Deep Convolutional Generative Adversarial Networks

Fundamentals and Trends in Vision and Image Processing

Hallison Paz

IMPA, November 11 2021

Key contributions

- Proposal of architecture guidelines for training GANs using convolutions.
- Analysis of the potential of GANs as tools for unsupervised learning

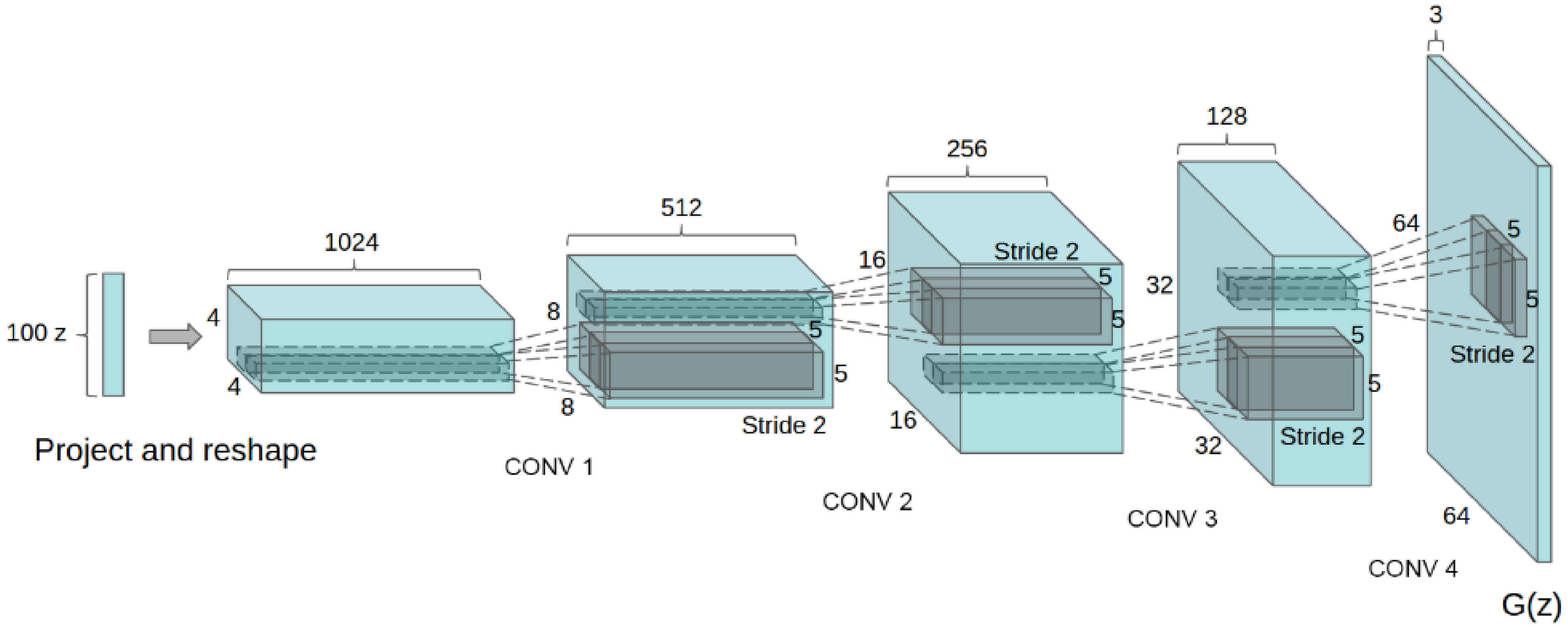
Why convolutions

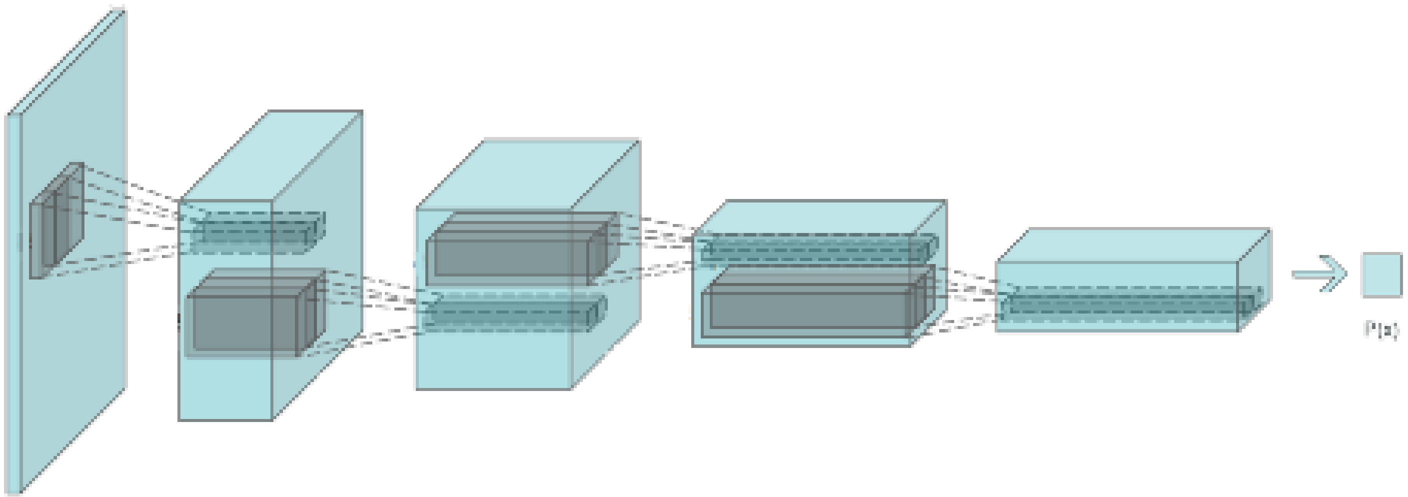
- Spatial coeherence
- Translation invariance
- Multiscale
- Already a good strategy for analysis

