

# Análisis de Redes Convolucionales y Técnicas de Explicabilidad para Arquitecturas Neuronales Profundas para la Segmentación de Instancias

## Trabajo de Fin de Grado

**Autor:** Juan Manuel Rodríguez Gómez

**Tutores:** Francisco Javier Merí de la Maza, Pablo Mesejo Santiago,  
Marilyn Bello García

24 de Junio de 2025

Doble Grado en Ingeniería Informática y Matemáticas



UNIVERSIDAD  
DE GRANADA

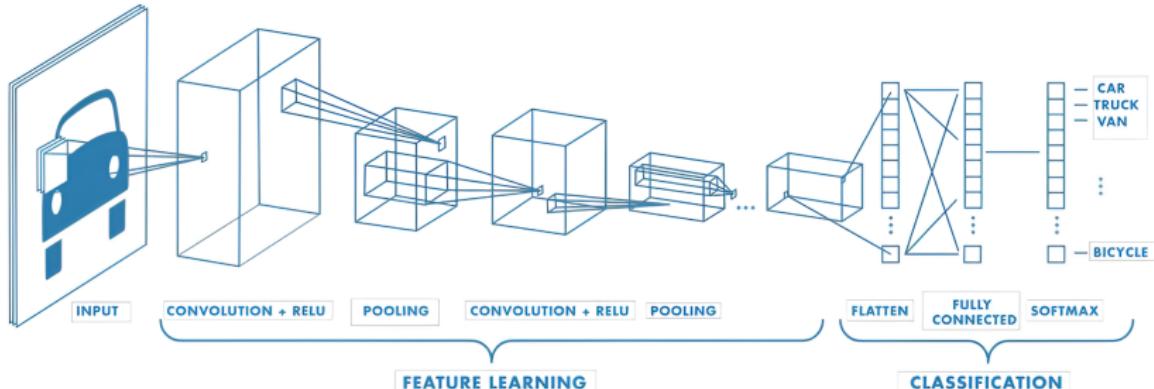
# Primera Parte: Análisis de Redes Convolucionales

- ① Introducción
- ② Teoría de Mallat sobre Redes de Dispersión
- ③ Generalización de la Teoría de Mallat

## 1 Introducción

## Redes neuronales convolucionales

- Gran rendimiento empírico, con bases teóricas aún en desarrollo.
  - Buscamos profundizar en su **modelización matemática**.

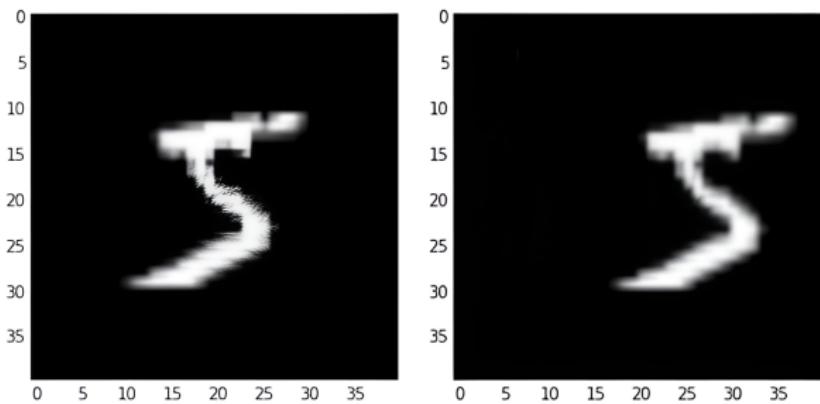


**Figura 1:** Diagrama esquemático de la arquitectura de una red neuronal convolucional. Imagen extraída de [Mat24b].

[Mat24b] MathWorks. Redes neuronales convolucionales. <https://es.mathworks.com/discovery/convolutional-neural-network.html>, 2024.

## Primera propiedad: Invarianza por traslaciones

- Identificar un objeto sin importar dónde esté en la imagen.



**Figura 2:** Ejemplo de invarianza por traslaciones: el dígito 5 se reconoce correctamente pese a su cambio de posición. Imagen extraída de [KA18].

---

[KA18] E. Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. CoRR, abs/1801.01450, 2018.

## Primera propiedad: Invarianza por traslaciones

## Invarianza por translaciones

Sea  $(T_t f)(x) := f(x - t)$  la traslación de  $f \in L^2(\mathbb{R}^d)$  por  $t \in \mathbb{R}^d$ .

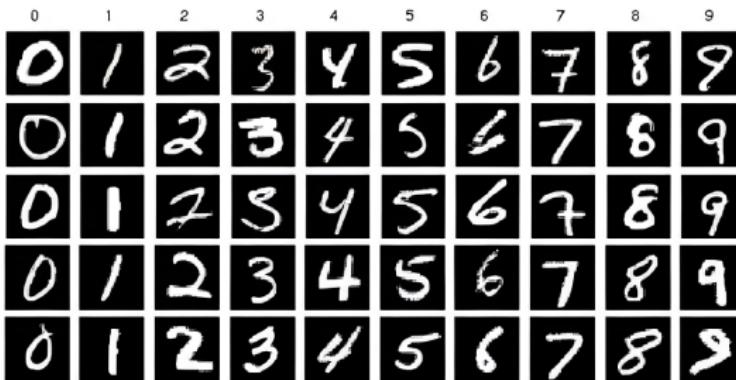
Sea  $\mathcal{H}$  un espacio de Hilbert. Decimos que un operador

$\Phi : L^2(\mathbb{R}^d) \rightarrow \mathcal{H}$  es *invariante por traslaciones* si:

$$\Phi(T_t f) = \Phi(f), \quad \forall f \in L^2(\mathbb{R}^d), \quad \forall t \in \mathbb{R}^d.$$

## Segunda propiedad: Invarianza frente a pequeñas deformaciones

- Identificar un objeto a pesar de cambios menores en su forma.



**Figura 3:** Ejemplo de invarianza a deformaciones: los dígitos se reconocen correctamente pese a variaciones en su forma. Imagen extraída de [LYP16].

---

[LYP16] S. Lim, S. Young, y R. Patton. An analysis of image storage systems for scalable training of deep neural networks. ResearchGate, 2016.

## Segunda propiedad: Invarianza frente a pequeñas deformaciones

### Acción de difeomorfismos cercanos a traslaciones

Sea  $\tau : \mathbb{R}^d \rightarrow \mathbb{R}^d$  un campo de desplazamiento. El difeomorfismo  $\mathbb{1} - \tau$  transforma un punto  $x \in \mathbb{R}^d$  en  $x - \tau(x)$ . Definimos entonces la *acción del difeomorfismo*  $\mathbb{1} - \tau$  sobre una función  $f \in L^2(\mathbb{R}^d)$  como  $(T_\tau f)(x) := f(x - \tau(x))$ .

### Invarianza frente a pequeñas deformaciones

Un operador invariante por traslaciones  $\Phi : L^2(\mathbb{R}^d) \rightarrow \mathcal{H}$  se dice *Lipschitz-continuo* por la acción de difeomorfismos  $C^2$  si existe una constante  $c > 0$  tal que, para todo  $f \in L^2(\mathbb{R}^d)$  y para todo  $\tau \in C^2(\mathbb{R}^d)$ , se cumple:

$$\|\Phi(f) - \Phi(T_\tau f)\| \leq c\|f\| (\|Jac(\tau)\|_\infty + \|Hess(\tau)\|_\infty).$$

## Objetivo

Desarrollar un operador  $\Phi$  que sea invariante por translaciones y frente a pequeñas deformaciones, y que modelice el comportamiento de una red neuronal convolucional.

