



Unione europea
Fondo sociale europeo



Regione Emilia-Romagna



Educazione
Ricerca
Emilia-Romagna

DATA LAB

GUARDA AVANTI

Big Data, nuove competenze
per nuove professioni.



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA



ALMA MATER STUDIORUM
UNIVERSITAS BOLOGNENSIS



Università
degli Studi
di Ferrara



UNIVERSITÀ
DI PARMA



POLITECNICO
MILANO 1863
POD TERRITORIALE DI
PACENZA



UNIVERSITÀ
CATTOLICA
del Sacro Cuore

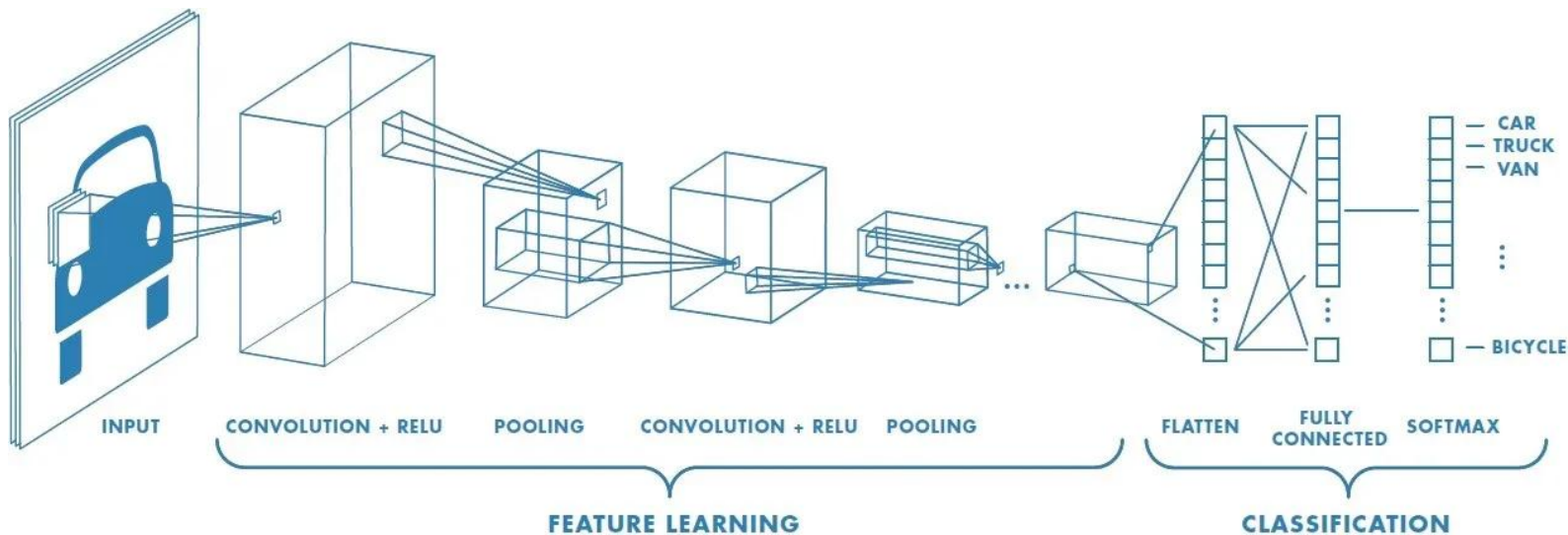
"Anticipare la crescita con le nuove competenze sui Big Data - Edizione 3" Operazione Rif. PA 2021-16029/RER approvata con DGR
n° 927 del 21 giugno 2021 e co-finanziata dal Fondo Sociale Europeo PO 2014-2020 Regione Emilia-Romagna

Deep Learning

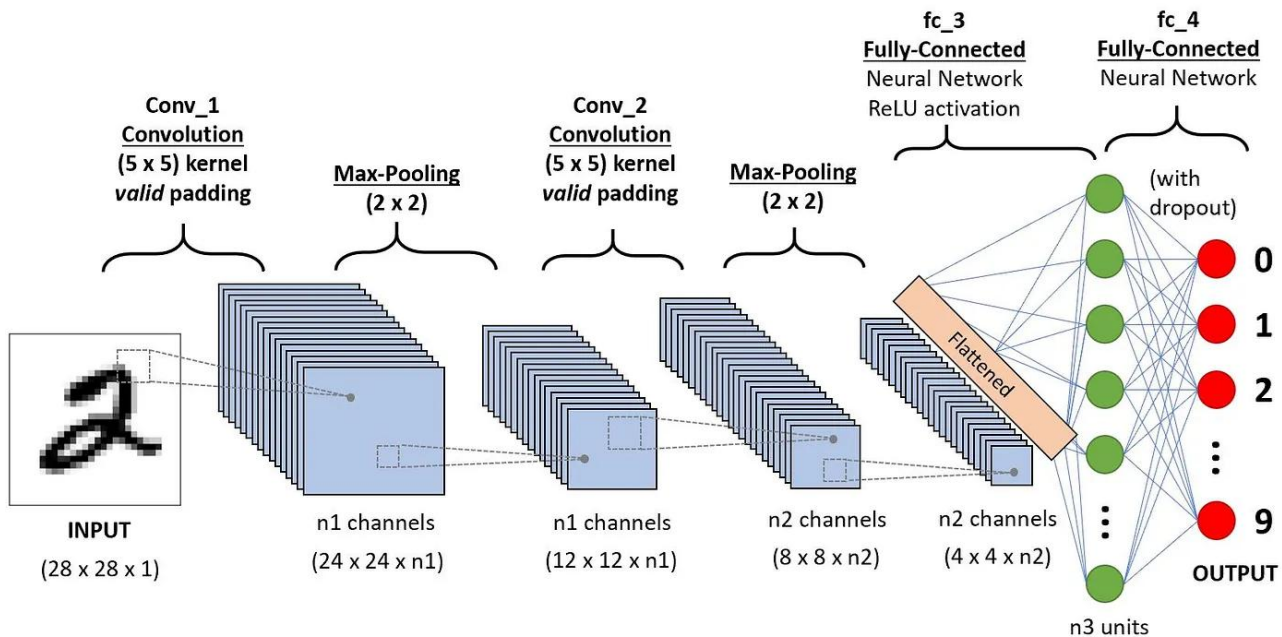


What is CNN?

- Convolutional Neural Network is a Deep Learning algorithm that can take an input image, and process the image, and classify one from the other.
- The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.
- Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.



