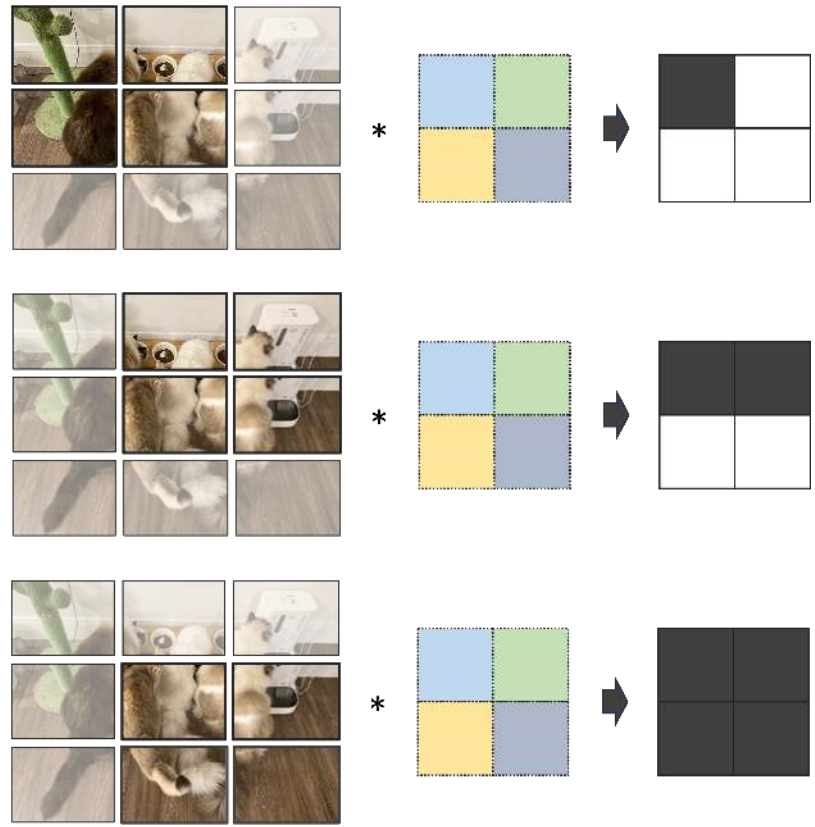
Design Principles of Convolutional Networks

Recap: Convolution

Convolution is a spatially anchored linear operation

- Fast, memory-efficient
- Preserves image structures



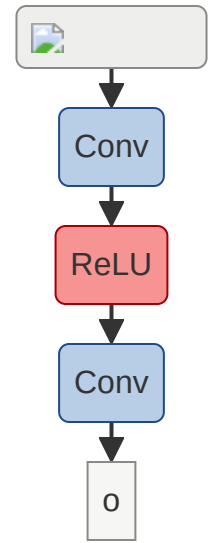
Recap: Convolutional Network

Alternate

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

Use stride

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



Design of Convolutional Networks

After stride

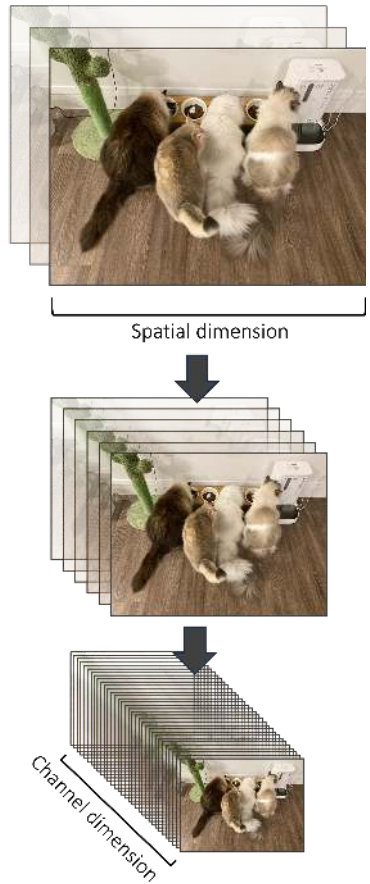
- More Channels (wider network)
- Lower resolution (width / height)

What does that mean for number of activations?

- Total activations shrink by $2\times$ after stride
- $C \times W \times H \to 2C \times \frac{W}{2} \times \frac{H}{2}$

Is this a good idea throughout the network?

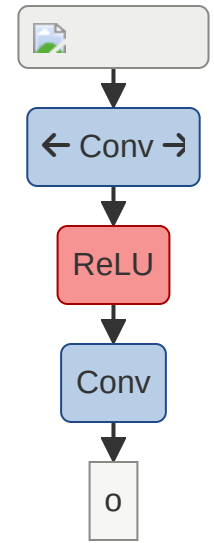
- No. Will lose information every stride
- Solution: Expand dimension in first layer



Design of ConvNets

Wide first layer

- Channels $64 - 96$
- Large kernel size $7 - 16$
- Strided $(2 - 16)$



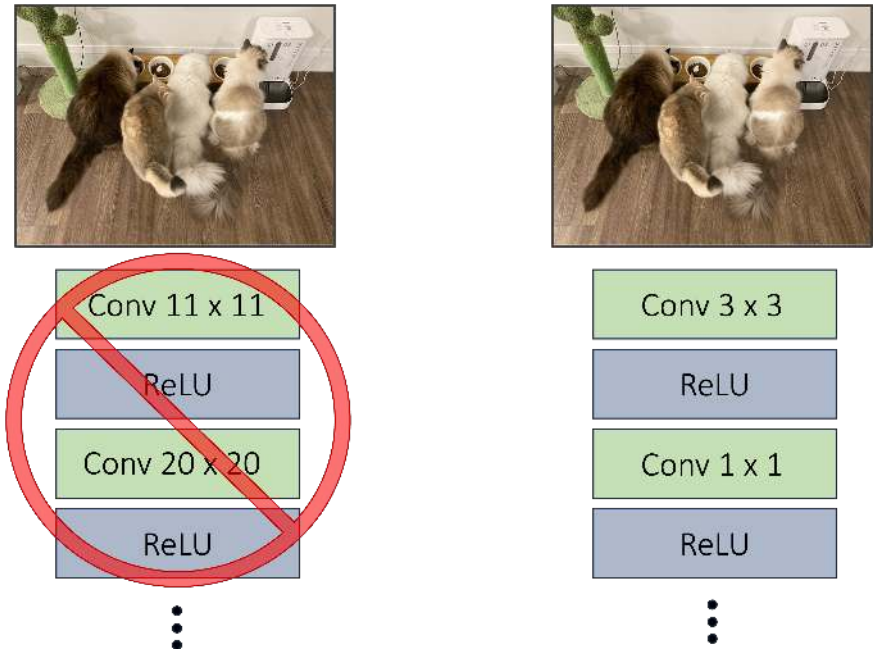
Keep Kernels Small

Use 3×3 or 1×1 (almost) everywhere

- Avoid using large kernels (e.g., 11×11)
- **Exception**: first layer often uses large kernel

Why?

- Saves computation
- More layers in sequence often better



Repeat Patterns

Use repeatable *pattern*

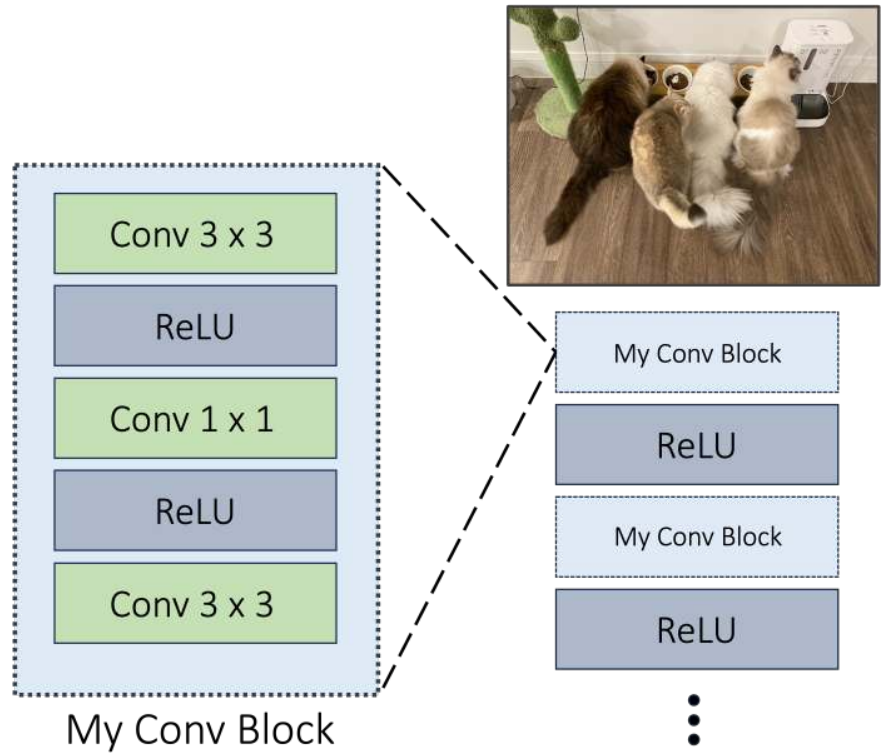
- This is called ***block***.
- Include nonlinearities, normalizations, residuals

Repeat *block* multiple times

- Optionally stride within block
- **Exception**: first layers are different

Why?

- Saves time: developing, debugging, tuning



Make It All Convolutional

Avoid using linear layers

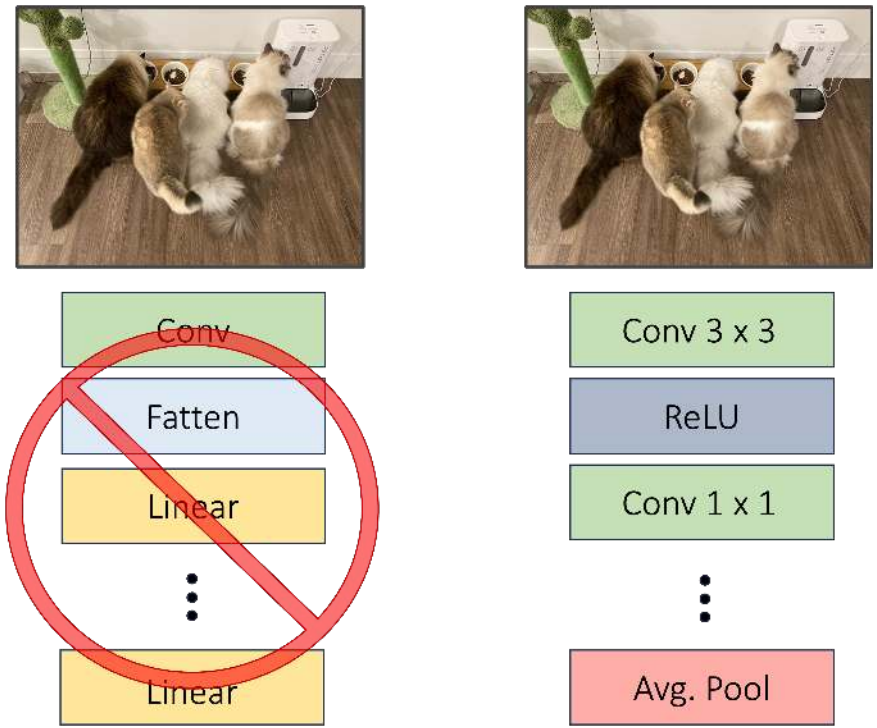
- Too many parameters

Average in the end

- Global average pooling

As many shared parameters as possible

- Better training signal



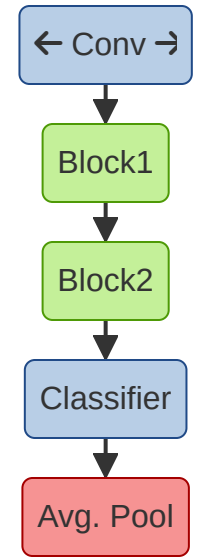
Design of ConvNets

Wide first layer

- Channels $64 - 96$
- Large kernel size $7 - 16$
- Strided $(2 - 16)$

Repeating blocks

- $\leq 3 \times 3$ convolution
- All convolutional



Design Principles of Convolutional Networks - TL;DR

Increase channel dimension and decrease spatial dimensions

Keep kernels small

Repeat patterns