## 1 Introducción

## 2 Teoría de Mallat sobre Redes de Dispersión

## 3 Generalización de la Teoría de Mallat

## Limitaciones de la transformada de Fourier

## Transformada de Fourier

Sea  $f \in L^1(\mathbb{R}^d)$ . Se define la *transformada de Fourier* de  $f$  como:

$$\widehat{f}(\omega) := \int_{\mathbb{R}^d} f(x) e^{-2\pi i \langle x, \omega \rangle} dx = \int_{\mathbb{R}^d} f(x) [\cos(2\pi \langle x, \omega \rangle) - i \sin(2\pi \langle x, \omega \rangle)] dx$$

## Módulo de la transformada de Fourier

Sea  $\mathcal{H}$  un espacio de Hilbert. Definimos el operador  $\Phi : L^2(\mathbb{R}^d) \rightarrow \mathcal{H}$  como  $\Phi(f) := |\widehat{f}|$ .

Este operador  $\Phi$ :

- Es invariante por traslaciones  $|\widehat{T_t f}| = |\widehat{f}|$ .
  - No es invariante frente a pequeñas deformaciones  $\Rightarrow$   
**Debemos buscar otro operador.**

# Ondículas como alternativa a la transformada de Fourier

En nuestro desarrollo, haremos uso de una ondícula madre del siguiente tipo:

$$\psi(x) = e^{i\langle x, \eta \rangle} \Theta(x)$$

donde  $\Theta(x)$  es una función real con soporte en una bola de baja frecuencia centrada en  $x = 0$  y con radio del orden de  $\pi$ , y  $\eta$  es un vector que define la frecuencia central de la ondícula.

## Escalado y rotación de una ondícula madre

Una *ondícula madre escalada por un factor  $2^j$* , con  $j \in \mathbb{Z}$ , y *rotada por  $r \in G$* , siendo  $G$  el grupo finito de rotaciones, se define como:

$$\psi_{2^j r}(x) = 2^j \psi(2^j r^{-1} x).$$

# Transformada de Littlewood-Paley

Con dicha ondícula madre, generamos la siguiente base ortonormal de ondículas:

$$\{\psi_\lambda(x)\}_{\lambda=2^j r \in 2^{\mathbb{Z}} \times G}$$

## Transformada de Littlewood-Paley

Sea  $f \in L^2(\mathbb{R}^d)$  y sea  $\{\psi_\lambda\}_{\lambda \in 2^{\mathbb{Z}} \times G}$  una familia de ondeletas generadas por escalado y rotación de una ondícula madre. Para cada  $\lambda = 2^j r$ , definimos la *transformada de Littlewood-Paley* de  $f$  como la convolución de  $f$  con la ondícula  $\psi_\lambda$ :

$$(W[\lambda]f)(x) := (f * \psi_\lambda)(x) = \int_{\mathbb{R}^d} f(u) \psi_\lambda(x - u) du, \quad \forall x \in \mathbb{R}^d.$$

El operador  $W[\lambda]f$  es Lipschitz-continuo bajo la acción de difeomorfismos pero **no es invariante por traslaciones**.

## Problema con la transformada de Littlewood-Paley: La escala



**Figura 4:** Ondículas a menor escala detectan mejor los cambios rápidos en la señal; a mayor escala, captan variaciones más globales. Imagen extraída de [Mat24a].

A una cierta escala  $2^J$ , con  $J \in \mathbb{Z}$  fijo, solo se conservan las ondículas afectadas por un factor de escala  $2^j > 2^{-J}$ .

---

[Mat24a] MathWorks. Continuous wavelet transform and scale-based analysis. <https://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html>, 2024.

## Solución: Promediado de bajas frecuencias

Esto genera un umbral a partir del cual la base ortonormal de ondículas no podría identificar cambios de frecuencia.

Surge la necesidad de promediar las bajas frecuencias que no son captadas por estas ondículas en un dominio proporcional al factor  $2^J$ :

$$A_J f = f * \phi_{2^J}, \quad \text{con} \quad \phi_{2^J}(x) = 2^{-J} \phi(2^{-J}x).$$

Así, la transformada de ondículas, fijada una escala, se expresaría de la siguiente forma:

$$W_J f = \left\{ A_J f, (W[\lambda]f)_{\lambda \in \Lambda_J} \right\}$$

con  $\Lambda_J = \{\lambda = 2^j r : r \in G^+, 2^j > 2^{-J}\}$ , siendo  $G^+$  el cociente de  $G$  sobre  $\{-\mathbb{1}, \mathbb{1}\}$ .

## Convenciones para el posterior desarrollo de este trabajo

- Trabajaremos en el espacio de funciones reales de  $L^2(\mathbb{R}^d)$ .
- $\widehat{\psi}$  es una función real que satisface  $\widehat{\psi}(0) = \int \psi(x) dx = 0$  y  $|\widehat{\psi}(r\omega)| = |\widehat{\psi}(\omega)|$ ,  $\forall r \in G \Rightarrow W_J$  es unitario (preserva la norma y el producto escalar).
- $\widehat{\phi}(\omega)$  es real y simétrica, por lo que  $\phi$  también lo es.
- Las derivadas de  $\phi$  pertenecen a  $L^1(\mathbb{R}^d)$ .

## Propagador de dispersión

Condición para obtener un operador invariante por traslaciones

Si  $U[\lambda]$  es un operador definido en  $L^2(\mathbb{R}^d)$ , no necesariamente lineal pero que conmuta con traslaciones, entonces  $\int_{\mathbb{R}^d} (U[\lambda]f)(x) dx$  es invariante por traslaciones si es finito.

$W[\lambda]f = f * \psi_\lambda$  es un ejemplo trivial que cumple la condición anterior. Pero  $\int_{\mathbb{R}^d} \psi(x) dx = 0 \Rightarrow \int_{\mathbb{R}^d} (f * \psi)(x) dx = 0$ .

Definimos así el operador:

$$U[\lambda]f = M[\lambda](W[\lambda]f) = |f * \psi_\lambda|,$$

siendo  $M[\lambda]$  lo que llamamos *operador de demodulación*.

# Propagador de dispersión

## Camino

Un conjunto de elementos ordenados  $p = (\lambda_1, \lambda_2, \dots, \lambda_m)$  con  $\lambda_k \in \Lambda_\infty = 2^{\mathbb{Z}} \times G^+$  se conoce como *camino*. El camino vacío se representa por  $p = \emptyset$ .

## Propagador de dispersión

Un *propagador de dispersión* es un producto por un camino ordenado de operadores no conmutativos de la forma

$U[\lambda]f = M[\lambda](W[\lambda]f) = |f * \psi_\lambda|$ , con  $f \in L^2(\mathbb{R}^d)$ . Es decir:

$$U[p]f = U[\lambda_m] \cdots U[\lambda_2] U[\lambda_1]f = |||f * \psi_{\lambda_1}| * \psi_{\lambda_2}| \cdots * \psi_{\lambda_m}|,$$

siendo  $U[\emptyset] = \text{Id}$ .

Tenemos que  $\int_{\mathbb{R}^d} (U[p]f)(x) dx < \infty$ .

# Transformada de dispersión de ventana

## Transformada de dispersión de ventana

Sea  $J \in \mathbb{Z}$  y  $\mathcal{P}_J$  el conjunto de caminos finitos  $p = (\lambda_1, \lambda_2, \dots, \lambda_m)$ , con  $\lambda_k \in \Lambda_J$  y  $|\lambda_k| = 2^k > 2^{-J}$ . Definimos una *transformada de dispersión de ventana* para todo  $p \in \mathcal{P}_J$  como:

$$(S_J[p]f)(x) = ((U[p]f) * \phi_{2^J})(x),$$

donde la convolución con  $\phi_{2^J}$  localiza el propagador de dispersión en dominios proporcionales a  $2^J$ . También podemos escribir:

$$(S_J[p]f)(x) = (|||f * \psi_{\lambda_1}| * \psi_{\lambda_2}| \cdots * \psi_{\lambda_m}| * \phi_{2^J})(x).$$

En particular,  $S_J[\emptyset]f = f * \phi_{2^J}$ .

