



ugr | Universidad
de **Granada**

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA

Sistema de Recuperación de Imágenes basado en Términos Lingüísticos Locales

Autor
Lidia Sánchez Mérida

Director
Jesús Chamorro Martínez



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

—
Granada, 18 de junio de 2019

Sistema de Recuperación de Imágenes basado en Términos Lingüísticos Locales

Lidia Sánchez Mérida

Palabras clave: descriptor, imagen, consulta, comparador, etiqueta, distancia, local, máximo, mínimo, meda

Resumen

Debido al proceso de digitalización que, desde hace tiempo, se está desarrollando el volumen de imágenes que se encuentran en Internet o en una base de datos es gigantesco. Esto provoca la necesidad de desarrollar sistemas capaces de extraer y gestionar la información relevante asociada a las propias imágenes. Hasta el momento la gran mayoría de estos sistemas han utilizado sus propiedades gráficas, proporcionando la posibilidad de realizar consultas en base a sus características visuales, y no a los objetos que aparecen.

Es por ello por lo que en este Trabajo de Fin de Grado se ha implementado un sistema de recuperación de imágenes basado en las etiquetas producidas por una Red Neuronal Convolucionada, que es capaz de reconocer un amplio rango de diversos objetos. Así mismo se han desarrollado distintas métricas capaces de generar resultados diferentes en función de las necesidades del usuario. Cabe destacar que para ejecutar esta funcionalidad es necesario disponer de una imagen consulta que represente el papel protagonista cuando se realice la comparación. No obstante, para dotar a esta funcionalidad de una mayor flexibilidad, también se ha añadido una ampliación que permite realizar una consulta utilizando solamente el concepto lingüístico correspondiente al objeto que deseamos buscar. Además también podremos restringir la posición en la que este debe aparecer.

A lo largo de este documento se podrán encontrar todos los aspectos detallados de las funcionalidades descritas anteriormente junto con varios y diversos ejemplos ilustrativos que demuestren su funcionamiento.

Image Retrieval System Based on Local Linguistic Terms

Lidia Sánchez Mérida

Keywords: descriptor, image, query, comparator, label, distance, local, maximum, minimum, average

Abstract

Due to the digitalisation process, which has been developing for a long time, the number of images found on the Internet or in databases is enormous. This causes the need to develop systems capable of extracting and managing the important information related to the images. So far most of these systems use the graphic properties of the images so as to make queries based on their visual features, not the objects which appear.

It's for this reason that in this Final Degree Project I developed an image recovery system based on the labels produced by a Convolutional Neural Network, which is able to recognize a wide range of different objects. In addition to that, distinct metrics have been developed so as to generate different results depending on what the user wants. It should be pointed out that to test this usefulness it's necessary to have a query image to play de leading role in the comparison. Nevertheless, in order to provide this functionality with more flexibility, an extension has also been added which allows the user to make a query based on the word related to the object which we wish to search. Moreover, we can restrict the position in which this element should appear too.

Throughout this document you will be able to find all the detailed aspects related to those previous functionalities along with several and different illustrative examples which will demonstrate how they work.

Yo, **Lidia Sánchez Mérida**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con [REDACTED], autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Lidia Sánchez Mérida

Granada a 18 de Junio de 2019.

D. **Jesús Chamorro Martínez**, Profesor del Área de Visión por Computador del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Sistema de Recuperación de Imágenes basado en Términos Lingüísticos Locales***, ha sido realizado bajo su supervisión por **Lidia Sánchez Mérida**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 18 de junio de 2019.

El director:

Jesús Chamorro Martínez

Agradecimientos

Gracias por el apoyo que he recibido de los compañeros que me han ido acompañando por este arduo camino, a mis amigos por sus incansables ánimos y su gratificante compañía y, en especial, a mi madre quien siempre ha estado a mi lado porque sin ella nada de esto hubiese sido posible.

Gracias también a mi tutor Jesús por su paciencia, su entrega y por transmitir su infinito entusiasmo en todos los proyectos de los que forma parte. Ojalá hubiese más docentes que compartiesen sus inigualables cualidades humanas y su maravillosa forma de enseñanza.

Índice general

I Sistemas de recuperación enfocados en términos lingüísticos	20
1. Introducción	21
1.1. Motivación	22
1.2. Objetivos	22
1.3. Estructura de la memoria.....	23
2. Redes Neuronales Convolucionadas (CNN).....	24
2.1. Redes Neuronales	24
2.1.1. Aprendizaje	25
2.1.2. Arquitecturas	26
2.1.2.1. Una sola capa	28
2.1.2.2. Múltiples capas	29
2.1.2.3. Recurrentes.....	30
2.2. Redes Neuronales Convolucionadas (CNN)	32
3. Sistemas de Recuperación.....	34
3.1. Sistemas de recuperación de imágenes.	34
3.1.1. Rasgos primitivos	38
3.1.2. Rasgos lógicos	39
3.1.3. Atributos abstractos	42
3.2. Análisis de imágenes.	43
4. Métricas	49
4.1. Clasificación de los comparadores.	49
4.1.1. Distinta posición	49
4.1.2. Sin repetidos.....	55
4.1.3. Misma posición.....	60
5. Consultas en base a términos lingüísticos	66
5.1. Consulta global a la imagen.....	66

5.2.	Consulta local a la imagen.....	67
5.2.1.	Operador AND	68
5.2.2.	Operador OR.....	70
6.	Base de datos.....	73
6.1.	Requisitos considerados.....	73
6.2.	Creación de la base de datos de imágenes.....	74
II Desarrollo e implementación del software.....		77
7.	Organización y presupuesto	78
8.	Requisitos.....	81
8.1.	Requisitos de datos	81
8.2.	Requisitos funcionales.....	81
8.3.	Requisitos no funcionales	82
9.	Casos de uso.....	84
9.1.	Actores	84
9.2.	Diagrama de casos de uso.....	84
9.3.	Descripciones de los casos de uso.....	85
10.	Análisis	99
11.	Diseño	102
10.1.	Java Multimedia Retrieval (JMR).....	102
10.2.	Novedades con respecto a la JMR.....	103
10.2.1.	Métricas	103
10.2.2.	<i>LabelGriddedDescriptor</i>	104
10.3.	Diagrama de paquetes	105
10.4.	Diagrama de clases.....	106
10.4.1.	Diagrama de la interfaz	106
10.4.2.	Diagrama de los comparadores y descriptor	106
12.	Implementación.....	108
13.	Manual de usuario.....	110
13.1.	Primer bloque: imágenes.....	111
13.2.	Segundo bloque: consultas.....	111
13.3.	Tercer bloque: métricas internas a <i>LabelDescriptor</i>	113
13.4.	Cuarto bloque: base de datos.....	113
13.5.	Quinto bloque: consulta con una etiqueta.	114
13.6.	Sexto bloque: descriptores.....	115
13.7.	Ejemplos de uso.	116
13.7.1.	Consulta en función de una imagen sin base de datos.	116
13.7.2.	Consulta en base a una imagen utilizando una base de datos.	118

13.7.3. Consulta en base a una etiqueta.....	120
14. Conclusiones y futuras investigaciones.	122
Bibliografía	124

Índice de figuras

Figura 1: Esquema biológico de dos neuronas	25
Figura 2: Etapas principales para transmitir información entre dos neuronas.	28
Figura 3: Ejemplo de arquitectura de una sola capa.	29
Figura 4: Ejemplo de arquitectura multicapa.	30
Figura 5: Primer ejemplo de arquitectura recurrente con retroalimentación.	31
Figura 6: Segundo ejemplo de arquitectura recurrente con retroalimentación.	32
Figura 7: Estructura de una CNN.	33
Figura 8: Flujo general de un CBIR.	37
Figura 9: Histograma de color de la imagen consulta.....	38
Figura 10: Imagen consulta.....	38
Figura 11: Lista de imágenes resultante tras la comparación, de mayor a menor grado de similitud.....	39
Figura 12: Ejemplo de funcionamiento de IRIS.	41
Figura 13: Primer experimento con un objeto taza extrayendo la estructura del color de forma global.	44
Figura 14: Segundo experimento con un objeto maceta extrayendo la estructura del color de forma global.	45
Figura 15: Primer experimento con dos objetos y extrayendo el color medio de forma global.....	45
Figura 16: Segundo experimento con dos objetos extrayendo el color medio localmente.	46
Figura 17: Primer experimento con un objeto sacapuntas y extrayendo la estructura del color de forma global.	46
Figura 18: Segundo experimento con un objeto sacapuntas y extrayendo la estructura del color localmente.	47
Figura 19: Primer experimento con un objeto copa y usando la identificación mediante etiquetas globalmente.....	47
Figura 20: Primer experimento de la primera familia de comparadores – Al menos un objeto en cualquier posición.....	50
Figura 21: Segundo experimento de la primera familia de comparadores – Al menos un objeto en cualquier posición.....	51
Figura 22: Tercer experimento de la primera familia de comparadores – Al menos un objeto en cualquier posición utilizando la doble inclusión.....	51
Figura 23: Primer experimento de la primera familia de comparadores – La mayoría de los elementos.....	52
Figura 24: Segundo experimento de la primera familia de comparadores – La mayoría	

de los elementos utilizando la doble inclusión.....	53
Figura 25: Tercer experimento de la primera familia de comparadores – La mayoría de los elementos.....	53
Figura 26: Cuarto experimento de la primera familia de comparadores – La mayoría de los elementos utilizando la doble inclusión.....	54
Figura 27: Primer experimento de la primera familia de comparadores – En promedio.	55
Figura 28: Segundo experimento de la primera familia de comparadores – En promedio.....	55
Figura 29: Imagen de un trozo de césped.....	56
Figura 30: Imagen que representa un grupo de personas practicando deporte sobre un parque con césped.	56
Figura 31: Primer experimento de la segunda familia de comparadores – Al menos un objeto.....	57
Figura 32: Primer experimento de la segunda familia de comparadores – La mayoría de objetos.....	58
Figura 33: Imagen repleta de fresas.	58
Figura 34: Cuenco de fresas.....	58
Figura 35: Tarta de fresa con dos fresas decorativas.....	59
Figura 36: Tarta de queso con dos fresas decorativas.....	59
Figura 37: Segundo experimento de la segunda familia de comparadores – La mayoría de los objetos.....	59
Figura 38: Primer experimento de la segunda familia de comparadores – En promedio.	60
Figura 39: Primer experimento de la tercera familia – Al menos uno de los elementos.	61
Figura 40: Segundo experimento de la tercera familia – Al menos uno de los objetos.	62
Figura 41: Primer experimento de la tercera familia – La mayoría de objetos.	63
Figura 42: Segundo experimento de la tercera familia – La mayoría de los elementos.	63
Figura 43: Primer experimento de la tercera familia de comparadores – En promedio.	64
Figura 44: Segundo experimento de la tercera familia – En promedio.	65
Figura 45: Búsqueda en Google de una pelota de tenis situada a la derecha de la imagen.....	67
Figura 46: Consulta de una pelota de tenis situada a la derecha.	67
Figura 47: Consulta de una copa de beber situada en la parte superior de la imagen.	68
Figura 48: Consulta de un sacapuntas situado en la esquina inferior izquierda de la imagen.	69
Figura 49: Consulta de una taza situada en la esquina superior derecha de la imagen.	69
Figura 50: Consulta doble de una copa situada en la parte superior central de la imagen y otra situada en la parte inferior central.....	70
Figura 51: Consulta de un labial situado en la parte superior o inferior de la imagen.	71
Figura 52: Consulta de un pimiento situado a la derecha o a la izquierda de la imagen.	71
Figura 53: Consulta de un mechero situado en la parte central o en el lateral derecho de la imagen.	72
Figura 54: Pimiento situado debajo de la imagen.....	74
Figura 55: Mechero situado en la parte de debajo de la imagen.	74
Figura 56: Taza situada arriba de la imagen.	74
Figura 57: Pelota de tenis situada arriba de la imagen.....	74
Figura 58: Sacapuntas situado en la parte central de la imagen.	75
Figura 59: Labial situado en el centro de la imagen.	75
Figura 60: Copia situada a la derecha.	75

Figura 61: Goma de borrar situada la izquierda.	75
Figura 62: Taza situada a la derecha de la imagen.	75
Figura 63: Taza situada a la derecha de la imagen.	75
Figura 64: Dos labiales.	76
Figura 65: Imagen de una pelota y una taza.	76
Figura 66: Imagen de tres mecheros.	76
Figura 67: Imagen de una goma de borrar y dos sacapuntas.	76
Figura 68: Cuenco de fresas.	76
Figura 69: Campo de margaritas.	76
Figura 70: Diagrama de Gantt.	79
Figura 71: Ejemplo de una de las páginas del <i>javadoc</i> del proyecto.	108
Figura 72: Imagen de la vista principal de la aplicación.	111
Figura 73: Primer panel asociado a las operaciones con imágenes.	111
Figura 74: Segundo panel relacionado con los parámetros de las consultas.	111
Figura 75: Vista de la ventana de diálogo que contiene los parámetros de las consultas	112
Figura 76: Tercer panel relacionado con el comparador interno de <i>LabelDescriptor</i> .	113
Figura 77: Cuarto panel relacionado con las operaciones con bases de datos.	114
Figura 78: Quinto panel relacionado con las consultas en base a etiquetas con una sola posición.	114
Figura 79: Quinto panel relacionado con las consultas en base a etiquetas con una combinación de dos posiciones.	115
Figura 80: Lista desplegable de descriptores.	115
Figura 81: Cuadro de diálogo para seleccionar las imágenes a abrir.	116
Figura 82: Vista de las imágenes abiertas en la aplicación.	117
Figura 83: Vista del resultado del primer ejemplo de consulta.	118
Figura 84: Vista del cuadro de diálogo para abrir un fichero de base de datos.	119
Figura 85: Vista del resultado del segundo ejemplo de consulta.	120
Figura 86: Vista del resultado del tercer ejemplo de consulta basado en etiquetas con una posición.	121
Figura 87: Vista del resultado del cuarto ejemplo de consulta basado en etiquetas con dos posiciones.	121

Parte I

Sistemas de recuperación enfocados en términos lingüísticos

Capítulo 1

Introducción

Actualmente el uso de las imágenes está tan generalizado que es difícil encontrar una actividad en la que no haya imágenes involucradas. El hecho de que comenzase su digitalización ha originado la existencia de grandes bases de datos que almacenan diversas colecciones de imágenes de cualquier género. Por esta razón es crucial invertir cierto esfuerzo en desarrollar herramientas que sean capaces de lidiar con estos inmensos volúmenes de datos. Para ello la principal operación que deberán permitir consiste en recuperar un conjunto de imágenes en función de una serie de parámetros específicos. Con el fin de ejecutar esta acción se deberá extraer información descriptiva sobre las imágenes para, posteriormente, compararla con los datos descriptivos concretados por el usuario.

