UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

Sesión 1.1

Redes convolucionales

VGG, ResNet, MobileNet

TRANSFORMATEC

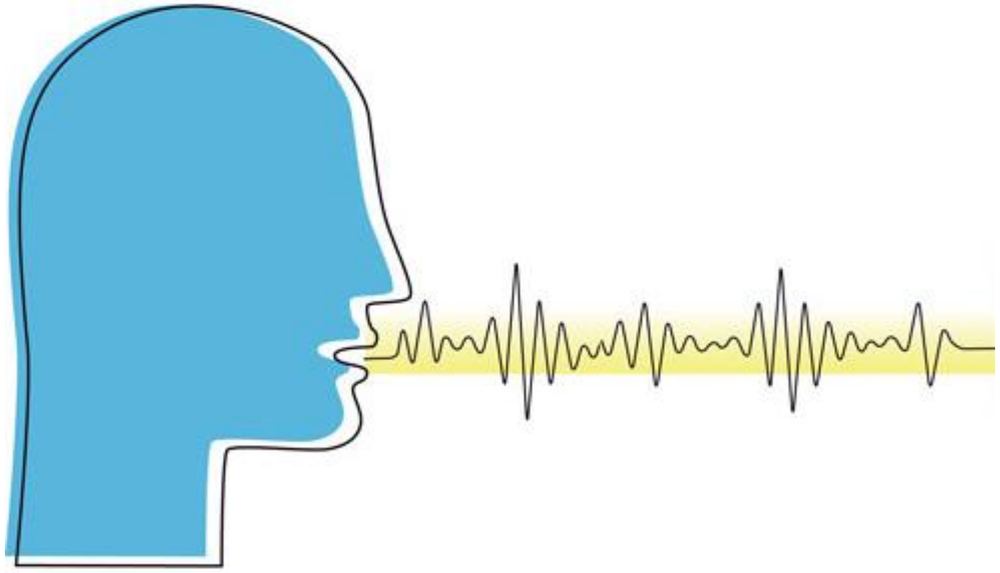
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1.

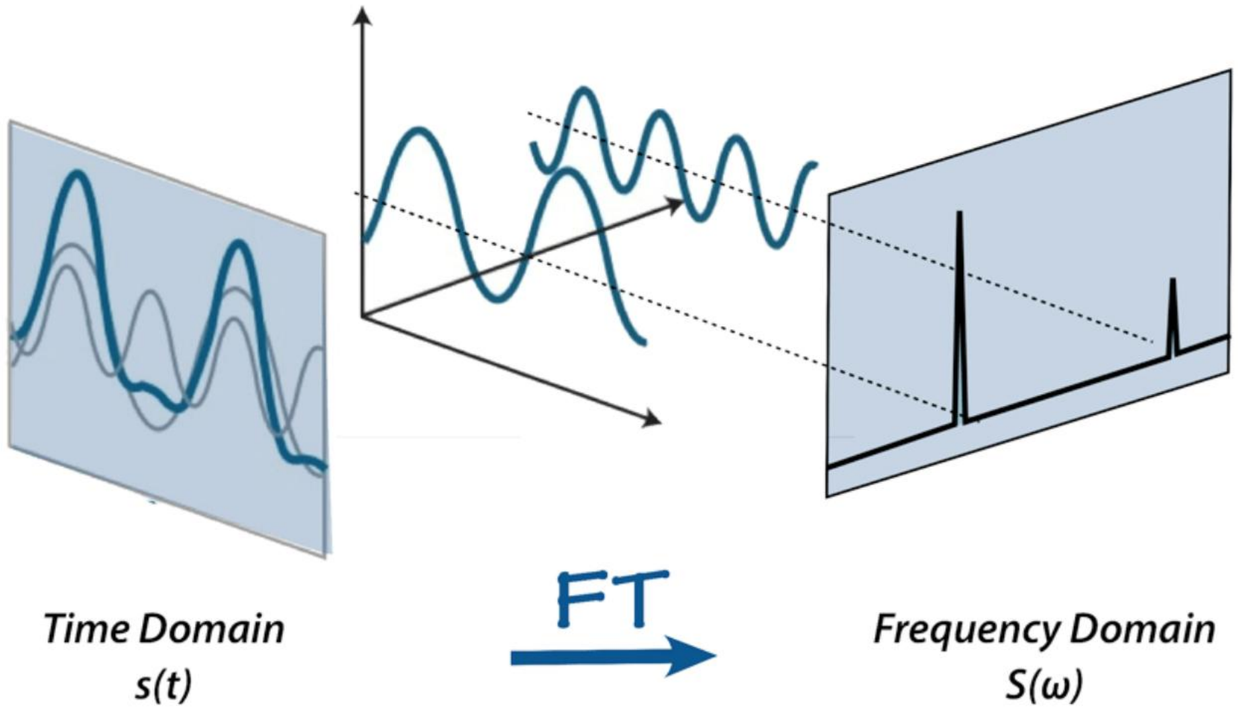
Filtros *digitales*

TRANSFORMATEC

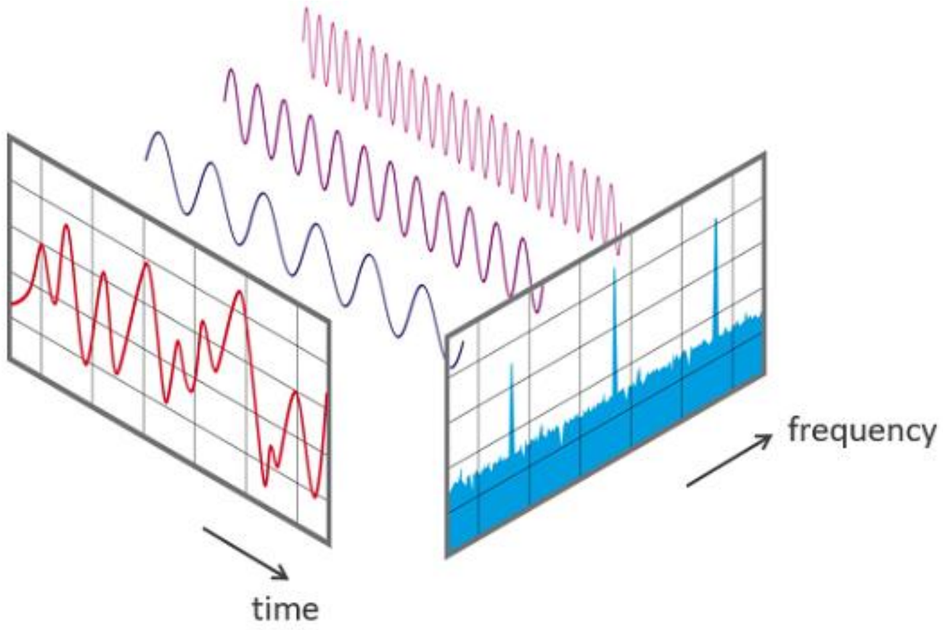
Señal



Dominio de *Frecuencia*



Dominio de *Frecuencia*



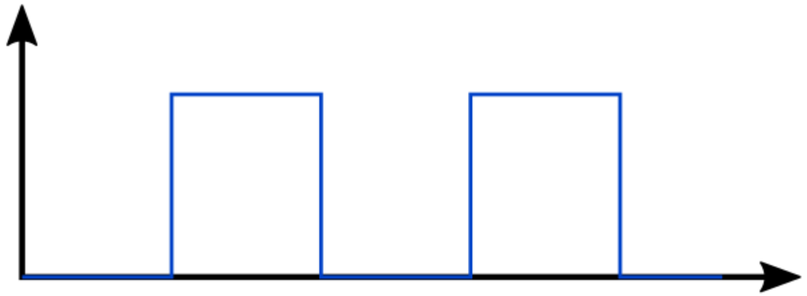
Serie de Fourier

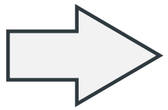
$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\frac{n\pi t}{L} + b_n \sin\frac{n\pi t}{L} \right)$$

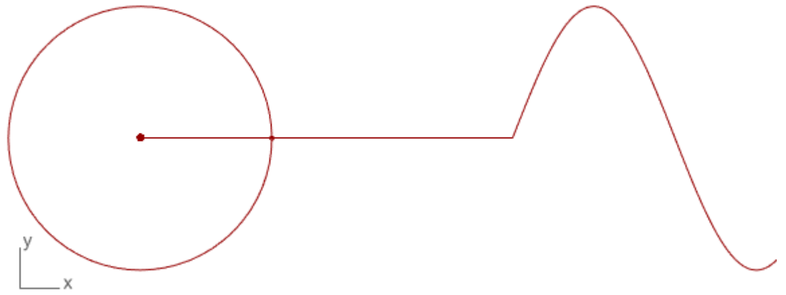
Transformada de Fourier

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i \omega t} \partial t$$

Dominio de *Frecuencia*

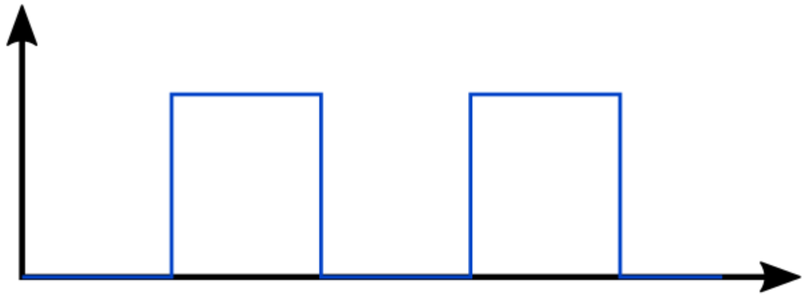


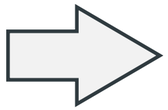


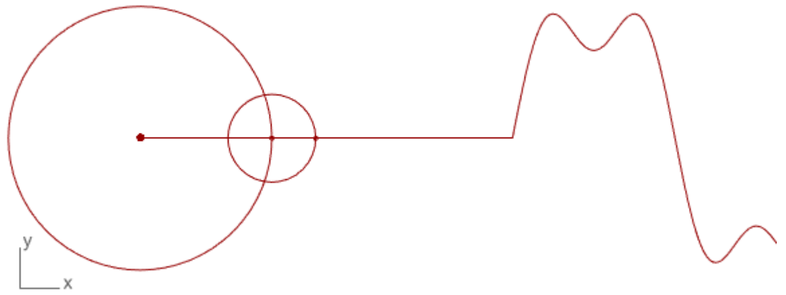


Primer componente

Dominio de *Frecuencia*

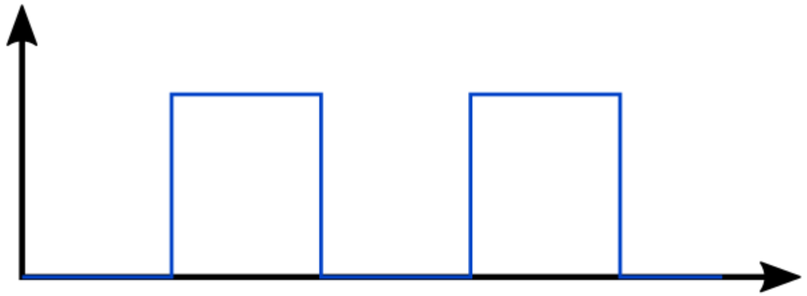


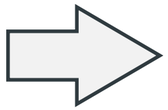


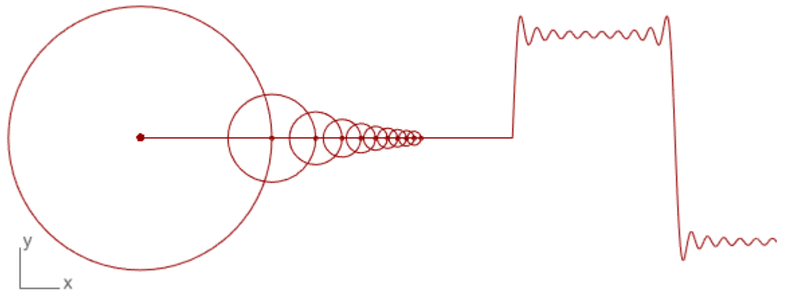


Primer y segundo componentes

Dominio de *Frecuencia*







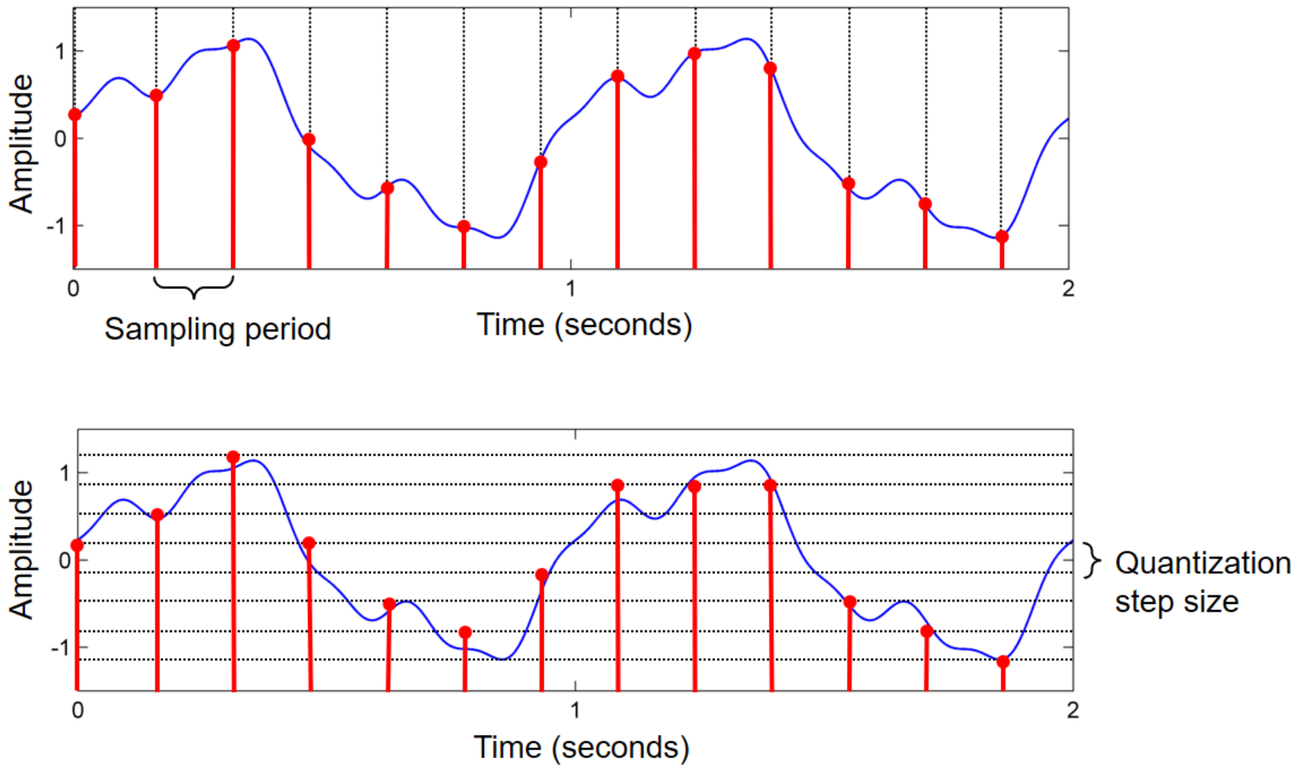
Primeros 10 componentes

Dominio de *Frecuencia*

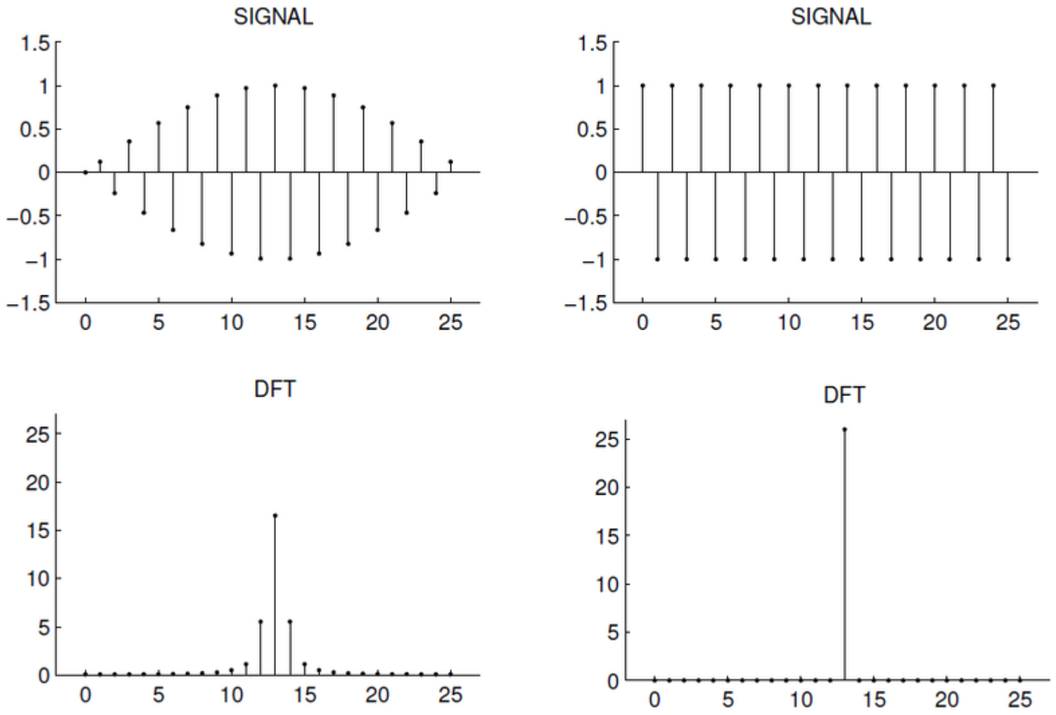
$$f(x) \qquad F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x} dx \qquad F(\omega)$$

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i \omega x} d\omega$$

Discretización



Discrete Fou*rier transform*

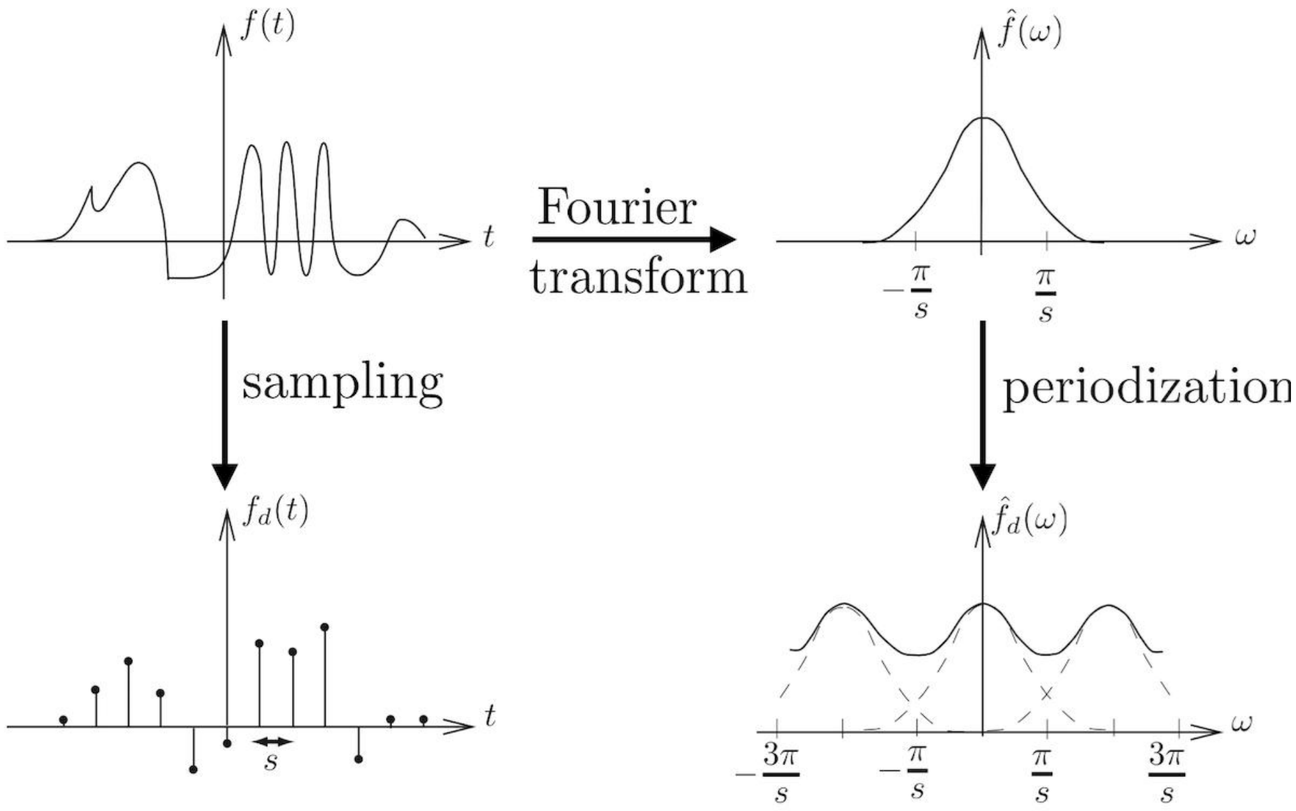


Transformada

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i\frac{2\pi}{N}kn}$$

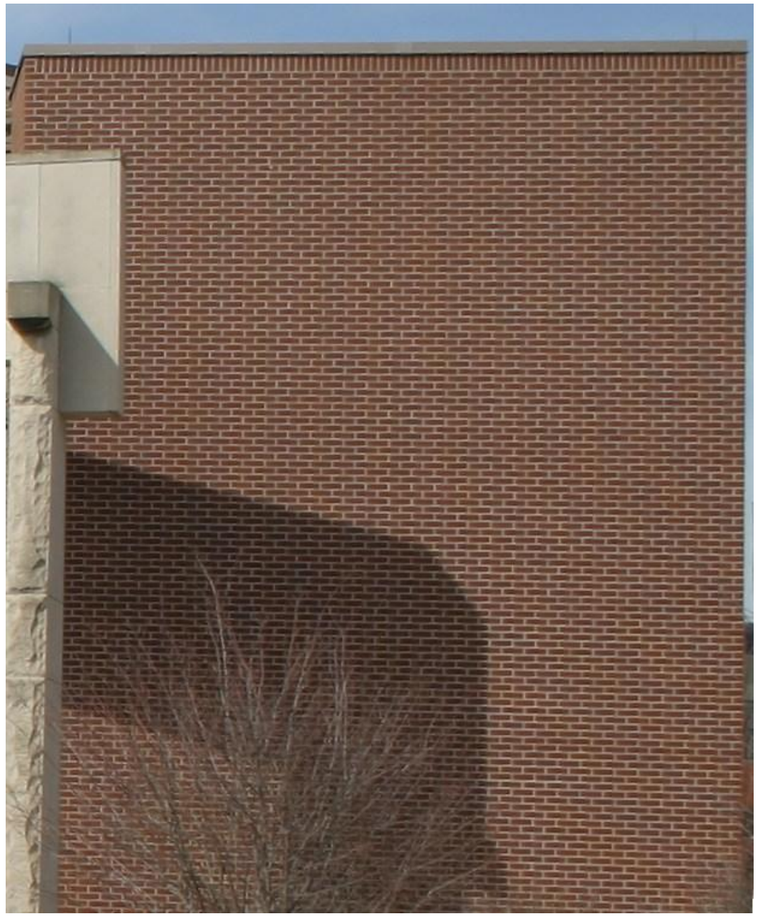
$$X_k = \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right]$$

Discrete Fou*rier transform*



Muestrear una señal equivale a periodizar su transformada de Fourier.

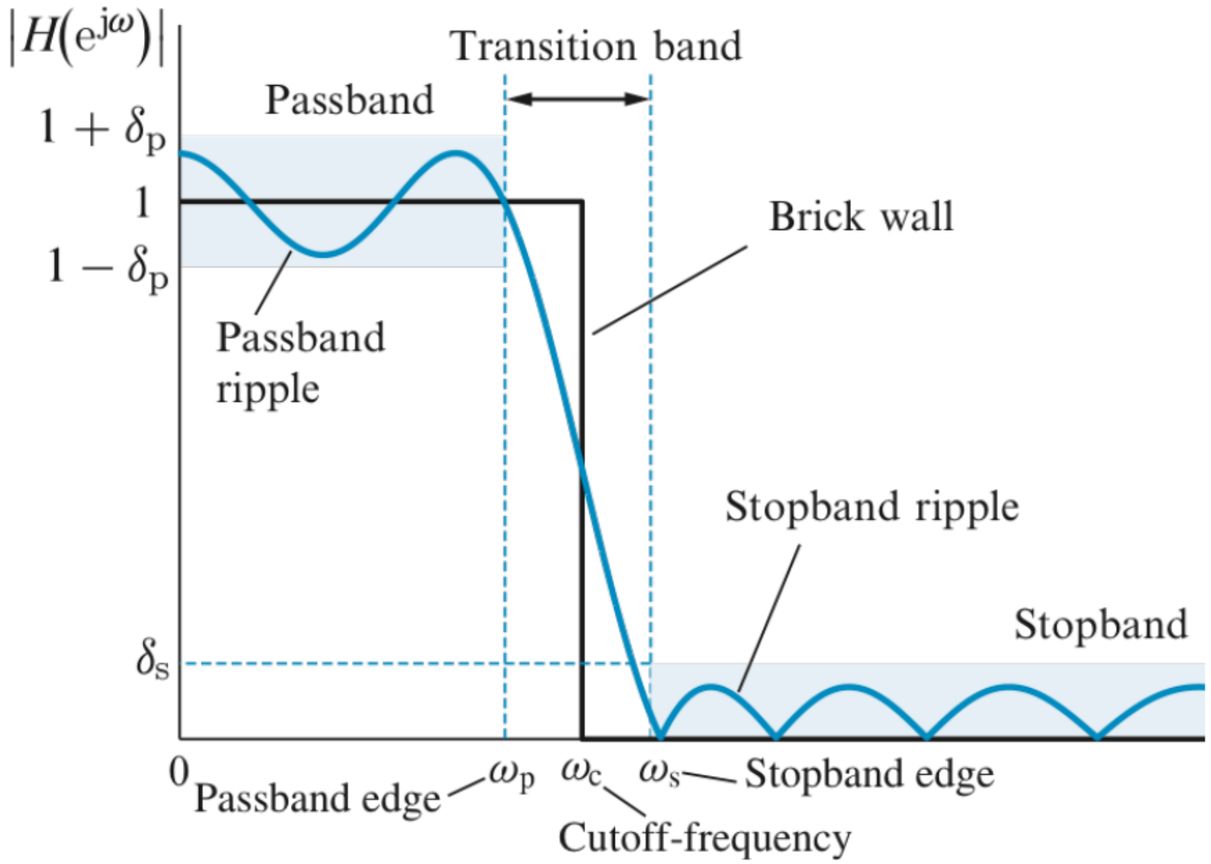
Alia*sing*



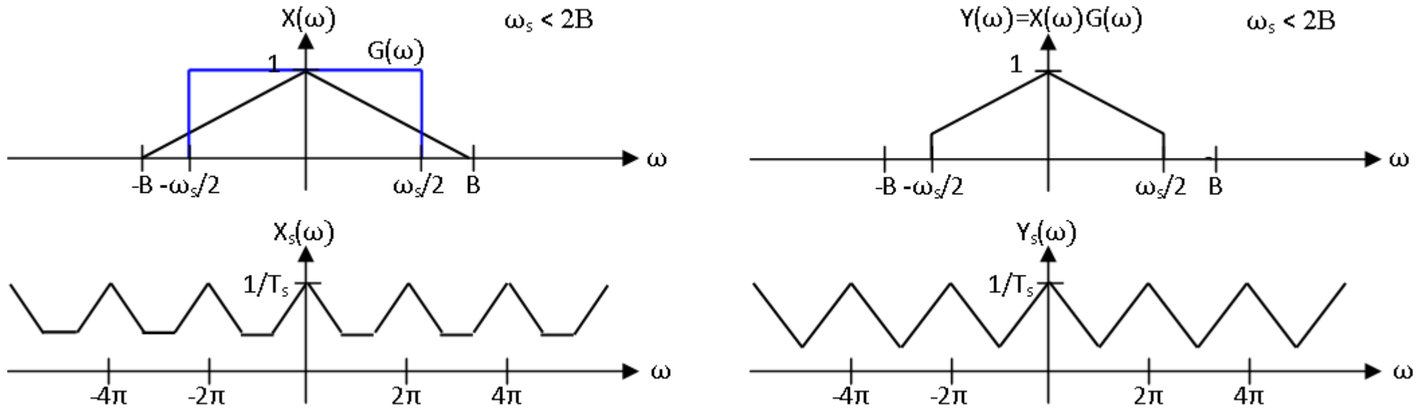
Submuestreo

Patrón de muaré

Filtro

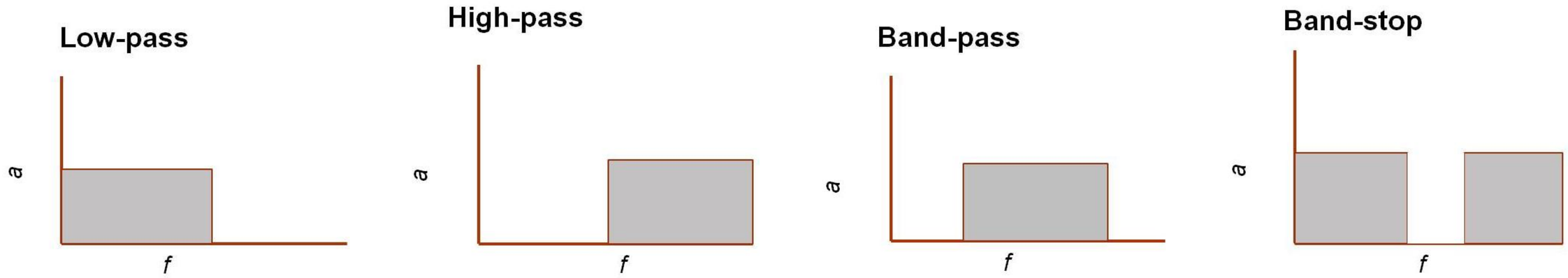

$|H(e^{j\omega})|$
Transition band
Passband
$1+\delta_p$
1
$1-\delta_p$
Brick wall
Passband ripple
Stopband ripple
Stopband
δ_s
0
Passband edge
ω_p
ω_c
ω_s
Stopband edge
Cutoff-frequency

