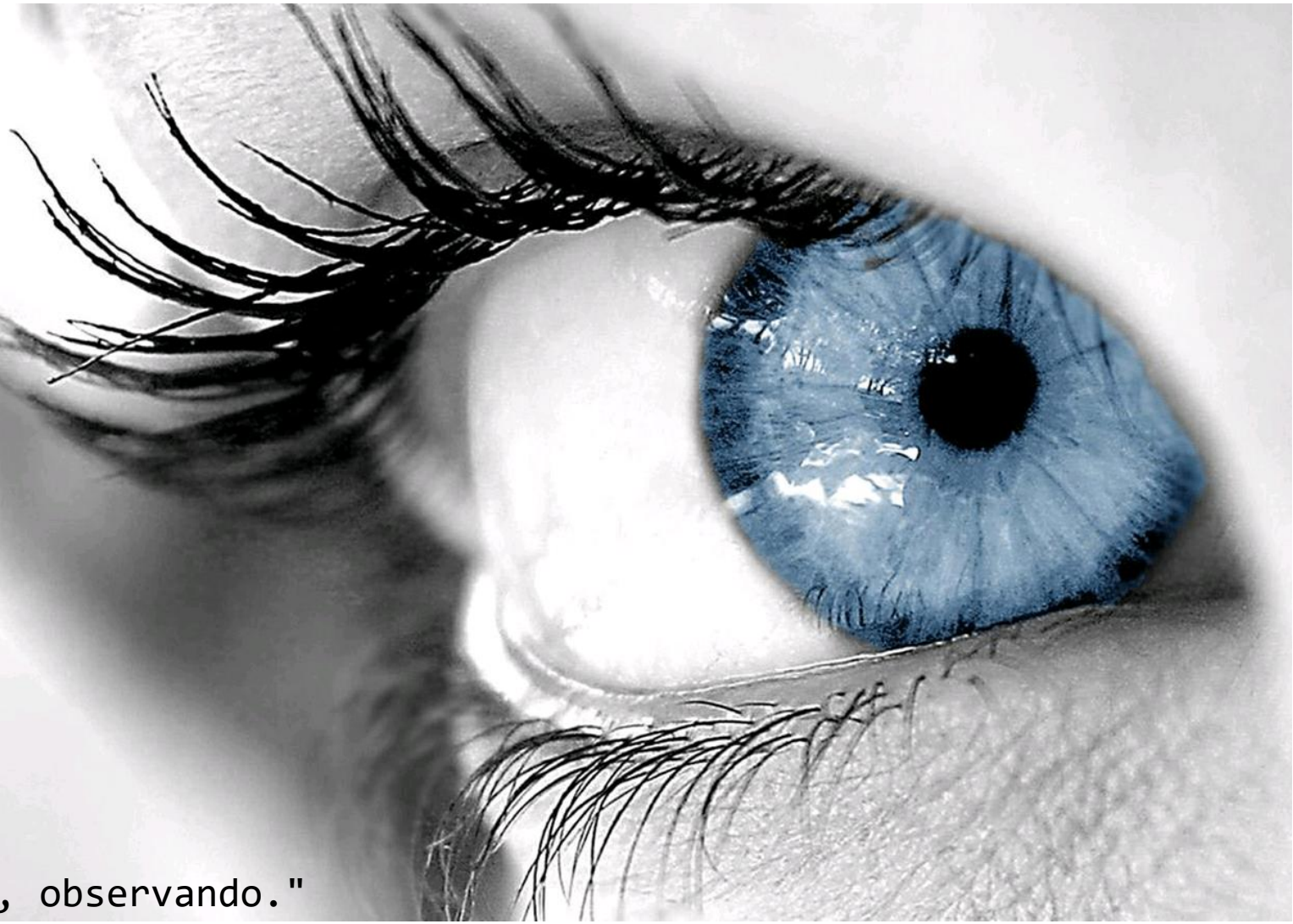
UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

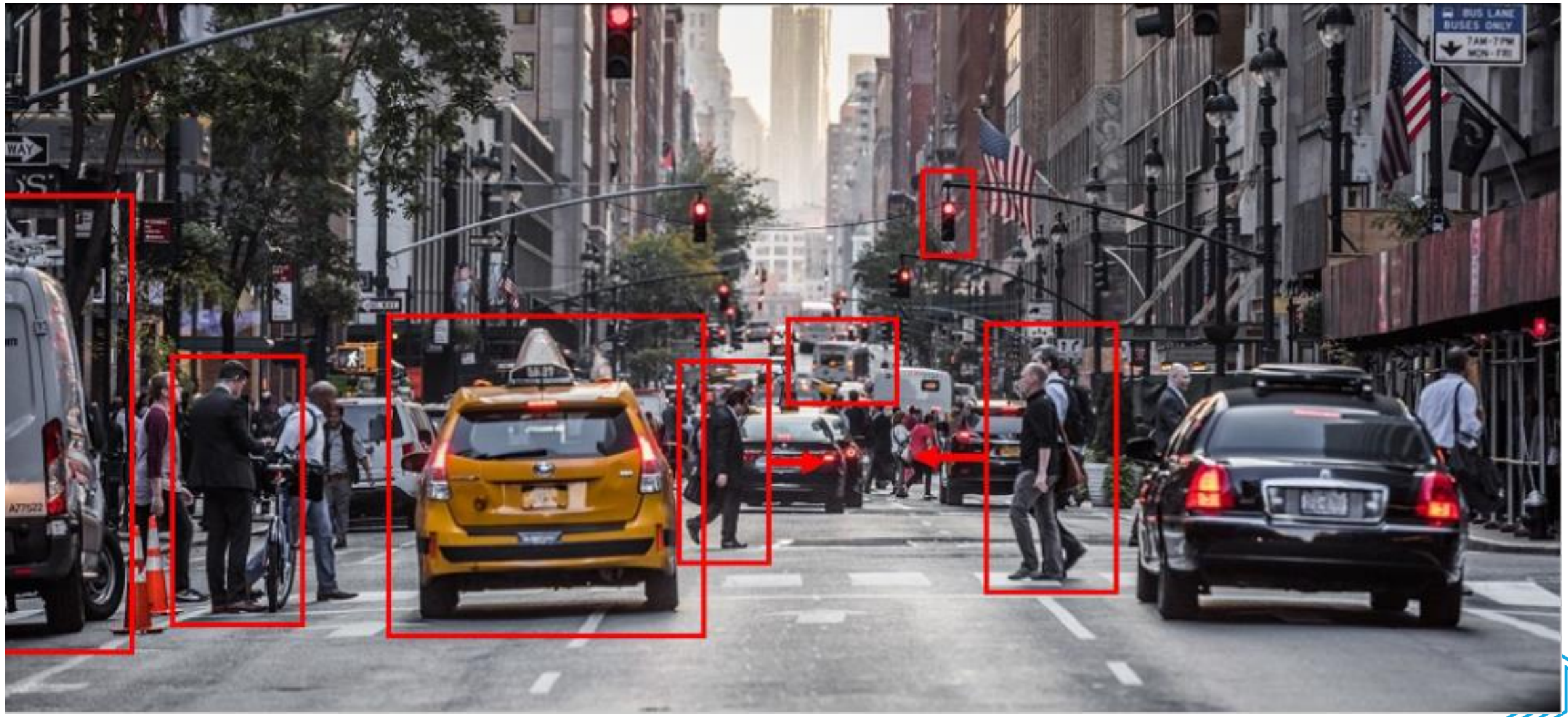
Escuela de
Ciencias Aplicadas
e Ingeniería

Visión por Computador Profunda



"Conocer qué hay y dónde está, observando."

Descubrir a partir de imágenes qué está presente en el mundo,
dónde están las cosas, qué acciones están ocurriendo,
y predecir y anticipar eventos en el mundo.



El auge e impacto de la visión por computador

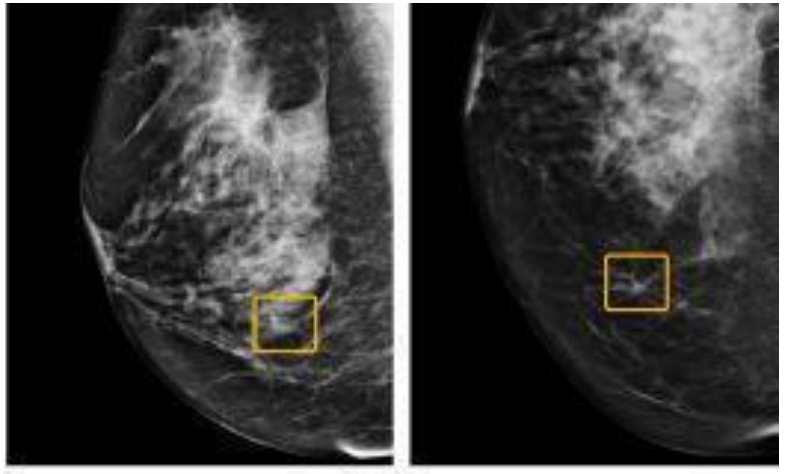
UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Robótica

Accesibilidad

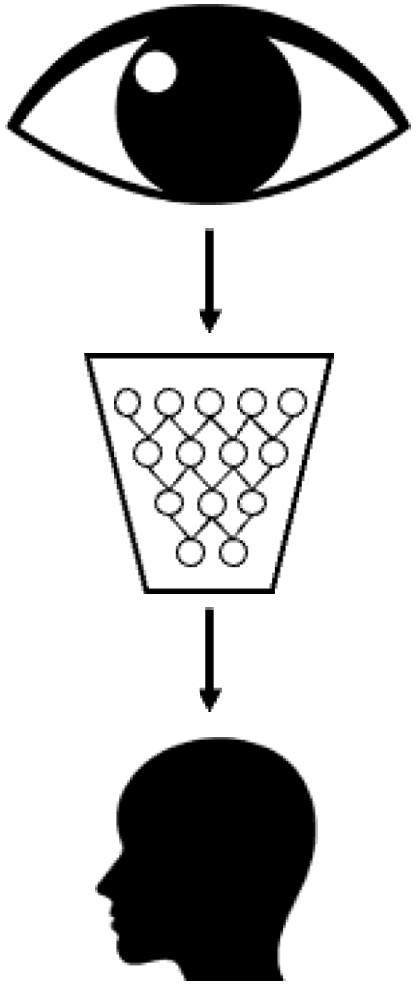
Biología y Medicina

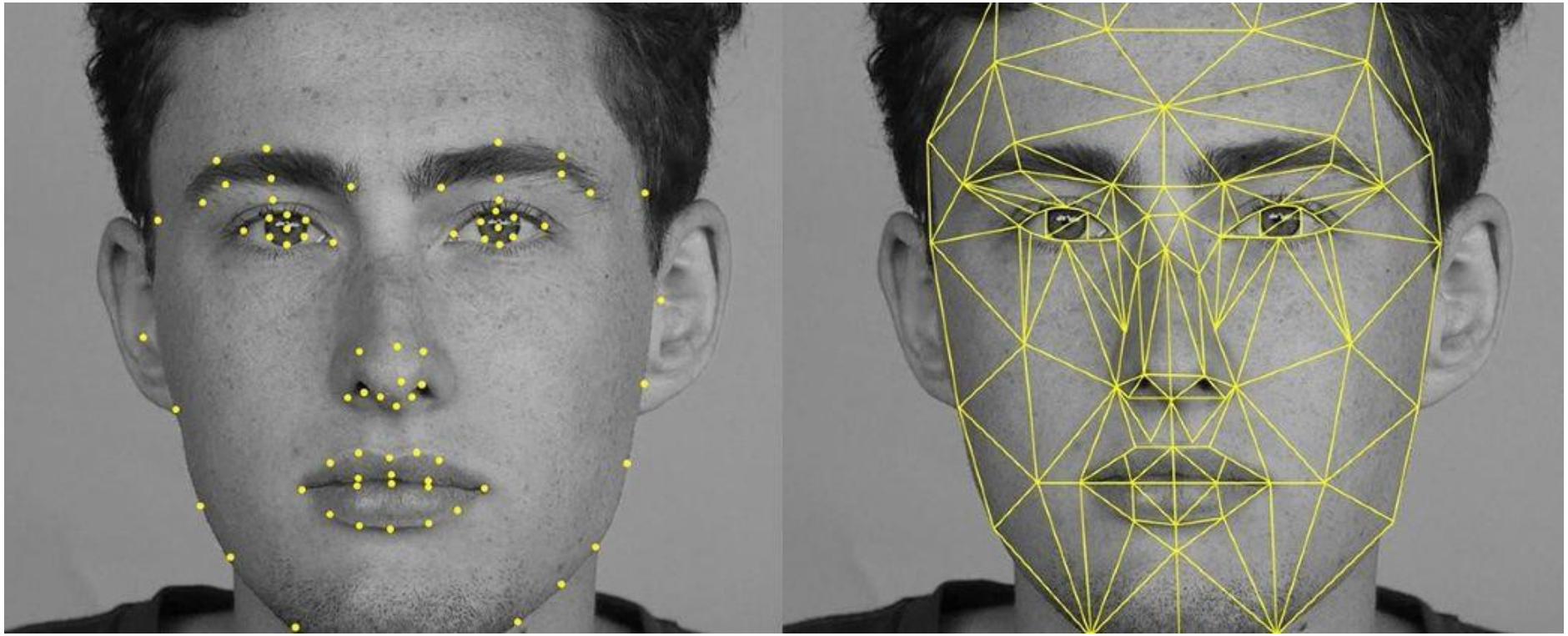


Computación móvil

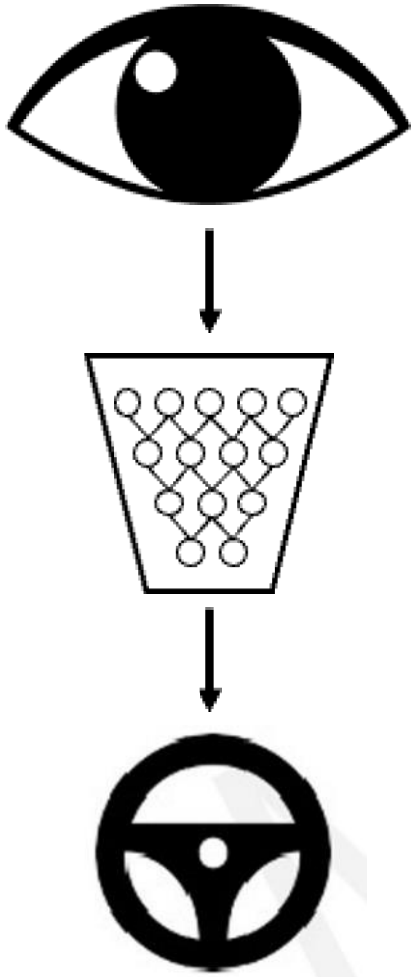
Conducción autónoma

Impacto: Detección y Reconocimiento Facial

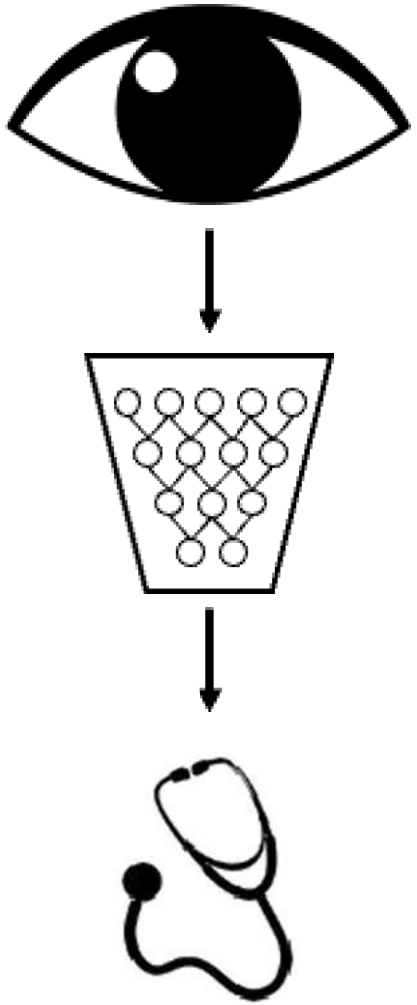




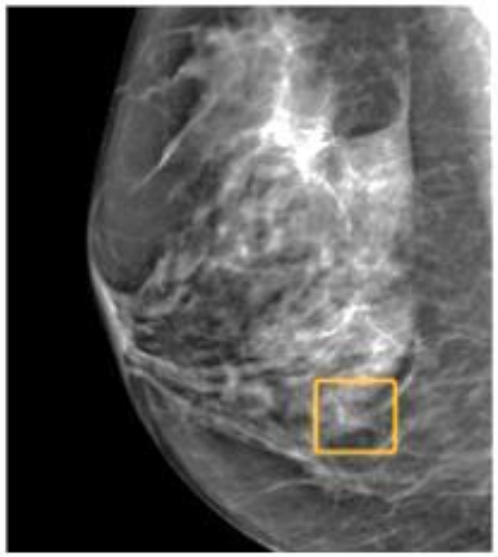
Impacto: Vehículos Autónomos



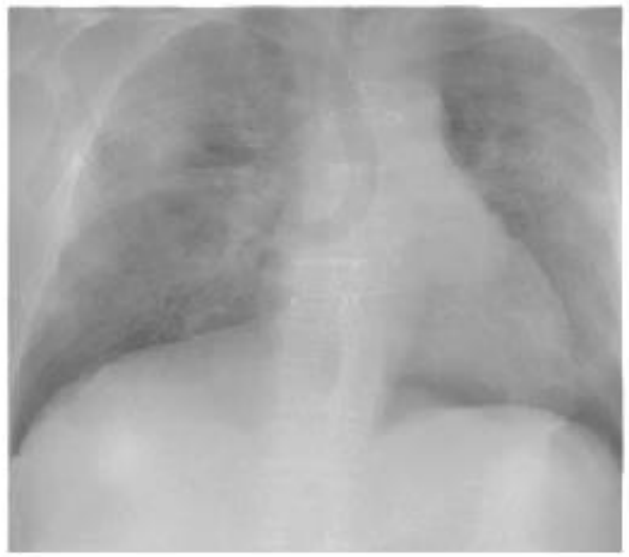
Impacto: Medicina, Biología, Atención Sanitaria