Los sistemas de hoy en día solo permiten recabar información acerca de las propiedades visuales de una imagen de manera global, lo que causa una considerable pérdida de precisión a la hora de realizar una consulta. Por ejemplo, si el usuario establece como imagen consulta aquella en la que aparecen dos objetos con distintas cualidades gráficas, tras realizar la comparación con un conjunto de imágenes almacenadas en una base de datos, la lista de imágenes resultante mostrará aquellas fotografías cuyas propiedades visuales se asemejen más a las características descriptivas del objeto que más destaque en la escena. Además, el uso exclusivo de propiedades gráficas para extraer los datos descriptivos de una imagen supone un claro inconveniente si tenemos en cuenta la diversidad de formas, colores y texturas que puede tener un mismo objeto.

Por todo lo explicado anteriormente este proyecto, en primer lugar, se enfocará en extraer los datos representativos de cada imagen de manera local con el objetivo de posibilitar un análisis de las propiedades de todos los objetos que aparezcan en la escena de manera individualizada. También introduciremos un novedoso procedimiento por el cual será posible reconocer los distintos objetos de la escena a partir de etiquetas lingüísticas, generadas por una *Red Neuronal Convolucionada* o CNN, que representen qué tipos de objetos son, independientemente de su aspecto físico. Además se añadirán diversas métricas para realizar consultas con diferentes resultados en función de lo que deseemos obtener. Por último, estas medidas junto con el proceso de etiquetado nos permitirán realizar búsquedas de objetos concretos en bases de datos de imágenes, a las cuales se les podrá, además, añadir la posición específica en la que queremos encontrar el

susodicho elemento.

1.1. Motivación

Los sistemas multimedia han progresado incrementalmente en su desarrollo convirtiéndose en poderosas herramientas capaces de extraer información descriptiva a partir de las propiedades visuales de las imágenes. En la época en la que surgió este procedimiento los resultados que proporcionaba eran más que suficientes para responder a las necesidades de entonces. No obstante, pese a que este procedimiento está hoy en día ampliamente extendido y es muy utilizado, se ha observado que esta técnica presenta una serie de inconvenientes que se han ido acentuando a consecuencia de la mayor exigencia en las demandas de los usuarios.

La primera desventaja reside en la escasa precisión de los datos extraídos puesto que estos se obtienen a partir de las características gráficas de una imagen completa. De esta forma las posibilidades de concreción en cuanto a una operación de recuperación son bastante limitadas puesto que debes ceñirte al conjunto de la escena de la imagen, no a los objetos que participan en ella. A este problema se le suma el inconveniente de la ambigüedad que representa la información obtenida en base a propiedades visuales. Y es que, teniendo en cuenta la amplia diversidad del aspecto físico del que disponen la gran mayoría de los objetos que nos rodean, podemos imaginar que probablemente realizar una consulta en base a, por ejemplo, su color medio no sea el mejor método para obtener unos resultados precisos.

Estos inconvenientes han motivado el desarrollo, desde hace poco tiempo, de otras técnicas innovadoras consistentes en identificar y dotar de etiquetas lingüísticas a los objetos de una imagen, independientemente de sus cualidades físicas. Por ello mi proyecto hará uso de esta metodología para independizar los resultados de una consulta, realizada en una base de datos de imágenes, de las cualidades gráficas de estas. Además se aplicará este procedimiento de manera local a la imagen con el objetivo de brindar la oportunidad de concretar más los resultados de dicha operación, puesto que así será posible identificar los distintos objetos que aparezcan en ella.

Además de lo anterior se explicarán las distintas medidas desarrolladas para realizar diversas consultas en función de los resultados que se deseen obtener. De esta forma habrá medidas cuyo principal objetivo sea el de limitar el conjunto de imágenes resultado de modo que estas tengan un grado de similitud muy parecido a la imagen consulta. O por el contrario se podrán realizar búsquedas superficiales de los objetos de una imagen consulta en la base de datos de imágenes. Todo ello irá acompañado de las pertinentes interpretaciones que estas medidas de distancia entre imágenes están sujetas.

1.2. Objetivos

Este proyecto se basa en el desarrollo de un sistema enfocado en la descripción local de imágenes utilizando los términos lingüísticos asociados a las susodichas áreas locales de las propias imágenes. Este procedimiento será aplicado tanto al ámbito de la descripción como al de consulta de imágenes. A continuación detallaremos los objetivos concretos que abordaremos en este proyecto:

1. Revisar el estado del arte en referencia a las arquitecturas CNN para determinar el etiquetado automático de las imágenes.
2. Desarrollar descriptores visuales enfocados en las áreas locales de las imágenes

- y basados en los términos aprendidos por la arquitectura CNN.
3. Integrar dichos descriptores en la biblioteca JMR (Java Multimedia Retrieval) de software libre.
 4. Desarrollar un sistema de recuperación de imágenes (CBIR) basado en los descriptores explicados anteriormente.

1.3. Estructura de la memoria

El proyecto ha sido dividido en dos diferentes bloques principales.

En el primero de ellos se tratarán todos los aspectos teóricos relacionados con las *Redes Neuronales Convolucionadas* o *CNN* que son las herramientas específicas que nos permitirán identificar los objetos que aparecen en las imágenes. En el siguiente capítulo será explicado, de manera detallada, el tema relacionado con los sistemas de recuperación de imágenes. Para ello se comenzará con una introducción sobre cómo han ido evolucionando a lo largo de la historia para después centrarnos en las capacidades existentes de los actuales y en las mejoras implementadas en este proyecto. A continuación se detallarán las diversas formas de estudio aplicadas a una imagen adjuntando ejemplos ilustrativos de cada una de ellas. Proseguiremos este proyecto con el ámbito relacionado con las métricas implementadas que han sido diseñadas con el fin de brindar diferentes posibilidades a los usuarios para realizar consultas en grandes bases de datos de imágenes obteniendo distintos resultados en función de lo que deseen. De nuevo para cada medida de distancia entre imágenes se adjuntará un ejemplo representativo del tipo de resultados que pueden proporcionar. Así mismo se continuará mostrando el potencial de la búsqueda de objetos implementada a través de sus etiquetas y con la posibilidad de restringir la aparición de dichos elementos a una posición determinada. Por último se explicará el proceso de creación de una base de datos propia, la cual ha sido requerida para demostrar el funcionamiento tanto de las consultas a través de imágenes como a través de los términos lingüísticos acompañados de una determinada localización.

El segundo bloque estará destinado al desarrollo del prototipo CBIR en el que se explicarán los requisitos que debía abordar y las novedosas extensiones implementadas que aporta. Así mismo se incluirán otros aspectos relacionados con la implementación del software, tales como los casos de uso, el diseño construido y la implementación. Para terminar con este bloque se presentará un manual de usuario con el objetivo de que cualquier persona puede utilizar este prototipo CBIR explorando todas las posibles operaciones que se encuentran disponibles.

Capítulo 2

Redes Neuronales Convolucionadas (CNN)

De forma introductoria y con carácter previo a la explicación acerca de las *Redes Neuronales Convolucionadas*, de las cuales utilizaremos una para la identificación de los objetos mediante etiquetas lingüísticas, procedemos a continuación a describir los aspectos esenciales de las *Redes Neuronales* de manera general.

2.1. Redes Neuronales

Su origen reside en el espectacular funcionamiento que demuestra el cerebro humano por estar compuesto de un conjunto de neuronas que trabajan de forma independiente y paralela con el objetivo de realizar tareas complejas simulando el trabajo en equipo. Además de esta cualidad, el desarrollo de esta parte de la computación también fue motivado por el hecho de que las susodichas neuronas tienen la capacidad de aprender.

En base a lo detallado anteriormente podemos entender el concepto de Red Neuronal como un sistema computacional que intenta imitar el potencial del aprendizaje humano basado en la interconexión de las neuronas de nuestro cerebro.

Una de las grandes ventajas que esconden este tipo de sistemas es que para hacer uso de ellos no hace falta programarlos desde cero, basta con entrenarlos en base a un gran volumen de ejemplos con el objetivo de que, posteriormente, sean capaces de reconocer diferentes muestras de los conceptos que han aprendido [1].

Existen dos similitudes entre la estructura neuronal humana y la computacional. La primera de ellas se refiere a que la composición de ambos sistemas está fundamentada en las interconexiones de elementos más pequeños capaces de ocuparse de una tarea diferente. Estas piezas son aquellas que conocemos con el nombre de neuronas, las cuales disponen de una información de entrada que reciben y de unos datos de salida que generan con el objetivo de que se transmitan a la siguiente neurona. Este hecho ocurrirá solamente si el estímulo a la neurona que tiene la información es lo suficientemente fuerte como para generar la energía necesaria para realizar esta acción.

Las neuronas biológicas están formadas por tres principales componentes: las dendritas cuya tarea fundamental es la de recopilar los impulsos eléctricos que viajan a través de toda la red neuronal para transmitirlos al núcleo de la neurona. Este último también forma parte de los tres elementos básicos, el cual se encarga de agrupar los susodichos impulsos nerviosos con el fin de procesarlos. Esta información recibida puede producir en la neurona un efecto estimulatorio o inhibitorio. Independientemente del impacto que produzcan los datos de entrada cuando hay una cantidad suficiente de ellos el núcleo genera un impulso con el objetivo de transmitir la información que retiene a la siguiente neurona. Para ello dispone de un tercer elemento denominado axón, el cual es considerado como la extensión del núcleo cuya tarea consiste en transmitir la esa señal generada por el núcleo hacia otra neurona con la que esté conectada. Esta transmisión de los impulsos eléctricos, denominada sinapsis, es posible gracias a la existencia de diversas interconexiones entre las dendritas de unas neuronas con los axones de otras. A continuación adjunto una imagen ilustrativa que representa un esquema de la estructura de dos neuronas biológicas con los elementos descritos anteriormente [1] [3].

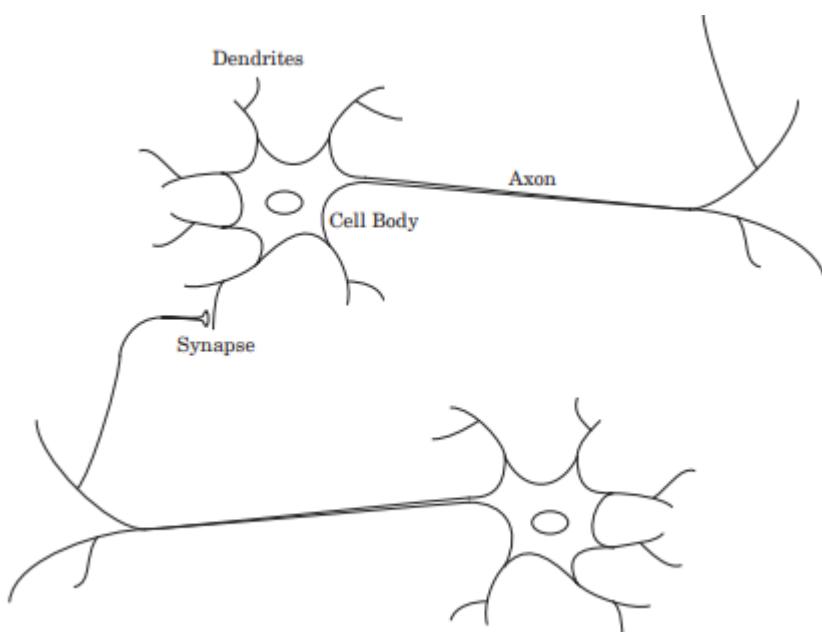


Figura 1: Esquema biológico de dos neuronas.

A partir de la explicación anterior sobre el funcionamiento básico de las redes neuronales biológicas se puede proceder a extraer todos los términos descritos al campo de las redes neuronales computacionales. En este ámbito una neurona es una unidad de procesamiento que dispone de un conjunto de funciones y que posee un estado interno denominado *estado de activación* [2]. Así mismo es capaz de recibir señales gracias a que se encuentra interconectada con otras neuronas que componen la red. Cada conexión existente está dotada de un conjunto de pesos que indican la influencia que tiene cada enlace entre dos pares de neuronas [3].

2.1.1. Aprendizaje

Este aspecto es el más relevante en el ámbito de las redes neuronales, puesto que dependiendo del modelo de aprendizaje que se haya establecido el sistema será capaz de abordar un tipo de problemas u otro. Además la habilidad de la que disponga la red para resolver los ejercicios que se le planteen también dependerá del tipo de entrenamiento al

que haya sido sometida.

El procedimiento que se sigue para entrenar a una red neuronal generalmente consiste en proporcionarle una batería de diversos ejemplos, pertenecientes al ámbito del que deseemos, posteriormente, que pueda resolver problemas. Así posibilitamos la modificación de los pesos relacionados con todas las conexiones que posea hasta que se cumpla alguno de los siguientes criterios de convergencia.

- Se establece un número determinado de ciclos de aprendizaje, es decir, se fija un número concreto de veces en las que se le va a introducir el conjunto completo de entrenamiento. Una vez finalizado este ciclo la red se da por entrenada.
- Mediante un valor de error. Para ello se deberá plantear una función de error y un valor para el susodicho de modo que se detenga el proceso de aprendizaje cuando la red sea capaz de producir un error menor que el establecido. El problema de este criterio es que podría darse el caso de que la red nunca llegase a alcanzar un error menor. Es por ello por lo que, generalmente, se le incluye un segundo parámetro de parada como por ejemplo un número determinado de ciclos. Si este segundo criterio ha de ser usado entonces podemos concluir que la red no ha llegado a originar una solución óptima y por lo tanto se debe volver a empezar el mismo procedimiento realizando ciertos ajustes.
- Cuando no surta efecto las modificaciones de los pesos. En este caso se detiene el proceso de aprendizaje pues, independientemente de los ejemplos propuestos, los pesos de sus conexiones no varían.

Un aspecto a tener en cuenta para que el entrenamiento pueda ser considerado como exitoso es que los ejemplos con los que se entrena la red cumplan las siguientes cualidades:

- El volumen de ejemplos debe ser lo suficientemente extenso como para que la red sea capaz de aprender el patrón común a todos ellos.
- Deben ser lo suficientemente variados como para que la red no se sobreajuste a los datos del conjunto de entrenamiento. Un ejemplo representativo que explica este concepto podría ser el siguiente. Supongamos que se desea entrenar una red neuronal para predecir la dirección de los futuros incendios. Para entrenarla se le proporcionan datos relacionados con algunos incendios ocurridos en el pasado, tales como la dirección del viento, la temperatura, entre muchos otros. Si llevamos a cabo el entrenamiento con el objetivo de que la red genere muy buenos resultados con el conjunto de ejemplos de entrenamiento se adaptará demasiado a los datos de estos y perderá la capacidad de generalización que necesita para realizar buenas predicciones futuras [2].