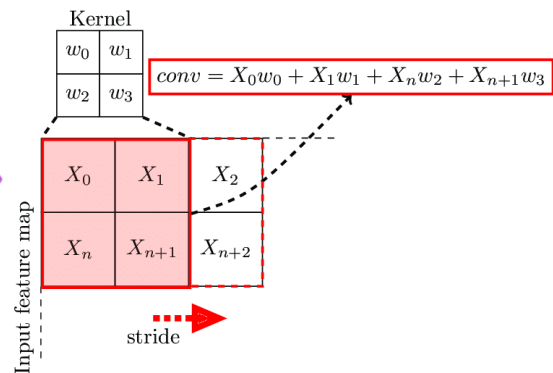
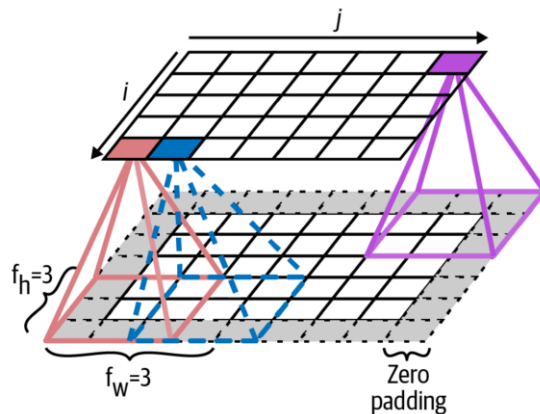
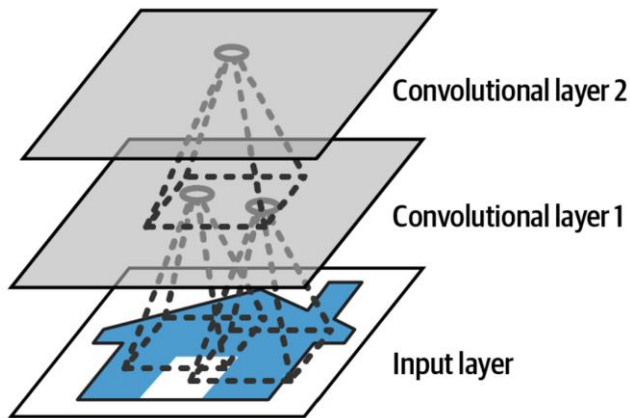
What is CNN?



CNN Terms

- CNN Layers
- Filters and Features Map
- Stride
- Padding
- Convolution Operation
- Pooling Layers
- Dropout Layers
- Activation Function
- Fully Connected Layers

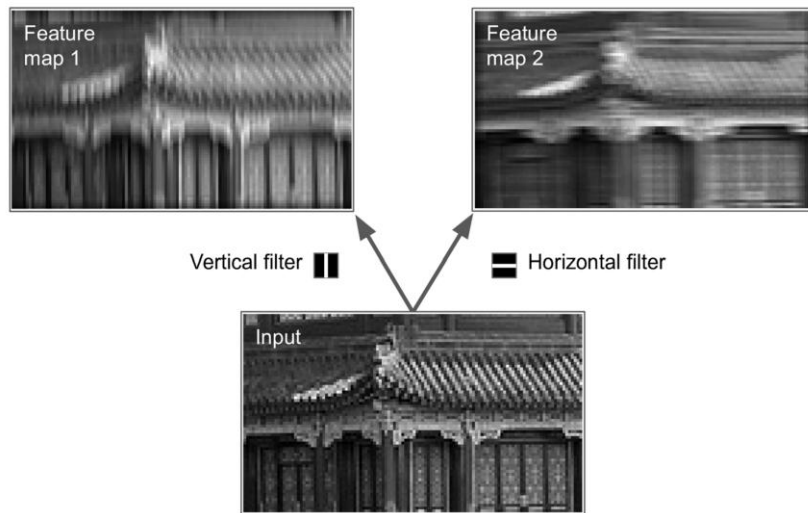
Convolutional Layer



- A convolution is a mathematical operation that slides one function over another and measures the integral of their pointwise multiplication
- Neurons in the first convolutional layer are not connected to every single pixel in the input image
- Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer

Filter (Kernel) and Feature Map

- A filter is a matrix of numbers that is smaller than the input image.
- The filter slides over the input image and performs a dot product operation with the pixels it overlaps.
- The resulting output value is added to a new matrix, which is called a feature map.
- Each filter is designed to extract a specific feature from the input image, such as edges, corners, or blobs.
- The values of the filter are learned during the training process of the CNN



Filter (Kernel) and Feature Map

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	1	1	0	0

Image

4	3	

Convolved
Feature

1	1	1 _{x1}	0 _{x0}	0 _{x1}
0	1	1 _{x0}	1 _{x1}	0 _{x0}
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1	1	0
0	1	1	0	0

Image

4	3	4

Convolved
Feature

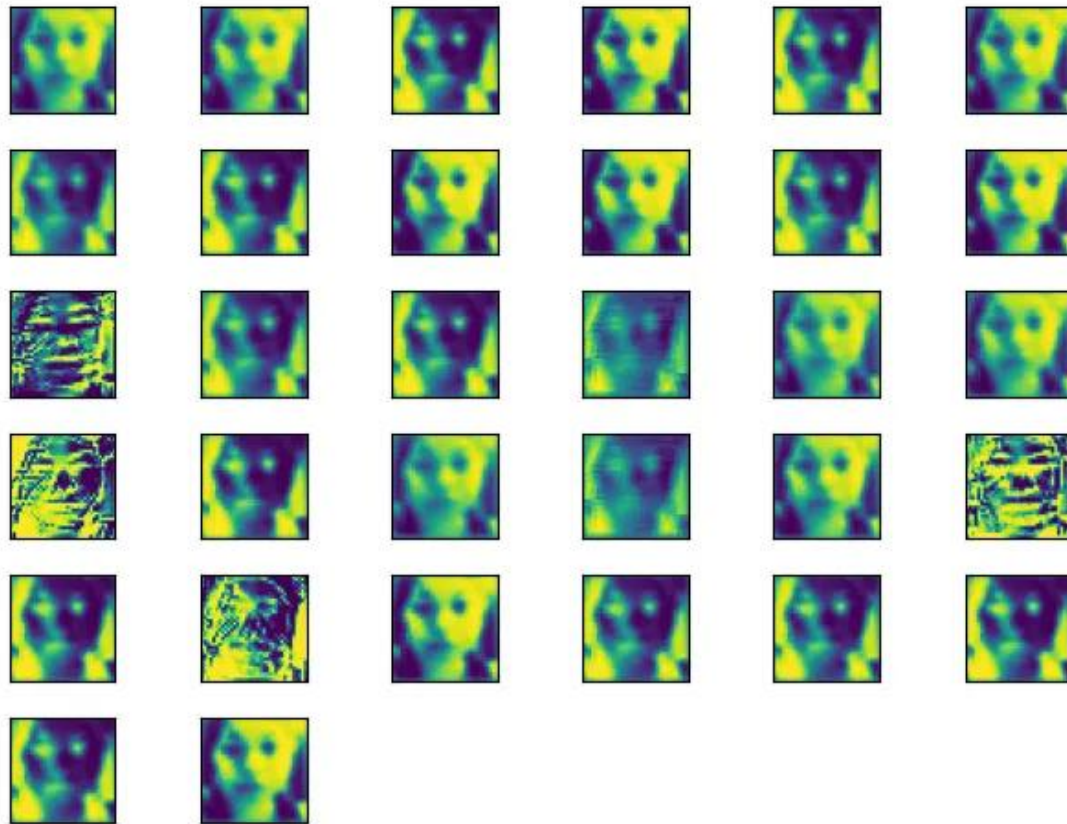
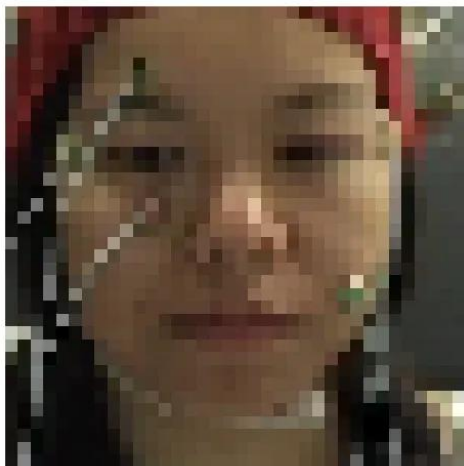
1	1	1	0	0
0 _{x1}	1 _{x0}	1 _{x1}	1	0
0 _{x0}	0 _{x1}	1 _{x0}	1	1
0 _{x1}	0 _{x0}	1 _{x1}	1	0
0	1	1	0	0