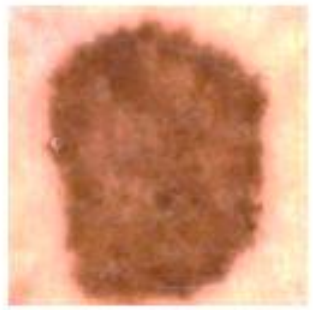
Cáncer de Mama



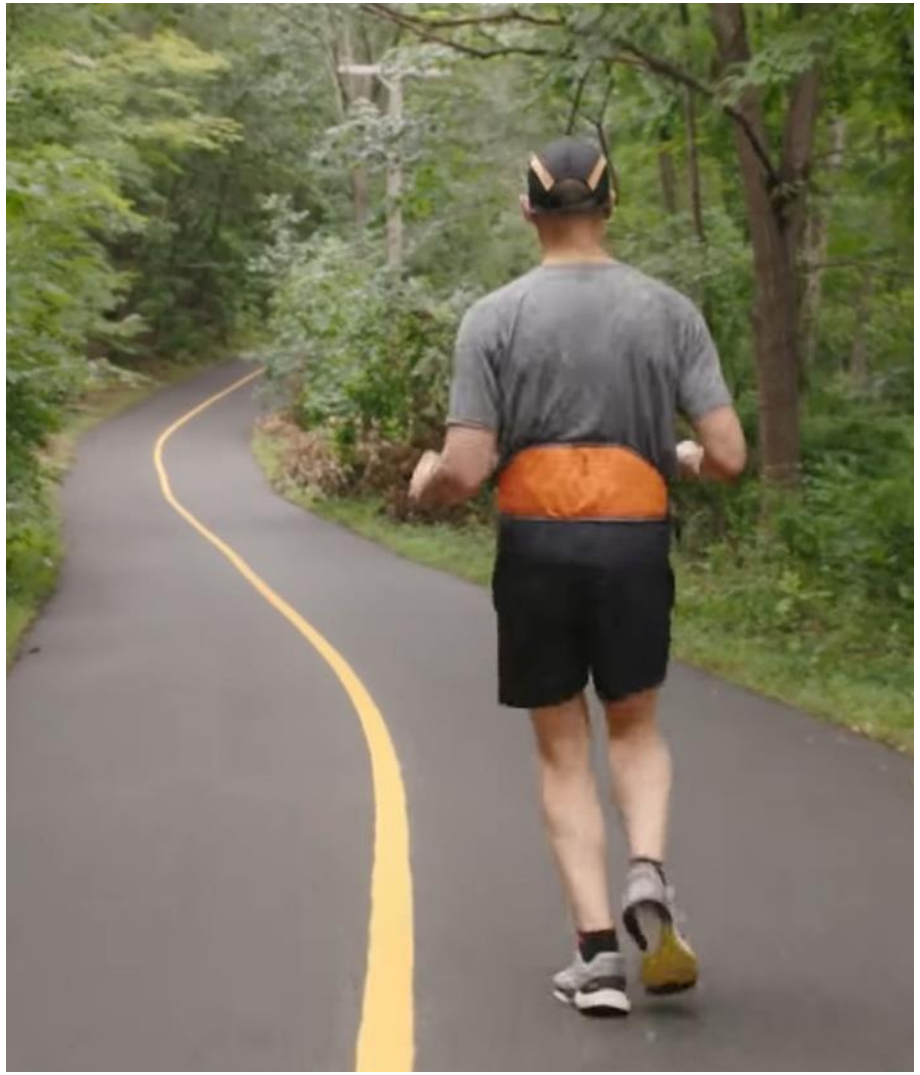
COVID 19

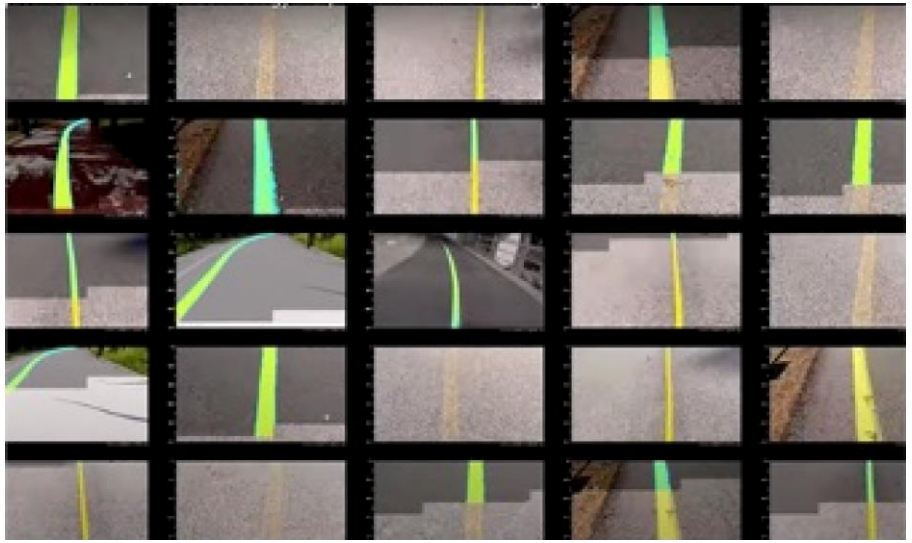


Cáncer de Piel



Impacto: Accesibilidad



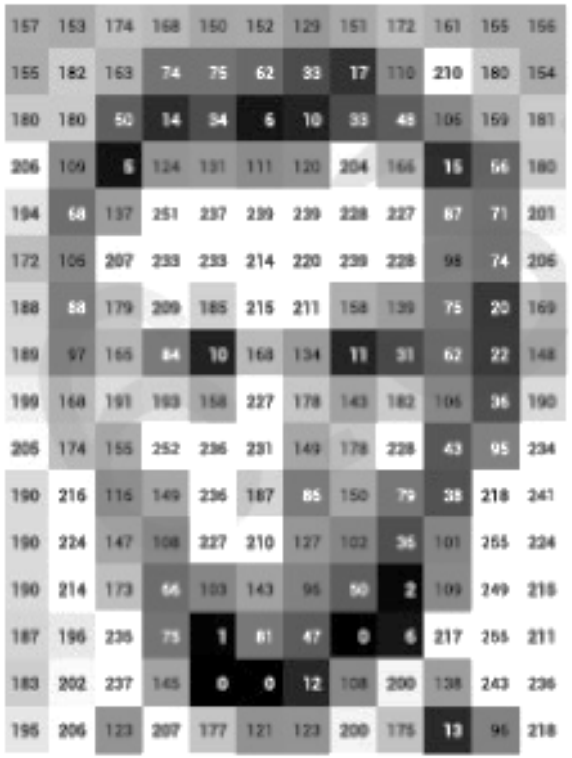


Lo que las computadoras “ven”

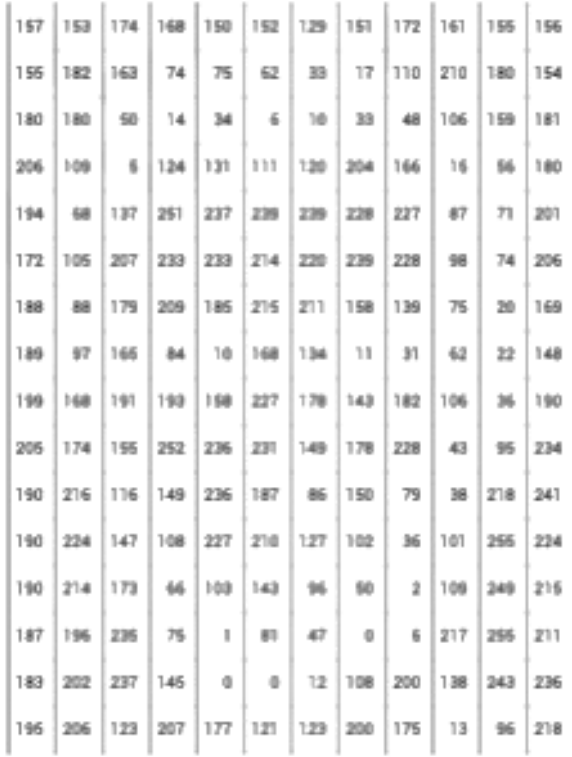
Las imágenes son números

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería



Lo que las computadoras ven

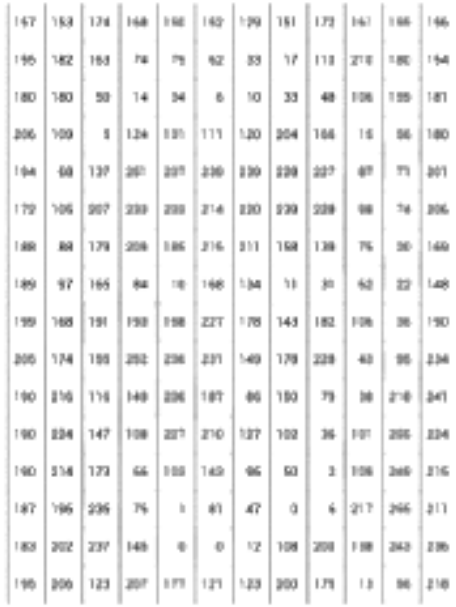


Una imagen es solo una matriz de números [0, 255],es decir,
1080×1080×3 para una imagen RGB.

Tareas en Visión por Computador

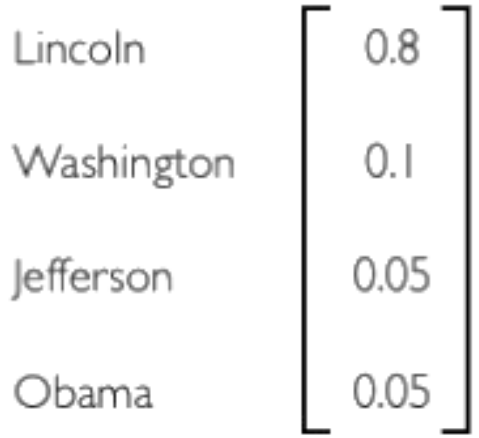
UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Imagen de entrada



Representación de píxeles

Clasificación



Regresión: La variable de salida toma un valor continuo.

Clasificación: La variable de salida toma una etiqueta de clase. Puede producir la probabilidad de pertenencia a una clase en particular.

Detección de Características de Alto Nivel

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

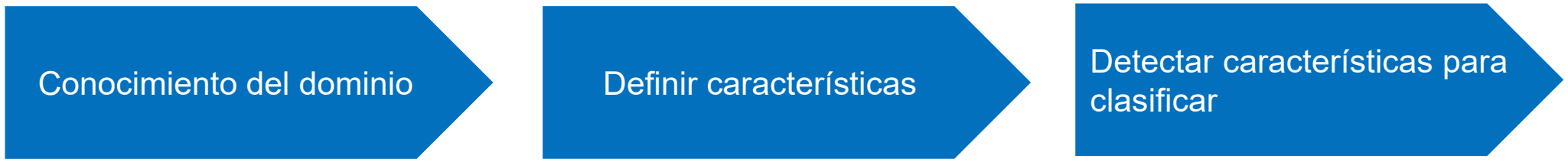
Identifiquemos las características clave en cada categoría de imagen:

Nariz, Ojos

Ruedas, Placa

Puerta, Ventanas

Extracción Manual de Características



Problemas?

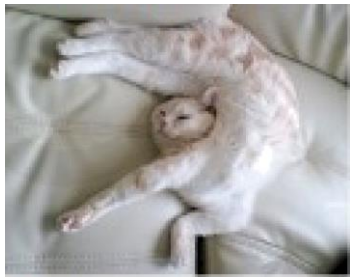
Extracción Manual de Características

Conocimiento del dominio → Definir características → Detectar características para clasificar

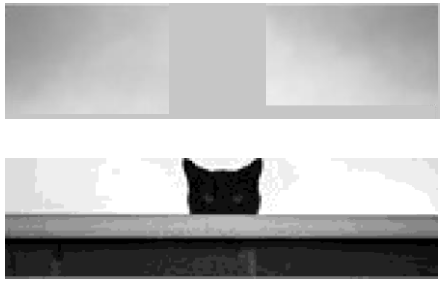
Variación de punto de vista

Variación de escala

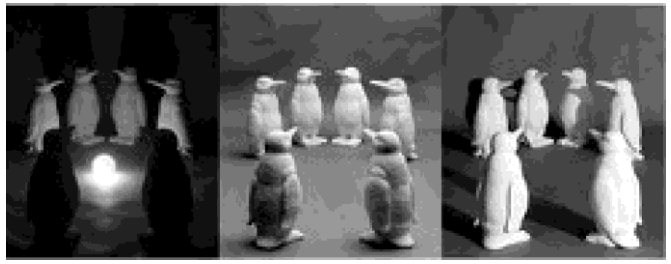
Deformación



Oclusión

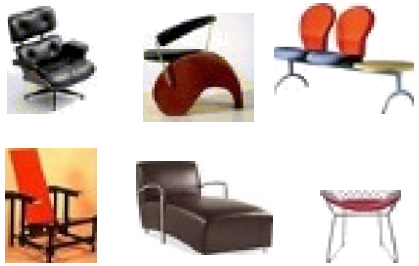


Condiciones de iluminación



Desorden de fondo

Variación intra-clase

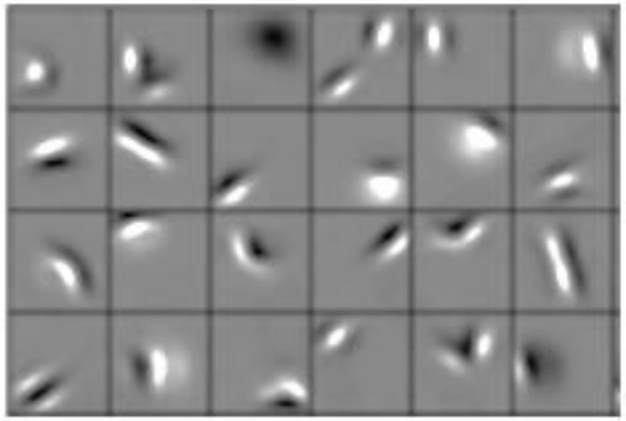


Aprendizaje de Representaciones de Características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Aprendemos una jerarquía de características directamente a partir de los datos en lugar de diseñarlas manualmente?

Características de bajo nivel:



Bordes, manchas oscuras

Características de nivel medio

Ojos, orejas, nariz

Características de alto nivel:

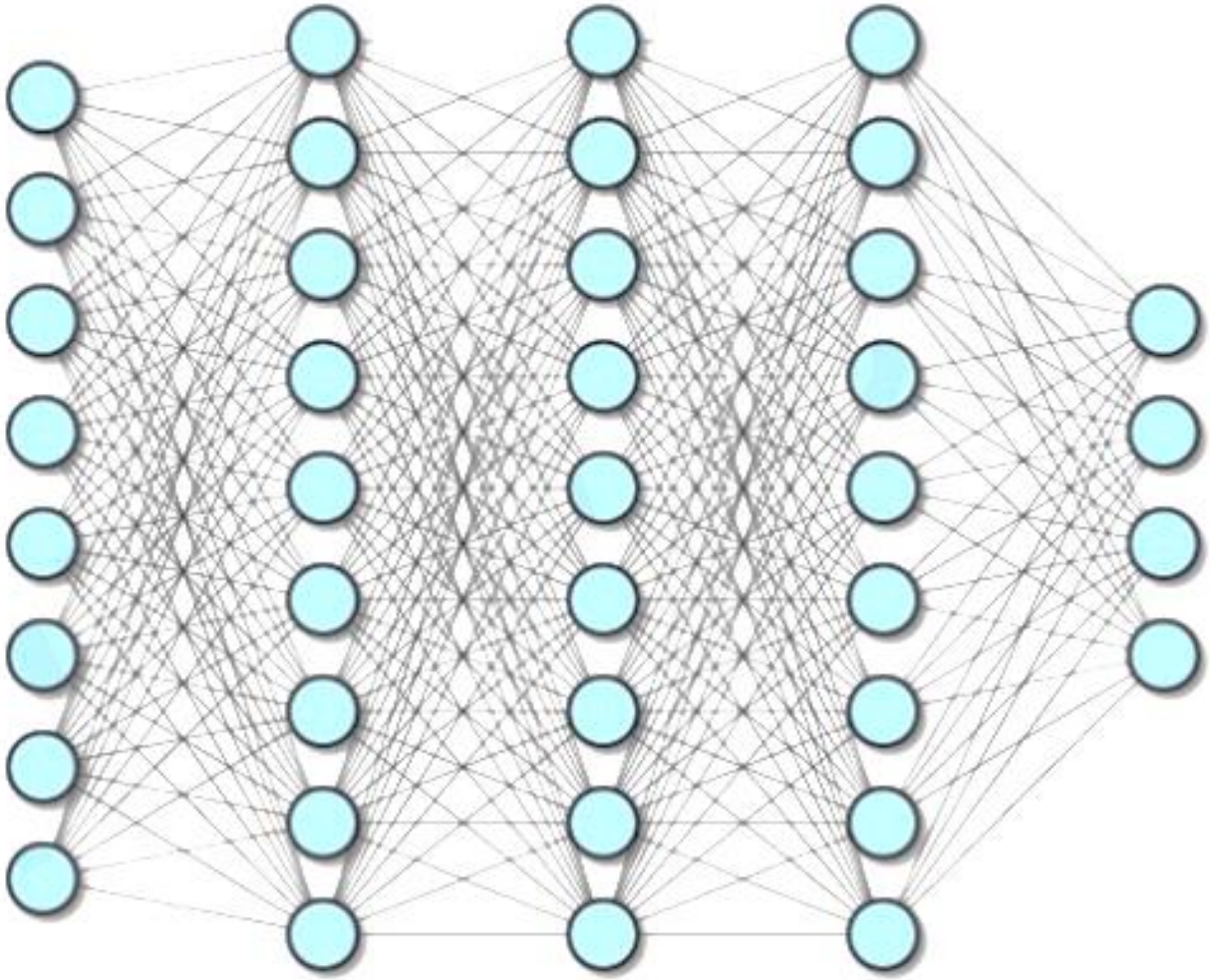
Estructura facial

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería

Aprendizaje de Características Visuales

Red Neuronal Totalmente Conectada

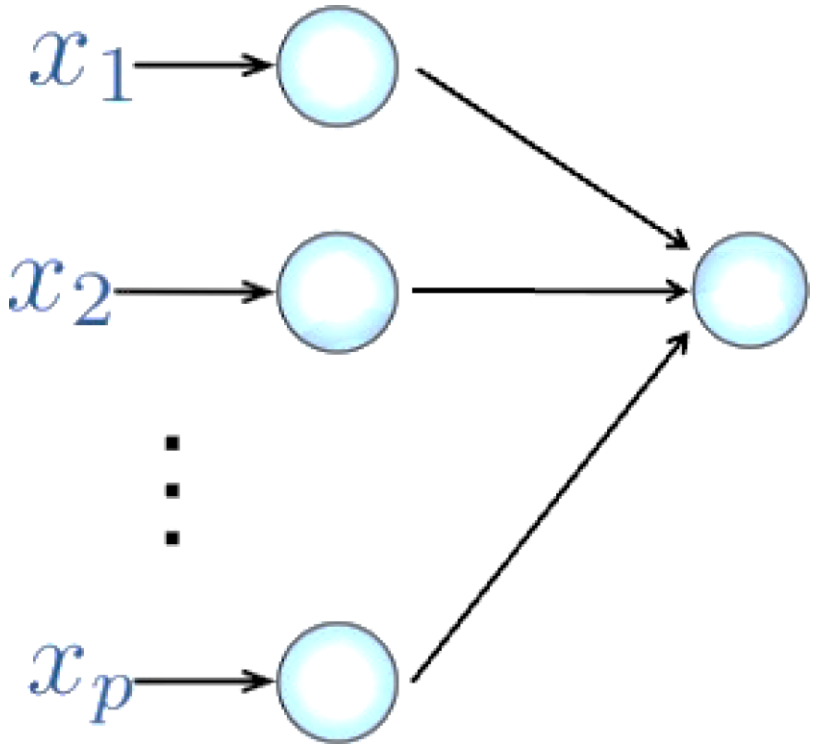


Red Neuronal Totalmente Conectada

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Entrada:

- Imagen 2D
- Vector de valores de píxeles



Totalmente Conectada:

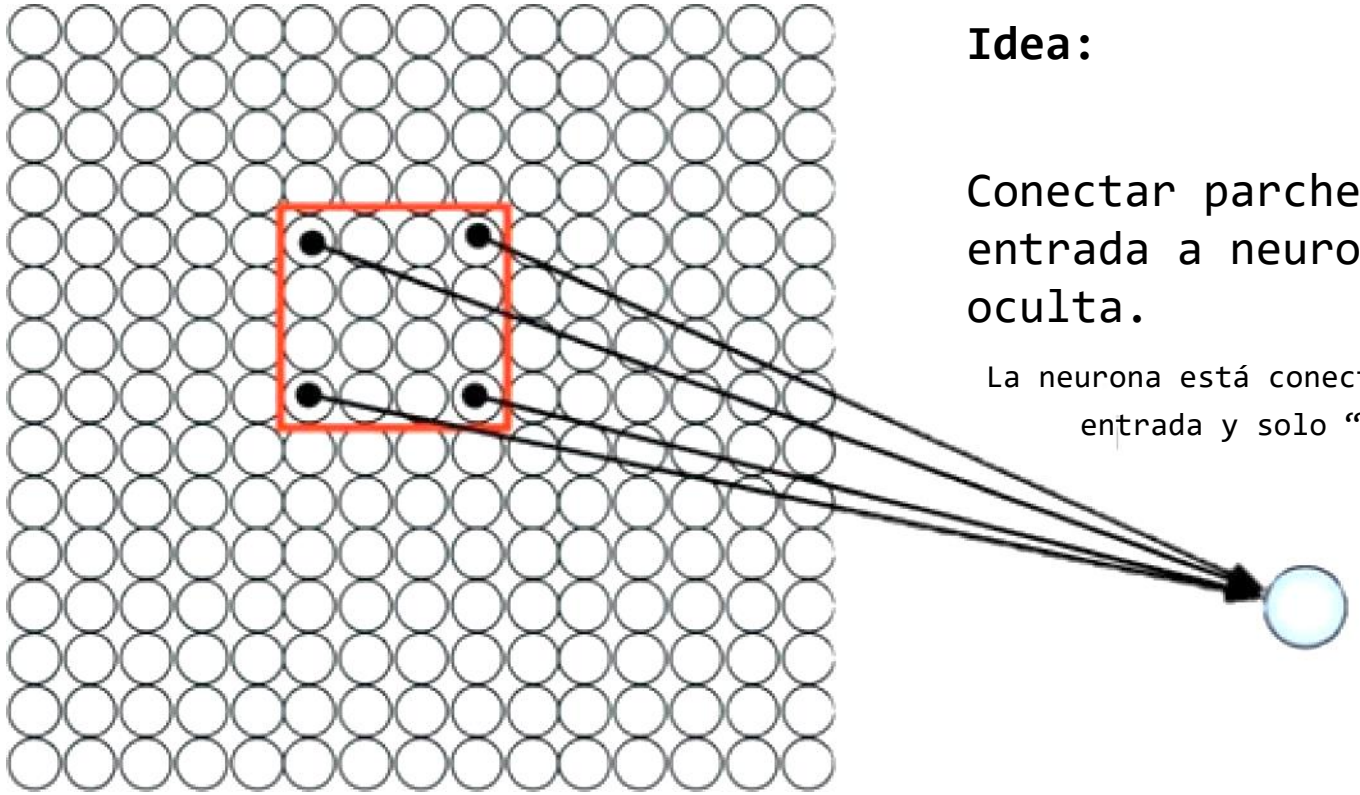
- Conecta cada neurona en la capa oculta con todas las neuronas en la capa de entrada
- ¡Sin información espacial!
- ¡Y muchísimos parámetros!

¿Cómo podemos usar la estructura espacial en la entrada para informar la arquitectura de la red?

Usando la Estructura Espacial

Entrada:

Imagen 2D, matriz de valores de píxeles

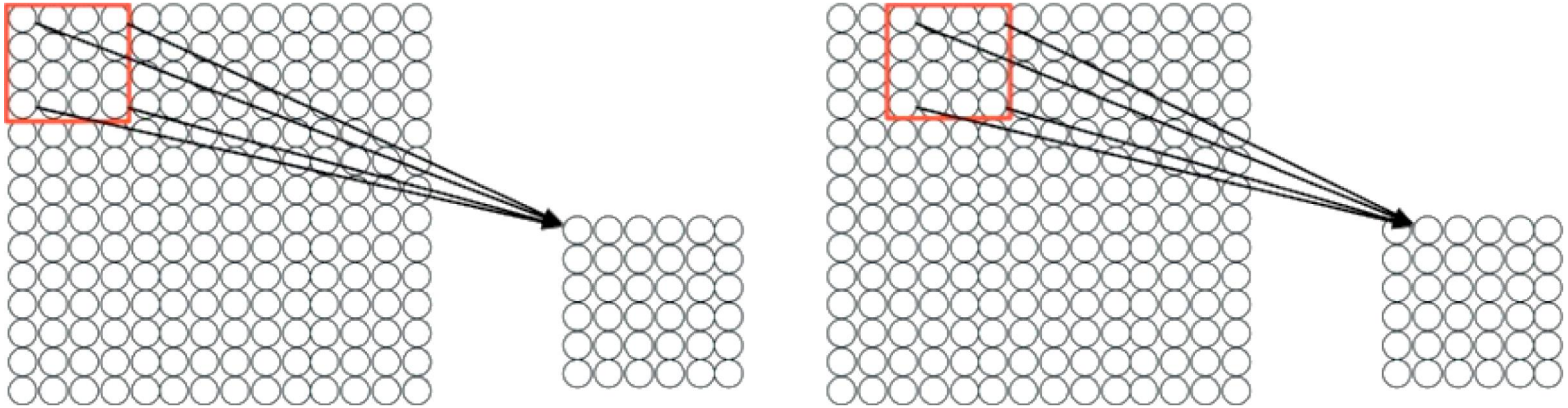


Idea:

Conectar parches de la entrada a neuronas en la capa oculta.

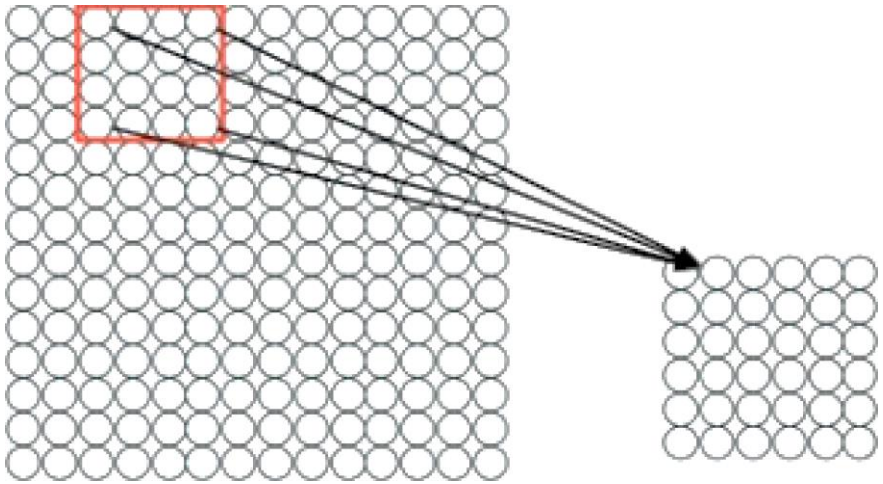
La neurona está conectada a una región de la entrada y solo “ve” esos valores.

Usando la Estructura Espacial



Conectar un parche en la capa de entrada a una única neurona en la capa siguiente.
Usar una ventana deslizante para definir las conexiones.
¿Cómo podemos ponderar el parche para detectar características particulares?

Extracción de Características con Convolución



- Filtro de tamaño 4×4: 16 pesos diferentes
- Aplicar este mismo filtro a parches de 4×4 en la entrada
- Desplazar 2 píxeles para el siguiente parche

Esta operación por parches es **convolución**

1. Aplicar un conjunto de pesos –un filtro– para extraer características locales
2. Usar múltiples filtros para extraer diferentes características
3. Compartir espacialmente los parámetros de cada filtro

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería

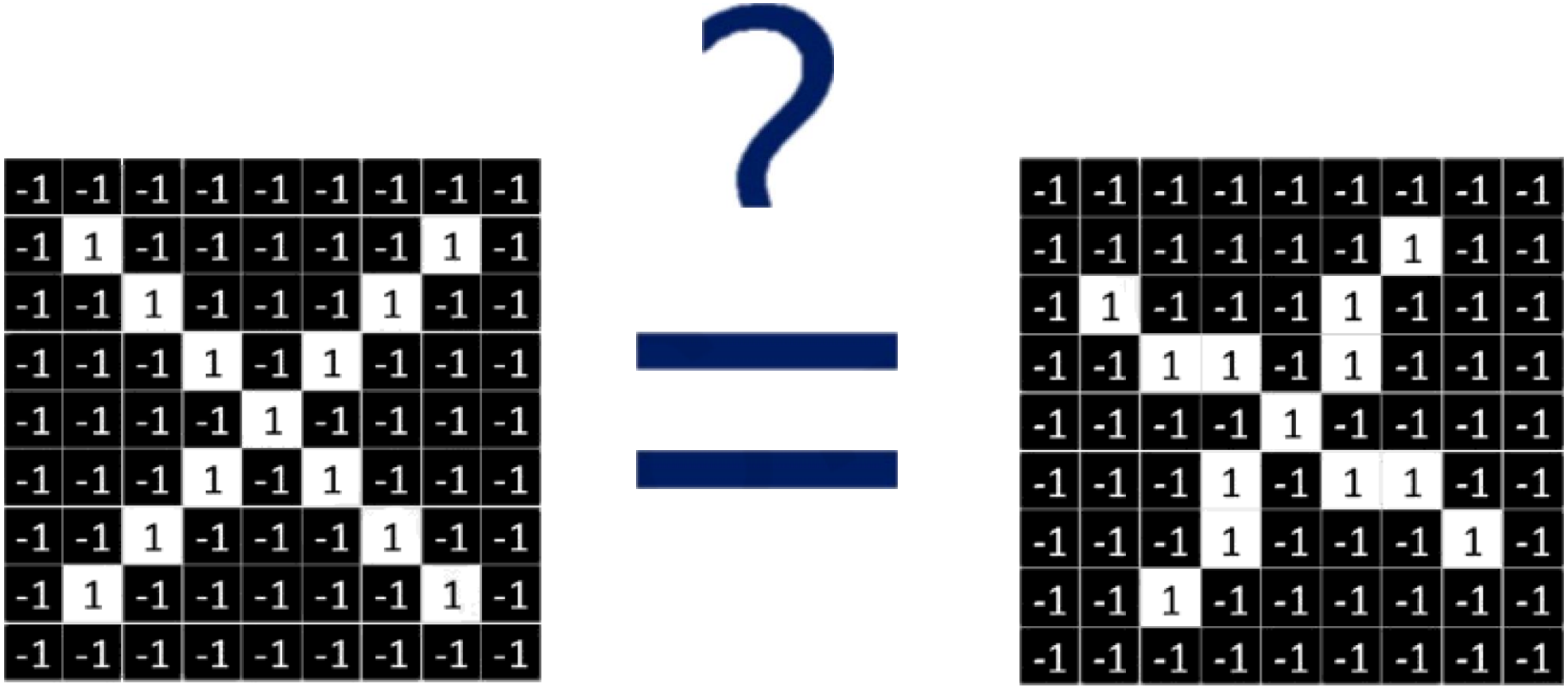
Extracción de Características y Convolución

Un Estudio de Caso

X ó X?

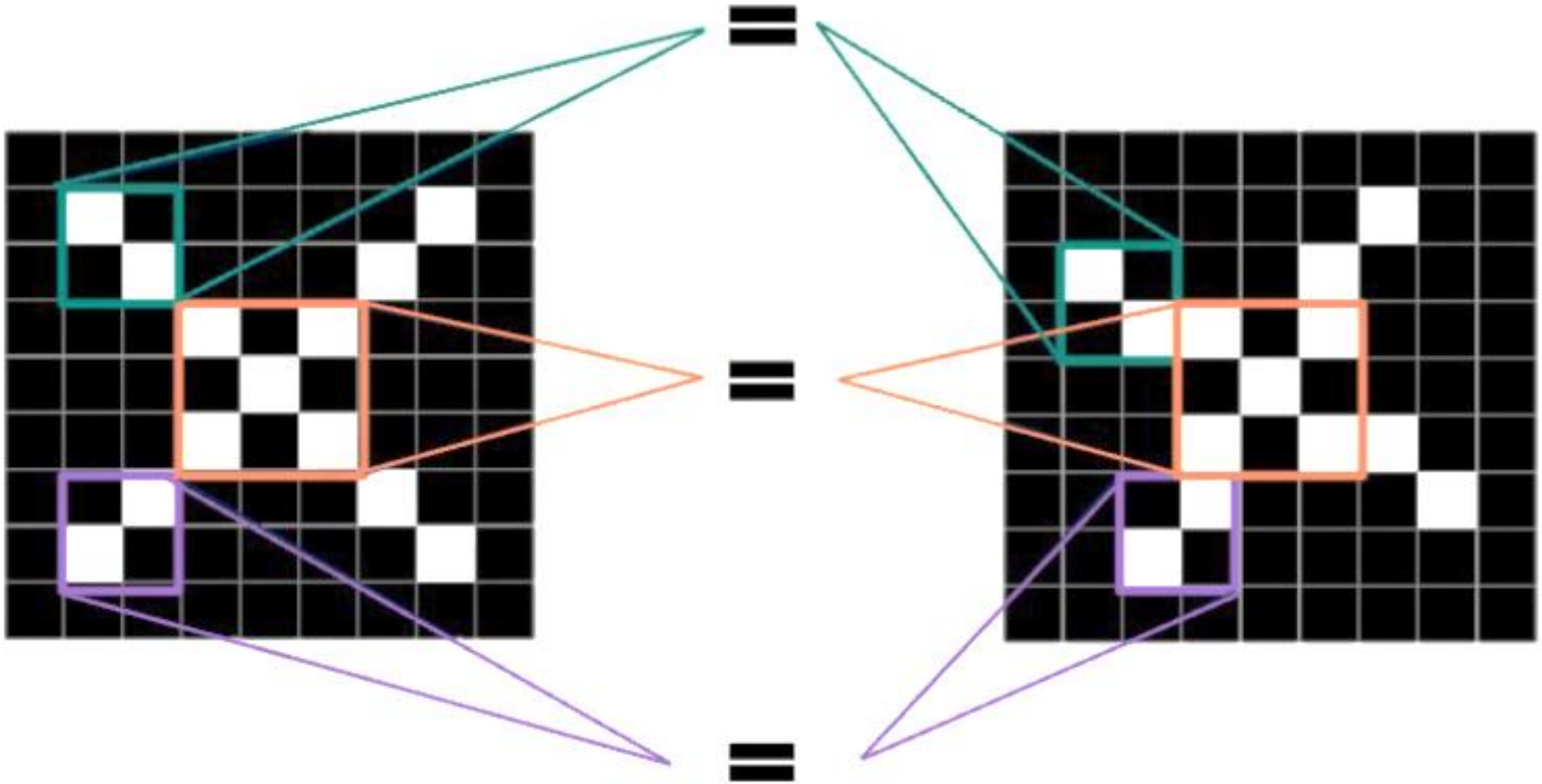
UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería



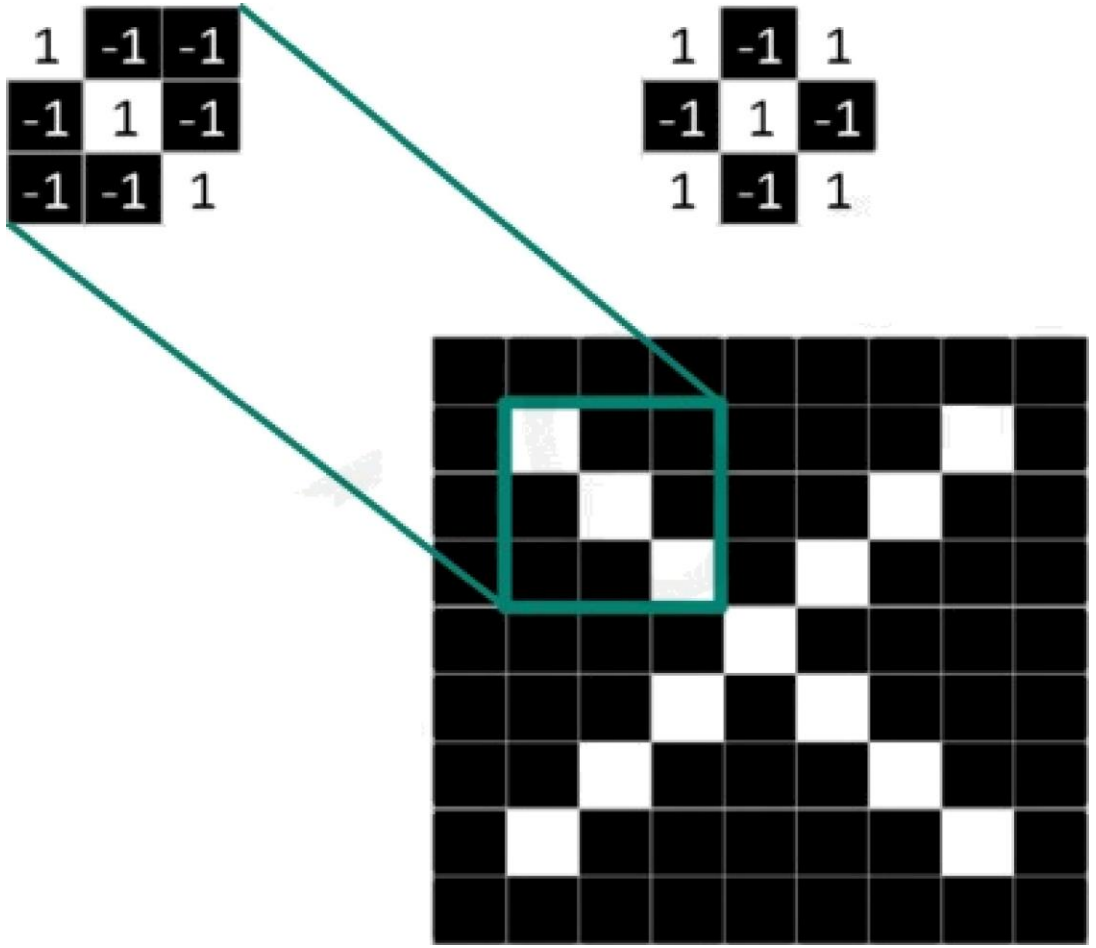
La imagen se representa como una matriz de valores de píxeles… ¡y las computadoras son literales!
Queremos poder clasificar una X como una X incluso si está desplazada, reducida, rotada o deformada.

Características de la X

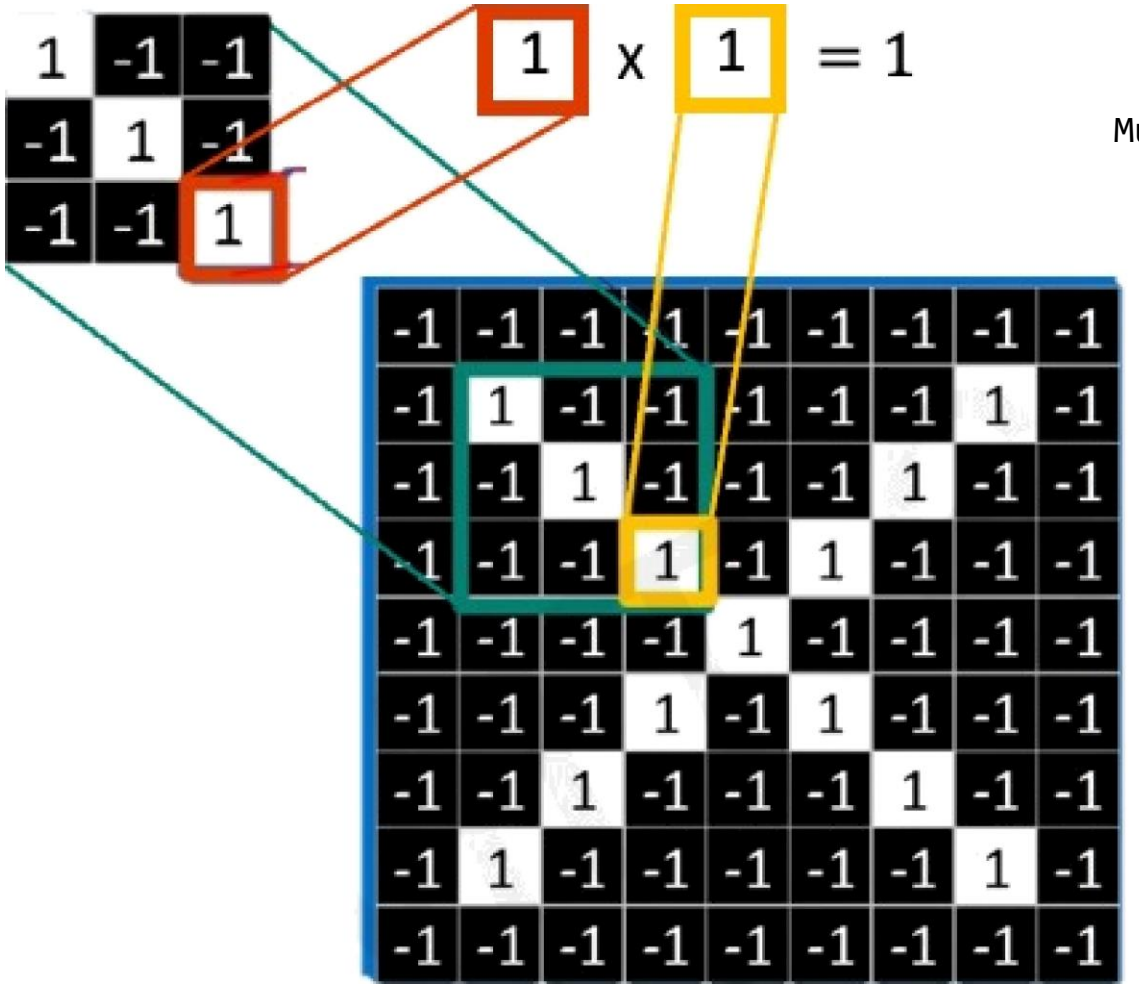


Filtros para Detectar Características de la X

Filtros



La Operación de Convolución



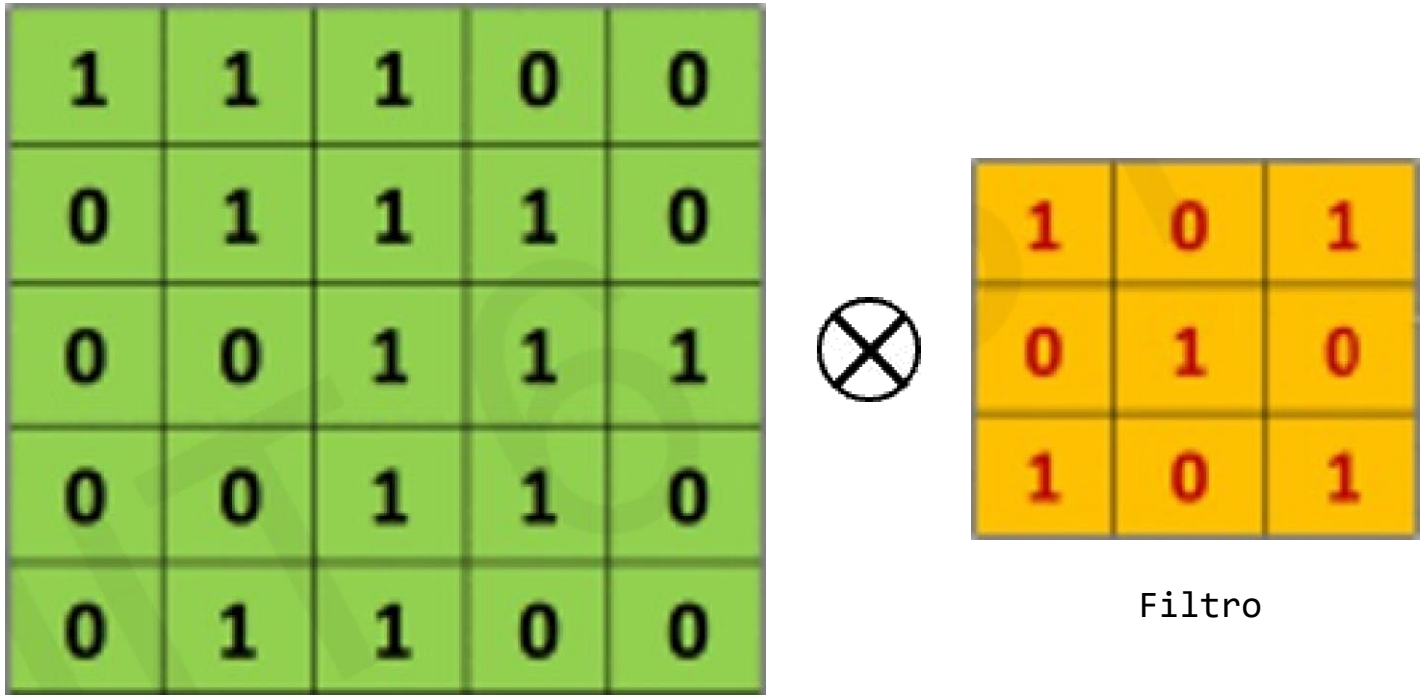
Multiplicación elemento a elemento

Sumar salidas

$$\odot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 9$$

La Operación de Convolución

Supongamos que queremos calcular la convolución de una imagen de 5×5 y un filtro de 3×3:



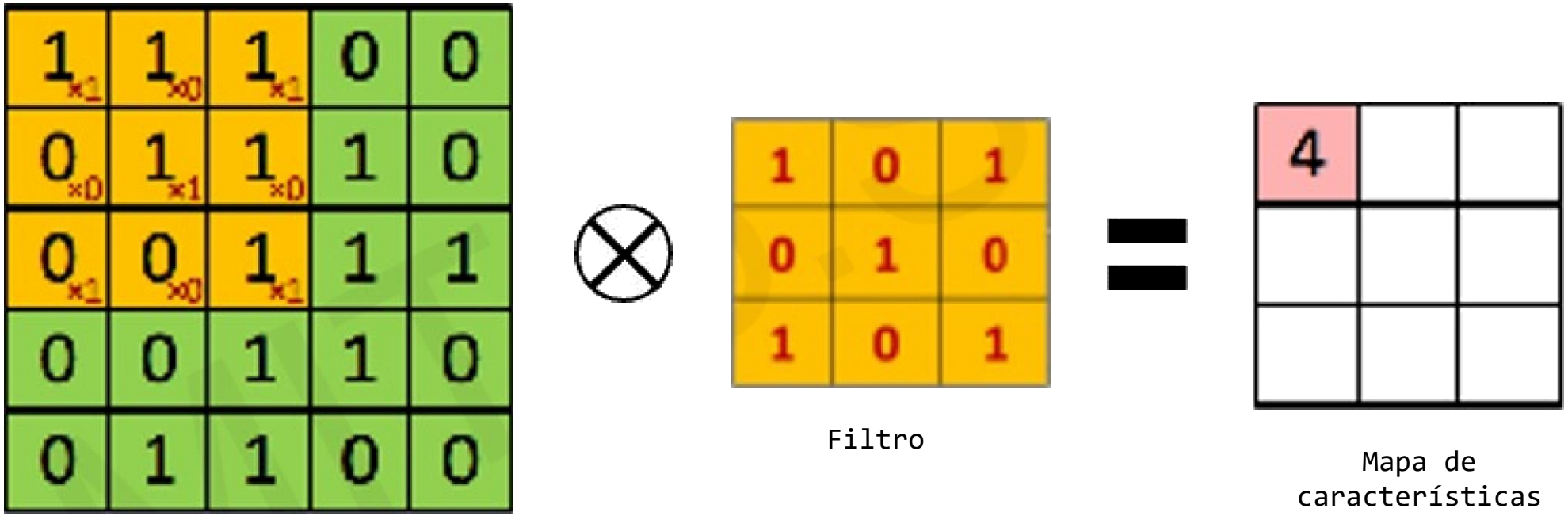
Imagen

Filtro

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas.

La Operación de Convolución

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



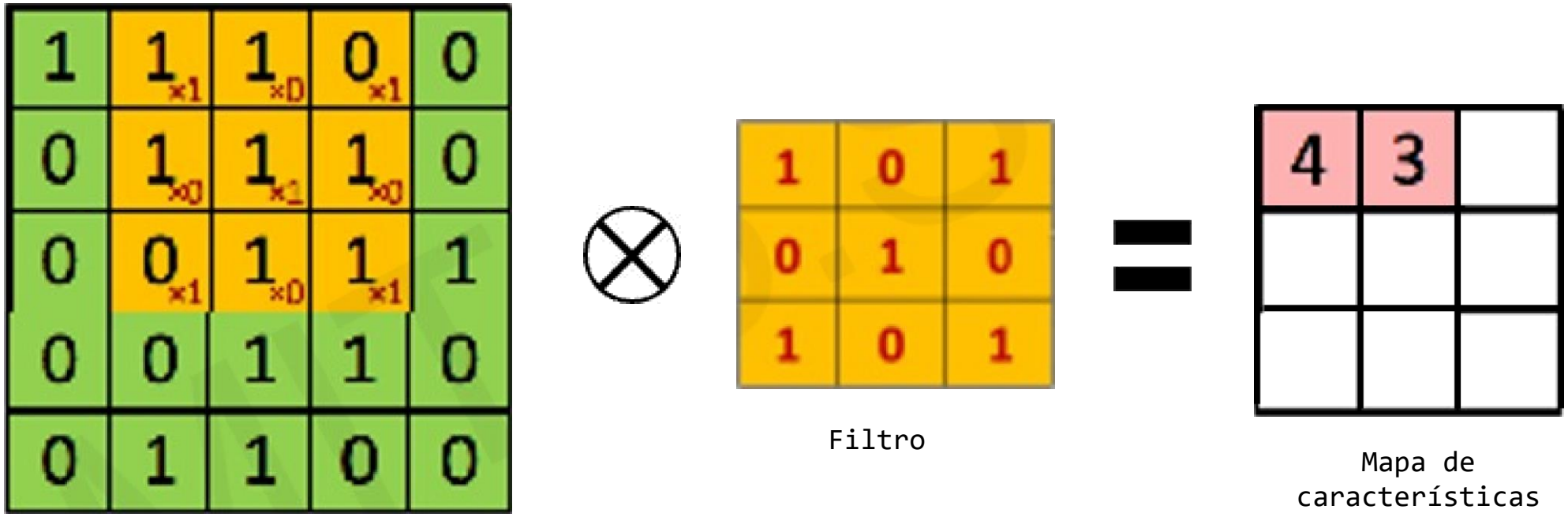
Filtro

Mapa de características

La Operación de Convolución

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



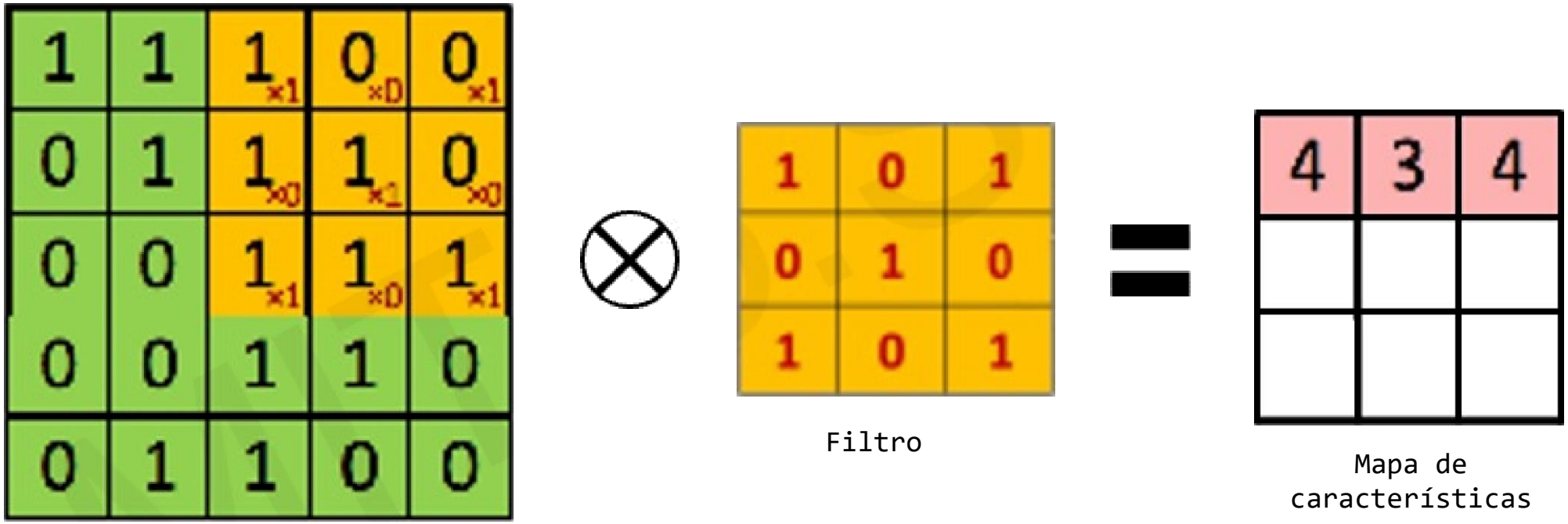
Filtro

Mapa de características

La Operación de Convolución

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:

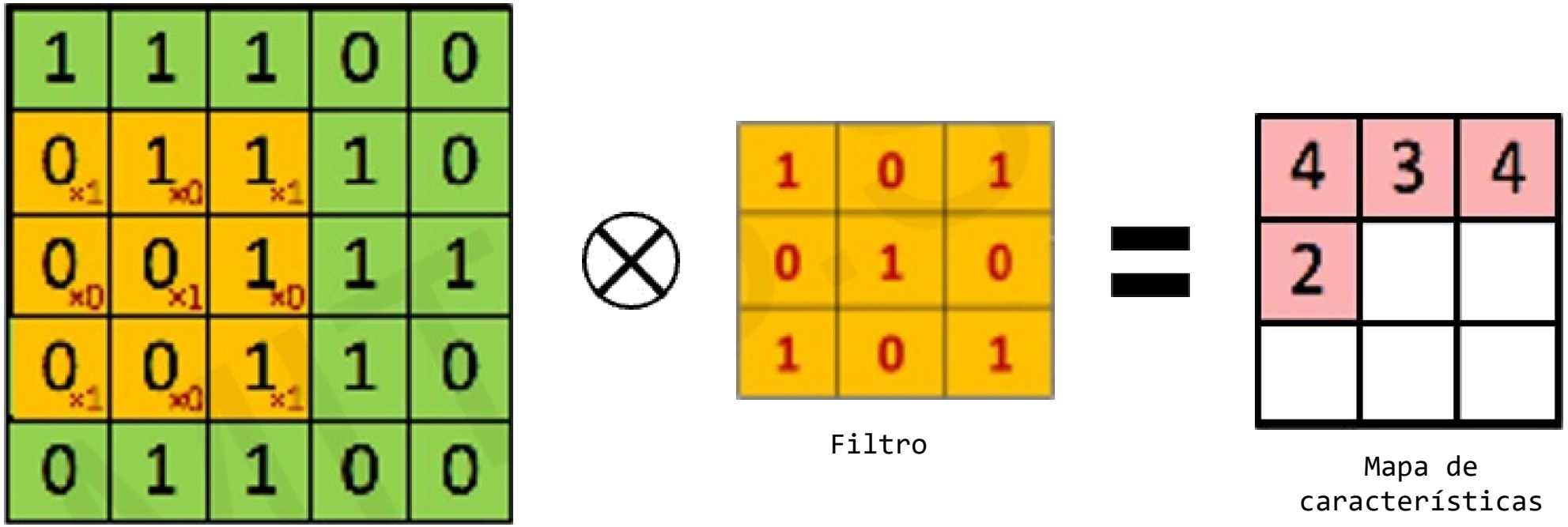


Filtro

Mapa de características

La Operación de Convolución

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:

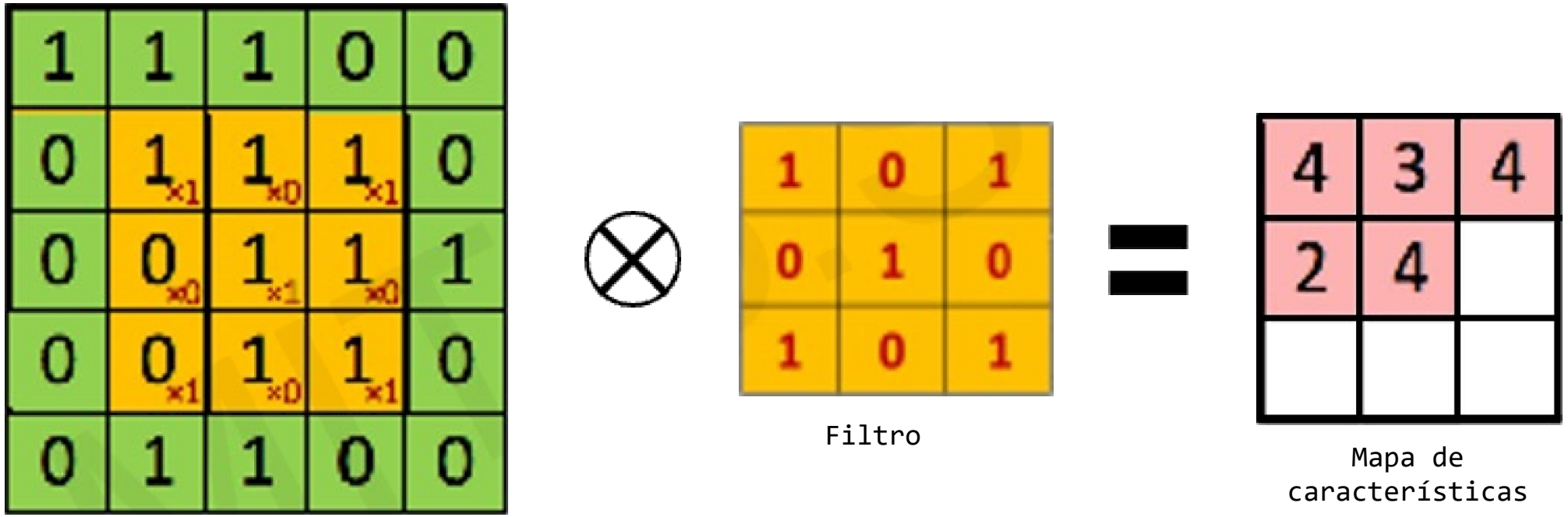


Filtro

Mapa de características

La Operación de Convolución

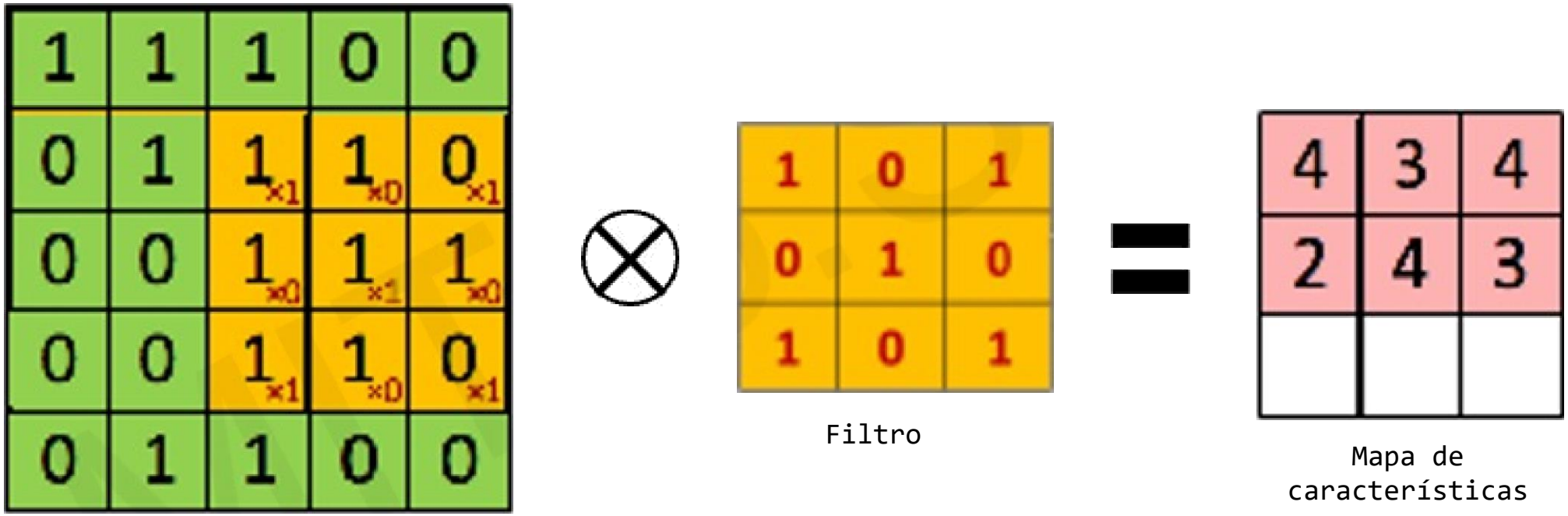
Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



La Operación de Convolución

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



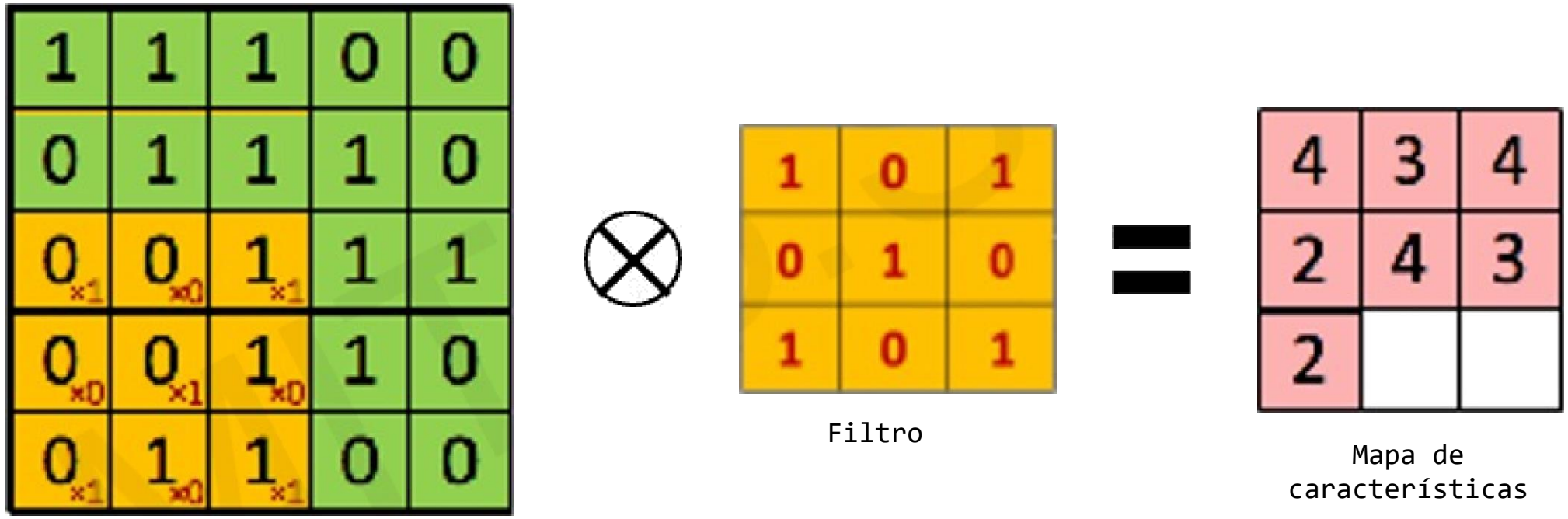
Filtro

Mapa de características

La Operación de Convolución

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:

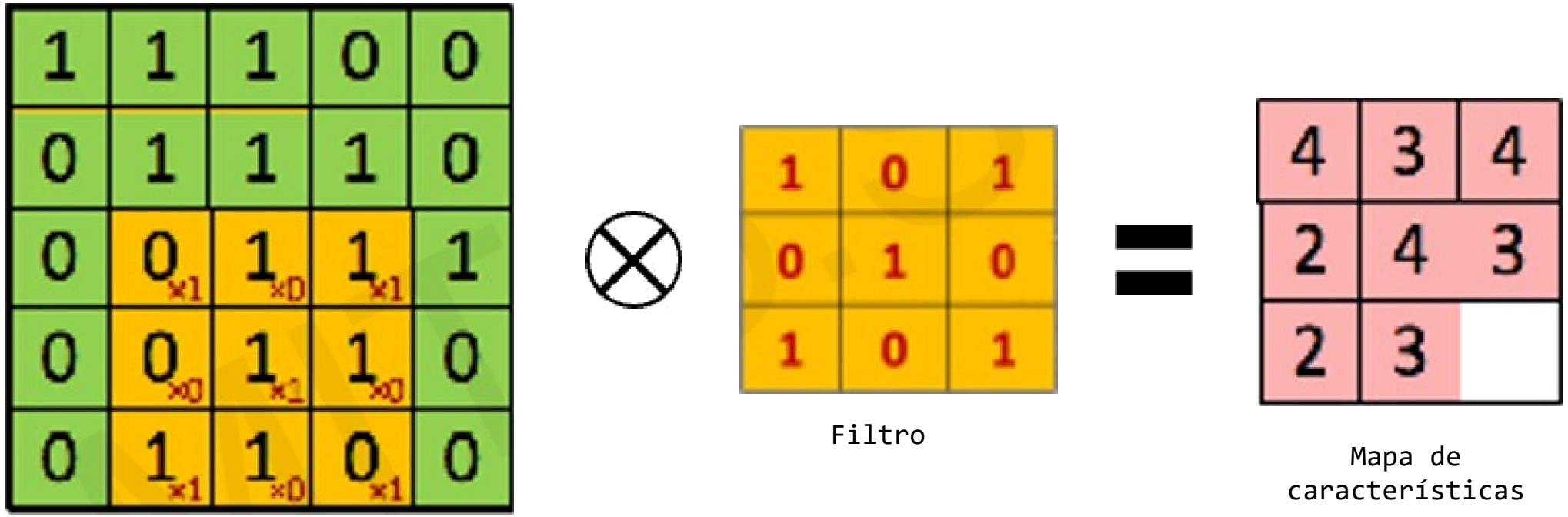


Filtro

Mapa de características

La Operación de Convolución

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



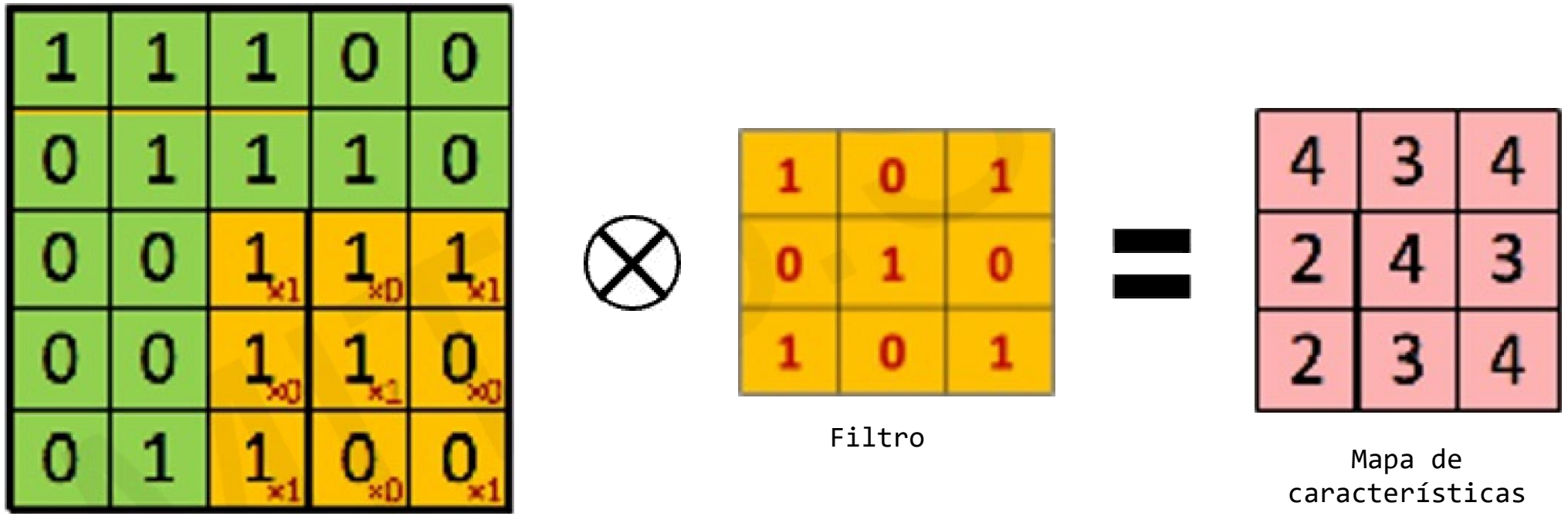
Filtro

Mapa de características

La Operación de Convolución

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Deslizamos el filtro 3×3 sobre la imagen de entrada, multiplicamos elemento a elemento y sumamos las salidas:



Filtro

Mapa de características

Producción de Mapas de Características

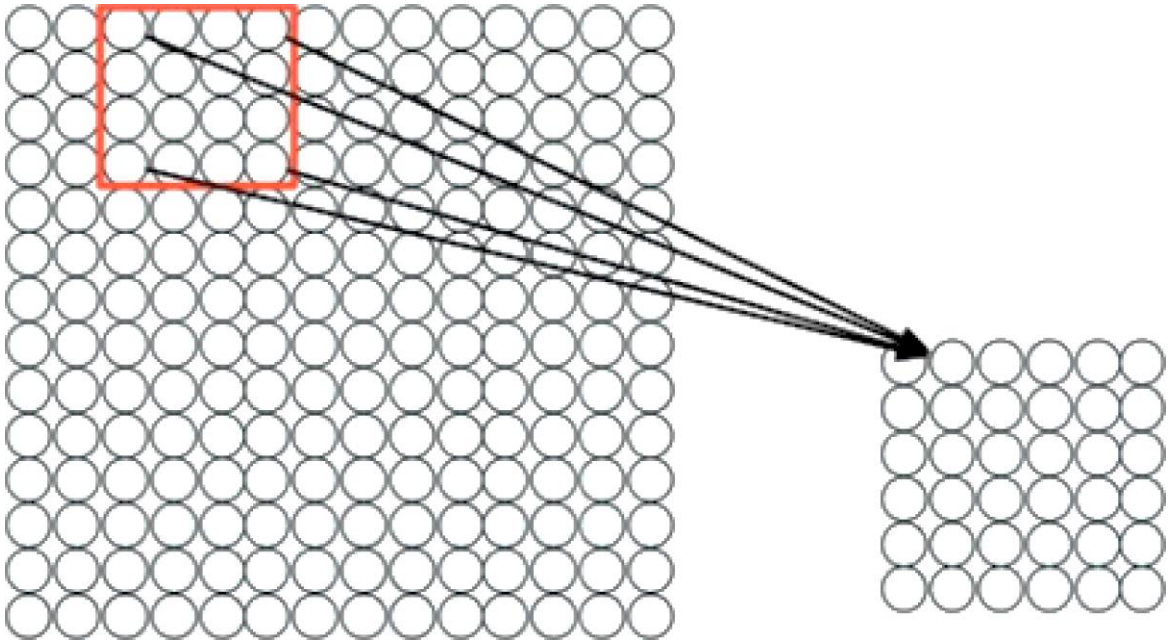
Original

Afilado

Detección de bordes

Detección de bordes “fuerte”

Extracción de Características con Convolución



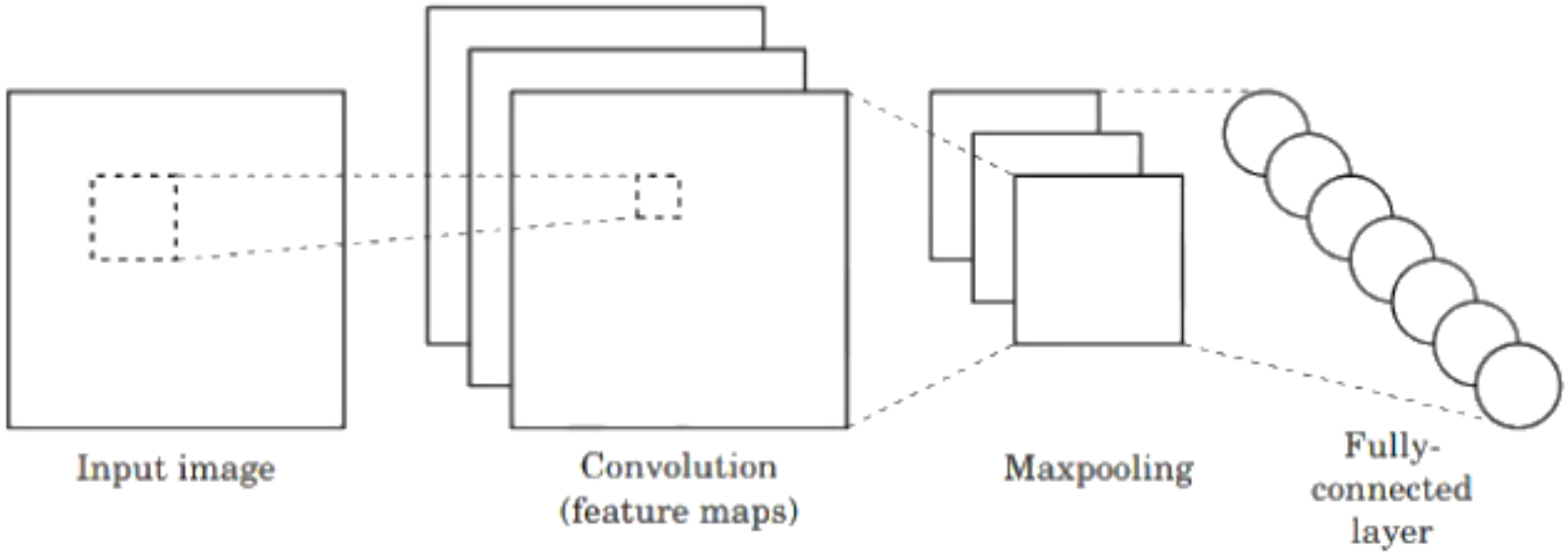
1. Aplicar un conjunto de pesos –un filtro– para extraer características locales
2. Usar múltiples filtros para extraer diferentes características
3. Compartir espacialmente los parámetros de cada filtro.

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería

Redes Neuronales Convolucionales (CNNs)

CNNs para Clasificación



`tf.keras.layers.Conv2D`

`tf.keras.activations.*`

`tf.keras.layers.MaxPool2D`

1. **Convolución: A**plicar filtros para generar mapas de características
2. **No linealidad: A** menudo ReLU.
3. **Pooling**: Operación de reducción de muestreo en cada mapa de características.

`torch.nn.Conv2d`

`torch.nn.ReLU,…`

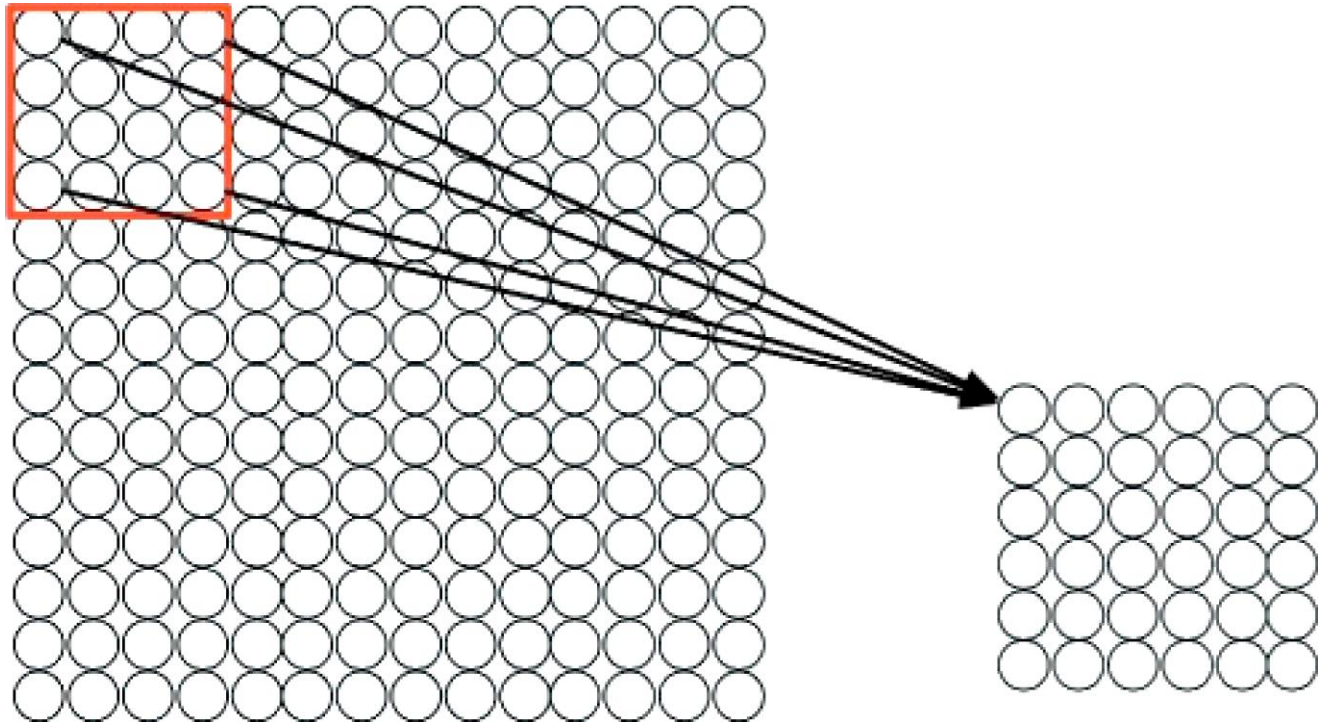
`torch.nn.MaxPool2d`

Entrenar el modelo con datos de imagen.
Aprender los pesos de los filtros en las capas convolucionales.

Capas Convolucionales – Conectividad Local

`tf.keras.layers.Conv2D`

`torch.nn.Conv2d`



Filtro 4×4: matriz de pesos W_{ij}

$$\sum_{i=1}^{4}\sum_{j=1}^{4} w_{ij}\, x_{i+p,j+q} + b$$

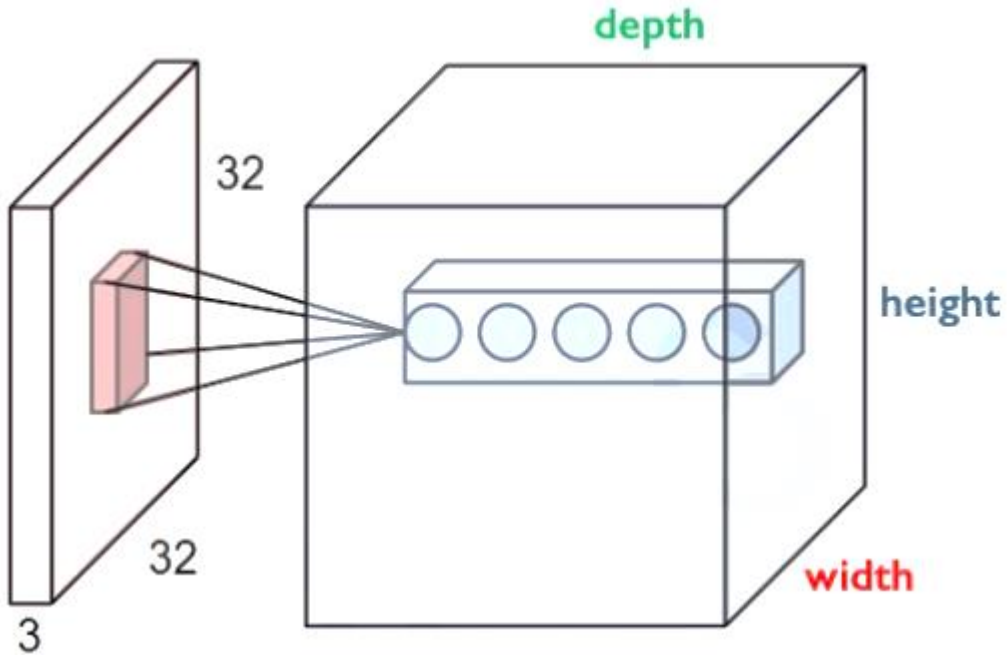
para la neurona (p,q) en la capa oculta

Para una neurona en la capa oculta:

- Tomar entradas de un parche
- Calcular la suma ponderada
- Aplicar sesgo (bias)

1. Aplicar una ventana de pesos
2. Calcular combinaciones lineales
3. Activar con una función no lineal)

CNNs: Disposición Espacial del Volumen de Salida



Dimensiones de la capa:

$$h \; x \; w \; x \; d$$

donde h y w son dimensiones espaciales y d (profundidad) = número de filtros

Stride:
Tamaño del paso del filtro

Campo Receptivo:
Ubicaciones en la imagen de entrada a las que una neurona está conectada

```
tf.keras.layers.Conv2D( filters=d, kernel_size=(h,w), strides=s )
```

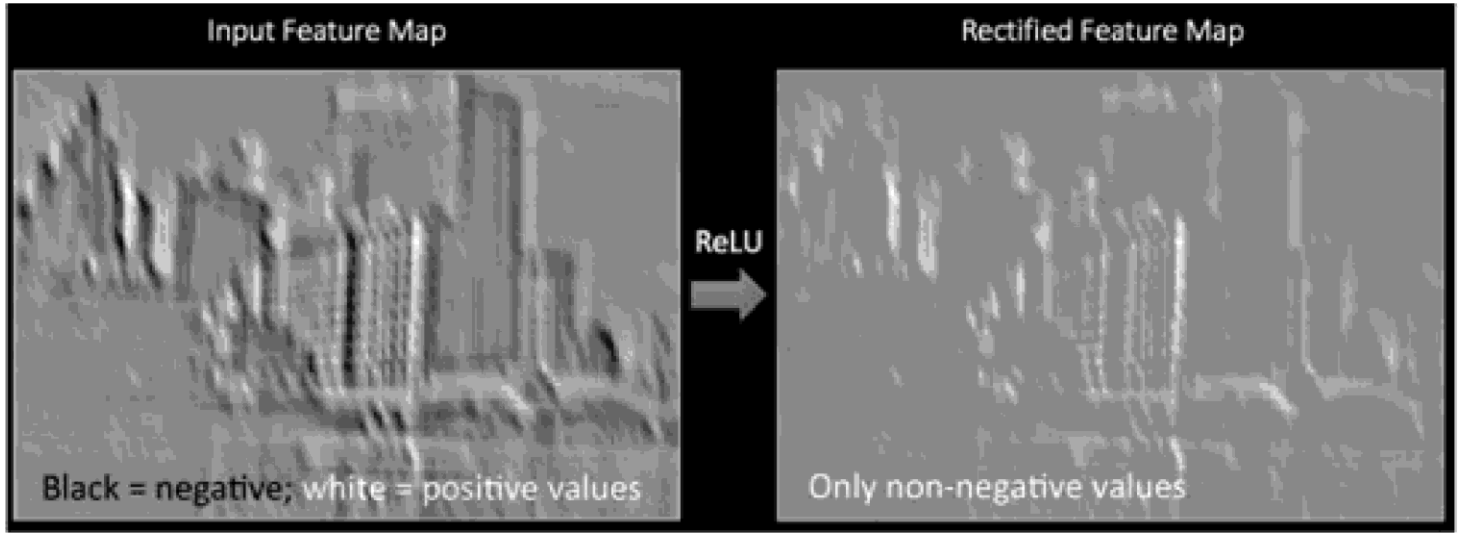
```
torch.nn.Conv2d( in_channels=3, out_channels=d, kernel_size=(h,w), stride=s )
```

¡Es necesario especificar la dimensionalidad de la entrada!

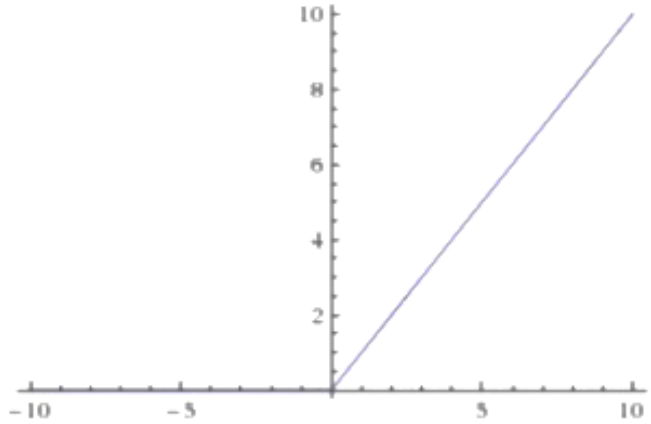
Introduciendo No Linealidad

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

- Aplicar después de cada operación de convolución (es decir, después de las capas convolucionales)
- ReLU: operación píxel a píxel que reemplaza todos los valores negativos por cero. ¡Operación no lineal!



Rectified Linear Unit (ReLU)

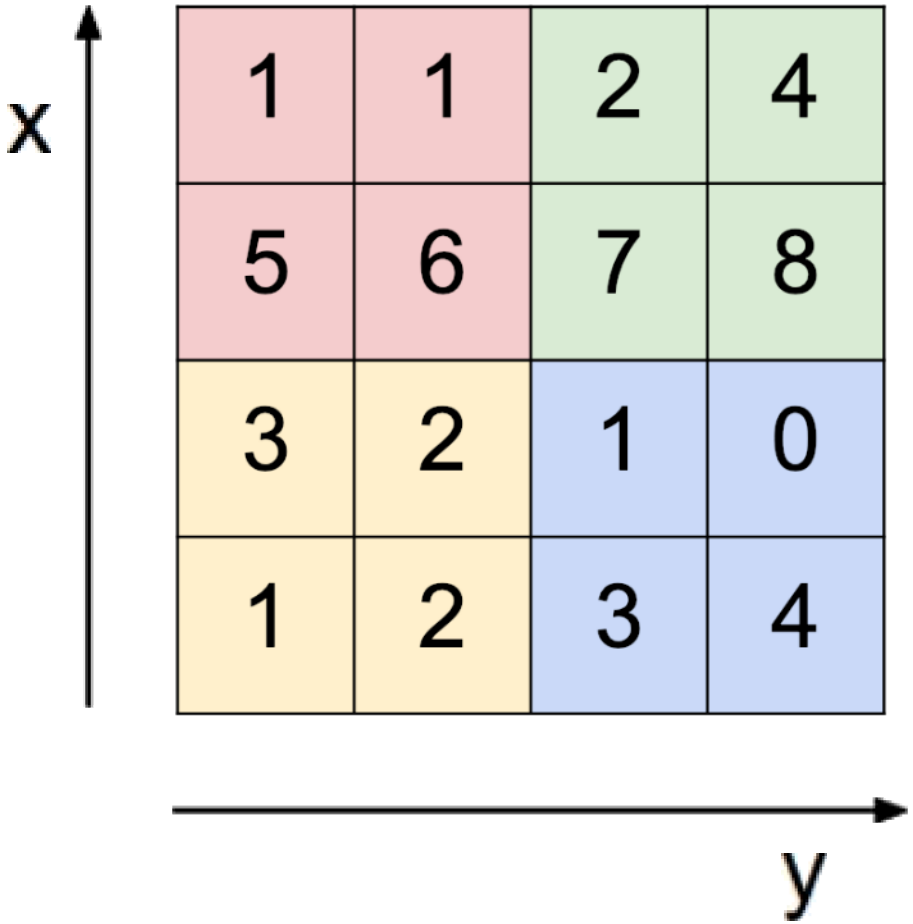


$$g(z) = \max(0, z)$$

`tf.keras.layers.ReLU`

`torch.nn.ReLU`

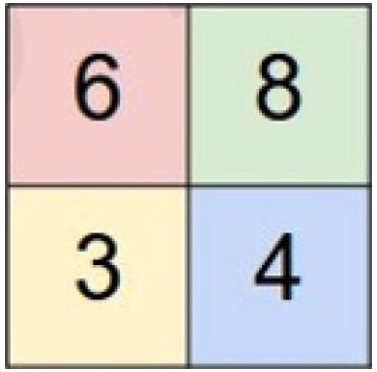
Pooling



Max pooling con filtros de 2×2 y stride 2

```
tf.keras.layers.MaxPool2D(
    pool_size=(2,2),
    strides=2
)
```

```
torch.nn.MaxPool2d(
    kernel_size=(2,2),
    stride=2
)
```



1. Reducción de dimensionalidad
2. Invariancia espacial

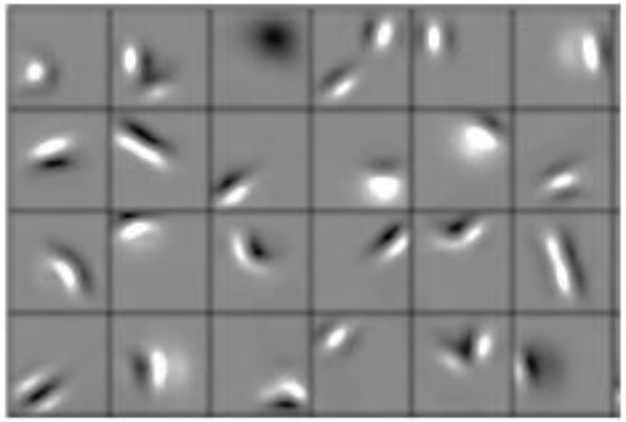
¿Cómo más podemos reducir la resolución y preservar la invariancia espacial?