Architecture guidelines

- Replace pooling layers with [fractional-]strided convolutions
- Use batch normalization
- Remove fully connected hidden layers for deeper architectures
- Use **ReLU** in generator, except for the output, which uses **Tanh**.
- Use **LeakyReLU** in the discriminator

Generator Architecture



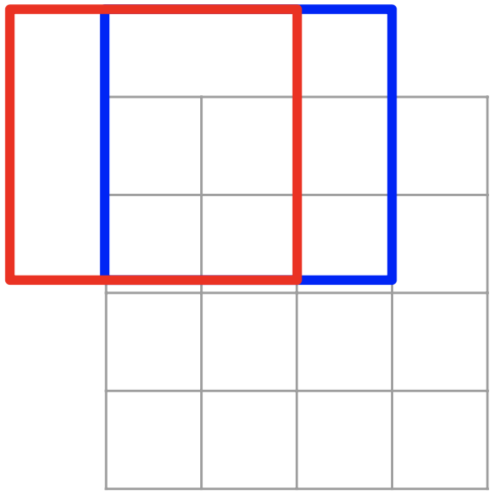


Discriminator Architecture

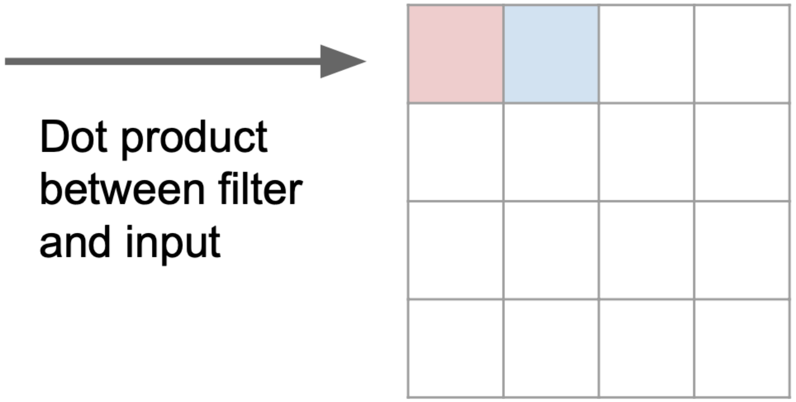
- Last layer is flattened and fed into a sigmoid output.