Image

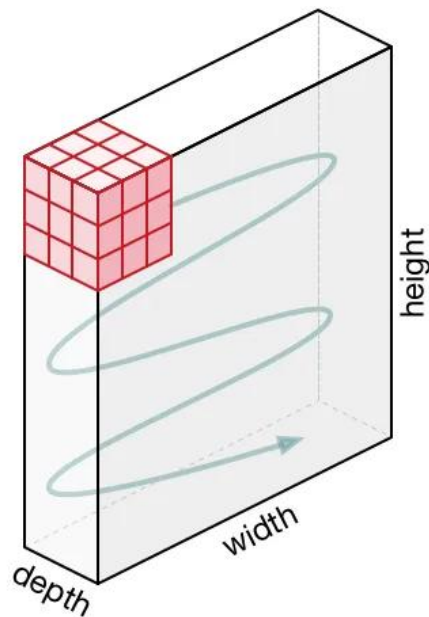
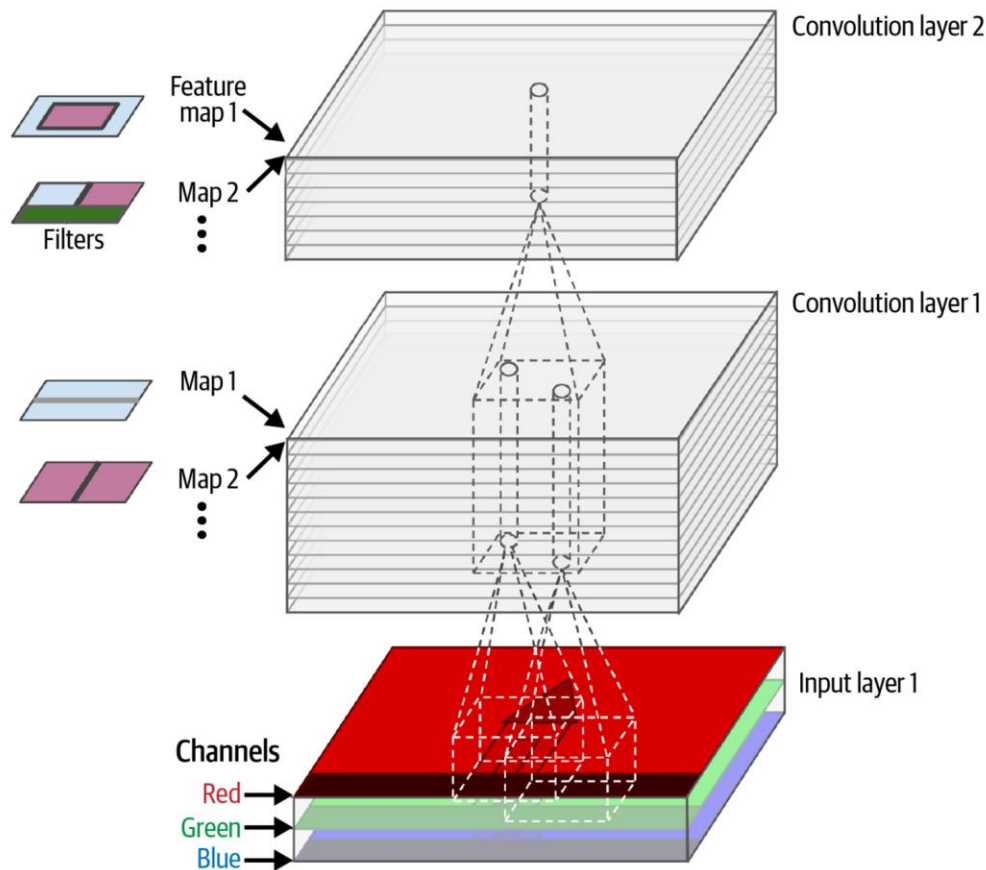
4	3	4
2		

Convolved
Feature

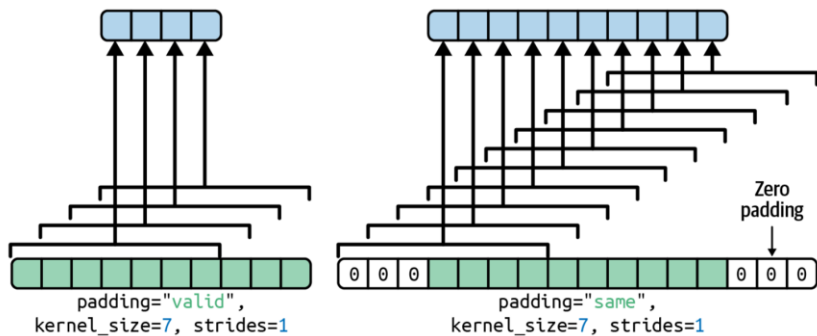
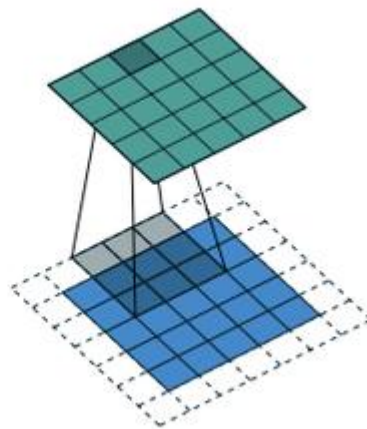
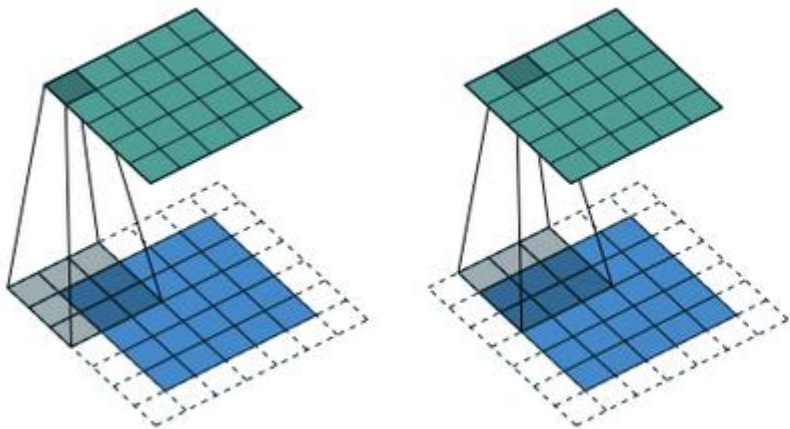
Filter (Kernel) and Feature Map



