Свертка

Свёртка (convolution) – это мат.операция функциональном анализе, которая при применении к двум функциям *f* и *g* возвращает третью функцию, ссоответствующую взаимно-корреляционной функции *f*(*x*) и *g*(−*x*). .

Интерпретация: «схожесть» одной функции с отражённой и сдвинутой копией другой.

Интуитивно: свёртка скользящим окном вычисляет в каждом месте *взвешенную сумму* соседних значений, отражая степень похожести участка данных на шаблон ядра. Если участок “похож” на ядро, результирующее значение свёртки будет большим, и наоборот

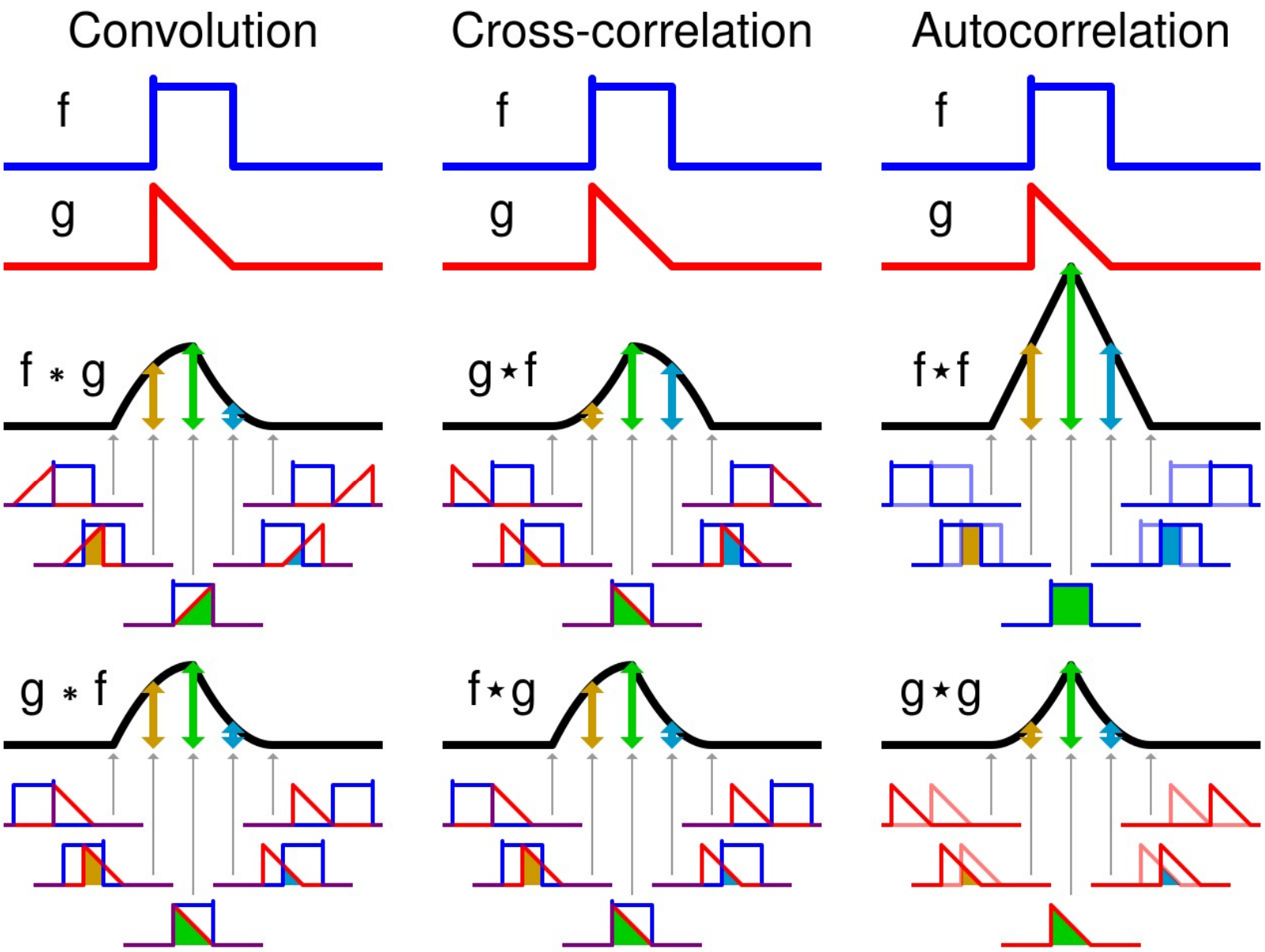
Непрерывная свёртка функций f(t) и g(t):

$$(f * g)(x) \ \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(y)\, g(x-y)\, dy = \int_{-\infty}^{\infty} f(x-y)\, g(y)\, dy.$$

Дискретная свёртка массивов A и B размера n×n и m×m даёт массив C размера (n−m+1)×(n−m+1):

$$C_{i,j} = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} A_{i+u,\, j+v}\ B_{u,v}$$

Графически: **наложение шаблона на данные**: форма одной функции “прикладывается” к другой при разных сдвигах и измеряется степень их перекрытия. Чем больше результат – тем больше текущий участок данных соответствует шаблону ядра.


Convolution
Cross-correlation
Autocorrelation
f
g
f ∗ g
g ⋆ f
f ⋆ f
g ∗ f
f ⋆ g
g ⋆ g

Свертка

В **теории сигналов** свёртка описывает отклик линейной системы на входной сигнал: выход есть свёртка входа с импульсной характеристикой системы. Например, в цифровой обработке сигналов фильтрация реализуется свёрткой входного сигнала с коэффициентами фильтра (ядром).

В **обработке изображений** многие классические фильтры (размытие, резкость, детектирование границ) – это свёртки изображения с некоторой ядрой-маской.

Свёртка широко применяется в **теории связных преобразований (преобразования сигналов из одной области в другую)**. По *теореме свёртки* в гармоническом анализе, свёртка во временно́й области соответствует покоординатному произведению спектров в частотной области. Это свойство лежит в основе ускоренного вычисления свёрток через быстрое преобразование Фурье (FFT) – так называемая *циклическая свёртка* через умножение спектров сигналов.

В **преобразовании Фурье** сама операция интеграла преобразования эквивалентна корреляции сигнала с базовой синусоидой - близко по смыслу к свёртке, учитывая симметричность синусоид.

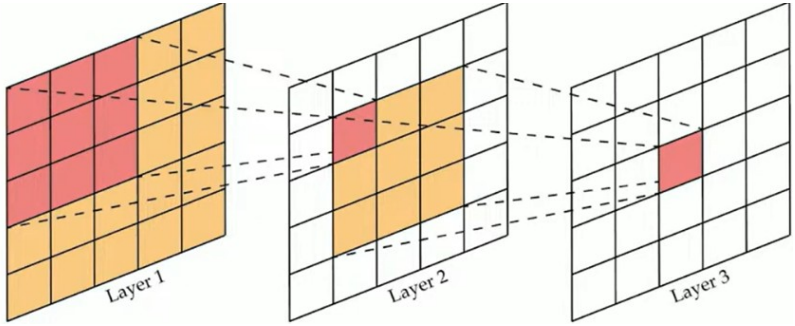
В **вейвлет-преобразовании** явно присутствует свёртка: сигнал разворачивается во временно-частотное представление путём свёртки исходного сигнала с масштабированным *вейвлетом*.

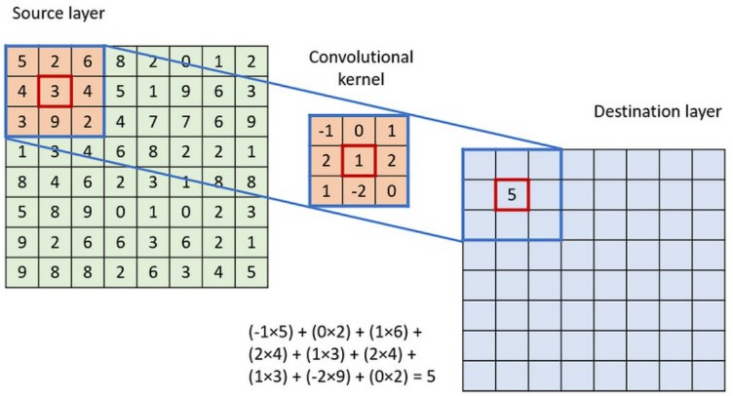
В **теории вероятностей** свёртка распределений используется для нахождения распределения суммы независимых случайных величин.