2.1.2. Arquitecturas

Continuaremos explicando el funcionamiento de las redes neuronales detallando cada una de las etapas y de los componentes que intervienen en el proceso relacionado con la transmisión de la información entre las neuronas. El primer paso previo al inicio de tal operación está relacionado con la recepción de los datos de entrada procedentes de otra

neurona. La información recibida a partir de la conexión de dos neuronas albergará un parámetro denominado peso cuyo principal objetivo es el de exaltar o inhibir a la neurona receptora. Tanto la información recibida como los pesos pasarán a ser procesados por la *función de propagación* que alberga la neurona receptora. La fórmula matemática del prototipo de la función de propagación se puede observar a continuación:

$$net_j = f_{prop}(d_{i1}, \dots, d_{in}, w_{i1j}, \dots, w_{ijn})$$

Tal y como se puede comprobar los argumentos que recibe se componen de los datos enviados por distintas neuronas y la relevancia, en forma de peso, que está asignada a la sinapsis por la que se han transmitido dicha información entre ellas.

Tras recibir los parámetros adecuados, la función de propagación se encarga de realizar la sumatoria del producto entre los datos recibidos de cada una de las neuronas por el peso de la conexión existente entre la neurona receptora y la que le envía la información. La fórmula matemática de este proceso se puede observar a continuación:

$$net = \sum_{i \in I} (d_i \times w_{ij})$$

Una vez calculado el valor resultante de la operación anterior se procede a ejecutar la siguiente fase en la que se encuentra una nueva función denominada *función de activación* o *función de transmisión*. Esta función se encarga de asignar el valor correspondiente al estado de activación que posee la propia neurona mediante un procedimiento en el que interviene un segundo componente conocido como *umbral*. Este es asignado de forma exclusiva a cada neurona que lleva a cabo esta función y representa el valor máximo que puede ser generado en la función de propagación. Es decir, el valor del umbral determinará cuántos datos puede almacenar cada neurona antes de transmitirlos a la siguiente. La fórmula matemática que representa su forma de proceder a realizar dichos cálculos se puede comprobar a continuación:

$$a_{j(t)} = f_{act}(net_{j(t)}, a_{j(t-1)}, \theta_j)$$

Así podemos observar que dispone de tres parámetros, el primero de los cuales obtiene el valor de la función de propagación para el dato actualmente recibido en la *neurona j*. El segundo se corresponde con el valor previo del estado de activación asociado a la misma neurona y, por último, el tercer argumento se corresponde con el umbral asignado de forma exclusiva a esta.

Cabe destacar que esta función es comúnmente compartida por todas las neuronas de la red o por un conjunto de ellas puesto que el proceso que permite la transmisión de información es el mismo para todas. No ocurre lo mismo con el umbral, ya que como hemos comentado anteriormente, es único para cada neurona y además está sujeto a cambios a consecuencia, por ejemplo, de un proceso de aprendizaje.

Con el fin de estructurar de manera clara e ilustrativa las etapas explicadas anteriormente se adjunta la siguiente figura en las que aparece cada una de ellas en el orden en el que se llevan a cabo [1].

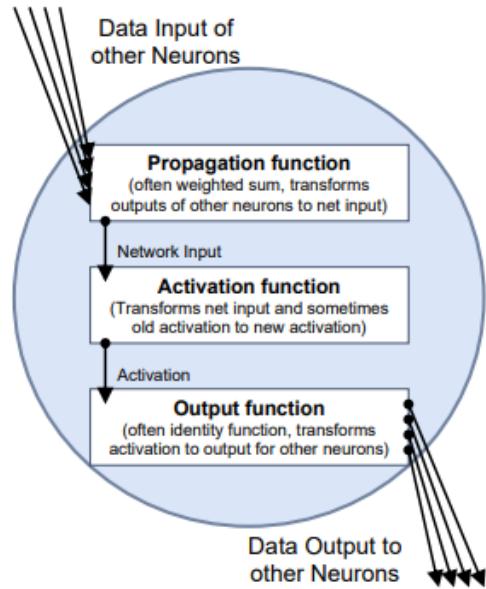


Figura 2: Etapas principales para transmitir información entre dos neuronas.

La segunda similitud entre una neurona biológica y una perteneciente a una red neuronal del ámbito de la computación, afirma que el funcionamiento del sistema completo dependerá del modelo de conexión por el que se relacionan las neuronas. Además el modelo de la arquitectura de la red neuronal también afecta al tipo de algoritmo que se utilizará para entrenarla [4].

Por lo general una red neuronal está principalmente compuesta por una capa de entrada que se encarga de recibir los datos, una capa de salida a la que se le encomienda mostrar los datos resultantes y entre medias de estas dos existen un conjunto de capas ocultas que serán las que calculen los resultados en función de los parámetros y los datos proporcionados [3].

Las tres arquitecturas principalmente conocidas y utilizadas se detallan a continuación.

2.1.2.1. Una sola capa

Esta es la arquitectura más sencilla que puede llegar a tener una red neuronal. En ella las neuronas se dividen en capas, en particular esta arquitectura solo cuenta con una capa en la que puede haber multitud de neuronas. Ellas serán las encargadas de realizar los cálculos computacionales necesarios para generar los resultados a partir de los datos de entrada. Además, cabe destacar que toda esta información es transmitida al completo a cada una de las neuronas de la capa de manera equitativa.

A causa de que cada capa alberga a todas las neuronas que la conforman, también incluye los diferentes elementos que las componen, muchos de los cuales ya hemos explicado con anterioridad, tales como los pesos para cada conexión entre dos neuronas y las funciones de transmisión de cada una de ellas. Por lo tanto cada una seguirá el procedimiento integrado en su función de transmisión y posteriormente aunarán los resultados obtenidos de cada dato de entrada en uno solo.

A continuación se mostrará una figura que ilustre esta arquitectura para aclarar la estructura que conlleva.

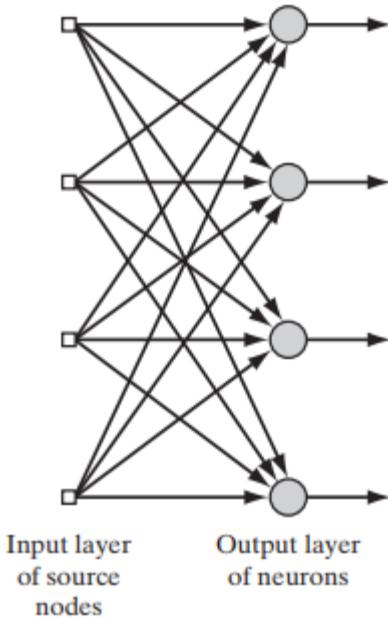


Figura 3: Ejemplo de arquitectura de una sola capa.

Si bien esta red cuenta con dos capas de neuronas, tal y como se puede comprobar, la primera destinada a recibir los datos de entrada no consta en el cómputo de capas total puesto que en ellas no se realiza ningún tipo de cálculo, su tarea principal es la de obtener la información de entrada [3] [4].

2.1.2.2. Múltiples capas

En esta segunda arquitectura intervienen una o más capas, en las que cada una puede disponer de un número diferente de neuronas. Dichas capas son conocidas como capas ocultas. Este adjetivo se les ha vinculado por el hecho de que no son perceptibles desde la entrada de datos o desde la salida de los resultados. El motivo de peso por el que añadir más capas intermedias entre la entrada de los datos y la salida está directamente relacionado con el propósito de posibilitar que la red neuronal se confeccione una visión global del problema. De este modo, la red neuronal será capaz de extraer un mayor rango de información útil, puesto que cuenta con un mayor número de sinapsis entre sus neuronas, con el objetivo de que le ayude a generar unos resultados más precisos.

A consecuencia de la existencia de varias capas, el proceso de recepción de los datos de entrada varía en esta arquitectura. Y es que, tal y como podremos observar en la siguiente figura que representa un ejemplo de esta, la capa que recoge los datos de entrada solo está conectada con la primera capa de cálculo. Es por ello por lo que las siguientes capas a partir de esta última recibirán los resultados obtenidos de las anteriores capas, por lo que esto simboliza una clara diferencia con la arquitectura anterior. En este caso la primera capa será la única que reciba la información original, mientras que el resto recogerán los resultados de las capas situadas en el nivel inmediatamente inferior. Con este procedimiento la red neuronal tiene la potestad de ir refinando y mejorando los resultados conforme se vayan transmitiendo a la siguiente capa. Una vez situados en la capa final, los cálculos que proporcione esta serán los resultados finales al problema planteado.

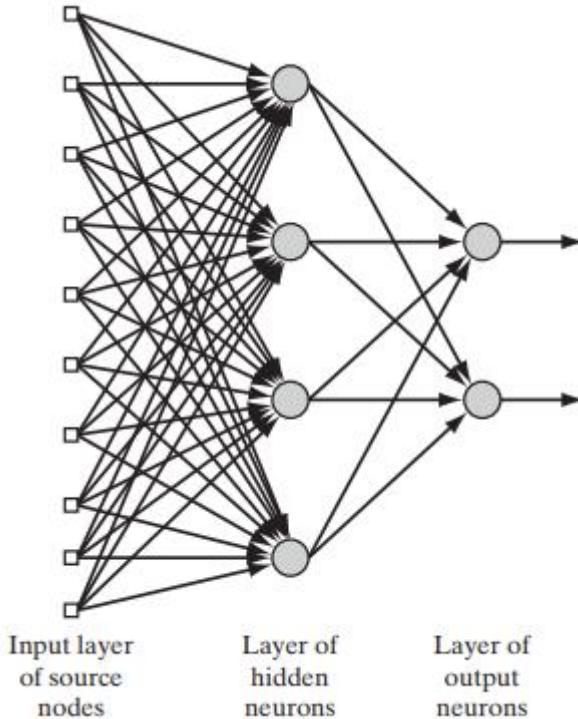


Figura 4: Ejemplo de arquitectura multicapa.

En la figura anterior podemos comprobar un ejemplo ilustrativo del modelo de arquitectura que nos ocupa, concretamente este cuenta con una capa de entrada, que recibe un total de diez datos, los cuales son transmitidos a la única capa oculta, que cuenta con hasta cuatro neuronas. Por último su capa de salida, compuesta por dos neuronas, son las encargadas de aunar los resultados de la capa oculta anterior para mostrar las soluciones calculadas. Este tipo de red neuronal es más bien conocida como una red 10-4-2, debido a su composición.

Cabe destacar que este ejemplo tiene la peculiaridad de que cada una de sus neuronas está conectada con todas las demás. A este modelo en particular, dentro de las arquitecturas multicapa, se le conoce como una *red neuronal completamente conectada*. Si por el contrario alguna de sus neuronas no está conectada con las que componen la capa adyacente, ya sea la capa anterior o la siguiente, se le denominará *parcialmente conectada* [4].

La principal similitud entre esta arquitectura y la anterior con una sola capa es que ambas deben ajustar el número de neuronas al problema que van a abordar. Así sus capas de entrada constarán de tantas neuronas como datos le proporcione el problema, y tantas neuronas en sus capas de salida como posibles resultados puedan originarse.

No obstante, el dilema se presenta en el ámbito de la arquitectura multicapa cuando está compuesta por más de dos capas. En este caso no hay ningún patrón o medio que pueda ayudarte a decidir el número de neuronas que deben conformar cada capa intermedia. Es aún, de hecho, un problema en el que todavía se está investigando con el objetivo de obtener el número de neuronas óptimo para cada capa oculta. Sin embargo la gran mayoría de redes neuronales que utilizan una arquitectura multicapa generalmente suelen disponer de dos capas o tres si el problema es más complejo [3].

2.1.2.3. Recurrentes

La principal diferencia entre este tipo de arquitectura y las dos explicadas anteriormente es que este modelo integra su propia retroalimentación, es decir, sus neuronas tomarán como datos de entrada aquellos resultados que han generado. Este procedimiento no era permitido en las otras arquitecturas, puesto que siempre transmitían los datos hacia las siguientes capas adyacentes.

Con el objetivo de entender el funcionamiento y la influencia que puede llegar a tener este proceso de retroalimentación, procedo a exponer dos ejemplos en los que cada uno estará compuesto de una estructura diferente. El primero de ellos, tal y como se puede observar en la figura que ilustra su composición, consta de una única capa de neuronas destinadas a realizar los cálculos pertinentes. Cada una de ellas tomará como datos de entrada las salidas resultantes de las otras neuronas. Por lo tanto, en este ejemplo la retroalimentación se refleja en el hecho en que cada neurona reutiliza los datos obtenidos de las demás pero no los suyos propios.

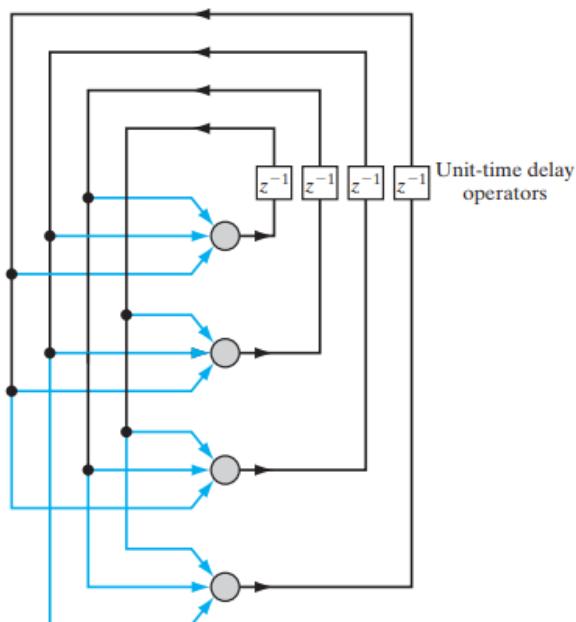


Figura 5: Primer ejemplo de arquitectura recurrente con retroalimentación.

En este segundo ejemplo el modelo consta de una única capa oculta compuesta por cuatro neuronas en la que, a diferencia del ejemplo anterior, la retroalimentación sí incluye la transmisión de los propios resultados que originan tanto a las entradas de las otras neuronas como a las suyas propias [4]. Este hecho incluye la necesidad de añadir un nuevo conjunto de elementos conocidos como *unidades de tiempo retardado*, los cuales estarán presentes en todas las neuronas de la capa de entrada, tal y como se puede comprobar en la siguiente imagen ilustrativa de este ejemplo particular. Su principal objetivo es producir un espacio temporal de espera para asegurarse de que todas las neuronas han acabado de calcular sus salidas antes de transmitirlas como entradas [3].

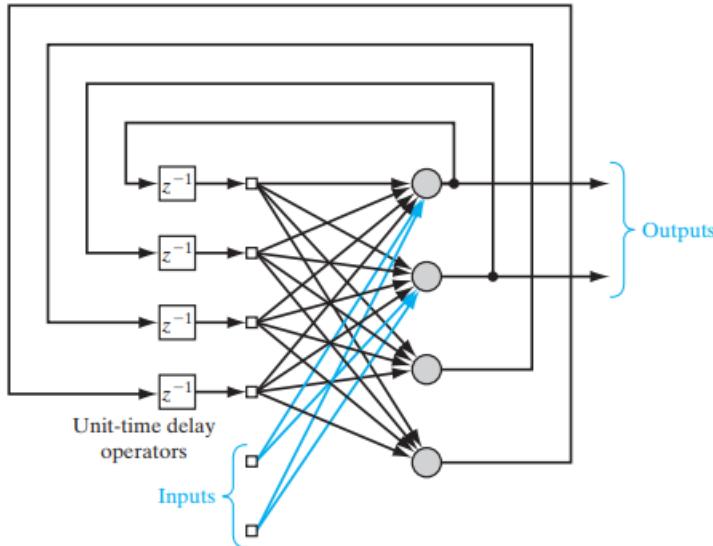


Figura 6: Segundo ejemplo de arquitectura recurrente con retroalimentación.