Anti-aliasing filter

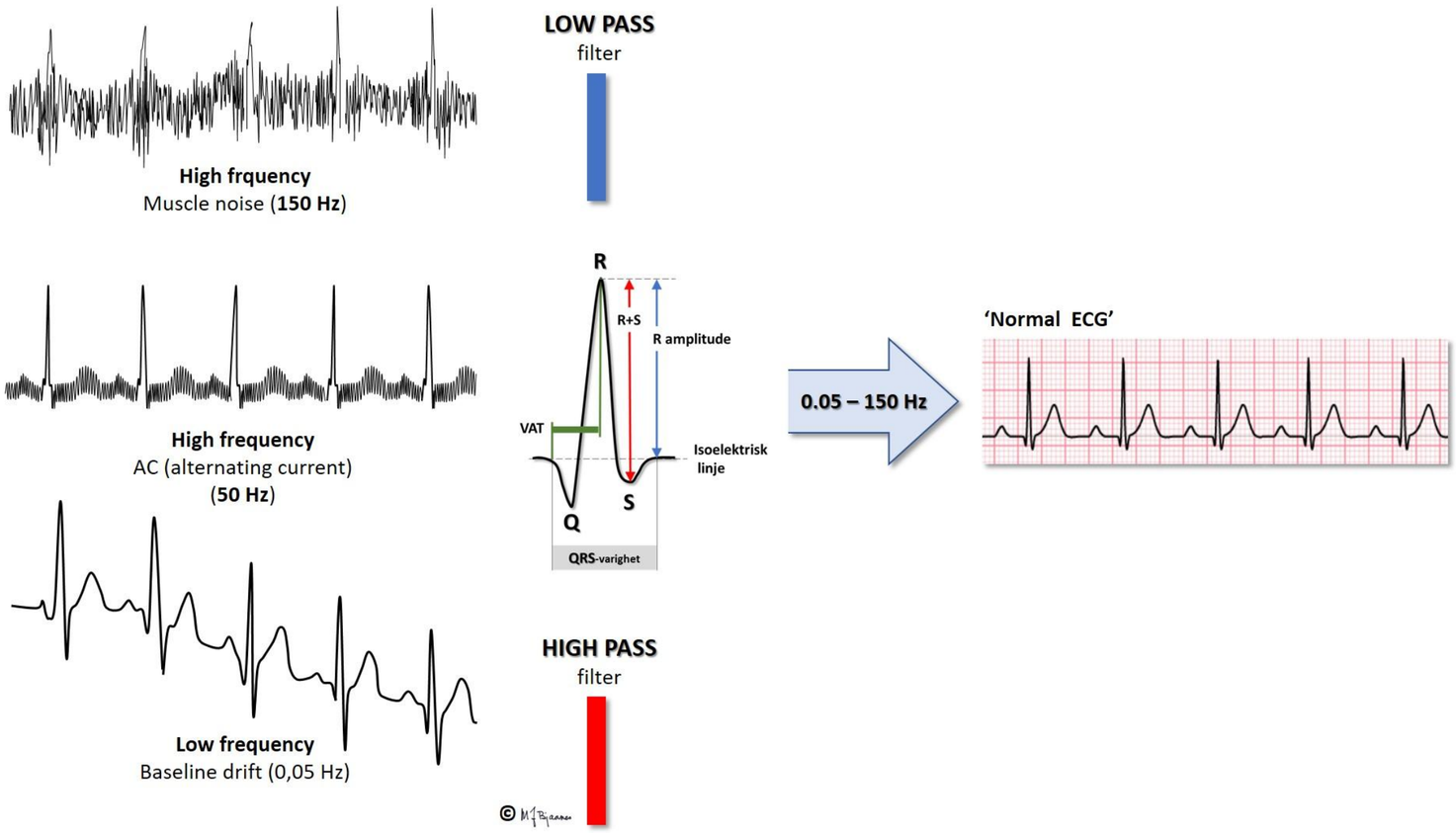


Frecuencia de Nyquist: $f_N = \frac{f_s}{2}$

Filter *types*

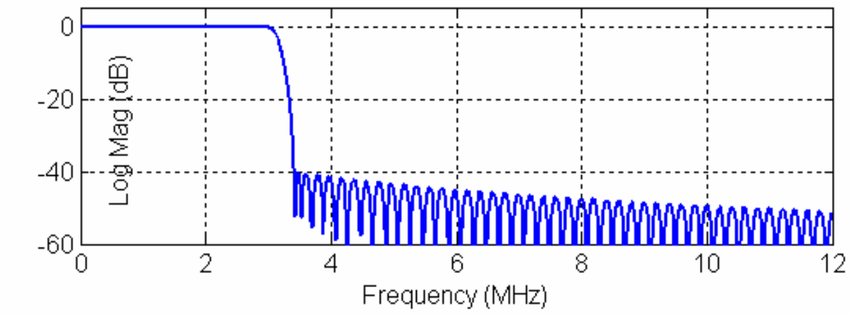


Filtro


LOW PASS
filter
High frquency
Muscle noise (150 Hz)
High frequency
AC (alternating current)
(50 Hz)
R
R+S
R amplitude
VAT
Isoelektrisk linje
Q
S
QRS-varighet
0.05 – 150 Hz
'Normal ECG'
HIGH PASS
filter
Low frequency
Baseline drift (0,05 Hz)
© M. Blanco

Filtro

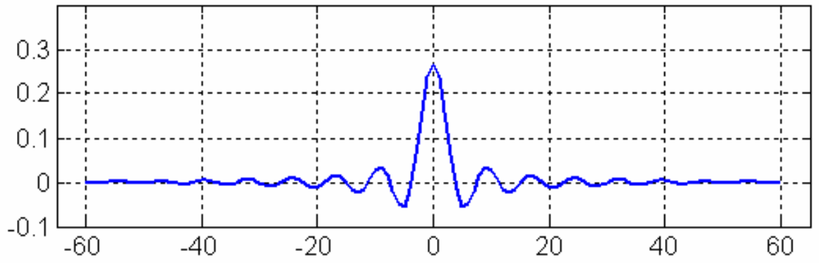
Low pass filter



$H[\omega]$

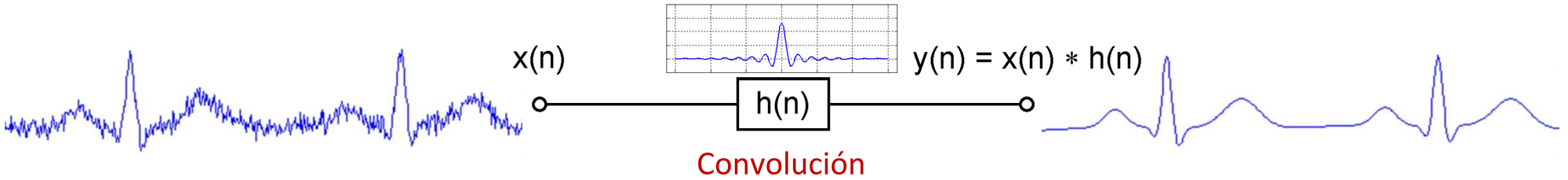
IFFT

Respuesta al impulso



$h[n]$

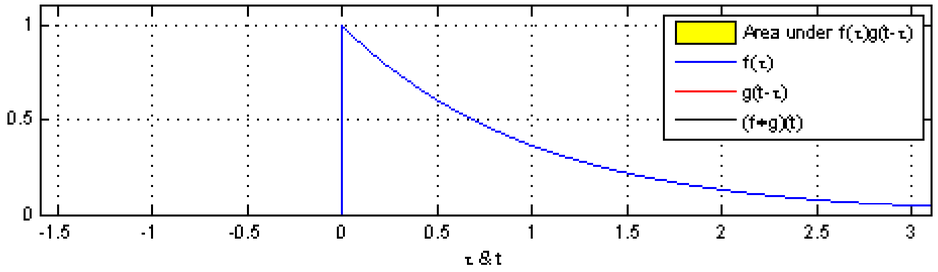
Filtro



Convolución

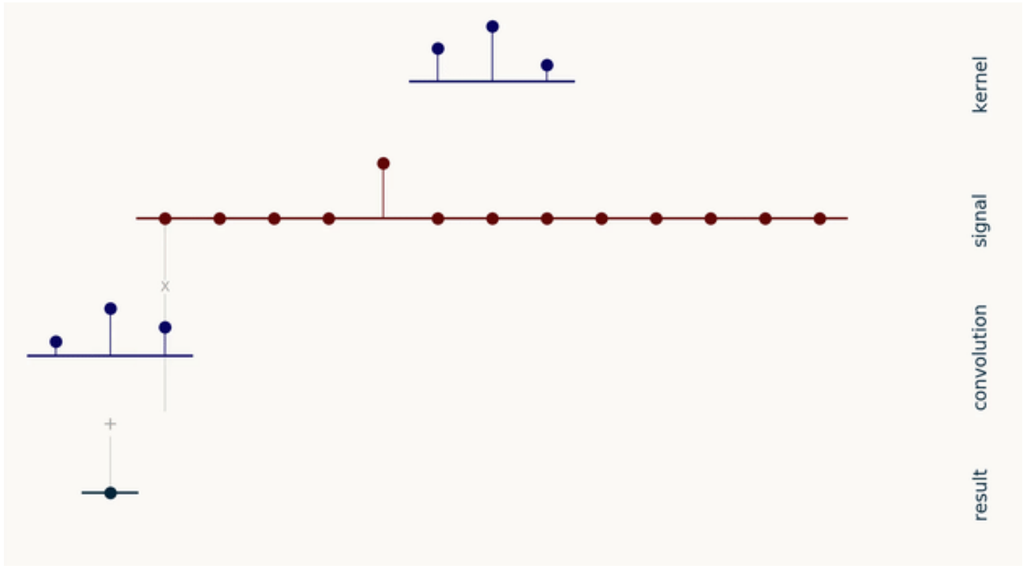
Convo*lution*

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\ d$$



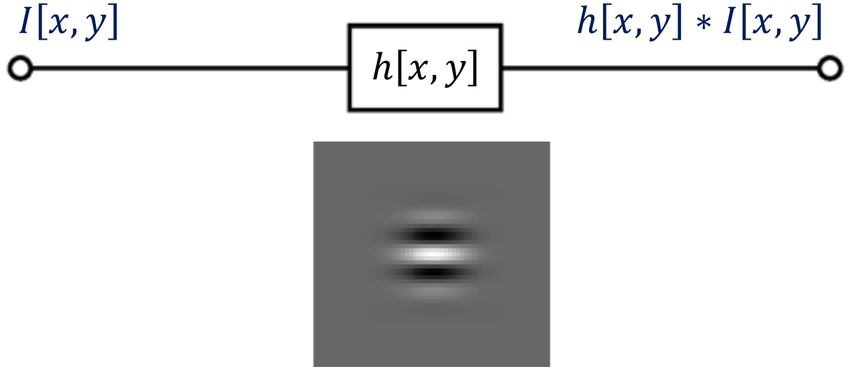
Discrete *convolution*

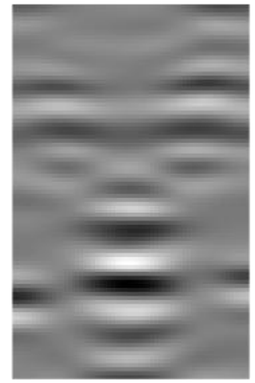
$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[k] \cdot g[n-k]$$



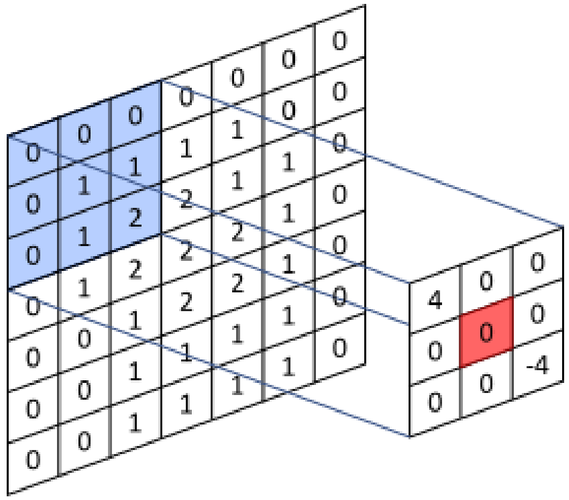
2D Discrete *convolution*

$$h[x,y] * I[x,y] = \sum_{i=-a}^{a} \sum_{j=-b}^{b} h[i,j] I[x-i, y-j]$$

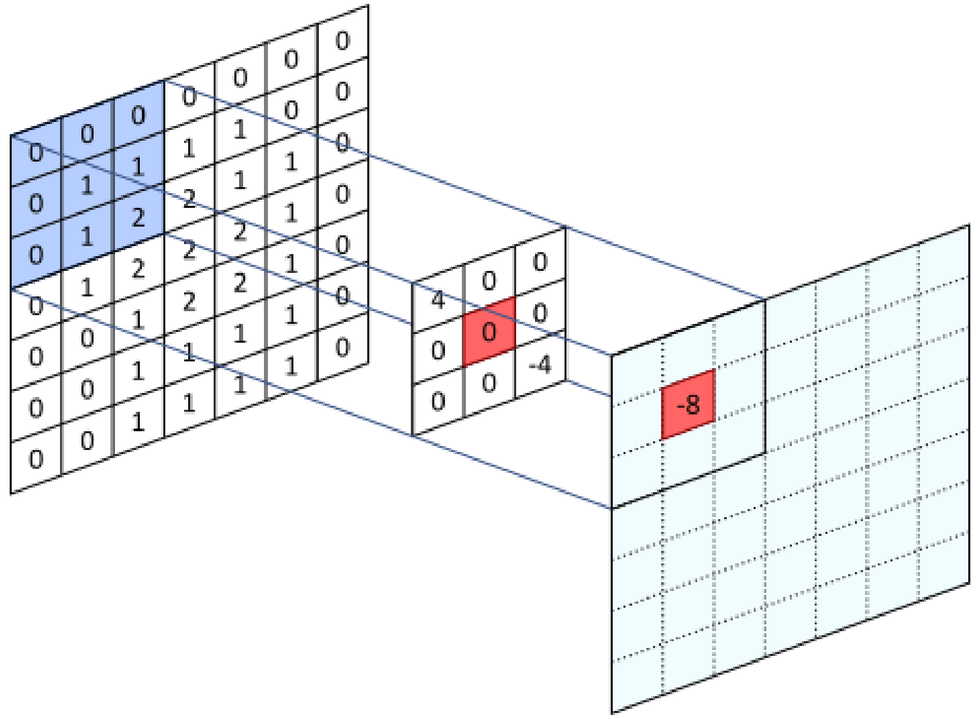




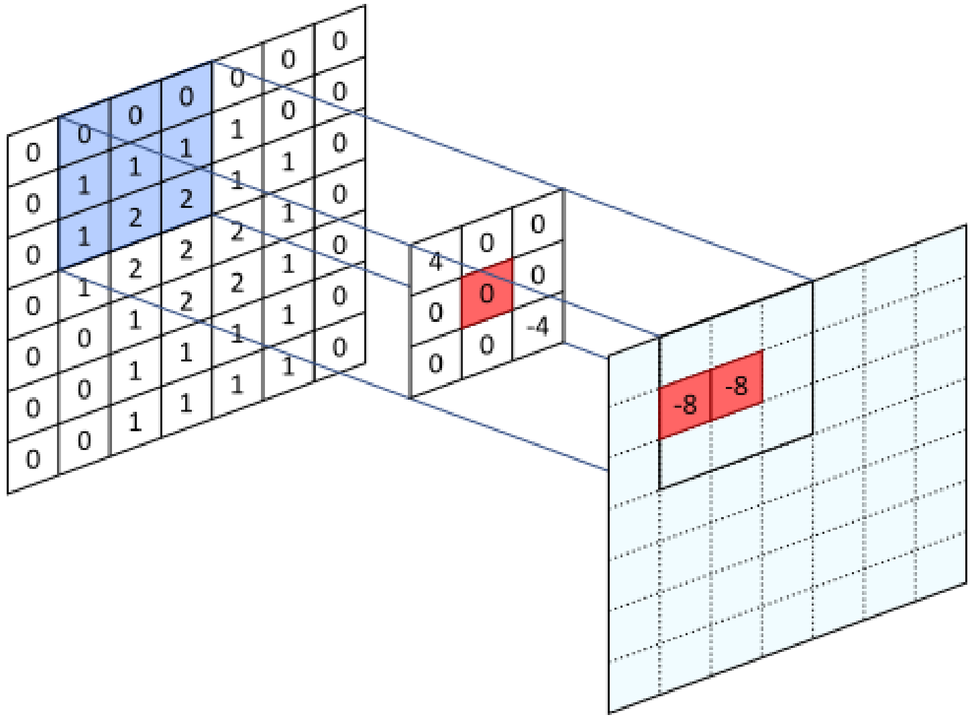
2D Discrete *convolution*



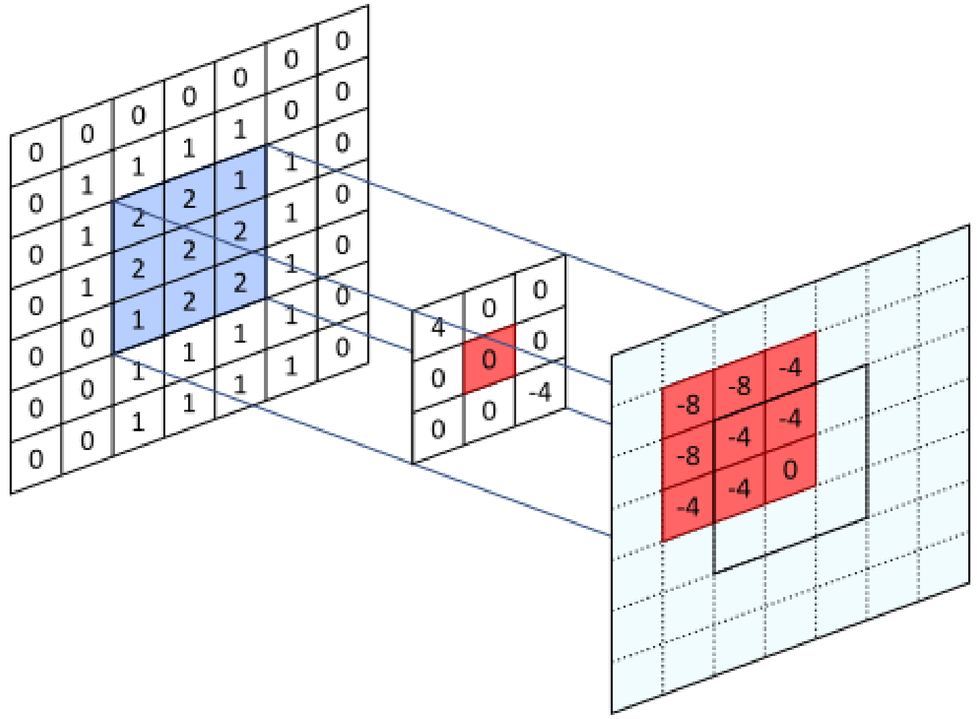
2D Discrete *convolution*



2D Discrete *convolution*



2D Discrete *convolution*



2D Discrete *convolution*

$I[x,y]$ → $h[x,y]$ → $h[x,y] * I[x,y]$

0	0	0	5	0	0	0
0	5	18	32	18	5	0
0	18	64	100	64	18	0
5	32	100	100	100	32	5
0	18	64	100	64	18	0
0	5	18	32	18	5	0
0	0	0	5	0	0	0

-1	-1	-1
-1	8	-1
-1	-1	-1

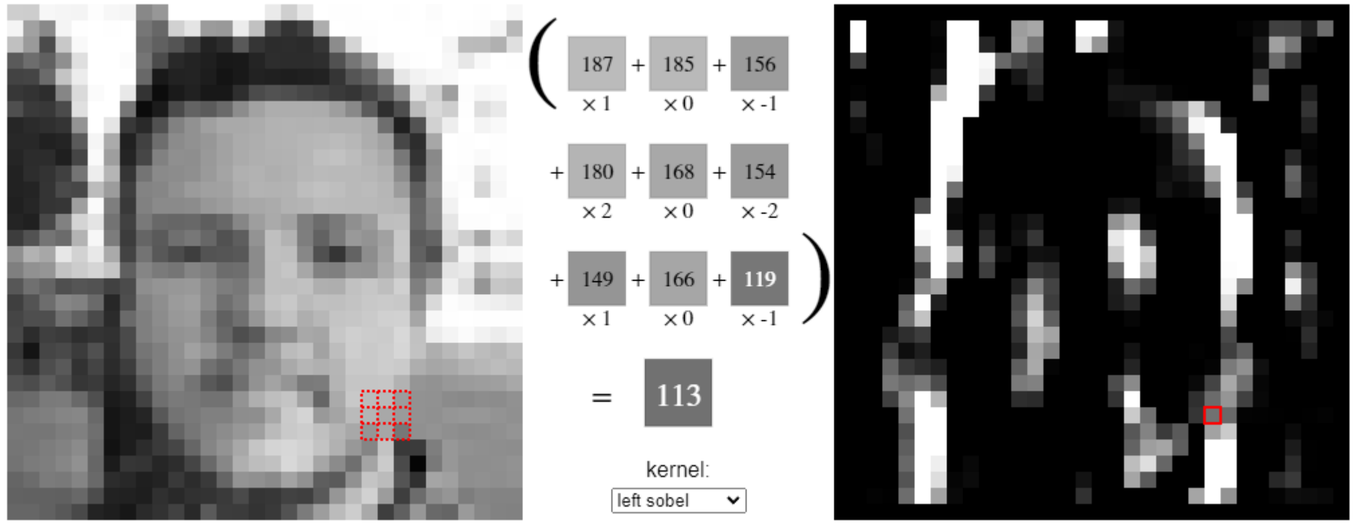
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Gaussian blur

Edge detection

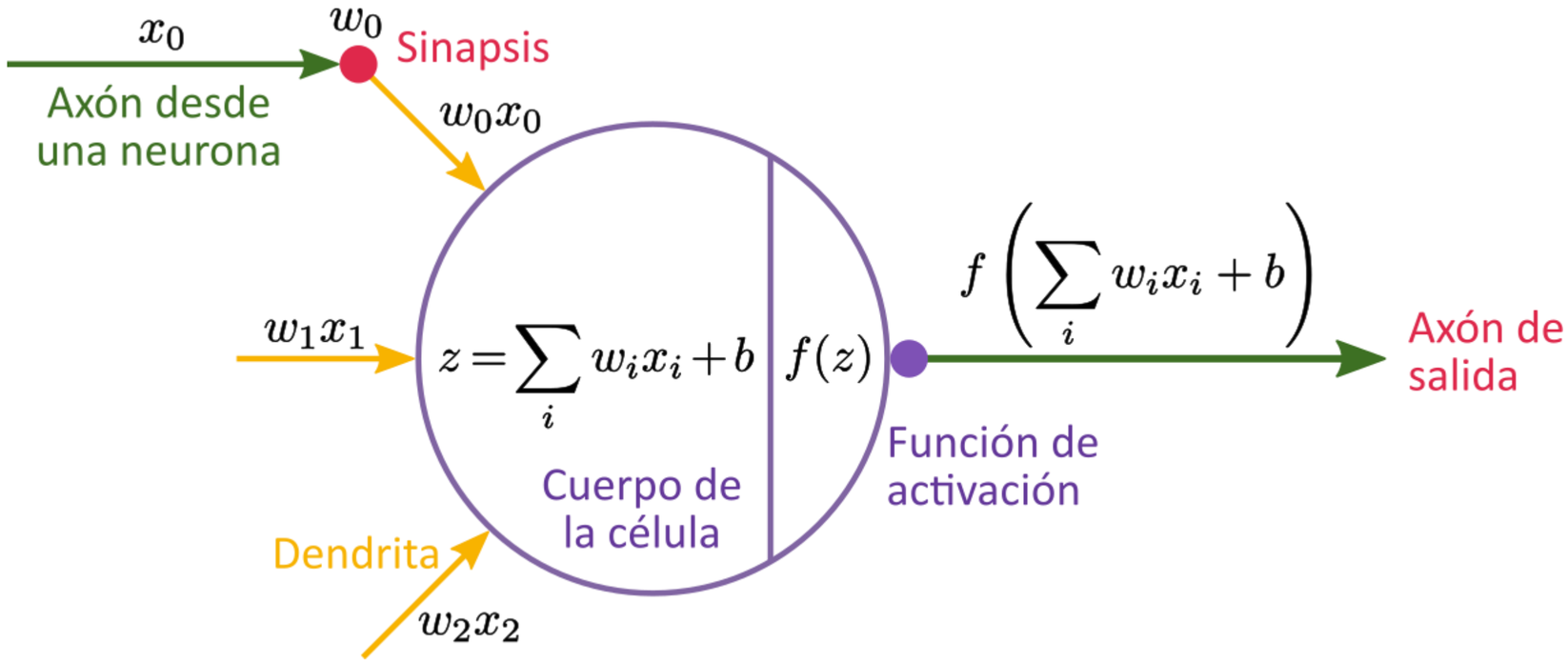
Laplacian filter

2D Discrete *convolution*



https://setosa.io/ev/image-kernels/

Red neuronal *artificial*



Frank Rosenblatt (1958) "The perceptron: a probabilistic model for information storage and organization in the brain". Psychological review 65.6 (1958): 386.

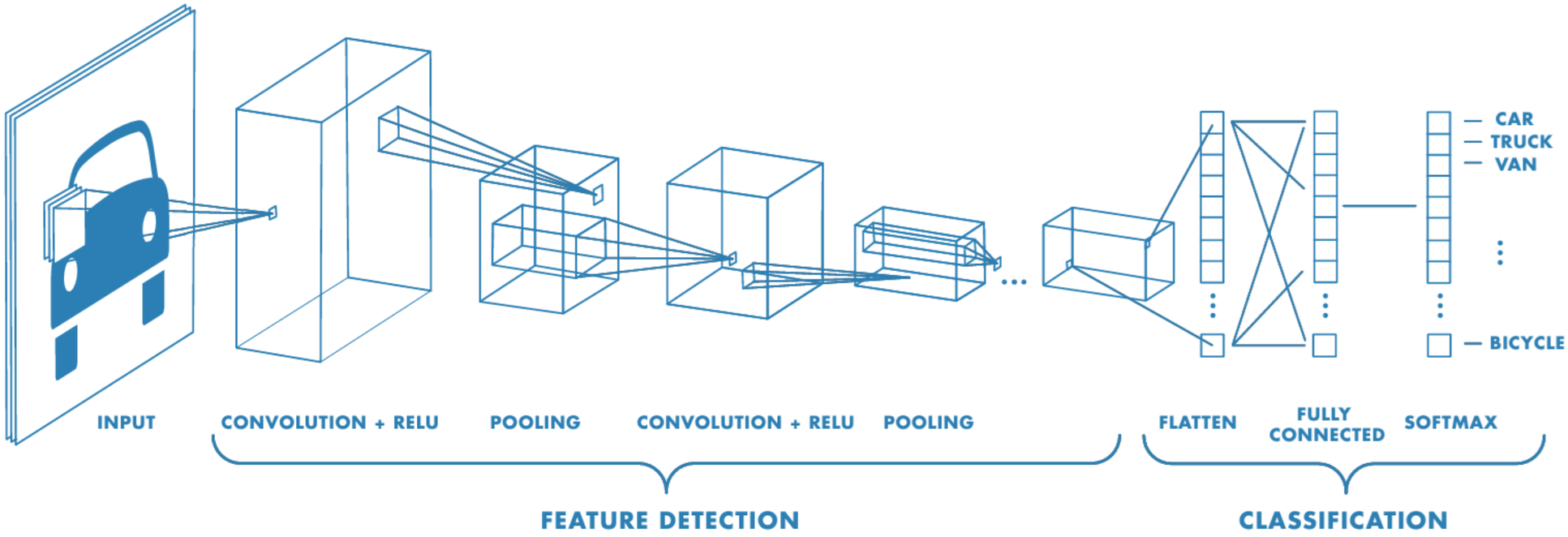
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

2.