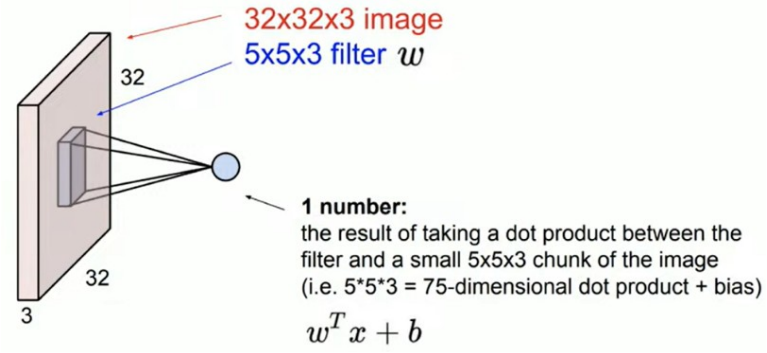
Свертка

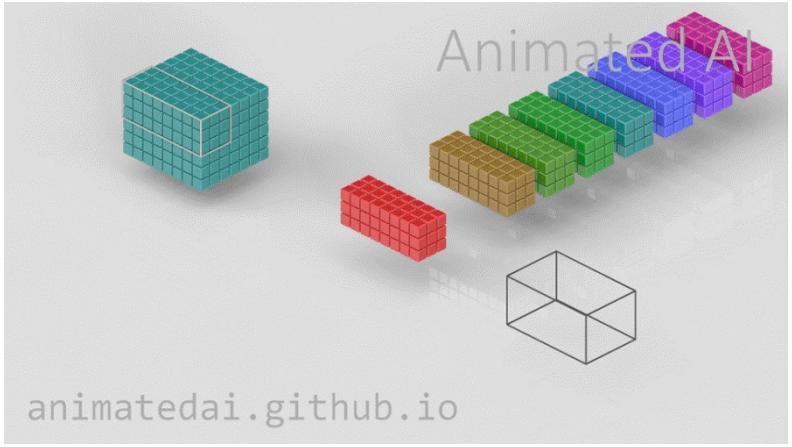
Вид свёртки	Где применяется	Особенности
Непрерывная свёртка (интеграл)	Анализ сигналов, физика, диффур-ры	Описывает линейные системы, сглаживание сигналов, объединяет функции. Частотная область: превращается в произведение спектров.
Дискретная свёртка (сумма)	Цифровая обработка сигналов и изображений	Реализуется алгоритмически для фильтрации, фильтров в изображениях (размытие, выделение краёв и т.д.). Результат зависит от наложения ядра на данные.
Циклическая свёртка	Спектральные методы (FFT), обработка периодических сигналов	Считается по модулю длины (с сигналами, дополнёнными нулями). Используется для быстрого вычисления свёртки через умножение в частотной области. Важна при периодических граничных условиях.
Корреляция (кросс-корреляция)	Поиск шаблонов в сигнале, машинное зрение	Похожа на свёртку, но без отражения ядра. Показывает степень сходства между сигналом и шаблоном при разных сдвигах. В библиотеках компьютерного зрения свёртка часто реализуется как корреляция (фактически, CNN используют корреляцию вместо строгой математической свёртки).
Свёртка в вейвлет-анализе	Временно́-частотный анализ сигналов, сжатие	Вейвлет-преобразование – интегральная свёртка сигнала с масштабированным вейвлетом. Позволяет выделять локальные частоты, анализировать сигнал на разных масштабах.
Свёртка распределений	Теория вероятностей, статистика	Свёрткой плотностей вероятностей получают распределение суммы двух независимых величин. Позволяет находить закон распределения результирующей случайной величины (например, сумма сигналов = свёртка их распределений).

Свертка в CNN: стандартная свёртка (standard/vanila convolution).

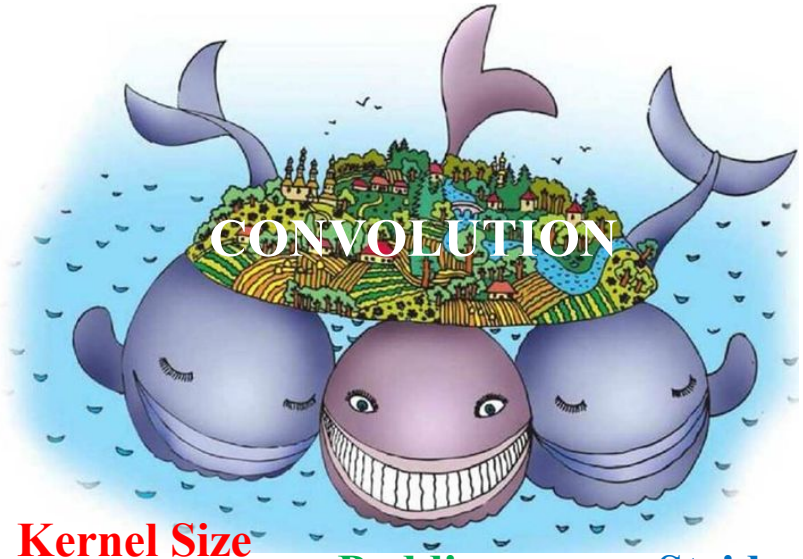








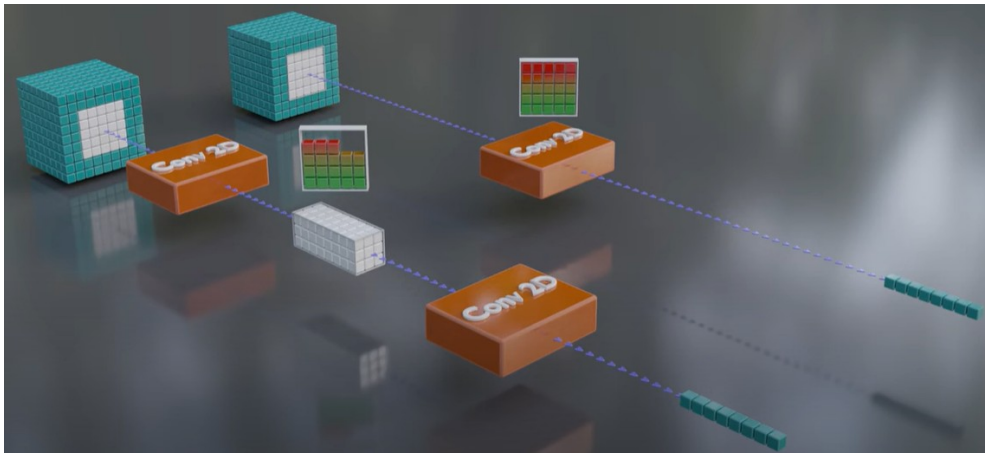
@animatedai



Kernel Size **Padding** **Stride**

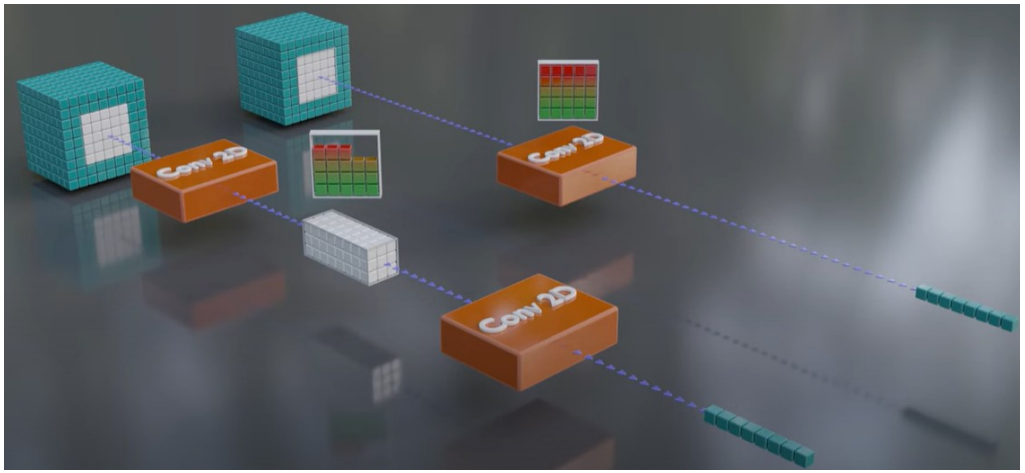
What is the best Kernel Size?