2.2. Redes Neuronales Convolucionadas (CNN)

Tras haber introducido los aspectos fundamentales de las redes neuronales con carácter general, procedo a explicar este tipo particular de red. Las CNN son redes neuronales con una arquitectura multicapa específicamente diseñadas para trabajar con datos cuya estructura sea similar a la de una cuadrícula, como por ejemplo una imagen cuyos píxeles están almacenados en una estructura de datos bidimensional [5]. Su principio básico está fundamentado en la introducción de una novedosa mejora que permite ignorar las transformaciones que puede sufrir una imagen con el objetivo de identificar el objeto independientemente de la forma que presente.

Un ejemplo ilustrativo que explique claramente este principio podría ser el caso que se ocupa de reconocer los dígitos numéricos. Los datos de entrada se corresponden con un conjunto de imágenes de los estos y la salida esperada es la identificación de cada uno de ellos. Para que la red sea capaz de reconocer un dígito independientemente de su aspecto se le debe preparar un entrenamiento con un conjunto de imágenes suficientemente amplio y variado como para que aprendiese a reconocer cada dígito sin importar las variaciones que este presente.

La estructura que compone a una CNN se puede observar en la siguiente figura, en la cual la segunda capa que ilustra se corresponde con la capa encargada de la convolución. Esta divide a la imagen en un conjunto de regiones con el objetivo de extraer las características de cada una de ellas. De este modo se aplica un novedoso método que consiste en obtener las propiedades de las regiones locales a la imagen en base a un conocido principio. Este afirma que aquellos píxeles que estén a una menor distancia muestran una mayor correlación, es decir, tienen un grado mayor de similitud por el hecho de encontrarse en una misma región de la imagen. Es por ello por lo que en el procedimiento que ejecuta esta capa los pesos serán los mismos a la hora de comparar las cualidades extraídas de una misma región.

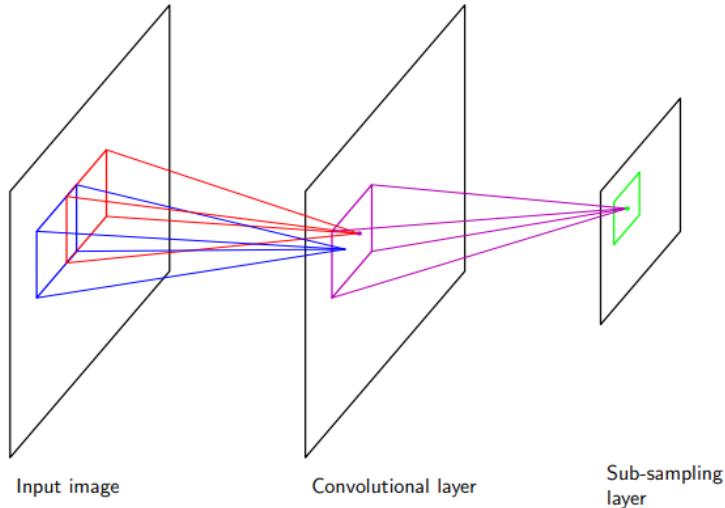


Figura 7: Estructura de una CNN.

Tras cada capa de convolución existe una capa computacional que es la encargada de realizar un submuestreo y calcular el valor promedio para cada una de las regiones de la imagen. Para ello esta última capa toma como entradas los valores resultantes de la capa de convolución y los divide en unidades de menor tamaño para calcular el promedio a menor escala. Cabe destacar que estas unidades más pequeñas son contiguas y no superpuestas, por lo que cada píxel solo formará parte de una de estas unidades. Con esta fragmentación adicional a cada una de las regiones se pretende reducir, aún más, la sensibilidad hacia las variaciones que puedan presentar ciertas regiones de las imágenes. De esta forma conseguimos que la CNN ignore estas peculiaridades y sea capaz de reconocer qué es lo que hay en cada región de una imagen [6].

En este proyecto se va a hacer uso de un determinado modelo de CNN denominada *ResNet (Residual Networks)*, el cual fue ganador del concurso *ImageNet* de 2015. Su principal ventaja reside en el avance que se tuvo que desarrollar para que esta red neuronal cuente con hasta 150 capas. Si bien es cierto que el modelo que estoy utilizando solo dispone de 50 capas, ambos pueden llegar a reconocer hasta 1000 términos, entre los cuales se pueden encontrar desde diversas especies de aves hasta objetos cotidianos que nos rodean diariamente [7].

Capítulo 3

Sistemas de Recuperación

3.1. Sistemas de recuperación de imágenes.

El primer significado acuñado a este término data del 1992 cuando Kato explicó el objetivo de sus experimentos, el cual residía en la recuperación automática de imágenes de una colección a través de ciertas características gráficas, tales como el color y la forma. Posteriormente este concepto ha ido evolucionando hasta el momento actual en el que se le conoce como el procedimiento a seguir para recuperar las imágenes deseadas de una base de datos en función de un conjunto de propiedades visuales que se pueden extraer directamente de una imagen.

A raíz de esta idea surgieron los denominados *CBIR (Content-Based Image Retrieval Systems)*, un conjunto de sistemas cuyo objetivo fundamental es aplicar este procedimiento y resolver el inconveniente de asociar términos descriptivos a las imágenes. Este aspecto fue considerado un problema como tal debido a la dificultad de obtener información útil de las imágenes sin llevar a cabo un procesamiento sobre ellas, es decir, sin reconocer las propiedades ni los objetos representados en la escena de cada imagen. Además surgía un nuevo inconveniente relacionado con el tipo de información descriptiva que se debía obtener de una imagen en función de su futura utilidad. Para determinar este aspecto se comenzó a plantear los diversos usos que los usuarios de imágenes podrían otorgarle. Centrándonos en el ámbito profesional existen una multitud de trabajadores que a diario deben utilizar imágenes para llevar a cabo sus tareas.

- Policía. Esta es una de las profesiones en la que más ha influido el hecho de disponer de imágenes para registrar todo tipo de actividades vinculadas con el crimen. Una de sus principales utilidades reside en el hecho de fotografiar las evidencias de una escena criminal para su posterior análisis. Junto a ella cabe destacar también la posibilidad de fotografiar a todas aquellas personas que hayan cometido un acto delictivo con el objetivo de realizar un seguimiento sobre su situación actual y de realizar una posible identificación en base a las características físicas de un sujeto.
- Medicina. Es otro de los ámbitos en los que el uso de las imágenes ha

contribuido de manera muy positiva en la representación de los resultados de ciertas pruebas, tales como los rayos X. De esta forma se puede realizar un diagnóstico y seguimiento de la dolencia del paciente consultando todas aquellas imágenes asociadas y realizando comparaciones entre ellas para comprobar el progreso realizado.

- Moda y diseño. En los diversos campos relacionados con esta materia ha sido crucial la aparición de las imágenes para su desarrollo. En la mayor parte de las tareas se integran imágenes con distintos fines, tales como la representación de los bocetos en el proceso creativo o la representación de productos que se encuentran tanto en la fase de desarrollo como aquellos que ya han salido al mercado.
- Editorial. Las imágenes han sido popularmente utilizadas para complementar el contenido de libros, periódicos y revistas desde hace bastantes años. La mayor parte de las editoriales cuentan con sus propias bases de datos de imágenes, las cuales están catalogadas en función de la materia a la que pertenezcan. Es en el ámbito electrónico, sobre todo, en el que los sistemas de recuperación de imágenes adquieren una mayor importancia debido al hecho de obtener la imagen correspondiente en función de la temática de una manera eficiente y rápida.
- Publicidad. Es otro de los ámbitos relacionados con el tema de la información, en la cual es indispensable utilizar imágenes estáticas o en movimiento que representen el producto o el servicio que se está comercializando. Es por ello por lo que las denominadas bibliotecas comerciales que contienen imágenes de stock, como *Getty Images* y *Corbis*, están experimentando un crecimiento exponencial muy relevante [9]. Se entiende por una fotografía de stock una imagen distribuida bajo ciertos tipos de licencias para uso creativo. De esta forma el fotógrafo mantiene el *copyright* pero a su vez permite que terceros usen sus fotografías a cambio de los pagos correspondientes a la licencia bajo la que se distribuyen sus imágenes. Por lo tanto es una alternativa para que las empresas busquen imágenes acordes a su proyecto sin la necesidad de contratar a un fotógrafo específicamente y, por otro lado, es una oportunidad para los fotógrafos de obtener unos beneficios medianamente regulares por su trabajo [10].
- Arquitectura e ingeniería. En ambas profesiones las imágenes son usadas con el objetivo de representar el aspecto de los proyectos que han sido acabados, como por ejemplo las fotografías de los interiores y exteriores de los nuevos edificios. De esta forma se mantiene un registro del progreso alcanzado en las distintas etapas de construcción del proyecto con el fin de poder informar al cliente del estado del susodicho [9].

Debido al gran abanico de usos posibles que se le pueden dar a una imagen surge la necesidad de extraer un conjunto de datos que la describan adaptándose a su futuro uso. Este desarrollo conllevaba dos principales problemas referentes al hecho de obtener dicha descripción. El primero de ellos reside en la inversión tanto monetaria como temporal que implica el esfuerzo de aquellos expertos cuya tarea consiste en establecer los términos descriptivos de cada imagen. Actualmente esta posibilidad es inalcanzable debido a la gigantesca cantidad de imágenes que circulan por Internet o que están almacenadas en una base de datos.

El segundo inconveniente, relacionado con el anterior, se basa en la ausencia de objetividad a la hora de que un experto describa una imagen. Con esta afirmación quiero decir que cada persona puede tener una visión distinta de una imagen en base a su forma de pensar. Por lo que el error común que se comete a la hora de llevar esta tarea a cabo es que el experto vincule un término descriptivo en función de lo que piensa que la imagen representa.

A pesar de estas dificultades estos sistemas se crearon con la fiel convicción de que, algún día en un futuro no muy lejano, las máquinas fuesen capaces de reconocer los objetos que componen una imagen a través de términos lingüísticos. Este aspecto es el que se ha estado desarrollando y en el que se continúa investigando para dotar a estos sistemas de un proceso de etiquetado más versátil a la hora de precisar de qué objeto se trata [8]. Mi proyecto se centrará especialmente en esta parte con el objetivo de poder reconocer y etiquetar a los objetos que aparecen en la escena de una imagen sin importar cuáles sean sus propiedades gráficas.

La cuestión principal de estos sistemas reside en escoger una clase de atributos a partir de los cuales se pueda realizar una recuperación de imágenes en base a su contenido. Si bien es cierto que los CBIR utilizan algunos procedimientos habitualmente relacionados con el procesamiento de imágenes y la visión por computador, su principal diferencia reside en el hincapié que realizan sobre la acción de recuperar imágenes en función de unos parámetros especificados. Además el mundo del procesamiento de imágenes va mucho más allá albergando otros tipos de operaciones como la mejora de imágenes, la compresión y el análisis que les atribuye el significado en función de su contenido. No obstante, ambos sistemas en sus respectivas investigaciones y desarrollos comparten ciertos problemas tales como:

- El estudio de los usos que hace la población de las imágenes así como el método a aplicar para llevar a cabo una búsqueda en una colección en base a ciertos parámetros concretos.
- Establecer el proceso adecuado para extraer la información relevante de la imagen a partir de su contenido.
- Reducir el espacio requerido para el almacenamiento de las grandes bases de datos de imágenes.
- La recuperación y acceso eficiente a las imágenes almacenadas en función de su contenido.
- Diseñar interfaces de usuario para trabajar con sistemas CBIR [9].

En relación al funcionamiento general de cualquier sistema de recuperación de imágenes podemos afirmar que, al menos, se compone de cuatro etapas principales, las cuales se pueden observar reflejadas en la siguiente figura.

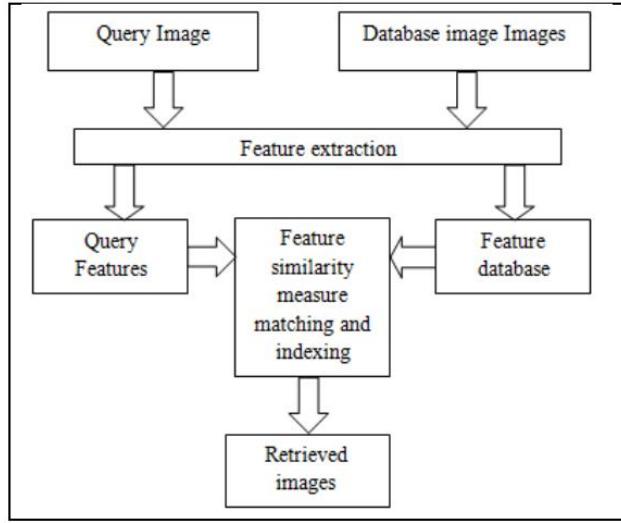


Figura 8: Flujo general de un CBIR.

La primera de ellas consiste en extraer la información descriptiva de las imágenes almacenadas en la base de datos en función de las propiedades que hayan sido seleccionadas para obtener dichos datos. Posteriormente se realizará el mismo procedimiento pero aplicado a la imagen consulta, la cual marcará el patrón a buscar entre las características seleccionadas a las imágenes de la base de datos. A continuación se procederá a realizar una comparación entre la imagen consulta y el resto de imágenes utilizando sus respectivas informaciones descriptivas en base a algún criterio que sea capaz de calcular la similitud entre la imagen consulta y el resto de imágenes. Tras haber realizado la comparación se presentarán las imágenes resultantes en un orden descendente que refleje un grado de similitud mayor cuanto menor sea la posición de la imagen. Es por ello que todas aquellas imágenes que ocupen los primeros puestos de la lista resultante serán clasificadas como las más parecidas a la imagen consulta en base a un determinada medida de distancia [11] [12].

Con el fin de almacenar la información descriptiva obtenida de una imagen se ha introducido un nuevo concepto denominado *descriptor*. Esta estructura de datos almacena un conjunto de valores resultante de aplicar algún tipo de operación a un conjunto de propiedades de una imagen. Por lo general las operaciones comúnmente aplicadas están relacionadas con la rama de las matemáticas y con las propiedades gráficas tales como el color, la textura o la forma. Un ejemplo representativo de este caso puede ser el cálculo del color medio con respecto a los valores *RGB* de la imagen.

No obstante también existen otra clase de descriptores cuya información, por ejemplo, está almacenada en forma de texto lingüístico.

