



UNIVERSIDAD  
DE GRANADA

Facultad de Ciencias  
Escuela Técnica Superior de Ingenierías Informática y de  
Telecomunicaciones

GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

# Localización de Regiones de Interés Utilizando Aprendizaje Profundo

Presentado por:  
Laura Gómez Garrido

Tutor:  
Jesús Chamorro Martínez  
*Ciencias de la Computación e Inteligencia Artificial*

Curso académico 2019-2020



# Localización de Regiones de Interés Utilizando Aprendizaje Profundo

Laura Gómez Garrido

Laura Gómez Garrido *Localización de Regiones de Interés Utilizando Aprendizaje Profundo.*  
Trabajo de fin de Grado. Curso académico 2019-2020.

**Responsable de  
tutorización**

Jesús Chamorro Martínez  
*Ciencias de la Computación e Inteligencia  
Artificial*

Grado en Ingeniería  
Informática y Matemáticas

Facultad de Ciencias  
Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicaciones

Universidad de Granada

#### DECLARACIÓN DE ORIGINALIDAD

D./Dña. Laura Gómez Garrido

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2019-2020, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 9 de julio de 2020

Fdo: Laura Gómez Garrido

*Dedicatoria (opcional)*

*Ver archivo preliminares/dedicatoria.tex*



# Índice general

Índice de figuras	IX
Índice de tablas	XI
Agradecimientos	XIII
Summary	XV
Introducción	XVII
<b>I. Estado del arte</b>	<b>1</b>
<b>1. Red Neuronal</b>	<b>3</b>
1.1. El problema de clasificar una imagen. . . . .	3
1.2. Clasificadores Lineales . . . . .	4
<b>II. Primera parte</b>	<b>7</b>
<b>2. Primer capítulo</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Elementos del texto . . . . .	10
2.2.1. Listas . . . . .	10
2.2.2. Tablas y figuras . . . . .	11
2.3. Entornos matemáticos . . . . .	11
2.4. Bibliografía e índice . . . . .	12
<b>III. Segunda parte</b>	<b>13</b>
<b>3. Segundo capítulo</b>	<b>15</b>
3.1. Primera sección . . . . .	15
<b>A. Primer apéndice</b>	<b>17</b>
<b>Glosario</b>	<b>19</b>
<b>Bibliografía</b>	<b>21</b>
<b>Índice alfabético</b>	<b>23</b>





## Índice de figuras

1.1. Cómo un ordenador ve una imagen . . . . .	3
2.1. Logotipo de la Universidad de Granada . . . . .	11



# Índice de tablas

2.1. Ejemplo de tabla . . . . .	11
---------------------------------	----



# Agradecimientos

Agradecimientos del libro (opcional, ver archivo preliminares/agradecimiento.tex).



## Summary

An english summary of the project (around 800 and 1500 words are recommended).

File: preliminares/summary.tex





## Introducción

De acuerdo con la comisión de grado, el TFG debe incluir una introducción en la que se describan claramente los objetivos previstos inicialmente en la propuesta de TFG, indicando si han sido o no alcanzados, los antecedentes importantes para el desarrollo, los resultados obtenidos, en su caso y las principales fuentes consultadas.

Ver archivo preliminares/introduccion.tex



## Parte I.

### Estado del arte

A continuación, explicaremos de forma concisa lo que es una *Red Neuronal* y una *Red Neuronal Convolucionada* para, seguidamente, hablar sobre los últimos avances en la localización de regiones de interés en imágenes.



# 1. Red Neuronal

## 1.1. El problema de clasificar una imagen.

Cuando observamos una imagen, podemos localizar varios elementos a partir de los cuáles esta se encuentra compuesta con tan sólo un vistazo. Sin embargo, para un ordenador no se trata de algo tan sencillo puesto que para él se trata de un gran conjunto de números que no tienen por qué tener relación alguna entre sí.



Figura 1.1.: Cómo un ordenador ve una imagen

Idealmente, en esta imagen desearíamos que fuera capaz de identificar que se trata de un perro, en concreto de un chiguagua, de pelaje blanco y manchas color café que se encuentra sobre un césped. Estos poquitos datos que para nosotros parecen tan triviales necesitan de horas y horas de computación para ser obtenidos a partir de una imagen cualquiera.