Kernel Size: 3x3 + 3x3 VS 5x5



Calculations: 72% VS 100%

Kernel Size: 3x3 + 3x3 + 3x3 VS 7x7



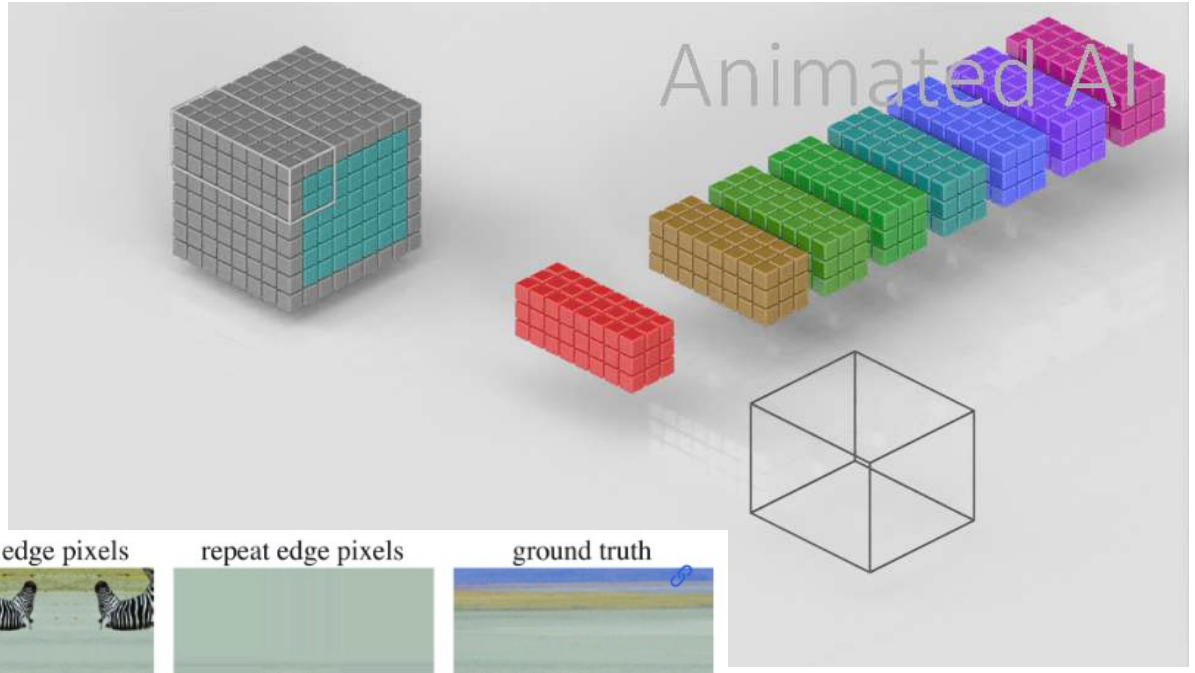
Calculations: 55% VS 100%

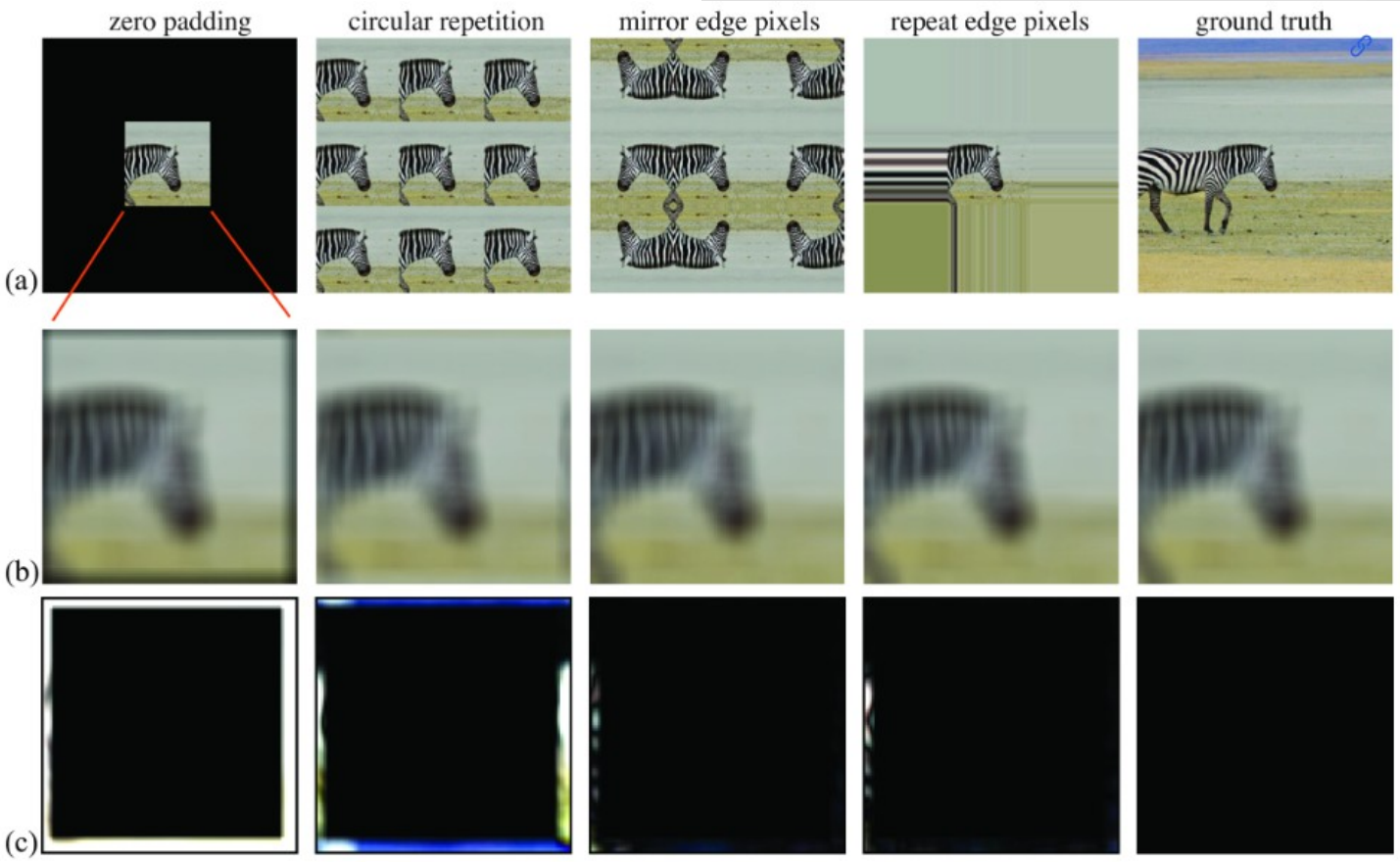
AlexNet (2012)
11x11 + 5x5 +3x3

ZFNet (2013)
7x7 + 5x5 +3x3

VGG (2014)
Only 3x3

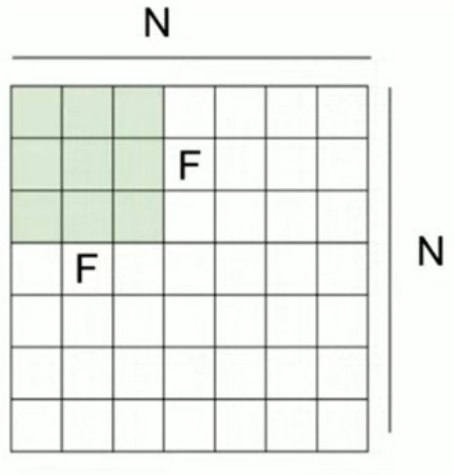
Padding

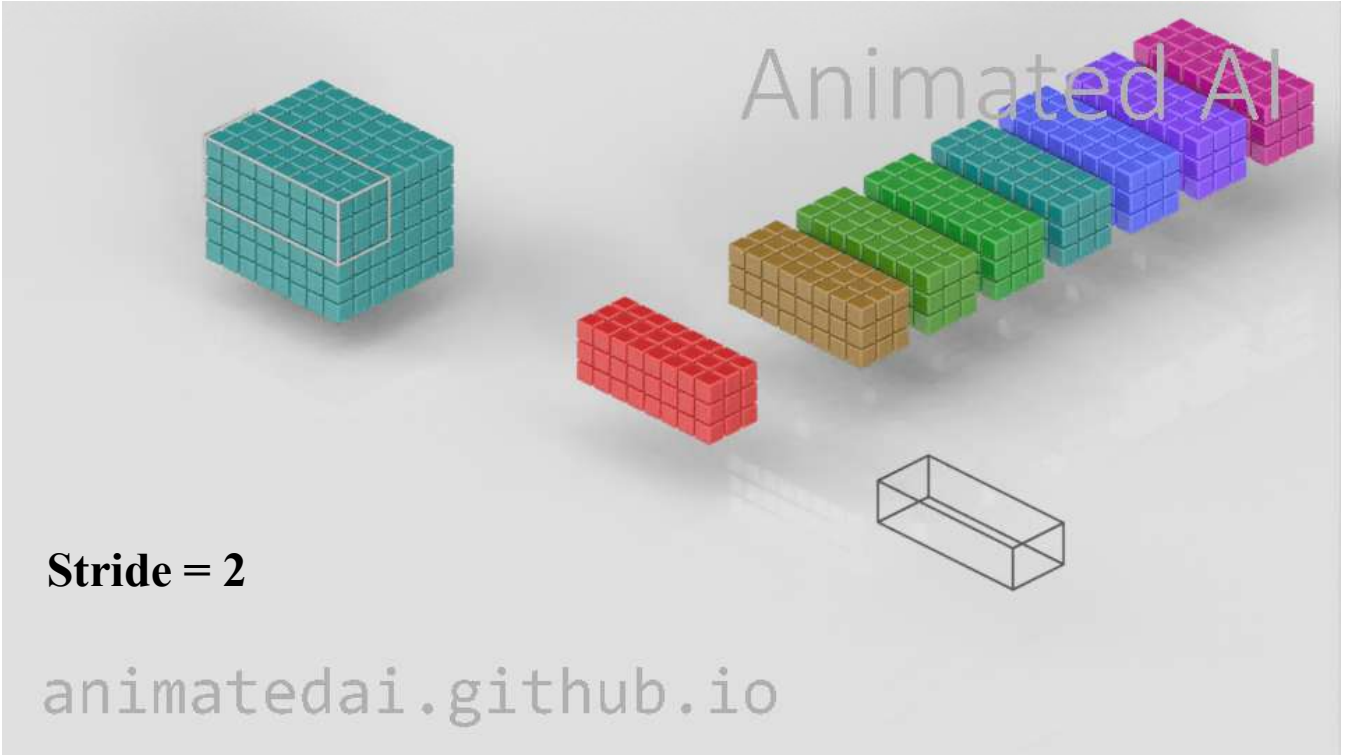

Animated AI


zero padding
circular repetition
mirror edge pixels
repeat edge pixels
ground truth
(a)
(b)
(c)