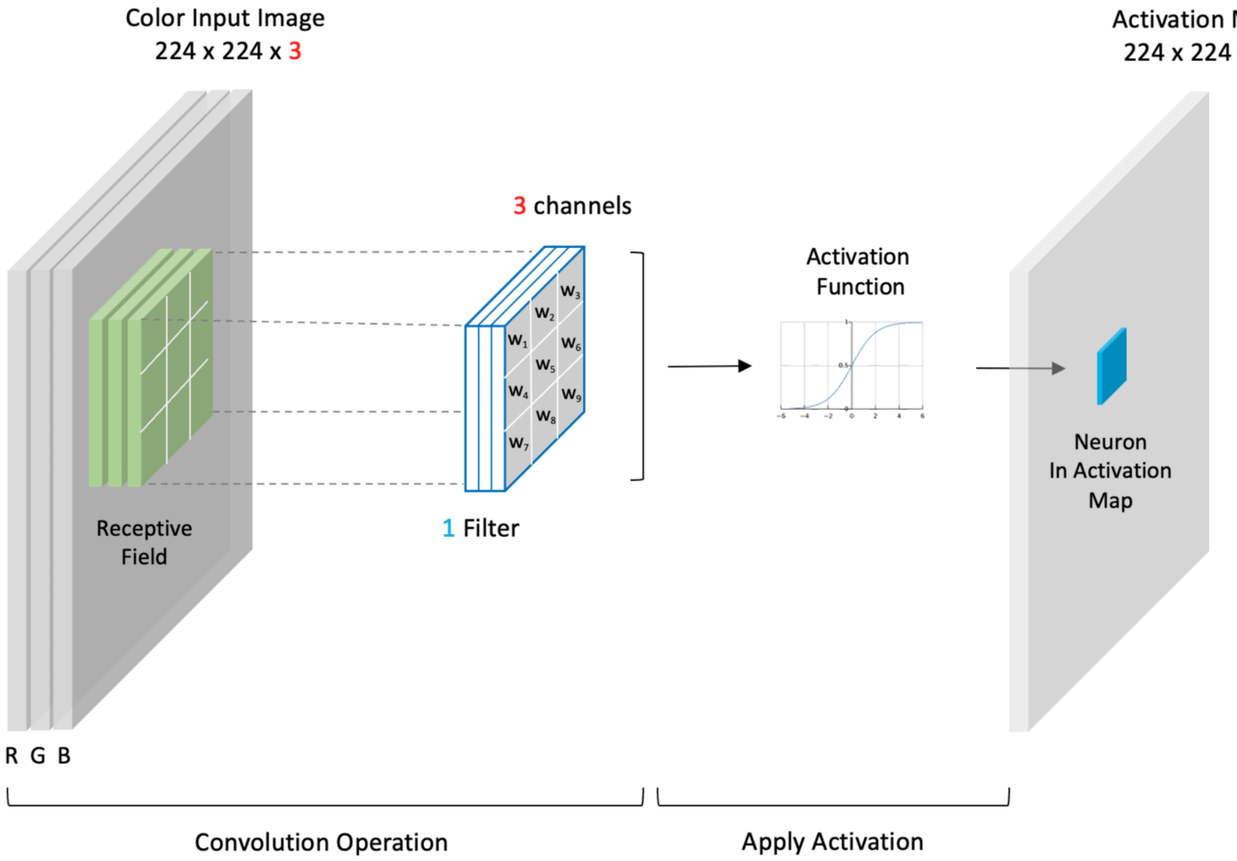
Convolutional
neural network

TRANSFORMATEC

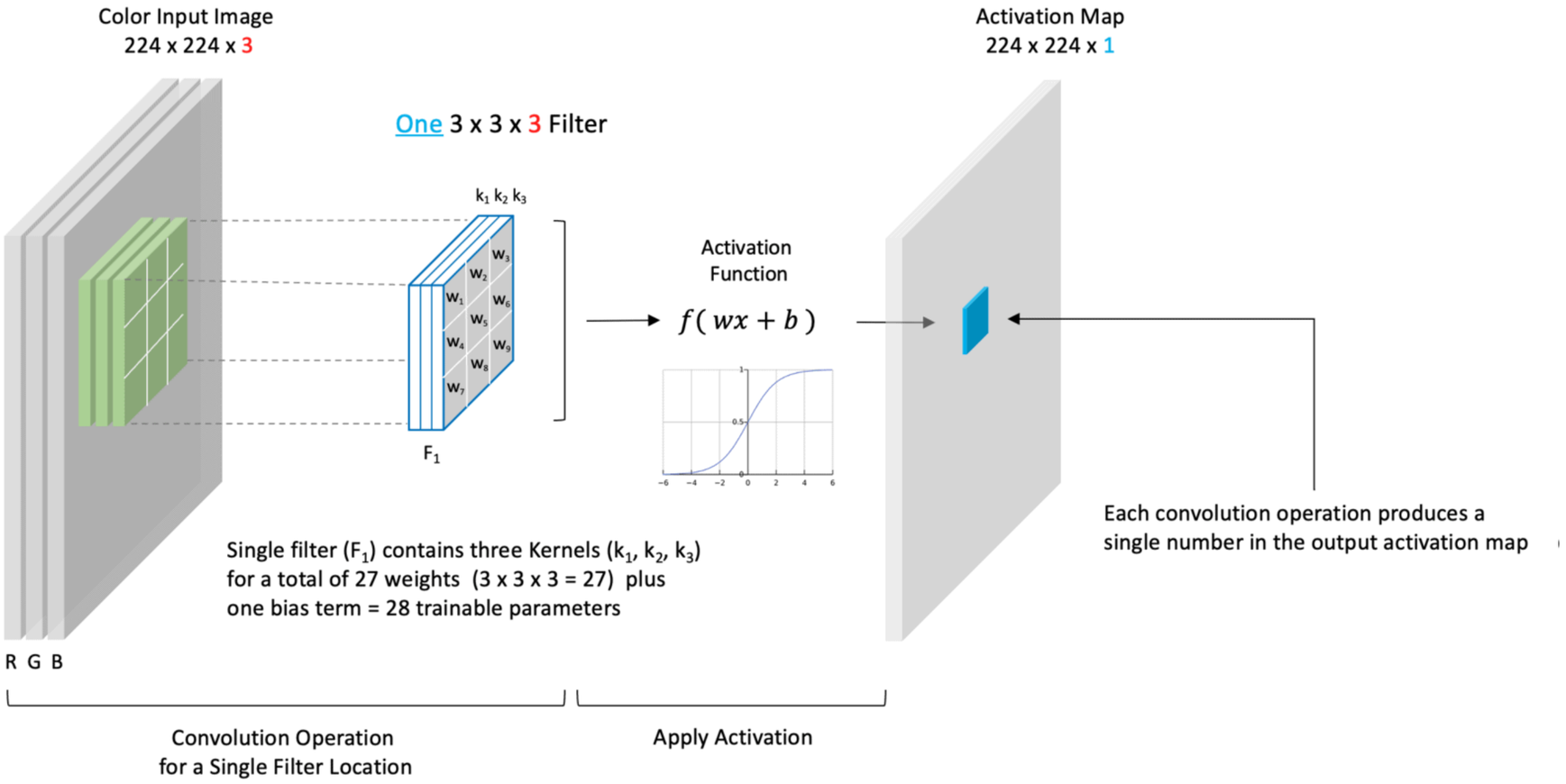
Convolutional *neural network*



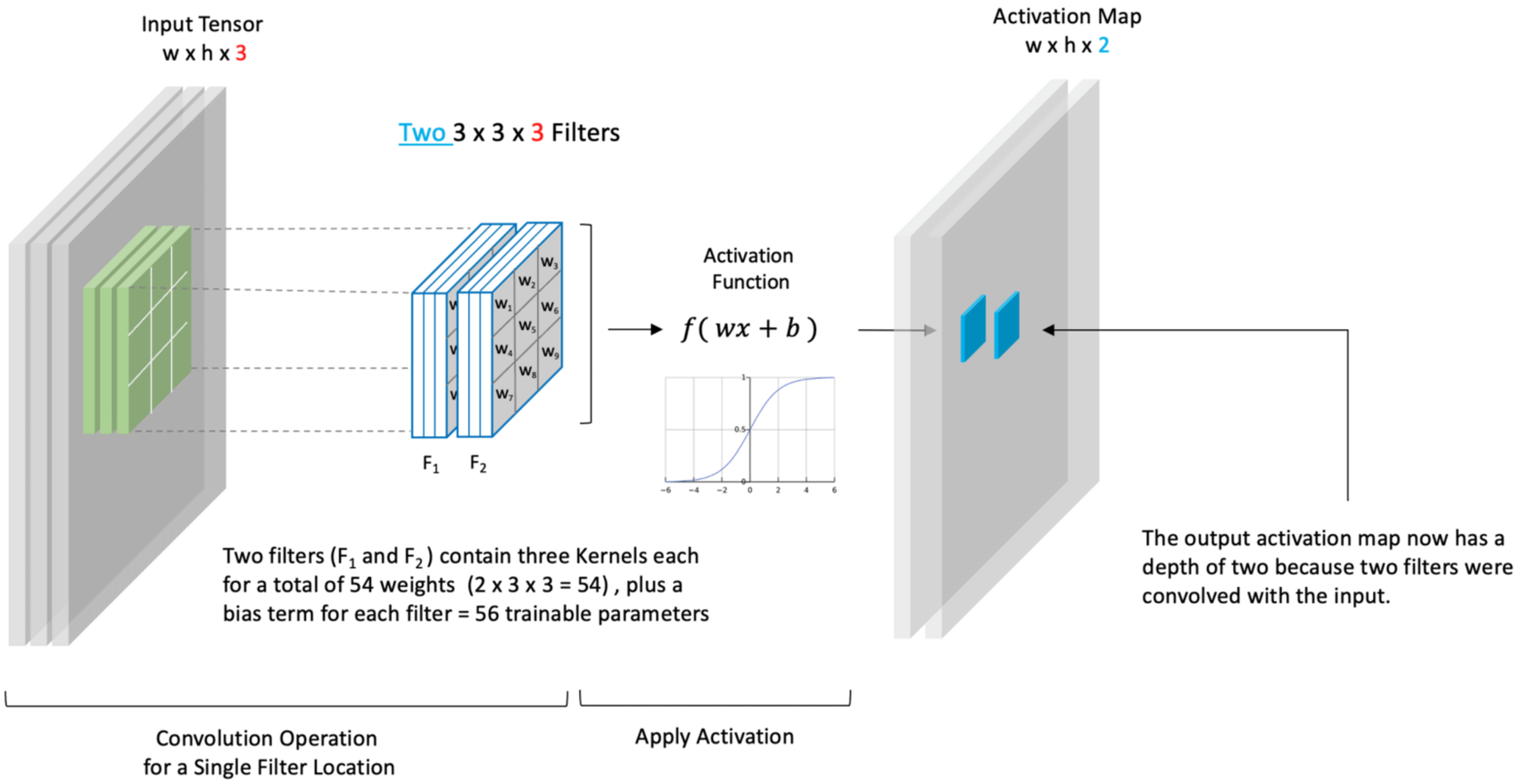
Convolutional *neural network*



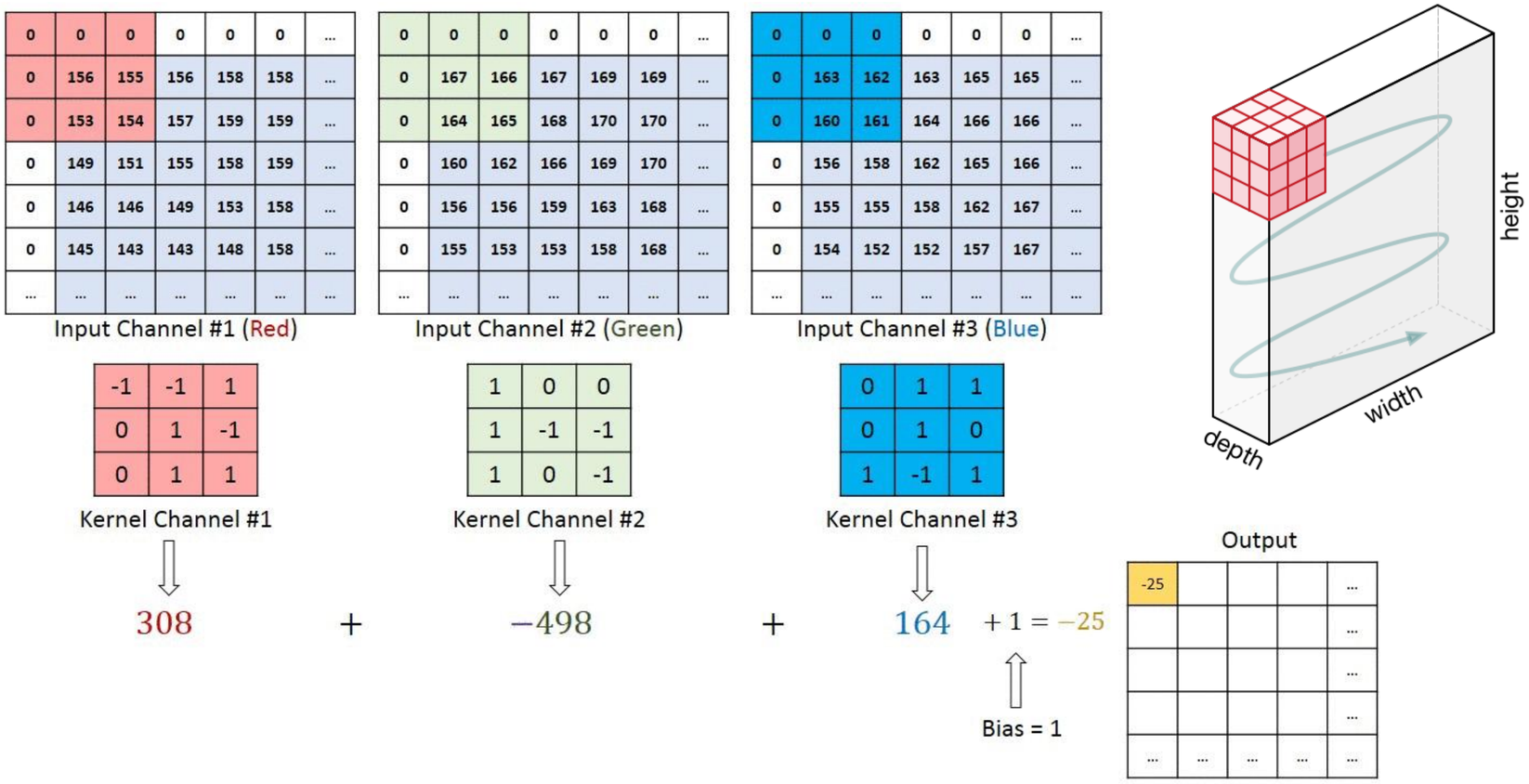
Convolutional *neural network*



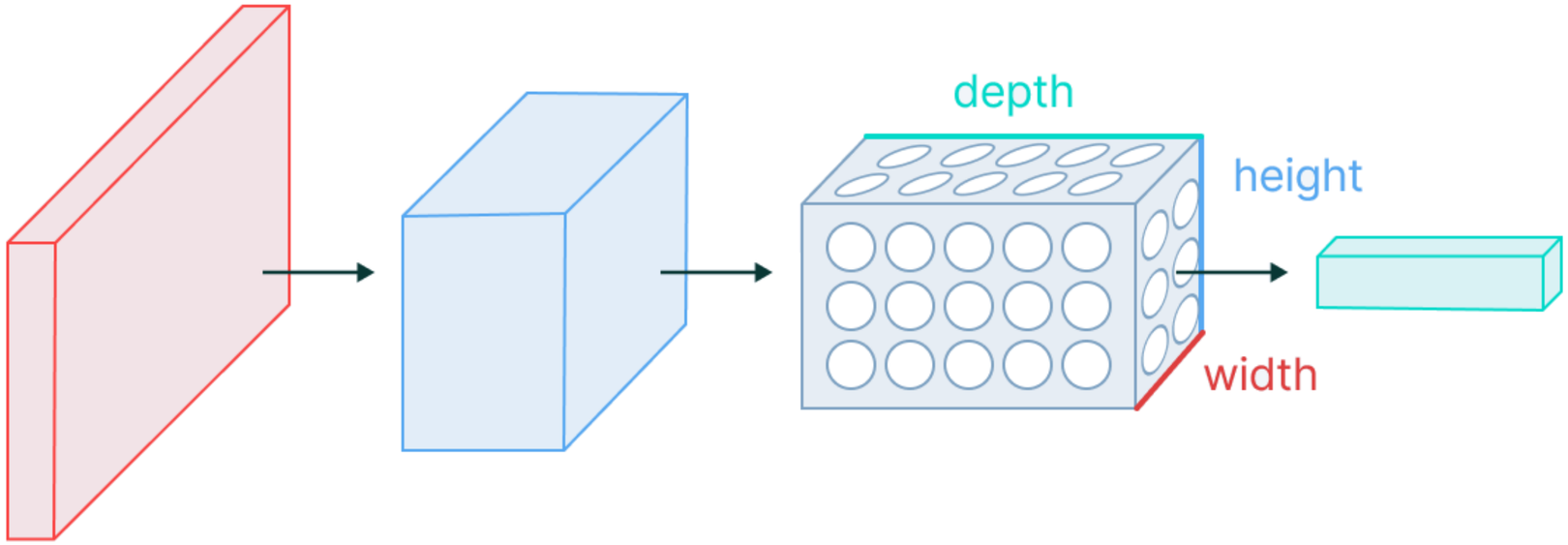
Convolutional *neural network*



Convolutional *neural network*

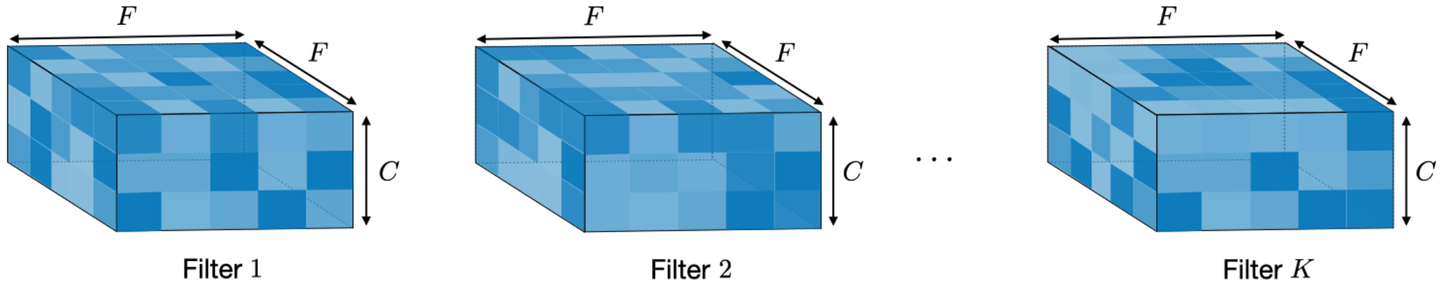


Convolutional *neural network*



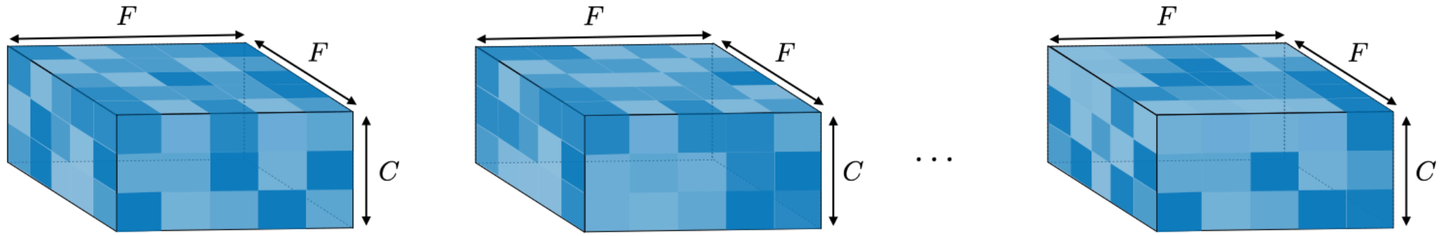
Convolutional *neural network*

Dimensiones del filtro

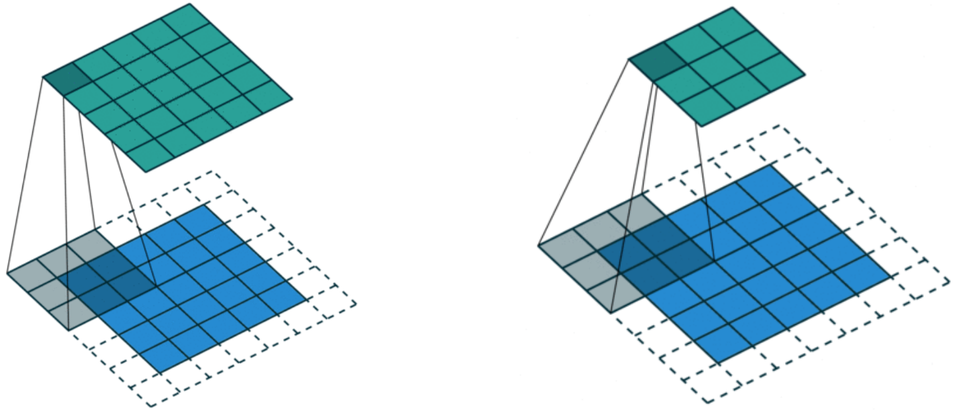


Convolutional *neural network*

Dimensiones del filtro



Stride

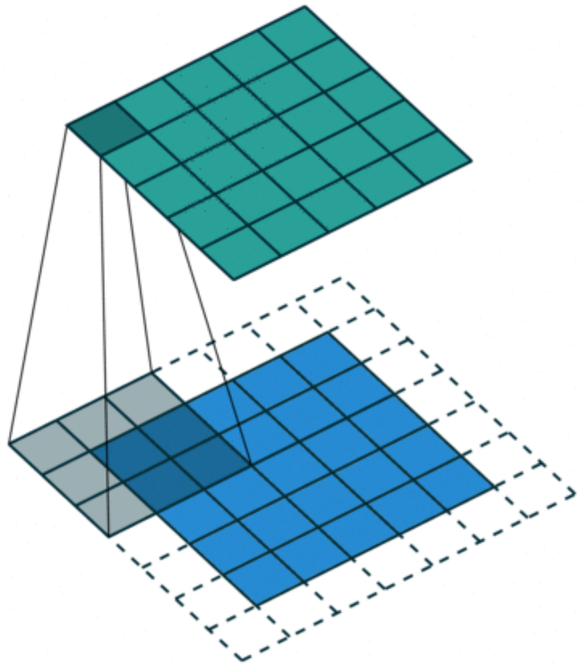


Stride 1

Stride 2

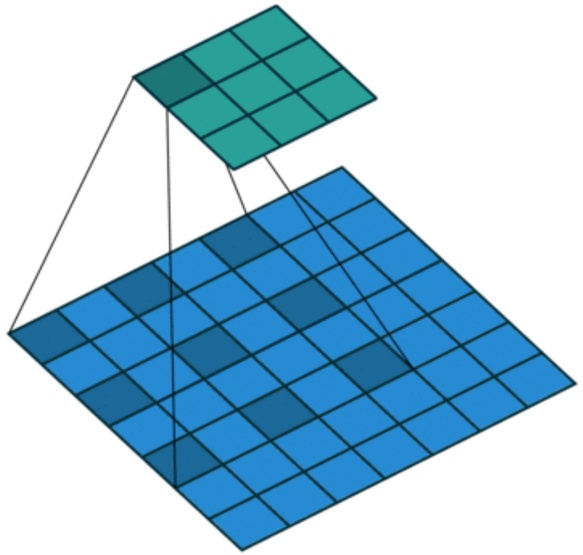
Convolutional *neural network*

Padding

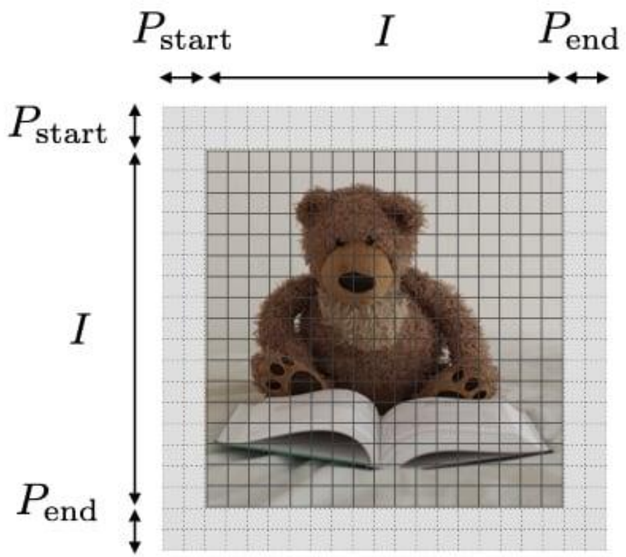


Convolutional *neural network*

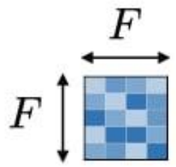
Dilated Convolution



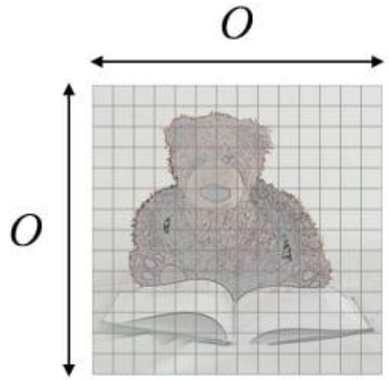
Convolutional *neural network*



Input



Filter

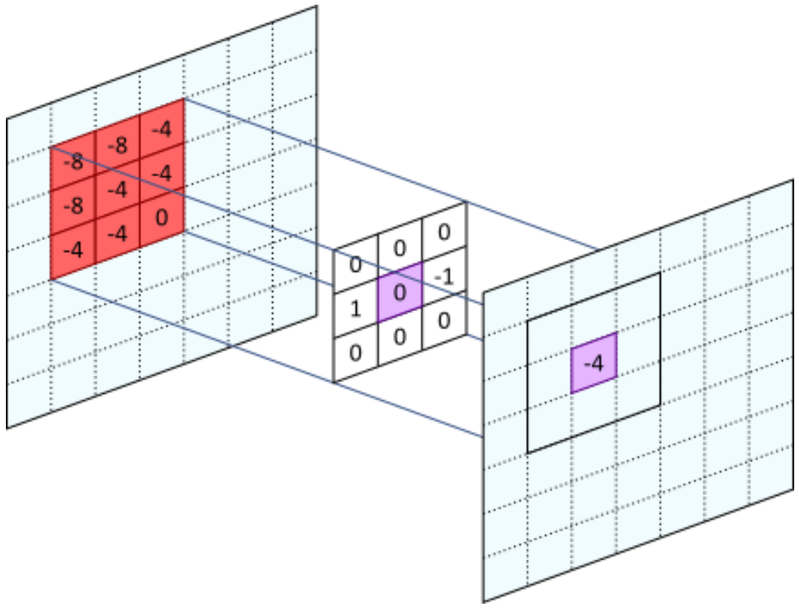


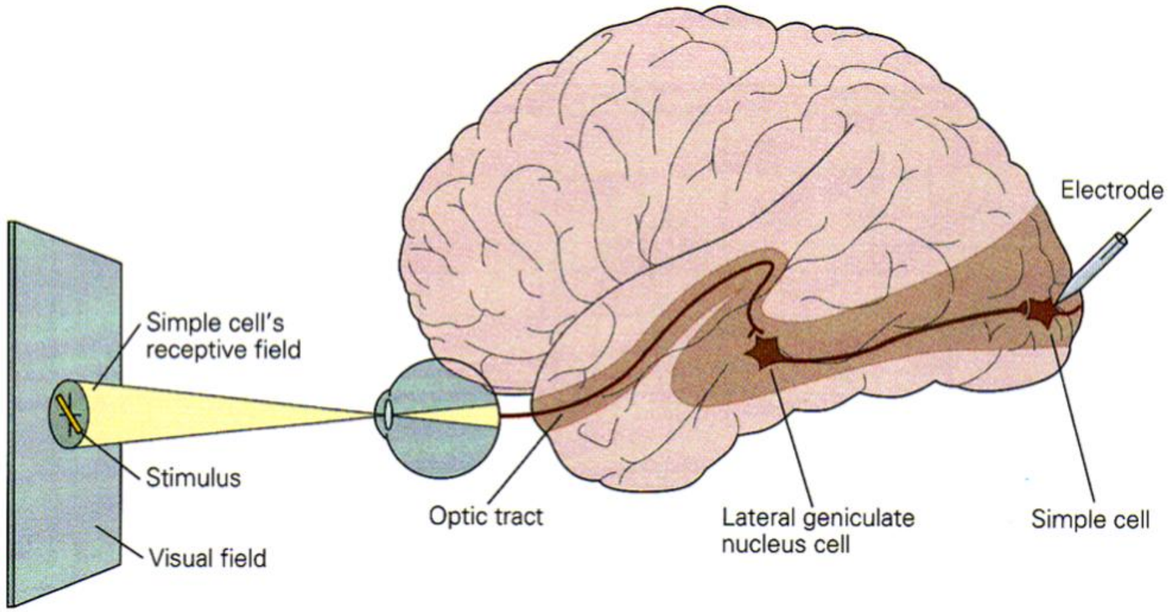
Output

$$O = \frac{I - F + P_{\text{start}} + P_{\text{end}}}{S} + 1$$

Convolutional *neural network*

Receptive field

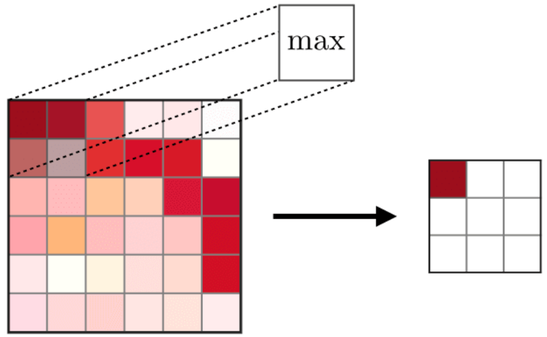




Convolutional *neural network*

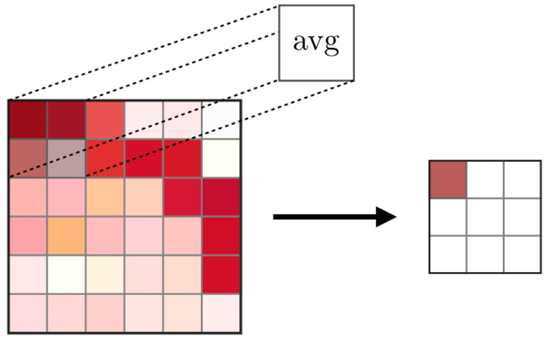
Pooling

Max Pooling



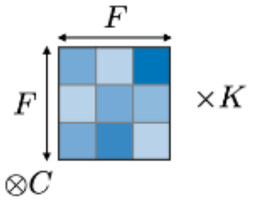
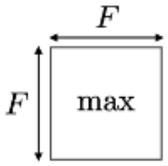
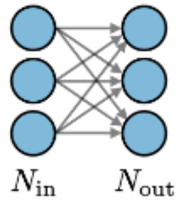
- Conserva las características detectadas.
- Más comúnmente utilizado.

Average Pooling

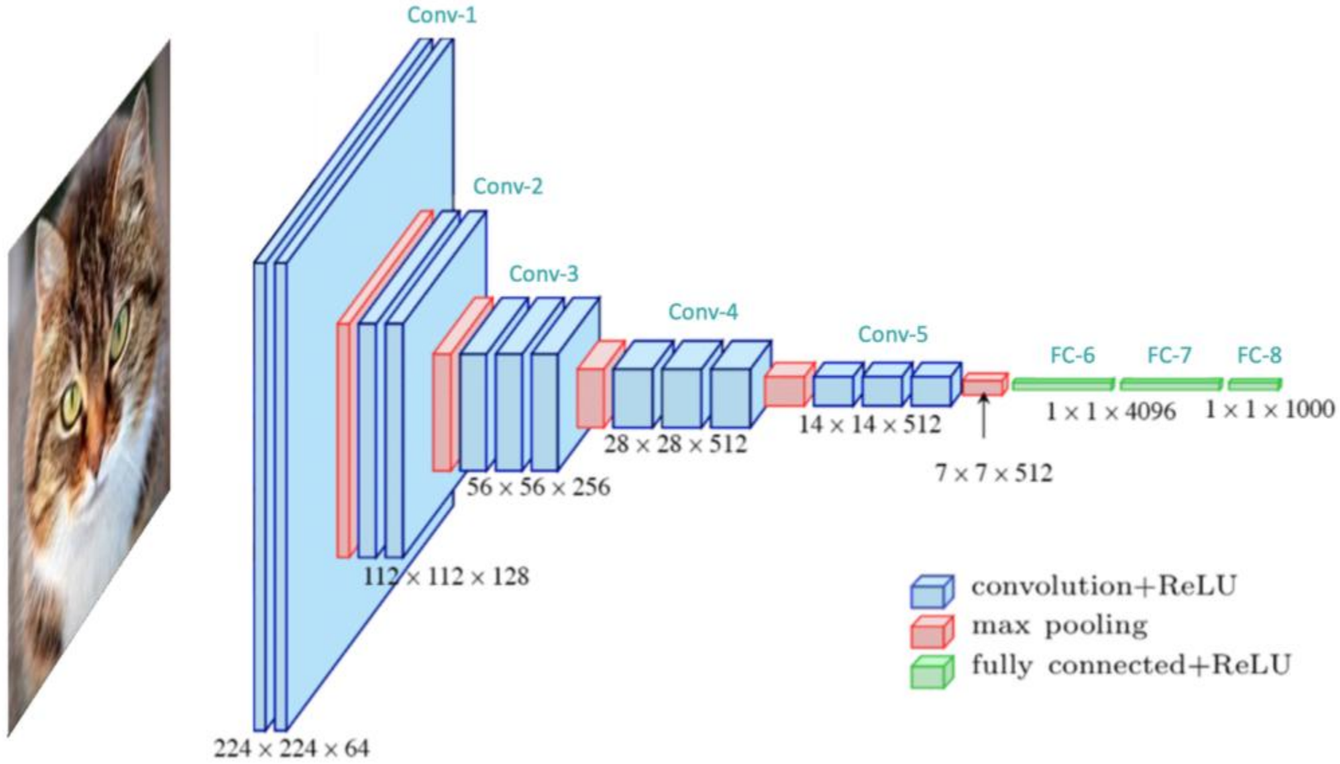


- Conserva los valores promedio de las características.
- Utilizado en LeNet.

