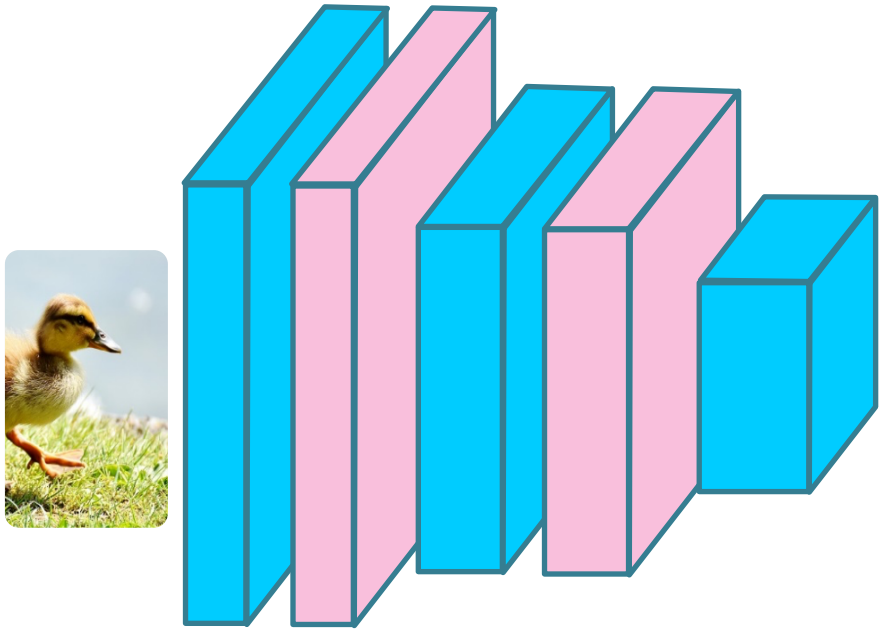
we exactly can't do pooling layers because this will downsample our images.
We want our segmentation to be crisp, not low quality.

Naïve approach: stack convolutional layers and add per-pixel softmax

We go deep but don't add pooling, too expensive

Semantic segmentation

Let's add pooling, which acts like down-sampling



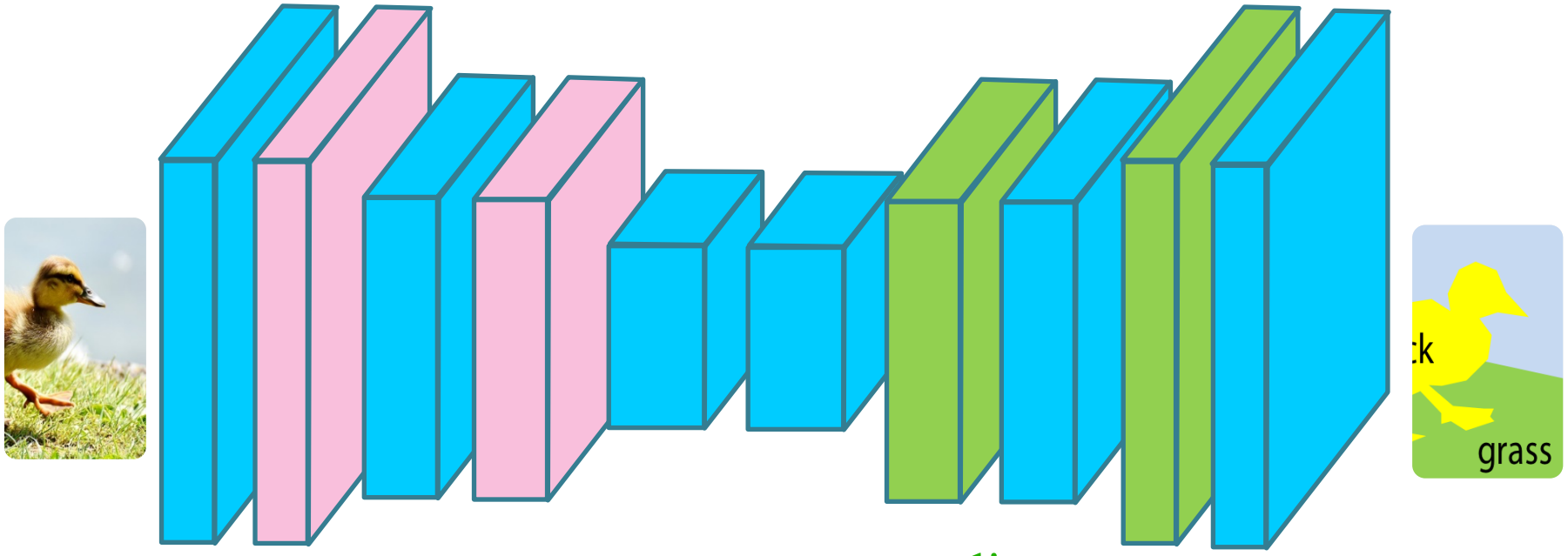
Wait a second!
We need to classify each pixel!