Pooling vs strided convolutions

Recall: Normal 3 x 3 convolution, stride 1 pad 1



Dot product between filter and input

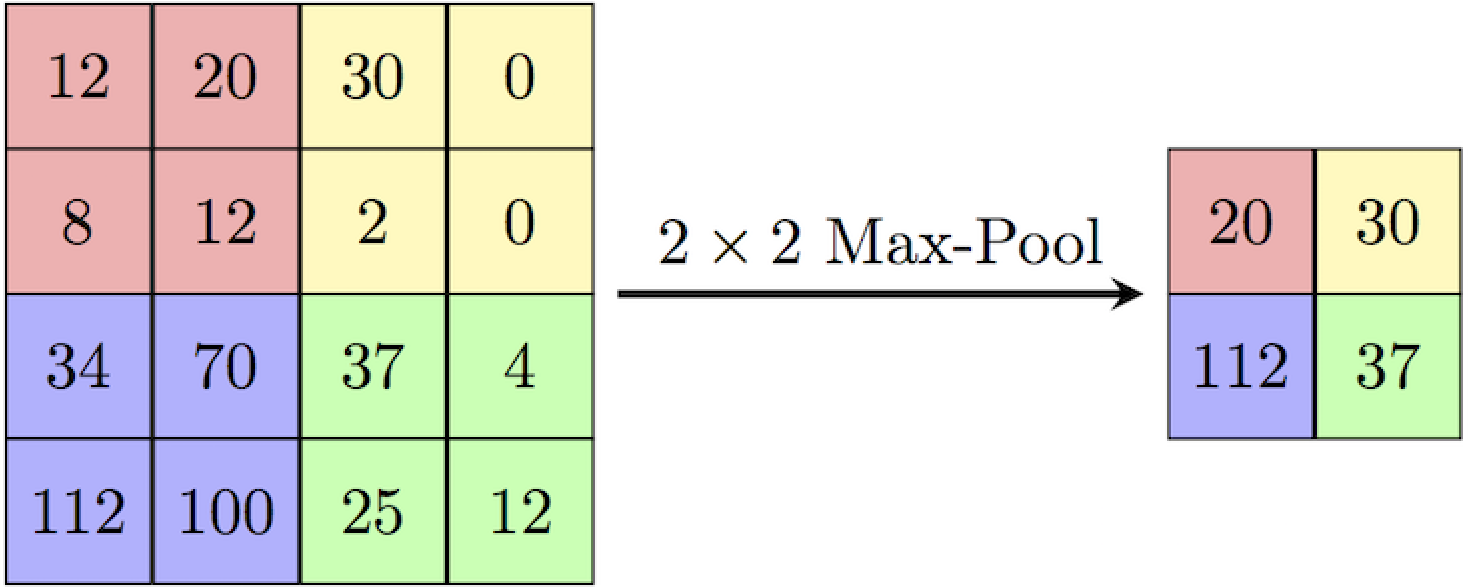


Input: 4 x 4

Output: 4 x 4

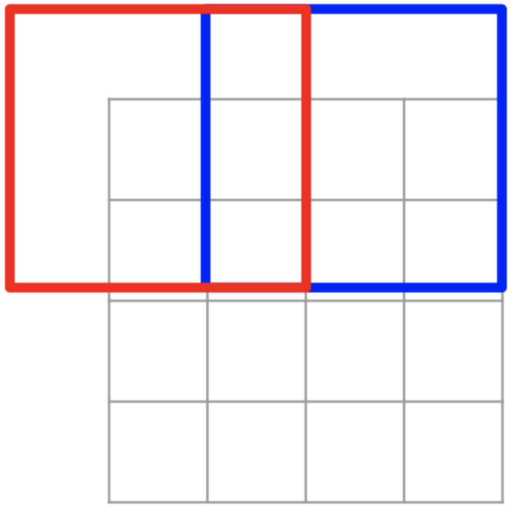
image from Fei-Fei Li & Justin Johnson & Serena Yeung (CS231n, 2017, lecture 11)

Downsampling with Pooling



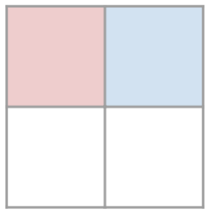
Downsampling with Strided convolutions

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input



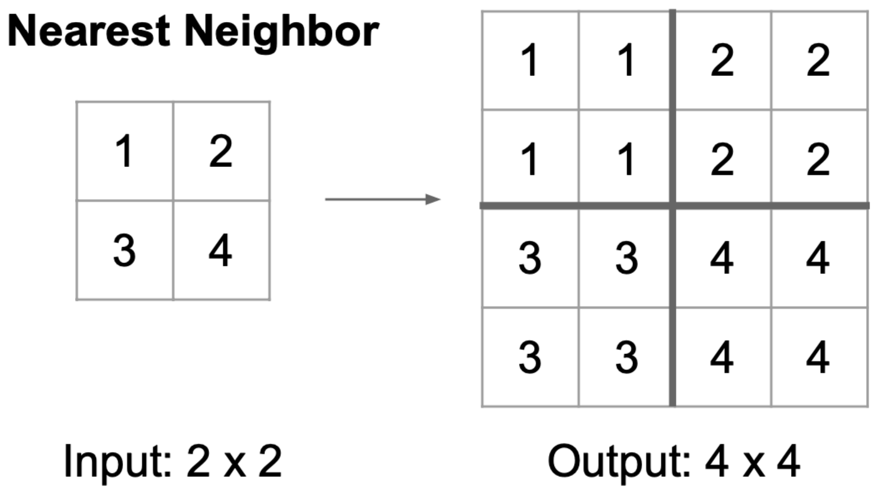
Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

image from Fei-Fei Li & Justin Johnson & Serena Yeung (CS231n, 2017, lecture 11)