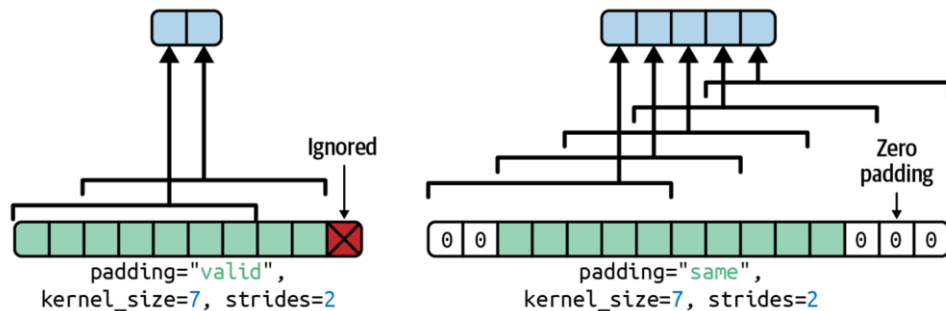
Filter (Kernel) and Feature Map



Padding and Strides



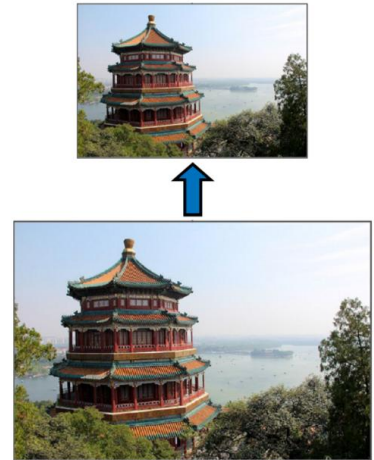
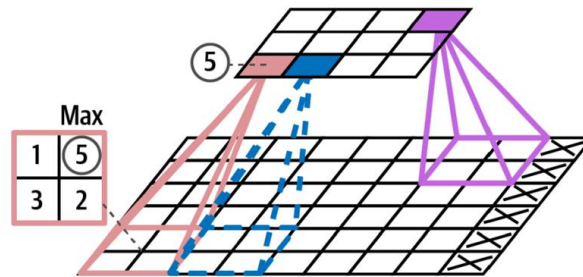
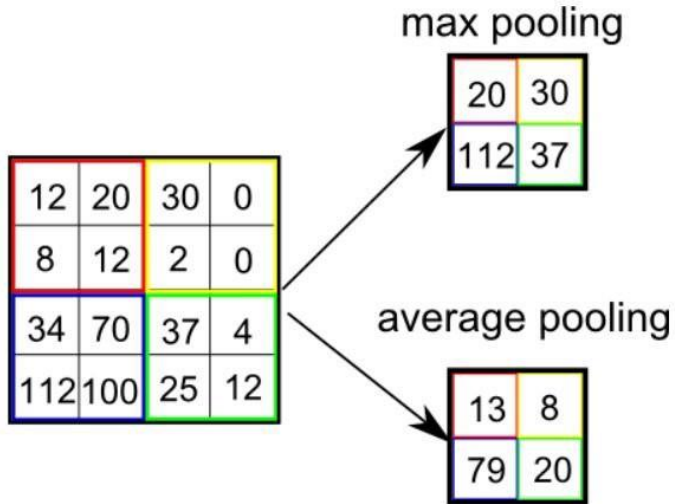
The two padding options, when strides=1



The two padding options, when strides=2


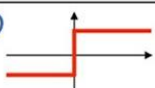
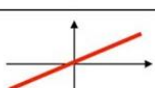

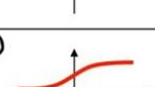

Pooling Layer

- Their goal is to reduce the computational load, the memory usage, and the number of parameters
- There are two types Max Pooling Layer and Average Pooling Layer
- It is useful for extracting dominant features

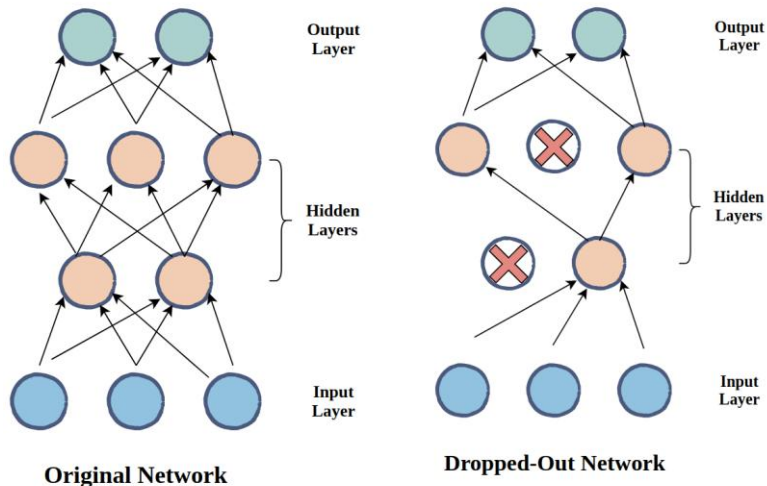


Activation Function

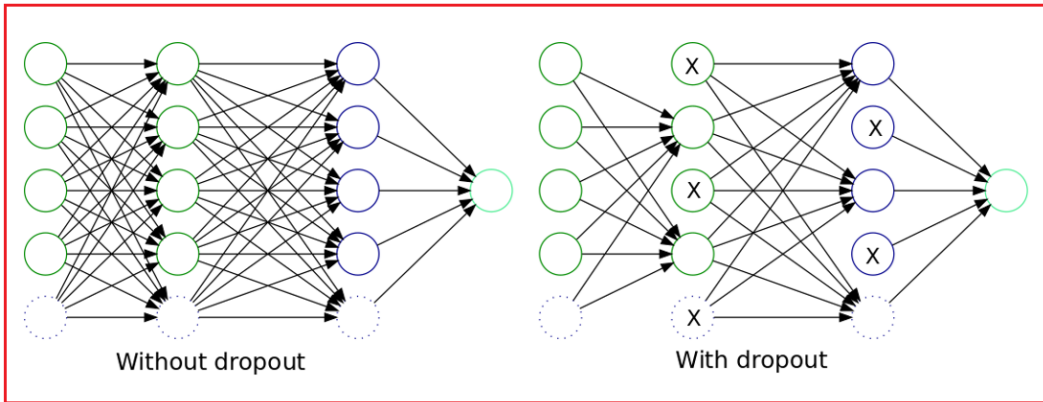
- The function introduces non-linearity to the network and enables it to learn complex patterns and relationships between the input and output.
- An activation function determines whether a neuron should be activated or not based on the input it receives.