Aprendizaje de Representaciones en CNNs Profundas

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Características de bajo nivel:



Bordes, manchas oscuras

Conv Layer 1

Características de nivel medio

Ojos, orejas, nariz

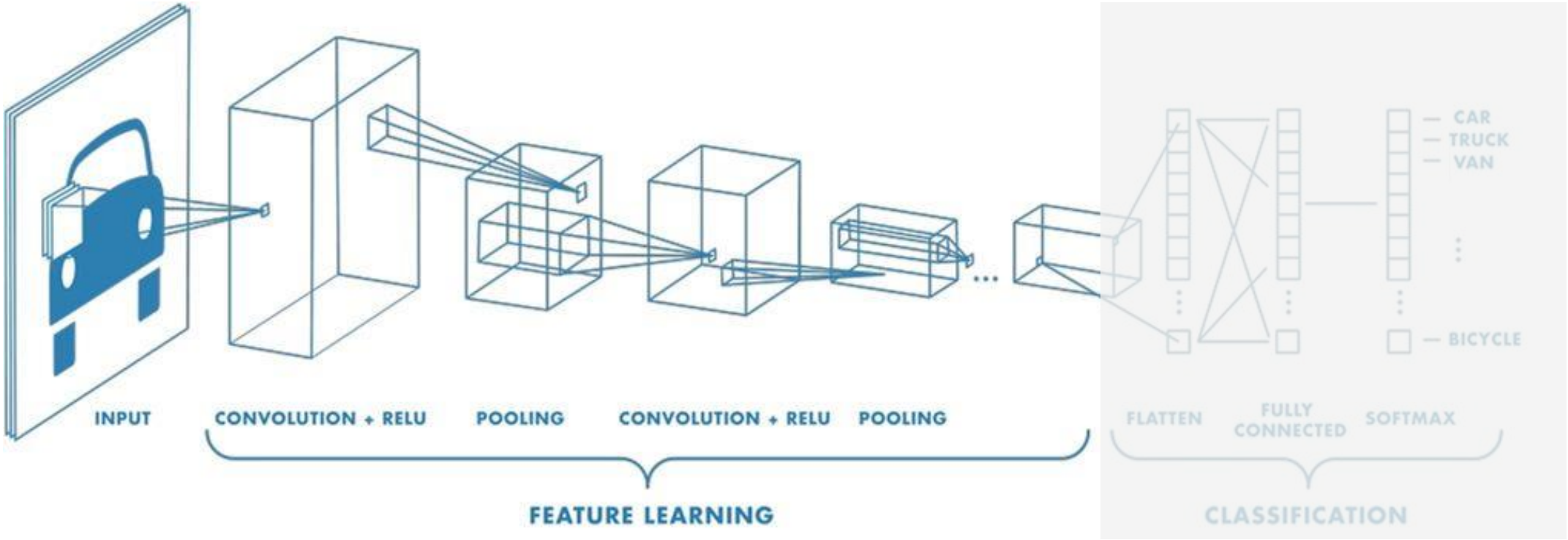
Conv Layer 2

Características de alto nivel:

Estructura facial

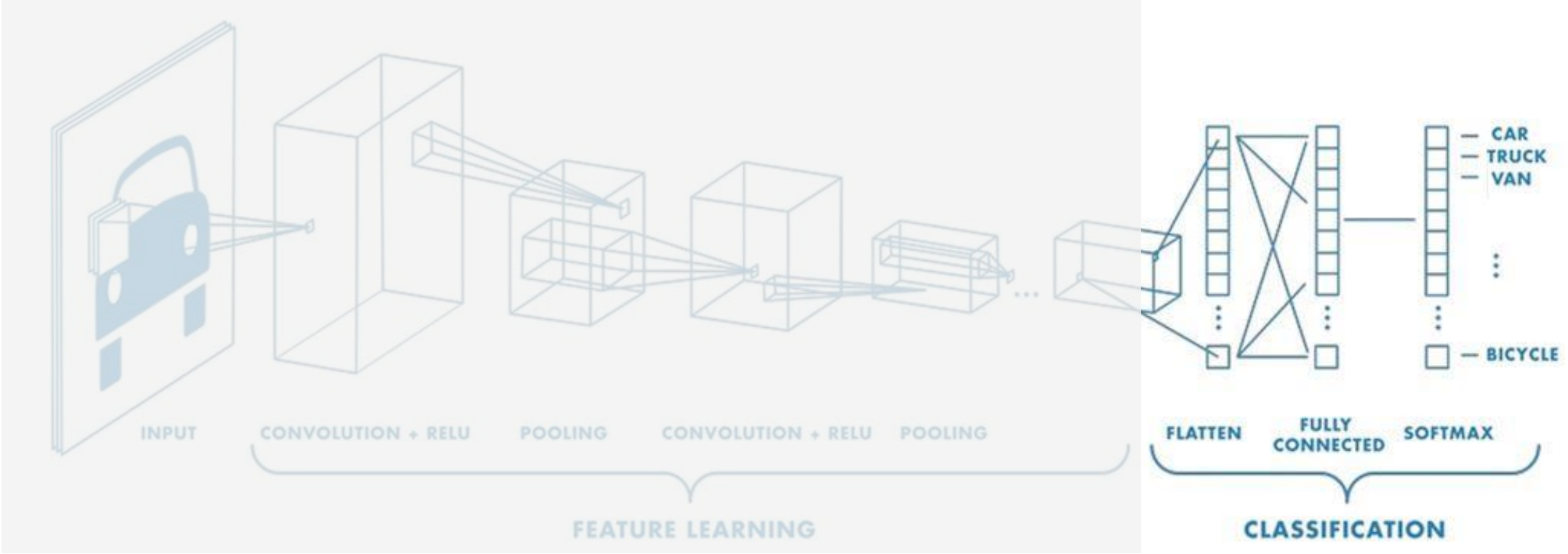
Conv Layer 3

CNNs para Clasificación: Aprendizaje de Características



1. Aprender características en la imagen de entrada mediante convolución
2. Introducir no linealidad mediante una función de activación (¡los datos del mundo real no son lineales!)
3. Reducir la dimensionalidad y preservar la invariancia espacial con pooling

CNNs para Clasificación: Probabilidades de Clase



- Las capas CONV y POOL generan características de alto nivel de la entrada.
- La capa totalmente conectada usa estas características para clasificar la imagen de entrada.
- El resultado se expresa como la probabilidad de que la imagen pertenezca a una clase en particular

$$\text{softmax}(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

Poniéndolo todo junto: CNN en TensorFlow

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

```python
import tensorflow as tf

def generate_model():
    model = tf.keras.Sequential([
        # first convolutional layer
        tf.keras.layers.Conv2D(32, filter_size=3, activation='relu'),
        tf.keras.layers.MaxPool2D(pool_size=2, strides=2),

        # second convolutional layer
        tf.keras.layers.Conv2D(64, filter_size=3, activation='relu'),
        tf.keras.layers.MaxPool2D(pool_size=2, strides=2),

        # fully connected classifier
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(1024, activation='relu'),
        tf.keras.layers.Dense(10, activation='softmax')   # 10 outputs
    ])
    return model
```

Poniéndolo todo junto: CNN en PyTorch

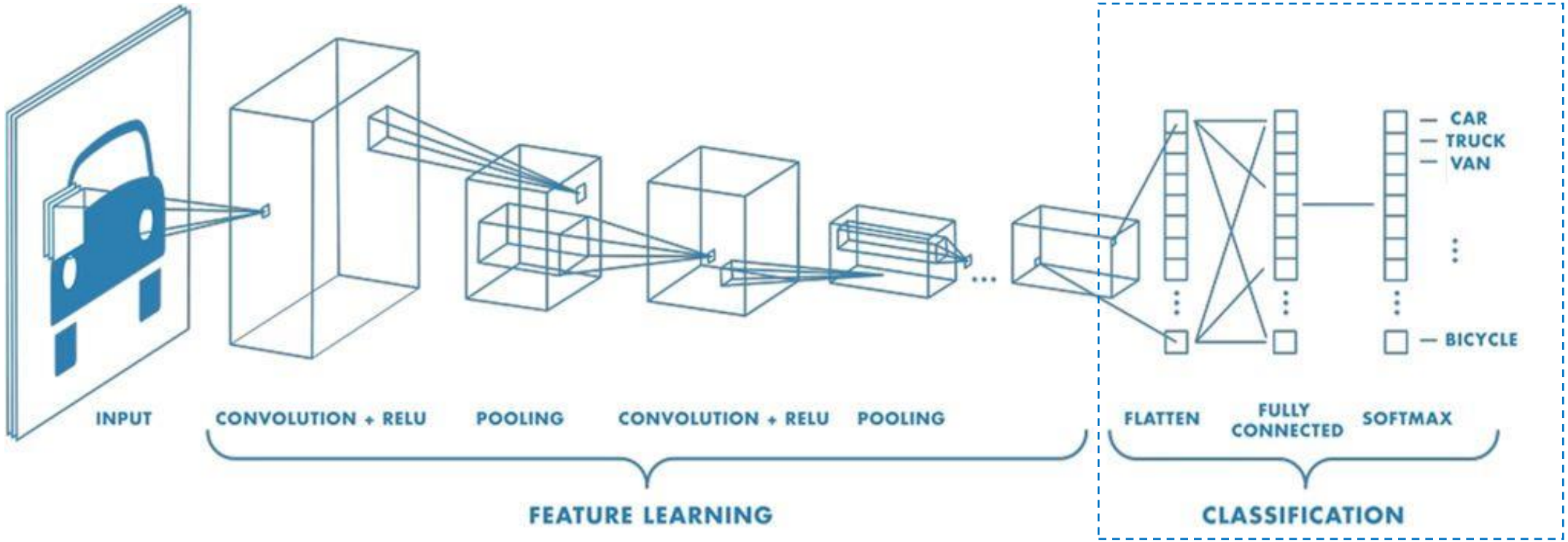
```python
import torch

def generate_model():
    model = nn.Sequential([
        # first and second convolutional layer
        torch.nn.Conv2d(in_channels=3,out_channels=32, filter_size=3),
        torch.nn.ReLU(),
        torch.nn.MaxPool2d(kernel_size=2, stride=2),

        torch.nn.Conv2d(in_channels=32, out_channels=64, filter_size=3),
        torch.nn.ReLU(),
        torch.nn.MaxPool2d(kernel_size=2, stride=2),

        # fully connected classifier
        torch.nn.Flatten(),
        torch.nn.Linear(64*6*6, 1024), # flattened dim after 2 conv layers
        torch.nn.ReLU(),
        torch.nn.Linear(1024, 10),  # 10 outputs
    ])
    return model
```

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería

Una Arquitectura para Muchas Aplicaciones

Una Arquitectura para Muchas Aplicaciones



Clasificación

Detección de objetos

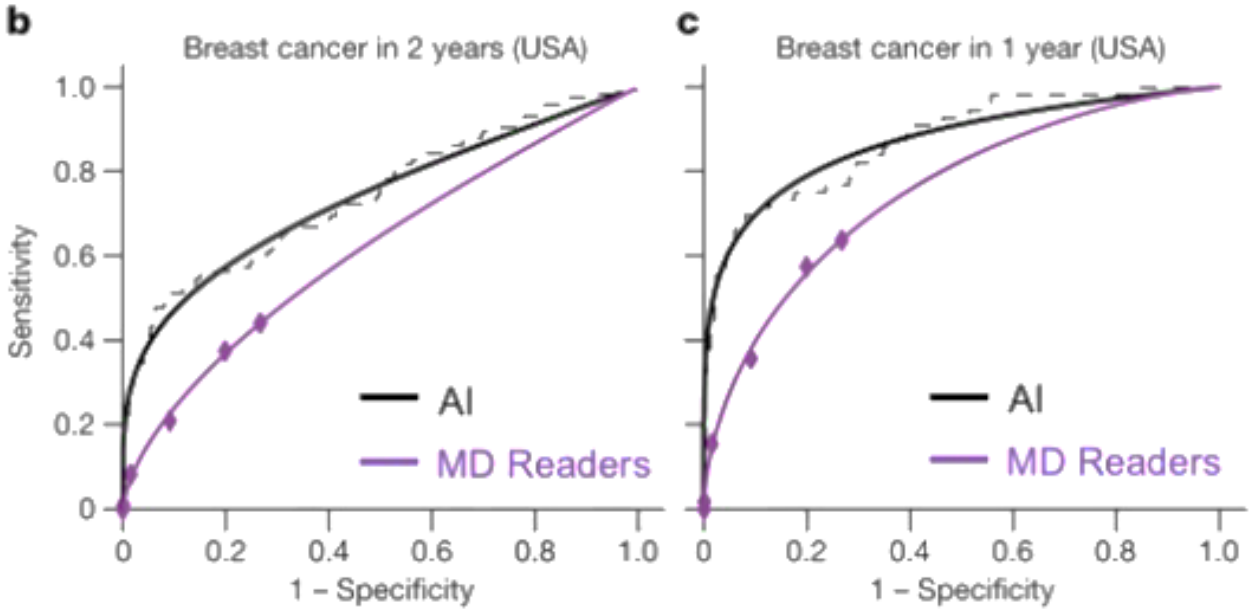
Segmentación

Control probabilístico

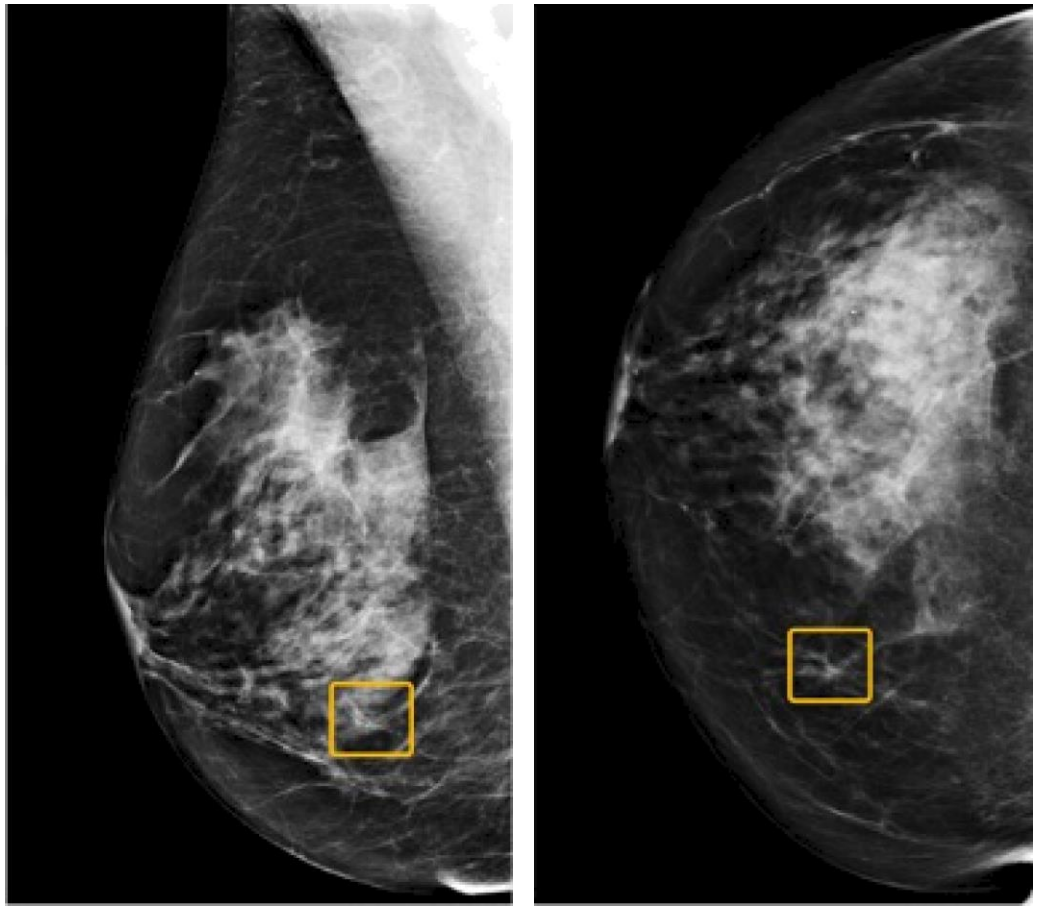
Clasificación: Detección de Cáncer de Mama