Upsampling strategies



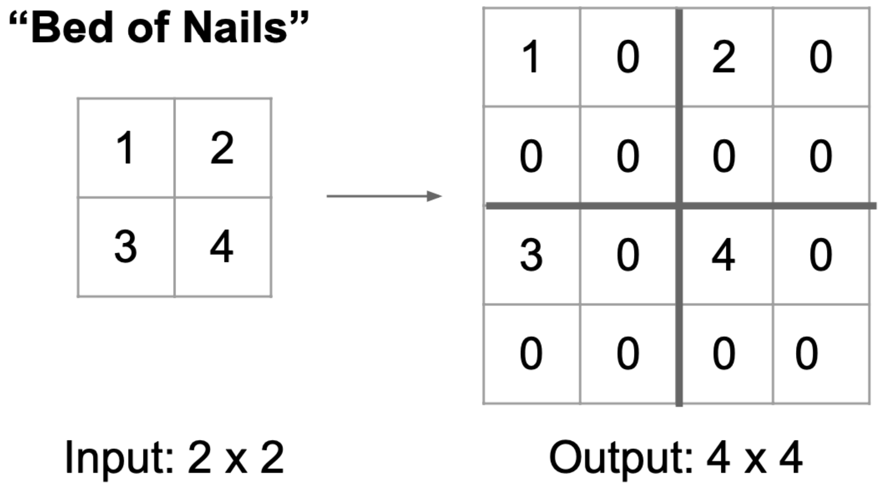
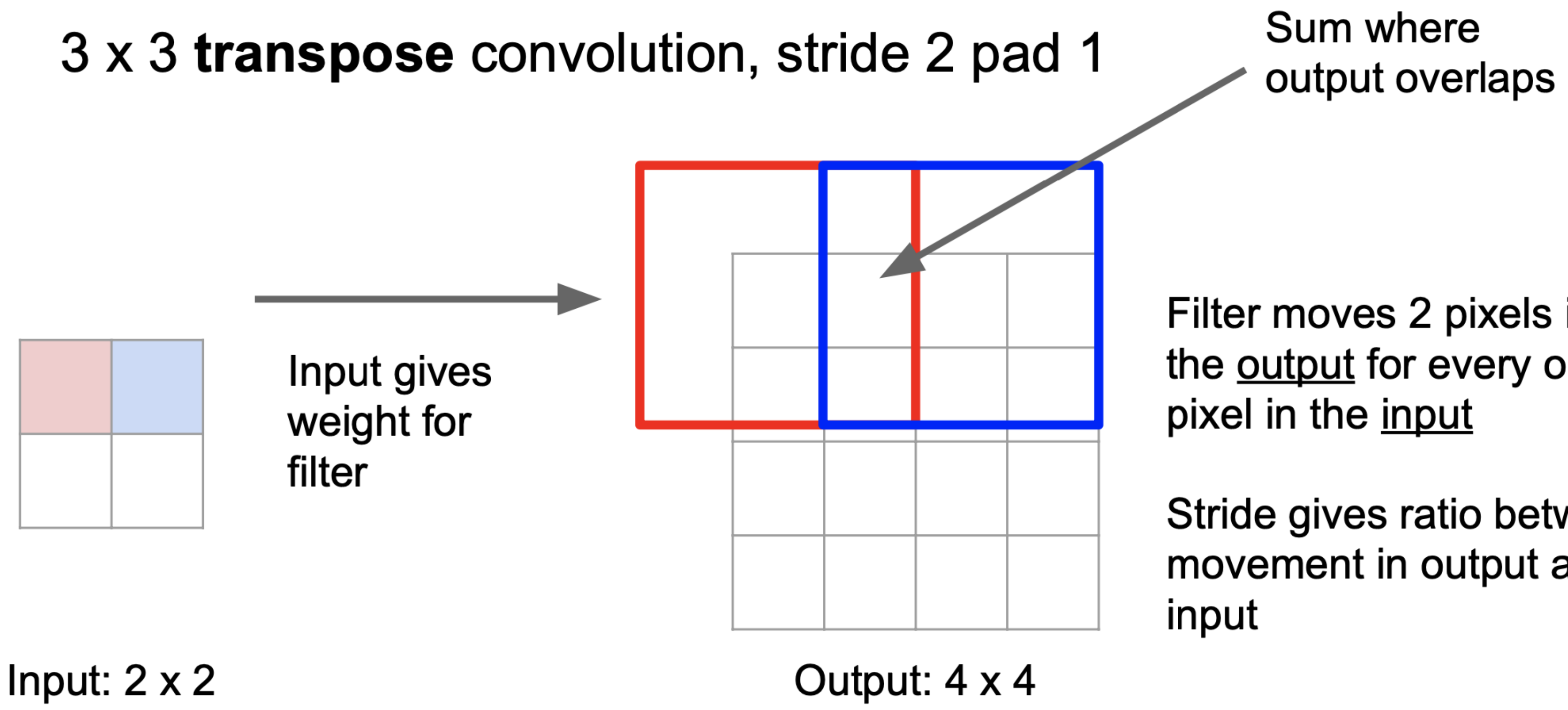