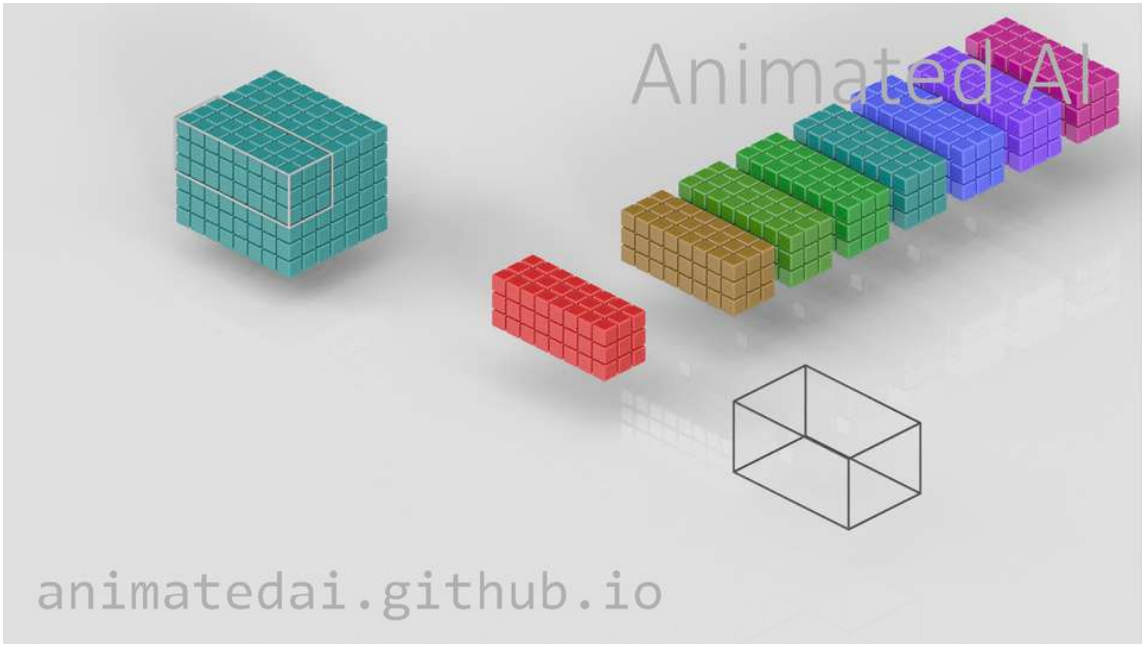
Stride

Output size:
(N - F) / stride + 1



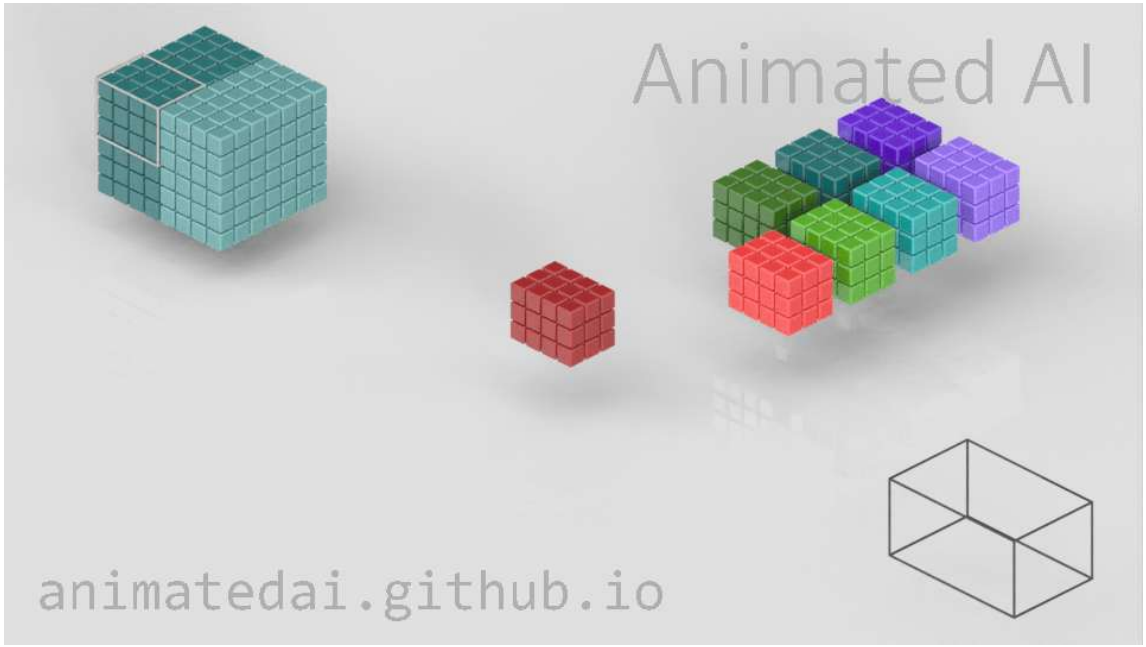


Group convolution

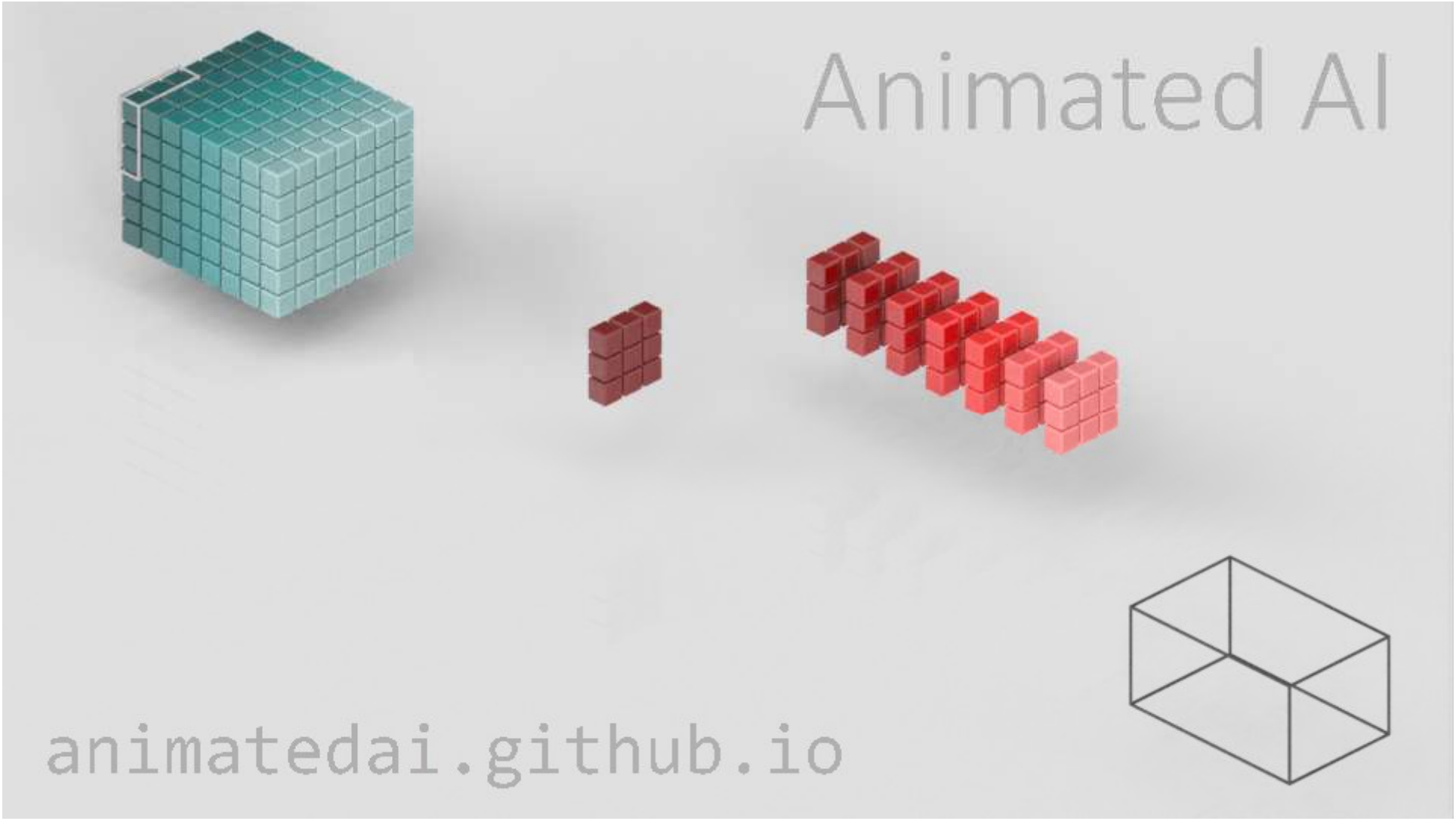


1 Group convolution

2 Group convolution



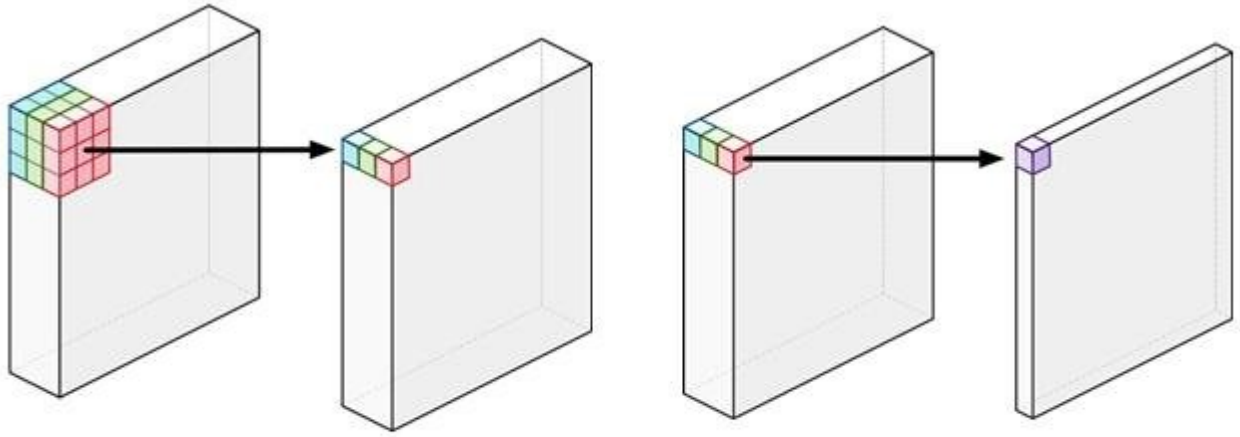
Depthwise convolution



Depthwise (8 Groups) convolution

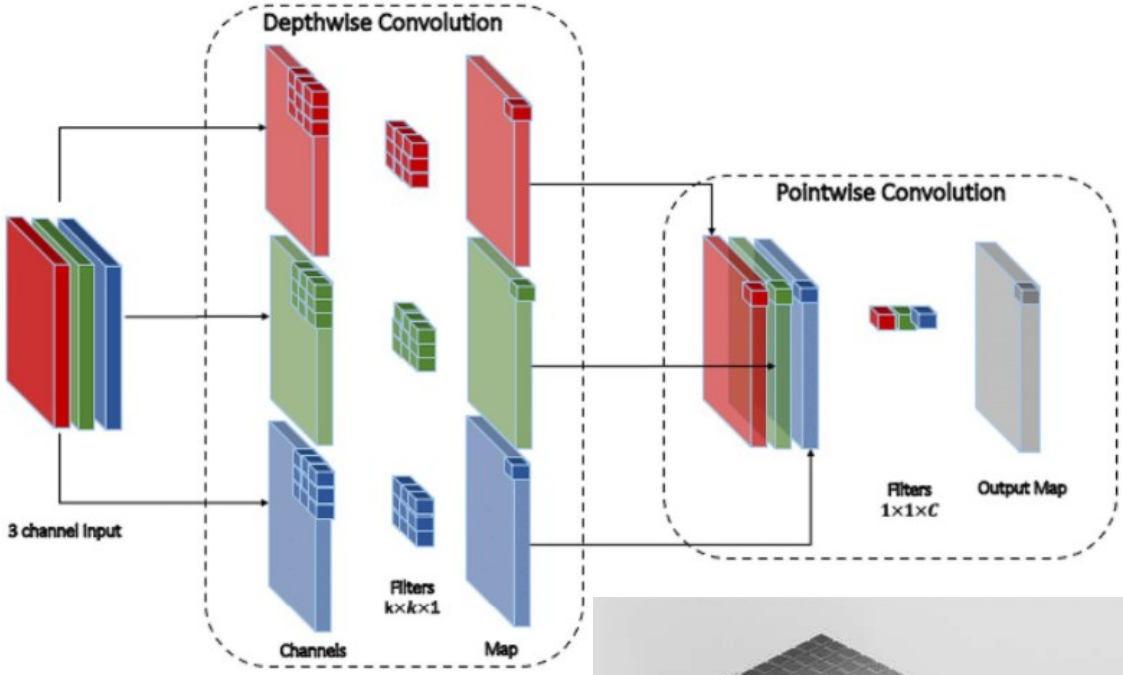
Pointwise Convolution

Initially 1x1 convolutions were proposed at Network-in-network(NiN). After they were highly used in GoogleNet architecture. Main features of such layers:
Reduce or increase dimensionality
Apply nonlinearity again after convolution
Can be considered as “feature pooling”
They are used in such way: we have image with size 32x32x100, where 100 means features, and after applying 20 1x1 convolutions filters we will get images with 32x32x20 dimensions.