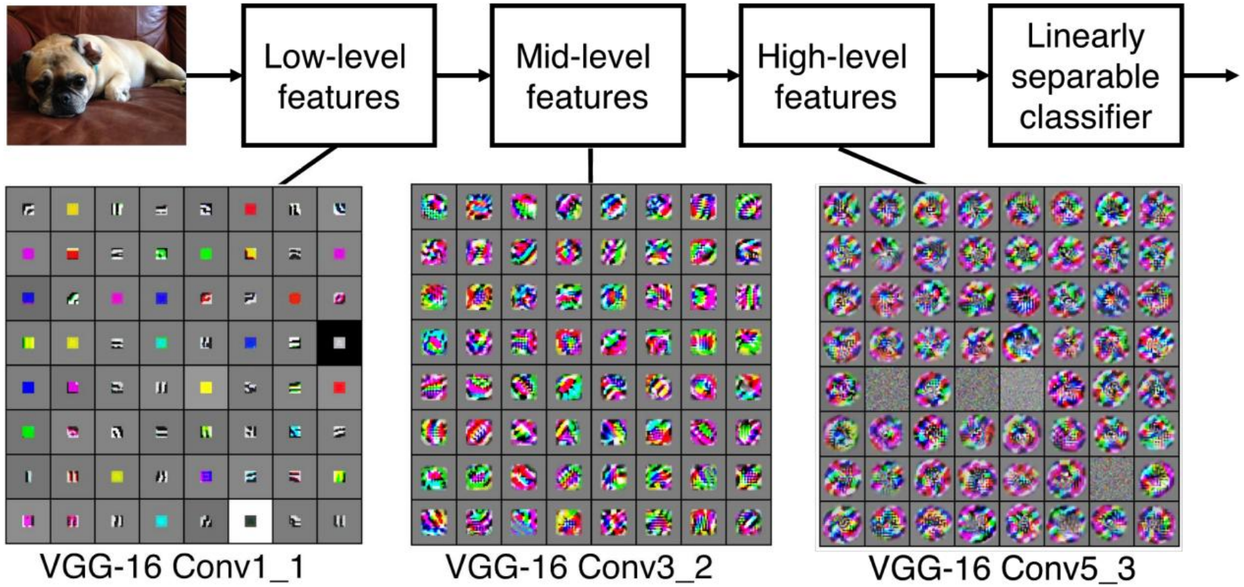
Convolutional *neural network*

	CONV	POOL	FC
Illustration			
Input size	$I \times I \times C$	$I \times I \times C$	N_{in}
Output size	$O \times O \times K$	$O \times O \times C$	N_{out}
Number of parameters	$(F \times F \times C + 1) \cdot K$	0	$(N_{\text{in}} + 1) \times N_{\text{out}}$
Remarks	• One bias parameter per filter • In most cases, $S < F$ • A common choice for K is $2C$	• Pooling operation done channel-wise • In most cases, $S = F$	• Input is flattened • One bias parameter per neuron • The number of FC neurons is free of structural constraints

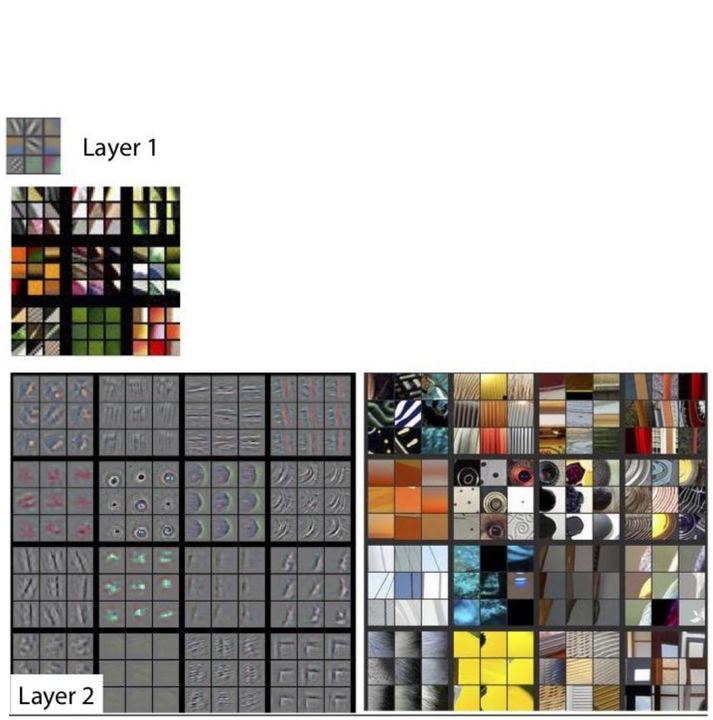
Convolutional *neural network*

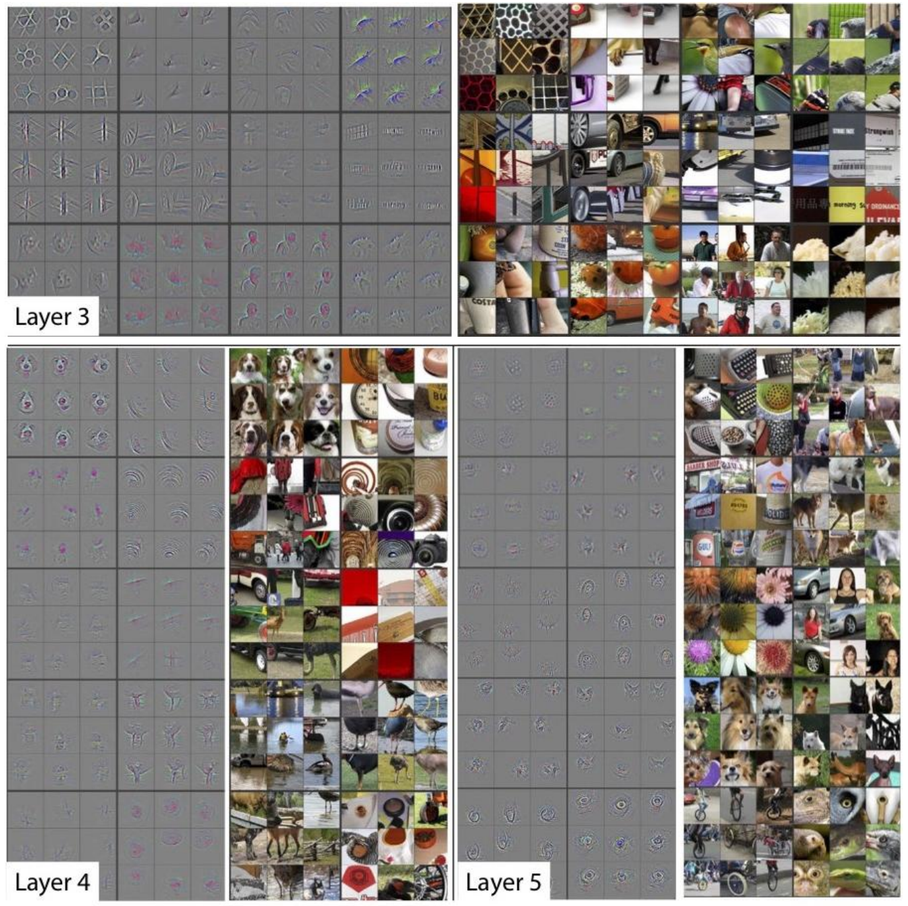


Convolutional *neural network*



Deep *representations*





Deep *representations*

Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

https://distill.pub/2017/feature-visualization/

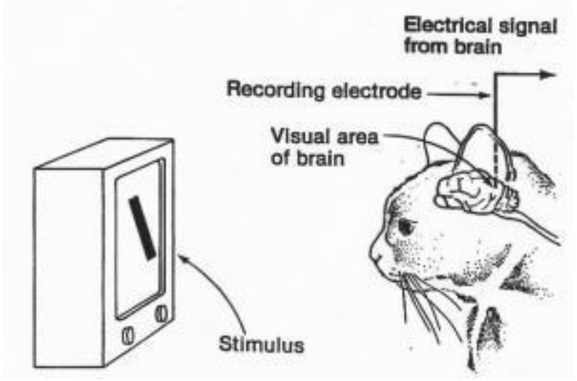
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

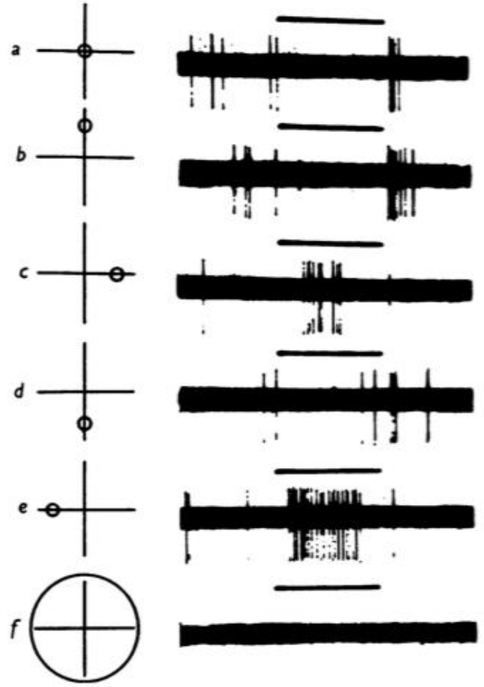
3.

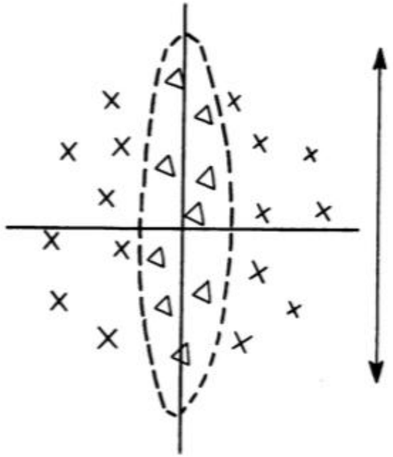
Arquitecturas *clásicas*

TRANSFORMATEC

Corteza *visual*

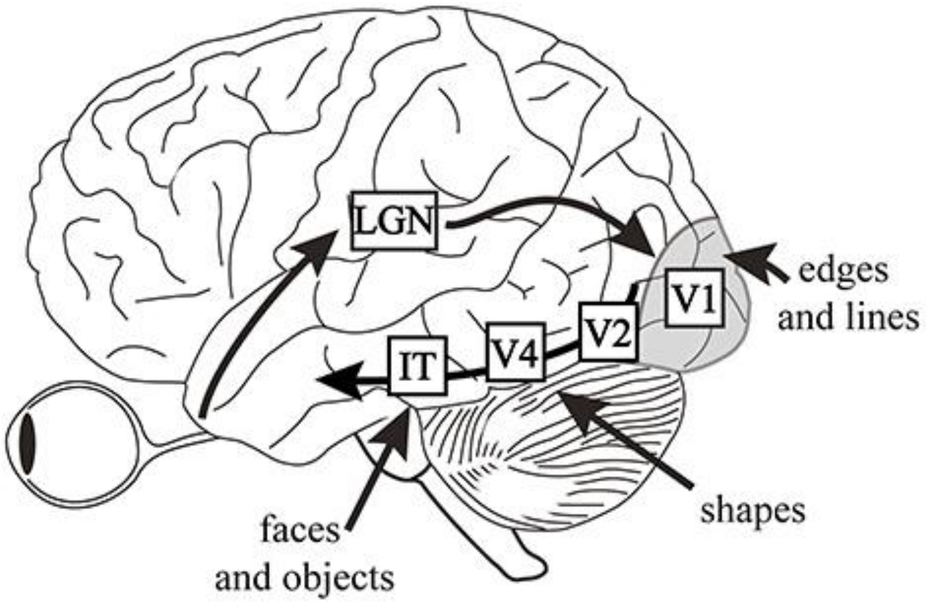


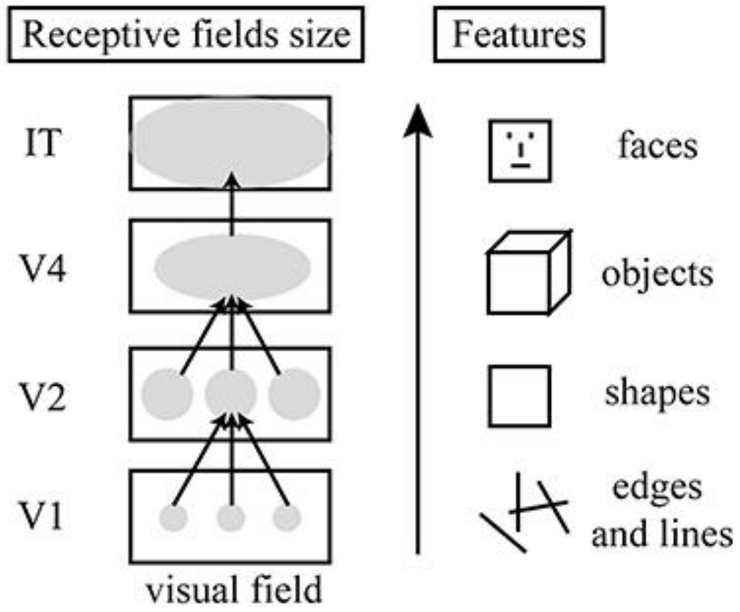




David Hubel et al. (1959) "Receptive fields of single neurones in the cat's striate cortex". The Journal of physiology 148.3 (1959): 574-591.

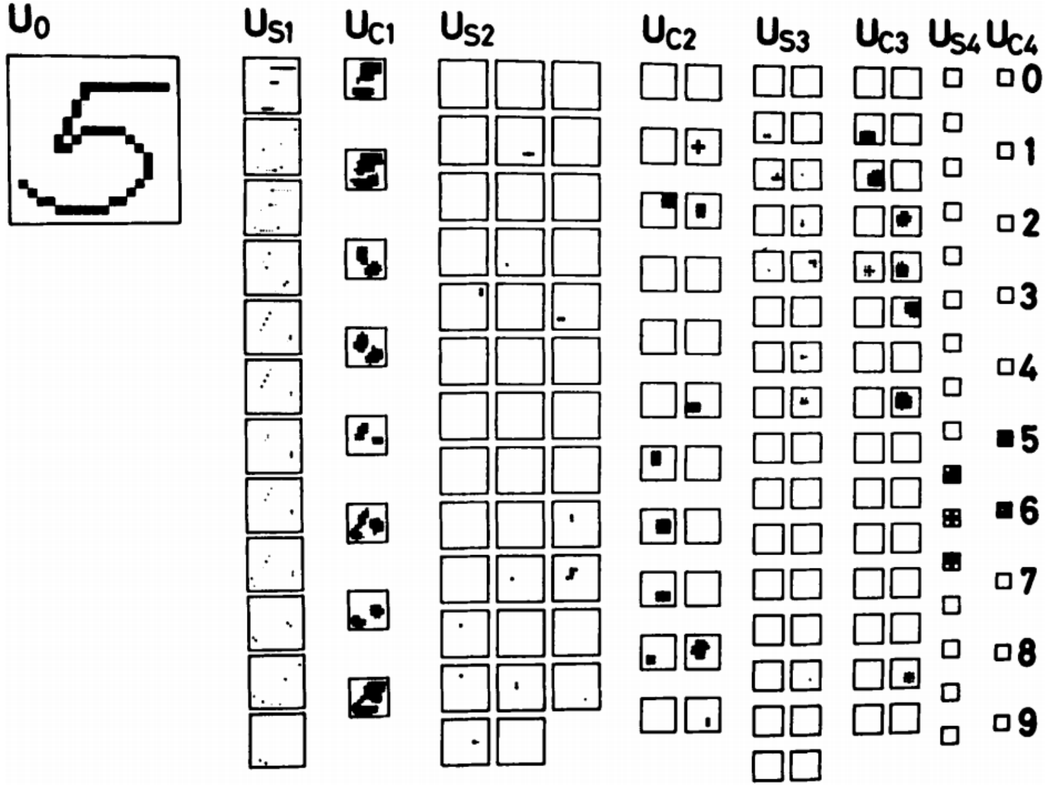
Corteza *visual*





David Hubel et al. (1959) "Receptive fields of single neurones in the cat's striate cortex". The Journal of physiology 148.3 (1959): 574-591.

Neocognitron



S-cell:

$$u_{Sl}(k_l, \boldsymbol{n}) = r_l \cdot \varphi\left(\frac{1 + \sum_{k_l=1}^{K_l-1} \sum_{v \in S_l} a_l(k_{l-1}, \boldsymbol{v}, k_l) \cdot u_{Cl-1}(k_{l-1}, \boldsymbol{n}+\boldsymbol{v})}{1 + \frac{2r_l}{1+r_l} \cdot b_l(k_l) \cdot v_{Cl-1}(\boldsymbol{n})}\right)$$

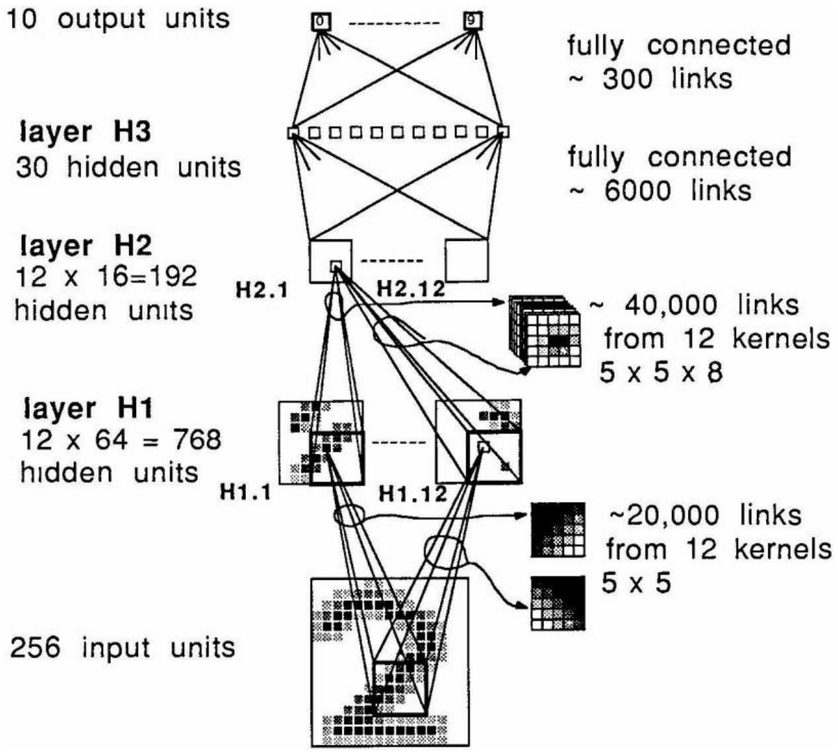
C-cell:

$$u_{Cl}(k_l, \boldsymbol{n}) = \psi\left(\frac{1 + \sum_{v \in D_l} d_l(\boldsymbol{v}) \cdot u_{Sl}(k_l, \boldsymbol{n}+\boldsymbol{v})}{1 + v_{Sl}(\boldsymbol{n})}\right)$$

donde $\varphi(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$ $\psi(x) = \varphi\left(\frac{x}{\alpha + x}\right)$

Fukushima, Kunihiko et al. (1983) "Neocognitron: A neural network model for a mechanism of visual pattern recognition". IEEE Transactions on Systems, Man, and Cybernetics .

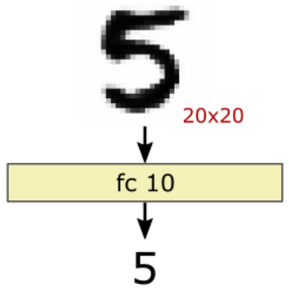
LeNet-1 *(1989)*



Yan LeCun (1989) "Generalization and network design strategies".
Connectionism in perspective.

LeNet

Perceptrón

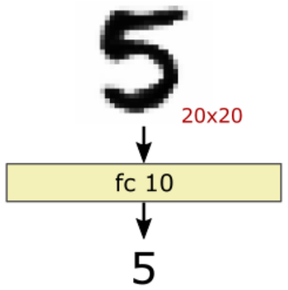


Error:
8.4%

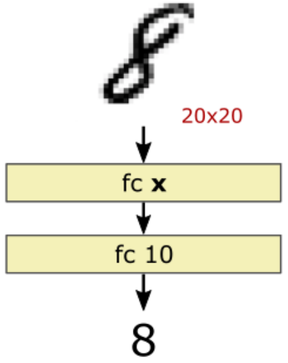
Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

LeNet

Perceptrón



MLP



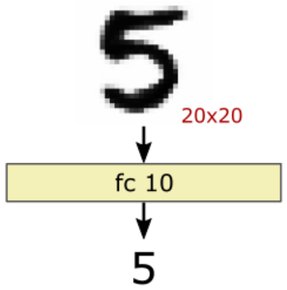
Error:
8.4%

Error:
3.6% (x:300)
3.8% (x:1000)

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

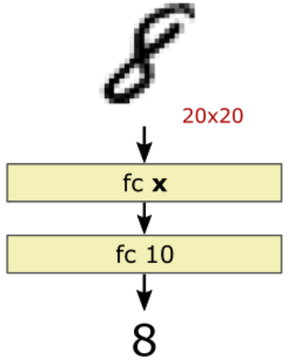
LeNet

Perceptrón



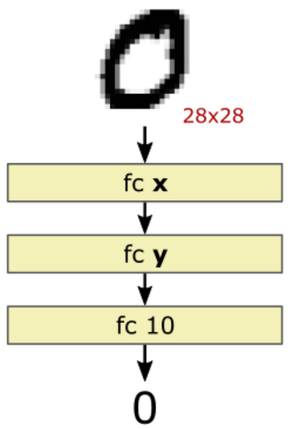
Error:
8.4%

MLP



Error:
3.6% (x:300)
3.8% (x:1000)

MLP

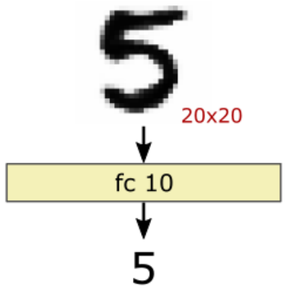


Error:
3.05% (x:300, y:100)
2.95% (x:1000,y:150)

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

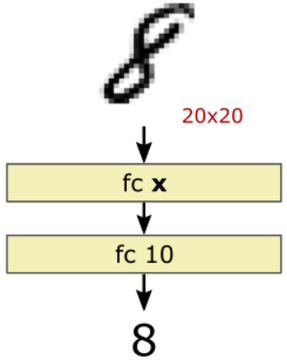
LeNet

Perceptrón



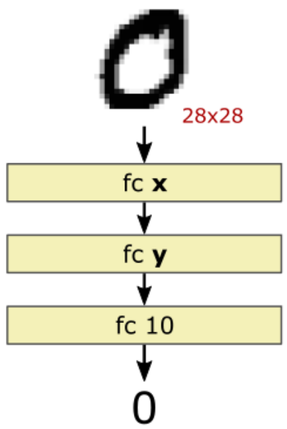
Error:
8.4%

MLP



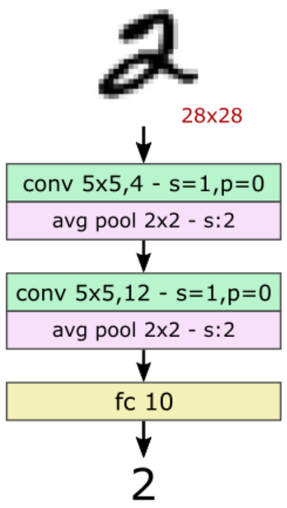
Error:
3.6% (x:300)
3.8% (x:1000)

MLP



Error:
3.05% (x:300, y:100)
2.95% (x:1000,y:150)

LeNet-1

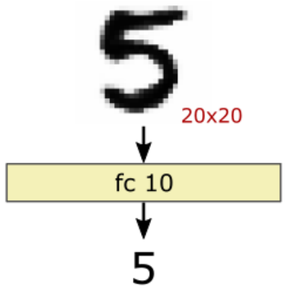


Error:
1.7%

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

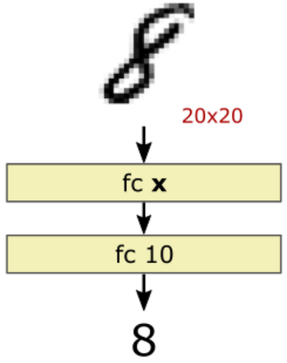
LeNet

Perceptrón



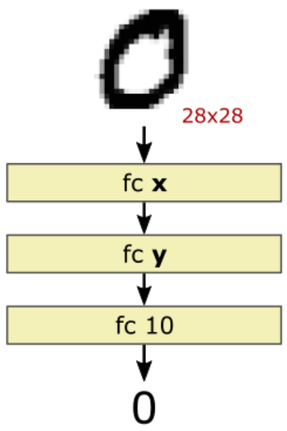
Error:
8.4%

MLP



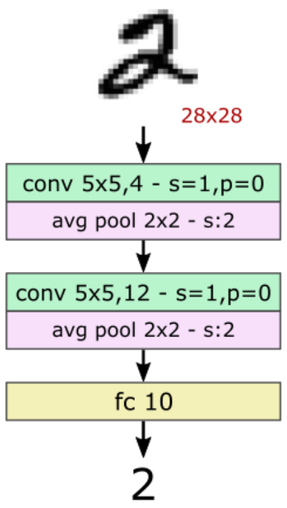
Error:
3.6% (x:300)
3.8% (x:1000)

MLP



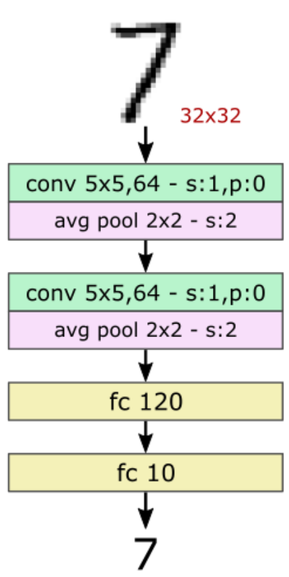
Error:
3.05% (x:300, y:100)
2.95% (x:1000,y:150)

LeNet-1



Error:
1.7%

LeNet-4

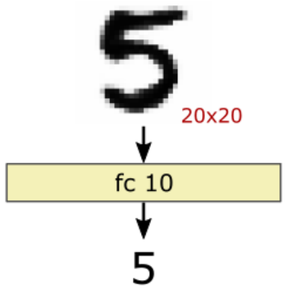


Error:
1.1%

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

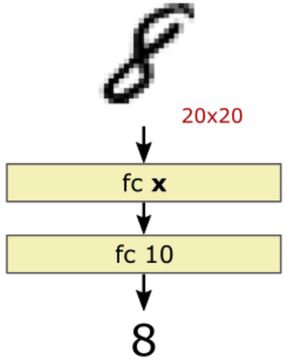
LeNet

Perceptrón



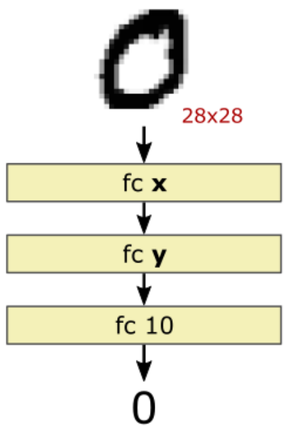
Error:
8.4%

MLP



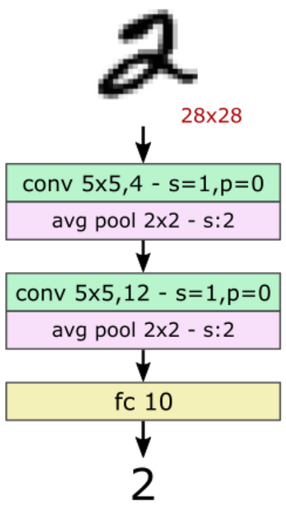
Error:
3.6% (x:300)
3.8% (x:1000)

MLP



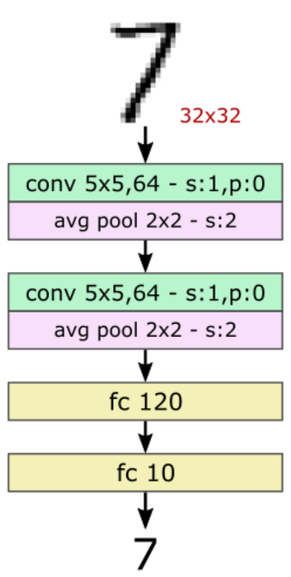
Error:
3.05% (x:300, y:100)
2.95% (x:1000,y:150)

LeNet-1



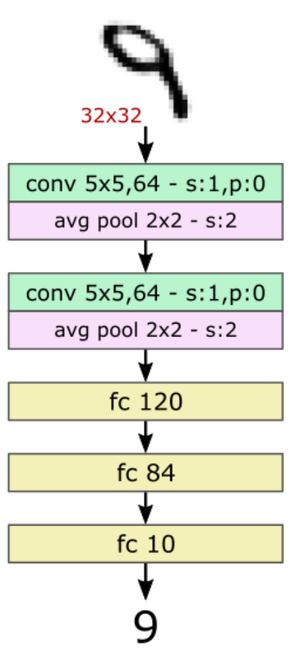
Error:
1.7%

LeNet-4



Error:
1.1%

LeNet-5

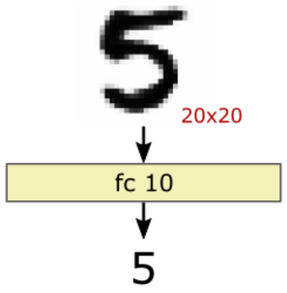


Error:
0.95%

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

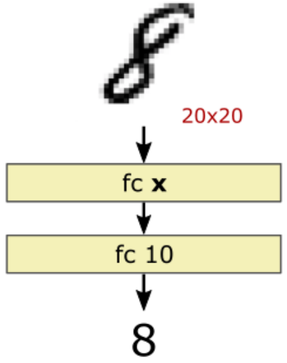
LeNet

Perceptrón



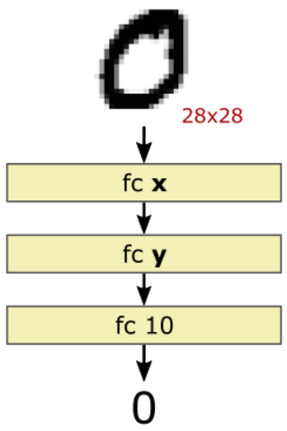
Error:
8.4%

MLP



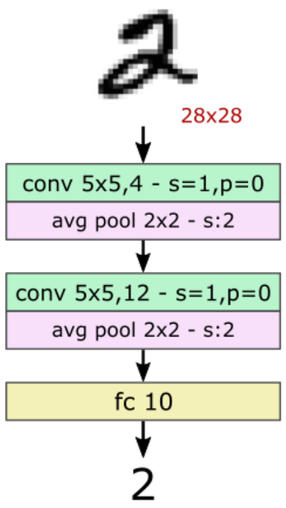
Error:
3.6% (x:300)
3.8% (x:1000)

MLP



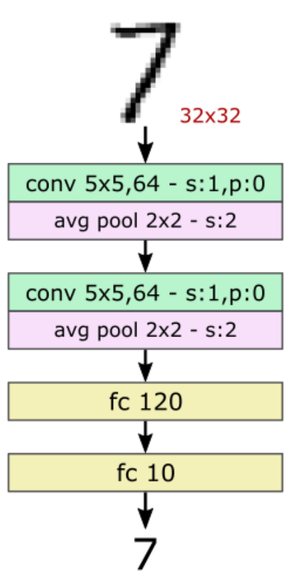
Error:
3.05% (x:300, y:100)
2.95% (x:1000,y:150)

LeNet-1



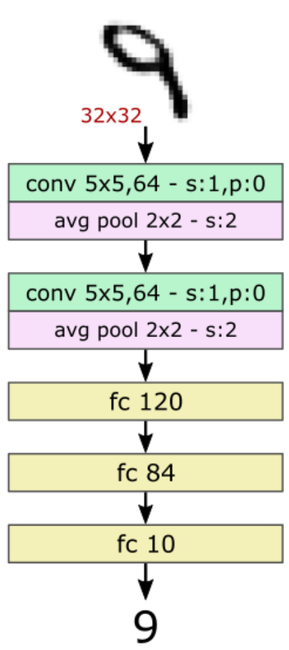
Error:
1.7%

LeNet-4



Error:
1.1%

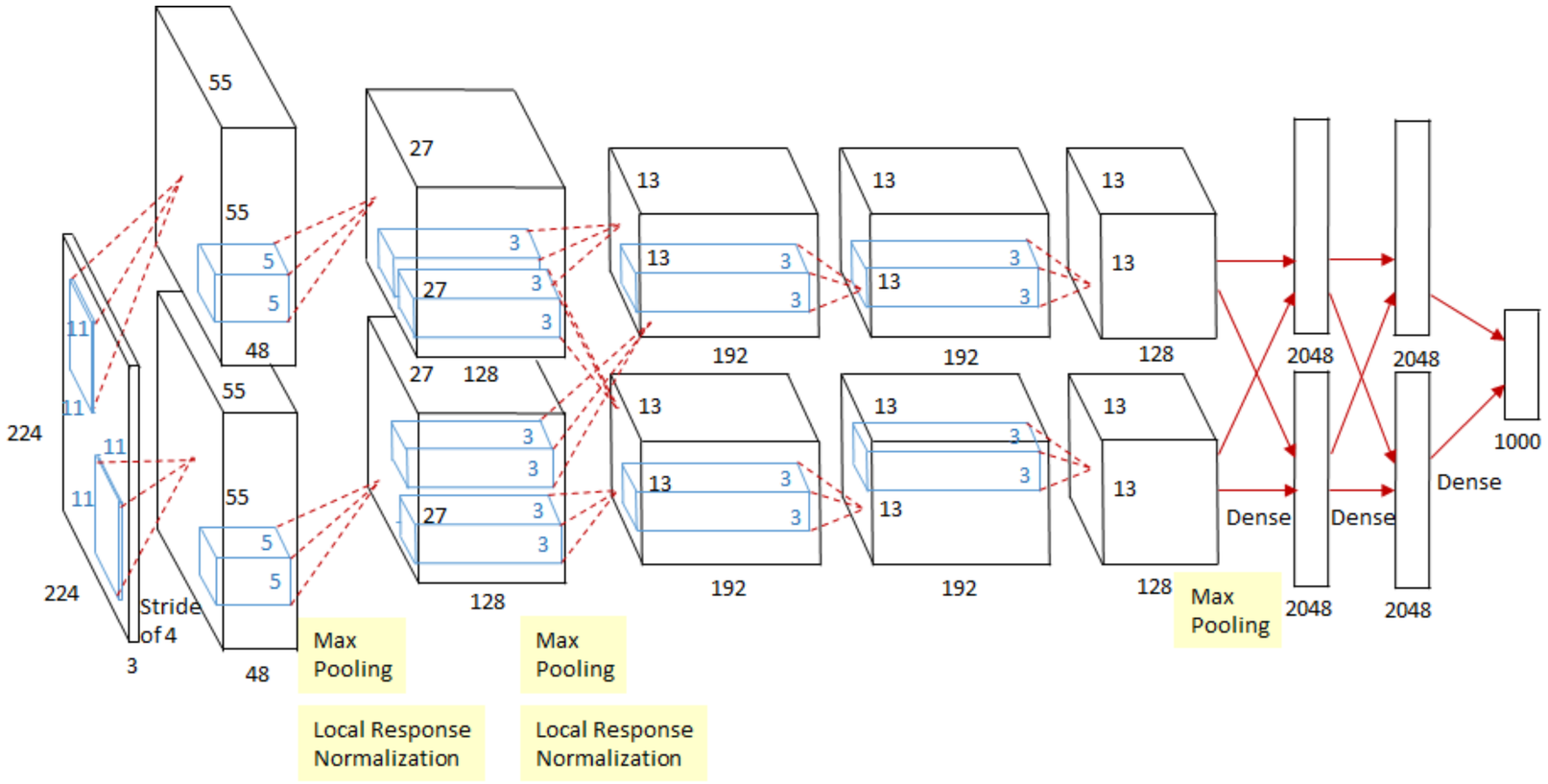
LeNet-5



Error:
0.95%

Yan LeCun et al. (1995) "Comparison of learning algorithms for handwritten digit recognition". International conference on artificial neural networks.