Dejamos todos estos detalles, a los cuales esperamos poder llegar en un futuro no muy lejano, y simplificamos el problema a tener un conjunto de etiquetas y buscar con cuál de

todas ellas tiene mayor relación nuestra imagen.

Una primera idea, sería utilizar utilizar *k-Nearest Neighbor Classifier*, en adelante *k-NN*. Este clasificador consiste en que para cada etiqueta se correspondan  $k$  imágenes y consideraremos como etiqueta idónea para nuestra clasificación aquella cuya distancia, siendo Manhattan y Euclídea las más comunes, entre los píxeles con nuestra imagen sea menor. Nos encontramos con que el tiempo de entrenamiento de nuestro clasificador sería ínfimo en comparación con el tiempo de clasificación, si bien se pueden hacer diversas mejoras que cambien estos hechos.

De esta sencilla propuesta, surgen diversos problemas. Al darle mayor importancia al valor concreto de los píxeles, en lugar de a las formas que de las que está compuesta la figura, nos encontramos con que valores como los colores de fondo pueden influir más en la clasificación que los propios píxeles de la figura que queremos clasificar. En el ejemplo de la imagen del chiguagua, si considerásemos las etiquetas verde y perro, tendríamos que por lo general como respuesta el color verde, pese a que estamos más interesados por la mascota en sí. Otro gran problema, sería la baja escalabilidad que nos proporciona esta solución, al incrementarse enormemente el costo computacional de clasificación conforme aumenta el número de etiquetas.

El siguiente paso, es buscar una forma de “memorizar” los datos de entrenamiento de forma que no tengamos que estar comparándolos con todos ellos cuando queramos clasificar una imagen. Lo que buscamos es poder valorar de alguna forma la imagen completa y a partir de esta “puntuación” conocer qué etiqueta le corresponde mejor a nuestra imagen. De esta forma, veríamos drásticamente reducido el tiempo de clasificación, sacrificando para ello el tiempo de entrenamiento, y podríamos ampliar en varias unidades la cantidad de datos de entrenamiento utilizados, aumentando así la precisión de nuestro clasificador.

### 1.2. Clasificadores Lineales

Antes de nada, vamos a comenzar contextualizando matemáticamente el entorno en el que nos encontramos. Una vez realizado esto podremos hablar correctamente de los clasificadores lineales y así poder extenderlos naturalmente a los conceptos de Red Neuronal y de Red Neuronal Convolucionada.

Sea  $D \in \mathbb{N}$  la dimensión de nuestras imágenes, por lo general el número de píxeles que estas poseen, y  $K \in \mathbb{N}$  la cantidad de etiquetas o categorías bajo las cuales pueden ser clasificadas. Siendo  $N \in \mathbb{N}$  el número de ejemplos que utilizaremos para entrenar nuestro clasificador, tendremos que para cada dato de entrenamiento  $x_i \in \mathbb{R}^D$   $i = 1, \dots, N$  le corresponde una etiqueta  $y_i \in 1, \dots, K$  tal que juntos conforman el par  $(x_i, y_i)$  de imagen y categoría a la que pertenece.

Para que no sea más fácil de entender este contexto, tomaremos como ejemplo el conjunto de datos CIFAR-10 que consiste en 60000 imágenes RGB de dimensión 32x32 y 10 categorías. De esta forma, tendríamos que  $D = 32 \cdot 32 \cdot 3 = 3072$ ,  $K = 10$  y  $N = 60000$ .

**Definición 1.1.** Definiremos la *función de puntuación* o *score function* como una función  $f : \mathbb{R}^D \rightarrow \mathbb{R}^K$  que asigna los píxeles de la imagen sin procesar una serie de puntuaciones para cada etiqueta.