Con el fin de establecer el contenido de los descriptores se llevaron a cabo diversos estudios de las múltiples propiedades de las imágenes que podían utilizarse para extraer la información descriptiva, tales como un patrón de color o textura concreto, la presencia de algunos objetos específicos en la escena de la imagen, una semántica concreta ligada al contenido que representa la imagen o simplemente basada en los metadatos de la propia imagen, quién la creó, dónde y cuándo [9]. A partir de estos resultados el profesor Eakins diseñó una clasificación aproximada de las propiedades de una imagen que se podrían utilizar para extraer la información descriptiva más relevante. De nuevo se fundamentó en las cuestiones que un usuario podía plantearse en relación a las cualidades de una imagen. Así la clasificación cuenta con

tres niveles principales de abstracción: rasgos primitivos, rasgos lógicos y atributos abstractos.

3.1.1. Rasgos primitivos

En este nivel las propiedades gráficas de la imagen se identifican con la materia prima que los CBIR utilizan para extraer la información descriptiva de una imagen. Las más comunes están directamente relacionadas con el color, la textura o la forma de los objetos que participan en la escena [9]. Gracias al progresivo desarrollo que se ha realizado en el campo de la visión por computador, hoy en día la gran mayoría de los sistemas de recuperación de imágenes pueden trabajar con este tipo de operaciones. Para ello se le brinda la posibilidad al usuario de establecer una imagen consulta con el objetivo de recuperar aquellas imágenes que se asemejen más a esta en función de las propiedades gráficas escogidas. En este ámbito ciertas características visuales proporcionan mejores resultados que otras, como por ejemplo el color. Esto se debe en mayor medida a la operación que realizan los CBIR consistente en calcular el histograma de color para cada imagen que participe en la consulta. Dicha operación fue introducida por primera vez en 1991 por *Swain* y *Ballard* y desde entonces se han ido introduciendo diversas mejoras que derivan en unos resultados mejores sobre todo cuando se integra con la localización espacial.

Además de los avances relacionados con el color también se han realizado evoluciones en el cálculo de la textura, la cual se extrae principalmente de propiedades visuales tales como el contraste, la rugosidad y la direccionalidad. Los resultados que proporciona esta cualidad gráfica son aún mejores si se combina con la propiedad anteriormente mencionada, el color.

Por último en este nivel también existe la posibilidad de reconocer las formas de los objetos que aparecen en las imágenes. Sin embargo el avance realizado no es comparable al progreso referente a las dos anteriores características gráficas puesto que este representa un desafío aún más complejo.

Algunos de los CBIR comerciales como QBIC y Virage son capaces de trabajar con estas medidas gráficas, bien de forma individual o de manera combinada, para realizar recuperaciones de imágenes. Un ejemplo representativo del tipo de consultas que se podría realizar con sistemas que utilicen estas medidas podría ser el siguiente: "Dado un conjunto de imágenes busque aquellas que alberguen cualquier tipo de objetos que tengan un color oscuro en una posición determinada, con una textura concreta o en aquellas cuyos objetos tengan una forma determinada" [9]. En particular, a continuación se muestra un ejemplo real de una consulta realizada en el sistema QBIC [1]. En las dos siguientes figuras podemos observar, respectivamente, a la izquierda la imagen consulta que se toma como referencia para llevar a cabo la recuperación de imágenes en una base de datos, y a la derecha su histograma de color, el cual será usado para realizar la comparación la imagen consulta y el resto de imágenes.

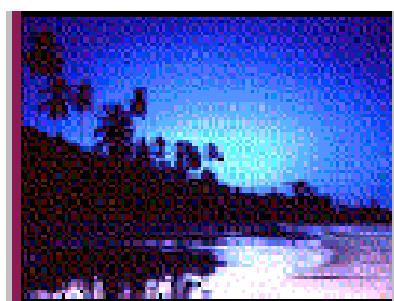


Figura 10: Imagen consulta.

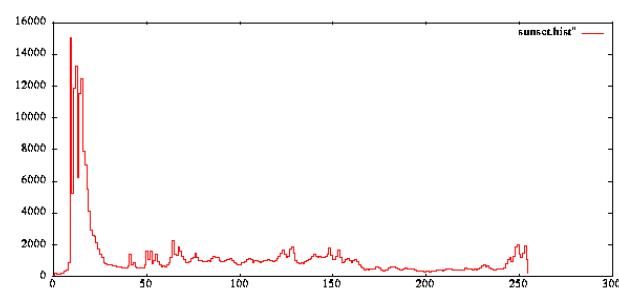


Figura 9: Histograma de color de la imagen consulta.

A continuación se presentan la lista de imágenes resultante de realizar la consulta en el sistema QBIC. Tal y como podemos comprobar, calculando el histograma de cada imagen en base al color y comparándolo con el de la imagen consulta, nos devuelve un resultado en el que todas las imágenes tienen en común el rango de tonos azules y negros que componen sus escenas pero en todas aparecen objetos de diverso tipo.

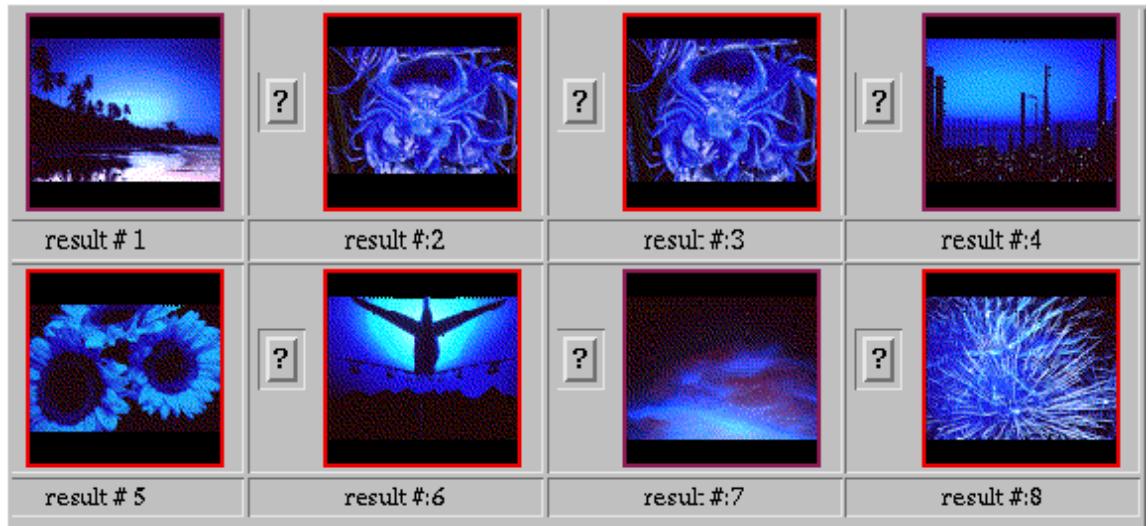


Figura 11: Lista de imágenes resultante tras la comparación, de mayor a menor grado de similitud.

Debido a la dependencia que genera el hecho de calcular los descriptores de las imágenes en función del proceso de extracción de sus propiedades visuales, estos sistemas cuentan con una ventaja y una desventaja. Por un lado este tipo de medidas son bastante objetivas puesto que solamente obedecen a las características del contenido de las imágenes sin la necesidad de utilizar una fuente externa de conocimiento [9]. No obstante, a su vez este hecho se corresponde con una desventaja puesto que el grado de precisión a la hora de obtener un conjunto de imágenes basadas en una semántica concreta es mínimo. Y es que, tal y como hemos podido comprobar en el anterior experimento, si realizamos una consulta en base a una propiedad gráfica, como por ejemplo el color, las imágenes resultantes se corresponderán con aquellas que cumplan los criterios visuales de la imagen consulta pero sin tener en cuenta los objetos que participan en esta ni su significado. Esta conclusión unida al aumento de demandas cada vez más exigentes por parte de los usuarios, los cuales ya desean que los sistemas tengan más en cuenta el contenido de las imágenes que sus cualidades visuales, fue lo que provocó el estudio de un segundo nivel de abstracción donde se comienza a introducir el concepto de la semántica en los CBIR [13].

3.1.2. Rasgos lógicos

En este segundo nivel se investiga la capacidad de inferencia sobre la identidad de los elementos que aparecen en la imagen con el fin de realizar recuperaciones de imágenes en base a un tipo determinado de objeto [9].

El primer CBIR que se destinó a llevar a la práctica este nivel fue el denominado *GRIM_DBMS* diseñado por *Rabbitti* y *Stanchev* en 1989, el cual era capaz de realizar sus propias interpretaciones dado un conjunto de dibujos con el objetivo de recuperar

aquellos dibujos lineales que perteneciesen a una categoría concreta. Un ejemplo representativo de este tipo de consultas podría ser aquel que buscase aquellas imágenes que se correspondiesen con planos de edificios de viviendas. Para ello el sistema analizaba los objetos que aparecían en los bocetos y les asociaba un conjunto de interpretaciones junto con la probabilidad de que estas fueran ciertas. De este modo se escogería aquella etiqueta que mayor número de posibilidades tuviese de representar el significado de una imagen. Pero a pesar de los continuos esfuerzos invertidos en este sistema y aunque su rango de interpretaciones era bastante limitado, no se han encontrado informes de resultados exitosos de recuperación de imágenes utilizando este sistema. No obstante se tomó como referencia del posible camino a seguir para desarrollar CBIR capaces de recuperar imágenes a partir de su semántica de forma automática.

A diferencia del nivel anterior, en este ámbito el progreso no ha sido tan notable puesto que surgen dos principales problemas. El principal está asociado con el reconocimiento de la escena de una imagen. Esta operación se puede llevar a cabo de manera global con el fin de identificar los objetos que aparecen en la imagen de manera generalizada, lo cual supone una importante ventaja a la hora de filtrar las imágenes sobre las que se realiza la consulta. Sin embargo también se puede realizar de forma local con el objetivo de identificar los elementos concretos que aparecen en la imagen.

El inconveniente reside en la complejidad de este tipo de consultas que no solo requieren un estudio de las cualidades físicas de los objetos sino, además, un proceso de inferencia para el cual se hace uso de una base de conocimiento externa que ayude a identificar un objeto en función de su apariencia. De nuevo vuelve a aparecer el principal problema que deriva de utilizar propiedades gráficas y que impide reconocer un elemento en base a su aspecto físico por sus múltiples formas, colores y texturas que puede llegar a tener. Es por esta razón por la que muchos expertos proponen el uso de términos lingüísticos de modo que se pueda simular, en cierta manera, la capacidad humana de identificar los objetos que nos rodean [13].

Un ejemplo de este tipo de sistema es el CBIR denominado *IRIS* que hace uso del color, la textura y la posición para averiguar cuál es el significado más probable de la escena de una imagen. En la siguiente figura podemos observar una captura del funcionamiento del sistema en cuestión, el cual está analizando la imagen de un paisaje natural compuesto por unas montañas nevadas a cuyas faldas se encuentran un bosque y una especie de lago. Debajo de la imagen podemos comprobar las etiquetas que genera el CBIR en base a las características visuales anteriormente mencionadas. En último lugar, en base a las interpretaciones locales que ha generado, proporciona una última etiqueta con la que representa el significado más probable de la imagen, y en efecto, la imagen muestra en su generalidad un paisaje con un bosque como principal protagonista.

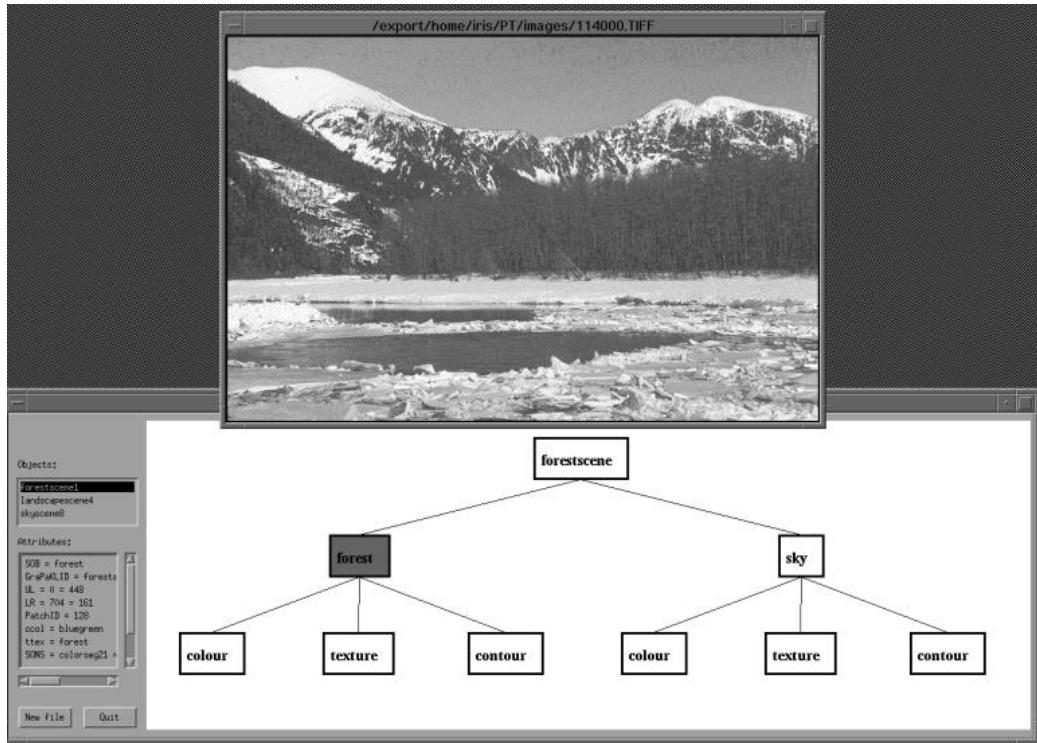


Figura 12: Ejemplo de funcionamiento de IRIS.

Así este sistema realiza un análisis gráfico para cada una de las posiciones especificadas averiguando cuál es el significado más probable para dicha región. Posteriormente se combinan las distintas interpretaciones locales para producir la etiqueta que represente el significado de la imagen. El último paso que realiza es convertir este término en descriptores lingüísticos, los cuales puedan usarse en sistemas de recuperación basados en texto.

Al igual que *GRIM_DBMS* este sistema realiza un procedimiento basado en un razonamiento complejo mediante una fuente de conocimientos externa, que ha sido previamente confeccionada por un experto, para permitir al sistema escoger la interpretación más probable a la escena que está analizando. Aunque sus buenos resultados dependen de que trabaje con un dominio más o menos restringido, este primer comienzo reforzó la idea de que los sistemas de recuperación de imágenes puedan utilizar un razonamiento deductivo automático similar al de los humanos para conseguir resultados más precisos [9] [13].

El segundo dilema al que se enfrenta esta variante de los sistemas CBIR es al consistente en recuperar imágenes de objetos o personas de manera individual. Un ejemplo de este tipo de consultas podría ser aquella que obtuviese imágenes en donde apareciese la torre Eiffel. En este caso también es necesario disponer de una base de conocimiento externa, pero a diferencia del ámbito anterior en el que para buscar un elemento determinado solo se necesitaba cierta habilidad para identificar algunos objetos, en esta segunda variante es además imprescindible el hecho de que pueda reconocer el objeto buscado mediante el nombre que se le ha asignado.