AlexNet



Alex Krizhevsky et al. (2012) "ImageNet Classification with Deep Convolutional Neural Networks". Advances in neural information processing systems.

AlexNet

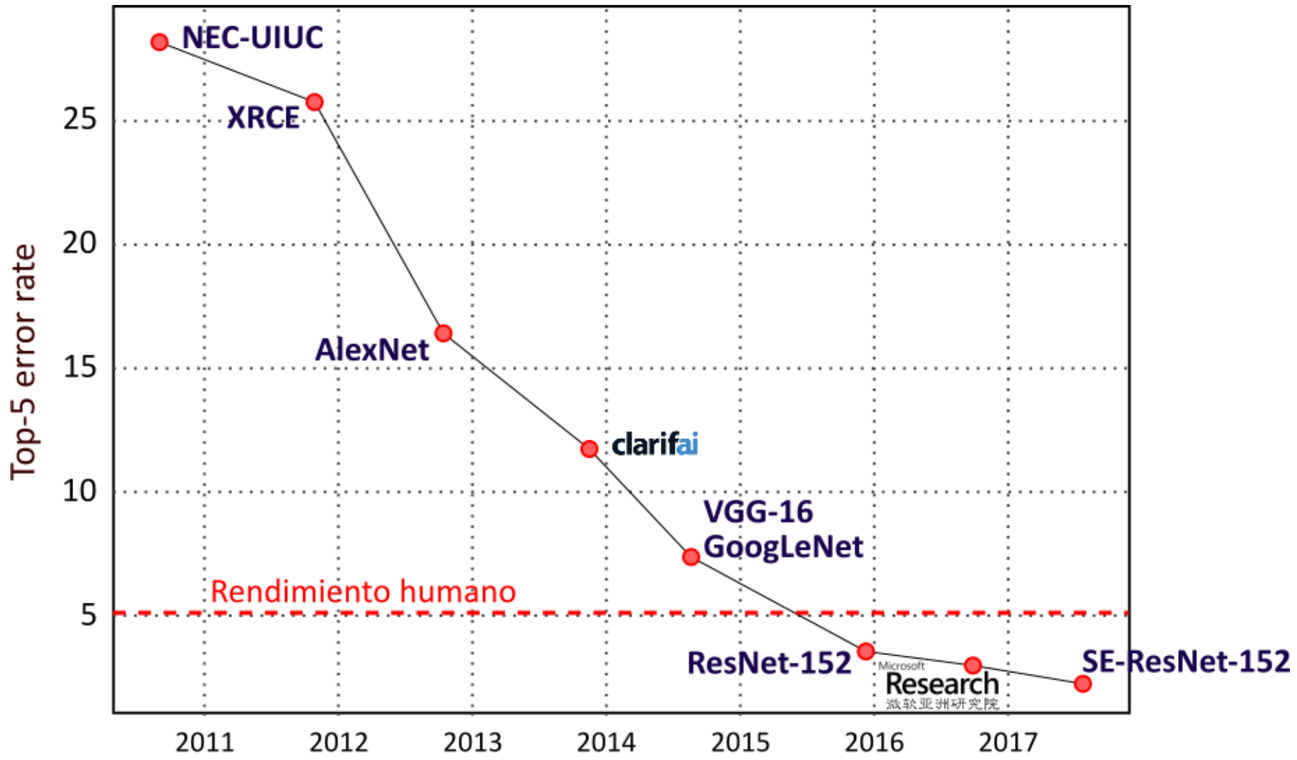
IMAGENET

- 14 197 122 imágenes
- 21 841 categorías

- 1 280 000 imágenes de entrenamiento
- 50 000 imágenes de evaluación
- 10 000 imágenes de prueba
- 1000 categorías

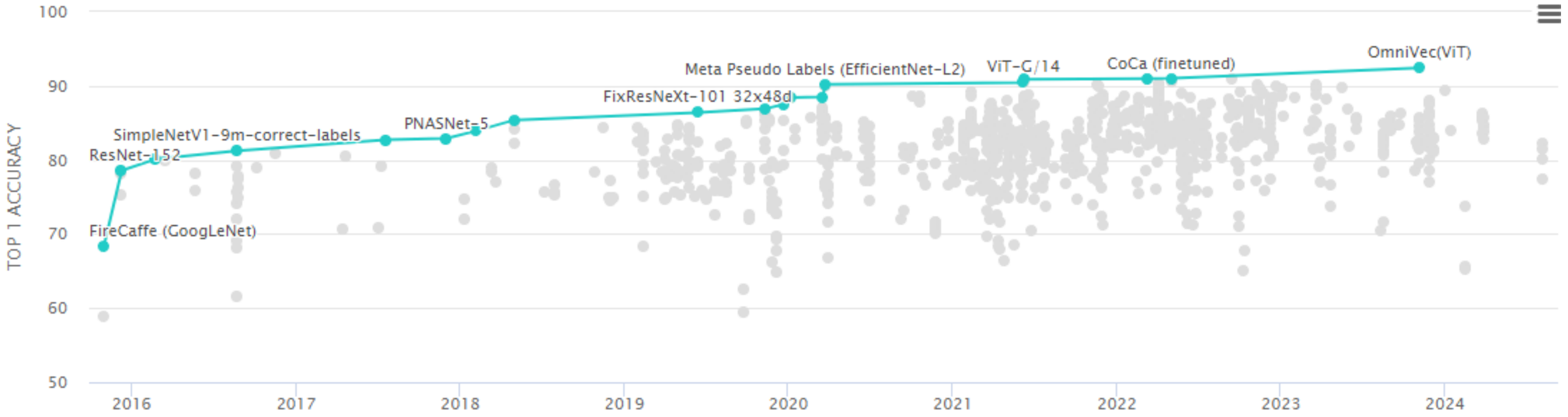
Jia Deng et al. (2009) "Imagenet: A large-scale hierarchical image database".
2009 IEEE Conference on Computer Vision and Pattern Recognition

AlexNet



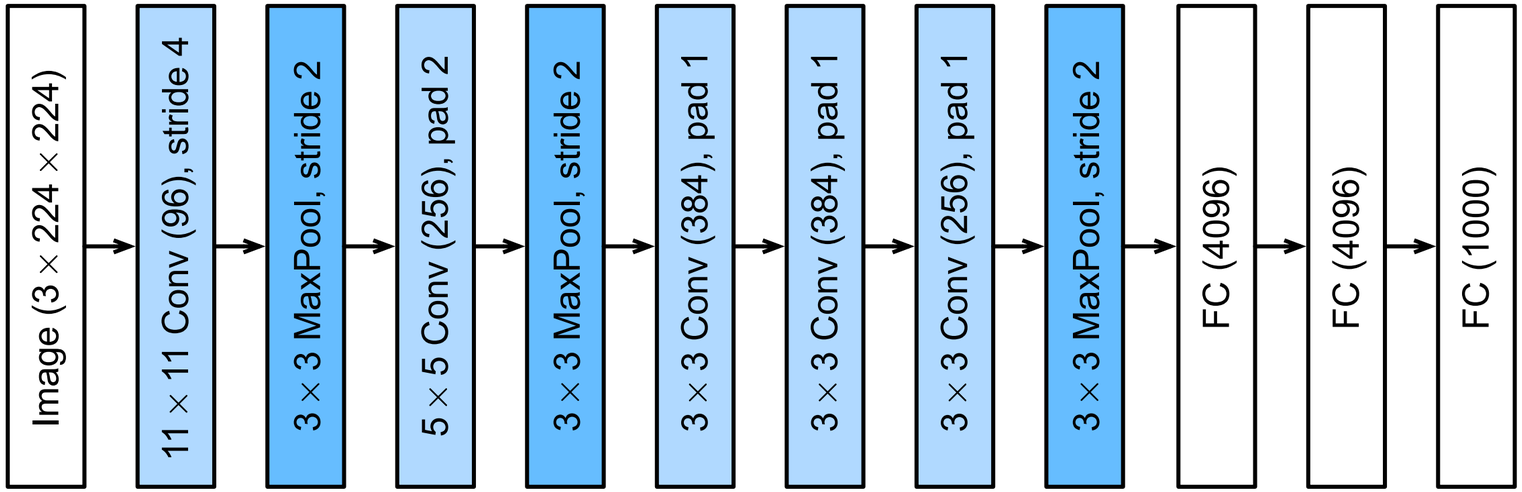
Jia Deng et al. (2009) "Imagenet: A large-scale hierarchical image database".
2009 IEEE Conference on Computer Vision and Pattern Recognition.

AlexNet



Jia Deng et al. (2009) "Imagenet: A large-scale hierarchical image database".
2009 IEEE Conference on Computer Vision and Pattern Recognition.

AlexNet



Alex Krizhevsky et al. (2012) "ImageNet Classification with Deep Convolutional Neural Networks". Advances in neural information processing systems.

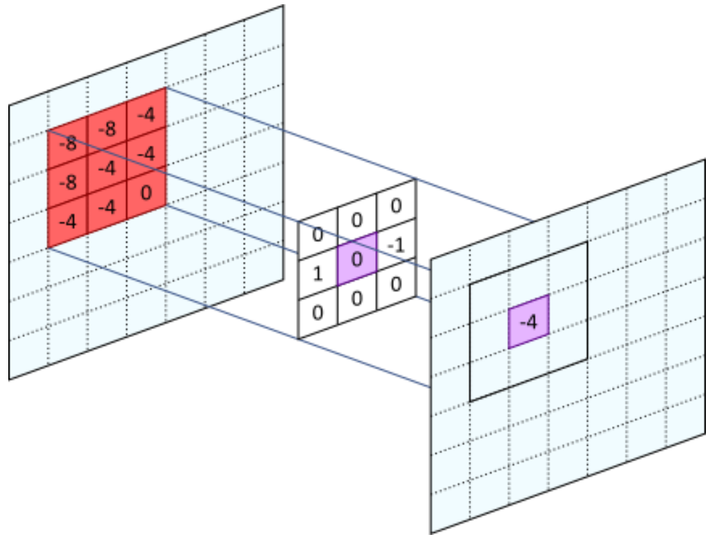
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

4.

VGG

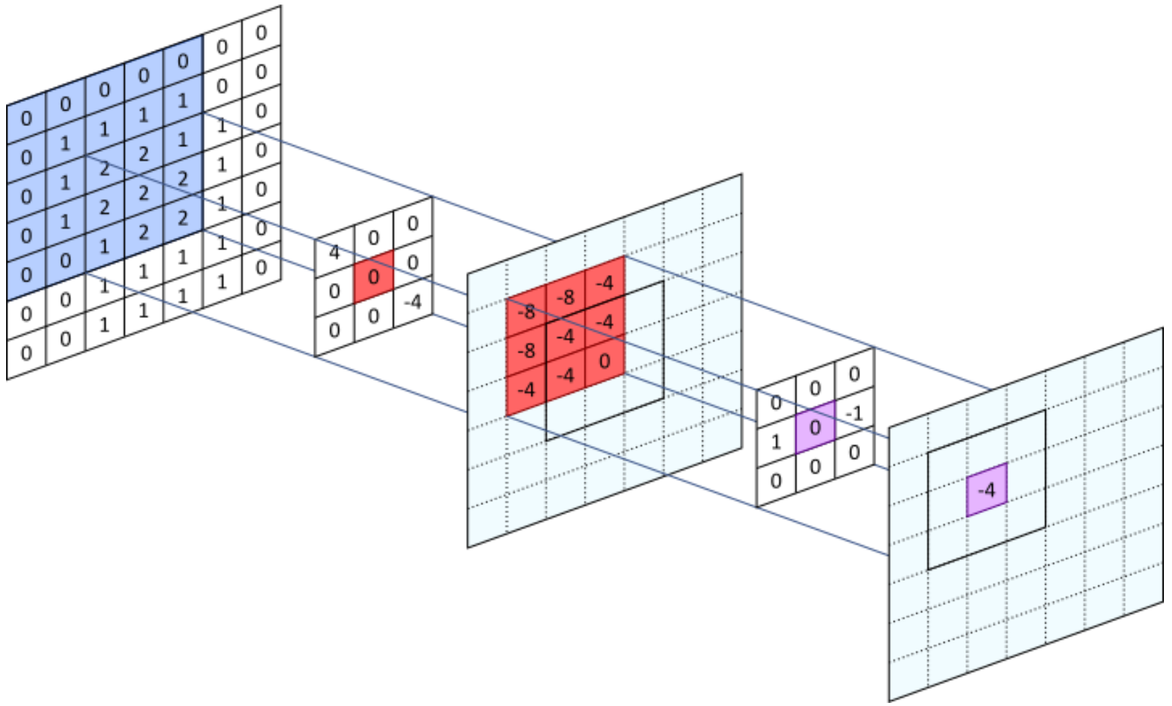
TRANSFORMATEC

VGG



Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

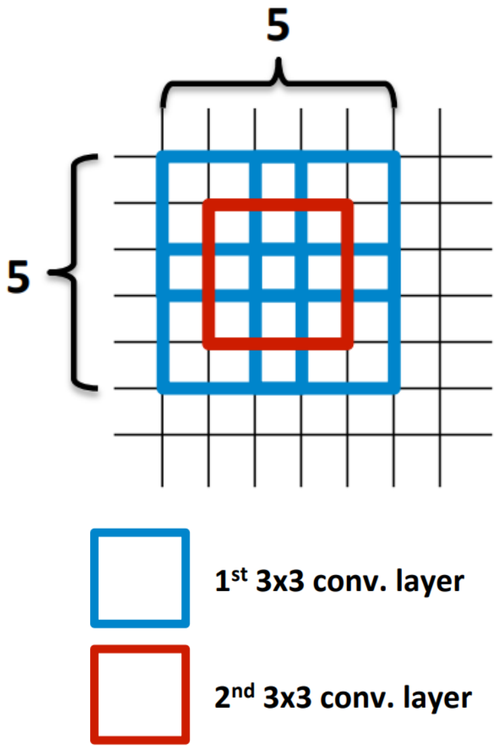
VGG



Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

VGG

Stacks of 3x3 Filters

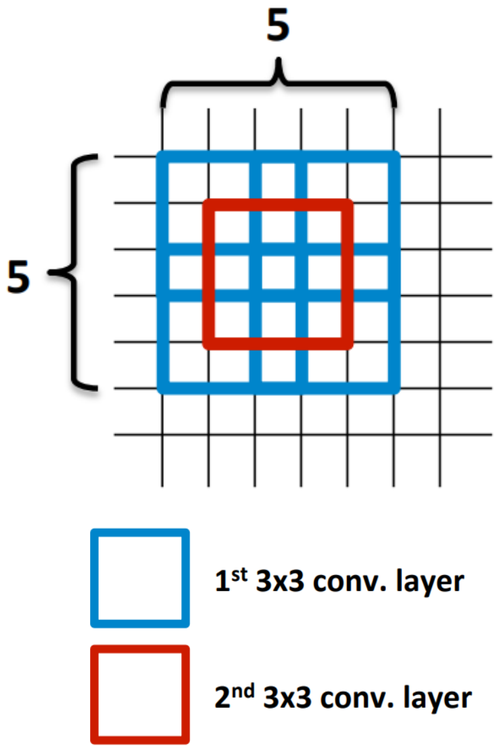


1 filtro 5×5, num. de parámetros 5×5=25
2 filtros 3×3, num. de parámetros 2×3×3=18
El número de parámetros se reduce en **28%**

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

VGG

Stacks of 3x3 Filters



1 filtro 5×5, num. de parámetros 5×5=25
2 filtros 3×3, num. de parámetros 2×3×3=18
El número de parámetros se reduce en **28%**

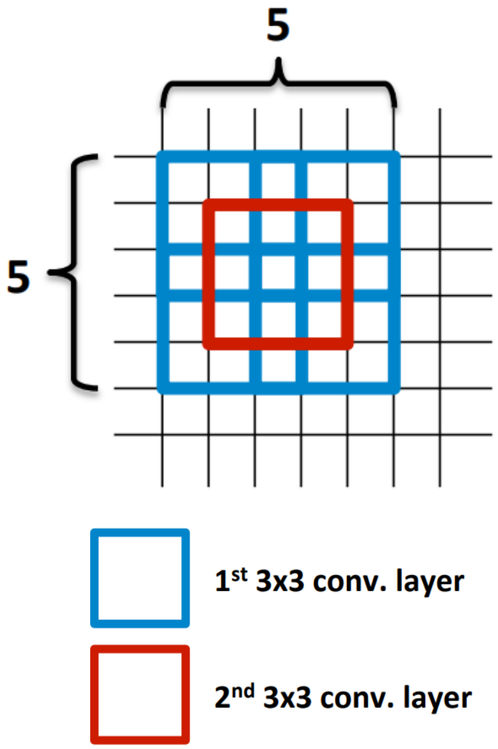
ZFNet

1 filtro 7×7, num. de parámetros: 7×7=49
3 filtros 3×3, num. de parámetros: 3×3×3=27
El número de parámetros se reduce en **45%**

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

VGG

Stacks of 3x3 Filters



1 filtro 5×5, num. de parámetros 5×5=25
2 filtros 3×3, num. de parámetros 2×3×3=18
El número de parámetros se reduce en **28%**

ZFNet

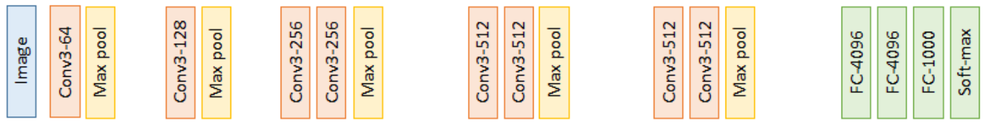
1 filtro 7×7, num. de parámetros: 7×7=49
3 filtros 3×3, num. de parámetros: 3×3×3=27
El número de parámetros se reduce en **45%**

AlexNet

1 filtro 11×11, num. de parámetros = 11×11=121
5 filtros 3×3, num. de parámetros = 3×3×5=45
El número de parámetros se reduce en **63%**

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

VGG

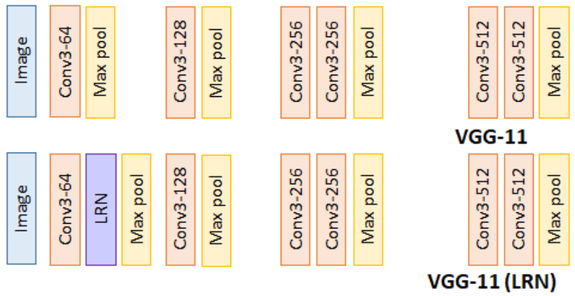


VGG-11

Number of Parameters (millions)	Top-5 Error Rate (%)
133	10.4

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

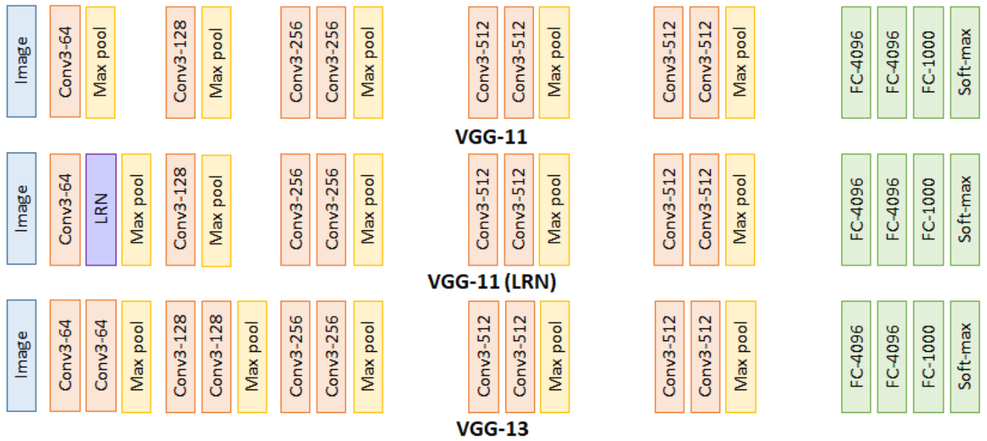
VGG



	Number of Parameters (millions)	Top-5 Error Rate (%)
VGG-11	133	10.4
VGG-11 (LRN)	133	10.5

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

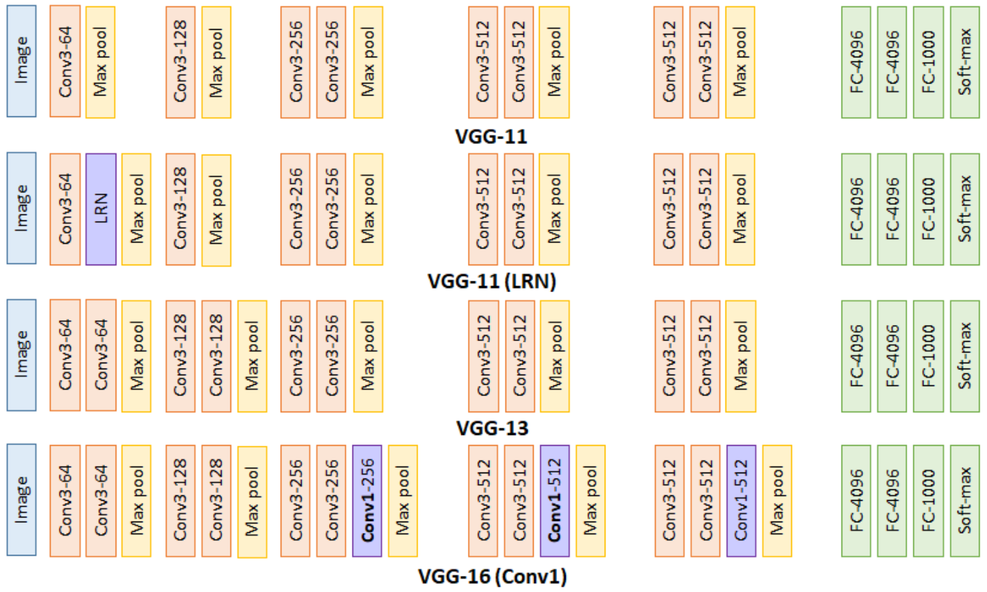
VGG



	Number of Parameters (millions)	Top-5 Error Rate (%)
VGG-11	133	10.4
VGG-11 (LRN)	133	10.5
VGG-13	133	9.9

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

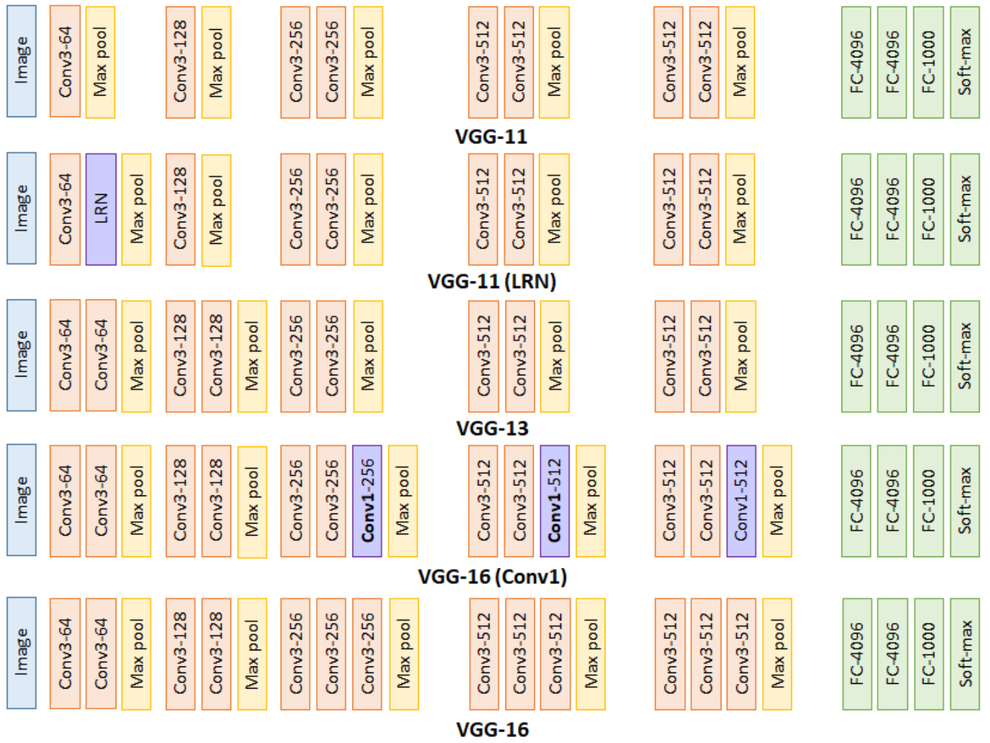
VGG



	Number of Parameters (millions)	Top-5 Error Rate (%)
VGG-11	133	10.4
VGG-11 (LRN)	133	10.5
VGG-13	133	9.9
VGG-16 (Conv1)	134	9.4

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

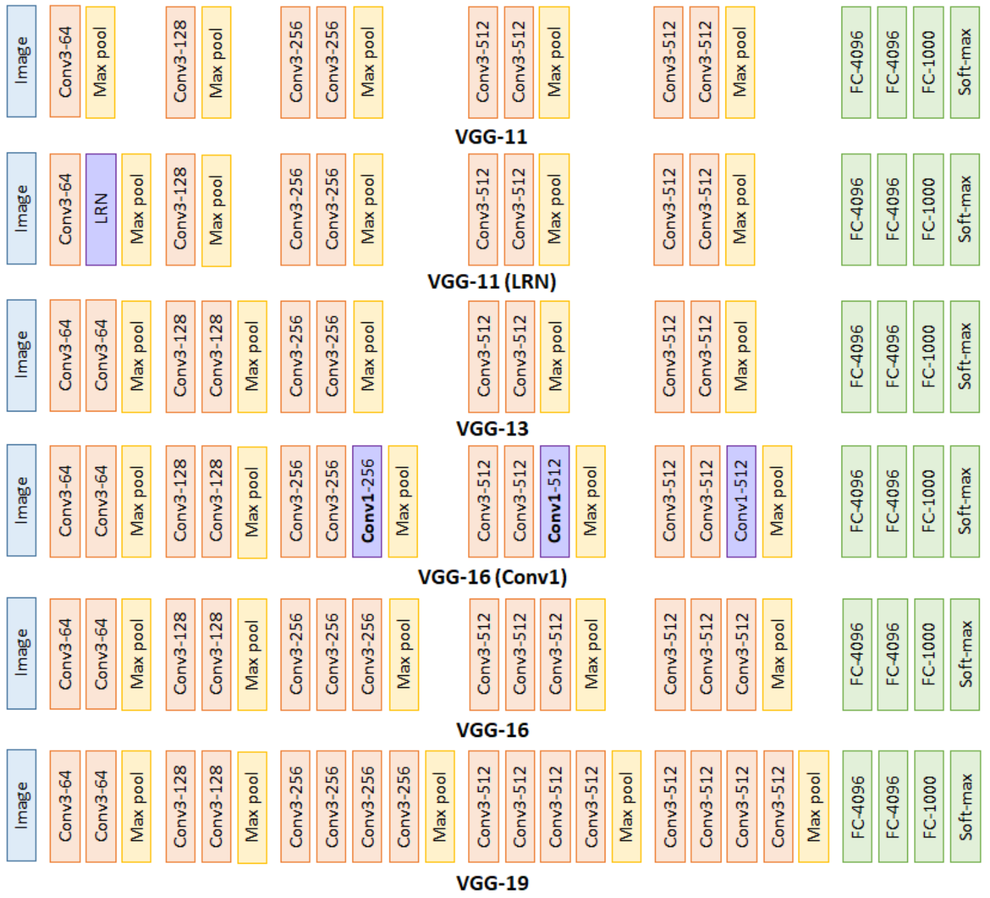
VGG



	Number of Parameters (millions)	Top-5 Error Rate (%)
VGG-11	133	10.4
VGG-11 (LRN)	133	10.5
VGG-13	133	9.9
VGG-16 (Conv1)	134	9.4
VGG-16	138	8.8

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

VGG



	Number of Parameters (millions)	Top-5 Error Rate (%)
VGG-11	133	10.4
VGG-11 (LRN)	133	10.5
VGG-13	133	9.9
VGG-16 (Conv1)	134	9.4
VGG-16	138	8.8
VGG-19	144	9.0

Karen Simonyan, Andrew Zisserman (2015) "Very Deep Convolutional Networks for Large-Scale Image Recognition". International Conference on Learning Representations (ICRL)

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

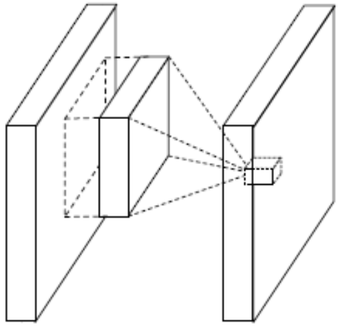
5.