a) depthwise convolution b) pointwise convolution

Depthwise Separable Convolution: Depthwise + Pointwise



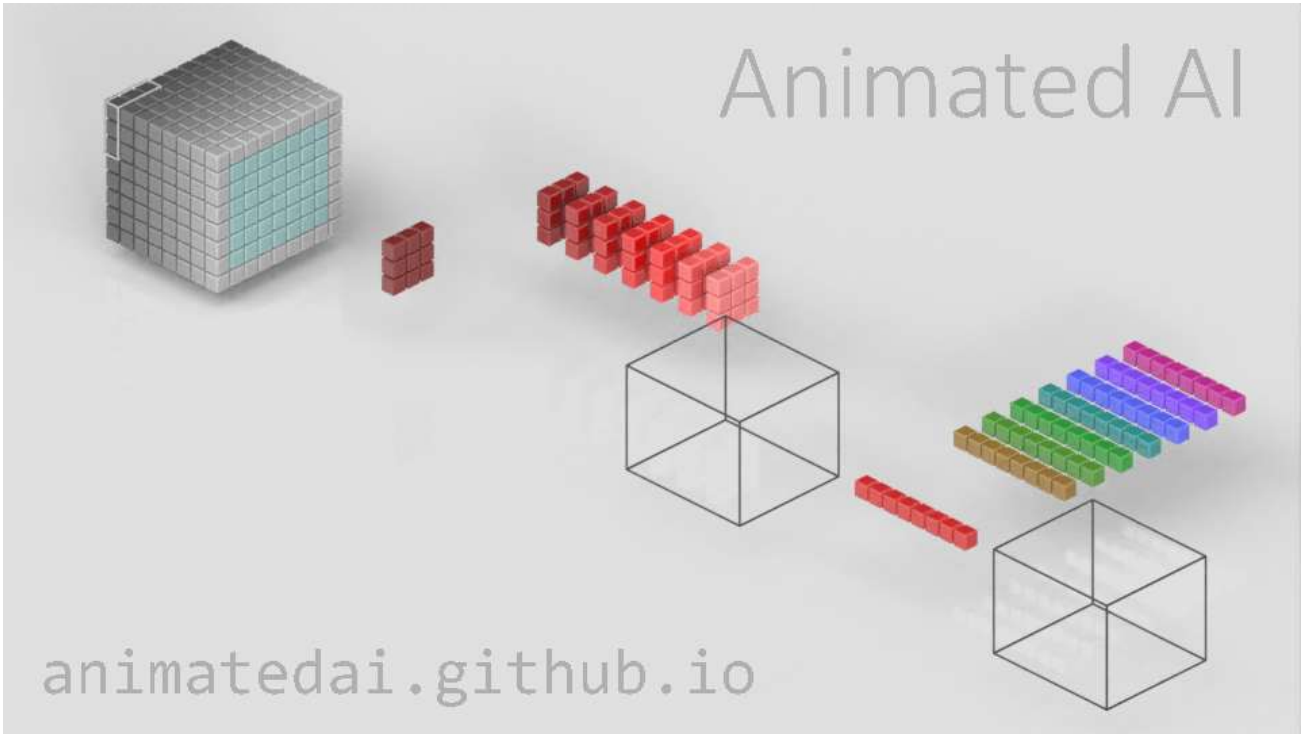
- computational cost
- the number of parameters

maintaining similar performance

Two steps:

Depthwise Convolution

Pointwise Convolution.



Depthwise Separable Convolution

Depthwise Convolution: A single convolution filter applies separately to each channel of the input. A dedicated kernel convolves each channel. In an RGB image with 3 channels, each channel receives its kernel, the output retains the same number of channels as the input.
Pointwise Convolution: This step uses a 1×1 convolution to combine the outputs of the depthwise convolution across the channels. This means it takes the depthwise convolved channels and applies a 1×1 convolutional filter to each pixel, combining information across the different channels. It integrates the features extracted independently by the depthwise step, creating an aggregated feature map.

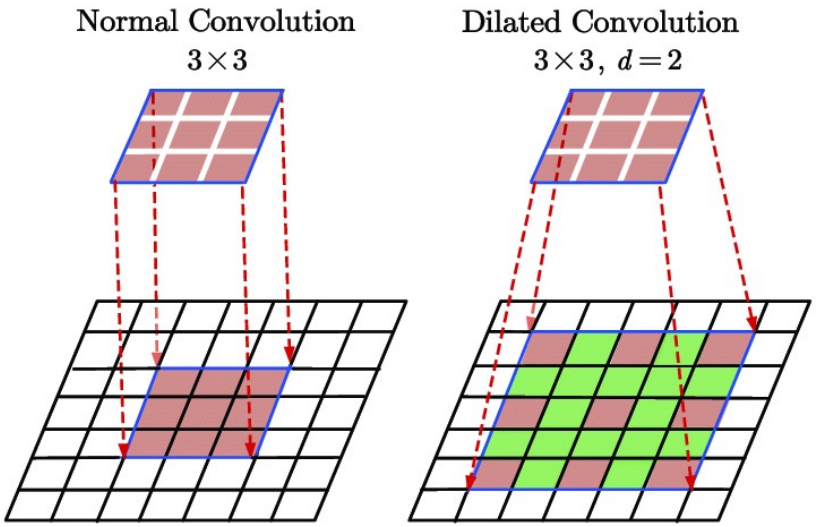
Standart Convolution:
Input Feature Map: 32 Channels
Output Feature Map: 64 Channels
Kernel Size for Convolution: 3 x 3
Standard Convolution:
Parameters =3×3×32×64
Total Parameters =18432

Depthwise Separable Convolution:
Depthwise Convolution:
Parameters= 3 x 3 x 32
Parameters=288
Pointwise Convolution:
Parameteres= 1 x 1 x32 x 64
Parameters= 2048
Total Prameters= 2336

Depthwise Separable Convolution

Depthwise separable convolutions are particularly prominent in models designed for mobile and edge computing, like the MobileNet architectures. These models are optimized for environments where computational resources, power, and memory are limited:
MobileNet Architectures: MobileNet models utilize depthwise separable convolutions extensively to provide lightweight deep neural networks. These models maintain high accuracy while being computationally efficient and small in size, making them suitable for running on mobile devices, embedded systems, or any platform where resources are constrained.
Suitability for Real-Time Applications: The efficiency of depthwise separable convolutions makes them ideal for real-time applications on mobile devices, such as real-time image and video processing, face detection, and AR and VR.

Dilated Convolution



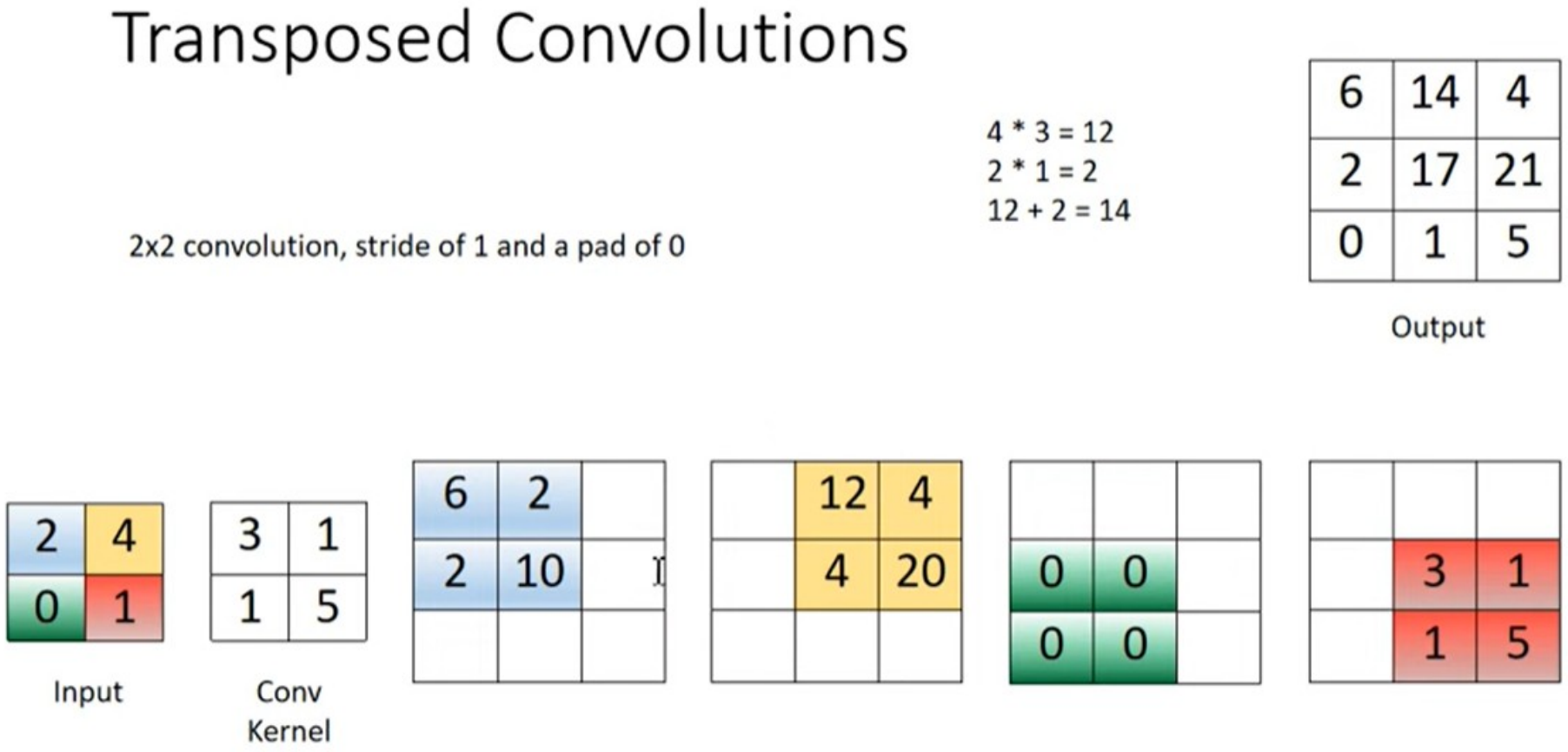
Increased Receptive Field: allows the receptive field of the network to grow exponentially with the depth of the network, rather than linearly. This is particularly useful in dense prediction tasks where contextual information from a larger area is beneficial for making accurate predictions at a pixel level.
Preservation of Resolution: Unlike pooling layers, which reduce the spatial dimensions of the feature maps, dilated convolutions maintain the resolution of the input through the network layers. This characteristic is crucial for tasks where detailed spatial relationships need to be preserved, such as in pixel-level predictions.
Efficiency: Dilated convolutions achieve these benefits without increasing the number of parameters, hence not increasing the model's complexity or the computational cost as much as increasing the kernel size directly would.

