

# Universidad de Alcalá

## Escuela Politécnica Superior

**Grado en Ingeniería Informática**

### **Trabajo Fin de Grado**

Estudio de técnicas de visión e inteligencia artificial aplicadas a un caso práctico

**Autor:** Pablo García García

**Tutor:** Adrián Domínguez Díaz

2024



**UNIVERSIDAD DE ALCALÁ**  
**ESCUELA POLITÉCNICA SUPERIOR**

**Grado en Ingeniería Informática**

**Trabajo Fin de grado**

**Estudio de técnicas de visión e inteligencia artificial aplicadas a  
un caso práctico**

Autor: Pablo García García

Tutor: Adrián Domínguez Díaz

**Tribunal:**

**Presidente:**

**Vocal 1º:**

**Vocal 2º:**

Fecha de depósito:



*“Computer Science is no more about computers than astronomy is about telescopes.”*  
Edsger W. Dijkstra



# Resumen

Palabras clave:





# Abstract

**Keywords:**



# Índice general

<b>Resumen</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>1. Introducción</b>	<b>13</b>
<b>2. Fundamentos de la Inteligencia Artificial y sus herramientas</b>	<b>15</b>
2.1. Tipos de aprendizaje y problemas . . . . .	15
2.2. Convolución y correlación cruzada . . . . .	16
<b>Bibliografía</b>	<b>20</b>

# Índice de figuras

2.1. Detección de bordes aplicando los kernel Sobel . . . . .	17
---	----

## Capítulo 1

# Introducción



## Capítulo 2

# Fundamentos de la Inteligencia Artificial y sus herramientas

### 2.1. Tipos de aprendizaje y problemas

Una vez entendida cuál es la idea de un modelo, es momento de discutir cómo se hace posible que este desempeñe correctamente su tarea. Esto se logra mediante un algoritmo de aprendizaje o entrenamiento que ajusta de manera óptima los parámetros del modelo. Normalmente se dispone de dos tipos de aprendizaje, **aprendizaje supervisado** y **aprendizaje no supervisado**[1].

En los algoritmos de aprendizaje supervisados, se dispone de un conjunto de datos o *dataset*, que contiene los valores de salida deseados para diferentes valores de entrada, normalmente recogiendo situaciones del pasado para poder extrapolar este conocimiento a situaciones del futuro. Los principales problemas que utilizan algoritmos de aprendizaje supervisado son los problemas de clasificación y de regresión. En los **problemas de clasificación**, se dispone de una serie de clases  $C_1, C_2, \dots, C_n$ , y para una serie de valores de entrada  $x_1, x_2, \dots, x_m$ , debe decidirse a qué clase pertenece dicha entrada. Un ejemplo sería decidir si un paciente va a sufrir un cierto tipo de cáncer dada su edad, peso, y otras constantes vitales. Algunos de los modelos más populares para llevar a cabo este tipo de tareas son árboles de decisión, máquinas de soporte vectorial, Naïve Bayes,  $k$ -vecinos, y redes neuronales; siendo estas últimas objeto de estudio en este trabajo. Otro tipo de problema popular a la hora de disponer de datos etiquetados, son los **problemas de regresión**, que se diferencia principalmente de la clasificación en que en este caso, los valores no son clases (valores discretos) sino valores continuos. Para resolver este tipo de problemas se suelen utilizar regresiones lineales y no lineales (exponencial, polinómica, etc). Un ejemplo de un problema de regresión sería predecir las horas que dormirá una persona dada su edad, horas trabajadas en el día, horas de recreo en el día, etc.

Por otro lado, los algoritmos de aprendizaje no supervisado no reciben los valores de salida esperados para una cierta observación (justo al contrario que en el caso supervisado), pues será trabajo del algoritmo encontrar relaciones y patrones entre los datos proporcionados. En este tipo de aprendizaje también se trata el problema de clasificación, sin embargo, es más común llamarlo **clustering** o **segmentación**, pues a priori no se conoce el número de clases y cuáles son, es el algoritmo el que deberá encontrar relaciones entre los datos para determinar esto. Algoritmos populares para realizar esta tarea son  $k$ -medias (y su variante  $k$ -medanas), clusterización jerárquica aglomerativa, modelos de mixtura gaussianos, y DBSCAN. Un ejemplo sencillo de este problema es detectar las diferentes regiones y objetos representados en una imagen, pues inicialmente no se conoce el número de regiones u objetos, y deben detectarse todas, asignando cada píxel de la imagen a cada una de ellas.