Need to do **un**pooling!

Yes. so we do the 'reverse'. Upsampling.

Semantic segmentation

Let's add pooling, which acts like down-sampling



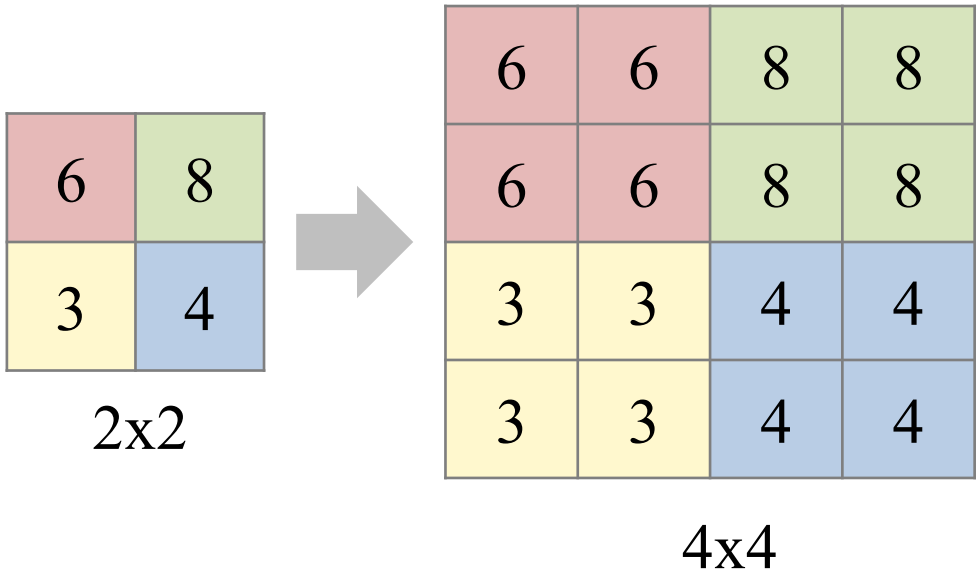
up-sampling

up-sampling

But how do we do up-sampling?

Nearest neighbor unpooling

Fill with nearest neighbor values

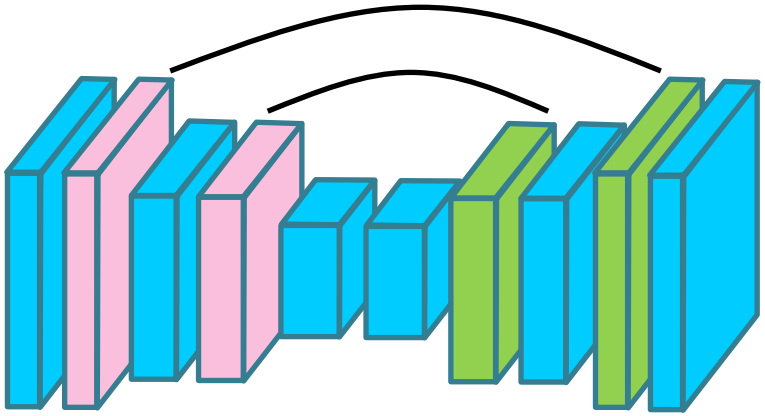


Pixelated and not crisp!

Can't just use this.

Max unpooling

Corresponding pairs of downsampling and upsampling layers



Remember which element was max during pooling, and fill that position during unpooling:

ooo. interesting.
So we remember the POSITION
of our previous maxpools.

1	2	4	3
2	5	2	0
1	5	2	1
7	3	3	8

4x4

5	4
7	8

2x2

…

1	2
3	4

2x2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

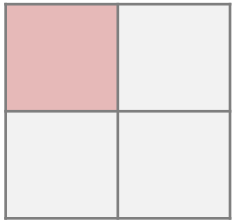
4x4

But there is something better.

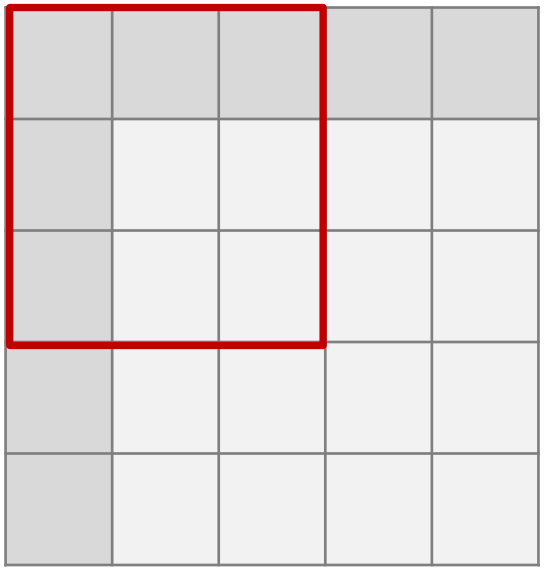
Learnable unpooling

- Previous approaches are not data-driven!
- We can replace max pooling layer with convolutional layer that has a bigger stride!
- What if we can apply convolutions to do unpooling?

Input: 2x2



Input gives weight for filter

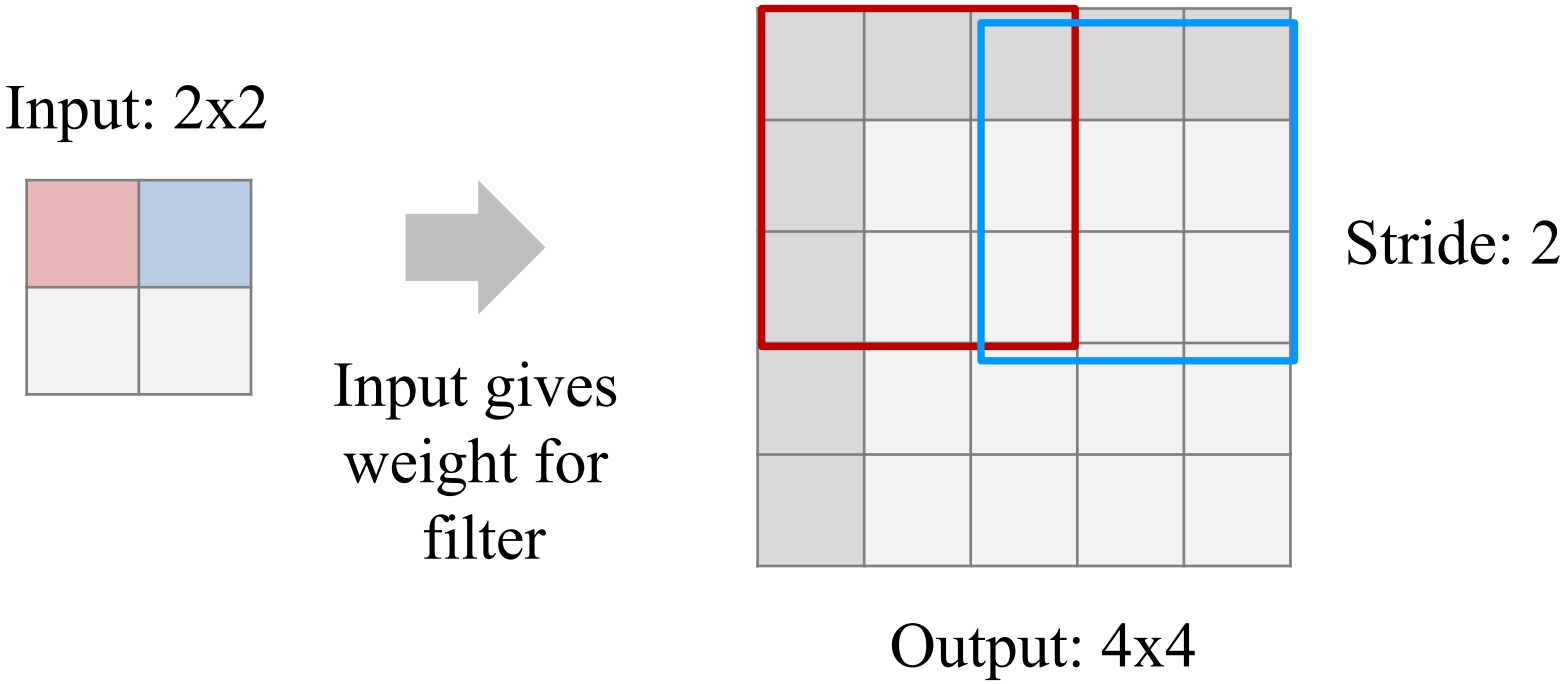


Output: 4x4

The step is to take the value from the red input cell, and then multiply it with the kernel's values to give the output in the red outlined box in the output 4x4.

Learnable unpooling

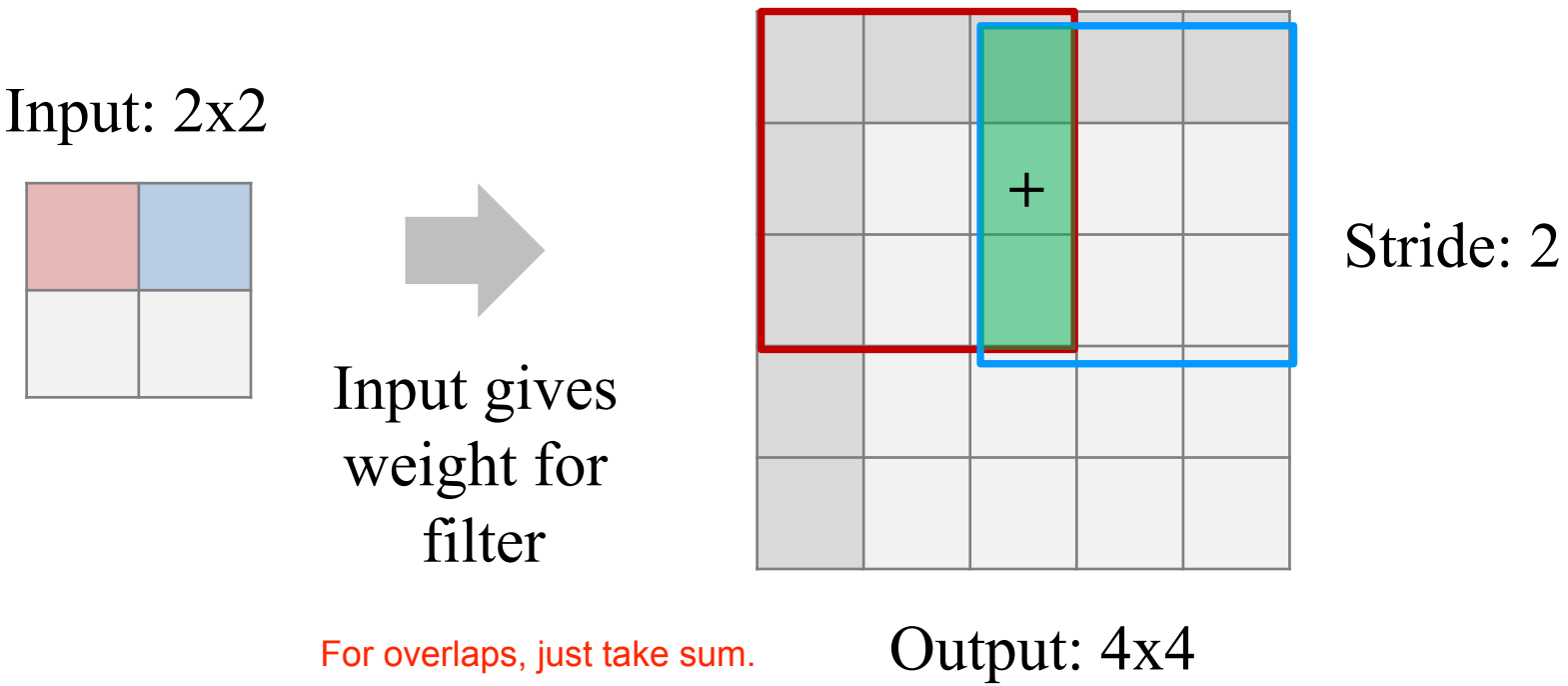
- Previous approaches are not data-driven!
- We can replace max pooling layer with convolutional layer that has a bigger stride!
- What if we can apply convolutions to do unpooling?