Semantic Segmentation: In semantic segmentation, the goal is to assign a class label to each pixel in an image. Dilated convolutions are extensively used in segmentation models like DeepLab, where capturing broader context without losing detail is crucial. By using dilated convolutions, these models can efficiently enlarge their receptive fields to incorporate larger contexts, improving the accuracy of classifying each pixel.

Transposed Convolution

Increase the spatial dimensions of an input tensor: It starts with the input, spreads it: adding zeros in between elements (upsampling), applies a kernel to produce a larger output.

Transposed Convolution generate or expand data dimensions, such as generating higher-resolution images from lower-resolution ones. Instead of mapping multiple input pixels into one output pixel, transposed convolution maps one input pixel to multiple outputs.



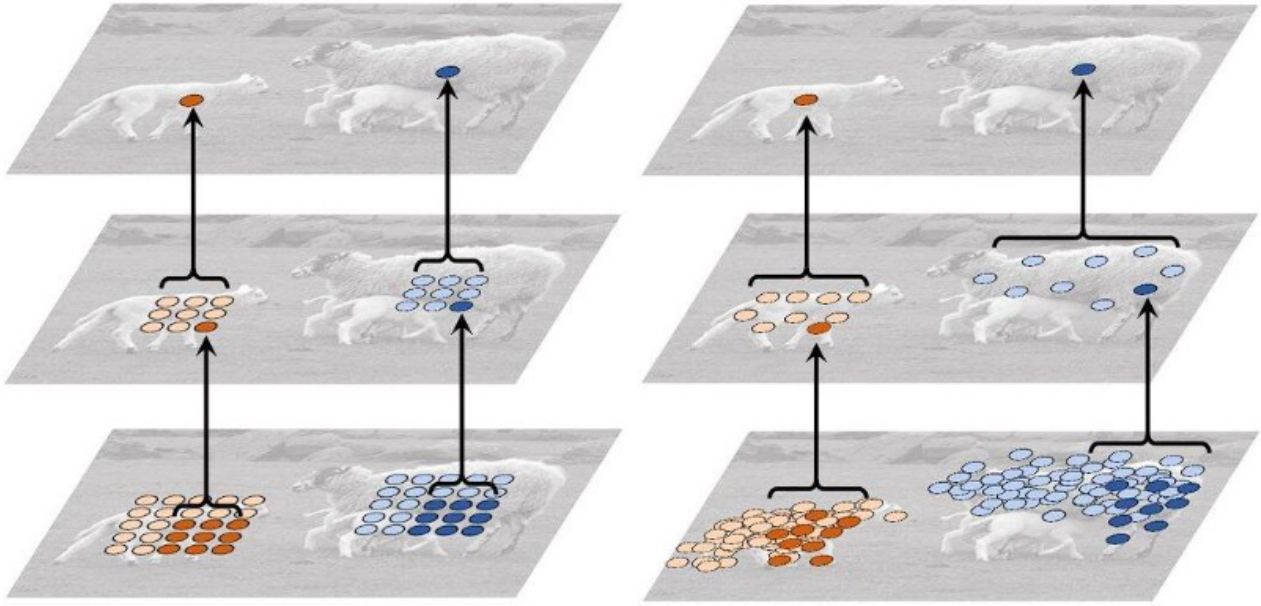
Transposed Convolution

Unlike standard convolution, where striding controls how far the filter jumps after each operation, in transposed convolution, the stride value represents the spacing between the inputs. For example, applying a filter with a stride of 2 to every second pixel in each dimension effectively doubles the dimensions of the output feature map if no padding is used.

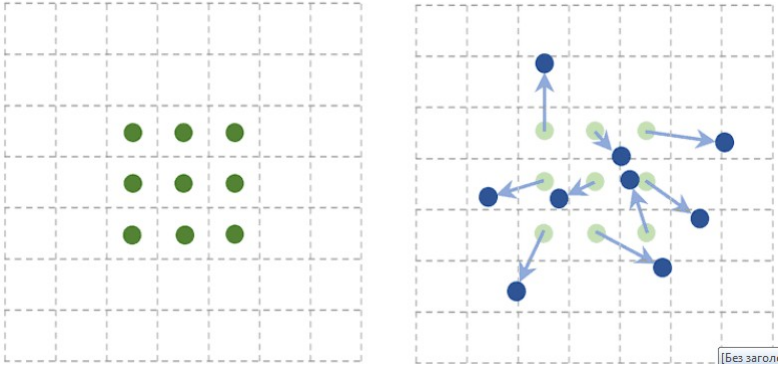
Using: The generator component of (GANs), decoder part of an AutoEncoder
In GANs, the generator starts with a random noise vector and applies several layers of transposed convolution to produce an output that has the same dimension as the desired data (e.g., generating a 64×64 image from a 100-dimensional noise vector). This process involves learning to upsample lower-dimensional feature representations to a full-resolution image.

Deformable Convolution

Deformable convolution is an advanced convolution operation that introduces learnable parameters to adjust the spatial sampling locations in the input feature map. This adaptability allows the convolutional grid to deform based on the input, making the convolution operation more flexible and better suited to handle variations in the input data.



(a) standard convolution (b) deformable convolution



Deformable convolutions have been successfully integrated into several state-of-the-art object detection frameworks, such as Faster R-CNN and YOLO, providing improvements in detecting objects with non-rigid transformations and complex orientations.

Частичная свёртка (partial convolution)

Специализированный тип свёртки, разработанный для работы с изображениями, содержащими пропущенные или маскированные области (например, для задачи *inpainting* – восстановления утраченных фрагментов).

Идея: вместе с входным изображением поступает бинарная *маска*, отмечающая допустимые пиксели (1 – доступен, 0 – пропуск).

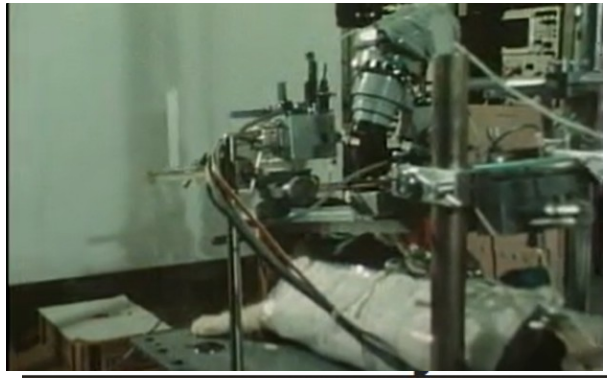
Частичная свёртка при вычислении каждого выходного пикселя учитывает *только те входные значения, где маска = 1*, игнорируя отсутствующие данные.

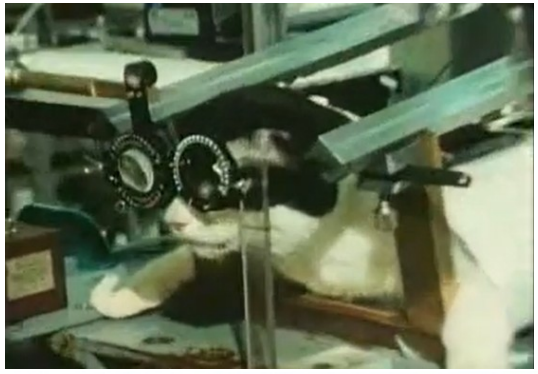
Формально, для окна входа XXX и бинарной маски MMM такого же размера выход считается как:

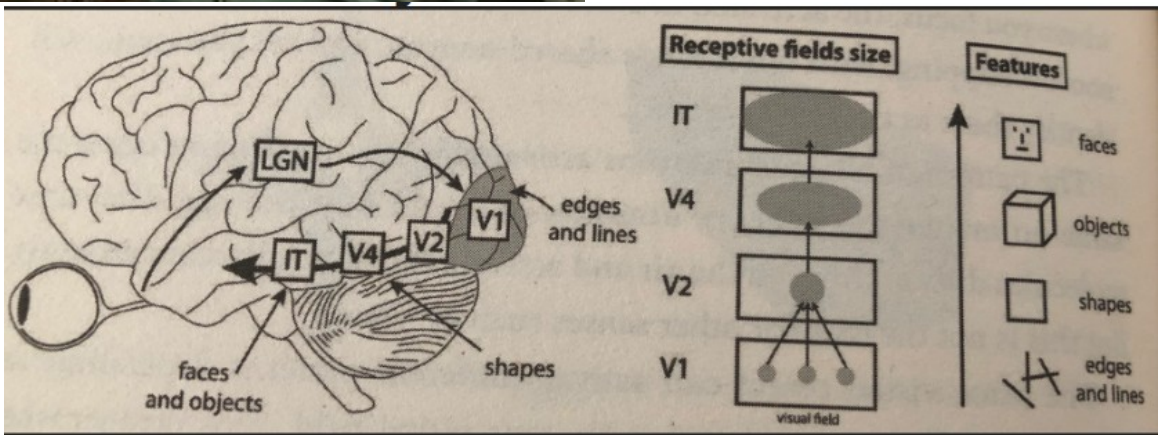
История возникновения сверток в искусственных нейронных сетях (CNN)

1957: Perceptron (Frank Rosenblatt)

