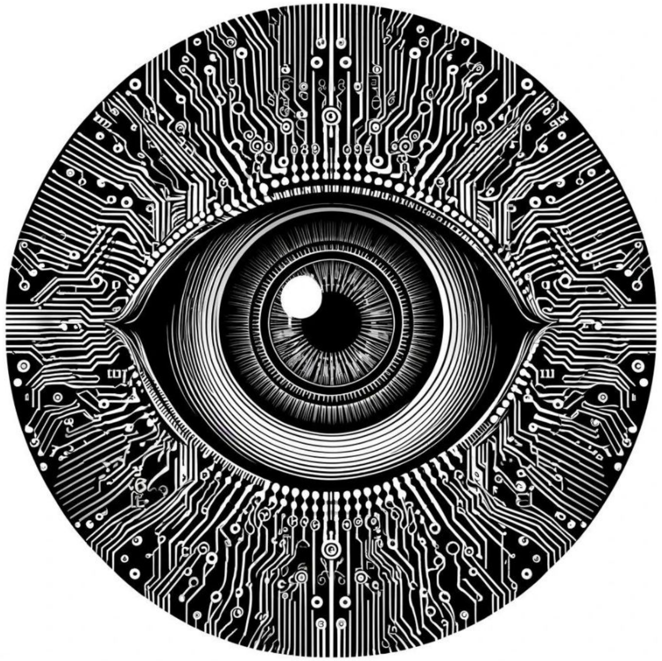
KI Service Zentrum by Hasso-Plattner-Institut HPI