# Comparativa con una red neuronal convolucional

- **Similitudes:**
  - Convolución: operador  $W[\lambda]$ .
  - Función de activación: operador  $M[\lambda]$  (módulo).
  - Si  $p$  tiene longitud  $m$ : los coeficientes  $S_J[p]f$  corresponden a los mapas de activación tras la capa  $m$  de la red.
- **Diferencias:**
  - En las redes neuronales convoluciones, los pesos se aprenden por retropropagación. En esta modelización, las ondículas son fijas y no se aprenden.
  - No se incluyen capas de pooling en esta modelización.

# Ondículas admisibles

## Ondículas admisibles

Una ondícula de dispersión  $\psi$  se considera *admissible* si existe un valor  $\eta \in \mathbb{R}^d$  y una función  $\rho \geq 0$ , con  $|\widehat{\rho}(\omega)| \leq |\widehat{\phi}(2\omega)|$  y  $\widehat{\rho}(0) = 1$ , tal que la función

$$\widehat{\Psi}(\omega) = |\widehat{\rho}(\omega - \eta)|^2 - \sum_{k=1}^{\infty} k \left( 1 - \left| \widehat{\rho}(2^{-k}(\omega - \eta)) \right|^2 \right),$$

satisface:

$$\alpha = \inf_{1 \leq |\omega| \leq 2} \sum_{j=-\infty}^{\infty} \sum_{r \in G} \widehat{\Psi}(2^{-j} r^{-1} \omega) \left| \widehat{\psi}(2^{-j} r^{-1} \omega) \right|^2 > 0.$$

## Propiedades de la transformada de dispersión de ventana

### Conservación de la norma

Si la ondícula es admisible, entonces, para todo  $f \in L^2(\mathbb{R}^d)$ , se cumple que:

$$\lim_{m \rightarrow \infty} \sum_{n=m}^{\infty} \|S_J[\Lambda_J^n]f\|^2 = 0,$$

y:

$$\|S_J[\mathcal{P}_J]f\| = \|f\|.$$

### No-expansividad en conjuntos de caminos

Para todo  $f, h \in L^2(\mathbb{R}^d)$  y  $J \in \mathbb{Z}$  se cumple que:

$$\|S_{J+1}[\mathcal{P}_{J+1}]f - S_{J+1}[\mathcal{P}_{J+1}]h\| \leq \|S_J[\mathcal{P}_J]f - S_J[\mathcal{P}_J]h\|.$$

Invarianza horizontal por traslaciones del operador  $S_j[p]f$

## Invarianza horizontal por traslaciones del operador $S_j[p]f$

Para ondículas de dispersión admisibles, se verifica que:

$$\forall f \in L^2(\mathbb{R}^d), \quad \forall t \in \mathbb{R}^d, \quad \lim_{J \rightarrow \infty} \|S_J[\mathcal{P}_J]f - S_J[\mathcal{P}_J](T_t f)\| = 0.$$

# Invarianza frente a pequeñas deformaciones del operador $S_J[p]f$

## Invarianza frente a pequeñas deformaciones del operador $S_J[p]f$

Existe una constante  $C \in \mathbb{R}$  tal que, para todo  $f \in L^2(\mathbb{R}^d)$ , con  $\|U[\mathcal{P}_J]f\|_1 < \infty$ , y todo  $\tau \in C^2(\mathbb{R}^d)$ , con  $\|\text{Jac}(\tau)\|_\infty \leq \frac{1}{2}$ , se cumple que:

$$\|S_J[\mathcal{P}_J](T_\tau f) - S_J[\mathcal{P}_J]f\| \leq C\|U[\mathcal{P}_J]f\|_1 K(\tau),$$

con:

$$K(\tau) = 2^{-J}\|\tau\|_\infty + \|\text{Jac}(\tau)\|_\infty \left( \log \left( \frac{\|\Delta\tau\|_\infty}{\|\text{Jac}(\tau)\|_\infty} \vee 1 \right) \right) + \|\text{Hess}(\tau)\|_\infty,$$

y para todo  $m \geq 0$ :

$$\|S_J[\mathcal{P}_{J,m}](T_\tau f) - S_J[\mathcal{P}_{J,m}]f\| \leq Cm\|f\|K(\tau),$$

siendo  $\mathcal{P}_{J,m}$  el subconjunto de  $\mathcal{P}_J$  de caminos de longitud estrictamente menor que  $m$ .

## 1 Introducción

## 2 Teoría de Mallat sobre Redes de Dispersión

## 3 Generalización de la Teoría de Mallat

## Generalización de la teoría de Mallat

Buscamos extender la teoría de Mallat sobre redes de dispersión para abarcar arquitecturas más generales:

- Incluir funciones de activación distintas del módulo (ReLU, sigmoide...).
  - Considerar diferentes tipos de filtros (no solo ondículas, también curvículas, crestículas...).
  - Modelizar también las capas de *pooling*.

## Resumen de la teoría de Mallat sobre redes de dispersión

## Vector de características de una red de dispersión

El vector de características  $\Phi_W(f)$  de una señal  $f \in L^2(\mathbb{R}^d)$  de dispersión es:

$$\Phi_W(f) := \bigcup_{n=0}^{\infty} \Phi_W^n(f), \quad \text{con } \Phi_W^0(f) := \{f * \psi_{(-J,0)}\},$$

$$\Phi_W^n(f) := \{(U[\lambda_1, \dots, \lambda_n]f) * \psi_{(-J, 0)}\}_{\lambda_1, \dots, \lambda_n \in \Lambda_W \setminus \{(-J, 0)\}},$$

donde cada  $\lambda = (j, k) \in \Lambda_W := \{(-J, 0)\} \cup \{(j, k) \mid j \in \mathbb{Z}, j > -J, k \in \{0, \dots, K-1\}\}$  representa un par escala-dirección, y:

$$U[\lambda_1, \dots, \lambda_n]f := |\cdots| |f * \psi_{\lambda_1}| * \psi_{\lambda_2} | \cdots * \psi_{\lambda_n}|$$

## Marcos semi-discretos

### Marco semi-discreto para $L^2(\mathbb{R}^d)$

Sea  $\{g_\lambda\}_{\lambda \in \Lambda} \subset L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$  un conjunto de funciones indexado por un conjunto numerable  $\Lambda$ . La colección:

$$\Psi_\Lambda := \{T_b(Ig_\lambda)\}_{b \in \mathbb{R}^d, \lambda \in \Lambda}$$

es un *marco semi-discreto* para  $L^2(\mathbb{R}^d)$  si existen constantes  $A, B > 0$  tales que:

$$A\|f\|^2 \leq \sum_{\lambda \in \Lambda} \int_{\mathbb{R}^d} |\langle f, T_b(Ig_\lambda) \rangle|^2 db = \sum_{\lambda \in \Lambda} \|f * g_\lambda\|^2 \leq B\|f\|^2, \quad \forall f \in L^2(\mathbb{R}^d),$$

donde  $\langle f, g \rangle := \int_{\mathbb{R}^d} f(x)\overline{g(x)}dx$  e  $(If)(x) := \overline{f(-x)}$  es la involución.

Las funciones  $\{g_\lambda\}_{\lambda \in \Lambda}$  se denominan *átomos* del marco  $\Psi_\Lambda$ , y los valores  $\langle f, T_b(Ig_\lambda) \rangle = (f * g_\lambda)(b)$  se llaman *coeficientes* del marco  $\Psi_\Lambda$ . Si  $A = B$ , el marco se denomina *marco ajustado*. Un marco ajustado con  $A = 1$  se llama *marco de Parseval*.