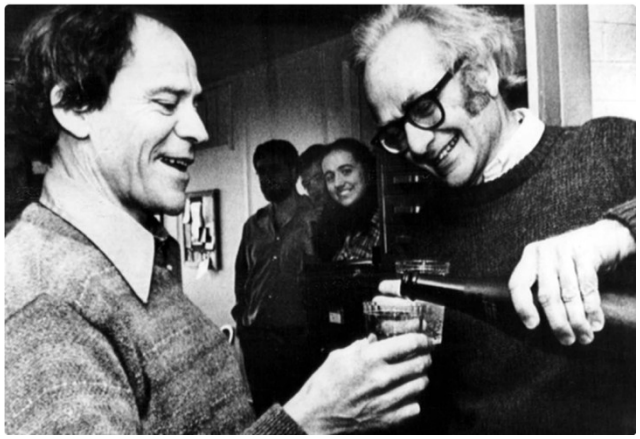
1958: David Hubel and Torsten Wiesel







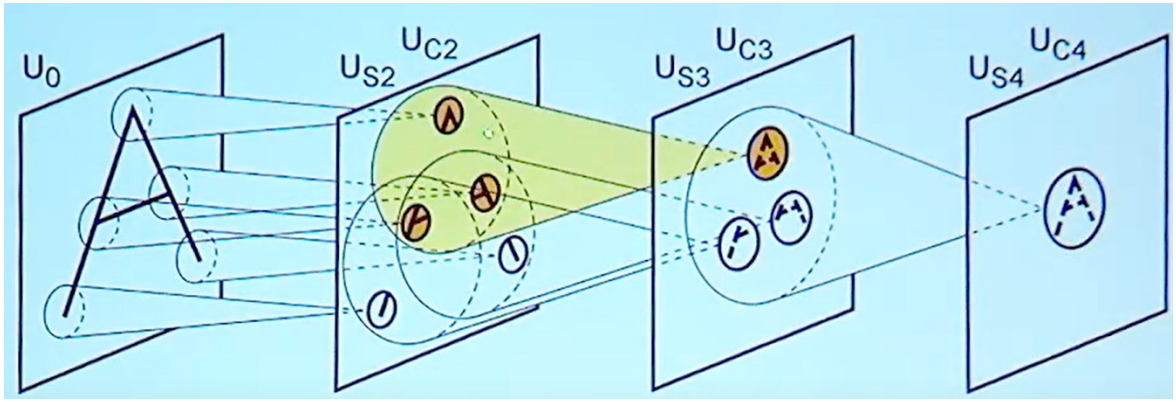
D H Hubel, T N Wiesel. Receptive fields of single neurones in the cat's striate cortex. J Physiol 1959;148:574-591.



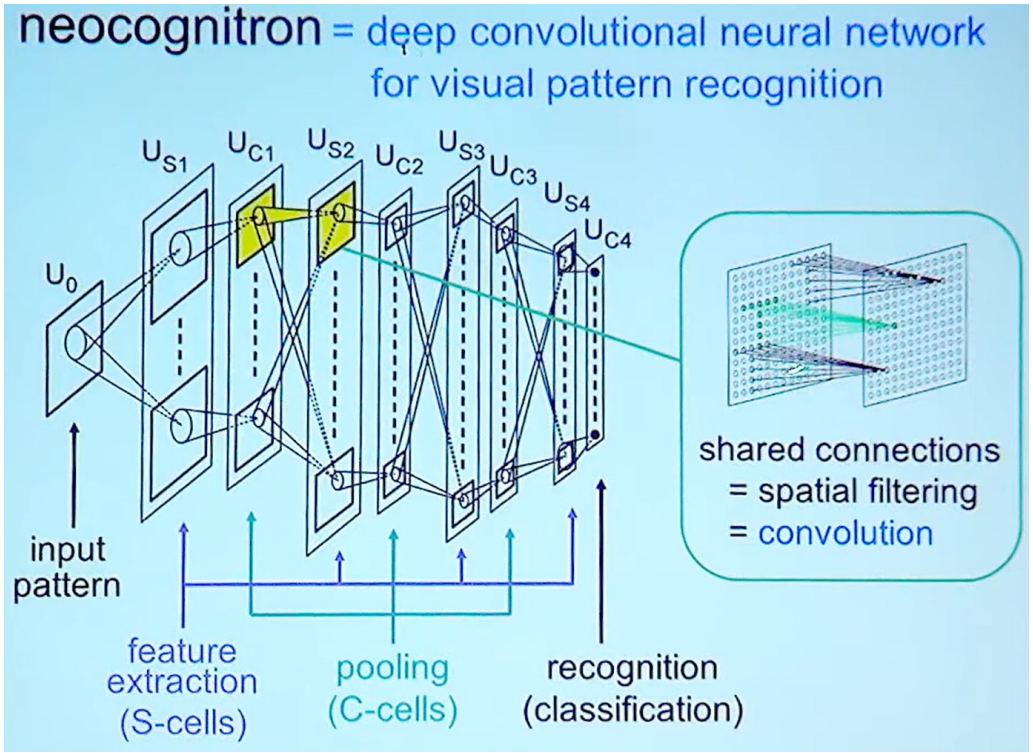
1981: the Nobel prize for physiology and medicine, along with Roger Sperry

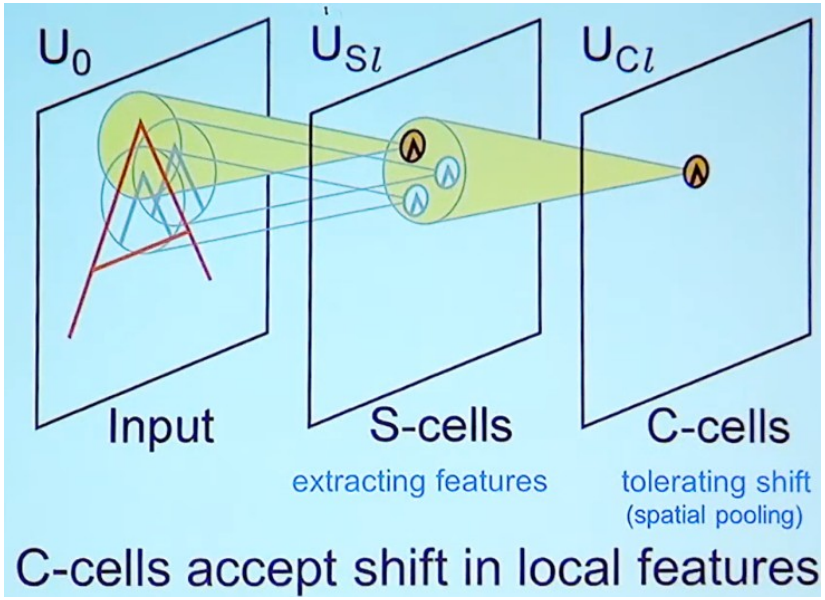
Simple cell of S-cell → **Complex cell of C-cell**

1979: Kunihiko Fukushima: Neocognitron



Fukushima, K. and Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Competition and cooperation in neural nets (pp. 267-285). Springer, Berlin, Heidelberg.



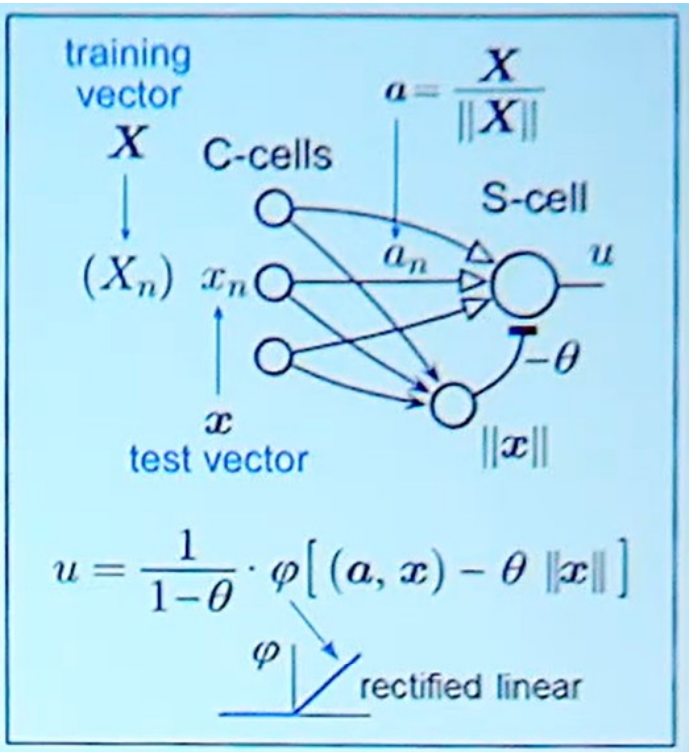


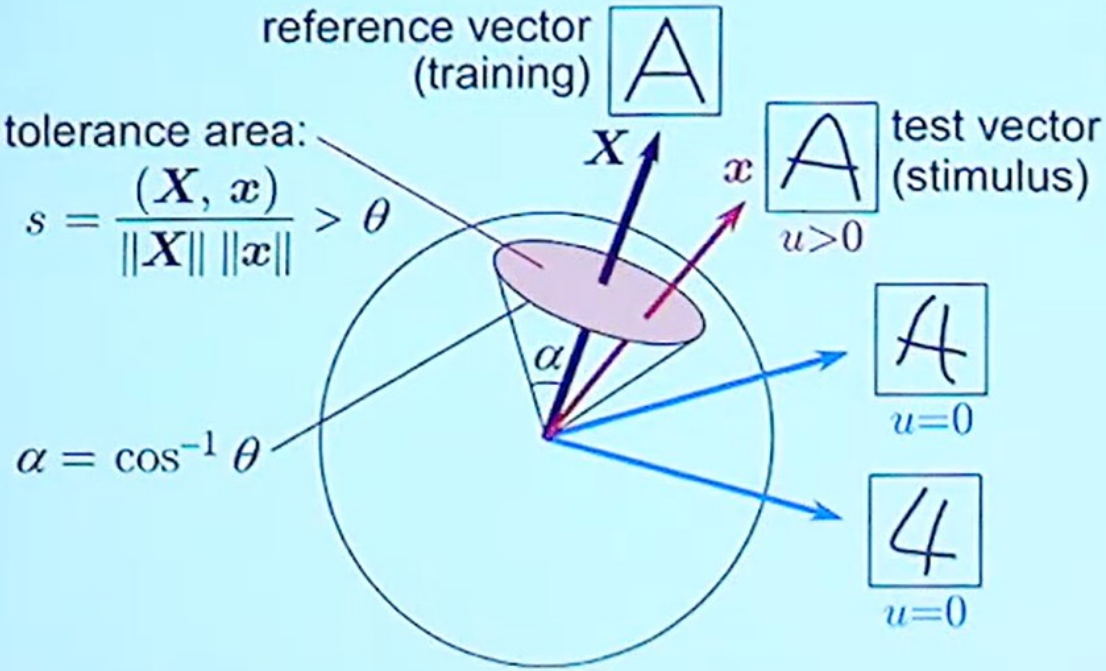
Feature extraction by an S-cell

$$u = \|\boldsymbol{x}\| \, \frac{\varphi[s-\theta]}{1-\theta}$$

$$s = \frac{(\boldsymbol{X}, \boldsymbol{x})}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|}$$

similarity between reference vector $\boldsymbol{X}$ and test vector $\boldsymbol{x}$