image from Fei-Fei Li & Justin Johnson & Serena Yeung (CS231n, 2017, lecture 11)

Fractional-strided convolutions

3 x 3 **transpose** convolution, stride 2 pad 1



Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

image from Fei-Fei Li & Justin Johnson & Serena Yeung (CS231n, 2017, lecture 11)

Fractional-strided convolutions

- Also called transposed convolution

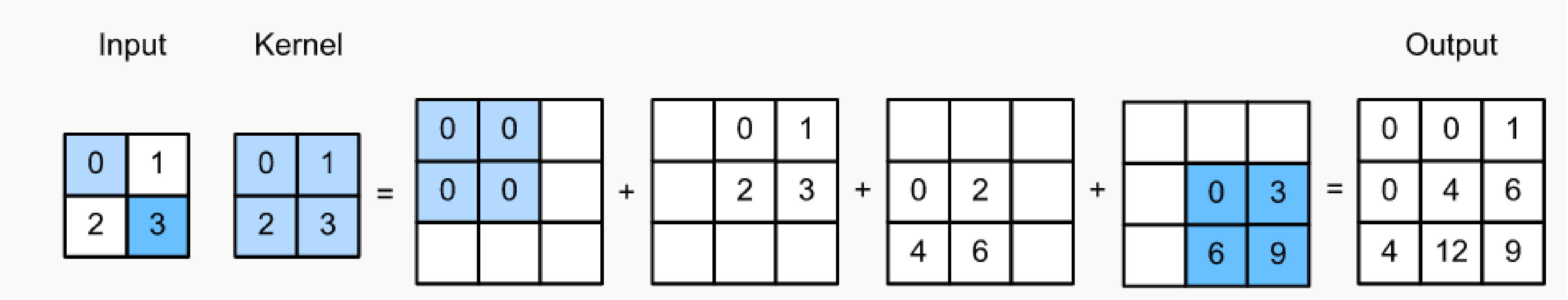


Image from Transposed Convolution Demystified

Batch Normalization

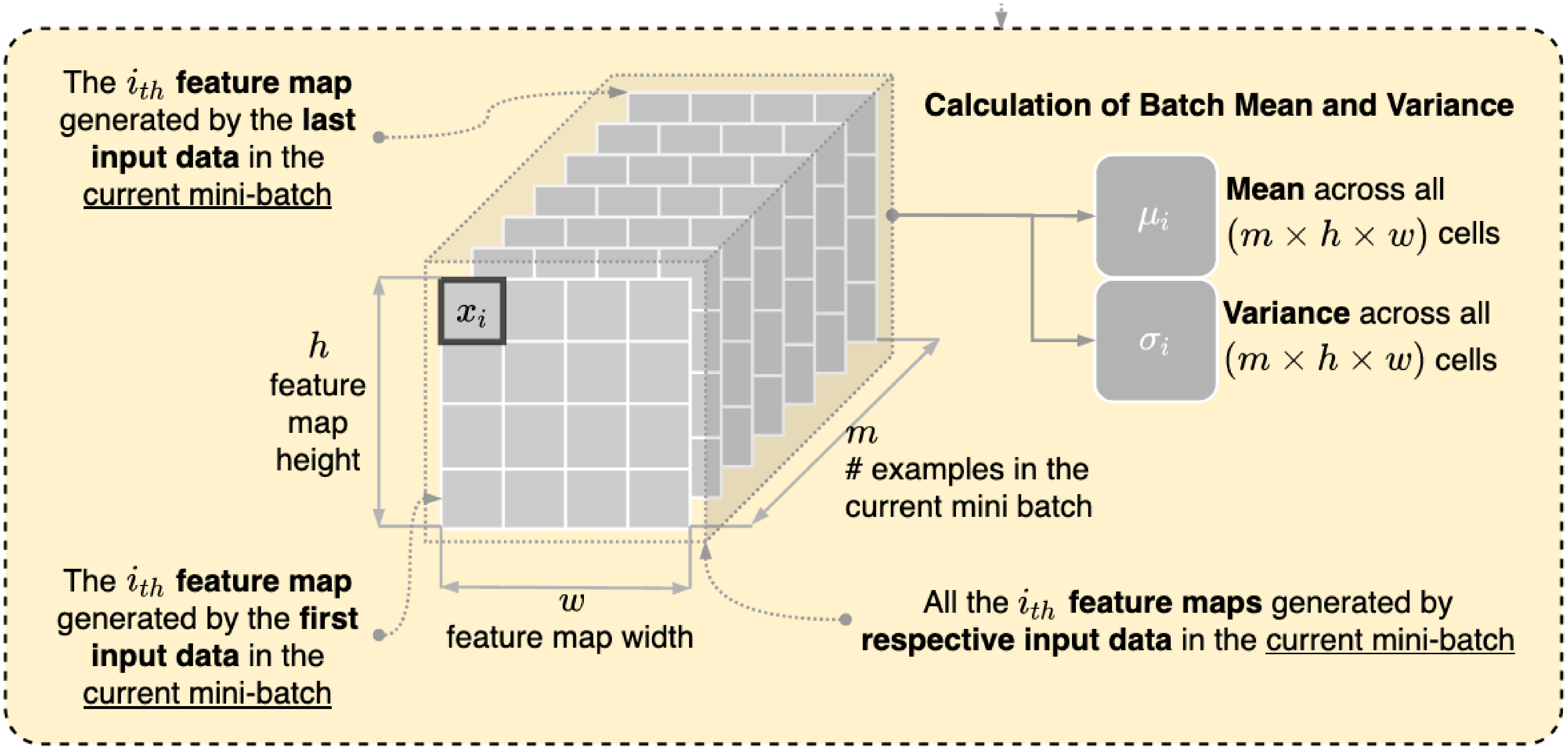
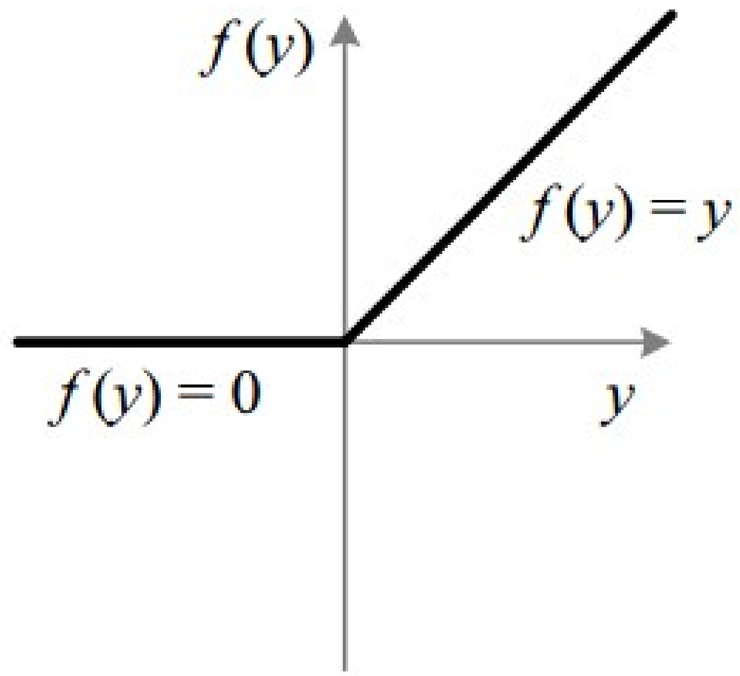
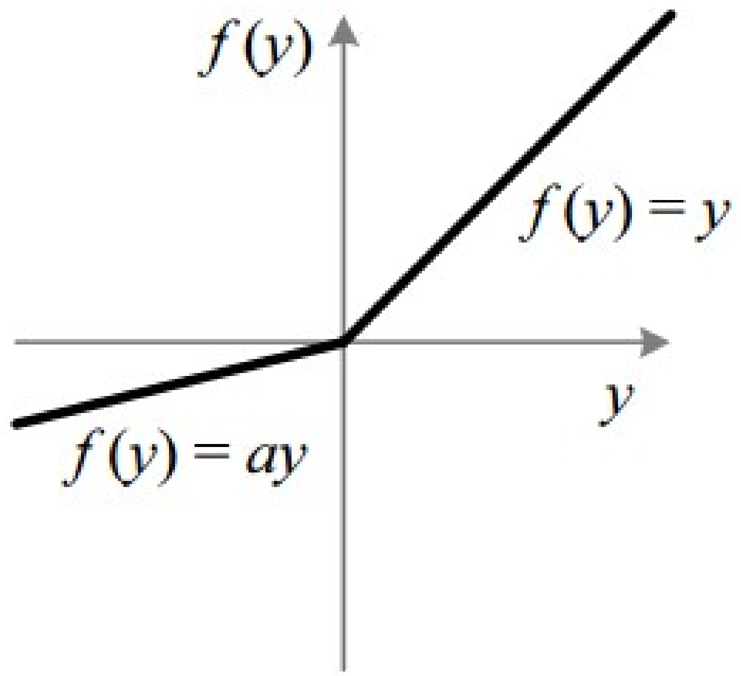