# Operador de *pooling*

## Operador de *pooling*

Sea  $M_n(f * g_{\lambda_n})$  el resultado de aplicar una no-linealidad Lipschitz-continua sobre una convolución en la capa n-ésima. Esta se puede someter a un proceso de *pooling* definido como:

$$f \mapsto S_n^{d/2} P_n(f)(S_n \cdot), \quad \text{con } f \in L^2(\mathbb{R}^d),$$

donde:

- $S_n \geq 1$  es el factor de *pooling*.
- $P_n$  es un operador Lipschitz-continuo que satisface que  $P_n f = 0$  cuando  $f = 0$ .

# Operador buscado: Extractor de características convolucional general

## Sucesión de módulos

Para  $n \in \mathbb{N}$ , sea  $\Psi_n = \{T_b(\lg_{\lambda_n})\}_{b \in \mathbb{R}^d, \lambda_n \in \Lambda_n}$  un marco semi-discreto para  $L^2(\mathbb{R}^d)$ , y sean  $M_n : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$  y  $P_n : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$  operadores Lipschitz-continuos con  $M_n f = 0$  y  $P_n f = 0$  para  $f = 0$ , respectivamente. Entonces, la sucesión de ternas:

$$\Omega := ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}},$$

se denomina *sucesión de módulos*.

# Operador buscado: Extractor de características convolucional general

## Operador $U_n$ y caminos

Sea  $\Omega = ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}}$  una sucesión de módulos. Sean  $\{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$  los átomos del marco  $\Psi_n$ , y sea  $S_n \geq 1$  el factor de *pooling* asociado a la capa  $n$ -ésima de la red.

Definimos el operador  $U_n$  asociado a la capa  $n$ -ésima de la red como:

$$U_n(\lambda_n, f) := U_n[\lambda_n]f := S_n^{d/2} P_n(M_n(f * g_{\lambda_n})) (S_n \cdot).$$

Para  $n \in \mathbb{N}$ , definimos el conjunto  $\Lambda_1^n := \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_n$ . Un conjunto ordenado  $q = (\lambda_1, \lambda_2, \dots, \lambda_n) \in \Lambda_1^n$  se llama un *camino*.

Para el camino vacío, establecemos  $\Lambda_1^0 := \{\emptyset\}$  y

$U_0[\emptyset]f := f$ , para todo  $f \in L^2(\mathbb{R}^d)$ .

## Operador buscado: Extractor de características convolucional general

## Extractor de características convolucional general

Sea  $\Omega = ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}}$  una sucesión de módulos. El *extractor de características*  $\Phi_\Omega$  basado en  $\Omega$  lleva  $L^2(\mathbb{R}^d)$  a su vector de características:

$$\Phi_\Omega(f) := \bigcup_{n=0}^{\infty} \Phi_\Omega^n(f),$$

donde:

$$\Phi_\Omega^n(f) := \{(U[q]f) * \chi_n\}_{q \in \Lambda_1^n}, \quad \forall n \in \mathbb{N},$$

siendo  $\chi_n$  el *átomo generador de salida* de la capa  $n$ .

# Condición de admisibilidad para sucesiones de módulos

## Sucesión de módulos admisible

Sea  $\Omega = ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}}$  una sucesión de módulos, con cotas superiores del marco  $B_n > 0$  y constantes de Lipschitz  $L_n, R_n > 0$  para los operadores  $M_n$  y  $P_n$ , respectivamente.

La condición:

$$\max \{B_n, B_n L_n^2 R_n^2\} \leq 1, \quad \forall n \in \mathbb{N},$$

se denomina *condición de admisibilidad para sucesiones de módulos*. Las sucesiones de módulos que satisfacen esta condición se denominan *admisibles*.

## Invarianza vertical por translaciones del operador $\Phi_\Omega$

Invarianza vertical por translaciones del operador  $\Phi_\Omega$

Sea  $\Omega = ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}}$  una sucesión de módulos admissible, sea  $S_n \geq 1$ ,  $n \in \mathbb{N}$ , el factor de *pooling*, y supongamos que los operadores  $M_n, P_n : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$  comutan con el operador de traslación  $T_t$ . Entonces:

- 1 Las características  $\Phi_{\Omega}^n(f)$  generadas en la capa  $n$ -ésima de la red satisfacen:

$$\Phi_{\Omega}^n(T_t f) = T_{t/(S_1 \dots S_n)}(\Phi_{\Omega}^n(f)), \quad \forall f \in L^2(\mathbb{R}^d), \quad \forall t \in \mathbb{R}^d, \forall n \in \mathbb{N}.$$

- 2 Si además existe una constante  $K > 0$  (que no depende de  $n$ ) tal que las transformadas de Fourier  $\widehat{\chi}_n$  de los átomos generadores de salida  $\chi_n$  satisfacen la *condición de decaimiento*:

$$|\widehat{\chi}_n(\omega)| |\omega| \leq K, \quad \text{c.p.d. } \omega \in \mathbb{R}^d, \quad \forall n \in \mathbb{N}_0,$$

entonces:

$$\|\Phi_{\Omega}^n(T_t f) - \Phi_{\Omega}^n(f)\| \leq \frac{2\pi|t|K}{S_1 \cdots S_n} \|f\|, \quad \forall f \in L^2(\mathbb{R}^d), \quad \forall t \in \mathbb{R}^d.$$

# Invarianza vertical por traslaciones del operador $\Phi_\Omega$

## Impacto del *pooling* en la invarianza vertical por traslaciones

Sea  $\Omega = ((\Psi_n, M_n, P_n))_{n \in \mathbb{N}}$  una sucesión de módulos admisible, sea  $S_n \geq 1$ ,  $n \in \mathbb{N}$ , el factor de *pooling*, y supongamos que los operadores  $M_n, P_n : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d)$  conmutan con el operador de traslación  $T_t$ .

Si, además, existe una constante  $K > 0$  (que no depende de  $n$ ) tal que las transformadas de Fourier  $\widehat{\chi}_n$  de los átomos generadores de salida  $\chi_n$  satisfacen la condición de decaimiento, entonces:

$$\|\Phi_\Omega^n(T_t f) - T_t(\Phi_\Omega^n(f))\| \leq \frac{2\pi|t|K|1 - (S_1 \cdots S_n)|}{|S_1 \cdots S_n|} \|f\|,$$

para todo  $f \in L^2(\mathbb{R}^d)$  y  $t \in \mathbb{R}^d$ .

# Segunda Parte: Técnicas de Explicabilidad para Arquitecturas Neuronales Profundas para la Segmentación de Instancias

- ① Introducción
- ② Fundamentos Teóricos
- ③ Estado del Arte
- ④ Experimentación
- ⑤ Conclusiones

## 1 Introducción

## 2 Fundamentos Teóricos

### ③ Estado del Arte

## 4 Experimentación

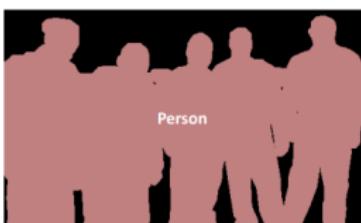
## 5 Conclusiones

## Contexto y motivación

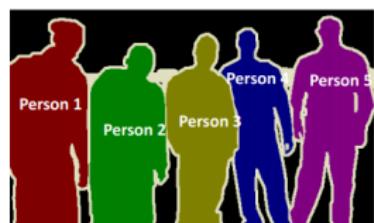
- **Segmentación de instancias:** Tarea de Visión por Computador que consiste en identificar y delimitar cada objeto individual en una imagen, proporcionando una máscara específica para cada instancia detectada.