Commonly Used Activation Functions			Range
1. Step function: $f(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$	①		$\{0, 1\}$
2. Signum function: $f(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	②		$\{-1, 1\}$
3. Linear function: $f(z) = x$	③		$(-\infty, \infty)$
4. ReLU function: $f(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$	④		$(0, \infty)$
5. Sigmoid function: $f(z) = \frac{e^x}{1+e^x}$	⑤		$(0, 1)$
6. Hyperbolic tan: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	⑥		$(-1, 1)$

Dropout

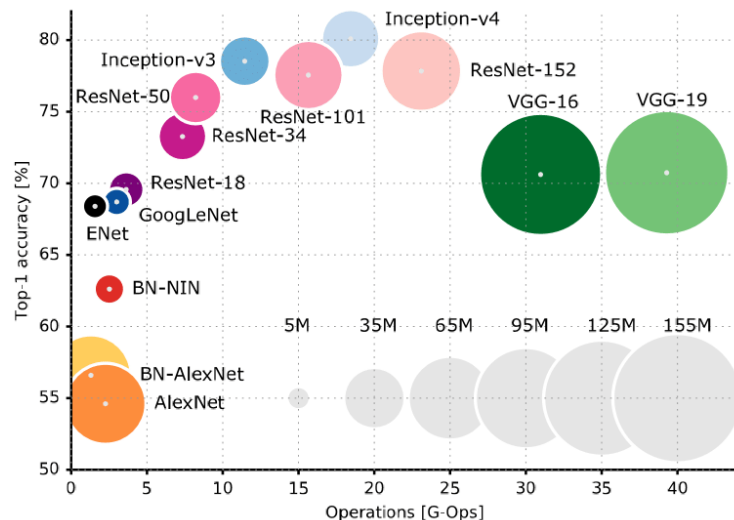
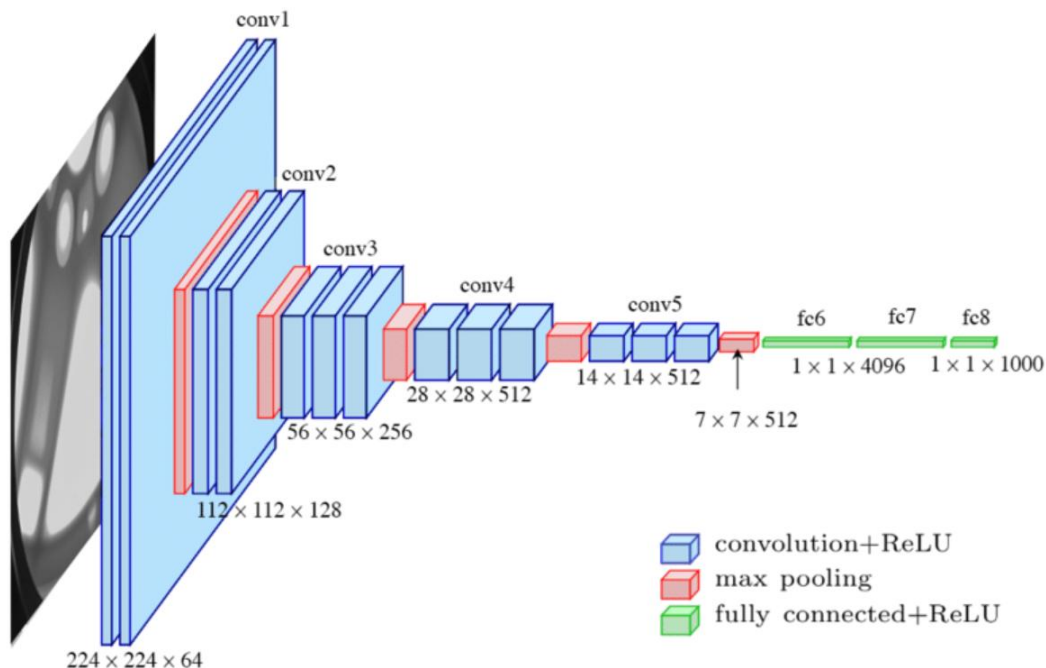


- Dropout is a regularization technique used in neural networks to prevent overfitting.
- It is a layer that randomly drops out or deactivates a specified number of neurons in the network during training
- Dropout is a technique used to prevent overfitting in neural networks by randomly deactivating neurons during training.



CNN Architectures

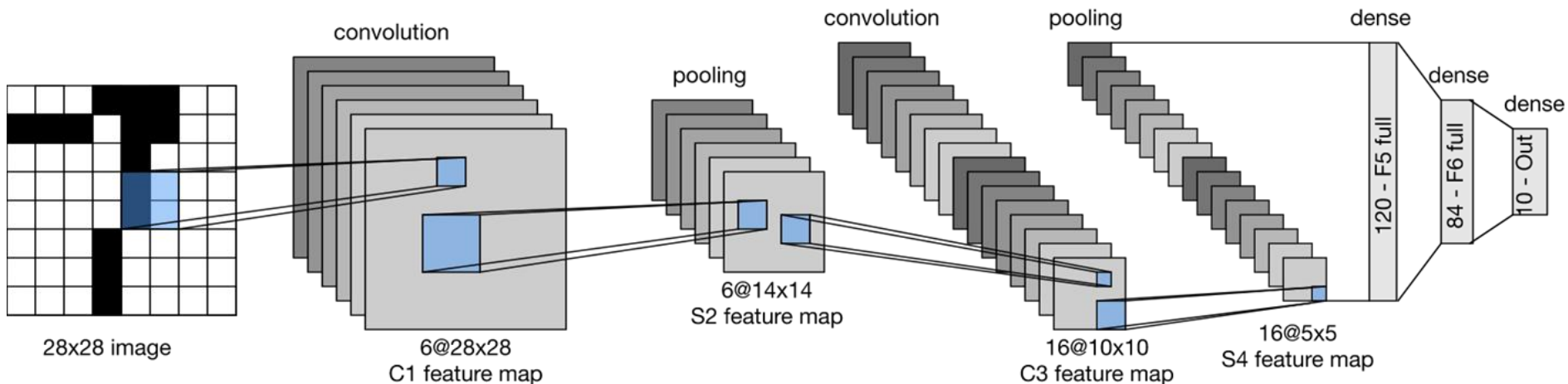
- Typical CNN architectures stack a few convolutional layers [+ReLU] then pooling layers, then another few convolutional layers [+ReLU] then another pooling layer so on.
- Do not use large kernel size. Typical size is 5x5 for first stage and then 3x3 is used



