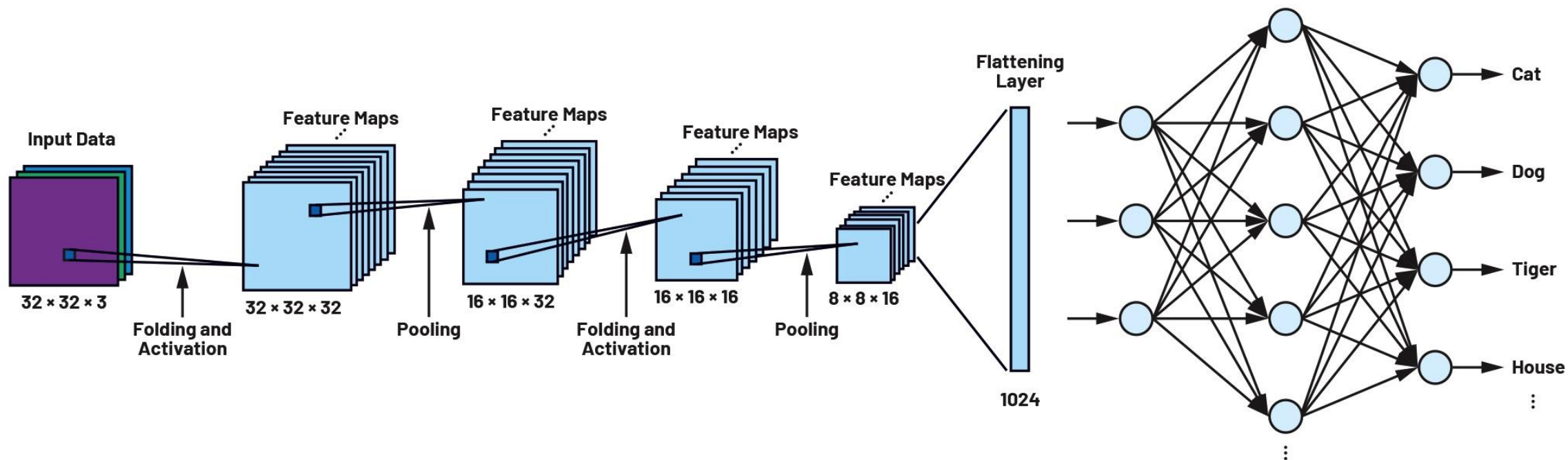
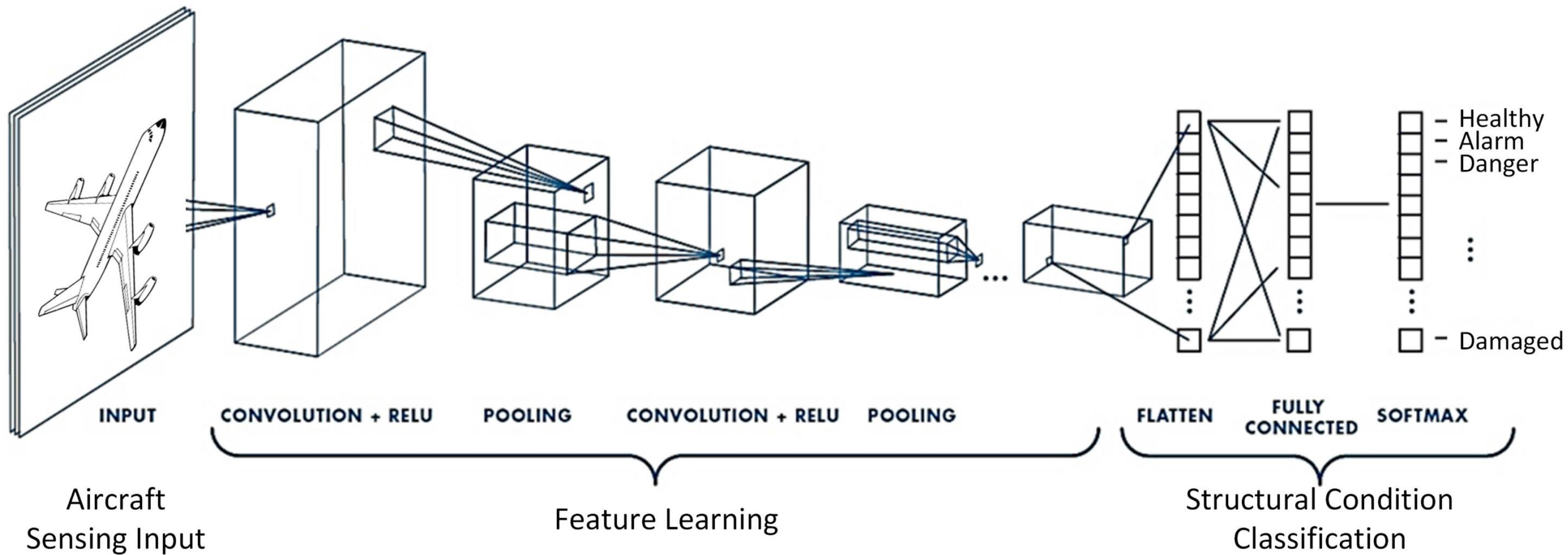