We multiply blue input cell value with the same kernel for the blue outline box in 4x4 output.

Learnable unpooling

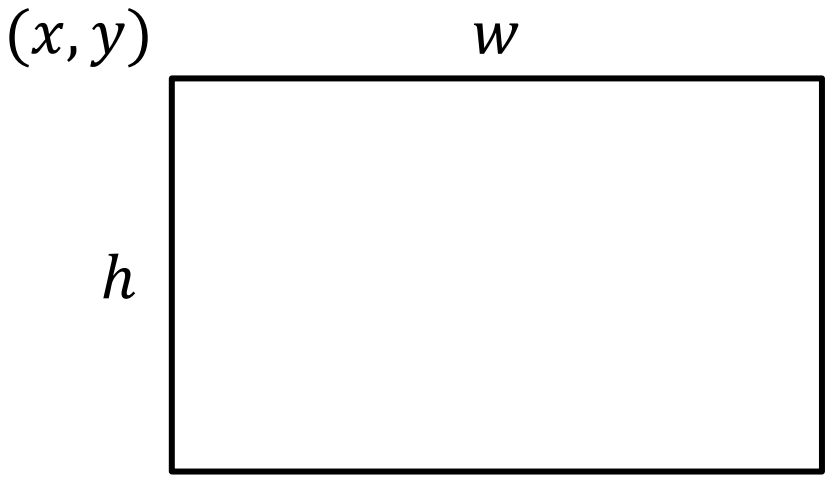
- Previous approaches are not data-driven!
- We can replace max pooling layer with convolutional layer that has a bigger stride!
- What if we can apply convolutions to do unpooling?



This is called TRANSPOSED convolution. It's not the best idea actually, see https://distill.pub/2016/deconv-checkerboard/

Object classification + localization

We need to find a bounding box to localize an object.