Upsampling and Channel Mixing with Convolutions



Antonio Rueda-Toicen

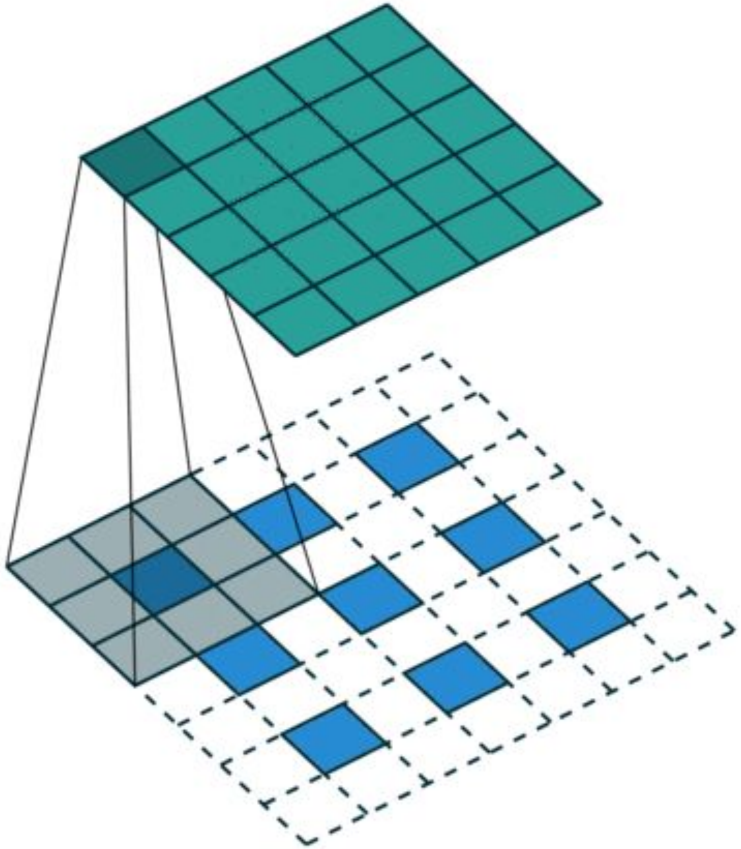
SPONSORED BY THE

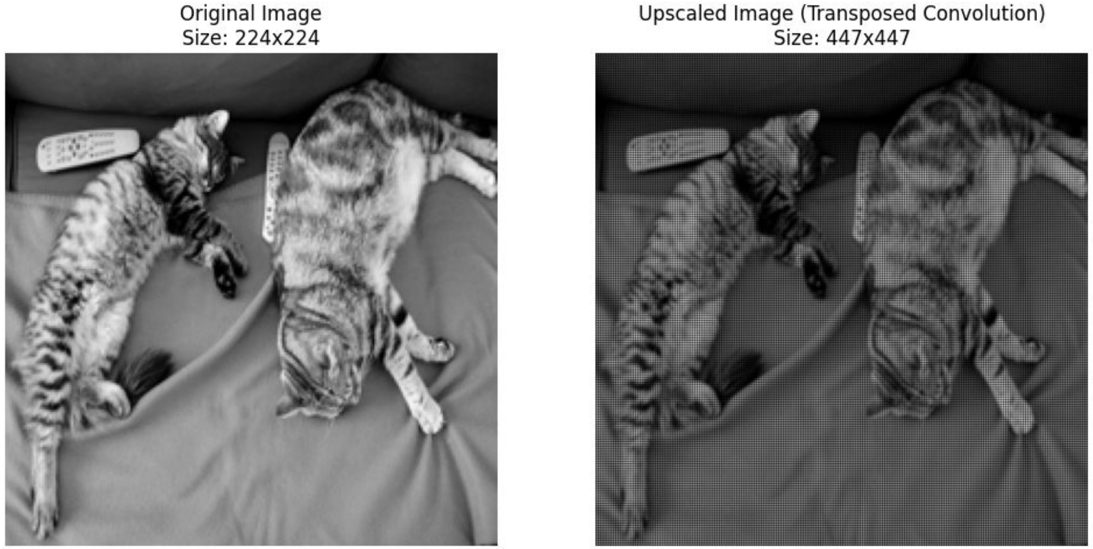
Federal Ministry
of Education
and Research

Learning goals

- Master upsampling and channel mixing with convolutions
- Use bilinear interpolation to resize images
- Understand hypercolumns and feature combination for image generation

Upsampling filters (aka transposed convolution / upconvolution / atrous convolution / deconvolution)



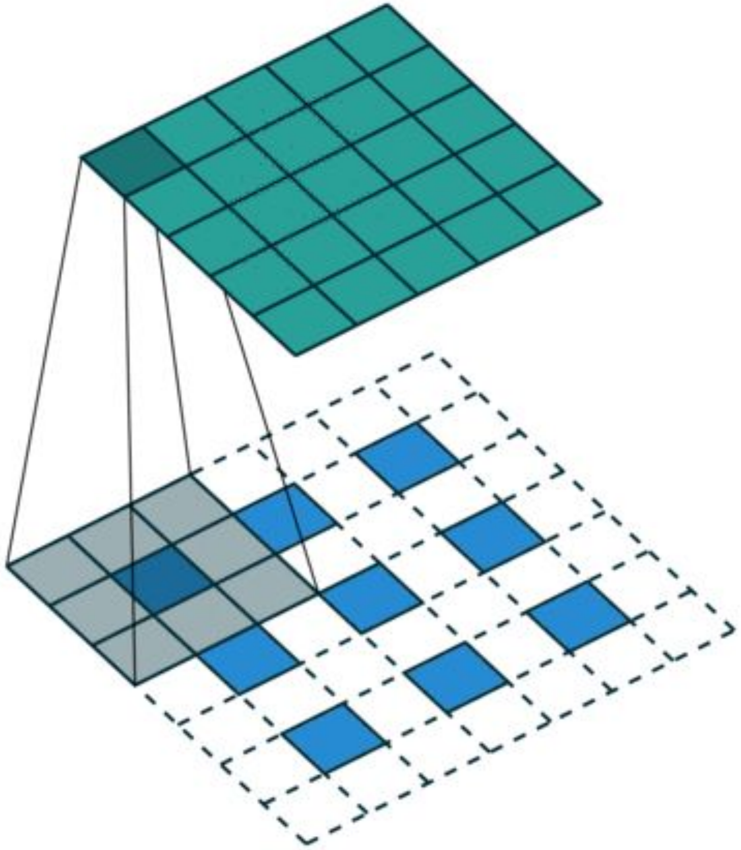


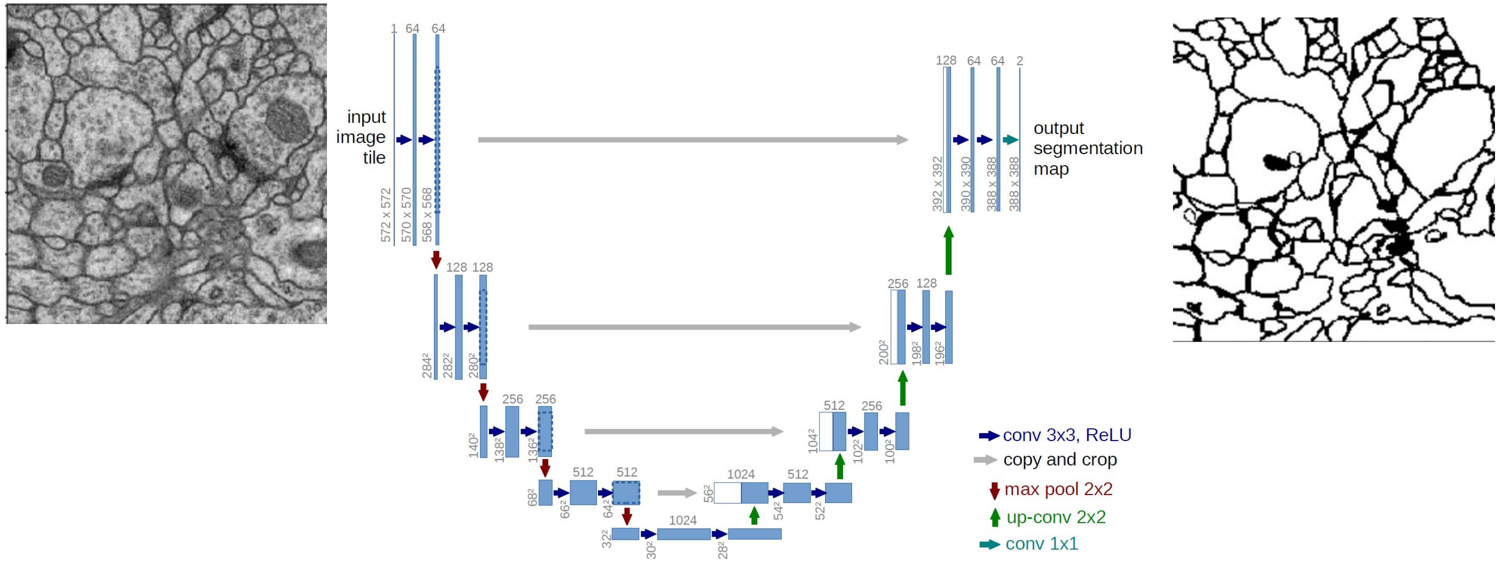
Upconvolution of 3x3 and stride = 1 padded with zeros

Effect: the output is a 5x5 image with ‘synthetic’ pixels

Activity: Conv2D in Pytorch (Colab notebook)

Usage of ‘up-convolution’: decoder paths in image generation models



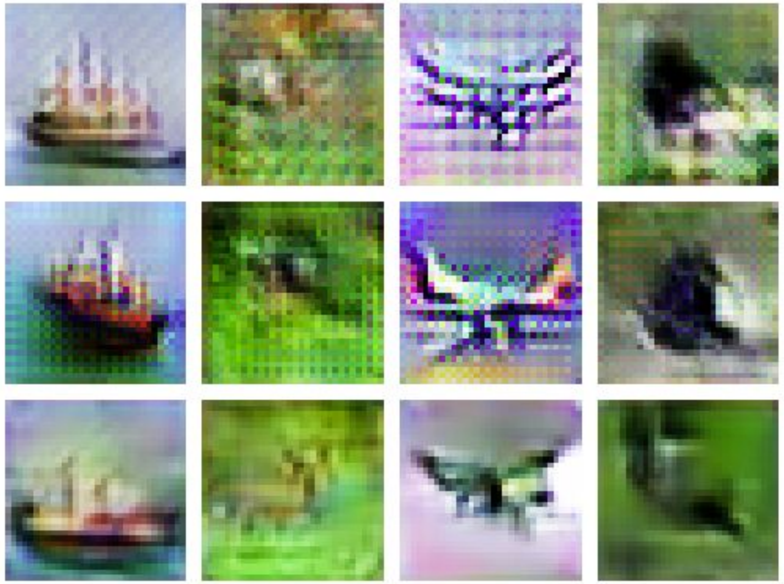


Upconvolution of 3x3 and stride = 1 padded with zeros

Effect: the output is a 5x5 image with ‘synthetic’ pixels

Activity: Conv2D in Pytorch (Colab notebook)

The checkerboard artifact from “deconvolution”

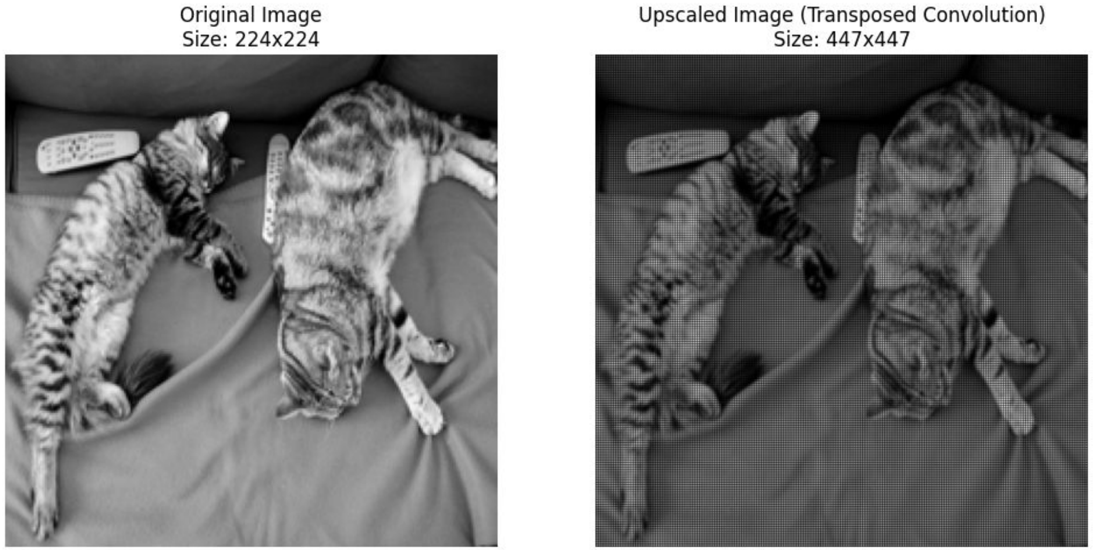


Deconv in last two layers.
Other layers use resize-convolution.
Artifacts of frequency 2 and 4.

Deconv only in last layer.
Other layers use resize-convolution.
Artifacts of frequency 2.

All layers use resize-convolution.
No artifacts.

Image from Deconvolution and Checkerboard artifacts



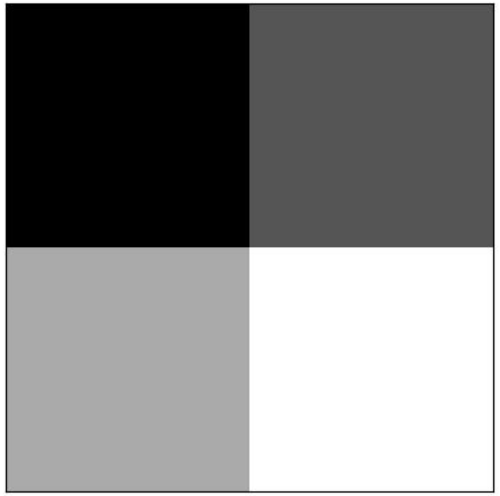
Original Image
Size: 224x224

Upscaled Image (Transposed Convolution)
Size: 447x447

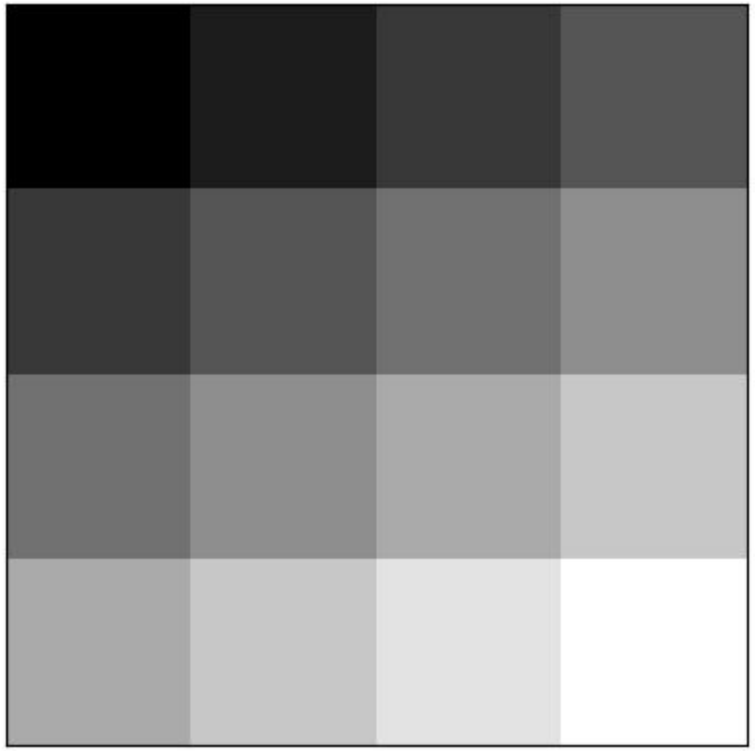
Problem: “checkerboard” artifacts

Bilinear interpolation

input



output



$$w_x = \frac{x - x_1}{x_2 - x_1}$$

$$w_y = \frac{y - y_1}{y_2 - y_1}$$

$(x_1, y_1) = (0, 0)$ coordinates of Q_{11}

$(x_2, y_2) = (1, 1)$ coordinates of Q_{22}

$(x, y) = (0.6, 0.7)$ target point

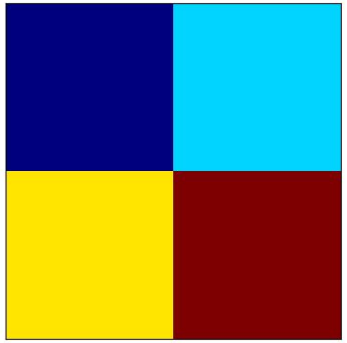
$$Q = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$$

$$f(x, y) = \begin{bmatrix} 1 - w_x & w_x \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} 1 - w_y \\ w_y \end{bmatrix}$$

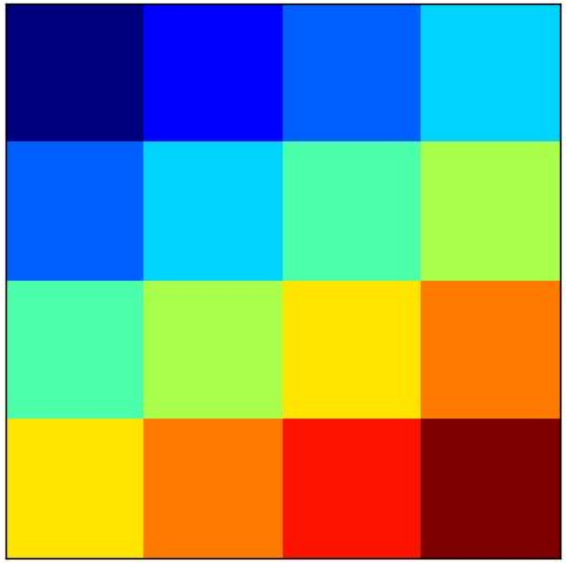
Bilinear interpolation in PyTorch

```
import torch
import torch.nn.functional as F

# Create a 1x1x2x2 tensor (batch_size x channels x height x width)
# Notice that the batch size and channel dimensions are created by wrapping
# the height and width tensor with two pairs of extra square brackets
input = torch.tensor([[[[10, 20],
                        [30, 40]]]],
                      dtype=torch.float32)

# Upscale to 4x4
output = F.interpolate(input, size=(4, 4), mode='bilinear',
align_corners=True)

import matplotlib.pyplot as plt
# The colormap is just for illustration of corner alignment
plt.imshow(input.squeeze(), cmap="jet")
plt.imshow(output.squeeze(), cmap="jet")
```

input (with colormap)
shape: 2x2



output (with colormap)
shape: 4x4



1x1 convolutions mix image channels

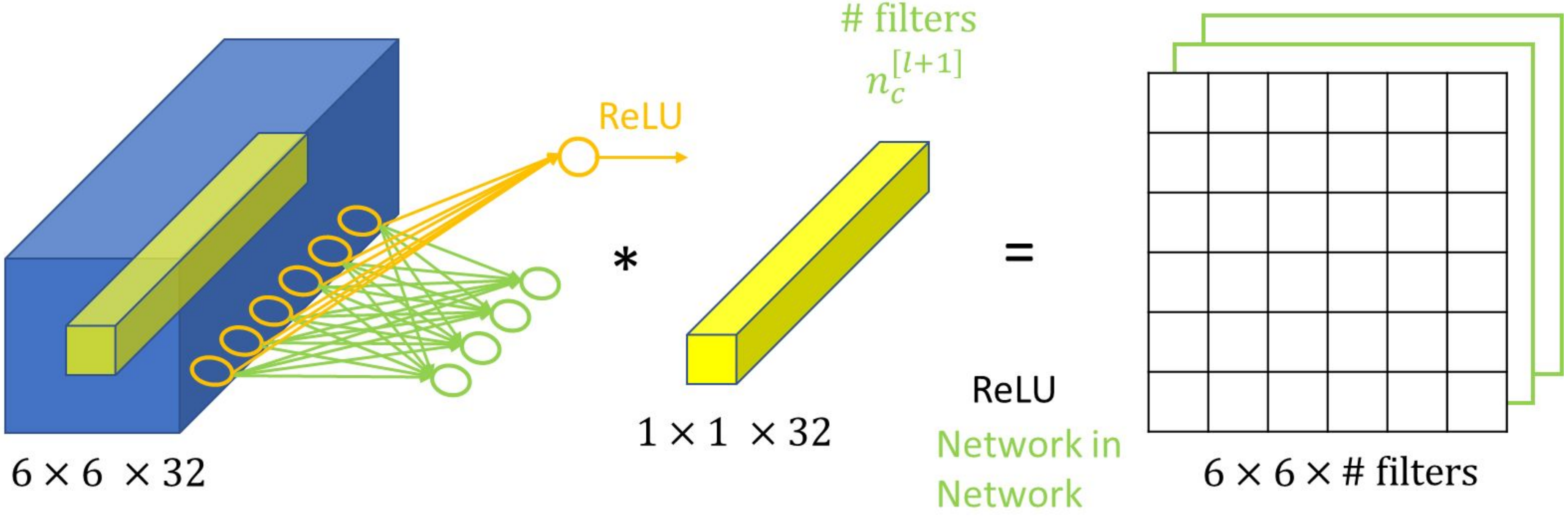
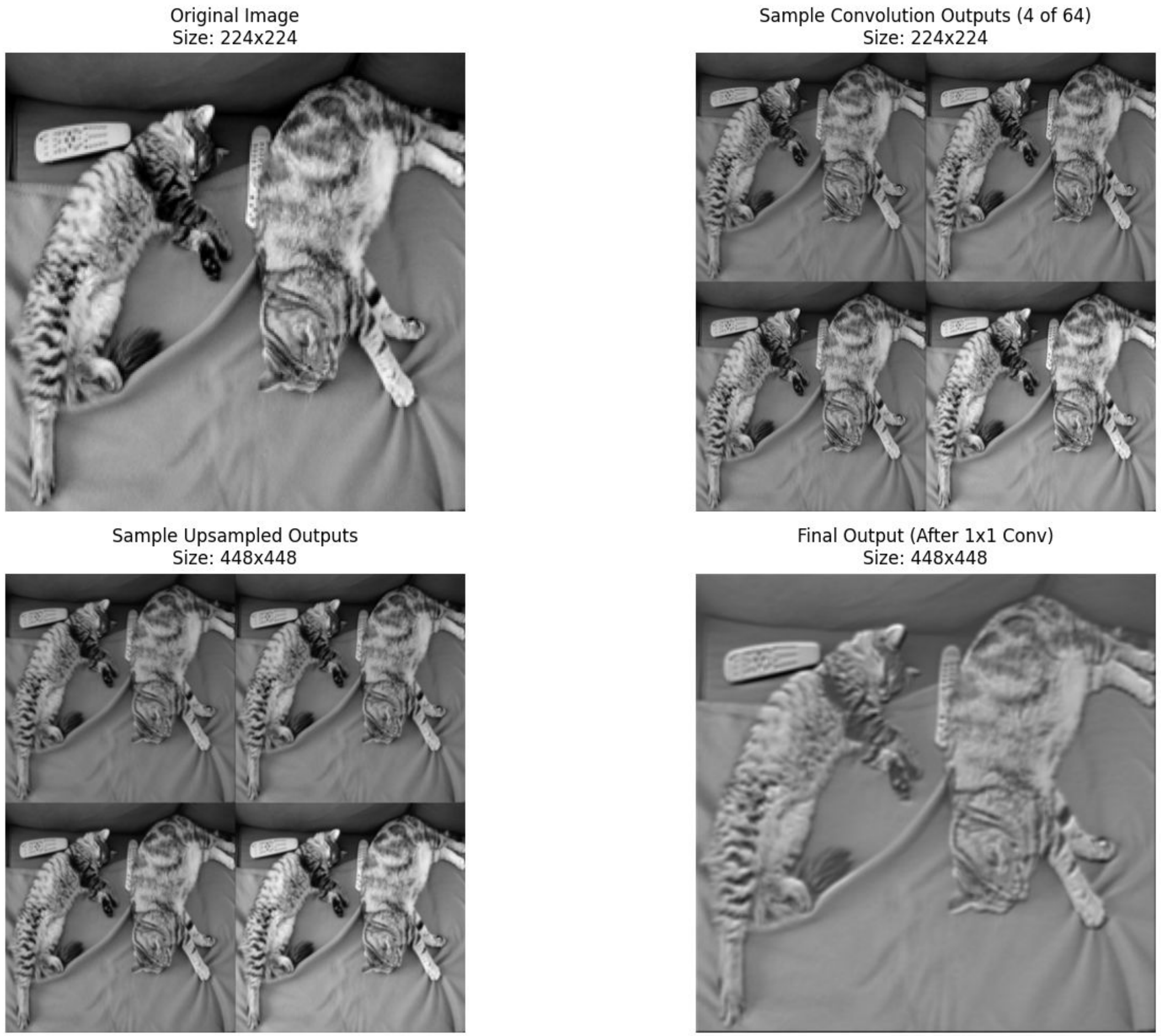


