

# Introducción a CNN y Reconocimiento de Emociones

## Diseño de Interfaces y UX

Dr. Hugo Arnoldo Mitre Hernández

CIMAT Zacatecas

Marzo 2026

- Interacción humano-máquina
- Monitoreo emocional en tiempo real
- Aplicaciones en sistemas embebidos

## Problema:

- Variabilidad facial
- Iluminación y oclusión
- Restricción computacional

# ¿Qué es una CNN?

## Idea clave

Las CNN aprenden características espaciales jerárquicas automáticamente.

- Convolución
- Extracción de características
- Clasificación

# Ecuación de la Convolución

$$S(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n)$$

Donde:

- $I(i + m, j + n)$ : valor del pixel en la imagen
- $K(m, n)$ : kernel o filtro
- $S(i, j)$ : salida (feature map)
- $\sum$ : suma sobre la región local

Interpretación:

- Detecta patrones locales (bordes, formas)
- Base del reconocimiento facial y emociones

# Convolución en imágenes de rostro

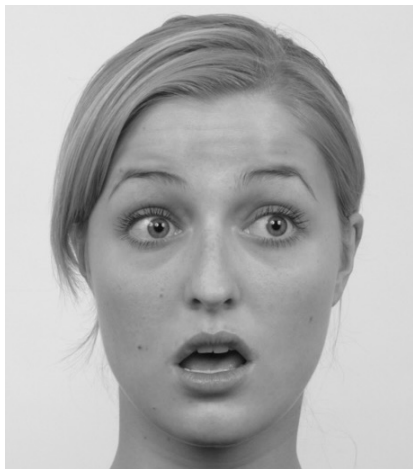


Imagen de entrada

Kernel (detector de bordes):

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Proceso:

- Se selecciona una región local (ej. ojo)
- Multiplicación elemento a elemento
- Suma de resultados
- Se genera un nuevo valor

# Convolución: cálculo paso a paso

Región de la imagen:

$$\begin{bmatrix} 52 & 55 & 61 \\ 62 & 59 & 55 \\ 63 & 65 & 66 \end{bmatrix}$$

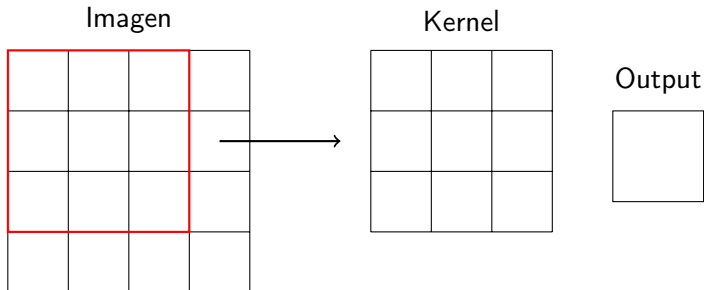
Kernel:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Resultado:

$$(-52 + 0 + 61) + (-62 + 0 + 55) + (-63 + 0 + 66) = 5$$

# Movimiento del kernel



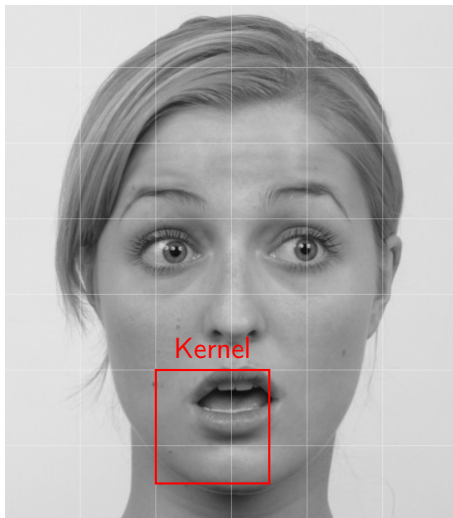
**El kernel recorre la imagen extrayendo características locales**