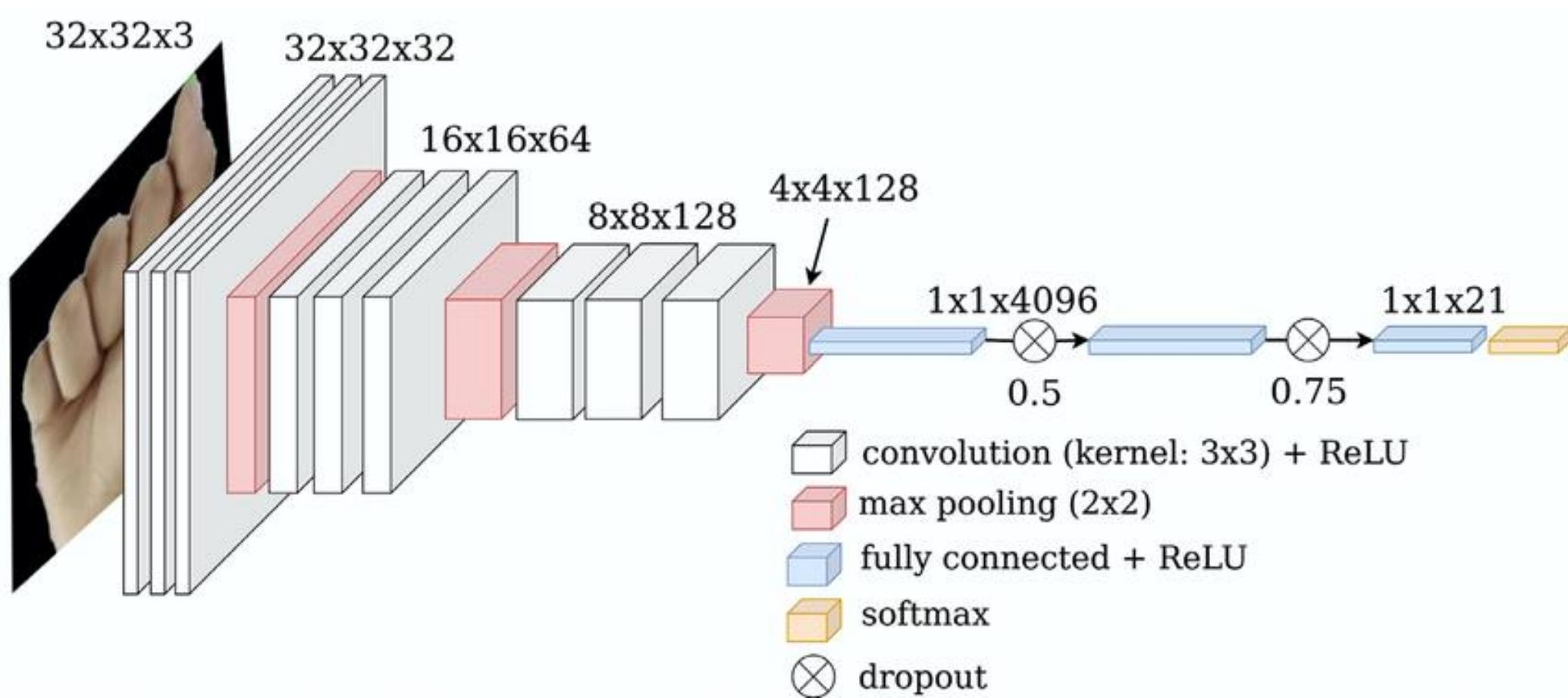