LeNet-5 [1998]

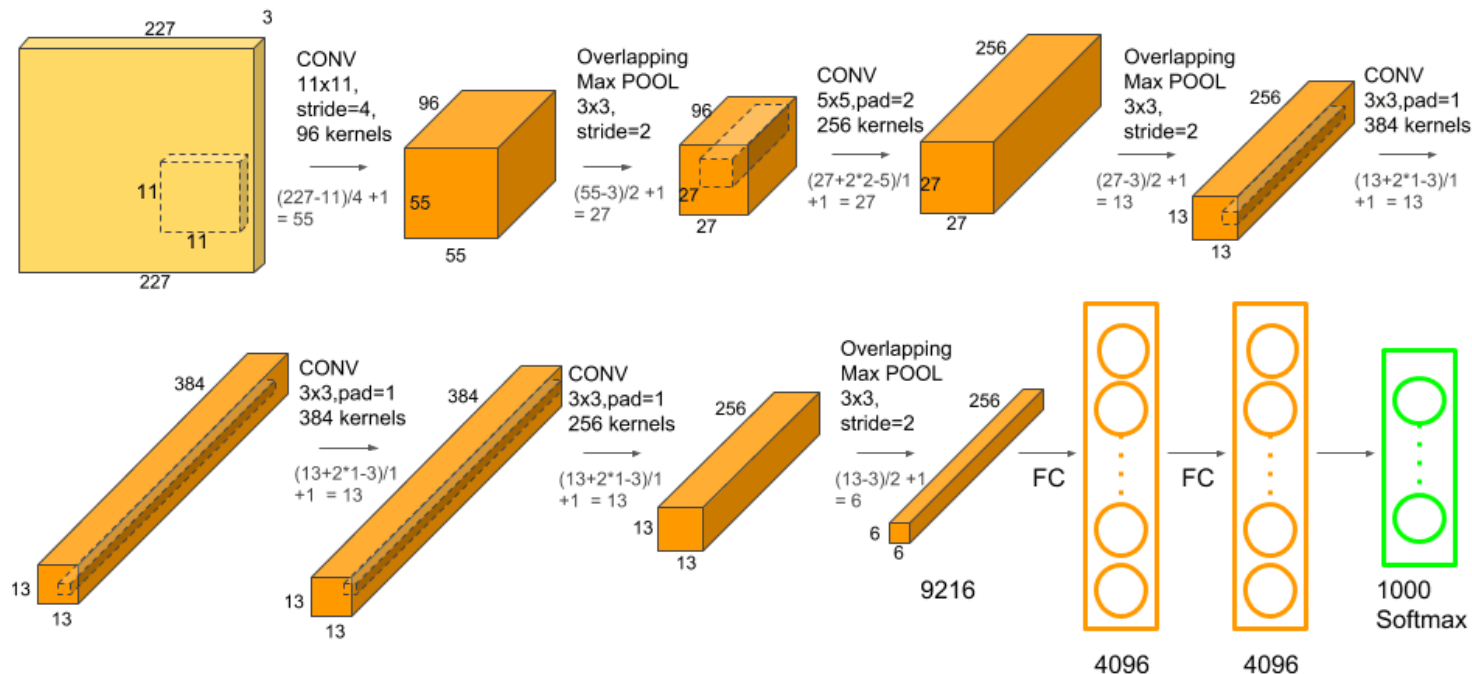
- It has been widely used for handwritten digit recognition.
- Major difference, now days we use ReLu
- It was used to recognize signature in bank cheques

Layer	Type	Maps	Size	Kernel size	Stride	Activation
Out	Fully connected	—	10	—	—	RBF
F6	Fully connected	—	84	—	—	tanh
C5	Convolution	120	1×1	5×5	1	tanh
S4	Avg pooling	16	5×5	2×2	2	tanh
C3	Convolution	16	10×10	5×5	1	tanh
S2	Avg pooling	6	14×14	2×2	2	tanh
C1	Convolution	6	28×28	5×5	1	tanh
In	Input	1	32×32	—	—	—



AlexNet [2012]

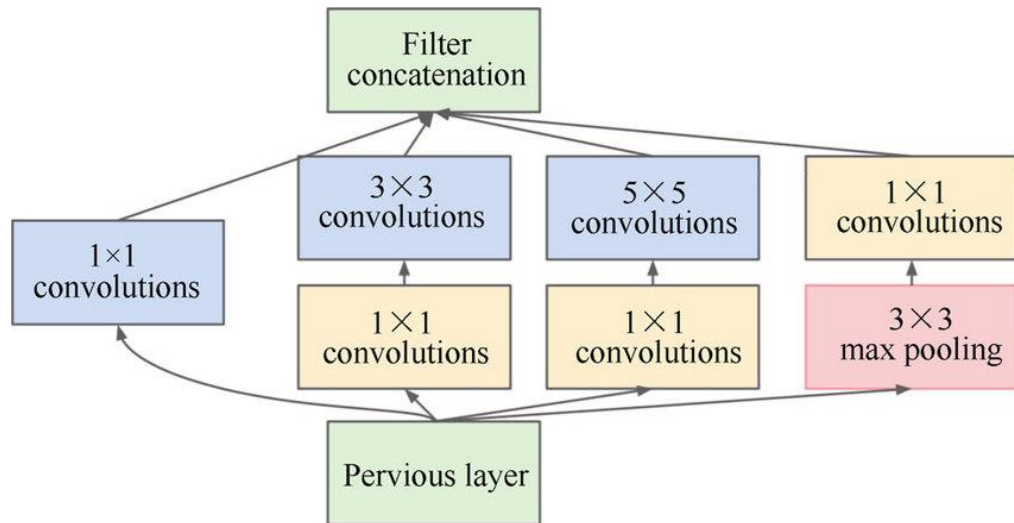
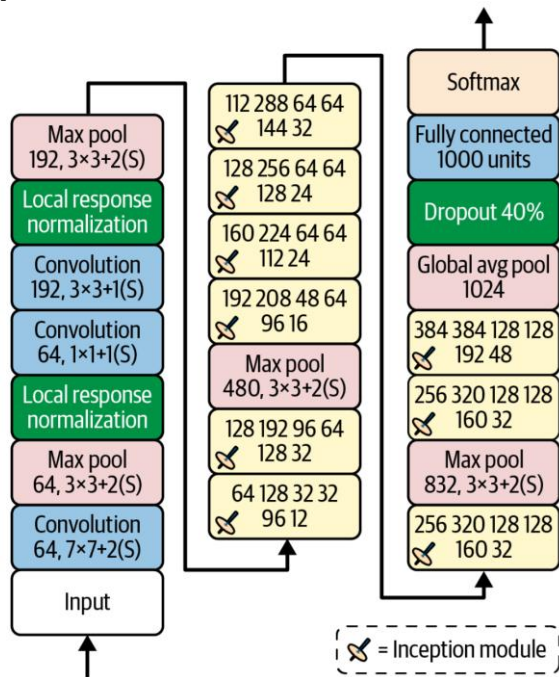
- The AlexNet CNN won the 2022 ILSVRC challenge by a large margin
- It is similar to LeNet-5, only much larger and deeper
- it was the first to stack convolutional layers directly on top of one another