## 2.2. Convolución y correlación cruzada

Una de las operaciones matemáticas más conocidas y que es más usada al trabajar con señales e imágenes es la llamada convolución, denotada por  $*$ , y dadas las funciones  $f(t)$  y  $g(t)$ , su convolución se define de la siguiente manera.

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

En este caso se está asumiendo que el dominio de  $f(\tau)g(t - \tau)$  es  $\mathbb{R}$ , lo que permite integrar sobre todo  $\mathbb{R}$ , de lo contrario, se modifica la definición para integrar solo sobre un intervalo  $[a, b]$ . En general, esta operación crea una nueva función a partir de otras dos, que indica cómo interactúan entre sí, y que permite aplicar filtros a señales e imágenes. Como se acaba de comentar para el caso de las imágenes, se puede tratar con señales que no dependan únicamente de una variable, pues estas se representan como una función de dos variables  $f(u, v)$ . En este caso, la convolución queda definida de la siguiente manera.

$$(f * g)(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta)g(u - \xi, v - \eta)d\xi d\eta$$

Partiendo de la primera definición mostrada, se pueden demostrar algunas propiedades útiles que cumple la convolución[2]:

- Conmutativa:  $f * g = g * f$
- Asociativa:  $f * (g * h) = (f * g) * h$
- Distributiva:  $f * (g + h) = (f * g) + (f * h)$
- Derivada:  $\frac{d}{dt}(f * g) = \frac{df}{dt} * g = \frac{dg}{dt} * f$
- Teorema de convolución:  $\mathcal{L}\{f * g\} = \mathcal{L}\{f\} \cdot \mathcal{L}\{g\}$

Si bien en las definiciones previas se ha tomado la integral y tanto  $\mathbb{R}$  como  $\mathbb{R}^2$  como dominios continuos sobre los que calcular la convolución, no se debe olvidar que las imágenes no dejan de ser matrices o funciones de dos variables con un dominio discreto, por lo que se debe presentar una definición adecuada a este caso[3].

$$(f * g)(u, v) = \sum_{i=-k}^k \sum_{j=-k}^k g(i, j)f(u - i, v - j)$$

En la expresión anterior, a la función  $g(u, v)$  se le llama filtro o *kernel* de convolución. A continuación se muestra un ejemplo de cómo calcular una convolución.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 7 & 8 & 9 \\ 9 & 8 & 7 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 14 & 15 & 16 \\ 16 & 15 & 14 \\ 16 & 15 & 14 \end{pmatrix}$$

$$\begin{aligned} 1 \cdot 1 + 2 \cdot 0 + 3 \cdot 0 + 5 \cdot 0 + 6 \cdot 1 + 7 \cdot 0 + 9 \cdot 0 + 8 \cdot 0 + 7 \cdot 1 &= 14 \\ 2 \cdot 1 + 3 \cdot 0 + 4 \cdot 0 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 0 + 8 \cdot 0 + 7 \cdot 0 + 6 \cdot 1 &= 15 \\ &\vdots \\ 7 \cdot 1 + 6 \cdot 0 + 5 \cdot 0 + 3 \cdot 0 + 2 \cdot 1 + 1 \cdot 0 + 3 \cdot 0 + 4 \cdot 0 + 5 \cdot 1 &= 14 \end{aligned}$$

Aplicando diferentes kernels de convolución a una imagen se pueden extraer diferentes tipos de características de una imagen, como por ejemplo bordes. El filtro Sobel es capaz de hacer esto con los kernels que se muestran a continuación, pues se comportan como aproximaciones de las derivadas parciales de la imagen en un punto teniendo en cuenta los píxeles cercanos[4].



$$G_u = \frac{\partial f(u, v)}{\partial u} \approx \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad G_v = \frac{\partial f(u, v)}{\partial v} \approx \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