## Object Detection



## Semantic Segmentation



## Instance Segmentation

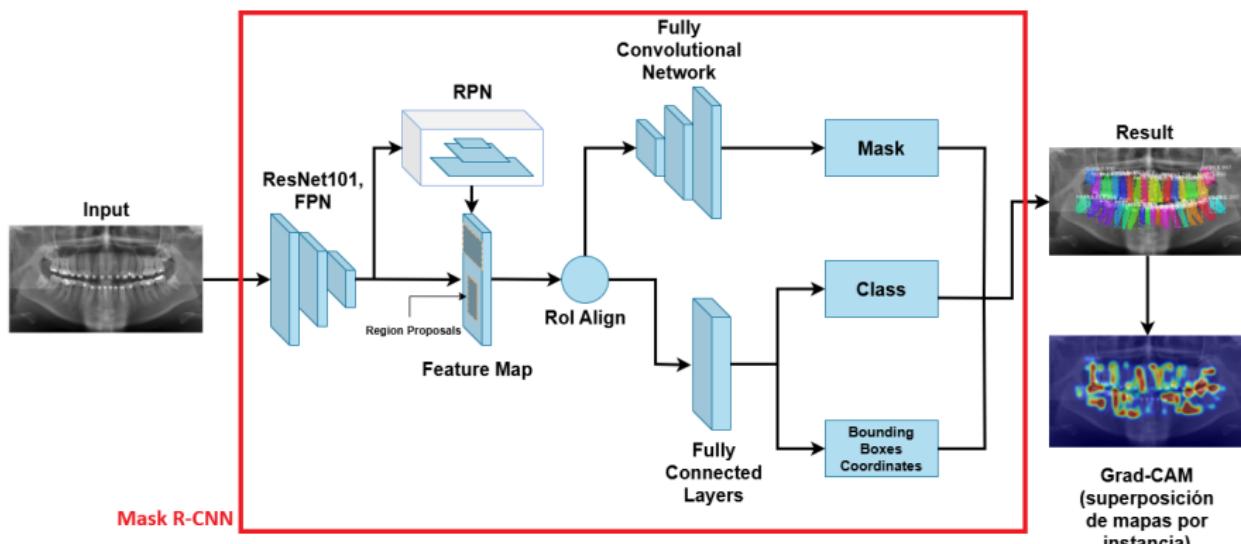
**Figura 1:** La segmentación de instancias combina la detección de objetos y la segmentación semántica. Imagen extraída de [Liu18].

[Liu18] Y. Liu. The confusing metrics of AP and MAP for object detection / instance segmentation. <https://yanfengliu.medium.com/the-confusing-metrics-of-ap-and-map-for-object-detection-3113ba0386ef>, 2018.

## Contexto y motivación

- Mask R-CNN es una de las arquitecturas más efectivas para esta tarea.
  - Sin embargo, no explica por qué segmenta así, comportándose como una **caja negra** y dificultando así la comprensión de sus decisiones.
  - **Solución:** Adaptar la técnica de explicabilidad Grad-CAM para visualizar qué partes de la imagen han influido en la predicción.

# Contexto y motivación



**Figura 2:** Resumen del enfoque propuesto: Segmentación de dientes con Mask R-CNN y explicación visual mediante Grad-CAM aplicado a cada instancia detectada.

# Objetivos

- ① Estudiar la **segmentación de instancias** y el funcionamiento de Mask R-CNN.
- ② Aplicar Mask R-CNN a **radiografías dentales panorámicas**.
- ③ Investigar el campo de la **Inteligencia Artificial Explicable**.
- ④ Implementar una **adaptación de Grad-CAM** en Mask R-CNN.
- ⑤ **Evaluar** la efectividad de Grad-CAM en Mask R-CNN.

## 1 Introducción

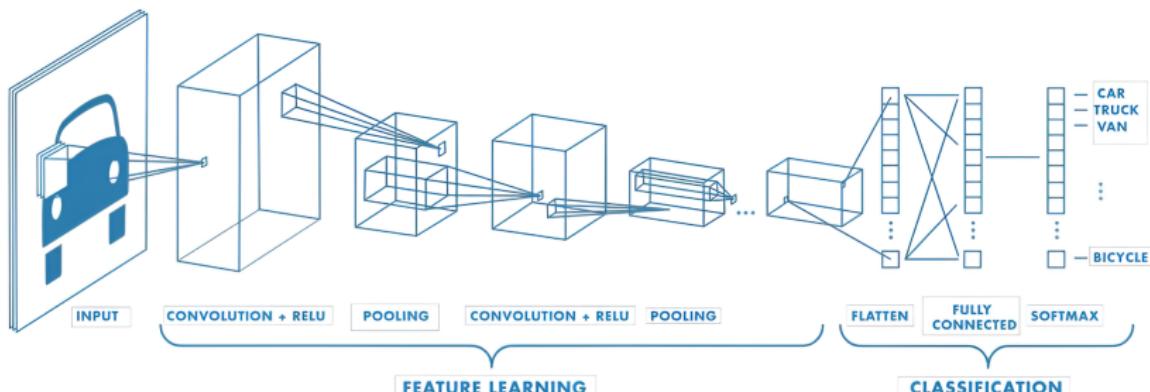
## 2 Fundamentos Teóricos

## 3 Estado del Arte

## 4 Experimentación

## 5 Conclusiones

## Redes neuronales convolucionales



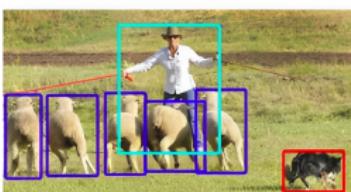
**Figura 3:** Diagrama esquemático de la arquitectura de una red neuronal convolucional. Imagen extraída de [Mat24b].

[Mat24b] MathWorks. Redes neuronales convolucionales. <https://es.mathworks.com/discovery/convolutional-neural-network.html>. 2024.

# Tareas de Visión por Computador



image classification



object detection



semantic segmentation

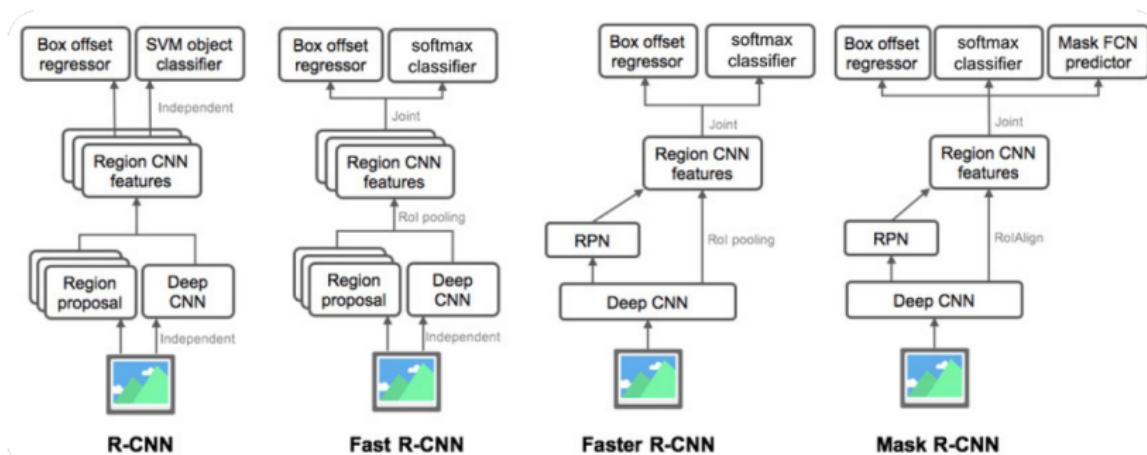


instance segmentation

**Figura 4:** Diferentes tareas de Visión por Computador aplicadas en una misma escena. Imagen extraída de [LOW+19].

[LOW+19] L. Liu, W. Ouyang, X. Wang, P.W. Fieguth, J. Chen, X. Liu, y M. Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2019.