Inception

TRANSFORMATEC

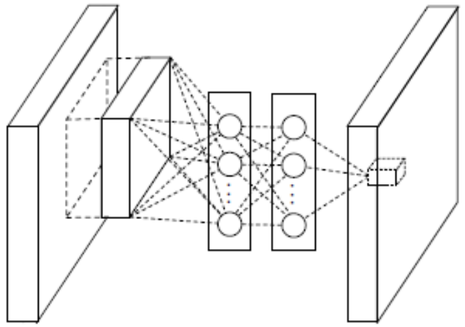
Network *in Network*

Linear Convolutional Layer



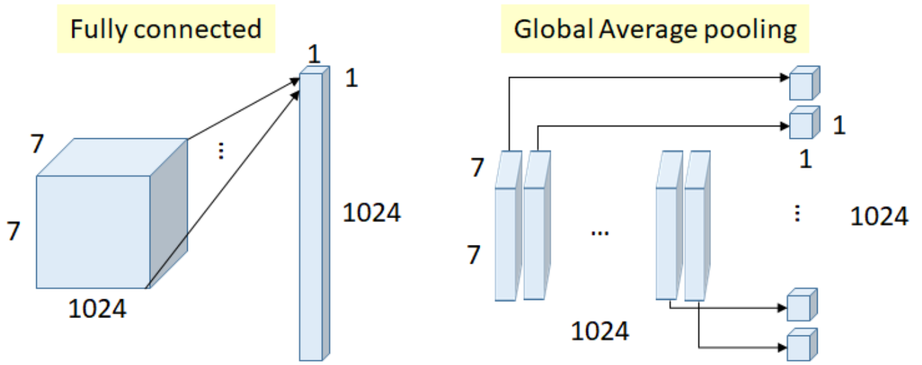
$$f_{i,j,k} = \max(w_k^T x_{i,j}, 0).$$

Mlpconv Layer



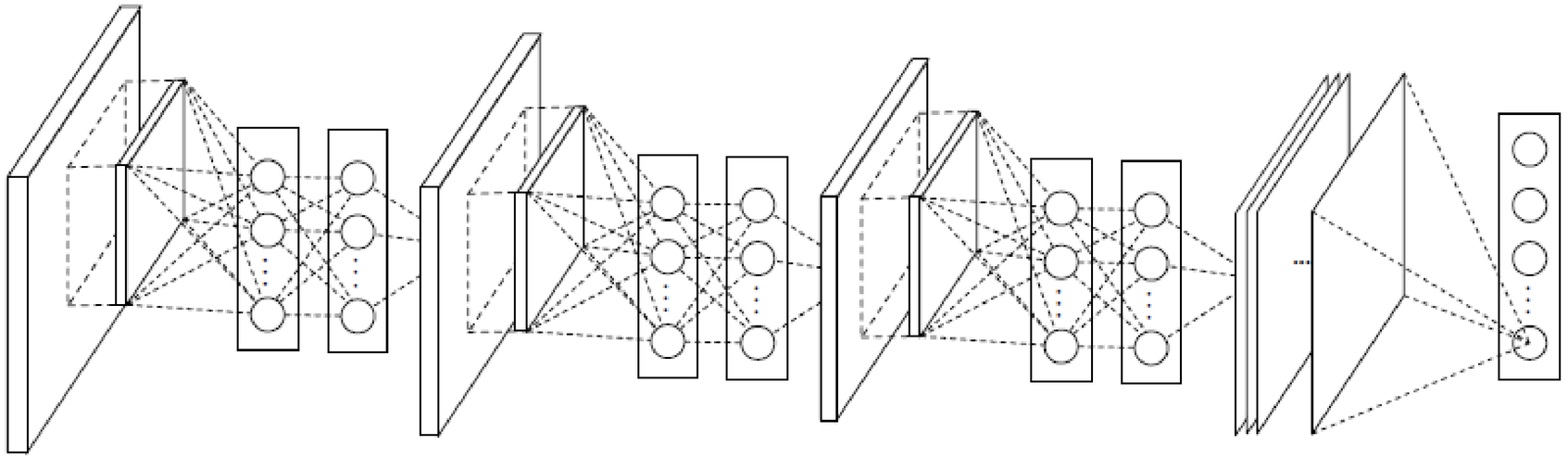
$$\begin{aligned} f^1_{i,j,k_1} &= \max({w^1_{k_1}}^T x_{i,j} + b_{k_1}, 0). \\ &\vdots \\ f^n_{i,j,k_n} &= \max({w^n_{k_n}}^T f^{n-1}_{i,j} + b_{k_n}, 0). \end{aligned}$$

Fully Connected Layer VS Global Average Pooling Layer



Lin, Min et al. (2013) "Network in Network". arXiv preprint arXiv:1312.4400.

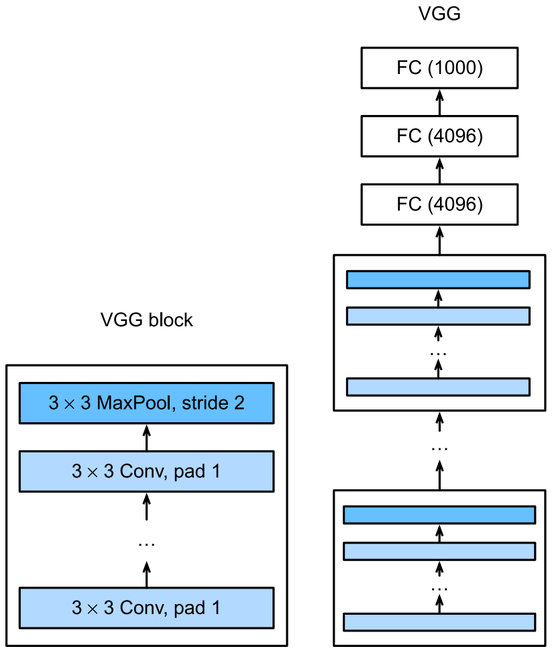
Network *in Network*

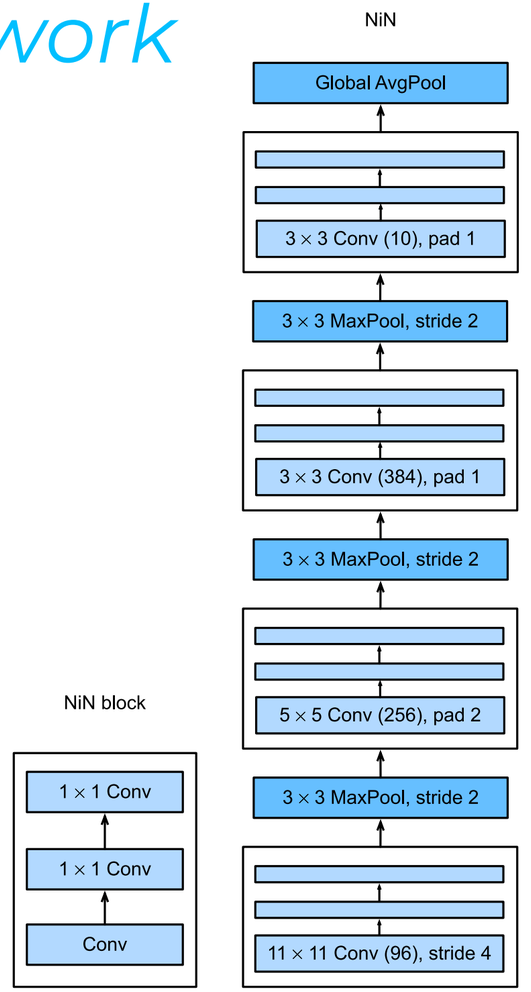


Mlpconv Layer | Mlpconv Layer | Mlpconv Layer | Global Average Pooling

Lin, Min et al. (2013) "Network in Network". arXiv preprint arXiv:1312.4400.

Network *in Network*

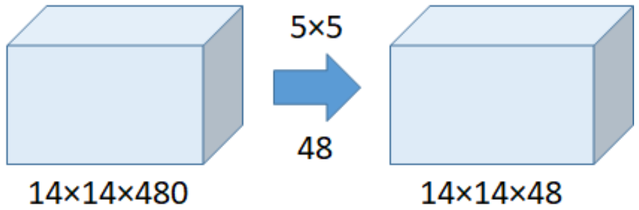




Lin, Min et al. (2013) "Network in Network". arXiv preprint arXiv:1312.4400.

GoogLeNet

Bottleneck

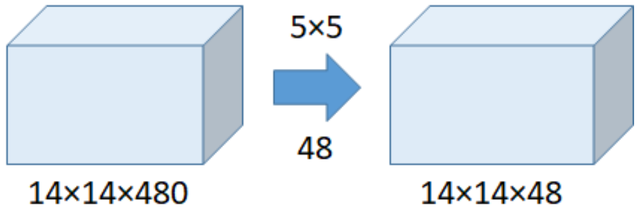


Número de operaciones = (14×14×48)×(5×5×480) = **112.9M**

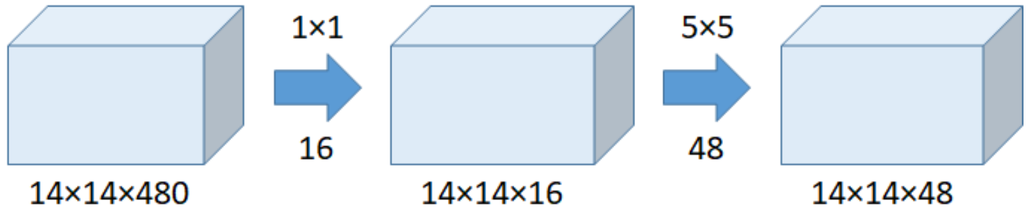
Christian Szegedy et al. (2015) "Going deeper with convolutions".
Proceedings of the IEEE conference on computer vision and pattern recognition.

GoogLeNet

Bottleneck



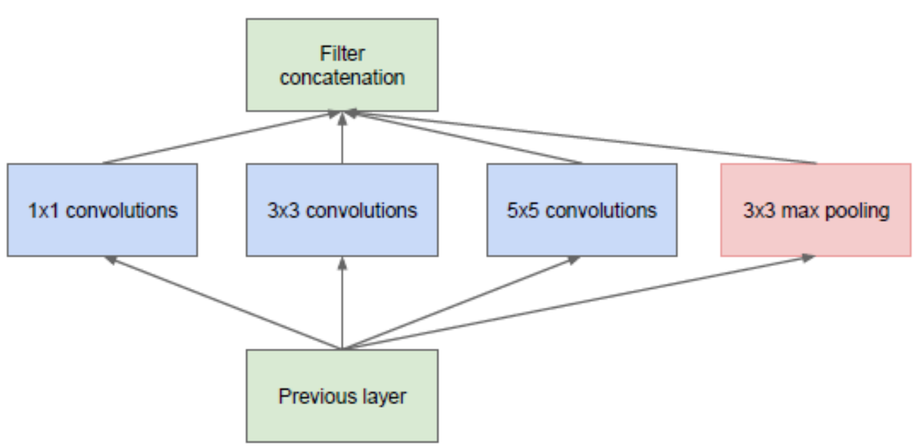
Número de operaciones = (14×14×48)×(5×5×480) = **112.9M**



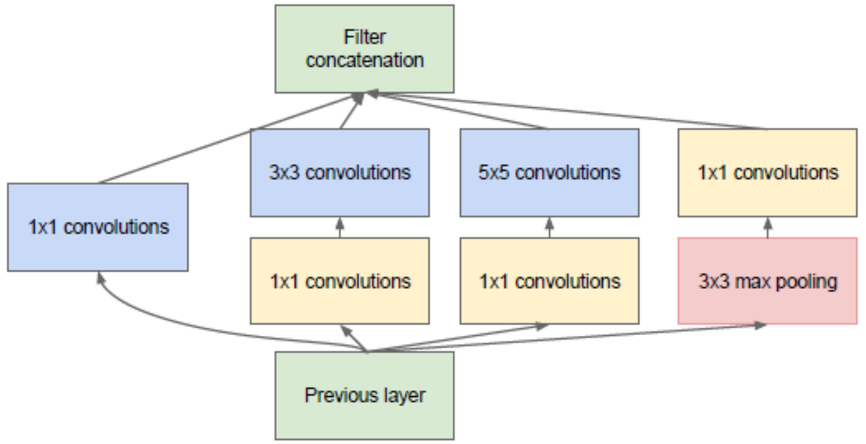
Número de operaciones 1×1 = (14×14×16)×(1×1×480) = 1.5M
Número de operaciones 5×5 = (14×14×48)×(5×5×16) = 3.8M
Número total de operaciones = 1.5M + 3.8M = **5.3M**

Christian Szegedy et al. (2015) "Going deeper with convolutions".
Proceedings of the IEEE conference on computer vision and pattern recognition.

GoogLeNet



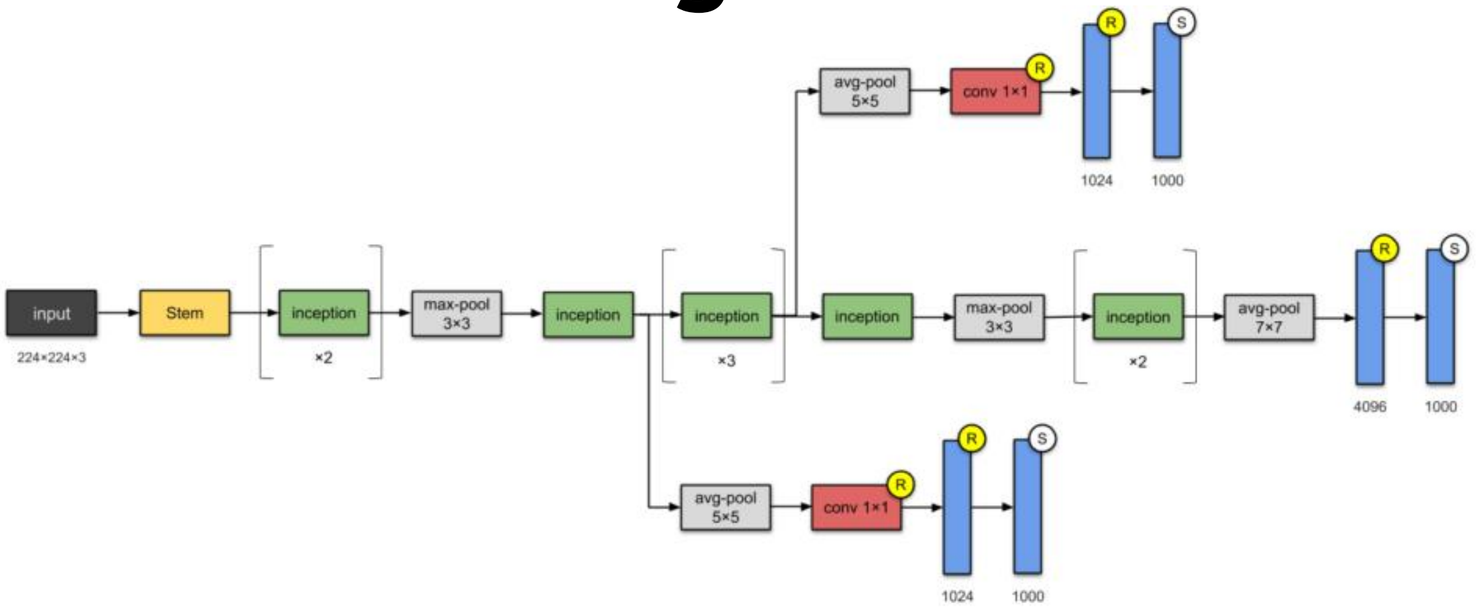
(a) Inception module, naïve version

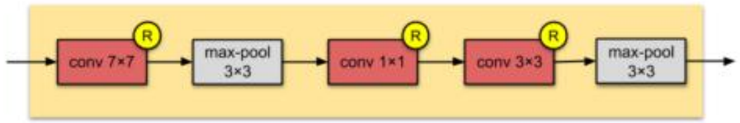


(b) Inception module with dimensionality reduction

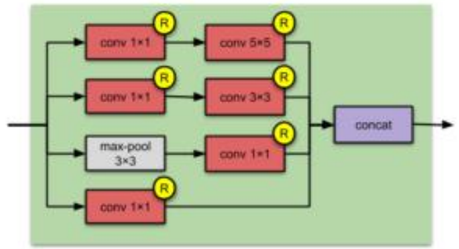
Christian Szegedy et al. (2015) "Going deeper with convolutions". Proceedings of the IEEE conference on computer vision and pattern recognition.

GoogLeNet





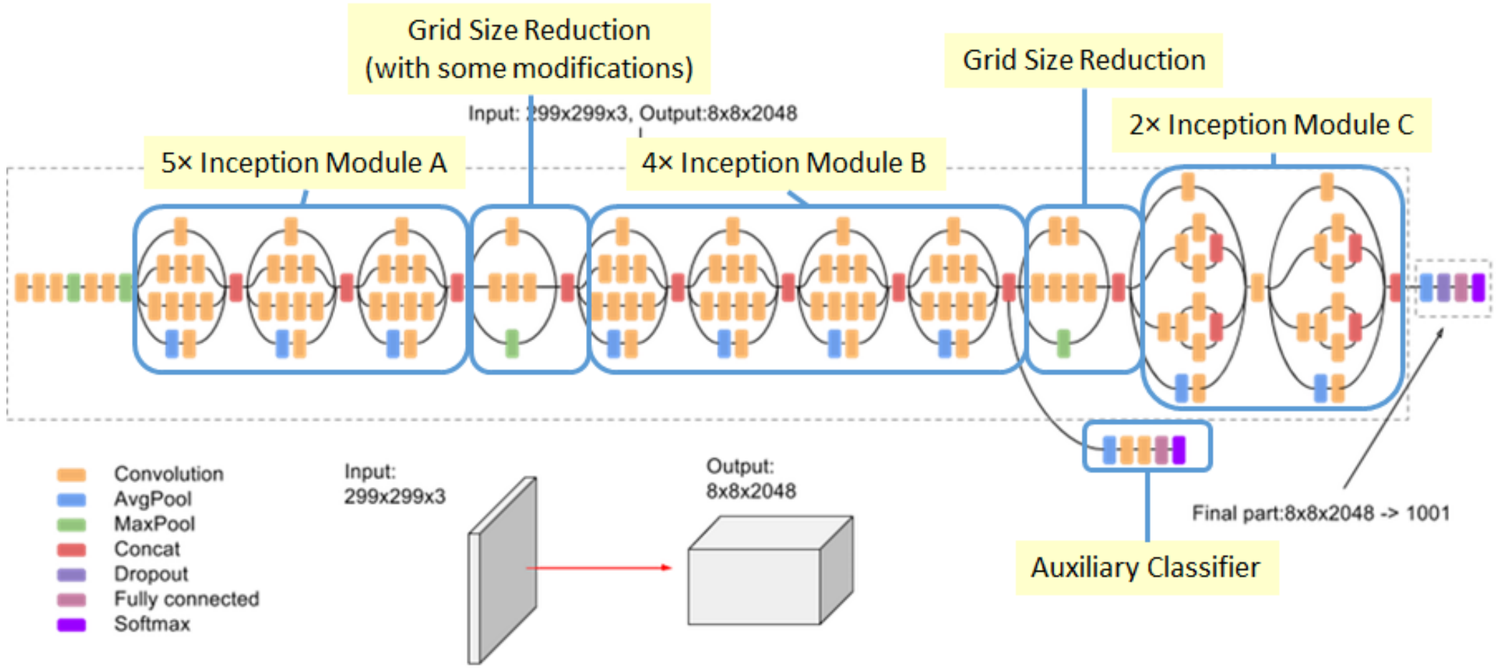
Stem



Inception module

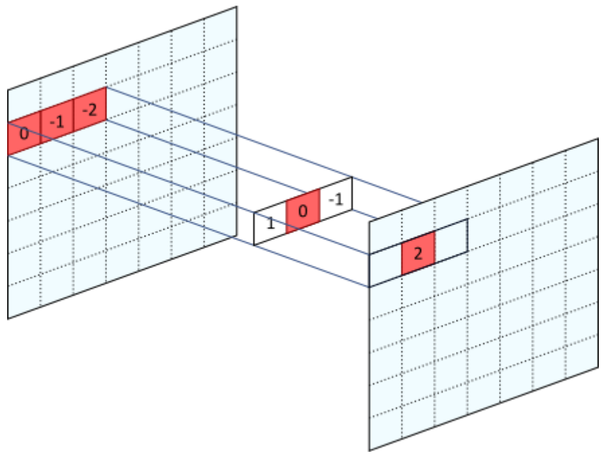
Christian Szegedy et al. (2015) "Going deeper with convolutions".
Proceedings of the IEEE conference on computer vision and pattern recognition.

Inception *v3*



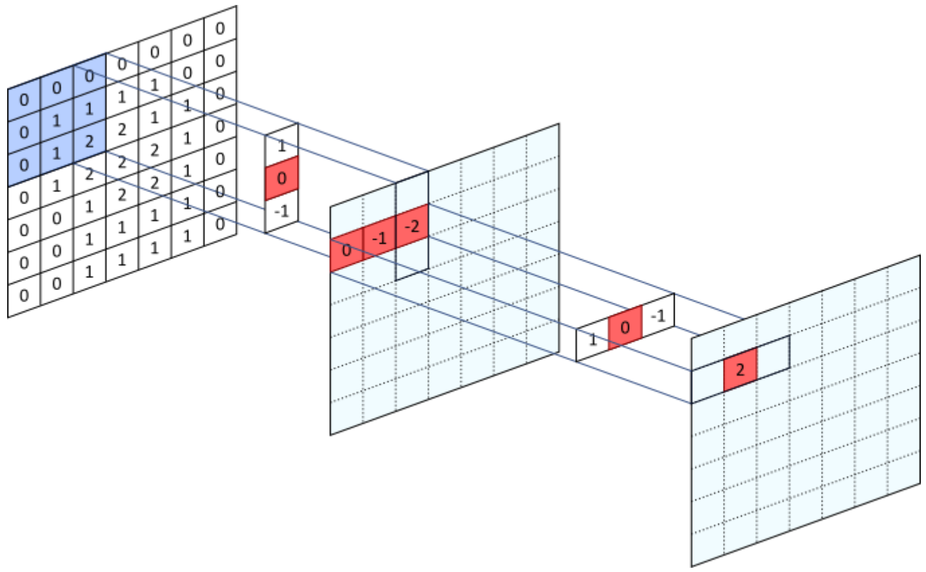
Christian Szegedy et al. (2015) "Rethinking the Inception Architecture for Computer Vision". Proceedings of the IEEE conference on computer vision and pattern recognition

Inception *v3*



Christian Szegedy et al. (2015) "Rethinking the Inception Architecture for Computer Vision". Proceedings of the IEEE conference on computer vision and pattern recognition

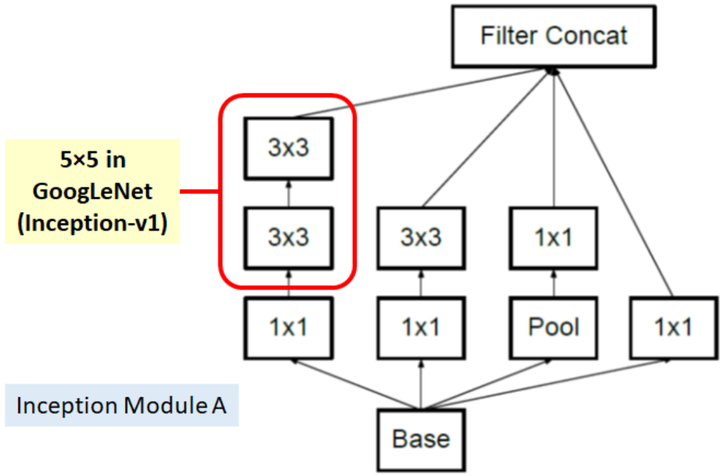
Inception *v3*



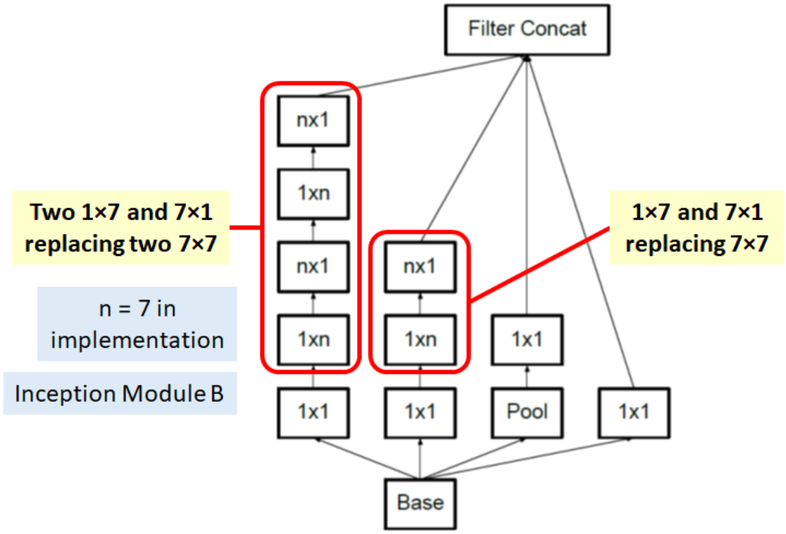
Christian Szegedy et al. (2015) "Rethinking the Inception Architecture for Computer Vision". Proceedings of the IEEE conference on computer vision and pattern recognition

Inception *v3*

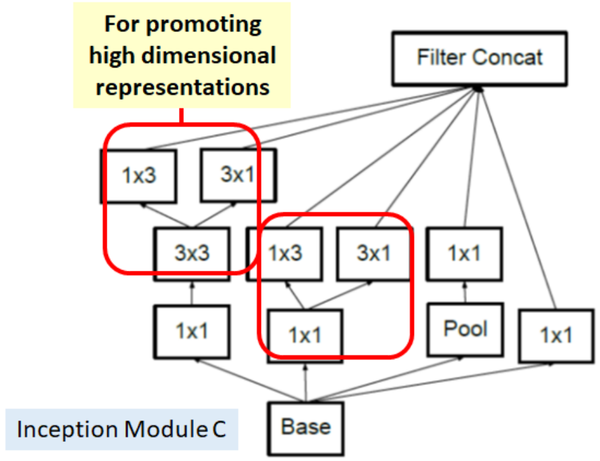
Inception Module A



Inception Module B

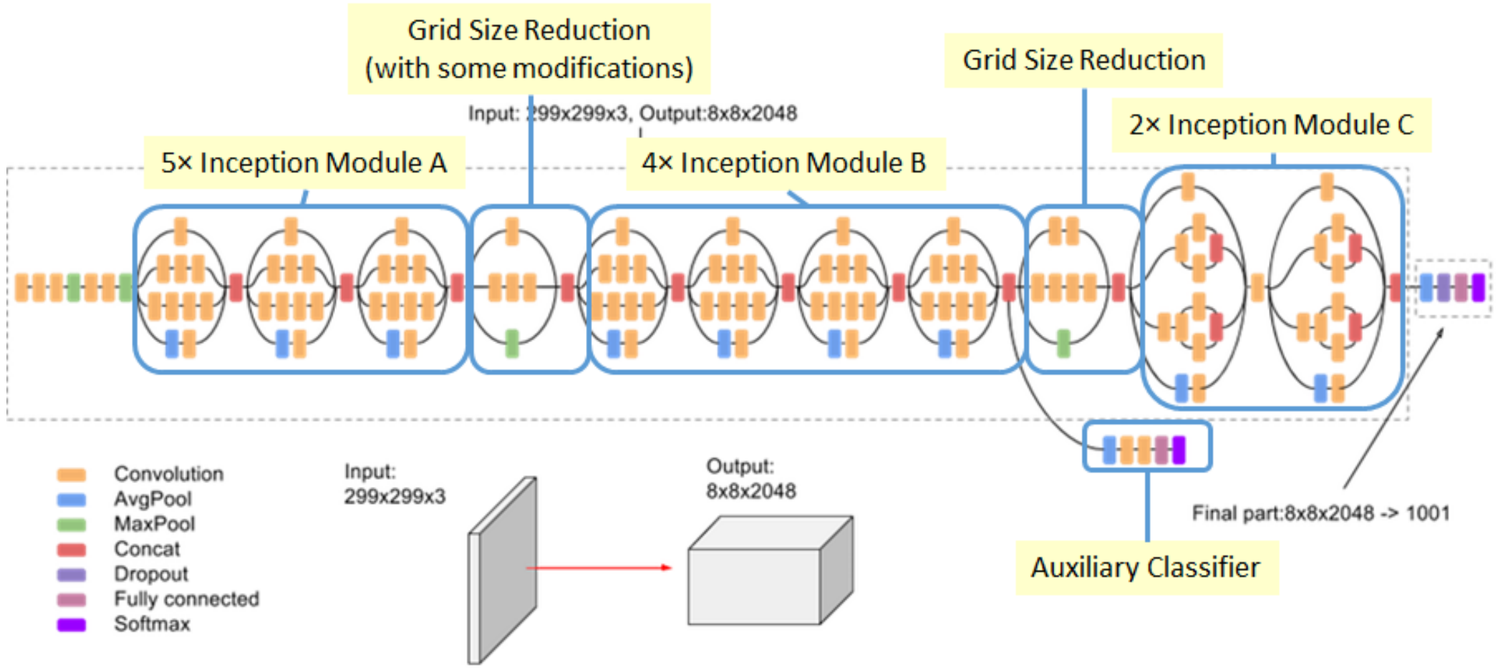


Inception Module C



Christian Szegedy et al. (2015) "Rethinking the Inception Architecture for Computer Vision". Proceedings of the IEEE conference on computer vision and pattern recognition

Inception *v3*



Christian Szegedy et al. (2015) "Rethinking the Inception Architecture for Computer Vision". Proceedings of the IEEE conference on computer vision and pattern recognition

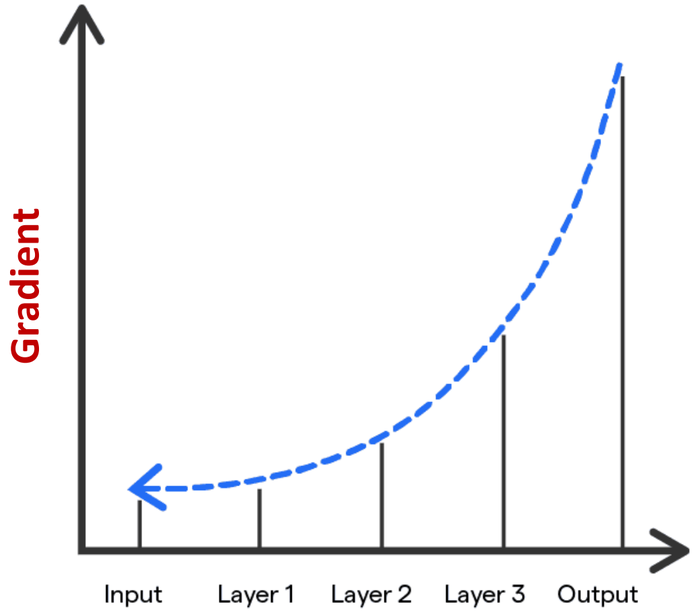
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

6.