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Evaluación internacional de un sistema de IA para la detección de cáncer de mama

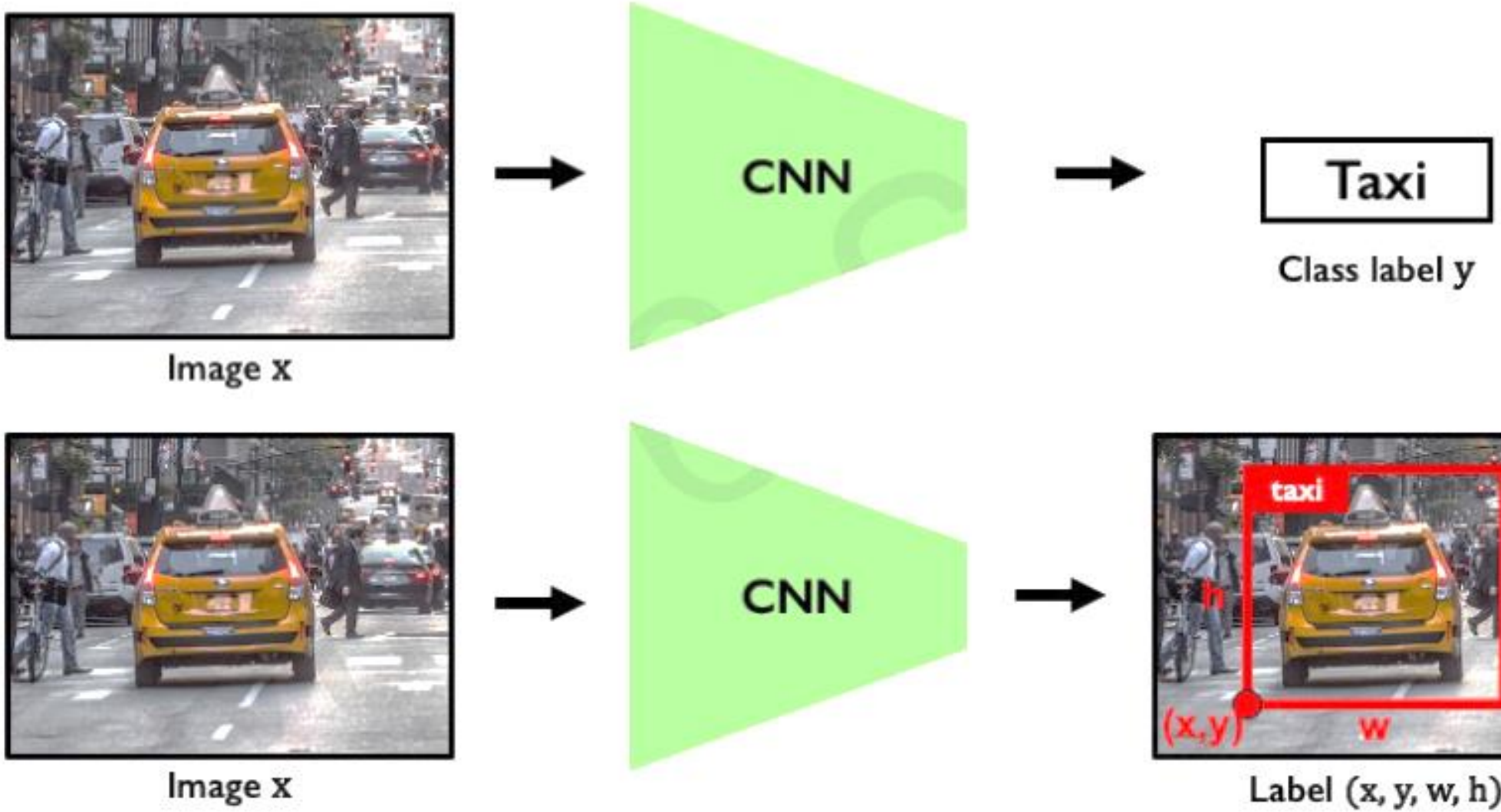


El sistema basado en CNN superó a radiólogos expertos en la detección de cáncer de mama a partir de mamografías.



Caso de cáncer de mama no detectado por el radiólogo pero sí detectado por la IA.

Detección de Objetos



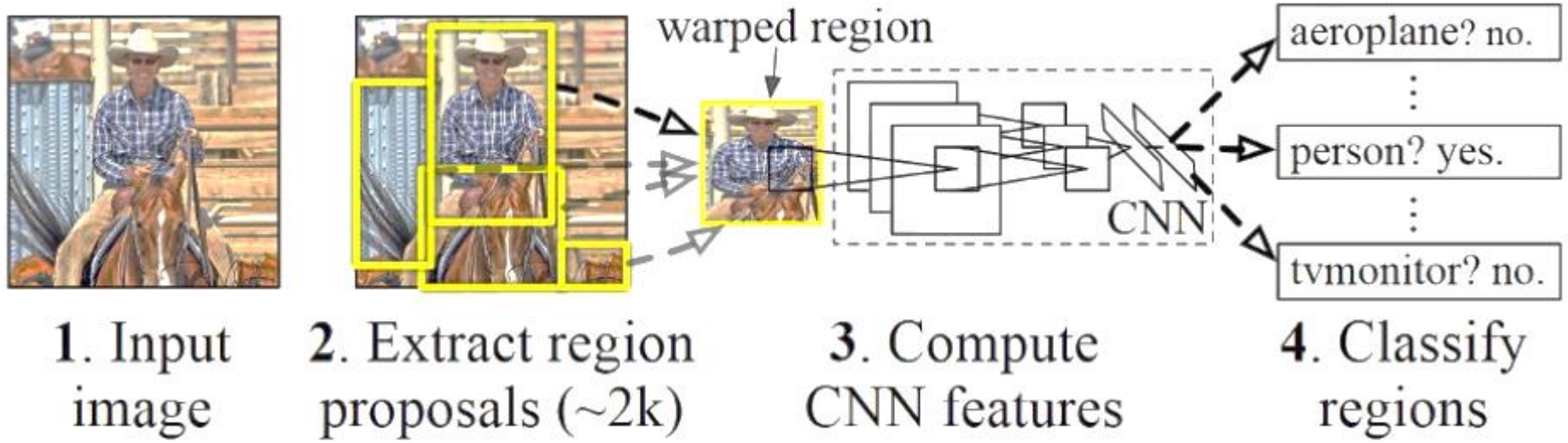
Detección de Objetos

Solución Ingenua para la Detección de Objetos

Problema: ¡Demasiadas entradas! Esto da como resultado demasiadas escalas, posiciones y tamaños

Detección de Objetos con R-CNNs

Algoritmo R-CNN: encontrar regiones que creemos que contienen objetos. Usar una CNN para clasificarlas.



Problemas:

1. ¡Lento! Muchas regiones; inferencia que consume mucho tiempo.
2. Frágil: propuestas de regiones definidas manualmente

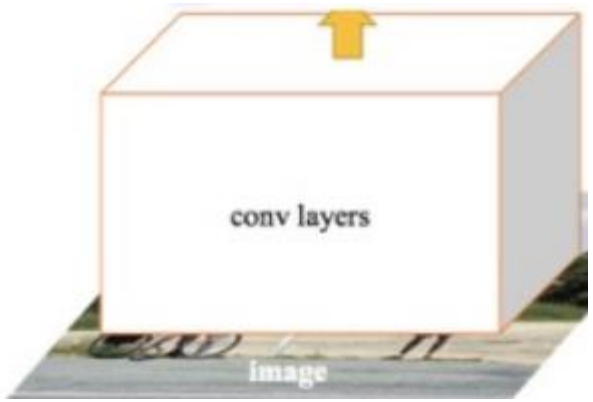
Faster R-CNN Aprende Propuestas de Regiones

Extracción de características sobre las regiones propuestas

Clasificación de regiones → detección de objetos

La imagen de entrada va directamente al extractor de características convolucional.

¡Rápido! Solo se ingresa la imagen una vez.



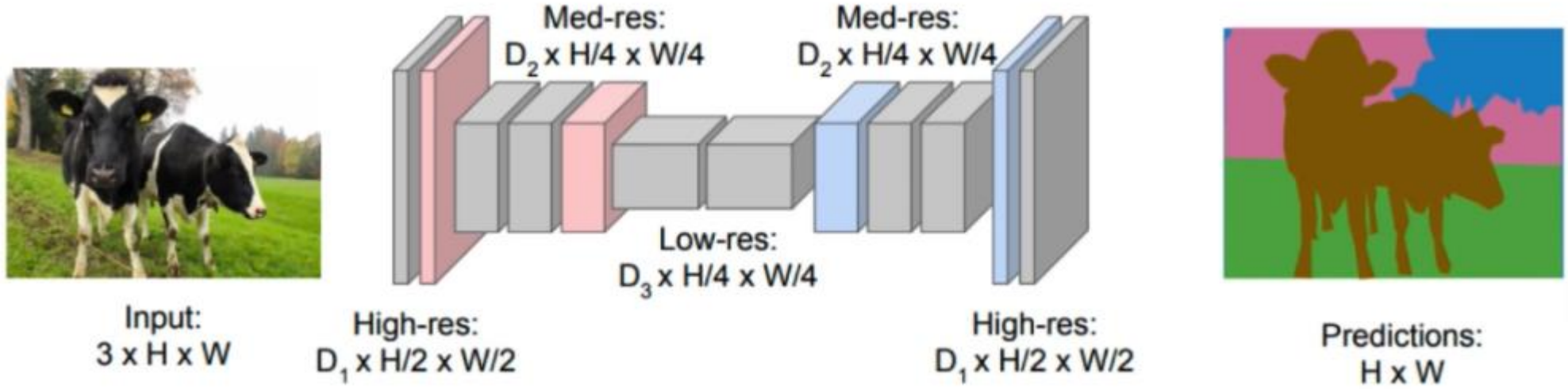
Red de propuestas de regiones para aprender regiones candidata Aprendido, basado en dato

Segmentación Semántica: Redes Totalmente Convolucionales

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

FCN: Fully Convolutional Network.

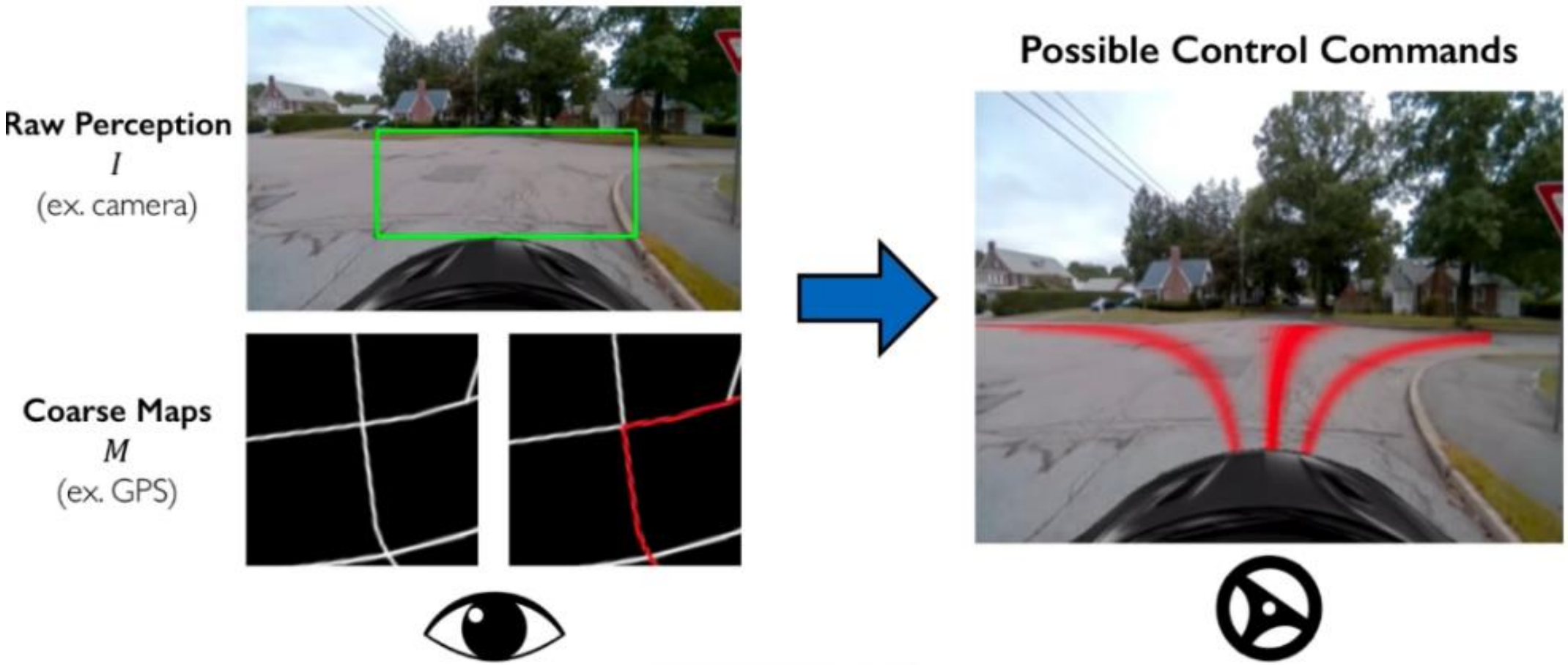
Red diseñada con todas las capas convolucionales, con operaciones de downsampling y upsampling.



`tf.keras.layers.Conv2DTranspose`

`torch.nn.ConvTranspose2d`

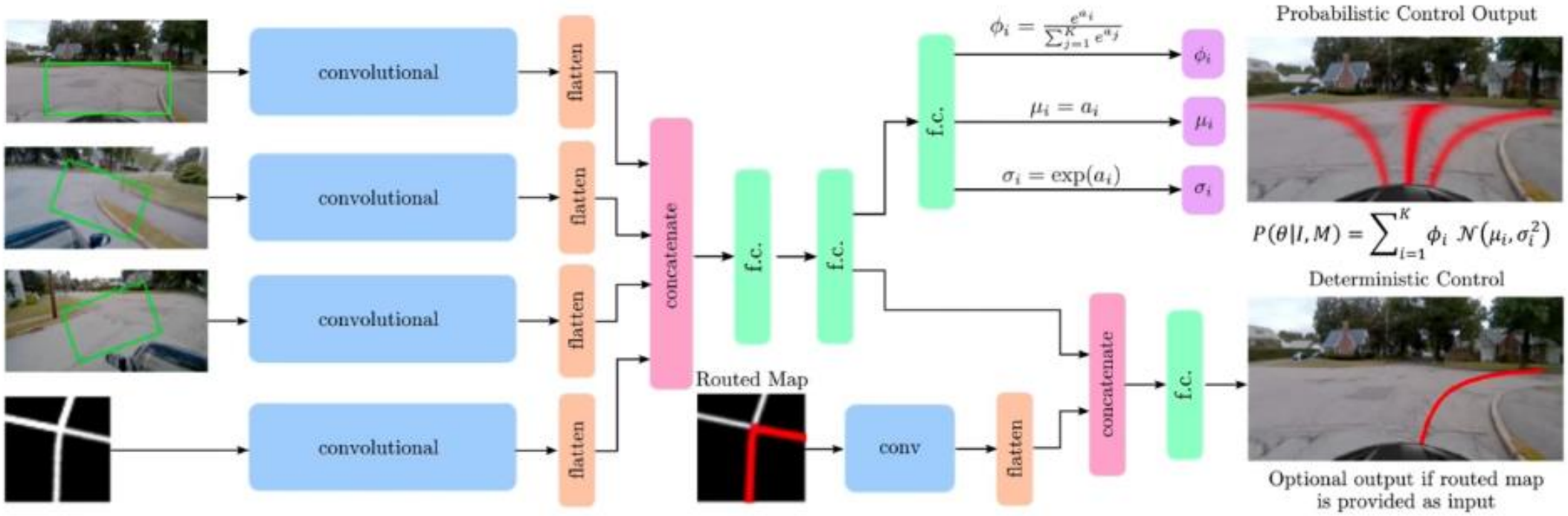
Control Continuo: Navegación a partir de la Visión



Marco de Extremo a Extremo para Navegación Autónoma

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Todo el modelo se entrena de extremo a extremo sin ninguna etiquetación o anotación humana.



$$L = -\log(P(\theta|I,M))$$


Auto ON
Navigation and Localization
Amini+ ICRA 2019.

Aprendizaje Profundo para Visión por Computador: Impacto

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Aprendizaje Profundo para Visión por Computador: Resumen

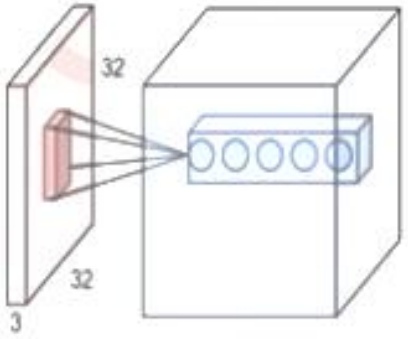
UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Fundamentos

- ¿Por qué visión por computador?
- Representación de imágenes
- Convoluciones para extracción de características

CNNs

- Arquitectura de CNN
- Aplicación a clasificación
- ImageNet



Aplicaciones

- Segmentación, descripción de imágenes, control
- Seguridad, medicina, robótica