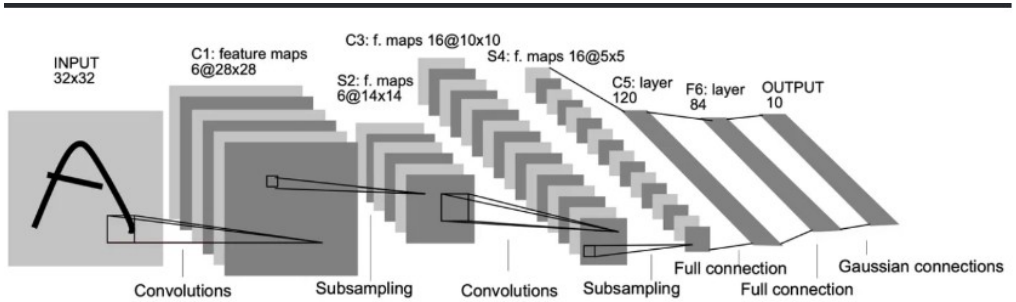


1989: Training CNN by Backpropagation

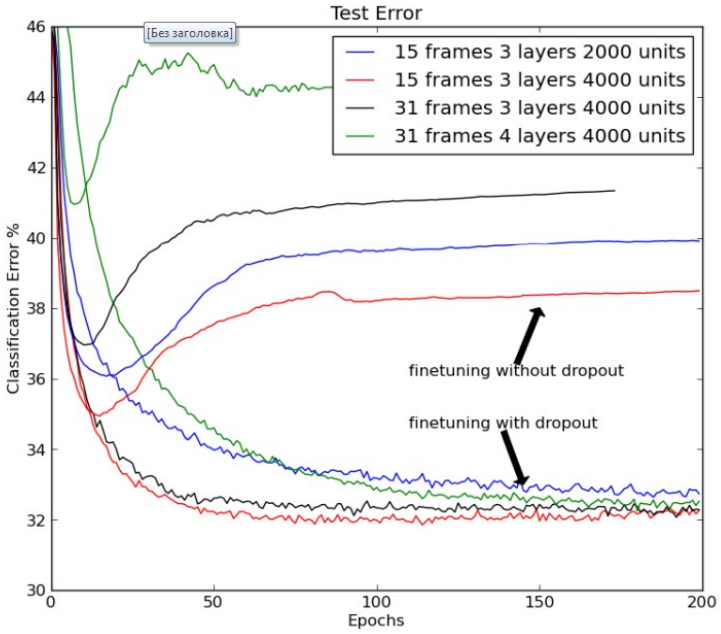
LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), pp.541-551.

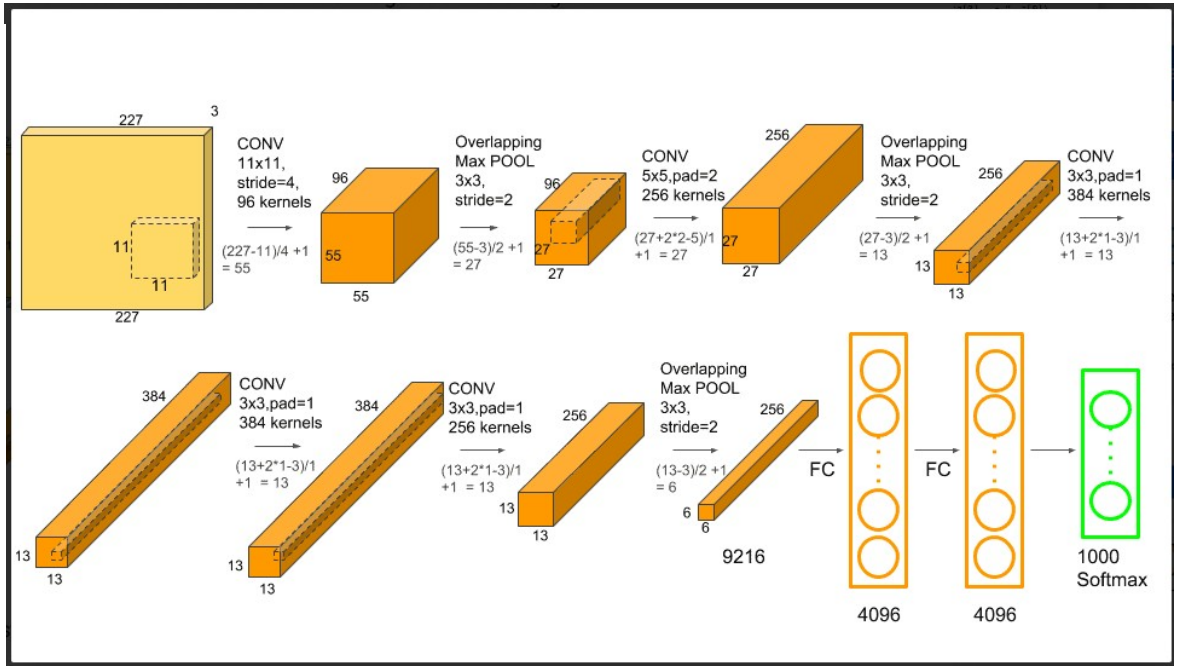
1998: End-to-end training of CNN: LeNet-5

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.



2012: AlexNet: Dropout+ Augmentation+ReLu + 2 x GPU Nvidia 3090





2014: VGG: Visual Geometry Group, University of Oxford

- Использование очень простых и повторяющихся блоков: свёртки размером только **3×3** и MaxPool 2×2.
- Показано, что глубокие сети с малым ядром (3×3) эффективнее по сравнению с более сложными свёртками.

2014: GoogLeNet/Inception (Google)

•Введение **Inception-модуля** — блока, параллельно вычисляющего несколько свёрток (1×1, 3×3, 5×5) и объединяющего результаты.
•Применение 1×1 свёрток для снижения вычислительной сложности («bottleneck»).

2015: ResNet (Microsoft)

•Введение **Residual Blocks**: добавление skip-соединений, позволяющих сети обучаться разностям между входом и выходом.
•Возможность обучения очень глубоких нейросетей (50, 101, 152 слоя и более).
•Практически решила проблему затухающего градиента.
•Стала базой для большинства современных архитектур CNN.

2016: DeepLab (Google), WaveNet (DeepMind):

•Введение **Dilated Convolution (Atrous Conv)** для сегментации изображений.
•Dilated convolution для генерации аудио и временных последовательностей, нашло применение и в CV.

2017: MobileNet, «Attention is All You Need» (Google):

•Depthwise Separable Convolution: мощные и компактные CNN для мобильных устройств и реального времени.
•Представлена архитектура Transformer (изначально для NLP), позднее сильно повлияла на Computer Vision.

2018:

•**EfficientNet (Google)**:
•Систематический подход масштабирования сети (depth, width, resolution).
•**Mask R-CNN (Facebook)**:
•Одновременная детекция и сегментация объектов (instance segmentation).

2019:

•**Vision Transformer (ViT)** (2020, начало исследований с 2019):
•Архитектура Transformer заменяет CNN в задачах классификации изображений.
•**Gated Convolution**: Динамическое управление важностью признаков (Image Inpainting).

2020:

CLIP и DALL-E (OpenAI):

Связь текста и изображений через contrastive learning, мощные мультимодальные модели.

YOLOv5 (Ultralytics):

Эффективная и точная real-time object detection сеть.

2021:

Swin Transformer (Microsoft):

Иерархический Transformer с окнами внимания, эффективен в распознавании и сегментации изображений.

2022:

Stable Diffusion и диффузионные модели:

Открытые модели, позволившие массово генерировать фотореалистичные изображения.

Segment Anything Model (SAM, Meta):

Универсальная модель сегментации любых объектов (zero-shot segmentation).

Период (год)	Название события	Инновация (что предложено)	Модель или Продукт
1957	Перцептрон (F. Rosenblatt)	Первая обучаемая модель нейрона	Perceptron
1958–1959	Эксперименты D. Hubel и T. Wiesel	Открытие иерархической структуры зрительной коры	Simple/Complex cells (V1-V4, cortex)
1979–1980	Neocognitron (K. Fukushima)	Первая сверточная нейросеть с иерархией	Neocognitron
1986	Метод обратного распространения ошибки (backpropagation)	Эффективный алгоритм обучения многослойных нейросетей	Backpropagation
1989–1998	LeNet и LeNet-5 (Yann LeCun)	Первая успешная CNN с обучением backpropagation	LeNet, LeNet-5
2006	Алгоритм обучения глубоких сетей (Hinton et al.)	Послойная предварительная тренировка (Pretraining)	Deep Belief Networks (DBN)
2012	AlexNet (Krizhevsky et al.)	Глубокая CNN, GPU-обучение, глубокое обучение	AlexNet
2014	VGG (Oxford Visual Geometry Group)	Простые и глубокие CNN (3×3 свёртки)	VGGNet (VGG16, VGG19)
2014	GoogLeNet/Inception (Google)	Многоуровневые параллельные модули (Inception-блок)	GoogLeNet (Inception-v1–v4)
2015	ResNet (Microsoft Research)	Остаточные связи (Residual connections), глубокие CNN	ResNet (ResNet-50, 101, 152)
2016	DeepLab (Google)	Dilated Convolution (Atrous Convolution)	DeepLab
2017	MobileNet (Google)	Depthwise Separable Convolution (эффективные CNN)	MobileNet
2017	Transformer (Google)	Архитектура Attention (изначально NLP)	Transformer
2018	Mask R-CNN (Facebook)	Детекция и сегментация объектов одновременно	Mask R-CNN
2019–2020	Vision Transformer (Google, 2020)	Transformer в задачах классификации изображений	ViT (Vision Transformer)
2020	CLIP, DALL-E (OpenAI)	Связь текста и изображений, генеративные модели	CLIP, DALL-E
2021	Swin Transformer (Microsoft)	Иерархический Transformer для изображений	Swin Transformer
2022	Segment Anything Model (Meta)	Универсальная сегментация объектов (zero-shot)	SAM (Segment Anything Model)