GoogLeNet [2014]

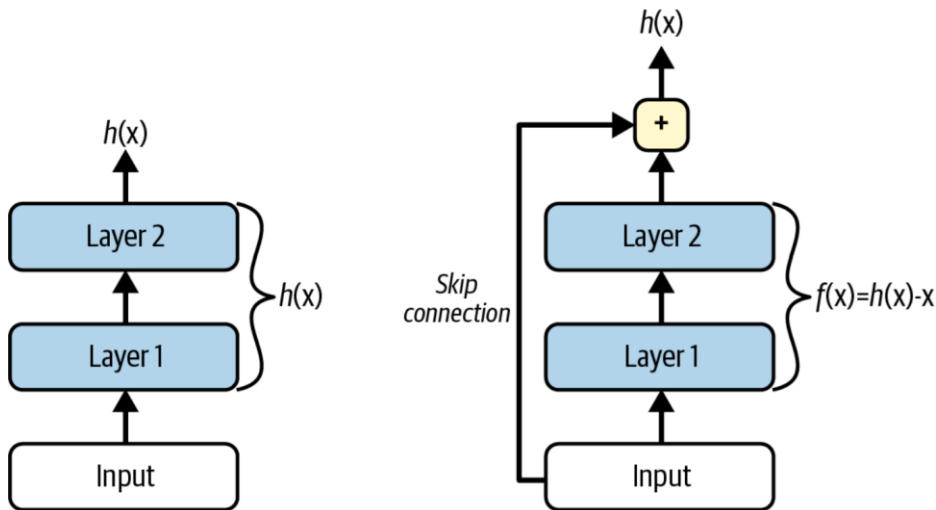
- The GoogLeNet architecture was developed by scientists at Google and it won the ILSVRC 2014 challenge by pushing the top-five error rate below 7%
- It was much deeper and used inception module
- The 1×1 convolution is used to reduce the depth or number of channels of the input

vc

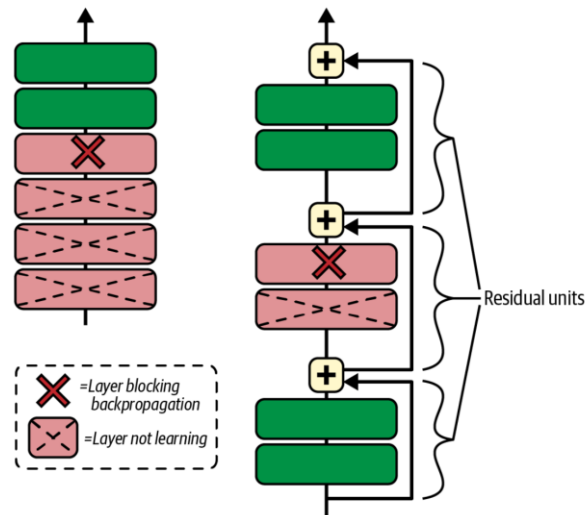


ResNet [2015]

- It has won the ILSVRC 2015 challenge using a Residual Network.
- The winning variant used an extremely deep CNN composed of 152 layers
- Models were getting deeper and deeper, with fewer and fewer parameters.
- It uses skip connections to reduce #params

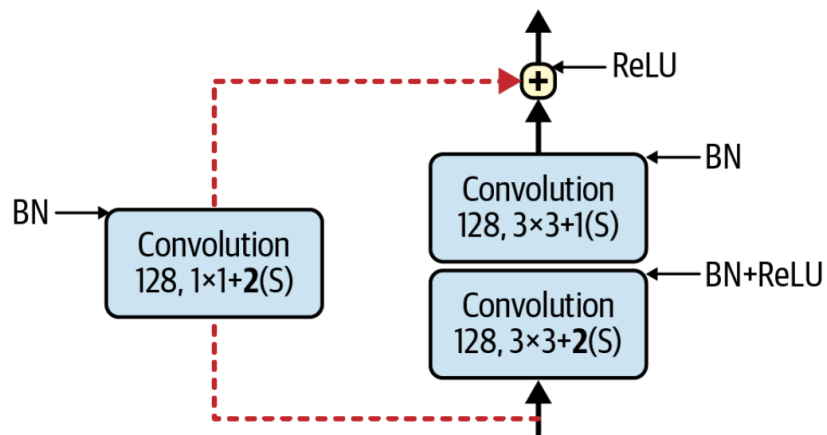


Residual learning

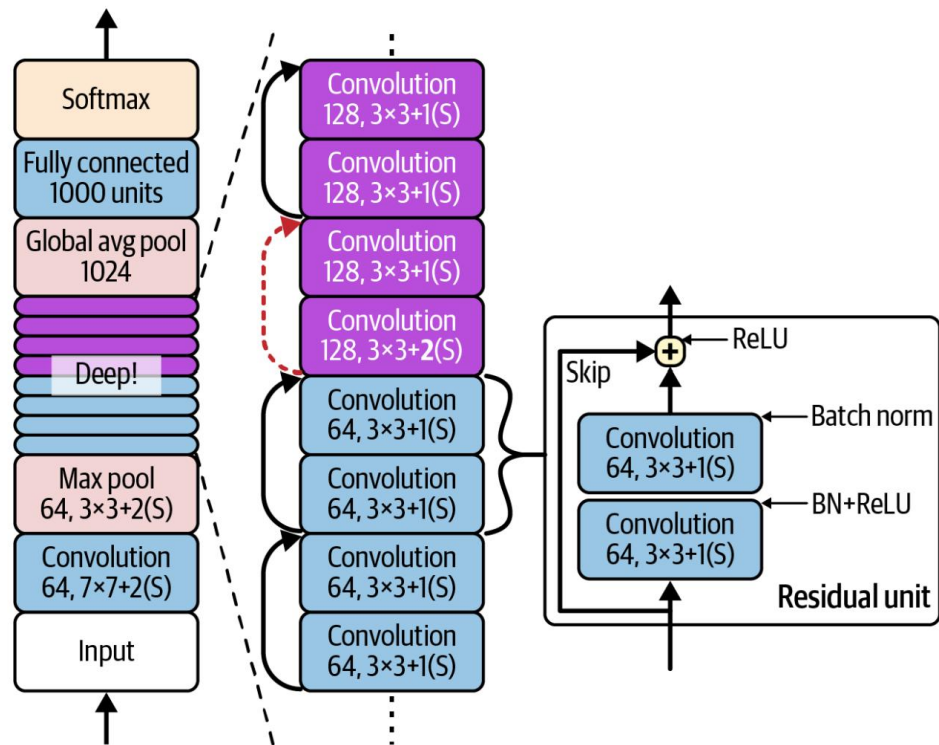


Regular vs Resnet

ResNet [2015]