El sistema más famoso aplicado a este enfoque es aquel que fue diseñado por *Forsyth* en 1997 con el objetivo de reconocer un amplio rango de seres vivos tales como seres humanos, ciertas especies de animales como los caballos y hasta algunos tipos de árboles en función de la durabilidad de sus hojas como son los árboles de hoja caduca. Para el desarrollo de este sistema se han utilizado un

conjunto de técnicas que se apoyan en la idea de construir un modelo de cada objeto que deseamos que el sistema reconozca. De esta forma se le van presentando al CBIR diversas imágenes en las que aparezca dicho objeto utilizando, para su identificación, distintas propiedades visuales como el color, la forma o la textura, así como su posición y el fondo de la fotografía. Con todo ello se pretende dotar al sistema de la capacidad de discernir cuándo un objeto forma parte de la escena de una imagen y cuándo no.

A pesar de este conjunto de procedimientos automáticos diseñados para identificar los objetos que se muestran en una imagen existen, también, una serie de métodos en los que interviene el usuario de forma directa. El primer ejemplo de este tipo de sistemas que utilizaban una de las técnicas en cuestión fue desarrollado por *Minka* en 1996, el cual permitía al usuario señalar una parte de la imagen con el objetivo de asignarle la etiqueta que, con mayor probabilidad, se adaptase a la semántica de lo que se mostraba en dicha región. No obstante, para realizar este procedimiento este sistema seguía basándose en la comparación con otras imágenes que tuviesen características gráficas similares. Por ello, con el fin de mejorar sus resultados, también se brindaba la posibilidad a los usuarios de enviar sus comentarios acerca de los errores que el sistema cometía y de las interesantes mejoras que se podían añadir.

Un segundo sistema que aplicaba estas técnicas adaptativas permitía al usuario establecer un determinado patrón de color, textura o forma junto con la relevancia de que apareciese en mayor o menor medida en la imagen para confeccionar su consulta. Una vez el usuario estaba de acuerdo con los resultados se establecía una etiqueta en base a ellos que posteriormente se almacenaba en una base de datos con el objetivo de que, si algún usuario volvía a hacer la misma consulta, el sistema no tuviese que volver a calcularla repitiendo este costoso procedimiento. De esta forma la base de datos era capaz de albergar cada término lingüístico ligado a un conjunto de propiedades visuales con el objetivo de agilizar y reducir el tiempo invertido en un futuro cuando se ejecutaran las mismas consultas.

Generalmente los sistemas pertenecientes a este nivel de abstracción son los más usados comparados con los del nivel anterior, puesto que es particularmente más práctico el hecho de reconocer un objeto en base a la etiqueta con la que más probablemente se identifique su contenido que en función de sus propiedades gráficas. Este hecho se demostró en los numerosos estudios que realizó *Peter Enser* en 1995 en los cuales se reflejaba que en la mayoría de las consultas que recibían ciertas fuentes de información, tales como las bibliotecas de imágenes destinadas a aparecer en periódicos, pertenecían al tipo de consultas propias de este nivel de abstracción [13].

3.1.3. Atributos abstractos

La intención en este último nivel es la de aplicar la inferencia al máximo exponente con el objetivo de realizar recuperaciones de imágenes en base a las implicaciones personales que transmiten [8] [9]. Este hecho, por ejemplo, aplicado al color se traduce a la apreciación que una persona puede realizar para clasificar esta propiedad como un color frío o cálido o para determinar si dos colores distintos combinan bien.

La versatilidad que ofrece este nivel de razonamiento es lo que permite que este tipo de sistemas sean capaces de llevar a cabo consultas en las que, por ejemplo, se especifique un tipo concreto de evento o actividad del tipo “Buscar imágenes de bailes folklóricos de un país determinado”. Además este modelo de sistemas también es apto para realizar un segundo tipo de consultas en las se encuentran involucrados sentimientos, emociones o estados de ánimo de cualquier variedad. Un ejemplo

representativo que refleje este potencial podría ser el siguiente: “Recuperar aquellas fotos que reflejen sufrimiento”.

La principal desventaja de los CBIR que aplican las técnicas de este tercer nivel de abstracción reside en el grado de subjetividad que se esconde tras el hecho de vincular una etiqueta lingüística con el contenido de una imagen. La razón principal de esta afirmación se fundamenta en las directrices que un experto debe proporcionar al sistema para que este sea capaz de identificar lo que transmite una imagen. Este punto puede llegar a ser muy controvertido en tanto en cuanto distintas personas pueden apreciar diferentes emociones al observar una misma imagen. No obstante este tipo de consultas también se pueden encontrar en ciertas bibliotecas que albergan imágenes destinadas a aparecer en periódicos o en artículos relacionados con el arte, aunque con una frecuencia considerablemente menor que las consultas pertenecientes al nivel anterior [9].

3.2. Análisis de imágenes.

La razón por la que se siguen desarrollando diversas metodologías para analizar imágenes se basa en la gran cantidad de información que estas albergan. Para llevar a cabo un estudio sobre imágenes debemos introducir el ámbito de la *minería de datos aplicada a imágenes*. Este campo se enfoca en extraer información relevante de una imagen en función de una serie de patrones concretos o de los datos específicos que se deseen obtener. Si bien es cierto que aún queda mucho por investigar, la gran mayoría de sus aplicaciones se han centrado en los sistemas de recuperación de imágenes. Para ello se han desarrollado métodos capaces de analizar y recuperar aquellas imágenes, almacenadas en grandes bases de datos, que tengan un mayor grado de similitud en función de las características de una imagen consulta.

Dependiendo de las cualidades que se utilicen para realizar el estudio de imágenes podemos clasificar los CBIR actuales en dos principales categorías. La primera abarca todos aquellos sistemas que extraen de cada imagen un texto descriptivo, el cual puede llegar a almacenar diversas propiedades tales como su tamaño, el tipo de imagen que es, la fecha en la que se realizó, quién la tomó, palabras clave relacionadas con su contenido o una descripción de su escena. Un ejemplo representativo del tipo de consultas que se podría realizar con este CBIR podría ser la siguiente: “Recuperar aquellas imágenes que hayan sido tomadas el 8 de enero de 2015 y cuyo tamaño sea mayor de 100 Kb”. No obstante, este tipo de sistemas presenta una serie de desventajas que han interrumpido su desarrollo. Dos de los principales inconvenientes están directamente relacionados con la tarea que debe desempeñar una persona para asignarle una descripción acorde a la escena de la imagen. Tal y como hemos comentado con anterioridad, esto supone una falta de objetividad puesto que dos personas distintas pueden explicar la representación de una imagen de dos formas diferentes. Además debemos tener en cuenta que con el gran volumen de imágenes que hoy en día se puede llegar a almacenar en una base de datos, esta tarea resultaría casi imposible de realizar.

Es el segundo tipo de CBIR en el que más se ha estado investigando y desarrollando, puesto que estos sistemas realizan una consulta en una base de datos de imágenes en función de las características gráficas de estas. Algunas de las propiedades más utilizadas son el color, con el cual se puede calcular el histograma que representa las tonalidades de la escena de una imagen, la textura o la forma de los objetos que se encuentran en ella [14]. No obstante, tal y como hemos puntualizado anteriormente, este tipo de características pueden no ser suficientes para obtener unos resultados precisos, es decir, a causa de la diversidad que presentan la gran mayoría de objetos en cuanto a su apariencia.

Con el objetivo de demostrar esta teoría procedo a adjuntar un ejemplo ilustrativo utilizando el software que he desarrollado para este proyecto. Para ello tendremos que observar en la siguiente figura, en primer lugar, a la imagen consulta la cual se corresponde con la fotografía de una taza roja. La consulta que se va a realizar consiste en buscar, en una base de datos propia que he elaborado, todas aquellas imágenes que presenten una misma estructura del color que la que tiene la imagen consulta. Esto se traduce en que la lista de imágenes resultante que se espera es aquella que contenga una serie de fotografías que presenten tonalidades rojas, por el color de la taza, blancas, por el fondo de la imagen, o grises, debido al suelo de la base donde se realizó la foto.

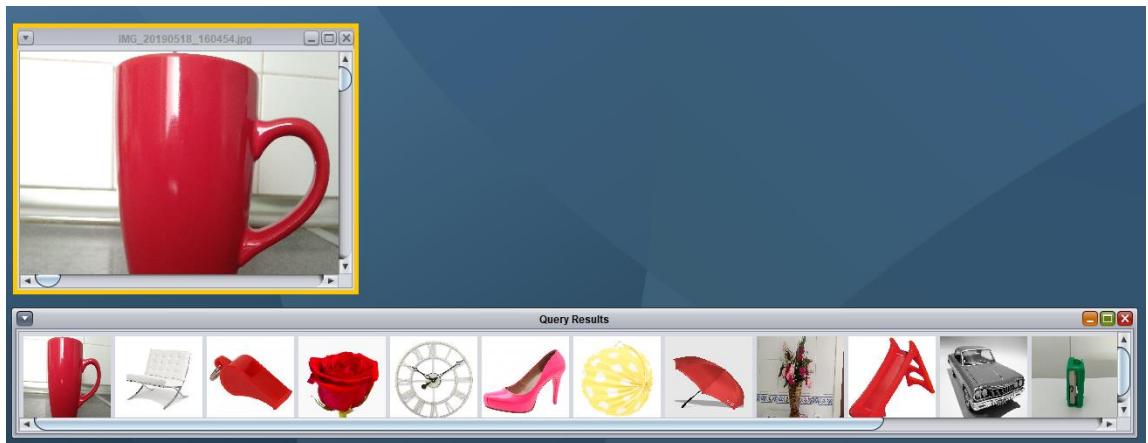


Figura 13: Primer experimento con un objeto taza extrayendo la estructura del color de forma global.

Tal y como podemos comprobar en la captura las imágenes recuperadas cumplen con las expectativas descritas previamente. Todas las fotografías que se han recuperado contienen alguno de los colores mencionados anteriormente, puesto que forman parte de la imagen consulta. No obstante podemos observar, a su vez, que las fotografías resultantes no tienen nada que ver con el elemento que se ha buscado, que en este caso es una taza. Todas ellas pertenecen a distintos tipos de objetos tales como un tobogán, una flor, un reloj, un silbato, entre otros.

Sin embargo este suceso no solo ocurre con este objeto, sino que esta tendencia está generalizada a cualquier tipo de elemento que busquemos. Un segundo ejemplo que demuestra esta teoría se puede ver a continuación, en el cual, en este caso, la imagen consulta se corresponde con una maceta rosa con plantas artificiales.

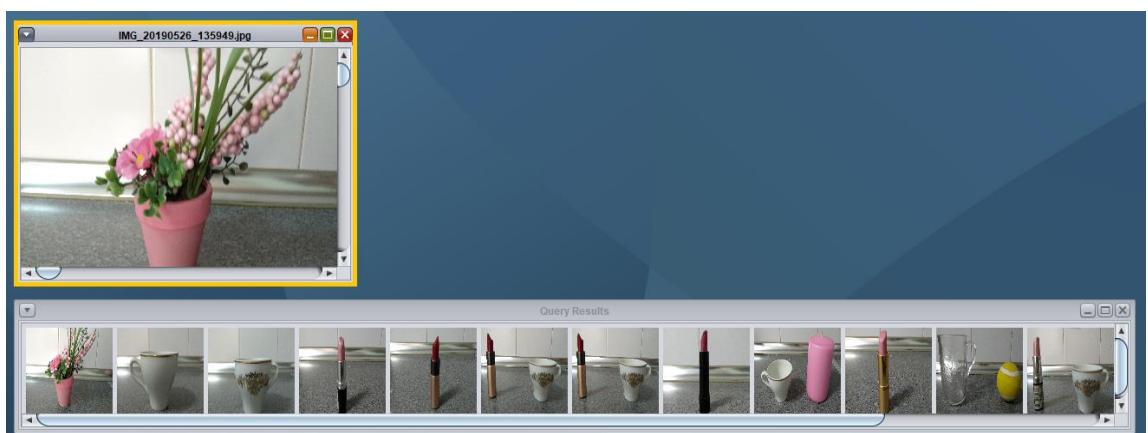


Figura 14: Segundo experimento con un objeto maceta extrayendo la estructura del color de forma global.

De nuevo, tal y como podemos comprobar, la lista de imágenes resultante presenta alguna de las tonalidades que contiene la imagen consulta. Es por ello por lo que en las imágenes recuperadas aparecen elementos rosas, como la vela o algunos pintalabios, así como otras en las que aparece una taza blanca a causa de que el fondo de la imagen consulta está compuesta por una pared del mismo color.

Una segunda peculiaridad, también muy común, en la mayoría de los estudios realizados a imágenes es que las características se extraen de forma generalizada, es decir, se analiza la imagen al completo. Esto puede suponer un problema en tanto en cuanto aparezcan diversos objetos en su escena. Un ejemplo que represente esta idea puede ser aquel en el que una imagen contenga dos elementos de propiedades gráficas muy distintas de manera que la información descriptiva extraída, de forma global, no sea suficientemente precisa como para generar buenos resultados. Para demostrarlo adjunto la siguiente figura que representa el experimento realizado con el software de este proyecto, en el cual la imagen consulta está compuesta por una taza roja y una vela rosa.

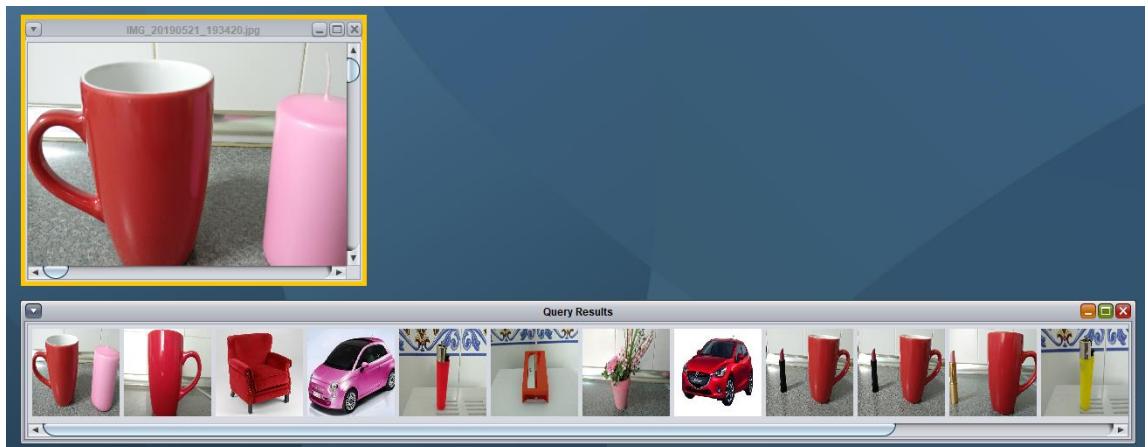


Figura 15: Primer experimento con dos objetos y extrayendo el color medio de forma global.