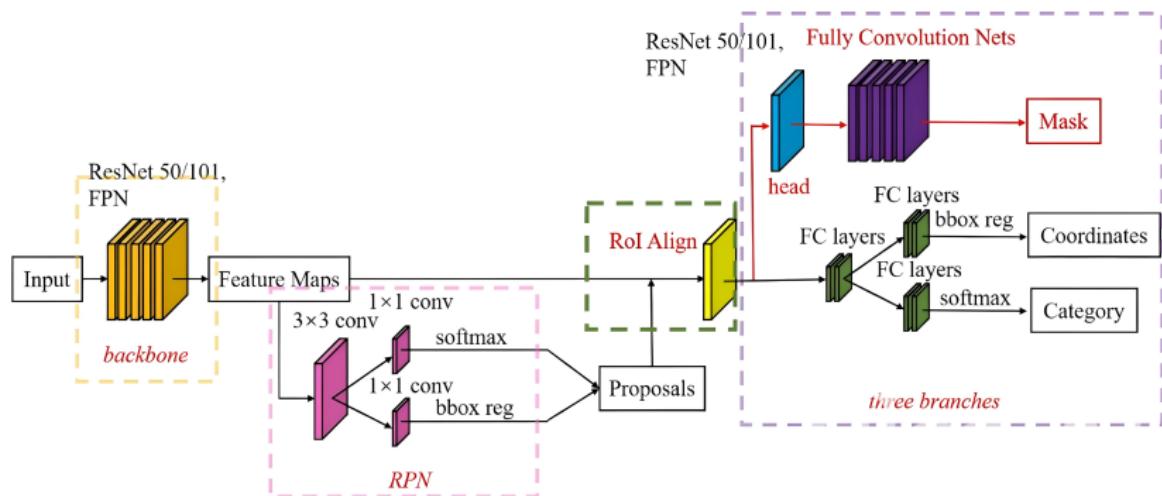
# Arquitecturas R-CNN



**Figura 5:** Evolución de la familia de arquitecturas R-CNN: desde R-CNN hasta Mask R-CNN. Imagen extraída de [Vin24].

[Vin24] A. Vina. ¿Qué es R-CNN? Un resumen rápido. <https://www.ultralytics.com/es/blog/what-is-r-cnn-a-quick-overview>, 2024.

# Mask R-CNN



**Figura 6:** Arquitectura de Mask R-CNN. Imagen extraída de [SKY23].

---

[SKY23] SKY ENGINE AI. What is Mask R-CNN? <https://skyengine.ai/se/skyengine-blog/119-what-is-mask-r-cnn>, 2023.

# Inteligencia Artificial Explicable (XAI)

- La **XAI** busca hacer comprensibles y confiables los modelos de IA, especialmente los de caja negra como las redes neuronales.
- Permite **entender cómo y por qué el modelo toma decisiones**, algo clave en ámbitos críticos (medicina, justicia, finanzas).
- **Tipos de explicabilidad:**
  - Según el momento en el que se aplica:
    - **Intrínseca:** El modelo es explicable por sí mismo (árboles de decisión).
    - **Post-hoc:** Se explica un modelo ya entrenado (Grad-CAM).
  - Según el alcance de las explicaciones que proporcionan:
    - **Global:** Explica todo el modelo.
    - **Local:** Explica predicciones concretas.
- **Propiedades deseables:** Comprensibilidad, fidelidad, estabilidad, justicia.

# Grad-CAM

- **Grad-CAM** (*Gradient-weighted Class Activation Mapping*) es una técnica de interpretabilidad post-hoc para redes neuronales convolucionales.
- Permite generar **mapas de calor visuales** que muestran las regiones más relevantes de una imagen para una clase específica.
- Se basa en los **gradientes** de la clase objetivo respecto a los mapas de características de una capa convolucional.
- No requiere modificar ni reentrenar el modelo.

# Grad-CAM

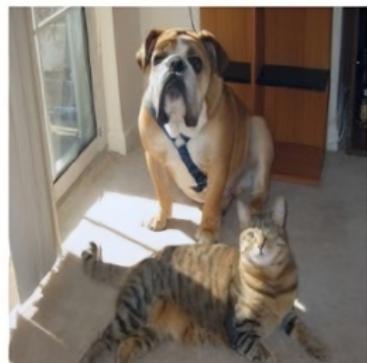
- **Funcionamiento:**

- ① Se elige una capa convolucional profunda y se obtienen sus **activaciones**  $A^k$ .
- ② Se calculan los **gradientes** de la clase objetivo  $y^c$  respecto a cada  $A^k$ .
- ③ Se realiza un **promedio global** de los gradientes para obtener los pesos  $\alpha_k^c$ .
- ④ Se combinan los mapas  $A^k$  con sus pesos  $\alpha_k^c$  y se aplica una **ReLU**:

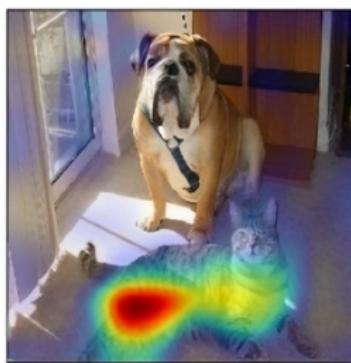
$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left( \sum_k \alpha_k^c A^k \right)$$

- ⑤ Se interpola y **superpone** el mapa de calor sobre la imagen original.

# Grad-CAM



Original Image



Grad-CAM 'Cat'



Grad-CAM 'Dog'

**Figura 7:** Visualización de mapas de calor generados por Grad-CAM para una imagen de entrada que contiene un perro y un gato. Imagen extraída de [SCD+17].

---

[SCD+17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, y D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, páginas 618–626, 2017.

## 1 Introducción

## 2 Fundamentos Teóricos

## 3 Estado del Arte

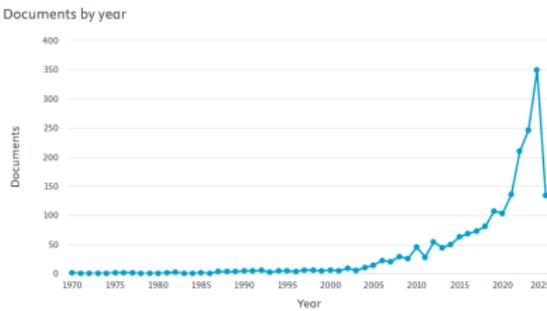
## 4 Experimentación

## 5 Conclusiones

Búsqueda Scopus - Segmentación dental con Mask R-CNN

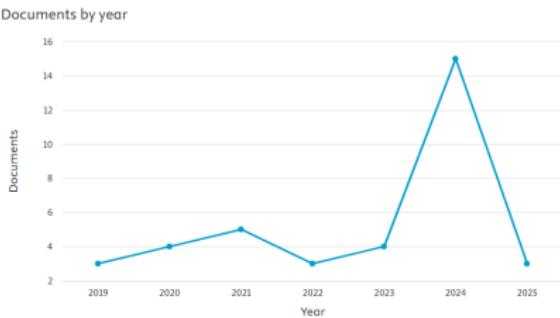
## Segmentación dental:

- 1992 publicaciones.
  - Crecimiento desde 2015  
(auge del Deep Learning).



## Usando Mask R-CNN:

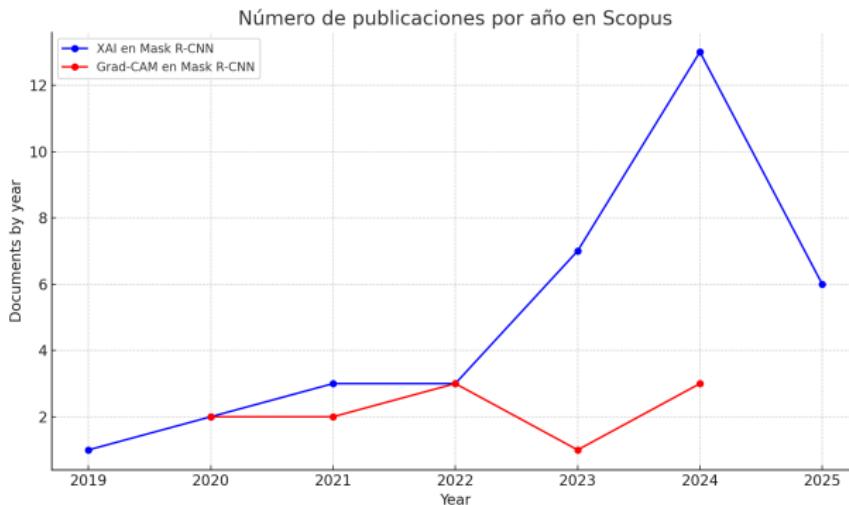
- Solo 37 publicaciones.
  - Línea de investigación reciente.
  - Resultados prometedores.



# Búsqueda Scopus - XAI aplicada a Mask R-CNN

## Explicabilidad en Mask R-CNN:

- 35 publicaciones en total sobre XAI aplicada a Mask R-CNN.
- Solo 11 aplican **Grad-CAM**.
- Línea emergente, con alto potencial de desarrollo.



## 1 Introducción

## 2 Fundamentos Teóricos

## 3 Estado del Arte

## 4 Experimentación

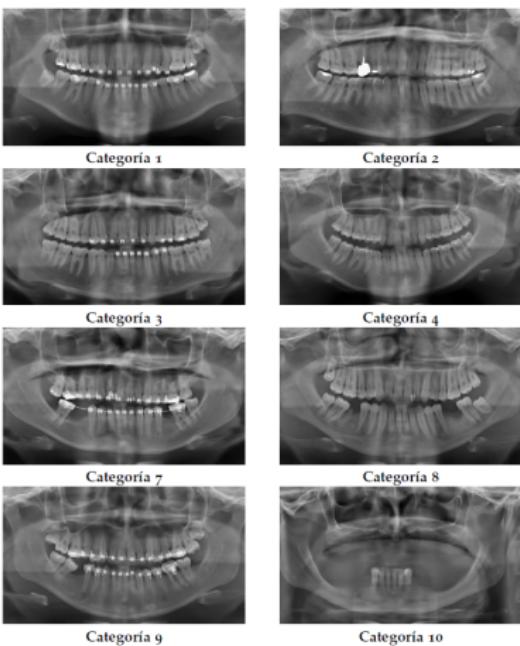
## 5 Conclusiones

# Conjunto de datos DNS Panoramic Images

- **543 imágenes** de radiografías dentales panorámicas, junto con su respectiva máscara dental.
  - **Diversidad:** dientes completos, restaurados, con aparatos o faltantes.
  - **Partición:** 376 entrenamiento (70 %), 87 validación (15 %) y 80 test (15 %), distribuidas equilibradamente por categoría.

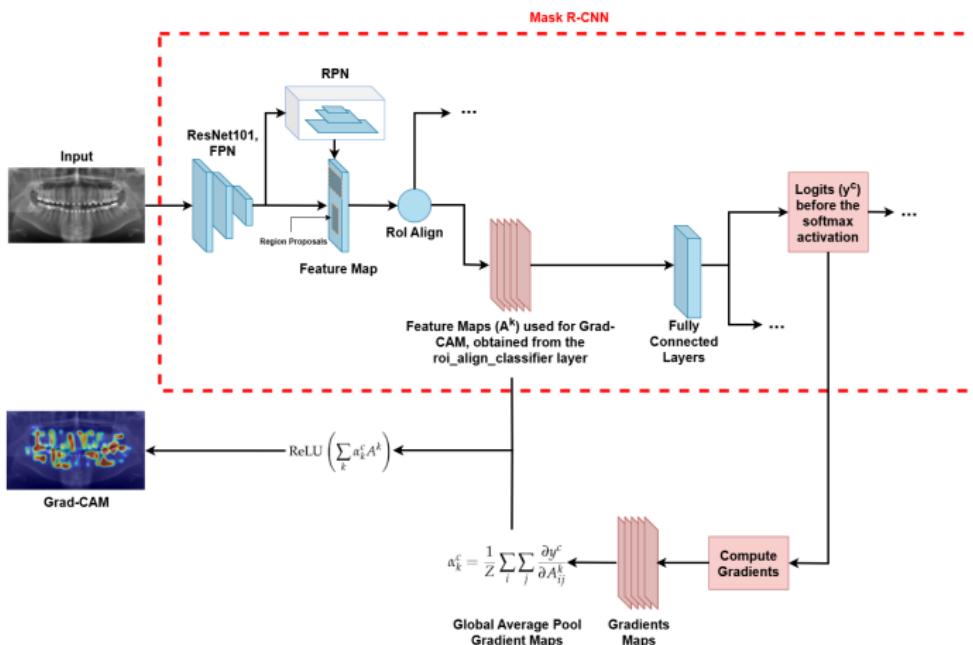


# Conjunto de datos DNS Panoramic Images



**Figura 8:** Ejemplos de radiografías dentales panorámicas según las categorías del conjunto de datos DNS Panoramic Images.

# Adaptación de Grad-CAM a Mask R-CNN



**Figura 9:** Esquema del proceso de adaptación de Grad-CAM a la arquitectura Mask RCNN.

## Métricas empleadas: Detección

- **Precision:** Porcentaje de predicciones que son correctas.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** Porcentaje de objetos reales correctamente detectados.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:** Media armónica entre Precision y Recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **AP@50:** Área bajo la curva Precision-Recall con  $\text{IoU} \geq 0.5$ .

$$\text{AP} = \int_0^1 P(R) dR$$

## Métricas empleadas: Segmentación

- **DSC:** Evalúa la superposición entre la máscara predicha y la real.

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

- **IoU:** Proporción de solapamiento entre las dos máscaras.

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

- **Hausdorff Distance:** Máxima distancia entre los bordes predicho y real.

$$HD(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(b, a) \right\}$$

## Métricas empleadas: Explicabilidad

- **Deletion:** Mide cuánto disminuye la confianza del modelo al eliminar píxeles relevantes.

$$\text{Deletion} = \int_0^1 p_c(t) dt$$

- **Insertion:** Mide cuánto aumenta la confianza del modelo al introducir píxeles relevantes.

$$\text{Insertion} = \int_0^1 p_c(t) dt$$

- **Stability:** Correlación entre mapas Grad-CAM de imágenes originales y perturbadas.

$$\text{Stability}(x) = \rho(M_x, M_{x+\delta})$$

- **IoU<sub>Grad-CAM</sub>:** Solapamiento entre atención y segmentación.

$$\text{IoU}_{\text{Grad-CAM}} = \frac{|A \cap B|}{|A \cup B|}$$

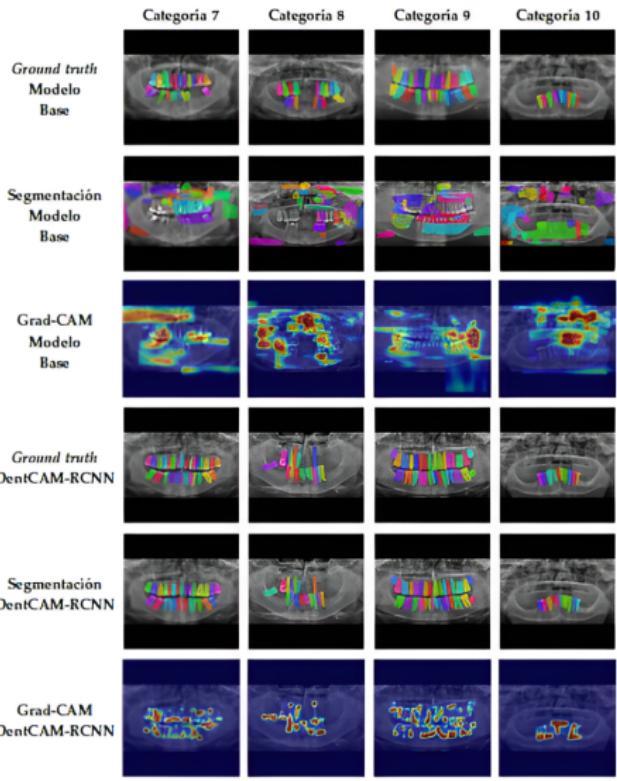
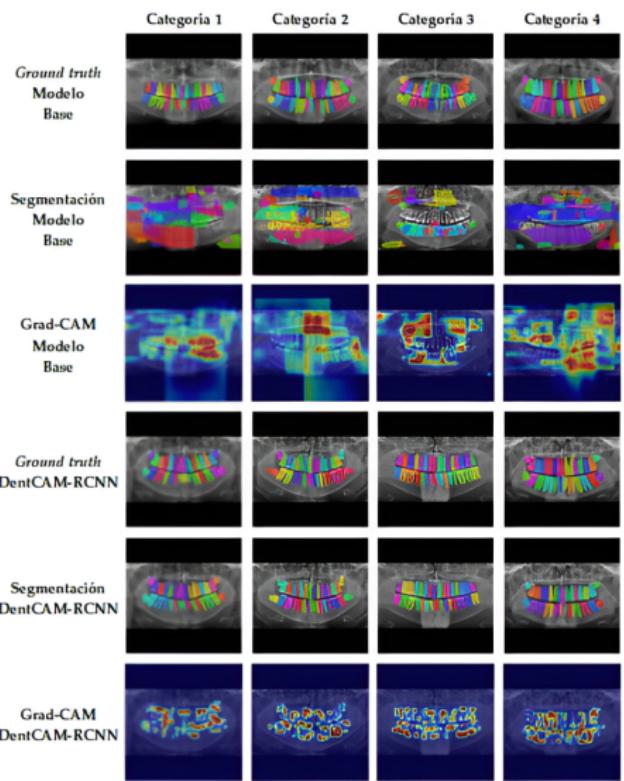
# Análisis Cuantitativo

- Modelos utilizados:
  - **Modelo base:** Mask R-CNN preentrenado en COCO, sin adaptación al dominio dental.
  - **Modelo ajustado (DentCAM-RCNN):** Mask R-CNN con ajuste fino en sus capas superiores sobre DNS Panoramic Images.
- Los resultados muestran valores promedio por categoría dental en el conjunto de test.

Modelo	Precision	Recall	F1	AP@50	Modelo	DSC	IoU	HD
Base	0.0523	0.0628	0.0571	0.0109	Base	0.0125	0.0097	30.73
Ajustado	<b>0.9736</b>	<b>0.9480</b>	<b>0.9604</b>	<b>0.9398</b>	Ajustado	<b>0.8776</b>	<b>0.7923</b>	<b>12.26</b>

Modelo	Deletion	Insertion	Stability	IoU <sub>Grad-CAM</sub>
Base	0.9938	0.9936	0.2472	0.1347
Ajustado	<b>0.1595</b>	<b>0.9965</b>	<b>0.8891</b>	<b>0.4365</b>

# Análisis Cualitativo



# Comparación con métodos clásicos de segmentación

- **Métodos clásicos de segmentación no supervisados:**  
Rendimiento muy variable dependiente del enfoque utilizado.  
Resultados extraídos de [JFR+18].
- **DentCAM-RCNN:** Supera ampliamente a estos métodos.

Método	Precision	Recall	F1-Score
Region growing	0.3500	0.6300	0.4400
Splitting/merging	0.8100	0.0800	0.1400
Global thresholding	0.5200	0.6900	0.5600
Niblack method	0.5100	0.8200	0.6100
Fuzzy C-means	0.6100	0.4500	0.4900
Canny	0.4500	0.1100	0.1700
Sobel	0.6600	0.0300	0.0600
Active contours w/o edges	0.5100	0.5700	0.5200
Level set method	0.4800	0.6800	0.5200
Watershed	0.4800	0.8200	0.5800
<b>DentCAM-RCNN</b>	<b>0.9776</b>	<b>0.9473</b>	<b>0.9622</b>

---

[JFR+18] G. Jader, J. Fontineli, M. Ruiz, K. Abdalla, M. Pithon, y L. Oliveira. Deep instance segmentation of teeth in panoramic x-ray images. *Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 400–407, 2018.

## Comparación con métodos clásicos de segmentación

	Categoría 1	Categoría 2	Categoría 3	Categoría 4
Region growing				
Splitting/ merging				
Global thresholding				
Niblack method				
Fuzzy C-means				
Canny				
Sobel				
Active contours without edges				
Level set method				
Watershed				
DentCAM-RCNN				

	Categoría 7	Categoría 8	Categoría 9	Categoría 10
Region growing				
Splitting/ merging				
Global thresholding				
Niblack method				
Fuzzy C-means				
Canny				
Sobel				
Active contours without edges				
Level set method				
Watershed				
DentCAM-RCNN				

# Comparación con estudios recientes de segmentación de instancias dentales en 2D

- DentCAM-RCNN demuestra una alta eficacia con un volumen de datos intermedio.

Estudio	Imágenes	F1-Score	DSC	IoU
Jader et al. [JFR+18]	193	0.8800	—	—
Lee et al. [LHK+20]	1024	0.8750	—	<b>0.8770</b>
Brahmi y Jdey [BJ23]	107	0.6300	—	—
Rubiú et al. [RBC+23]	1000	—	0.8700	—
DentCAM-RCNN	543	<b>0.9604</b>	<b>0.8776</b>	0.7923

---

[JFR+18] G. Jader, J. Fontineli, M. Ruiz, K. Abdalla, M. Pithon, y L. Oliveira. Deep instance segmentation of teeth in panoramic x-ray images. *Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 400–407, 2018.

[LHK+20] J.-H. Lee, S.-S. Han, Y.-H. Kim, C. Lee, y I. Kim. Application of a fully deep convolutional neural network to the automation of tooth segmentation on panoramic radiographs. *Oral Surg. Oral Med. Oral Pathol. Oral Radiol.*, 129(6):635–642, 2020.

[BJ23] W. Brahmi y I. Jdey. Automatic tooth instance segmentation and identification from panoramic x-ray images using deep cnn. *Multimedia Tools and Applications*, 83(18):55565– 55585, 2023.

[RBC+23] G. Rubiú, M. Bologna, M. Cellina, M. Cè, D. Sala, R. Pagani, E. Mattavelli, D. Fazzini, S. Ibba, S. Papa, y M. Ali. Teeth segmentation in panoramic dental x-ray using mask regional convolutional neural network. *Appl. Sci.*, 13(13):7947, 2023.

## 1 Introducción

## 2 Fundamentos Teóricos

## 3 Estado del Arte

## 4 Experimentación

## 5 Conclusiones

# Conclusiones

- Se ha desarrollado **DentCAM-RCNN**, un modelo basado en Mask R-CNN ajustado al dominio dental, en el que se ha integrado Grad-CAM para dotarlo de explicabilidad visual.
- Este modelo supera a métodos clásicos de segmentación, así como a estudios recientes de segmentación de instancias dentales en 2D que emplean Mask R-CNN.
- **Todos los objetivos planteados al inicio han sido cumplidos con éxito.**
  - ① Estudiar la segmentación de instancias y el funcionamiento de Mask R-CNN.
  - ② Aplicar Mask R-CNN a radiografías dentales panorámicas.
  - ③ Investigar el campo de la Inteligencia Artificial Explicable.
  - ④ Implementar una adaptación de Grad-CAM en Mask R-CNN.
  - ⑤ Evaluar la efectividad de Grad-CAM en Mask R-CNN.
- **Código y resultados disponibles públicamente en GitHub:**
  - [https://github.com/juanmaarg6/DentCAM\\_RCNN](https://github.com/juanmaarg6/DentCAM_RCNN).

Fin de la Presentación

Muchas gracias por su atención.

# ¿Preguntas?