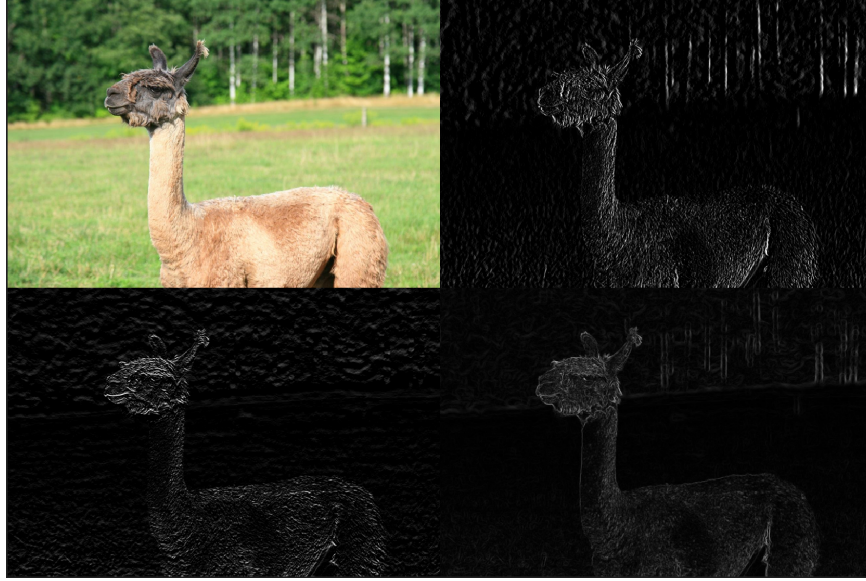


Figura 2.1: Detección de bordes aplicando los kernel Sobel

A continuación se va a calcular la convolución de la matriz del ejemplo anterior con  $G_u$ . Para comprobar los cálculos, se puede realizar esto con la función `convn` de MATLAB. Al realizar los cálculos a mano tal y como se ha mostrado en el ejemplo anterior, se obtiene como resultado la matriz  $A$ , mientras que MATLAB devuelve la matriz  $B$  como resultado de `convn(X,  $G_u$ , 'valid')`. ¿Qué acaba de suceder? ¿Está mal codificada la función de MATLAB? ¿Está mal realizado el ejemplo?

$$A = \begin{pmatrix} 4 & 4 & 4 \\ -4 & -4 & -4 \\ -4 & -4 & -4 \end{pmatrix} \quad B = \begin{pmatrix} -4 & -4 & -4 \\ 4 & 4 & 4 \\ 4 & 4 & 4 \end{pmatrix}$$

La respuesta a estas preguntas se podría resumir en que se ha realizado una “pequeña trampa” a la hora de calcular la convolución manualmente en el ejemplo, ya que no se ha aplicado correctamente la definición dada. ¿Qué sentido tiene hacer esto? En visión artificial y tratamiento de imágenes, muchos autores y librerías llaman convolución a la operación que se ha mostrado en el primer ejemplo, cuando en realidad no lo es y trae lugar a confusión. Dicha operación se llama correlación cruzada, denotada por  $(\star)$ , y que es muy similar a la convolución, pues su principal diferencia es que en la convolución “real”, el kernel se rota 180 grados antes de calcular la convolución “falsa” o correlación cruzada, es decir,  $X \star Y = X \star (R_{180} \cdot Y)$ , donde  $R_{180}$  es la matriz de rotación de 180 grados. La correlación cruzada de dos imágenes (matrices) se define de la siguiente manera[3].

$$(f \star g)(u, v) = \sum_{i=-k}^k \sum_{j=-k}^k g(i, j) f(u + i, v + j)$$

Esta operación sí es con la que realmente se aplican los filtros a las imágenes y con la que se trabaja en general en el campo de la visión artificial. Es importante ver que ahora, al contrario que con la convolución,  $f \star g \neq g \star f$ . Algunas librerías de visión artificial tratan a la correlación cruzada como convolución debido

al frecuente uso que tiene una sobre la otra y la forma similar que tienen de calcularse. Un ejemplo es OpenCV en la documentación de su función `filter2D`, donde se comenta que aplica una convolución cuando realmente aplica la correlación cruzada[5]. Finalmente, como se verá en próximos capítulos, las famosas redes neuronales convolucionales, no aplican convoluciones sino correlaciones cruzadas.



# Bibliografía

- [1] R. Szeliski, *Computer Vision: Algorithms and Applications, 2nd ed.* Springer International Publishing, 2022, ISBN: 978-3-030-34371-2. DOI: 10.1007/978-3-030-34372-9. dirección: <https://szeliski.org/Book/>.
- [2] López y H. I, “Método alternativo para calcular la convolución de señales en tiempo continuo,” 2009.
- [3] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016, págs. 326-330. dirección: <http://www.deeplearningbook.org>.
- [4] W. Gao, L. Yang, X. Zhang y H. Liu, “An improved Sobel edge detection,” *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, vol. 5, págs. 67-71, 2010. DOI: 10.1109/ICCSIT.2010.5563693.
- [5] *OpenCV: Image Filtering*. dirección: [https://docs.opencv.org/4.x/d4/d86/group\\_\\_imgproc\\_\\_filter.html](https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html).



Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá