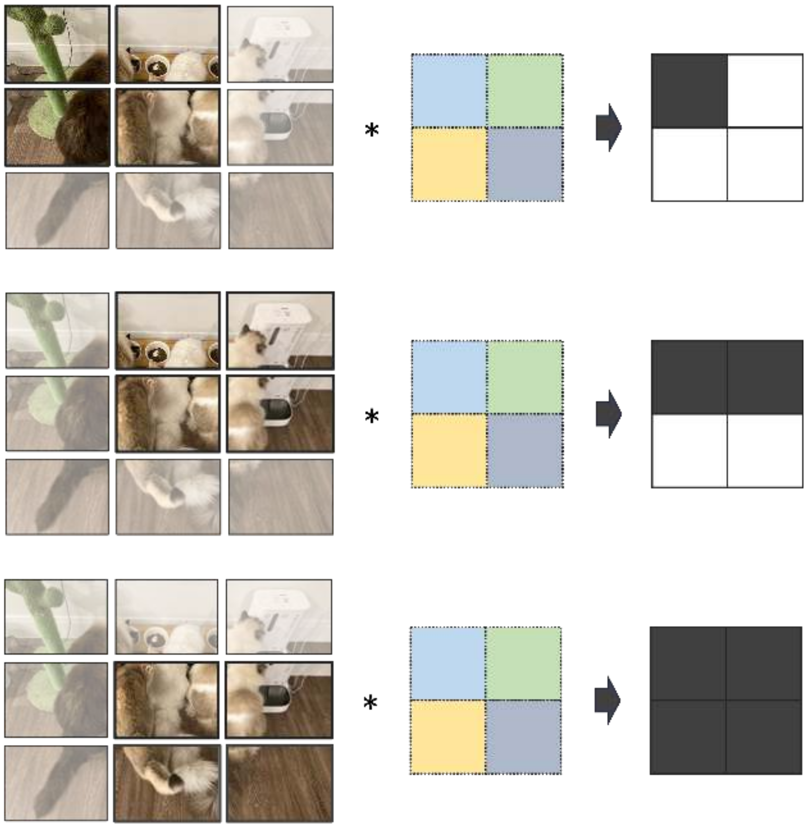
Structure of Convolutions

Recap: Convolution

Convolution is a spatially anchored linear operation

- Fast, memory-efficient
- Preserves image structures

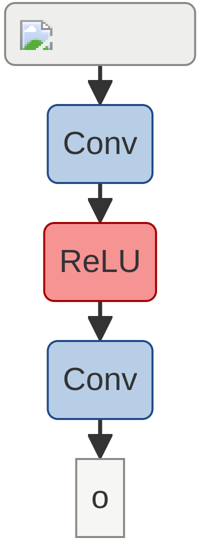


Convolutional Network

Alternate

- Convolution
- Non-linearity
- Normalization and residuals for deeper networks

Issue 1: Vanilla convolution shrinks inputs



Convolution Output Size

Output: $y \in \mathbb{R}^{C_2 \times (H-h+1) \times (W-w+1)}$

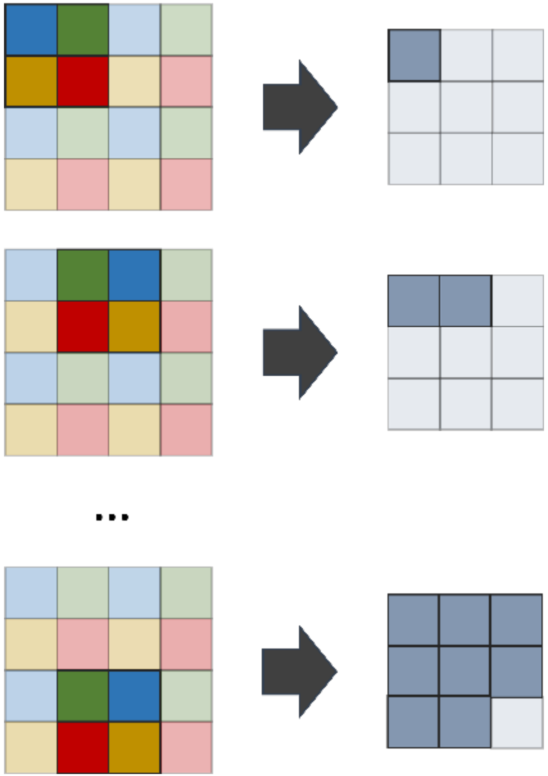
Output size depends on

- Input size: $H \times W$
- Kernel size: $h \times w$

A simple example:

- 4×4 input
- 2×2 kernel $\Rightarrow 3 \times 3$ output

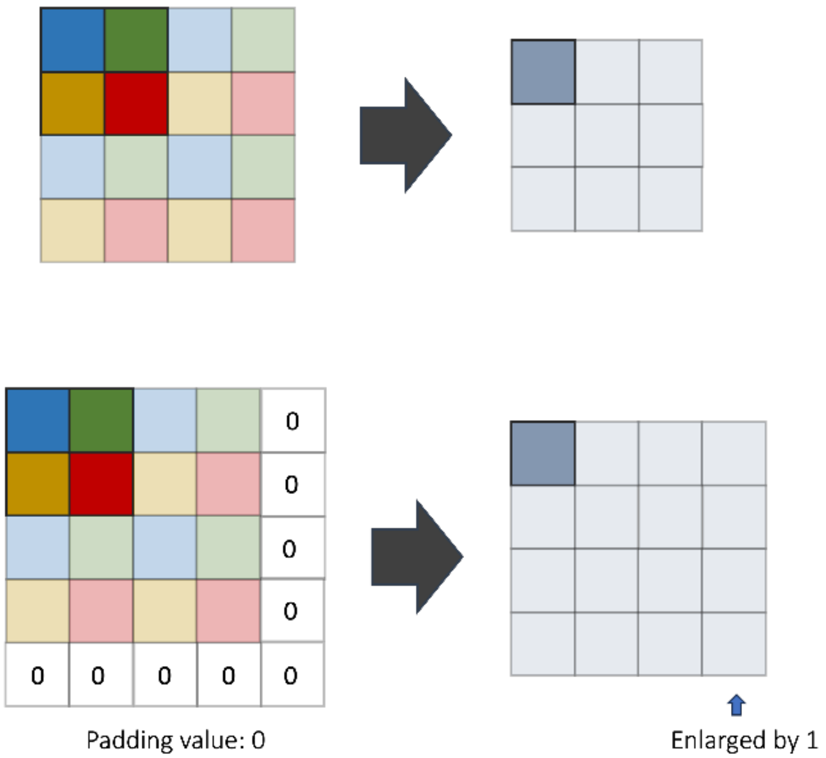
Output size shrinks down with convolutions!



Convolution With Padding

Padding: Pad the *input* to match the output size

- Pad the input with a constant (e.g., 0)
- Output size grows by 1

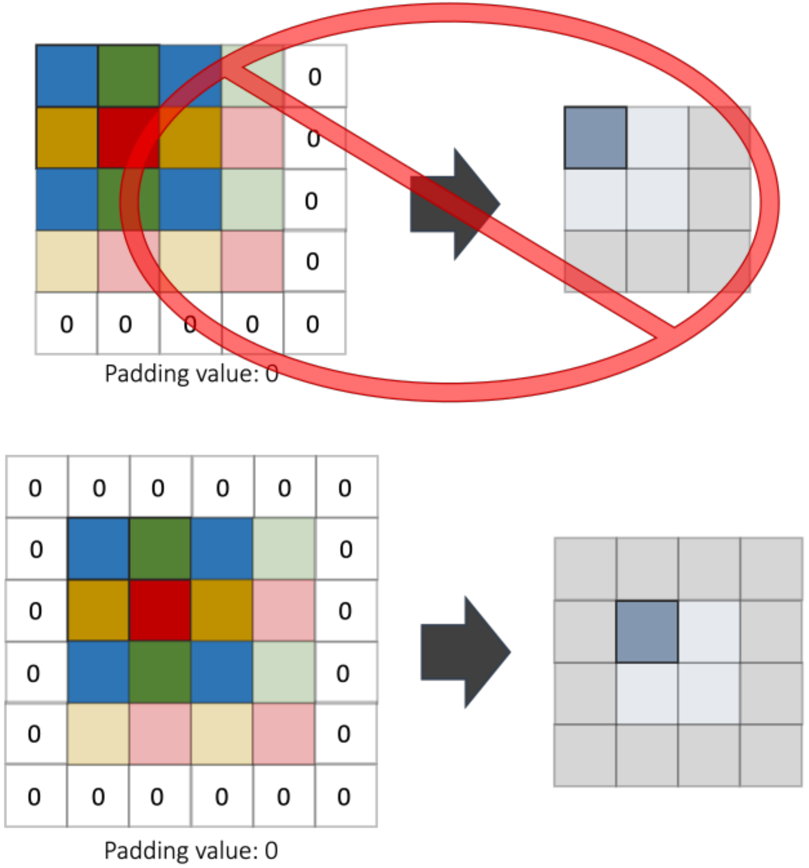


Convolution With Padding

What happens if the image is padded only on one-end?

- Image context shift! (gray: padded area)

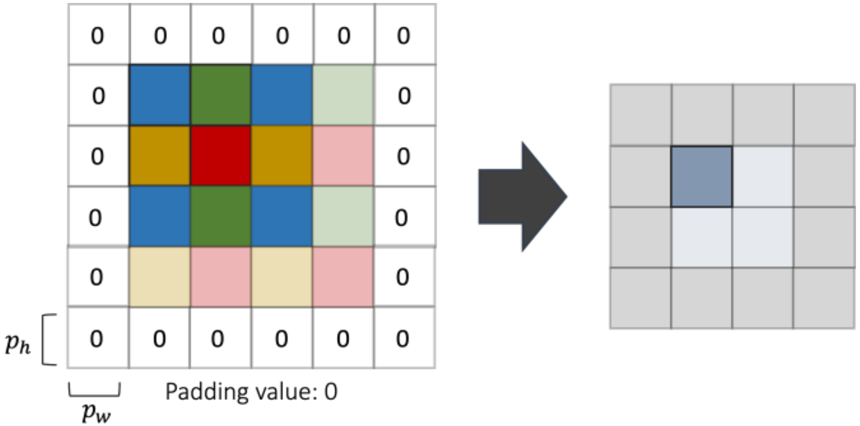
Solution: Always pad symmetrically



Convolution With Padding

Add p_w and p_h of a value c in each dimension

Input	$C_1 \times H \times W$
Padded Input	$C_1 \times (H + 2p_h) \times (W + 2p_w)$
Kernel	$C_1 \times C_2 \times h \times w$
Output	$C_2 \times (H + 2p_h - h + 1) \times (W + 2p_w - w + 1)$



For simplicity: Pad to retain size

- $w = 2p_w + 1, h = 2p_h + 1$

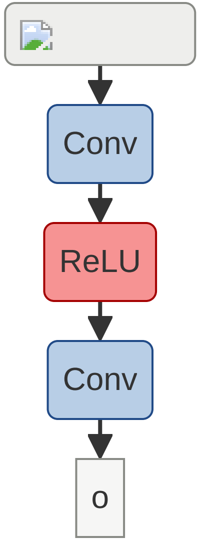
Convolutional Network

Alternate

- Convolution
- Non-linearity
- Normalization and residuals for deeper networks

✓ ~~Issue 1: Vanilla convolution shrinks inputs~~

Issue 2: Vanilla convnets get very slow as C grows



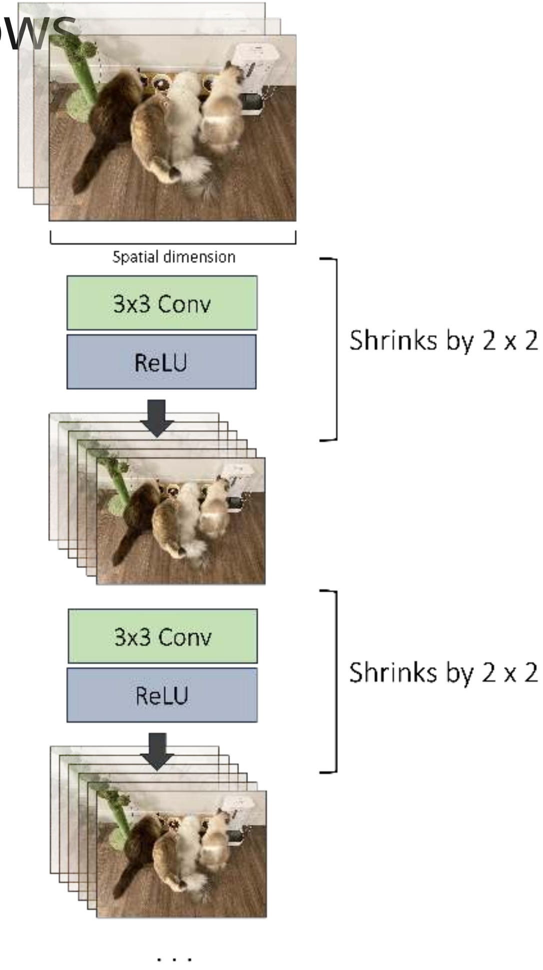
Vanilla convnets get very slow as C grows

Computational cost of convolution

$$O(WHwhC_1C_2)$$

Solution: Shrink W and H

- Shrink W and H
- Increase C



Convolution With Stride

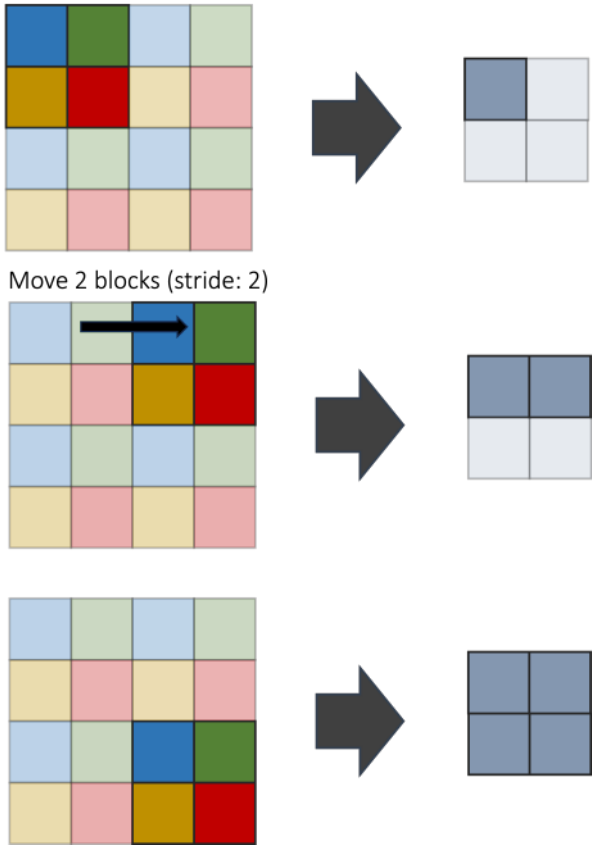
Only compute every s_w / s_h-th output

- Skip (do not compute) outputs

Input	$C_1 \times H \times W$
Kernel	$C_1 \times C_2 \times h \times w$
Output	$C_2 \times \left(\left\lfloor \frac{H-h+2p_h}{s_h} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{W-w+2p_w}{s_w} \right\rfloor + 1\right)$

Advantages

- Computational efficiency later layers
- Larger receptive field later layers

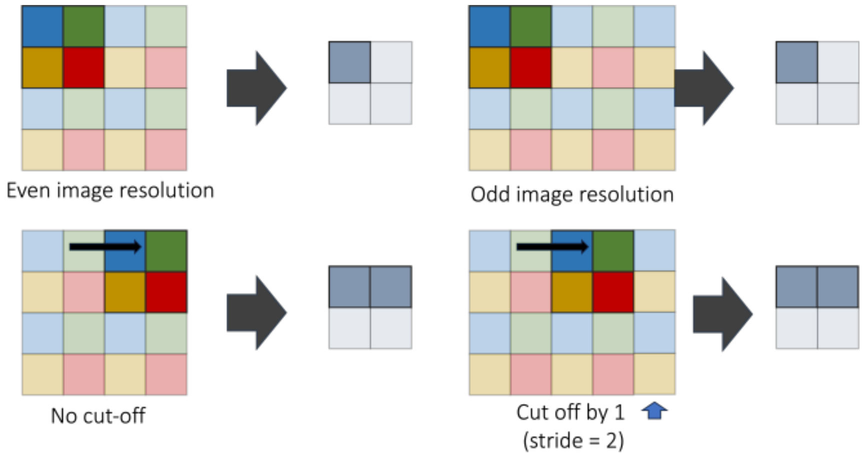


Stride and Rounding

Output size: $\left(\left\lfloor\frac{H-h+2p_h}{s_h}\right\rfloor+1\right)\times\left(\left\lfloor\frac{W-w+2p_w}{s_w}\right\rfloor+1\right)$

What if $W-w+2p_w$ is not divisible by s_w?