Image from https://stackoverflow.com/questions/65613694/calculation-of-mean-and-variance-in-batch-normalization-in-convolutional-neural

Relu x Leaky-Relu





Training and Results

Training details

- Training images scaling to the range of the tanh activation function [-1, 1]
- SGD with a mini-batch size of 128
- Weights initialized from a zero-centered Normal distribution (std=0.02).
- In the LeakyReLU, the slope was set to 0.2
- Adam optimizer using lr=0.0002 and momentum term $\beta_1 = 0.5$

Results

Generated bedrooms after one epoch of training on LSUN

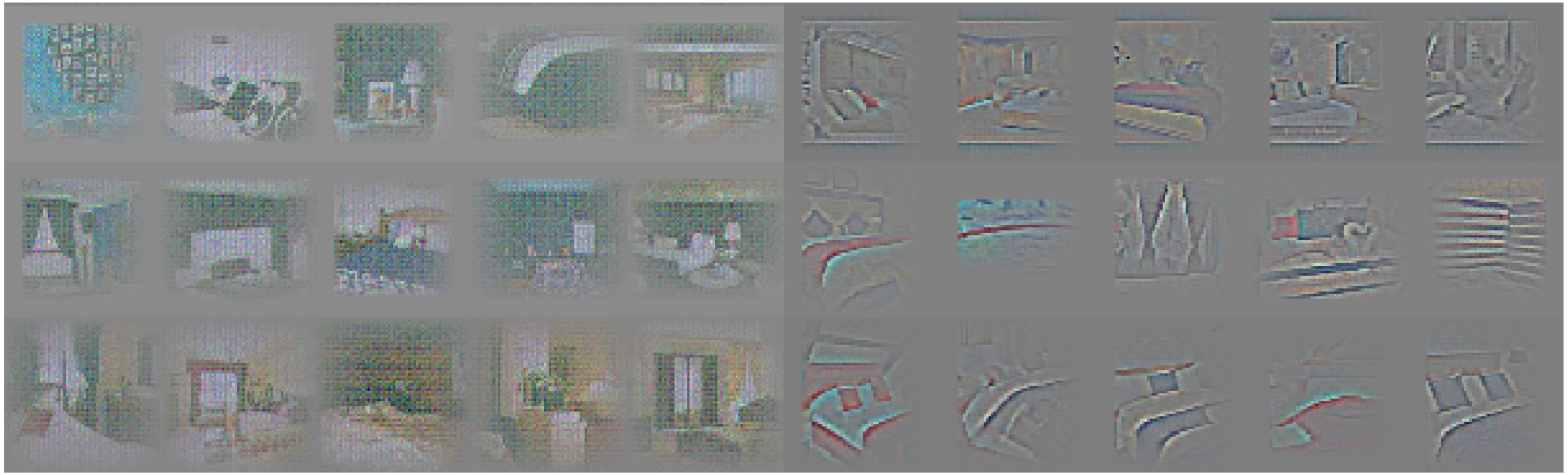
Results

Generated bedrooms after five epochs of training on LSUN

GANs as Tools for Unsupervised Learning

Visualizing features activations

- The discriminator learned relevant features of the scene



Random filters Trained filters

Using features for supervised learning

- Using discriminator's features to train a SVM classifier

Table 1: CIFAR-10 classification results using our pre-trained model. Our DCGAN is not pre-trained on CIFAR-10, but on Imagenet-1k, and the features are used to classify CIFAR-10 images.