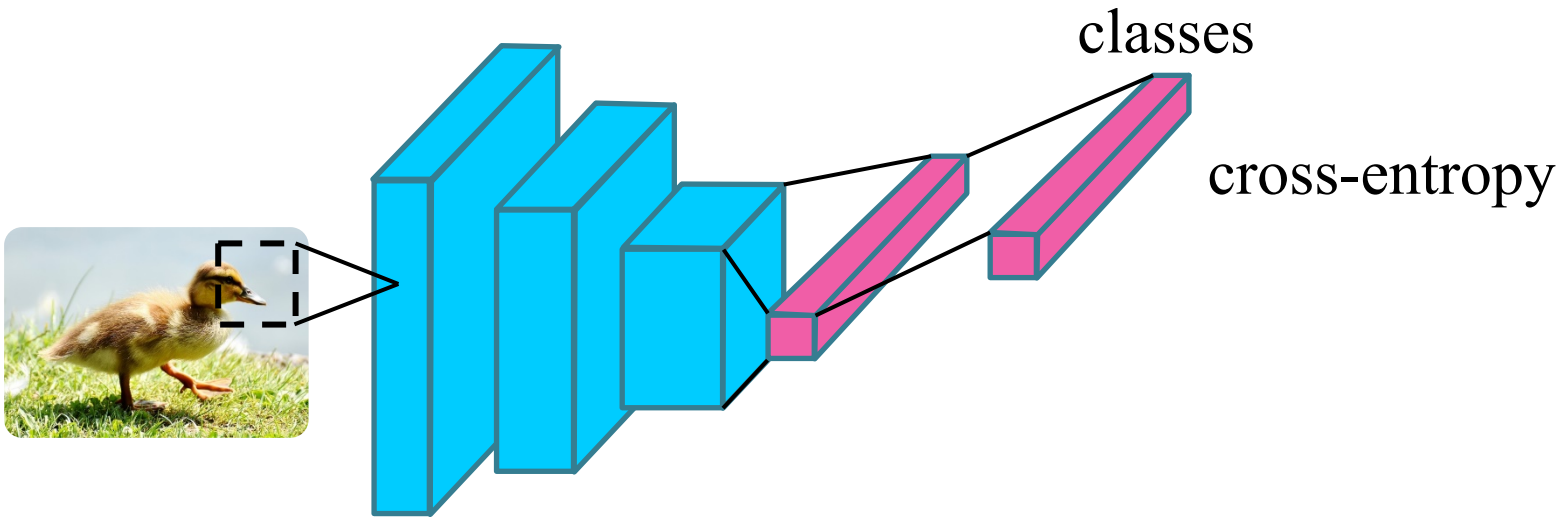


We will use regression for (x, y, w, h)!

(x,y) is the coordinate of the upper left corner of box.
w,h is width and height.

Object classification + localization

Classification network:

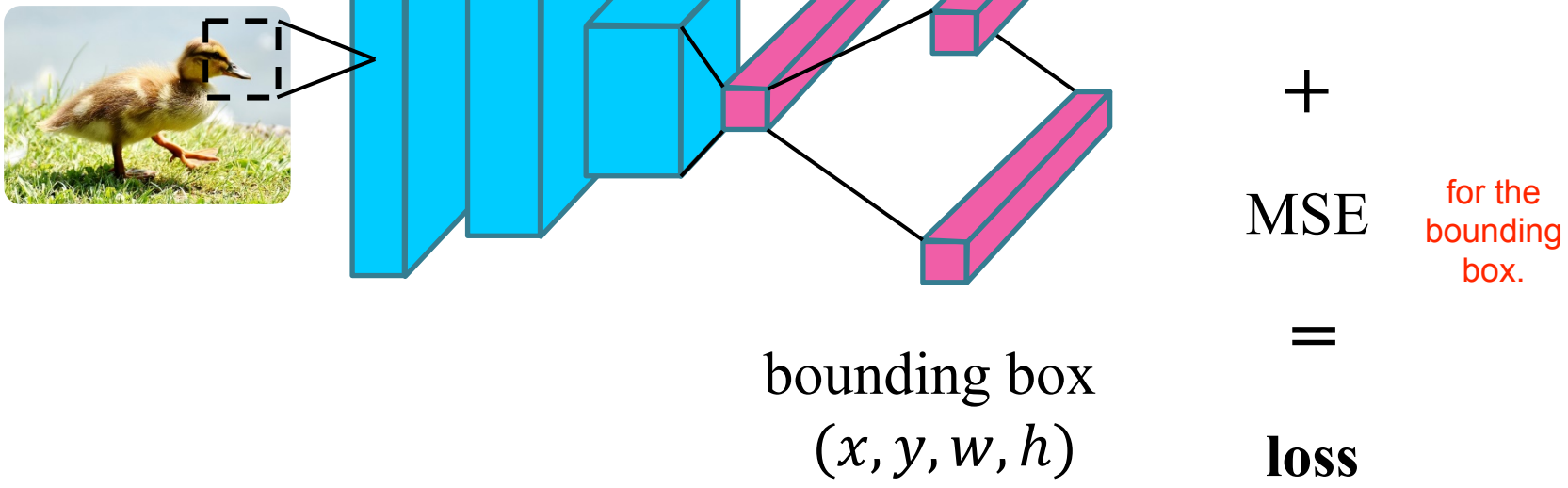


Do we need a second network?

No! Just use the same feature extractors, and train a smaller network on top of it.

Object classification + localization

Classification + localization network:



Summary

- In this video we took a sneak peek into other computer vision problems that successfully utilize convolutional neural networks.
- This video concludes our introduction to neural networks for images!