La función de puntuación nos revela la probabilidad que tiene cada imagen de pertenecer a cada una de la distintas etiquetas de las que disponemos. Dicho esto, podemos definir un *clasificador lineal* o *linear classifier* como aquel que utiliza una función de puntuación lineal, es decir, de la forma:

$$f(x) = W \cdot x + b \quad W \in M_{K,D}(\mathbb{R}) \quad b \in \mathbb{R}^K \quad \forall x \in \mathbb{R}^D$$

Dicho esto, existen diferentes tipos de clasificadores lineales y la principal diferencia entre ellos reside en la función de pérdida que utilicen a la hora de entrenar la red.

**Definición 1.2.** Una *función de pérdida* o *loss function* es aquella que durante el entrenamiento de un clasificador se encarga de penalizar las etiquetas incorrectas.





## **Parte II.**

### **Primera parte**

Si el trabajo se divide en diferentes partes es posible incluir al inicio de cada una de ellas un breve resumen que indique el contenido de la misma. Esto es opcional.



## 2. Primer capítulo

Al inicio de cada capítulo puede incluirse un breve resumen. Esto es opcional.

### 2.1. Introducción

Este documento es una plantilla para la elaboración de un trabajo fin de Grado siguiendo las [directrices](#) de la comisión de Grado en Matemáticas de la Universidad de Granada que, a fecha de marzo de 2019, son las siguientes:

- La memoria debe realizarse con un procesador de texto científico, preferiblemente (La)TeX.
- La portada debe contener el logo de la UGR, incluir el título del TFG, el nombre del estudiante y especificar el grado, la facultad y el curso actual.
- La contraportada contendrá además el nombre del tutor o tutores.
- La memoria debe necesariamente incluir:
  - un índice detallado de capítulos y secciones,
  - un resumen amplio en inglés del trabajo realizado (se recomienda entre 800 y 1500 palabras),
  - una introducción en la que se describan claramente los objetivos previstos inicialmente en la propuesta de TFG, indicando si han sido o no alcanzados, los antecedentes importantes para el desarrollo, los resultados obtenidos, en su caso y las principales fuentes consultadas,
  - una bibliografía final que incluya todas las referencias utilizadas.
- Se recomienda que la extensión de la memoria sea entre 30 y 60 páginas, sin incluir posibles apéndices.

Para generar el pdf a partir de la plantilla basta compilar el fichero `libro.tex`. Es conveniente leer los comentarios contenidos en dicho fichero pues ayudarán a entender mejor como funciona la plantilla.

La estructura de la plantilla es la siguiente<sup>1</sup>:

- Carpeta **preliminares**: contiene los siguientes archivos
  - dedicatoria.tex** Para la dedicatoria del trabajo (opcional)
  - agradecimientos.tex** Para los agradecimientos del trabajo (opcional)
  - introduccion.tex** Para la introducción (obligatorio)
  - summary.tex** Para el resumen en inglés (obligatorio)

---

<sup>1</sup>Los nombres de las carpetas no se han acentuado para evitar problemas en sistemas con Windows

## 2. Primer capítulo

El resto de archivos de dicha carpeta no es necesario editarlos pues su contenido se generará automáticamente a partir de los metadatos que agreguemos en `libro.tex`

- Carpeta **capitulos**: contiene los archivos de los capítulos del TFG. Añadir tantos archivos como sean necesarios. Este capítulo es `capitulo01.tex`.
- Carpeta **apendices**: Para los apéndices (opcional)
- Carpeta **img**: Para incluir los ficheros de imagen que se usarán en el documento.
- Carpeta **paquetes**: Incluye dos ficheros
  - hyperref.tex** para la configuración de hipervínculos al generar el pdf (no es necesario editarlo)
  - comandos-entornos.tex** donde se pueden añadir los comandos y entornos personalizados que precisemos para la elaboración del documento. Contiene algunos ejemplos
- Fichero `library.bib`: Para incluir las referencias bibliográficas en formato bibtex. Son útiles las herramientas [doi2bib](#) y [OttoBib](#) para generar de forma automática el código bibtex de una referencia a partir de su DOI o su ISBN.
- Fichero `glosario.tex`: Para incluir un glosario en el trabajo (opcional).
- Fichero `libro.tex`: El documento maestro del TFG que hay que compilar con  $\text{\LaTeX}$  para obtener el pdf. En dicho documento hay que cambiar la *información del título del TFG y el autor así como los tutores*.

## 2.2. Elementos del texto

En esta sección presentaremos diferentes ejemplos de los elementos de texto básico. Conviene consultar el contenido de `capitulos/capitulo01.tex` para ver cómo se han incluido.

### 2.2.1. Listas

En  $\text{\LaTeX}$  tenemos disponibles los siguientes tipos de listas:

Listas enumeradas:

1. item 1
2. item 2
3. item 3

Listas no enumeradas

- item 1
- item 2
- item 3

Listas descriptivas

**termino1** descripción 1

**termino2** descripción 2

### 2.2.2. Tablas y figuras

En la **Tabla 2.1** o la **Figura 2.1** podemos ver...

Agrupados		
cabecera	cabecera	cabecera
elemento	elemento	elemento
elemento	elemento	elemento
elemento	elemento	elemento

Tabla 2.1.: Ejemplo de tabla



Figura 2.1.: Logotipo de la Universidad de Granada

## 2.3. Entornos matemáticos

**Teorema 2.1.** *Esto es un ejemplo de teorema.*

**Proposición 2.1.** *Ejemplo de proposición*

**Lema 2.1.** *Ejemplo de lema*

**Corolario 2.1.** *Ejemplo de corolario*

**Definición 2.1.** *Ejemplo de definición*

**Observación 2.1.** *Ejemplo de observación*

Y esto es una referencia al **Teorema 2.1**.

Identidad Pitagórica (2.1)

$$\cos^2 x + \sin^2 x = 1 \quad (2.1)$$

La fórmula de Gauss-Bonnet para una superficie compacta  $S$  viene dada por:

$$\int_S K = 2\pi\chi(S)$$

## 2.4. Bibliografía e índice

Esto es un ejemplo de texto en un capítulo. Incluye varias citas tanto a libros [Eul82, Eul84, Eul85] como a recursos online [Eul] (páginas web). Ver el fichero library.bib.

Además incluye varias entradas al índice alfabético mediante el comando \index

**Parte III.**

**Segunda parte**





## 3. Segundo capítulo

Mathematicians have tried in vain to this day to discover some order in the sequence of prime numbers, and we have reasons to believe that it is a mystery into which the human mind will never penetrate.

---

*(Leonard Euler)*

### 3.1. Primera sección



## A. Primer apéndice

Los apéndices son opcionales.

Archivo: `apendices/apendice01.tex`



## Glosario

La inclusión de un glosario es opcional.

Archivo: `glosario.tex`

$\mathbb{R}$  Conjunto de números reales.

$\mathbb{C}$  Conjunto de números complejos.

$\mathbb{Z}$  Conjunto de números enteros.



## Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [Eul] Leonhard Euler. [https://en.wikipedia.org/wiki/Leonhard\\_Euler](https://en.wikipedia.org/wiki/Leonhard_Euler). Recurso online. Accedido el 14 de marzo de 2019. [Citado en pág. 12]
- [Eul82] Leonhard Euler. *Commentationes mechanicae ad theoriam machinarum pertinentes. Vol. III. Leonhardi Euleri Opera Omnia, Series Secunda: Opera Mechanica et Astronomica, XVII.* Orell Füssli, Zürich, 1982. Edited and with a preface by Charles Blanc and Pierre de Haller. [Citado en pág. 12]
- [Eul84] Leonhard Euler. *Elements of algebra*. Springer-Verlag, New York, 1984. Translated from the German by John Hewlett, Reprint of the 1840 edition, With an introduction by C. Truesdell. [Citado en pág. 12]
- [Eul85] Leonhard Euler. An essay on continued fractions. *Math. Systems Theory*, 18(4):295–328, 1985. Translated from the Latin by B. F. Wyman and M. F. Wyman. [Citado en pág. 12]





# Índice alfabético

Gauss-Bonnet  
fórmula, **11**

Identidad pitagórica, **11**

Leonard  
Euler, **12**