ResNet

TRANSFORMATEC

Vanishing gradient problem



Supongamos una red con L capas, y cada capa l tiene un conjunto de pesos W_l y un conjunto de activaciones A_l. En un forward pass, para una capa l, tenemos:

$$Z_l = W_l A_{l-1} + b_l \qquad A_l = \sigma(Z_l)$$

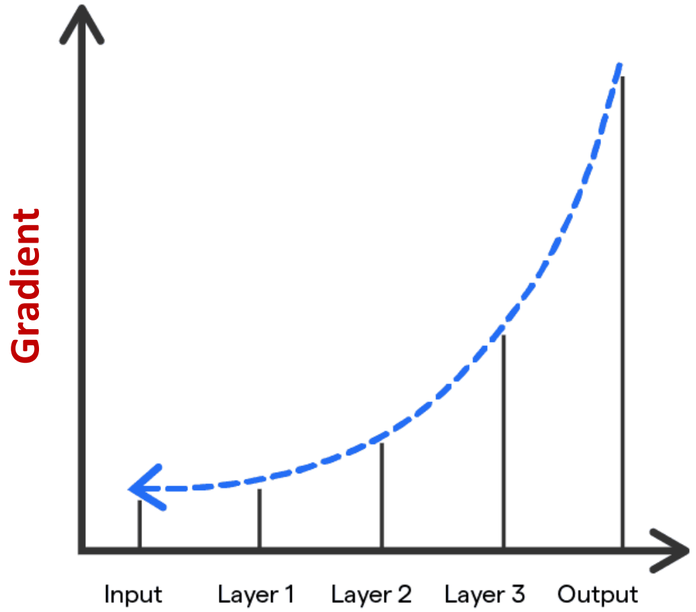
Consideremos la derivada de la función de pérdida $\mathcal{L}$ con respecto a los pesos en la capa l:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \delta_l A_{l-1}^{\top} \qquad \delta_l = \frac{\partial \mathcal{L}}{\partial Z_l} = \left(W_{l+1}^{\top}\delta_{l+1}\right) \odot \sigma'(Z_l)$$

Podemos desarrollar esto recursivamente:

$$\delta_l = \left(W_{l+1}^{\top}\left(W_{l+2}^{\top}\left(\cdots\left(W_L^{\top}\delta_L\right)\right)\right)\right) \odot \sigma'(Z_l)$$

Vanishing gradient problem



Supongamos una red con L capas, y cada capa l tiene un conjunto de pesos W_l y un conjunto de activaciones A_l. En un forward pass, para una capa l, tenemos:

$$Z_l = W_l A_{l-1} + b_l \qquad A_l = \sigma(Z_l)$$

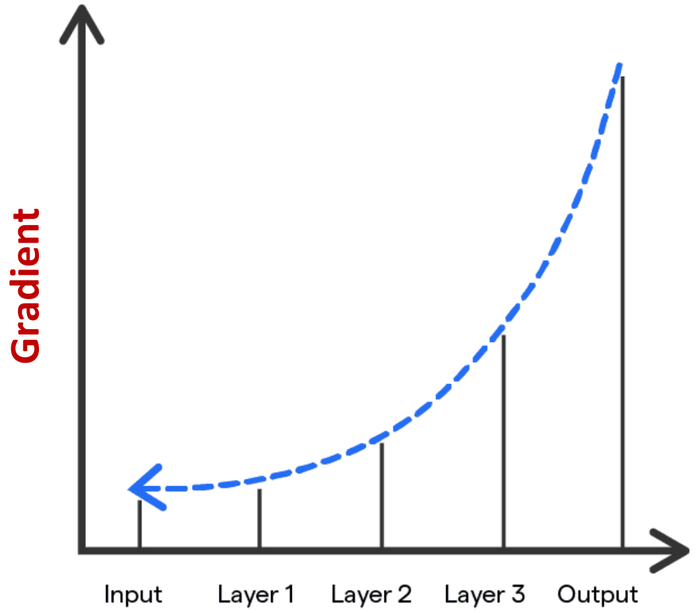
Consideremos la derivada de la función de pérdida $\mathcal{L}$ con respecto a los pesos en la capa l:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \left(\prod_{k=l+1}^{L} W_k^\top \odot \sigma'(Z_{k-1}) \right) \delta_L A_0^\top$$

Para las funciones: $\sigma(z) = \tanh(z)$, $\sigma(z) = \text{Sigmoid}(z)$ ⇒ $|\sigma(z)| < 1$

Entonces: $\prod_{k=l+1}^{L} \sigma'(Z_{k-1}) \to 0$ cuando $L - l$ es muy grande

Vanishing gradient problem



Supongamos una red con L capas, y cada capa l tiene un conjunto de pesos W_l y un conjunto de activaciones A_l. En un forward pass, para una capa l, tenemos:

$$Z_l = W_l A_{l-1} + b_l \qquad A_l = \sigma(Z_l)$$

Consideremos la derivada de la función de pérdida $\mathcal{L}$ con respecto a los pesos en la capa l:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \left(\prod_{k=l+1}^{L} W_k^\top \odot \sigma'(Z_{k-1}) \right) \delta_L A_0^\top$$

Para el caso de: $\sigma(z) = \mathrm{ReLU}(z)$

- Cuando $Z_{k-1} > 0$ entonces $\sigma'(Z_{k-1}) = 1$.
- Cuando $Z_{k-1} \leq 0$ entonces $\sigma'(Z_{k-1}) = 0$.

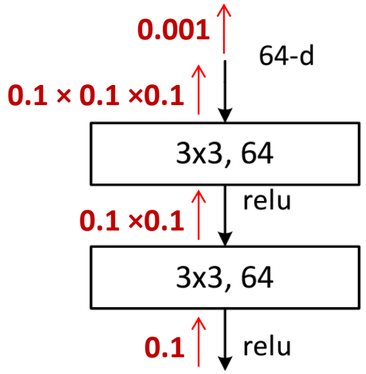
Dead ReLU!

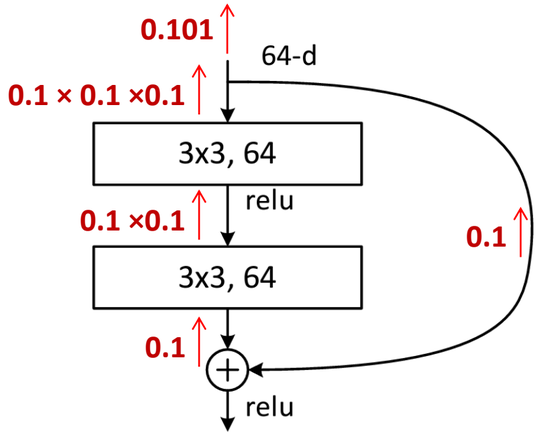
Algunas neuronas no son entrenadas

ResNet

Solución: skip-connection $Z_l = \mathcal{F}(x) + Z_{l-1}$

El **gradiente del error** en la capa $l-1$: $\delta_{l-1} = \frac{\partial \mathcal{L}}{\partial Z_{l-1}} = \frac{\partial \mathcal{L}}{\partial Z_l}\left(\frac{\partial Z_l}{\partial Z_{l-1}}\right) = \frac{\partial \mathcal{L}}{\partial Z_l}\left(\frac{\partial \mathcal{F}}{\partial Z_{l-1}} + 1\right)$



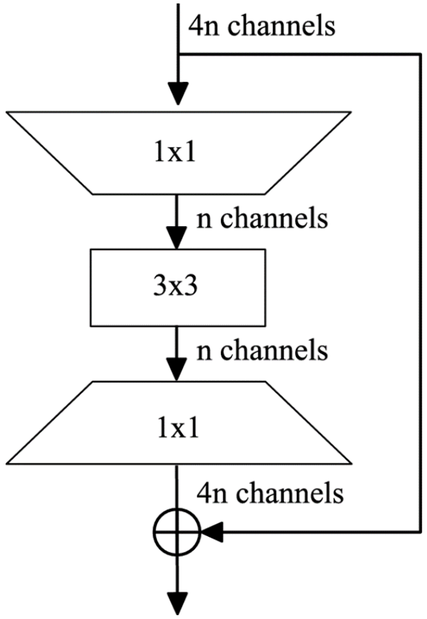


Kaiming He et al. (2015) "Deep Residual Learning for Image Recognition".
Proceedings of the IEEE conference on computer vision and pattern recognition

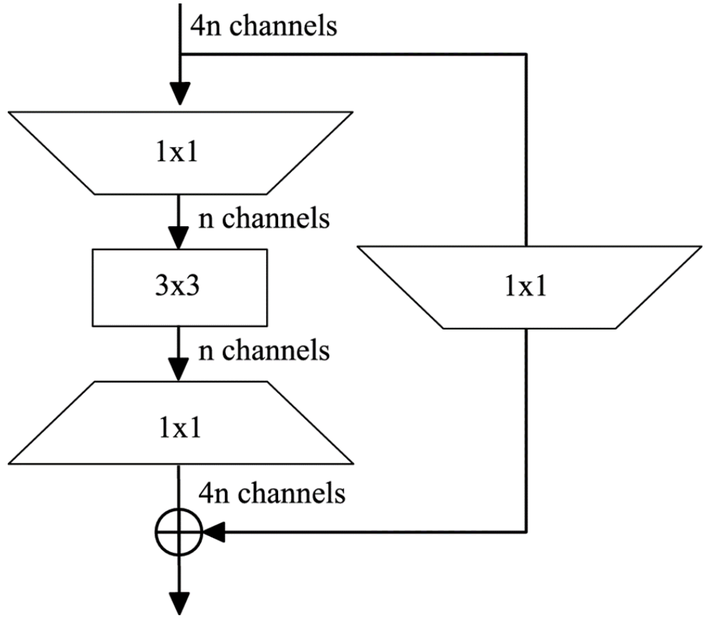
ResNet

Bottleneck architecture

Identity Shortcuts

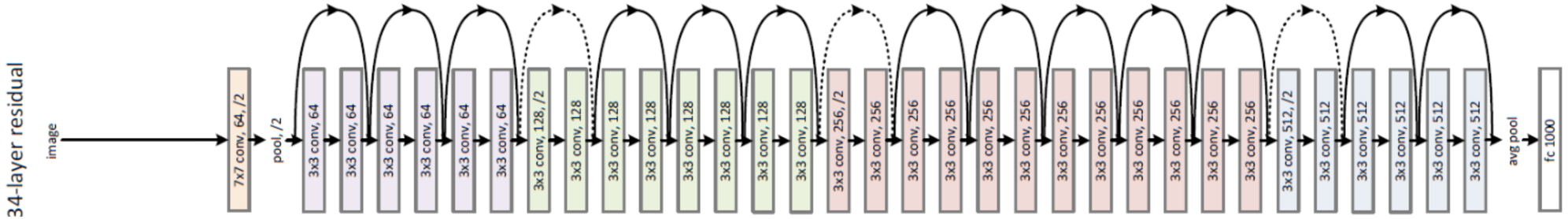


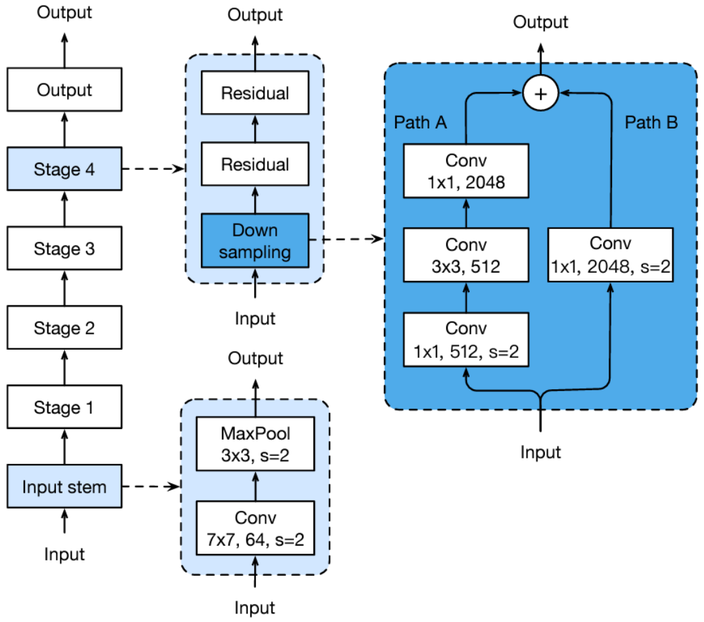
Projection Shortcuts



Kaiming He et al. (2015) "Deep Residual Learning for Image Recognition".
Proceedings of the IEEE conference on computer vision and pattern recognition

ResNet





Kaiming He et al. (2015) "Deep Residual Learning for Image Recognition".
Proceedings of the IEEE conference on computer vision and pattern recognition

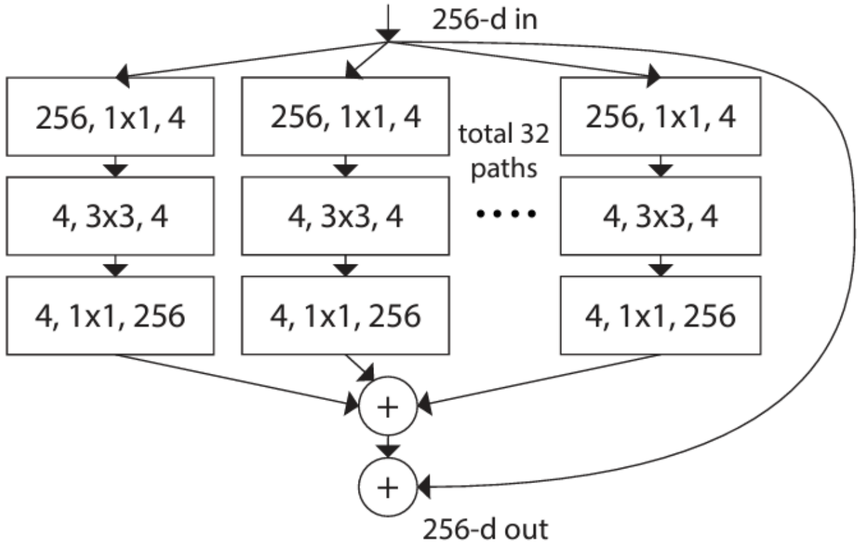
ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

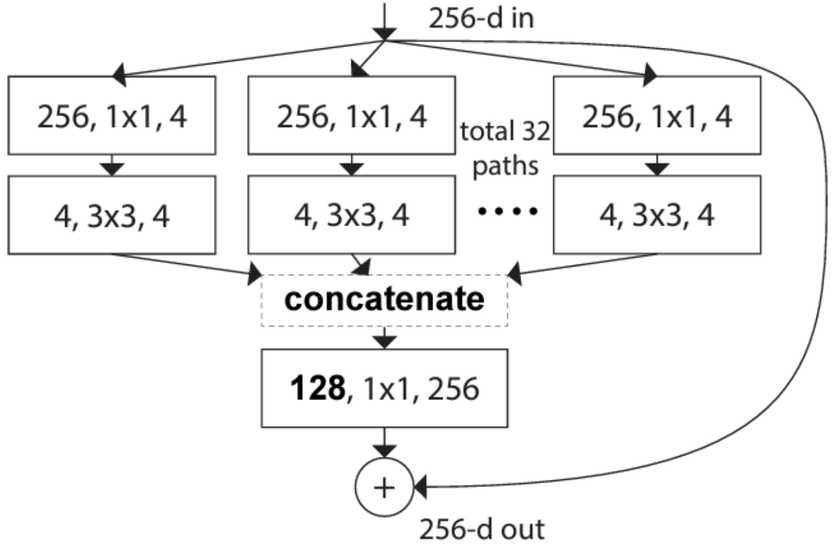
Kaiming He et al. (2015) "Deep Residual Learning for Image Recognition".
Proceedings of the IEEE conference on computer vision and pattern recognition

ResNeXt

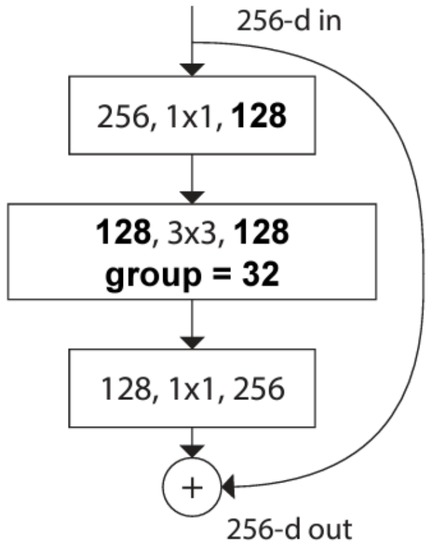
Multiple Paths Configuration



Concatenation and Multiple Paths Configuration



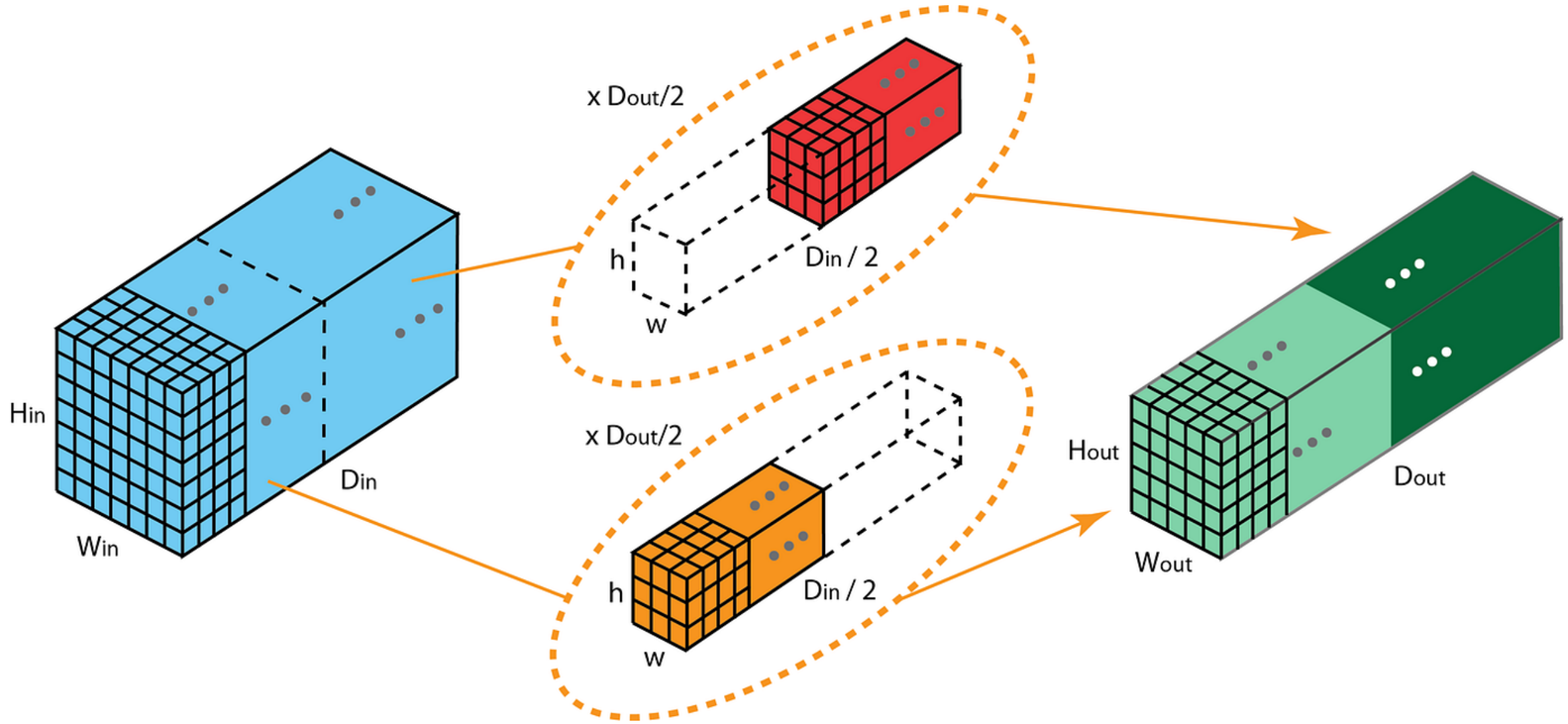
Grouped Convolution Configuration



Saining Xie et al. (2017) "Aggregated Residual Transformations for Deep Neural Networks". Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

ResNeXt

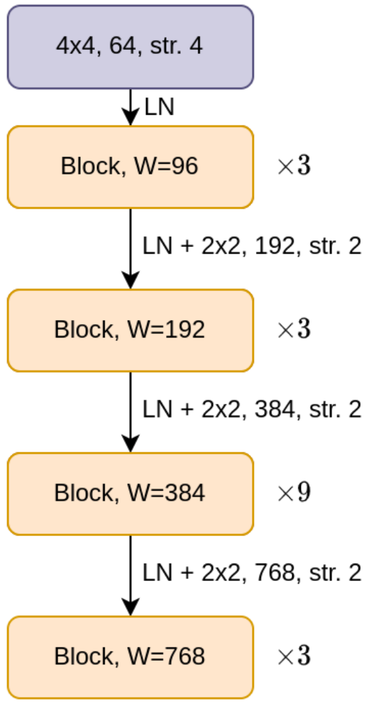
Grouped Convolutions



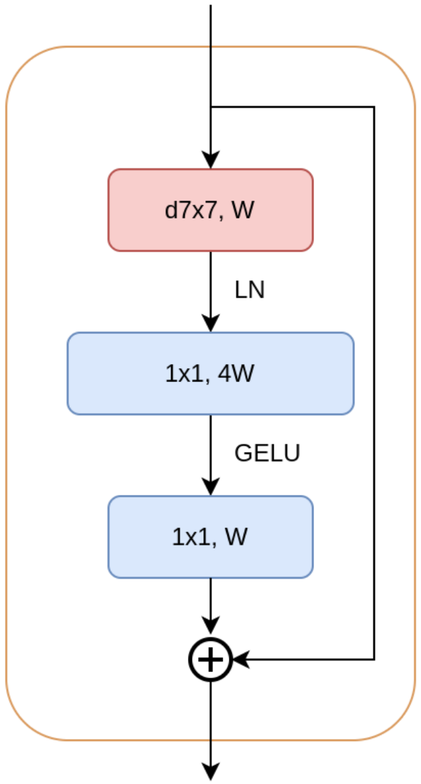
Saining Xie et al. (2017) "Aggregated Residual Transformations for Deep Neural Networks". Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

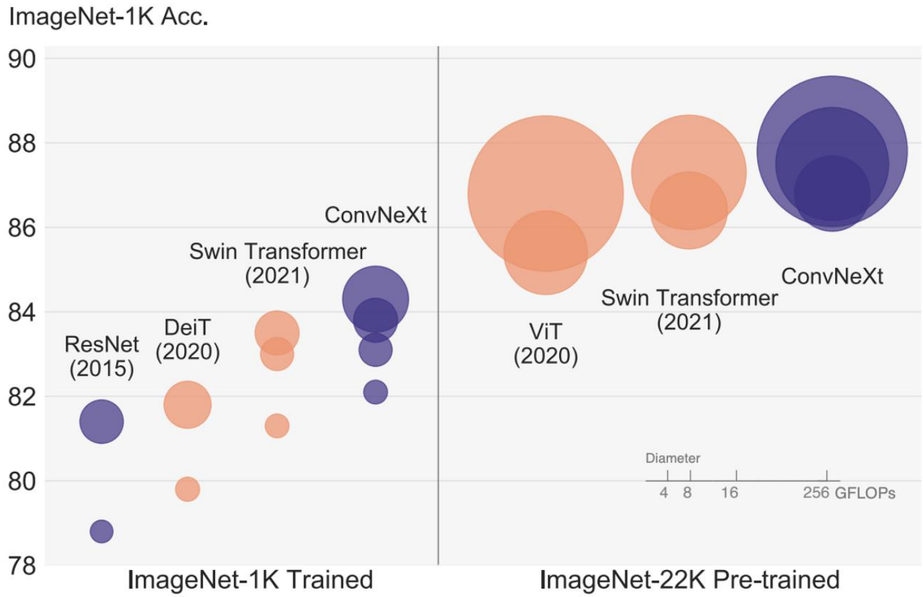
ConvNeXt

ConvNeXt Architecture



Block





Params: 28.6M, FLOPs: 4.5G, IN-1k acc: 82.0%

Zhuang Liu et al. (2022) "A ConvNet for the 2020s".
Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11976-11986).

ConvNeXt

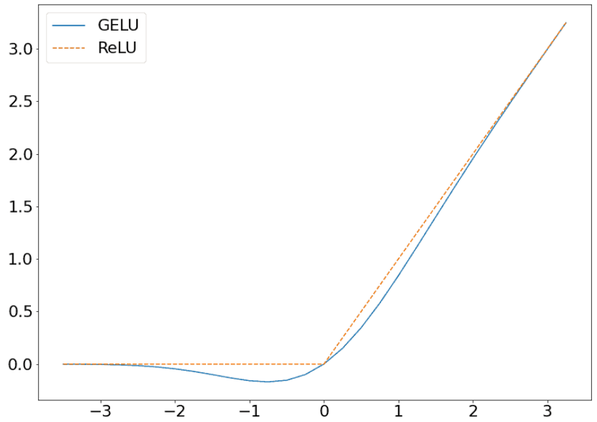
GELU activation

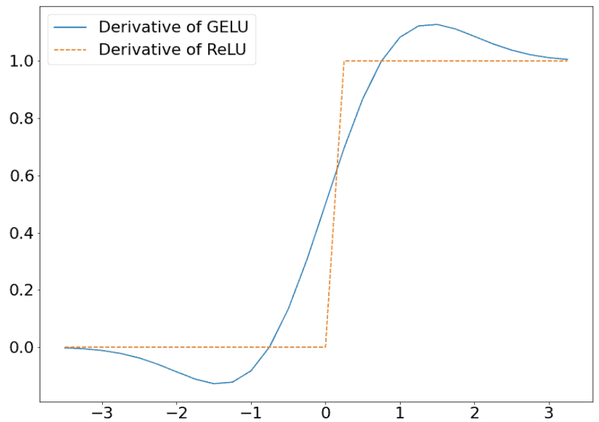
$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

Podemos aproximar la GELU con:

$$\text{GELU}(x) \approx 0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right]\right)$$

$$\text{GELU}(x) \approx x\sigma(1.702x)$$

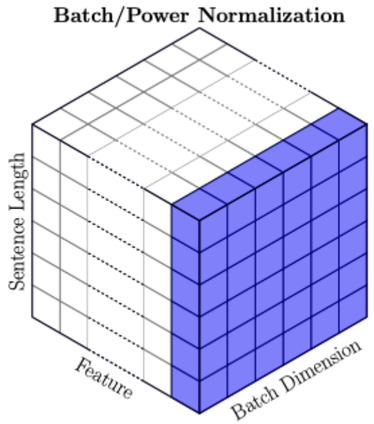


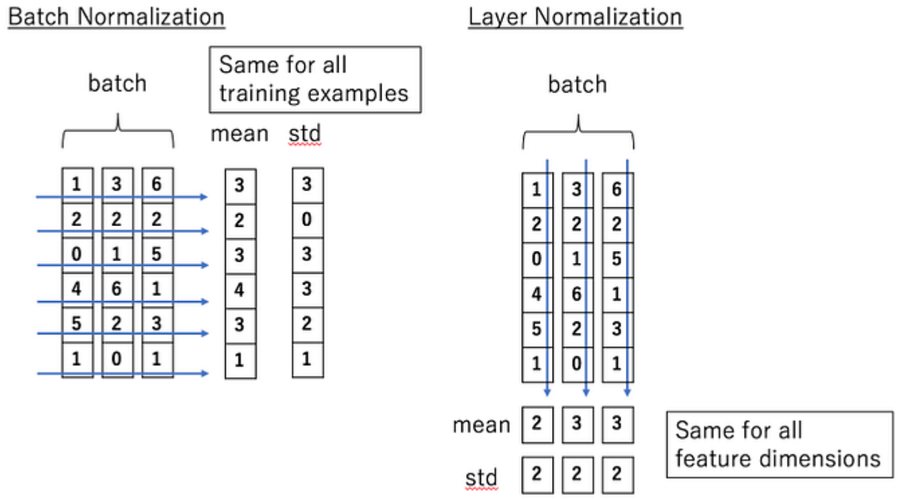


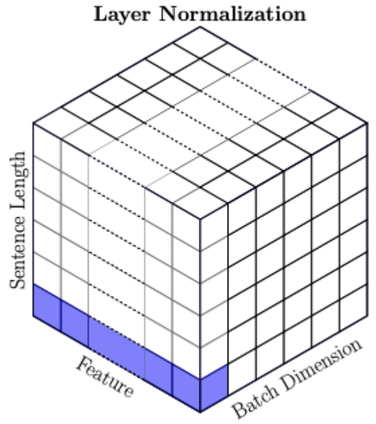
Dan Hendrycks, and Kevin Gimpel (2016) "Gaussian Error Linear Units (GELUs)". arXiv preprint arXiv:1606.08415.

ConvNeXt

Layer Norm







$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\mathrm{Var}[x] + \varepsilon}} \cdot \gamma + \beta$$

Jimmy Lei Ba et al. (2016) "Layer Normalization". stat, 1050, 21.

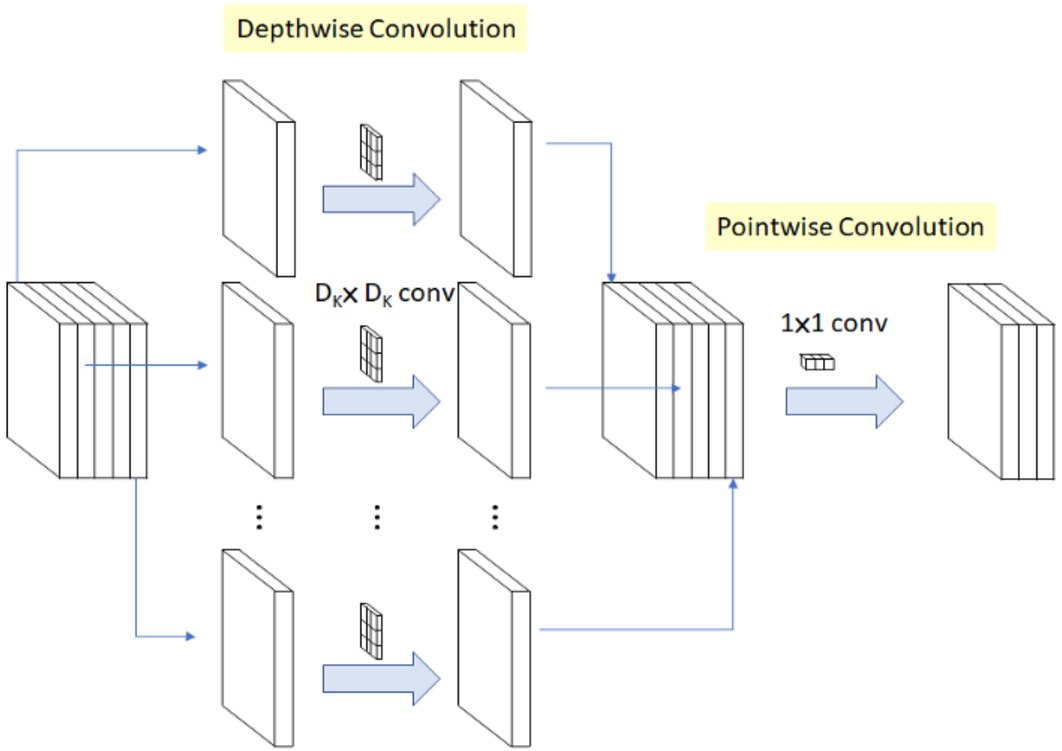
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

7.

MobilNet

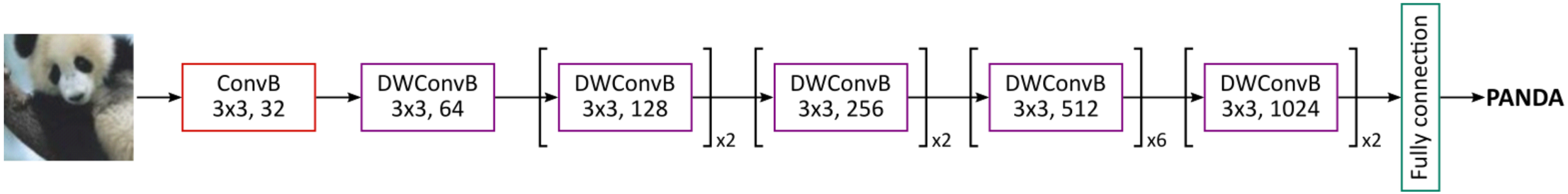
TRANSFORMATEC

MobilNet

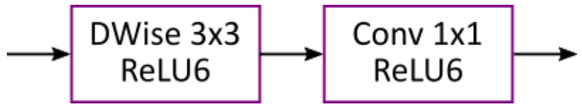


Andrew G. Howard et al. (2017) "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". arXiv preprint arXiv:1704.04861.

MobilNet

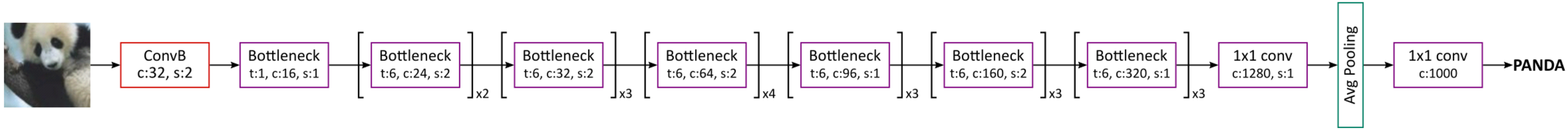


DWConvB

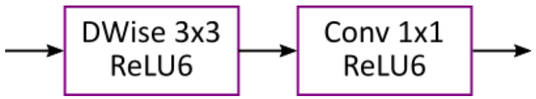


Andrew G. Howard et al. (2017) "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". arXiv preprint arXiv:1704.04861.

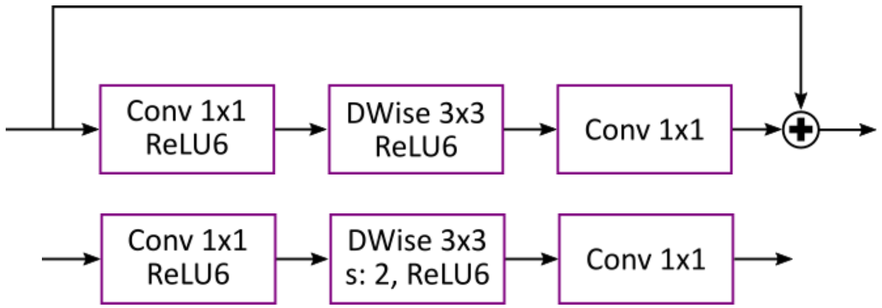
MobilNetV2



MobileNetV1

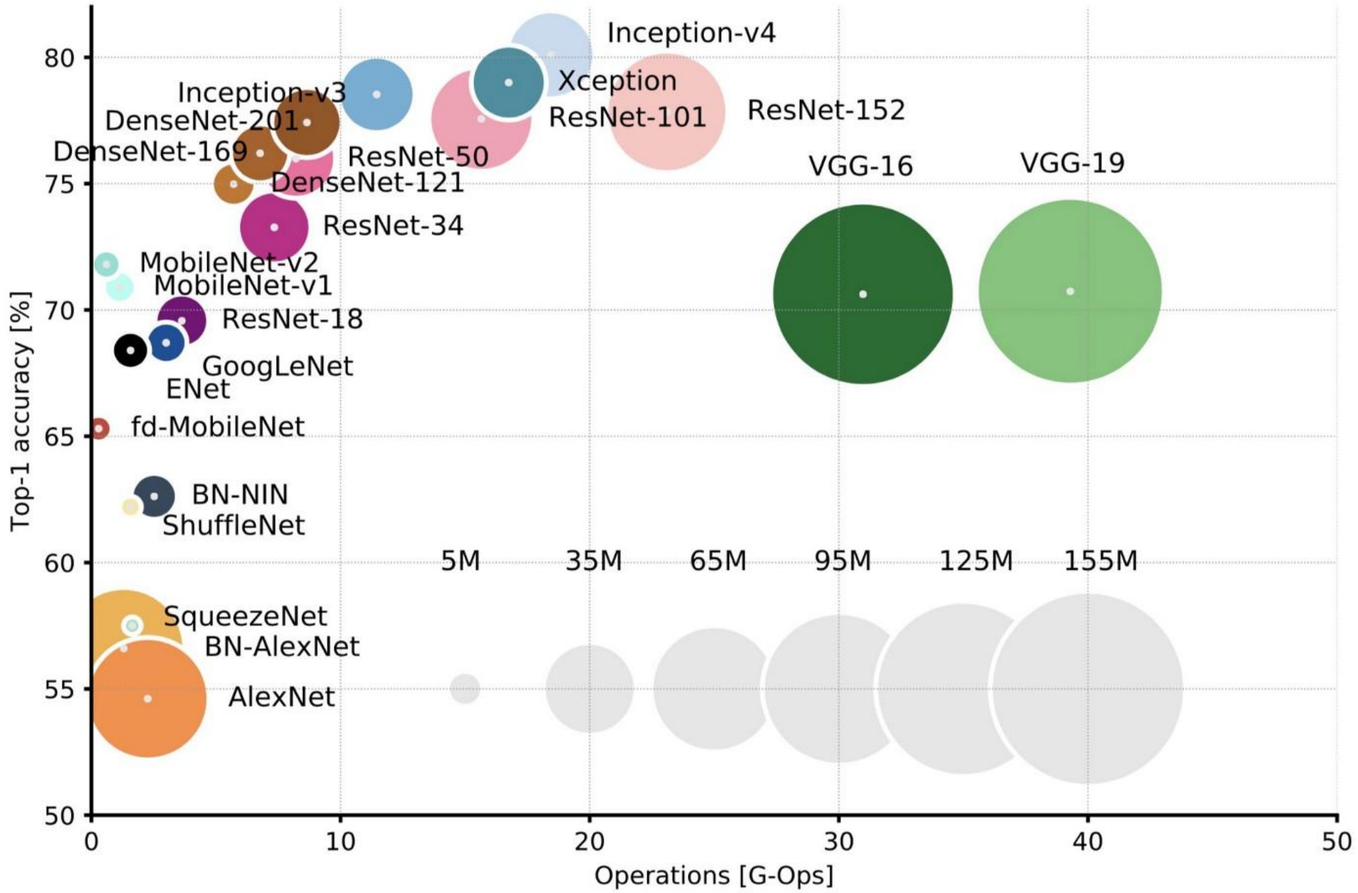


MobileNetV2



Mark Sandler et al. (2018) "MobileNetV2: Inverted Residuals and Linear Bottlenecks".
Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

MobilNetV2



Mark Sandler et al. (2018) "MobileNetV2: Inverted Residuals and Linear Bottlenecks". Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

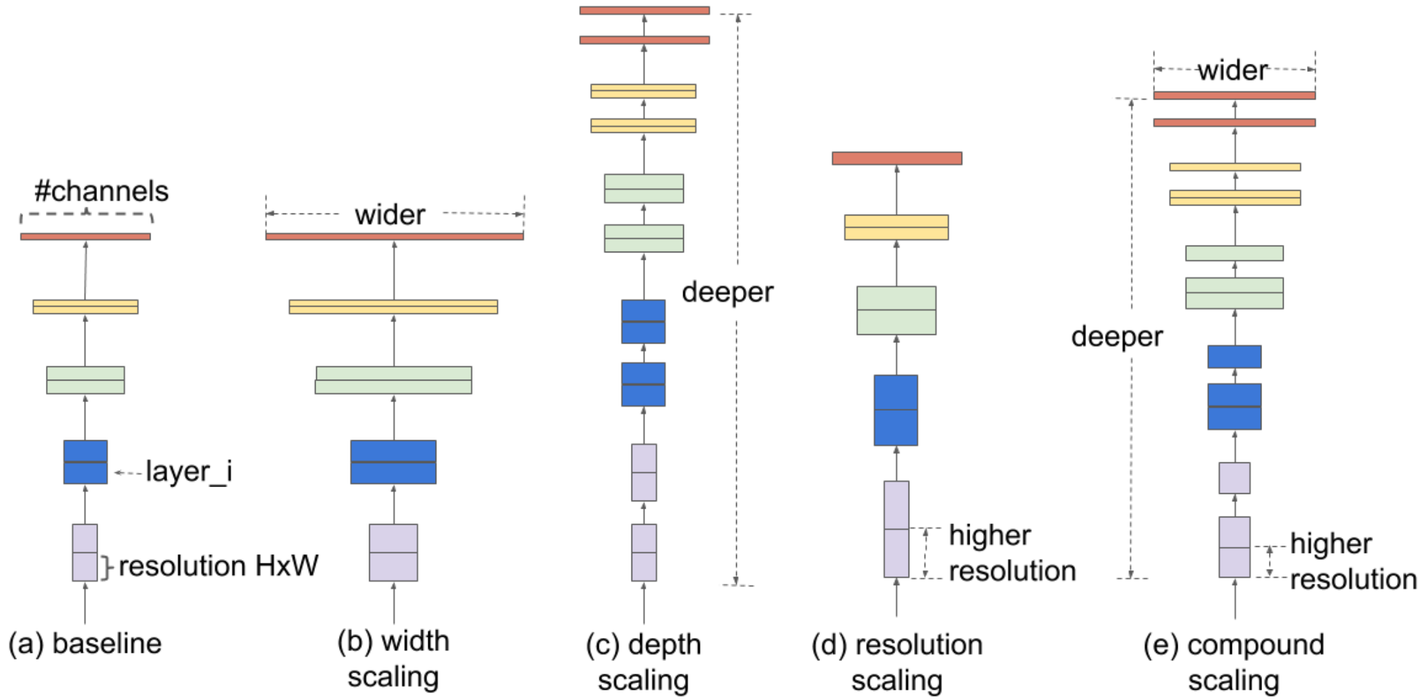
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3.

Efficient*Net*

TRANSFORMATEC

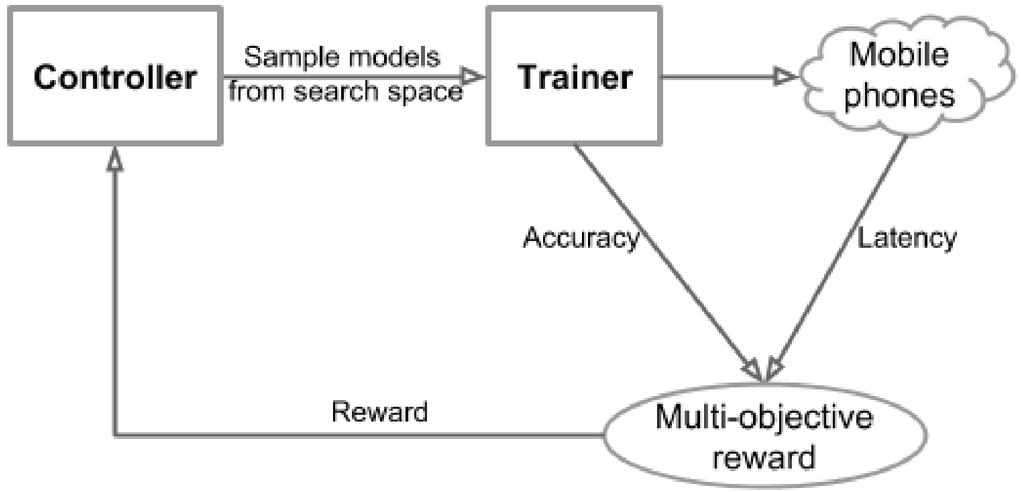
EfficientNet



$$\text{depth: } d = \alpha^\phi$$
$$\text{width: } w = \beta^\phi$$
$$\text{resolution: } r = \gamma^\phi$$
$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

Mingxing Tan et al. (2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". International Conference on Machine Learning. PMLR, 2019.

MnasNet

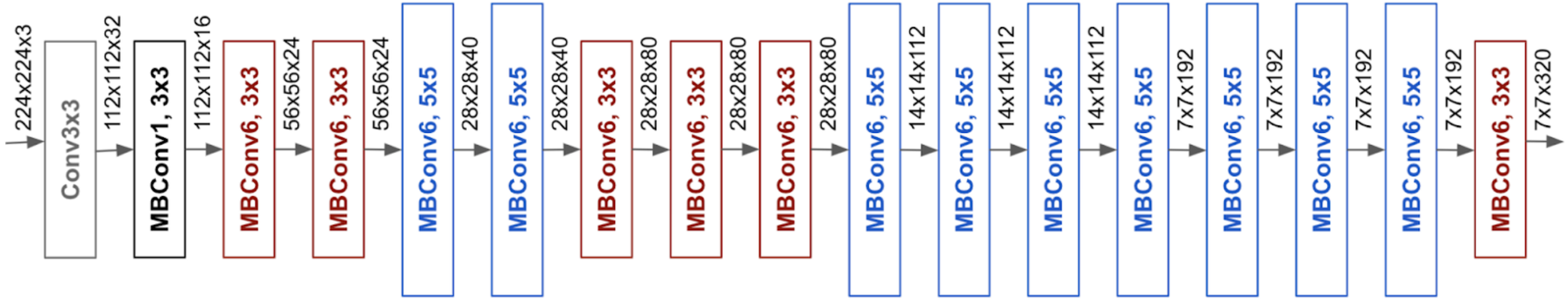


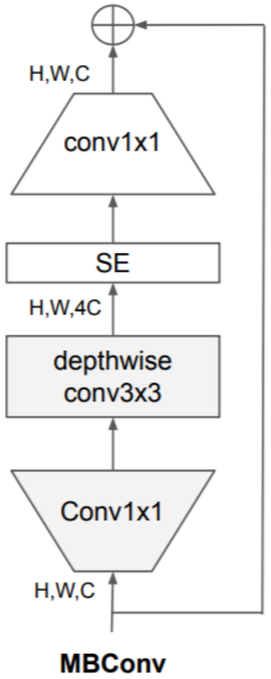
$$\underset{m}{\text{maximize}} \quad ACC(m) \times \left[\frac{LAT(m)}{T}\right]^{w}$$

$$w = \begin{cases} \alpha, & \text{if } LAT(m) \leq T \\ \beta, & \text{otherwise} \end{cases}$$

Mingxing Tan et al. (2019) "MnasNet: Platform-aware neural architecture search for mobile".
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

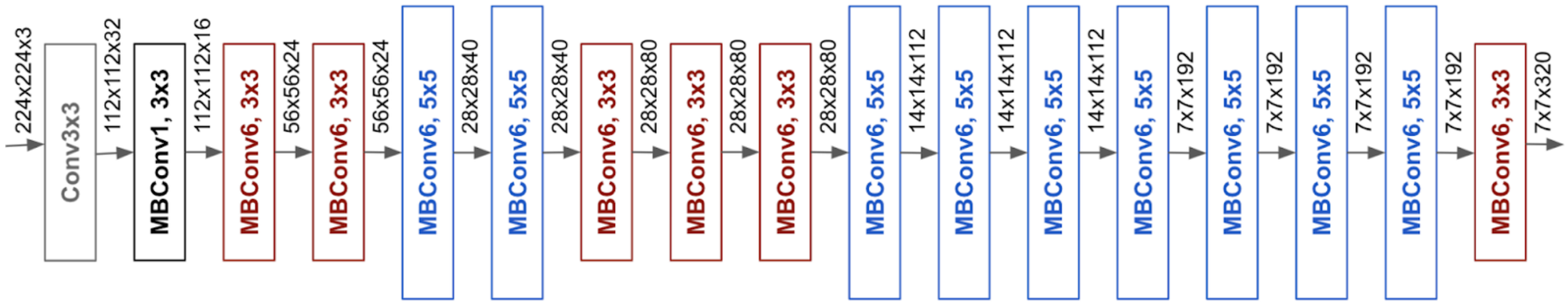
Efficient*Net*

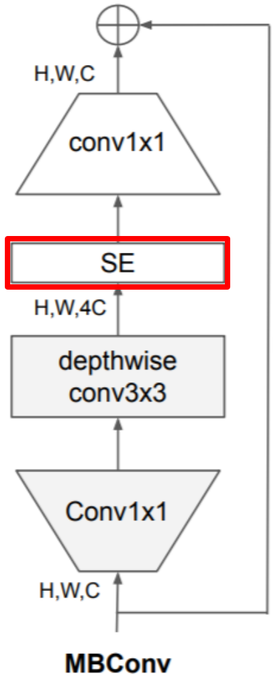




Mingxing Tan et al. (2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". International Conference on Machine Learning. PMLR, 2019.

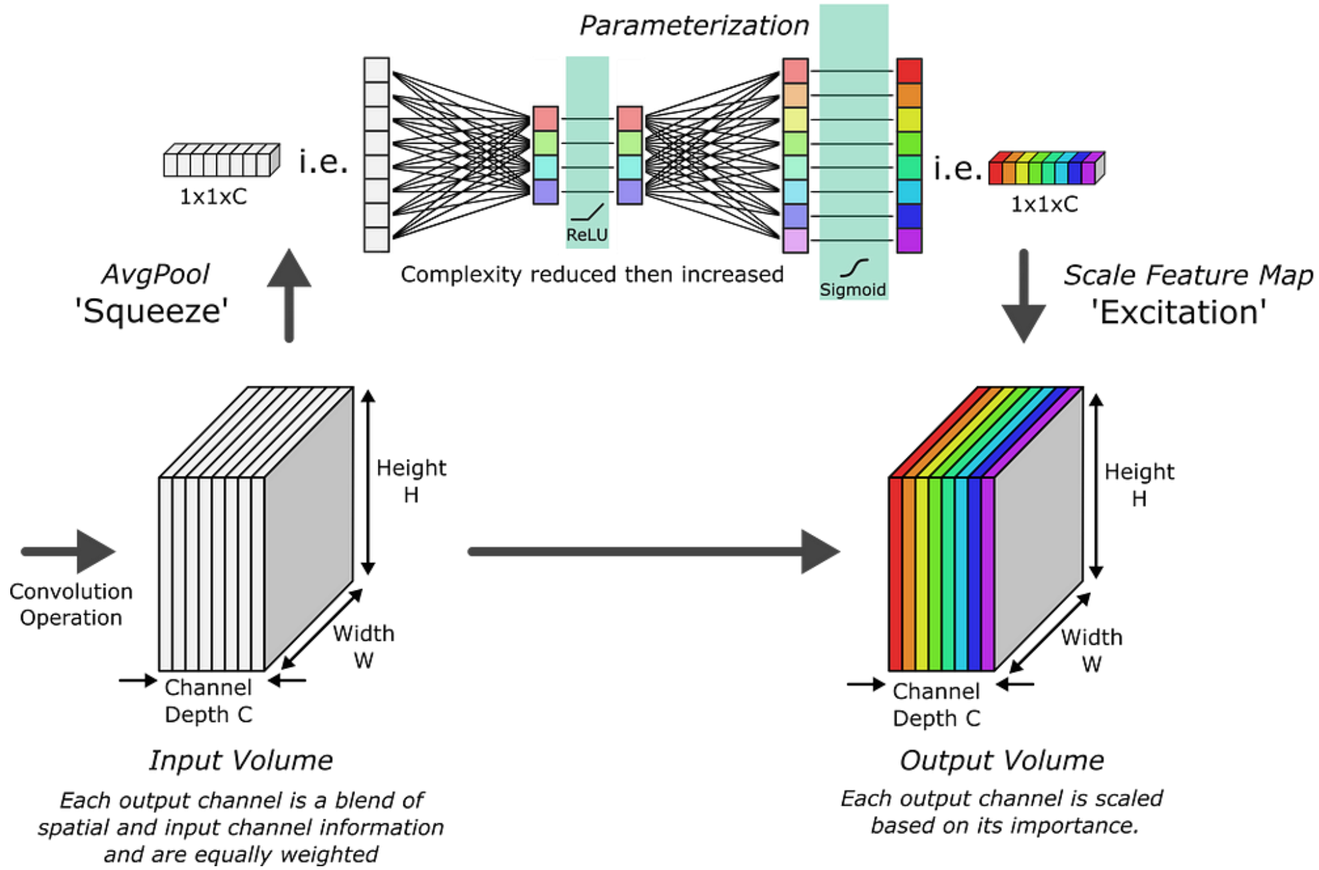
Efficient*Net*





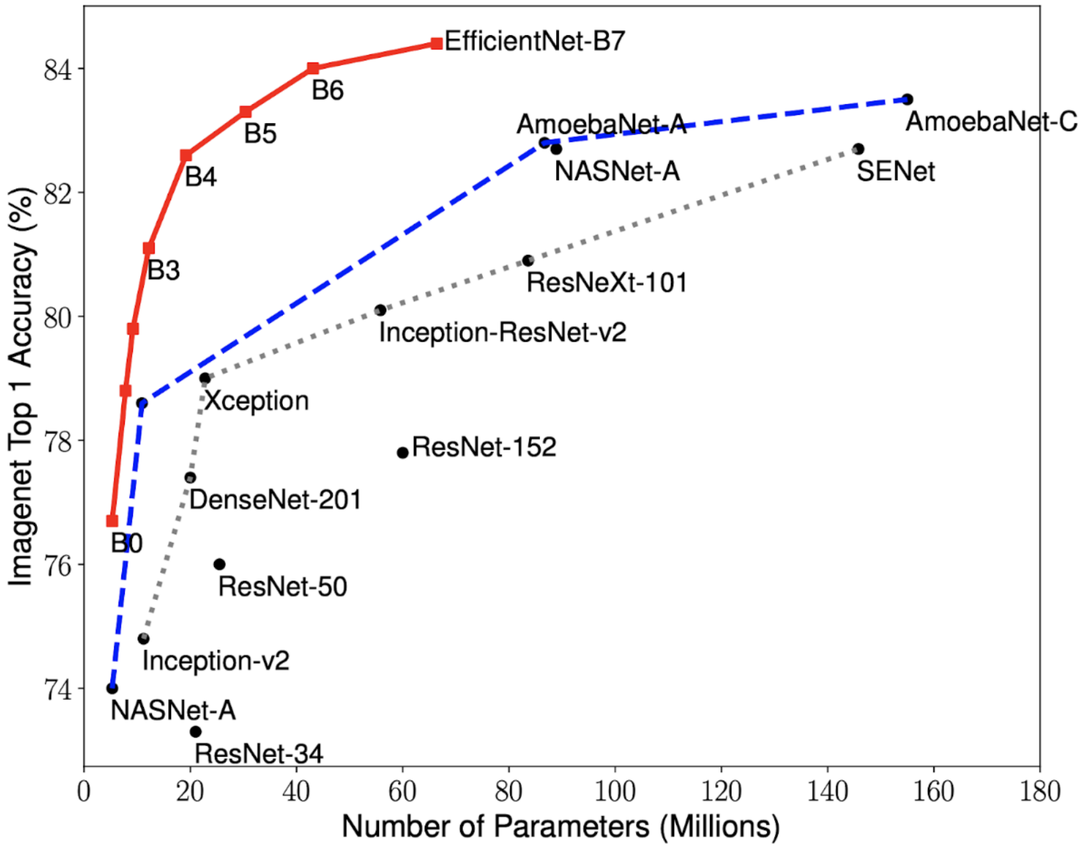
Mingxing Tan et al. (2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". International Conference on Machine Learning. PMLR, 2019.

Squeeze-and-*Excitation*



Jie Hu et al. (2018) "Squeeze-and-Excitation Networks".
Proceedings of the IEEE conference on computer vision and pattern recognition.

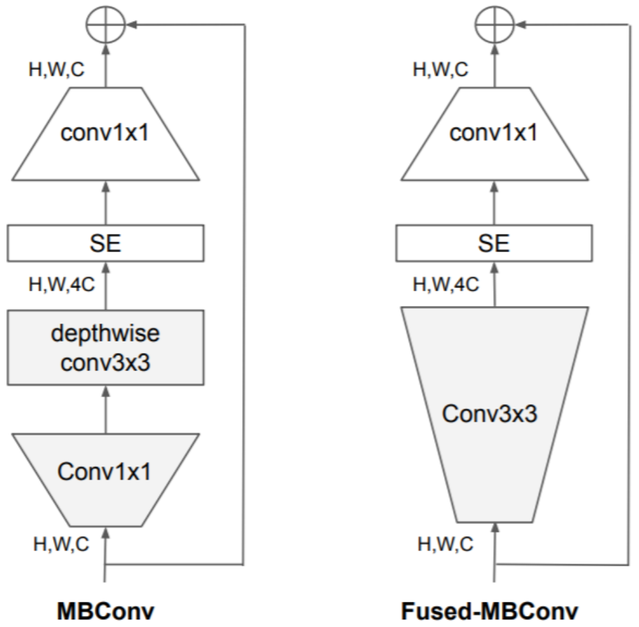
EfficientNet



Mingxing Tan et al. (2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". International Conference on Machine Learning. PMLR, 2019.

EfficientNetV2

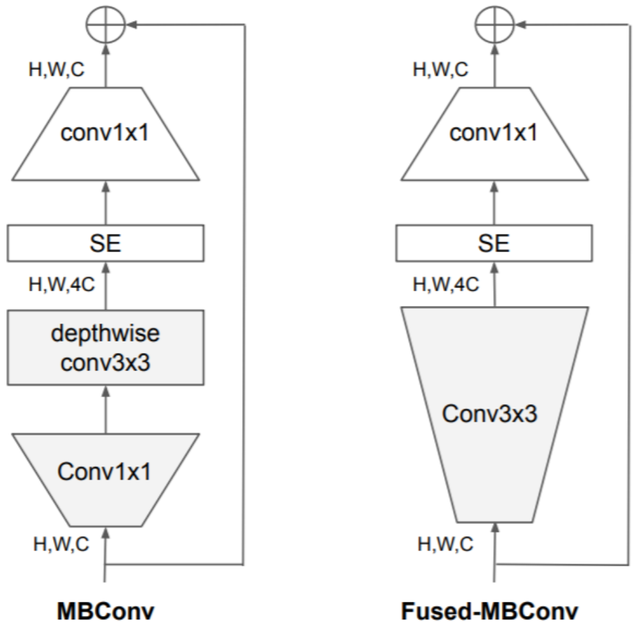
Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	272	15
7	Conv1x1 & Pooling & FC	-	1792	1



Mingxing Tan et al. (2021) "EfficientNetV2: Smaller Models and Faster Training". arXiv preprint arXiv:2104.00298.

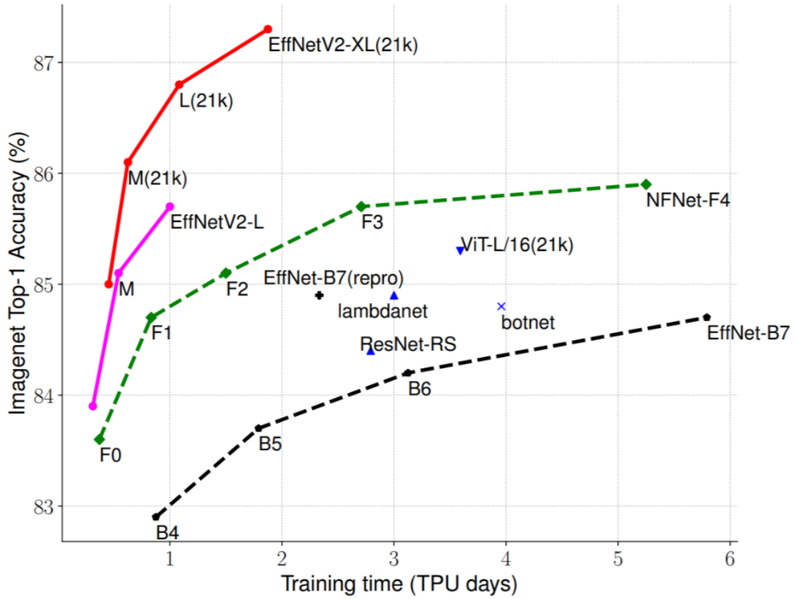
EfficientNetV2

	Params (M)	FLOPs (B)	Top-1 Acc.	TPU imgs/sec/core	V100 imgs/sec/gpu
No fused	19.3	4.5	82.8%	262	155
Fused stage1-3	20.0	7.5	83.1%	362	216
Fused stage1-5	43.4	21.3	83.1%	327	223
Fused stage1-7	132.0	34.4	81.7%	254	206



Mingxing Tan et al. (2021) "EfficientNetV2: Smaller Models and Faster Training". arXiv preprint arXiv:2104.00298.

EfficientNetV2



(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Mingxing Tan et al. (2021) "EfficientNetV2: Smaller Models and Faster Training". arXiv preprint arXiv:2104.00298.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

GRACIAS

Victor Flores Benites

TRANSFORMATEC