Image from "What does a 1x1 convolution do?" by Andrew Ng

Upsampling and channel mixing with 1x1 convs



Original Image
Size: 224x224

Sample Convolution Outputs (4 of 64)
Size: 224x224

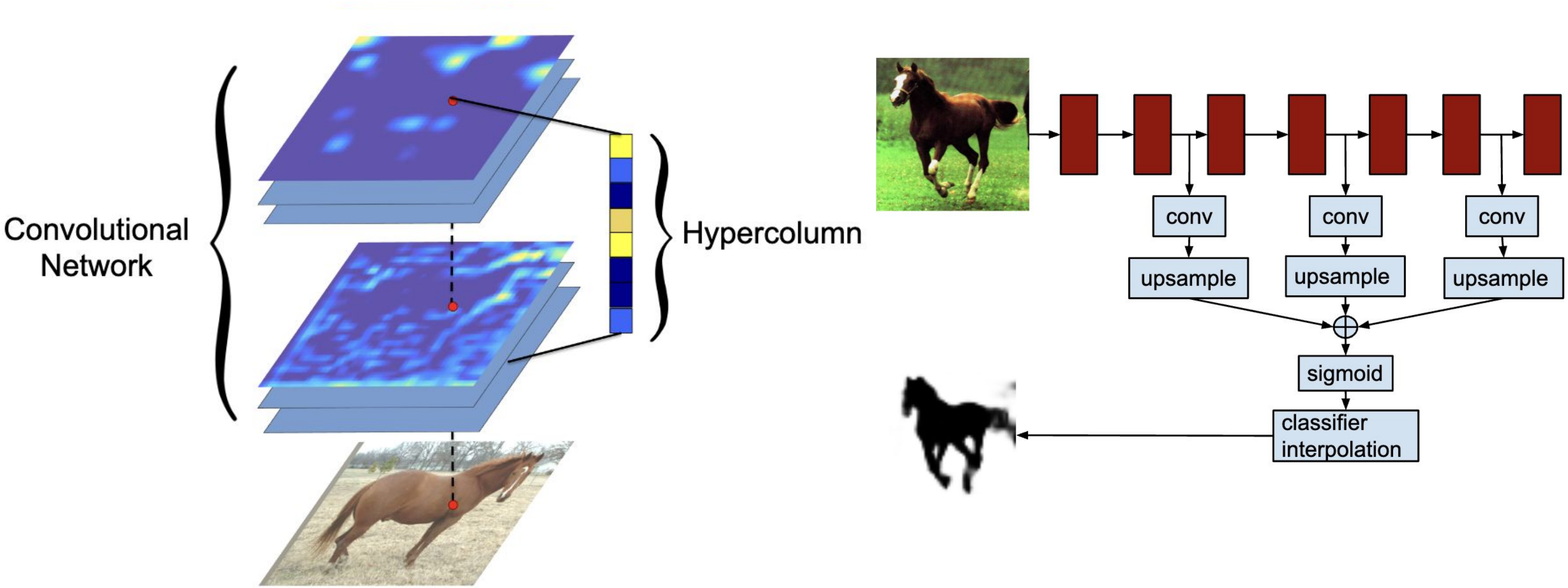
Sample Upsampled Outputs
Size: 448x448

Final Output (After 1x1 Conv)
Size: 448x448

```
layers = nn.Sequential(
    nn.Conv2d(
        in_channels=1,
        out_channels=64,
        kernel_size=3,
        padding=1,
        bias=False
    ),
    nn.Upsample(
        scale_factor=2,
        mode='bilinear',
        align_corners=True
    ),
    nn.Conv2d(
        in_channels=64,
        out_channels=1,
        kernel_size=1,
        bias=False
    )
)
```

Note: no checkerboard artifacts

Combining “hypercolumns” is a common use of 1x1 convolutions



Images from “Hypercolumns for Object Segmentation and Fine-grained Localization”

Summary

1x1 convolutions mix channel information

- Mixing channels allows us to create new images by combining learned features

Transposed convolutions enable upsampling (with checkerboard artifacts)

- Bilinear interpolation and channel mixing is a better alternative

Further reading and references

A guide to convolution arithmetic for deep learning

- https://arxiv.org/abs/1603.07285

Network in network (1x1 convolutions)

- https://arxiv.org/abs/1312.4400

Hypercolumns for object segmentation and fine-grained localization

- https://openaccess.thecvf.com/content_cvpr_2015/papers/Hariharan_Hypercolumns_for_Object_2015_CVPR_paper.pdf

SPONSORED BY THE

Federal Ministry
of Education
and Research