Tal y como podemos comprobar en la captura, las imágenes resultantes contienen, en su mayoría, alguno de los dos colores que aparecen en la imagen consulta, es decir, gran parte de ellas muestran objetos rojos o rosas, independientemente de la categoría a la que estos pertenezcan. Sin embargo podemos contemplar que ciertas imágenes muestran tonalidades muy distintas de ambos colores, llegando a aparecer otras totalmente diferentes como la relacionada con el encendedor amarillo. Con el fin de solucionar este primer inconveniente en mi proyecto se propone una novedosa idea consistente en aplicar una rejilla a la imagen. De esta forma se extraerá la información descriptiva de cada región en lugar de tomar la imagen completa. Para demostrar esta teoría procedo a realizar una consulta tomando como imagen consulta la misma que en el experimento anterior.



Figura 16: Segundo experimento con dos objetos extrayendo el color medio localmente.

Si comparamos los resultados de este último experimento con los proporcionados en el anterior podremos comprobar que en este ejemplo actual, en todas las fotos resultantes, se muestra uno de los colores principales que alberga la imagen consulta. Esto es debido a que, en este caso, se han extraído cuatro colores medios correspondientes a las cuatro regiones en las han sido divididas todas las imágenes que se pueden ver.

Otro ejemplo que demuestra este avance es el que se presenta a continuación. En esta primera captura repetimos el proceso de extracción, en este caso, de la estructura del color de manera global a la imagen. Para ello usaremos como imagen consulta aquella en la que aparece un sacapuntas azul en la parte derecha.



Figura 17: Primer experimento con un objeto sacapuntas y extrayendo la estructura del color de forma global.

De nuevo, tal y como podemos comprobar, existen un gran número de imágenes que presentan alguna de las dos tonalidades principales de la imagen consulta, en este caso azul o gris. No obstante hay dos imágenes, en las que aparece una copa de cristal respectivamente, que se encuentran en la lista de las fotos más parecidas. Si bien este hecho no es cierto, a consecuencia de extraer la estructura del color de la imagen consulta de forma global, el azul detectado en las tres imágenes en cuestión es suficiente motivo para colocar las dos fotografías de las copas entre las imágenes más parecidas.

No ocurre lo mismo en este siguiente experimento donde se obtiene la misma propiedad que en el ejemplo anterior, la estructura del color, pero aplicando una rejilla que divide verticalmente a la imagen en dos mitades. Y es que tal y como podemos comprobar en la siguiente captura, todas las imágenes resultantes presentan alguno de los dos principales colores de la imagen consulta, azul o gris.

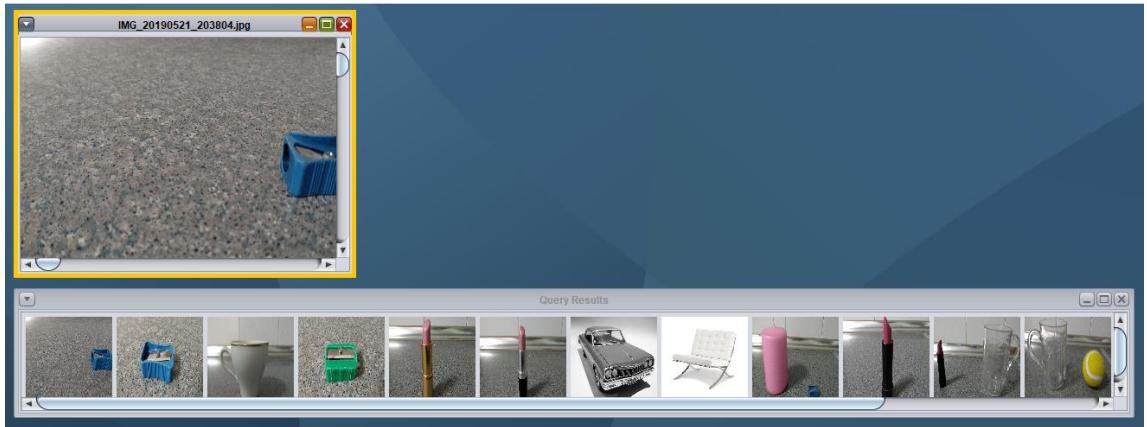


Figura 18: Segundo experimento con un objeto sacapuntas y extrayendo la estructura del color localmente.

Si bien la introducción de este procedimiento, consistente en extraer las propiedades gráficas de forma local en todas las regiones en las que se divide una imagen, ha supuesto una notable mejoría en los resultados mostrados en los anteriores experimentos, seguimos arrastrando el mismo problema comentado en capítulos anteriores. Todos los resultados son acorde a una determinada característica visual y no al tipo de elemento que aparece en la imagen consulta. Es por ello por lo que mi proyecto añade una nueva funcionalidad para posibilitar el hecho de realizar consultas en función de la escena de una imagen y sin importar sus propiedades visuales. Para ello utilizaremos la ya explicada red neuronal convolucional o CNN denominada *ResNet* con el fin de que nos ayude a identificar cada objeto y asignarle su etiqueta correspondiente.

Comenzaremos a mostrar la gran ventaja que supone esta nueva metodología con el siguiente ejemplo ilustrativo. En él llevaremos a cabo una recuperación de todas aquellas imágenes que contengan el mismo elemento que alberga la imagen consulta. En este caso, tal y como se puede comprobar, aparece una copa de cristal.

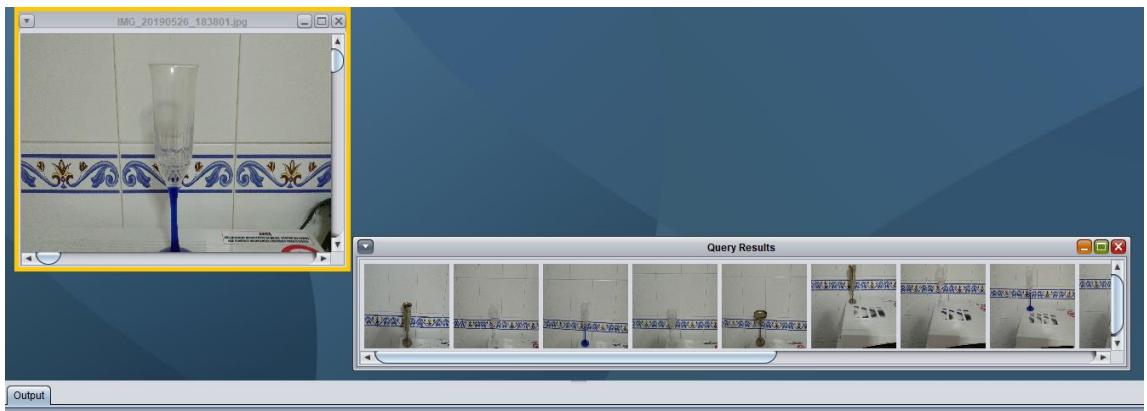


Figura 19: Primer experimento con un objeto copa y usando la identificación mediante etiquetas globalmente.

En este experimento cabe destacar que la red neuronal ha sido capaz de identificar el objeto de la imagen consulta, etiquetándolo como *goblet* lo cual se traduce como “copa” en español. Este hecho se puede comprobar abajo a la derecha en el texto subrayado de azul. Gracias a la identificación y asignación de un término lingüístico hemos conseguido un resultado en el que, tal y como podemos observar, solo aparecen imágenes de copas de distintas características físicas y en diferentes posiciones.

Si bien es cierto que comienza a quedar patente el potencial de esta herramienta en el ejemplo anterior, este procedimiento, asociado a la identificación de un elemento mediante una etiqueta lingüística, demuestra una gran ventaja cuando en la imagen consulta aparecen varios objetos. Este hecho nos permite realizar consultas en las que el CBIR tendrá en cuenta los términos lingüísticos asociados a los objetos que la CNN sea capaz de reconocer. La versatilidad de esta metodología se verá, además, extendida en los siguientes capítulos gracias a la implementación en mi proyecto de diversas métricas. Con ellas el usuario podrá definir el grado de precisión de la consulta, el cual está relacionado con el número de objetos que deberán aparecer en las imágenes que se recuperen. También será posible restringir la posición de los susodichos a la localización en la que se encuentren los diversos objetos de la imagen consulta. Todas las distintas medidas implementadas serán explicadas detalladamente y con ejemplos representativos en el siguiente capítulo.

Capítulo 4

Métricas

Una métrica o comparador es un procedimiento que calcula la similitud entre dos imágenes, independientemente de la información descriptiva que haya sido extraída previamente. La razón por la que he implementado diversas métricas se fundamenta en proporcionar al usuario distintas formas de realizar una consulta. Así, tal y como explicaremos en este capítulo, se establecerán varios parámetros que controlarán ciertos aspectos tales como encontrar uno o varios objetos de los que aparezcan en la imagen consulta, o restringir la posición en la que estos se localizan.

Fundamentalmente los comparadores implementados se pueden clasificar en tres categorías en base a sus peculiaridades: aquellos que tienen en consideración la posición de los elementos, los que buscan los objetos en cualquier posición y, por último, un tercer tipo especial de comparadores cuyo nivel de restricción es llevado hasta el máximo exponente.

4.1. Clasificación de los comparadores.

A continuación se detallarán las explicaciones correspondientes a cada comparador describiendo su funcionamiento e ilustrándolo con diversos ejemplos representativos para facilitar su comprensión.

4.1.1. Distinta posición

En esta categoría se encuentran aquellas métricas en las que no importa la posición en la que se encuentren los objetos de la imagen consulta. Es por ello por lo que son ideales para que el usuario busque elementos en particular que se pueden encontrar en cualquier localización. Además cuentan con otra ventaja adicional, y es que sus descriptores no se ven obligados a contar con el mismo número de datos descriptivos, por lo que dos imágenes con rejillas de distintas dimensiones pueden ser comparadas.

No obstante el hecho de que dos imágenes puedan ser comparadas pese a que sus descriptores tengan distintos tamaños supone, a su vez, un inconveniente. Como este proyecto está diseñado especialmente para utilizar los términos lingüísticos proporcionados por la CNN, puede darse el caso en el que alguna de las etiquetas de la imagen consulta se encuentre en la otra imagen, con la que se está comparando, pero no viceversa. Esto puede ocurrir en tanto en

cuanto la CNN proporcione diversas etiquetas para una región de una imagen. Si solo realizamos la comparación en un único sentido (imagen consulta → otra imagen) y aparece esta circunstancia, las dos imágenes serán consideradas como parecidas.

Esta operación se denomina ***inclusión simple***, la cual consiste en comprobar si las etiquetas de una región de la imagen consulta aparecen en la región de la imagen con la que se está comparando. Dicho cálculo pertenece al ámbito de los conjuntos algebraicos, y por ello se puede formular matemáticamente de la siguiente manera:

$$A = B \leftrightarrow (A \subset B)$$

El otro método que sí que realiza la comprobación anterior en los dos sentidos para comprobar que existen todas las etiquetas en ambas regiones se denomina ***doble inclusión***. Su nombre explica la doble comparación que realiza dadas dos regiones de dos imágenes distintas. Tal y como ocurría en el caso anterior esta operación también se encuentra dentro del campo asociado a los conjuntos matemáticos y por tanto su fórmula es la siguiente:

$$A = B \leftrightarrow (A \subset B) \wedge (B \subset A)$$

A continuación se detallarán las tres operaciones comunes a los comparadores desarrollados pero adaptadas a las características de cada una de las familias.

4.1.1.1. *Al menos un elemento*

En esta clase de métrica dos imágenes A y B se las considerará como similares si al menos uno de los elementos de A se encuentra en cualquier región de la imagen B . La fórmula matemática asociada a este procedimiento se puede observar a continuación:

$$d = \min (\min(|A_i - B_j|, \forall i, \forall j))$$

El **primer ejemplo** ilustrativo de esta métrica se presenta a continuación. En él se toma como imagen A o imagen consulta aquella que contiene una taza rosa y una pelota de tenis verde. Si aplicamos este comparador las imágenes resultantes que se esperan son todas aquellas en las que aparezca al menos uno de los dos objetos en cualquier localización.

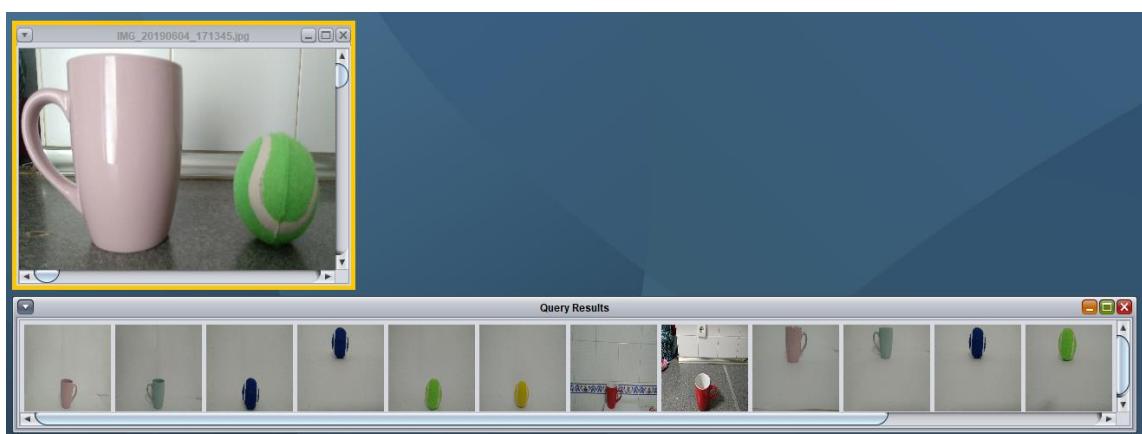


Figura 20: Primer experimento de la primera familia de comparadores – Al menos un objeto en cualquier posición.

Tal y como se puede observar, en la lista de imágenes resultante aparecen fotografías tanto de

tazas como de pelotas de tenis de diversos colores y en distintas posiciones.

Un **segundo ejemplo** aplicando esta misma métrica es el que se expone a continuación. En este caso la imagen A está compuesta por tres objetos, dos mecheros y una goma de borrar. De nuevo los resultados que se esperan son aquellas fotografías en las que al menos aparezca uno de los objetos en cualquier posición.

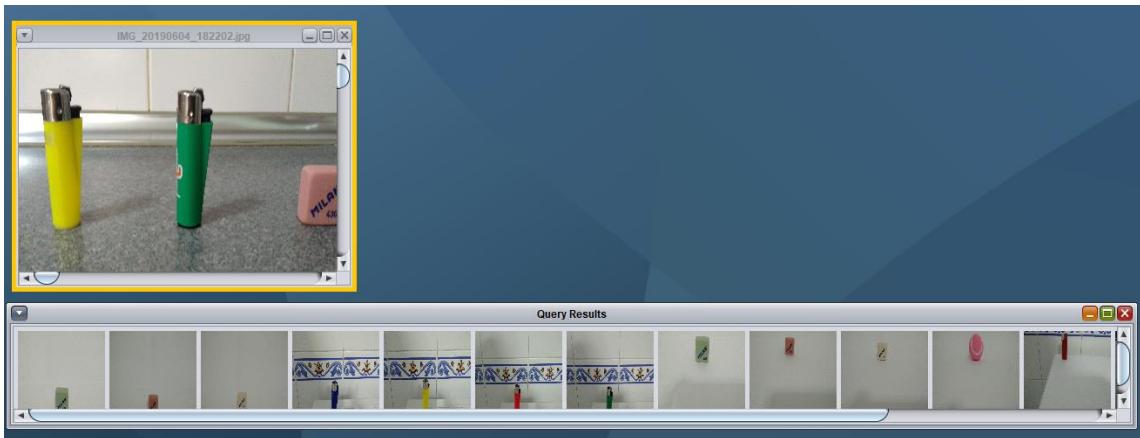


Figura 21: Segundo experimento de la primera familia de comparadores – Al menos un objeto en cualquier posición.