Convolutional Neural Networks







¿qué es lo que pasa con las imágenes en el DL?

- Topología de píxeles
- Invariancia de la traslación
- Problemas de iluminación y contraste
- Conocimiento del sistema visual humano
- Los píxeles tienden a tener valores similares
- Bordes y formas

Recordemos qué es un kernel

Un kernel es una cuadrícula de pesos superpuestos a la imagen, centrados en un píxel.

La salida es sobre el píxel centrado

Se utiliza para técnicas tradicionales de procesamiento de imágenes: Desenfocar, enfocar, detección de bordes, relieve, etc.

Vertical Line Detector

-1	1	-1
-1	1	-1
-1	1	-1

Horizontal Line Detector

-1	-1	-1
1	1	1
-1	-1	-1

Corner Detector

-1	-1	-1
-1	1	1
-1	1	1

Operación de convolución

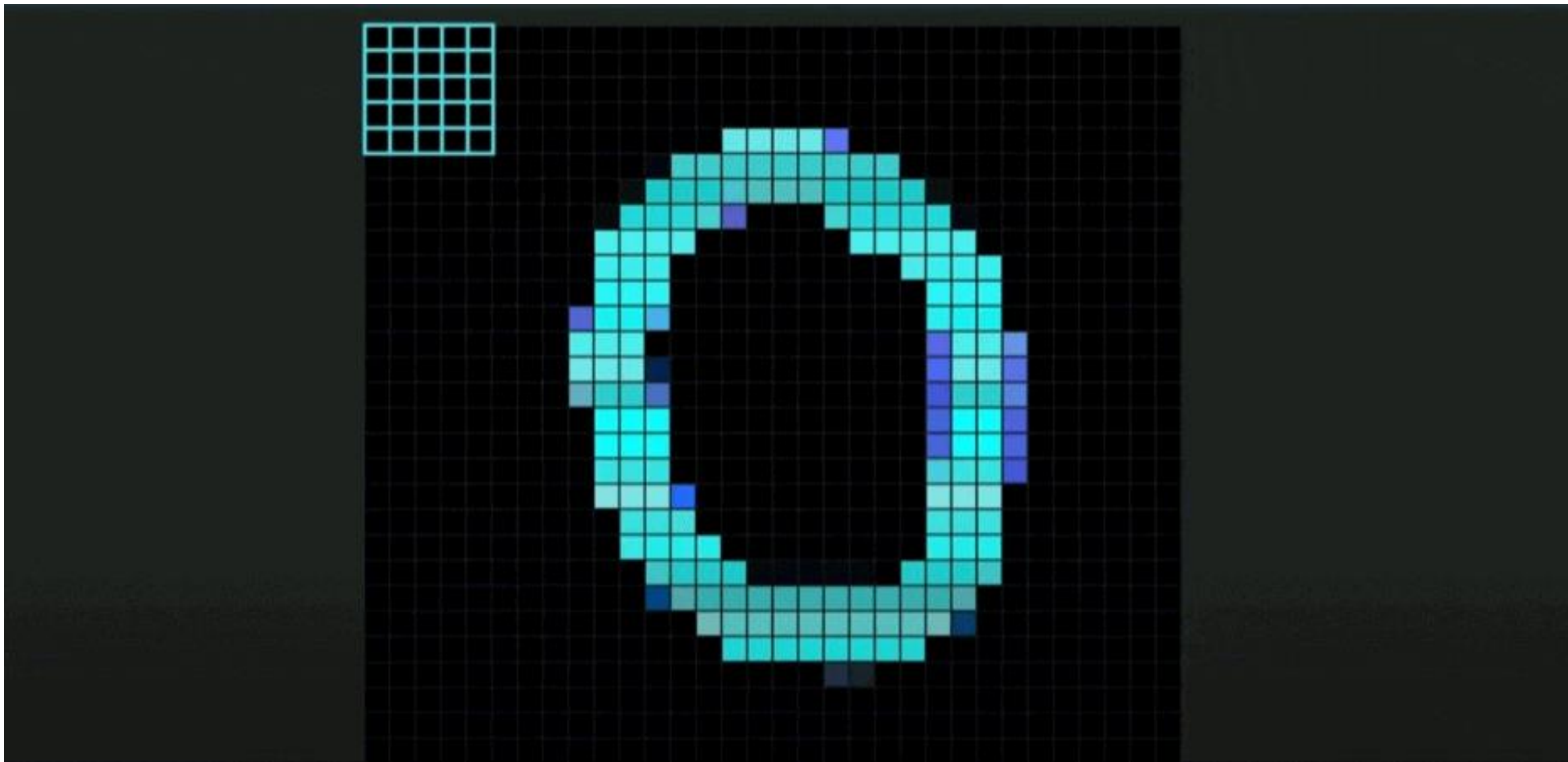
$$Y(i, j) = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} X(i + m, j + n) \cdot K(m, n)$$

Cuándo hay C canales

$$Y(i, j) = \sum_{c=0}^{C-1} \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} X(c, i + m, j + n) \cdot K(c, m, n)$$