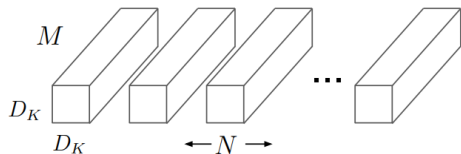
**Skip connection when changing
feature map size and depth**



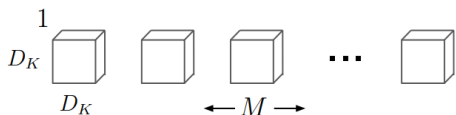
Resnet Architecture

MobileNet [2017]

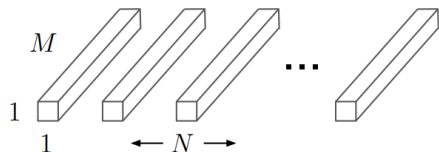
- Efficient Convolutional Neural Networks for Mobile Vision Applications
- Depthwise convolution applies a single filter to each input channel
- The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Table 5. Narrow vs Shallow MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.75 MobileNet	68.4%	325	2.6
Shallow MobileNet	65.3%	307	2.9

Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

EfficientNet [2020]

- EfficientNet was developed by scaling up the base architecture using a compound scaling method that optimizes network depth, width, and resolution.
- The compound scaling method involves scaling the depth, width, and resolution of the network together, rather than independently, to achieve better performance with fewer parameters.

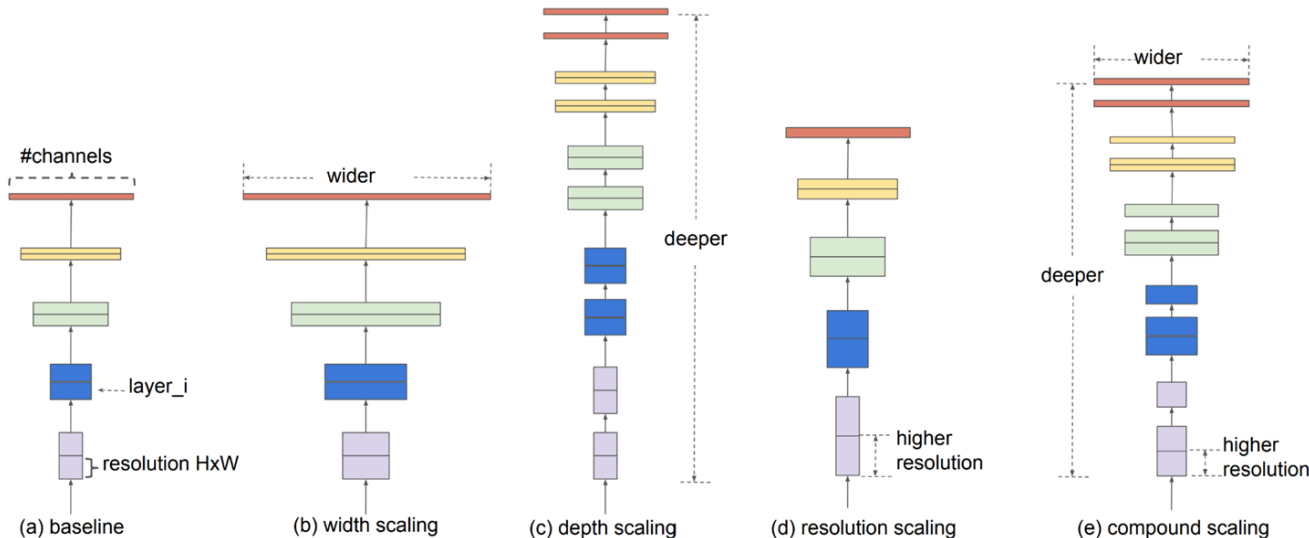


Figure 2. Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

EfficientNet [2020]

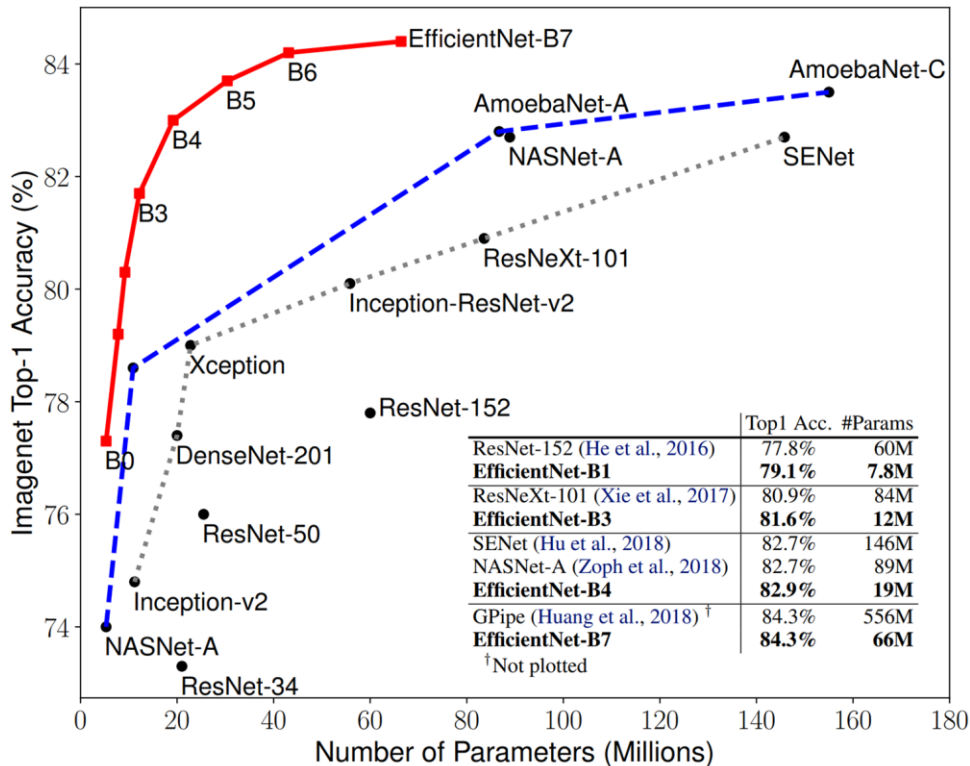


Table 1. EfficientNet-B0 baseline network – Each row describes a stage i with \hat{L}_i layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels \hat{C}_i . Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1