- We round down
- Cut content



Convolutional Network

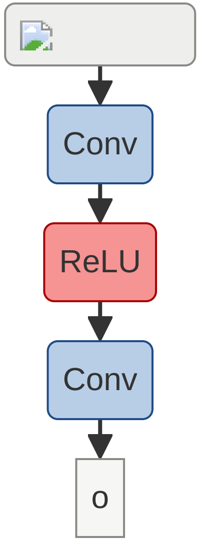
Alternate

- Convolution
- Non-linearity
- Normalization and residuals for deeper networks

✓ ~~Issue 1: Vanilla convolution shrinks inputs~~

✓ ~~Issue 2: Vanilla convnets get very slow as C grows~~

What is my CNN is still too slow?



Group Convolution

Computational cost of convolution

$$O(WHwhC_1C_2)$$

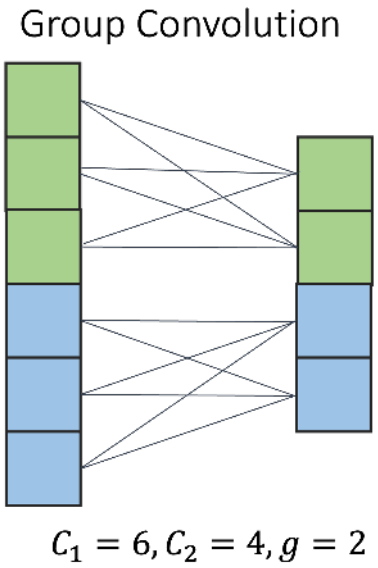
More efficient type of convolution

- Splits channels into g groups

$$O(WHwhC_1C_2) \to O\left(\frac{WHwhC_1C_2}{g}\right)$$

Reduced computation cost

$$O(C_1 \cdot C_2) \to O(C_1 \cdot C_2/g)$$



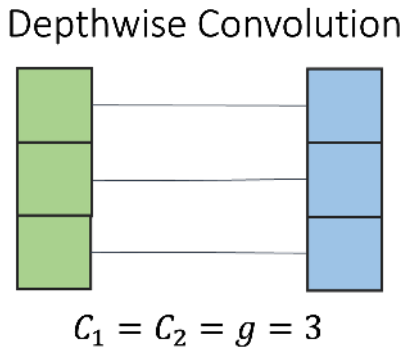
Special Convolution: **Depthwise Convolution**

An extreme case

- Special grouping where $C_1 = C_2 = g$
- Reduce computation cost:

$$O(WHwhC_1C_2) \to O\left(WHwhC_1\right)$$

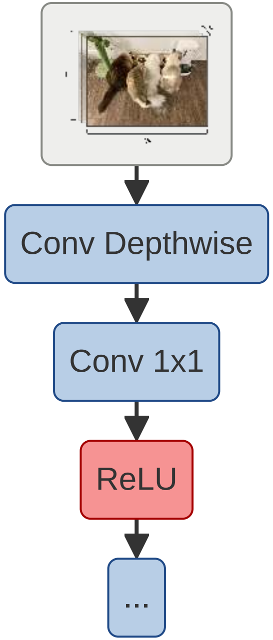
- What cloud go wrong?



Special Convolution: **Depthwise Convolution**

Depthwise convolution cannot communicate across channels.

- Example: RGB image
- No reasoning about colors, just red-ness, green-ness, blue-ness



Solution: Always add 1×1 convolution! [1]

- Depthwise Convolution: $O\left(WHwhC_1\right)$
- 1x1 Convolution: $O\left(WHC_1C_2\right)$

1. Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv:1704.04861 (2017). ↩

Hyperparameters of Convolutions

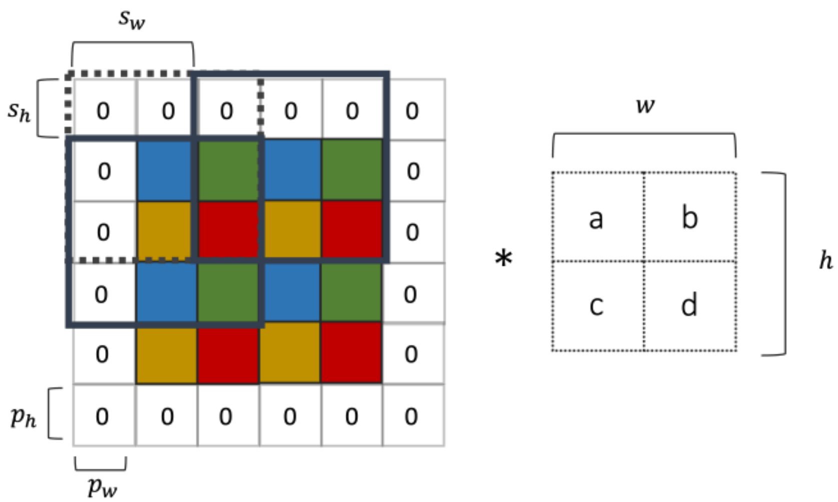
Output channels: C_2

Groups: g

Kernel size: w and h

Padding: p_w and p_h

Stride: s_w and s_h



Hyperparameters of Convolutions - The Illusion of Choice

Stride: $s_w = s_h \in \{1, 2\}$

Kernel size: $w = h \in \{1, 3\}$ (mostly)

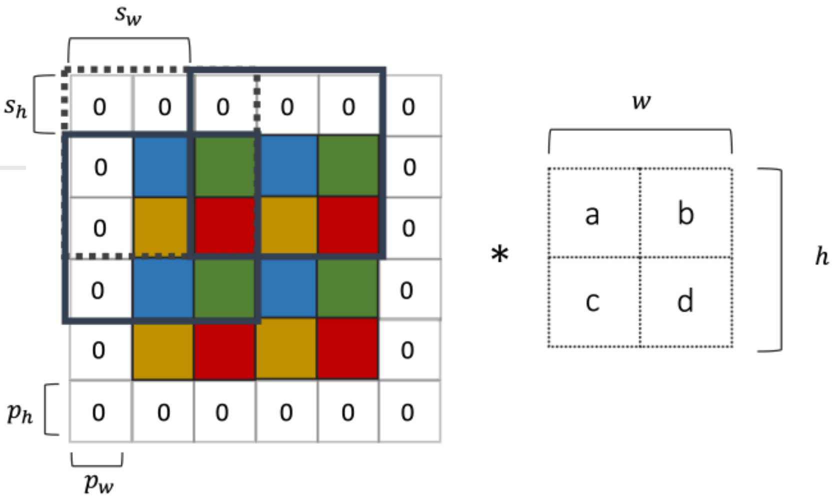
Output channels: $C_2 = C_1 s_w = C_1 s_h$ (powers of 2)

- Keeps computation constant $O(WHwhC_1C_2)$

Groups: g (expert use only. Ignore.)

Padding: $p_w = \frac{w-1}{2}$ and $p_h = \frac{h-1}{2}$

- Do not change activation size



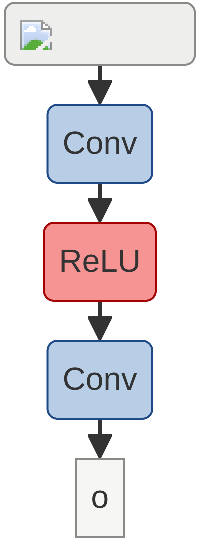
Convolutional Network

Alternate

- Convolution
- Non-linearity
- Normalization and residuals for deeper networks

Use stride

- Trade channels for spatial resolution
- Larger receptive field
- More global patterns



Convolutional Operators and Their Structure - TL;DR

Use padding and striding to control output size of convolution layers

Group and **depthwise** convolution reduce computation cost

Depthwise convolution cannot mix information across channels