Primeras ideas de una red convolucional

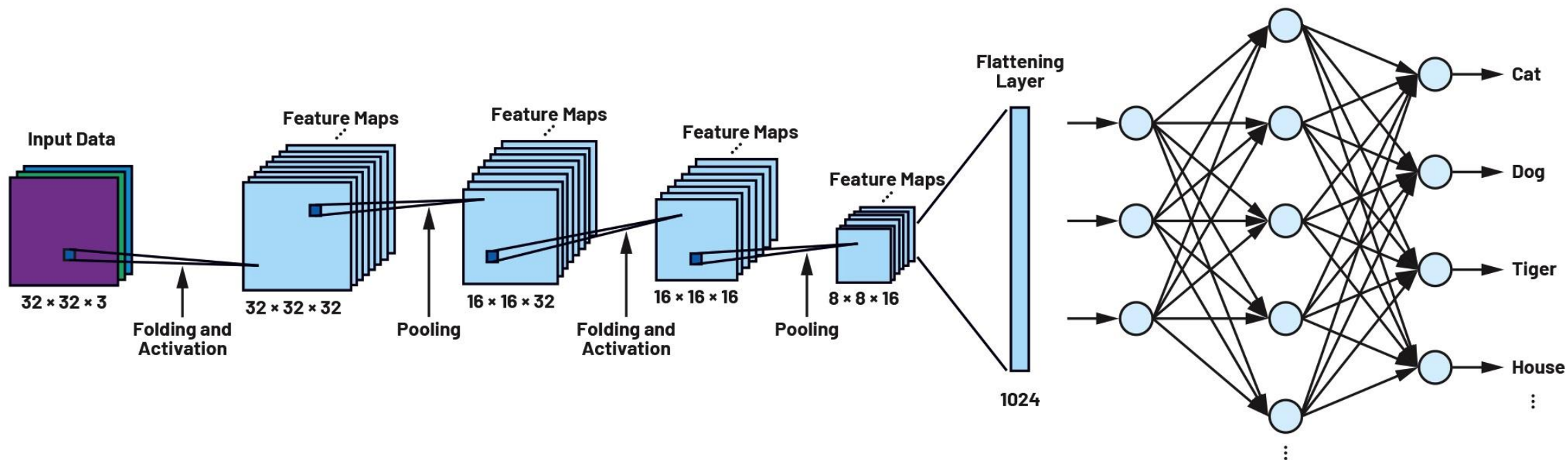
- La red neuronal aprende cuáles kernels son más útiles
- Reduce el número de parámetros, en gran medida porque utiliza el mismo kernel a través de toda la imagen.

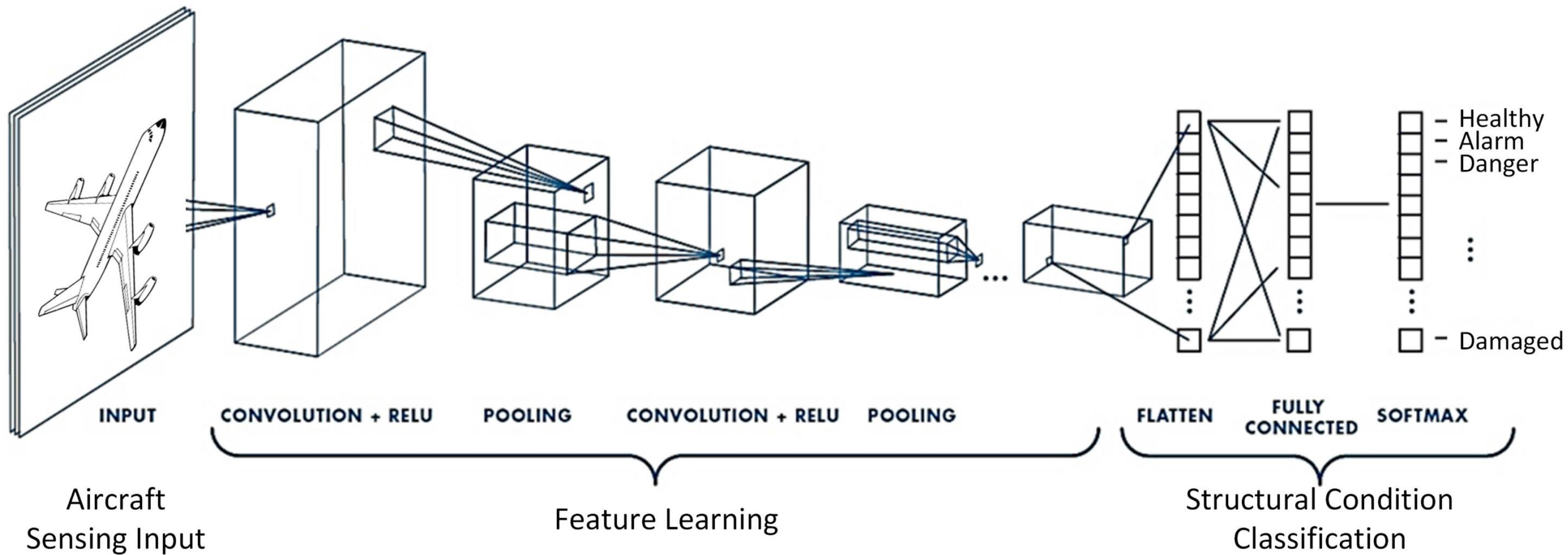


Primera fase: Feature Map/ Convolution + ReLU

Convolución de kernel tridimensional

- Utilización de convoluciones en un filtro
- Tres kernels 3x3 apilados para cada dimensión
- Incremento de las multiplicaciones a 27 para una salida





Una capa convolucional toma como entrada un **tensor de datos** (por ejemplo, una imagen con altura, ancho y canales) y produce como salida un **tensor de activaciones**, llamado **mapa de características** (*feature map*).

$$z_{ij}^{(k)} = (w^{(k)} * x)_{ij} + b^{(k)}$$

donde:

- x es la entrada (ej., la imagen o el mapa de activaciones anterior),
- $w^{(k)}$ es el **k-ésimo filtro (kernel)** de la capa,
- $*$ es la operación de convolución,
- $z_{ij}^{(k)}$ es el resultado en la posición (i, j) del **k-ésimo feature map**,
- $b^{(k)}$ es el bias del filtro k ,
- después se aplica una función de activación $a_{ij}^{(k)} = \sigma(z_{ij}^{(k)})$.

Aquí empiezan a entrar una serie de hiperpárametros para el bloque de feature map

1. **Tamaño del kernel** (por ej., 3x3, 5x5)
2. **Stride** (cuánto se mueve el filtro)
3. **Padding** (si se agregan bordes para mantener tamaño)
4. **Número de filtros** (define profundidad de salida)

1. Tamaño del Kernel

- Número de píxeles que un kernel "ve" simultáneamente.
- Se suelen usar números impares para que haya un píxel central.
- No es necesario que el kernel sea cuadrado.

Padding

- Al usar kernels directamente, se producirá un efecto de borde.
- Los píxeles cercanos al borde no se usarán como píxeles centrales, ya que no hay suficientes píxeles circundantes.
- El relleno añade píxeles adicionales alrededor del marco, de modo que los píxeles de la imagen original se convierten en píxeles centrales a medida que el kernel se desplaza por la imagen.
- Los píxeles añadidos suelen tener valor cero (relleno de ceros).

Sin Padding

1	2	0	3	1
1	0	0	2	2
2	1	2	1	1
0	0	1	0	0
1	2	1	1	1

input

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

kernel

-2		

output

Con Padding

0	0	0	0	0	0	0
0	1	2	0	3	1	0
0	1	0	0	2	2	0
0	2	1	2	1	1	0
0	0	0	1	0	0	0
0	1	2	1	1	1	0
0	0	0	0	0	0	0

input

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

kernel

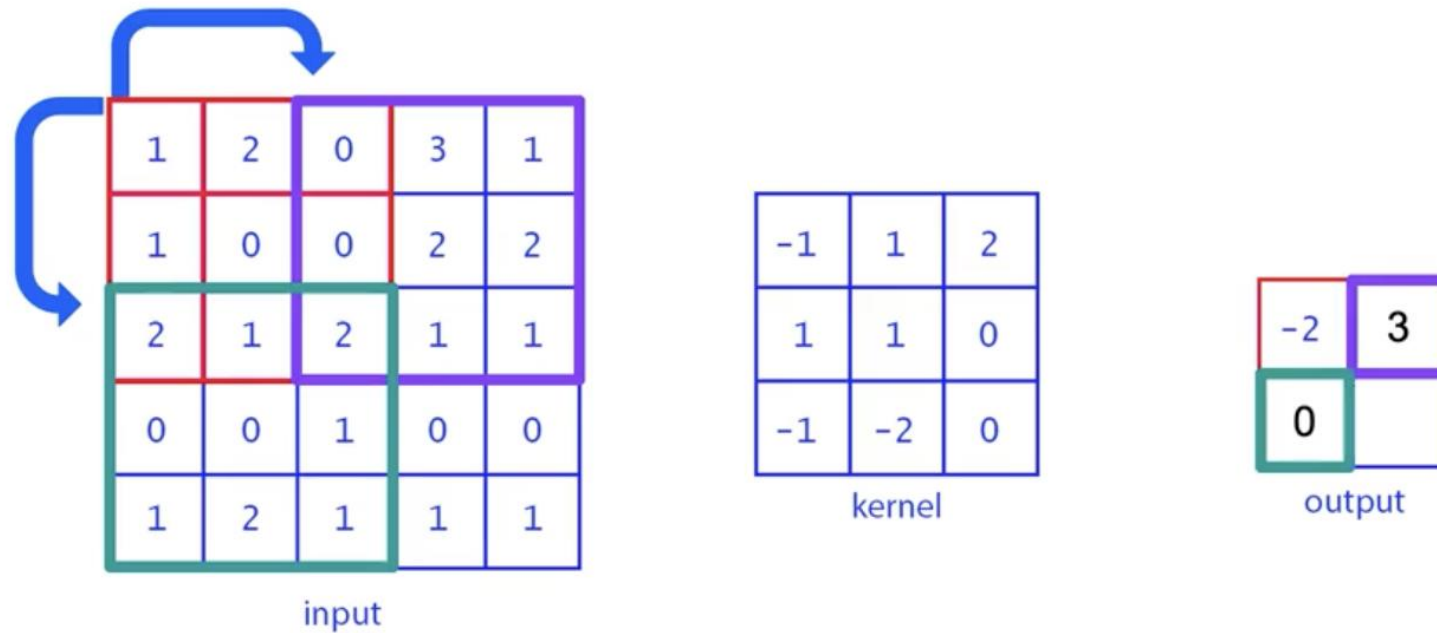
-1				

output

Stride/Zancada o paso

- - El tamaño del paso a medida que el núcleo se mueve por la imagen.
- - Puede ser diferente para pasos verticales y horizontales (pero suele tener el mismo valor).
- - Cuando la zancada es mayor que 1, reduce la dimensión de salida.

2 pasos sin padding



2 pasos con padding

0	0	0	0	0	0	0
0	1	2	0	3	1	0
0	1	0	0	2	2	0
0	2	1	2	1	1	0
0	0	0	1	0	0	0
0	1	2	1	1	1	0
0	0	0	0	0	0	0

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

kernel

-2	2	
3		

output

Profundidad

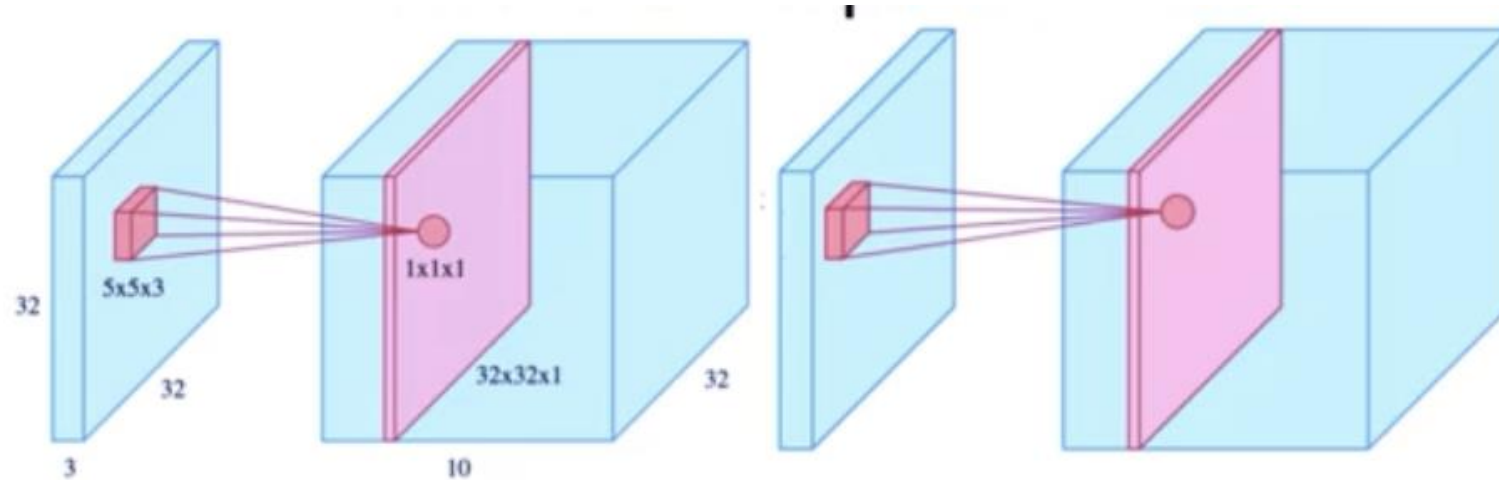
El número de canales se denomina "profundidad".

Por lo tanto, el kernel tendrá una "profundidad" igual al número de canales de entrada.

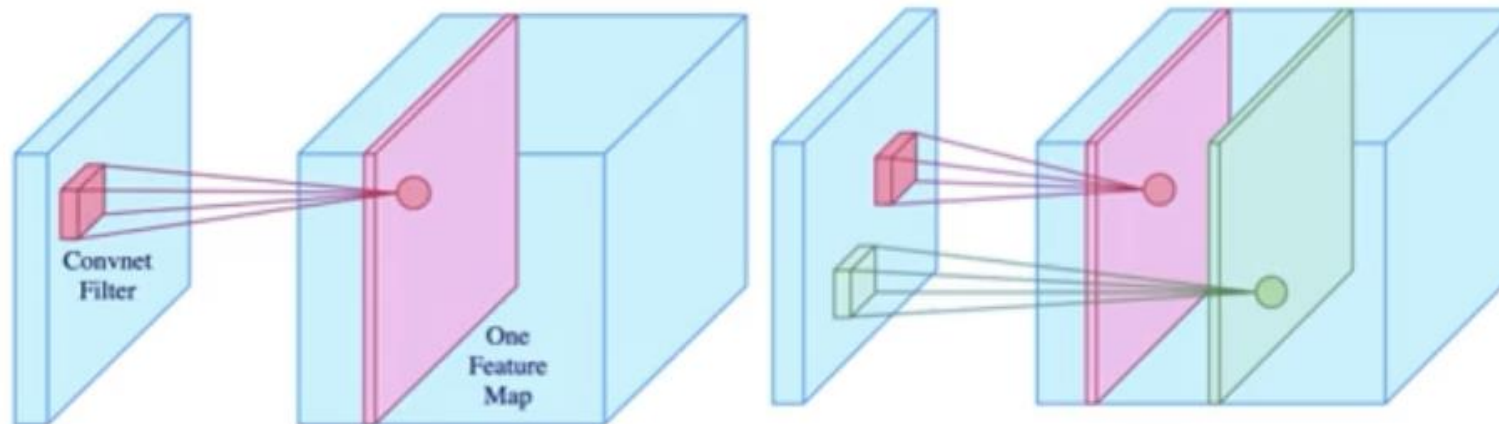
Un kernel de 5 x 5 en una imagen RGB.

- Habrá $5 \times 5 \times 3 = 75$ pesos

Capa Convolucional



Note : 10 filters of size 5x5x3 are deployed



Cuántas neuronas hay ¿Cuántas neuronas hay en una capa convolucional?

En una red convolucional, la cantidad de "neuronas" no está determinada por el tamaño del kernel, sino por el tamaño del *feature map* de salida.

Cada posición donde aplicas el filtro (o kernel) es una **neurona**.

Si tienes:

- Imagen de entrada: tamaño $H \times W$,
- Filtro de tamaño $F \times F$,
- Stride S ,
- Padding P ,

$$\text{Output height} = \frac{H - F + 2P}{S} + 1$$

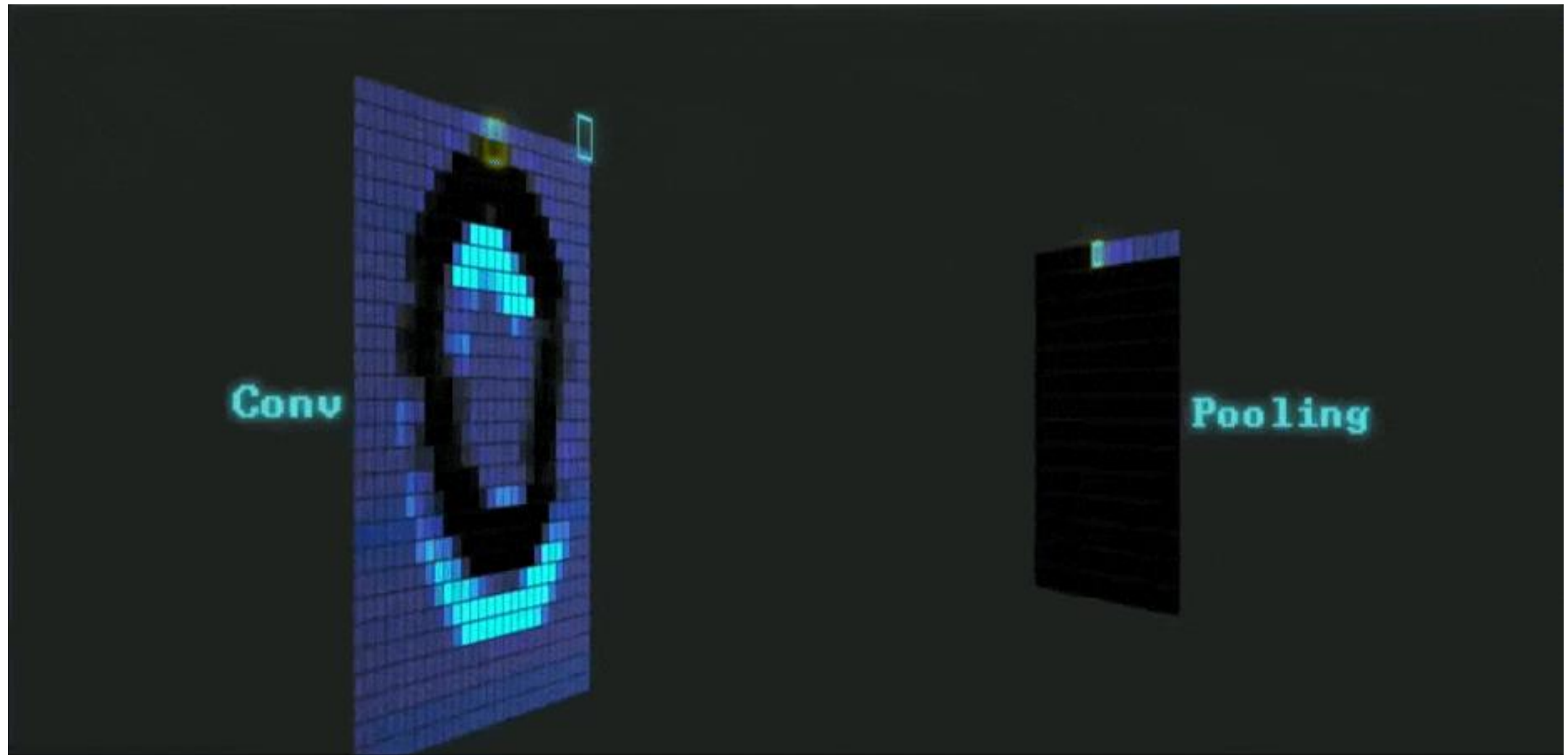
$$\text{Output width} = \frac{W - F + 2P}{S} + 1$$

$$\text{Neurons} = \text{Output height} \times \text{Output width} \times \text{número de filtros}$$

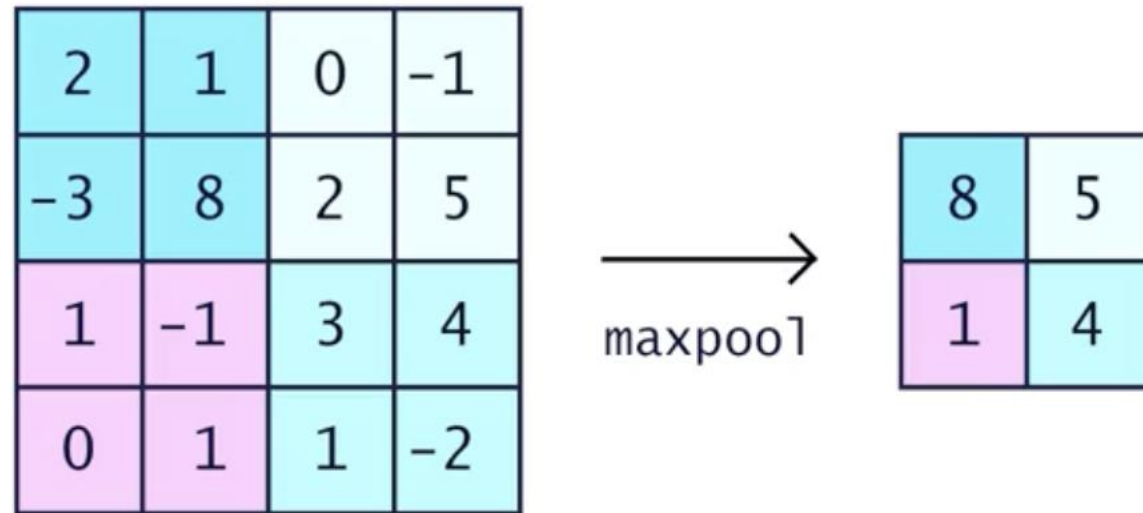
Segunda fase: Pooling

Idea: Reducir el tamaño de la imagen asignando un fragmento de píxeles a un único valor.

No tiene parámetros, aunque existen diferentes tipos de operaciones de agrupación.



Max pool



Average Pool