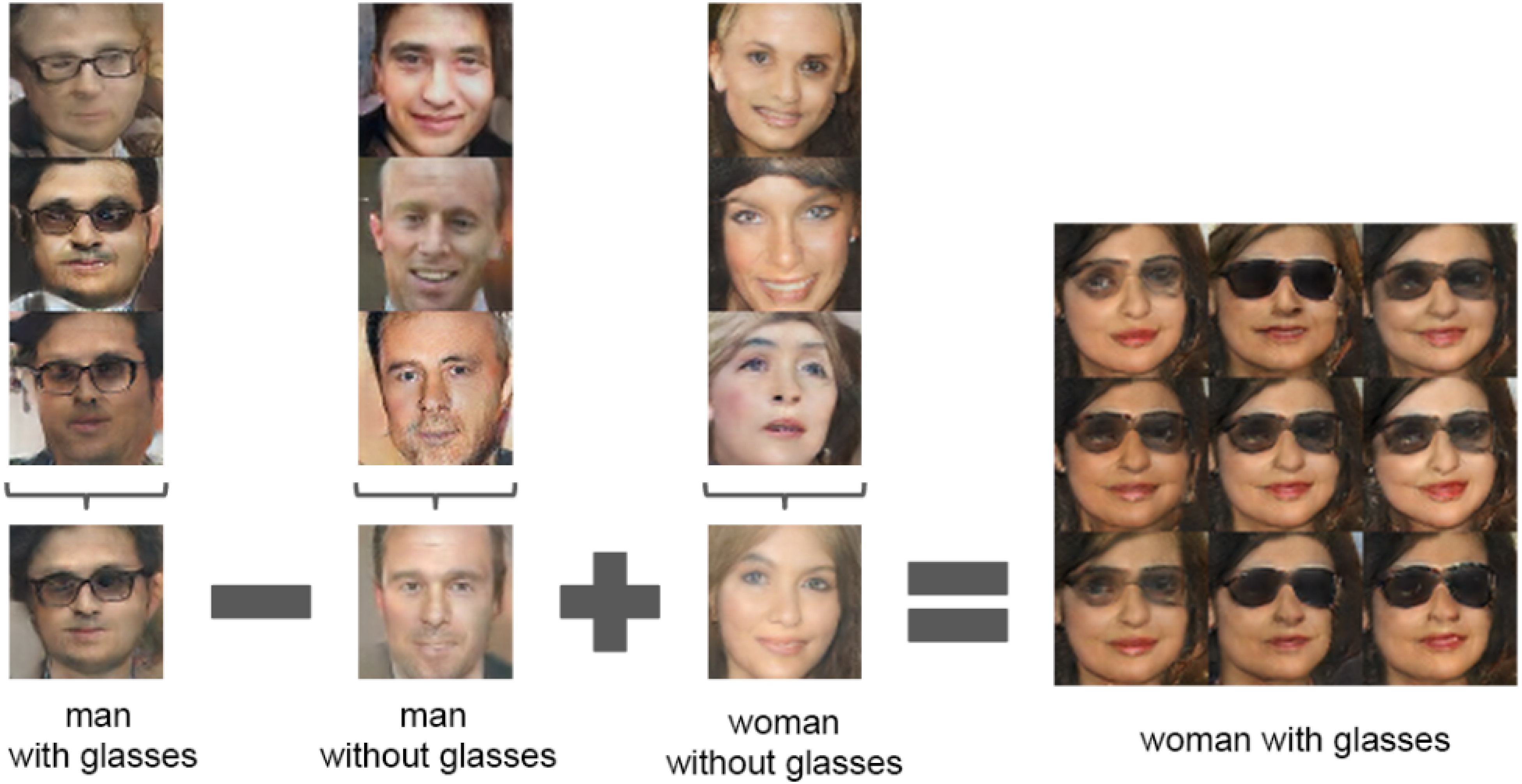
Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% ($\pm$0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% ($\pm$0.7%)	3200
View Invariant K-means	81.9%	72.6% ($\pm$0.7%)	6400
Exemplar CNN	84.3%	77.4% ($\pm$0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% ($\pm$0.4%)	512

Walking in the latent space

Walking in the latent space

- Interpolation has semantics;
    - It learns a manifold.
- An interesting test to validate results

Vector Arithmetic



Let's see it in practice...

...next class! #statytuned