# Convolución: movimiento del kernel



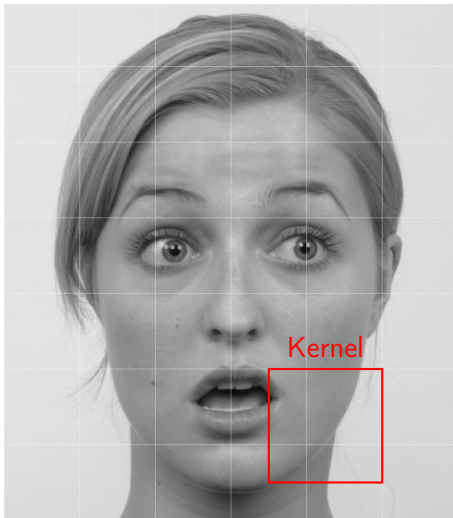
Paso 1: Se selecciona una región

# Convolución: movimiento del kernel



Paso 2: El kernel se desplaza

# Convolución: movimiento del kernel



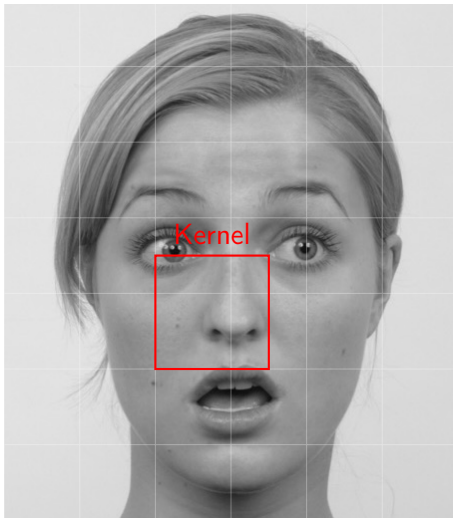
Paso 3: Se detectan patrones locales

# Convolución: movimiento del kernel



Paso 4: Se cubre otra zona del rostro

# Convolución: movimiento del kernel



Paso 5: Se construye el mapa de características

**Región de entrada**

$$\begin{bmatrix} 52 & 55 & 61 \\ 62 & 59 & 55 \\ 63 & 65 & 66 \end{bmatrix}$$

**Kernel**

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

## Cálculo

$$(-52 + 0 + 61) + (-62 + 0 + 55) + (-63 + 0 + 66)$$

$$= 5$$

## Salida parcial

$$\begin{bmatrix} 5 \\ \end{bmatrix}$$

Kernel se mueve → nuevo valor

$$\begin{bmatrix} 5 & 8 & 2 \end{bmatrix}$$

## Continuación

$$\begin{bmatrix} 5 & 8 & 2 \\ 10 & 12 & 6 \end{bmatrix}$$

## Mapa de características final

$$\begin{bmatrix} 5 & 8 & 2 \\ 10 & 12 & 6 \\ 7 & 9 & 4 \end{bmatrix}$$

### Interpretación:

- Valores altos  $\rightarrow$  bordes detectados
- Valores bajos  $\rightarrow$  regiones planas

# Convolución en CNN (imagen a color)

$$S_k(i, j) = \sum_{c=1}^C \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} I_c(i + m, j + n) W_{k,c}(m, n) + b_k$$

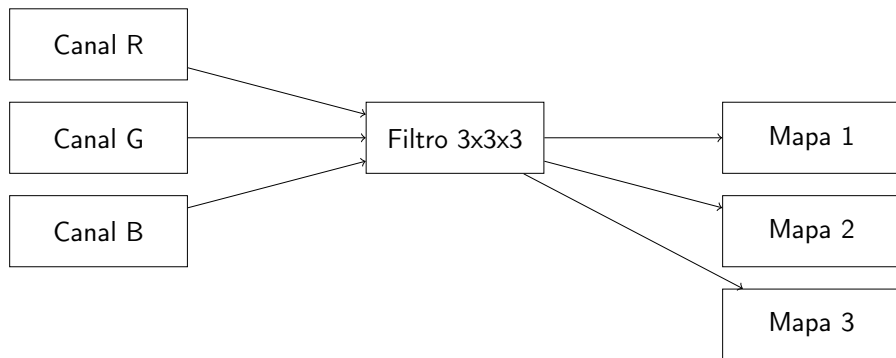
Donde:

- $I_c$ : imagen de entrada (canal  $c$ )
- $W_{k,c}$ : pesos del filtro  $k$
- $b_k$ : bias
- $S_k$ : feature map

Idea clave:

- Cada filtro aprende un patrón diferente del rostro

# Convolución en CNN (multicanal)



**Cada filtro procesa todos los canales y genera múltiples feature maps**

# Aplicación en reconocimiento de emociones

- Filtro 1 → bordes (ojos, boca)
- Filtro 2 → textura facial
- Filtro 3 → forma de sonrisa

## Resultado:

- Feature maps → representación del rostro
- Clasificación → emoción (feliz, enojo, triste)

# ¿Qué aprende la CNN en rostros?

- Bordes → ojos, boca, nariz
- Texturas → piel
- Formas → sonrisa, ceño

## Interpretación:

- Valores altos → presencia de rasgos faciales
- Valores bajos → zonas sin información relevante

## Resultado final:

- La red combina estos patrones para reconocer emociones

## Filtros clásicos:

- Bordes → ojos, boca, cejas
- Sobel → orientación de líneas
- Blur → reducción de ruido
- Sharpen → detalles finos
- Gabor → textura y orientación

## En CNN modernas:

- Los kernels se aprenden automáticamente
- Cada filtro detecta un patrón del rostro

# ¿Qué aprende una CNN para emociones?

- Capas iniciales → bordes
- Capas intermedias → ojos, boca
- Capas profundas → emociones

## Ejemplo:

- Sonrisa → curvatura de labios
- Enojo → cejas inclinadas
- Tristeza → caída de boca

## 1. Detectores de bordes

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Detecta contornos (ojos, boca)

## 2. Filtros Sobel

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Detecta orientación de líneas

## 3. Blur (suavizado)

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Reduce ruido

### Aplicación:

- Bordes → ojos, boca
- Líneas → cejas
- Suavizado → robustez

## 4. Sharpen (realce)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Resalta detalles finos

## 5. Gabor (textura)

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi fx)$$

Detecta orientación y textura

## Aplicación en emociones:

- Textura → arrugas (enojo)
- Formas → sonrisa (felicidad)
- Detalles → microexpresiones

### Idea clave

Estos filtros inspiran lo que aprende una CNN automáticamente

# ¿Los kernels se diseñan o se aprenden?

## Filtros clásicos (visión por computadora)

- Diseñados manualmente
- Ejemplos:
  - Sobel
  - Edge detection
  - Gabor
- Detectan patrones específicos

## CNN (ej. Light CNN-9)

- Kernels aprendidos automáticamente
- Ajustados mediante entrenamiento
- Optimizados para la tarea (emociones)

### Limitación

No se adaptan automáticamente a los datos

### Ventaja

Aprenden patrones complejos del rostro

## Idea clave:

La red aprende automáticamente qué características son importantes.

# Gracias