Tal y como se muestra en la captura la mayoría de las imágenes resultantes contienen alguno de los dos tipos de objetos, sin embargo si nos fijamos un poco más en la lista de fotografías, podremos comprobar que en la imagen número 11 hay una pelota rosa de tenis. Este es un claro ejemplo del inconveniente, explicado anteriormente, que arrastra la **inclusión simple**. En este caso la foto de la pelota contiene el siguiente conjunto de etiquetas: `{tennis_ball,rubber_eraser}`, las cuales en español se traducen como pelota de tenis y goma de borrar. Debido a que la comparación se está haciendo en un único sentido, desde la imagen consulta a la otra, y teniendo en cuenta que en ambas imágenes está la etiqueta referente a la goma de borrar, la foto de la pelota de tenis aparece en los resultados de la consulta aunque no tenga ningún parecido.

Para obtener unos resultados más precisos forzamos a que la comparación se realice en ambos sentidos, aplicando para ello, la **doble inclusión**. Esto se demuestra en el **tercer experimento** en el que se replica la consulta del ejemplo anterior aunque añadiendo esta doble comparación. Las imágenes resultantes se pueden observar en la siguiente captura.

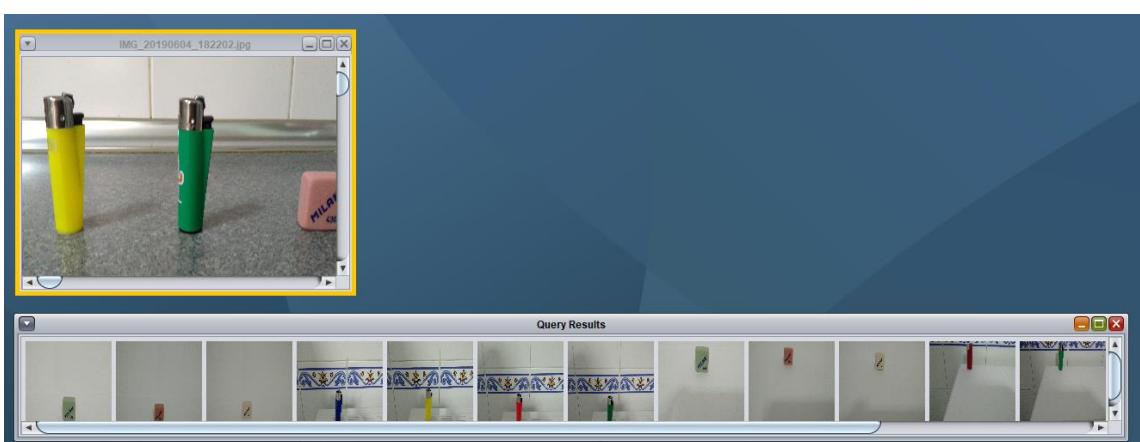


Figura 22: Tercer experimento de la primera familia de comparadores – Al menos un

objeto en cualquier posición utilizando la doble inclusión.

Tal y como podemos comprobar, reemplazando la inclusión simple por la doble inclusión, obtenemos unos resultados en los que todas las imágenes mostradas contienen, al menos, uno de los objetos que aparece en la imagen consulta.

4.1.1.2. *La mayoría de los elementos*

En esta segunda métrica dos imágenes A y B son parecidas si las diferencias entre todas sus regiones son mínimas. Esta descripción se traduce en la práctica al caso en el que dadas dos imágenes, si alguna de ellas tiene una región muy distinta de la otra, este comparador las clasificará como diferentes. Como podremos comprobar es una medida muy restrictiva en la que se consiguen imágenes que contienen la gran mayoría de los objetos que aparecen en la imagen consulta, en este caso, en distintas posiciones. Su fórmula matemática se puede ver a continuación:

$$d = \max (\min(|A_i - B_j|, \forall i, \forall j))$$

El **primer ejemplo** que demuestra el comportamiento descrito anteriormente es el siguiente, en el que se observa una imagen consulta con dos objetos, un labial y un mechero amarillo. Las fotografías que esperamos que muestre son todas aquellas en las que los dos elementos aparezcan, aunque en distintas posiciones.

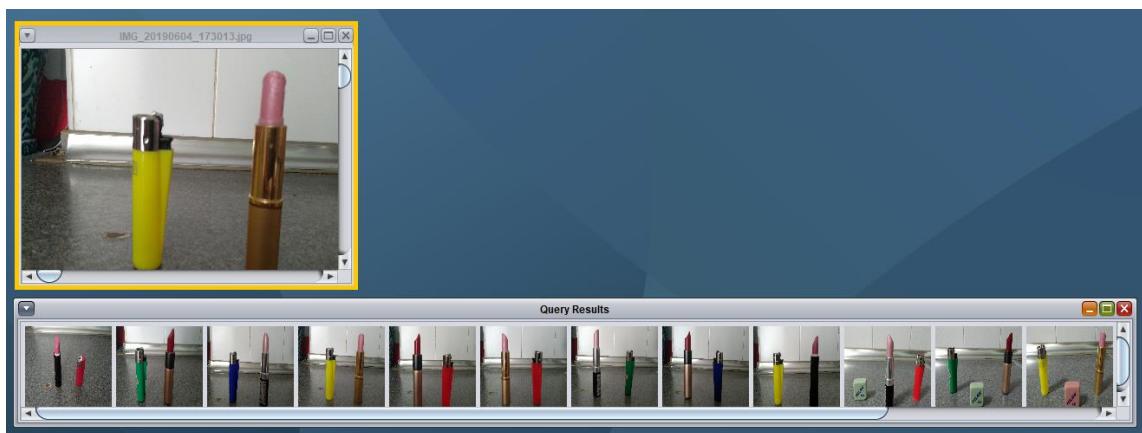


Figura 23: Primer experimento de la primera familia de comparadores – La mayoría de los elementos.

En los resultados mostrados podemos comprobar que en todas las imágenes aparecen los dos objetos, en distintas posiciones y con diferentes características físicas. No obstante, si nos fijamos en las tres últimas imágenes de la lista podemos comprobar que, además de los dos elementos de la imagen consulta, también aparece una goma de borrar. Aunque han sido clasificadas como similares esto no es del todo cierto puesto que la imagen consulta no cuenta con este último elemento. De nuevo vuelve a quedar patente la falta de precisión cuando usamos la **inclusión simple** en algunos casos.

Si bien es cierto que los resultados del **siguiente experimento**, tomando la misma imagen consulta pero aplicando la **doble inclusión**, no muestran cambios significativos en las primeras imágenes, sí podemos notar que las últimas fotografías destacadas anteriormente ya no se encuentran en la lista resultante.

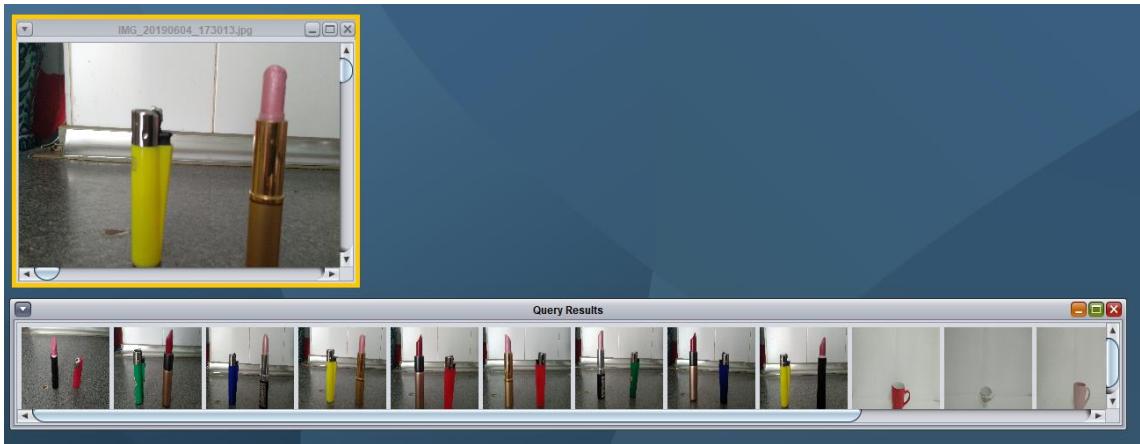


Figura 24: Segundo experimento de la primera familia de comparadores – La mayoría de los elementos utilizando la doble inclusión.

Como resultado de esta doble comparación podemos observar que, tras la última imagen que cumple el requisito de que únicamente aparezcan los dos objetos mostrados en la imagen consulta, la lista resultante muestra el resto de fotografías en el orden en el que se ha realizado la comparación, puesto que ninguna de ellas contiene los dos elementos fotografiados.

Un **ejemplo característico** es el que se explica a continuación. En este experimento la imagen consulta contiene tres elementos del mismo tipo, es decir, tres sacapuntas de diversos colores. Si aplicamos esta métrica los resultados esperados consistirían en mostrar imágenes en las que aparecieran todas las etiquetas lingüísticas que contiene la imagen consulta. Los resultados se pueden comprobar en la siguiente captura.

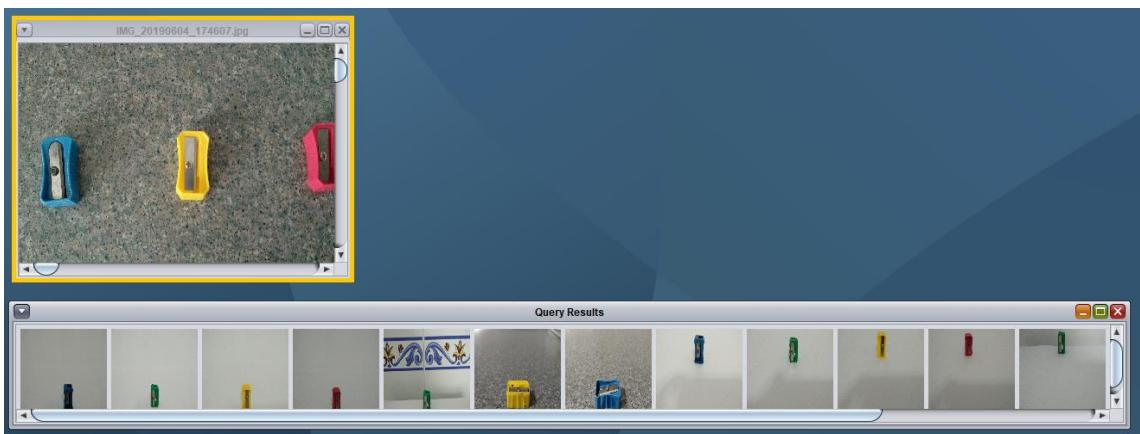


Figura 25: Tercer experimento de la primera familia de comparadores – La mayoría de los elementos.

Tal y como podemos comprobar en la lista resultante, efectivamente, aparecen imágenes que contienen el término lingüístico asociado a un sacapuntas. No obstante, el hecho de haber utilizado la **inclusión simple**, ha provocado que se muestren fotografías que no son estrictamente parecidas a la imagen consulta. Y es que, si bien es cierto que aparece el mismo tipo de elemento, en la imagen consulta hay tres sacapuntas. De nuevo queda patente que, para ciertos ejemplos, no basta con realizar la comparación en un único sentido, sino que es necesario comprobar que aparece cada una de las etiquetas en los descriptores de ambas imágenes.

Si realizamos la misma consulta pero aplicando la **doble inclusión** podremos comprobar como

la lista resultante varía considerablemente.

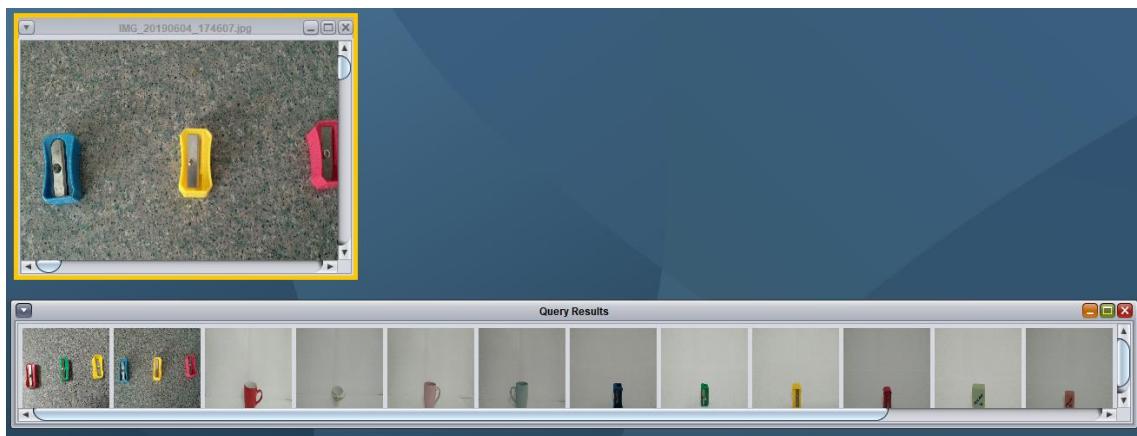


Figura 26: Cuarto experimento de la primera familia de comparadores – La mayoría de los elementos utilizando la doble inclusión.

En este caso podemos comprobar que las únicas fotografías que son realmente parecidas a la imagen consulta son las dos primeras que aparecen en la lista, en las cuales aparecen tres sacapuntas de distintos colores. A partir de ellas las imágenes restantes se ordenan en función del orden de consulta que se haya realizado.

4.1.1.3. *En promedio*

En esta última métrica asociada a esta primera familia se considerará que dos imágenes son afines si en promedio sus regiones son similares, es decir, si en ambas aparecen los mismos elementos pero en distinta posición. Si bien es una métrica bastante estricta lo es en menor medida comparada con la anterior. La fórmula matemática que representa el procedimiento que sigue se expone a continuación:

$$d = \text{mean} (\min(|A_i - B_j|, \forall i, \forall j))$$

El **primer ejemplo** que ilustra el funcionamiento de este comparador es el representado en la siguiente figura. Como imagen consulta aparece una fotografía cuyos dos elementos son un sacapuntas y una goma de borrar. Las imágenes que se espera que esta métrica recupere son aquellas en las que, en promedio, aparezcan ambos objetos.

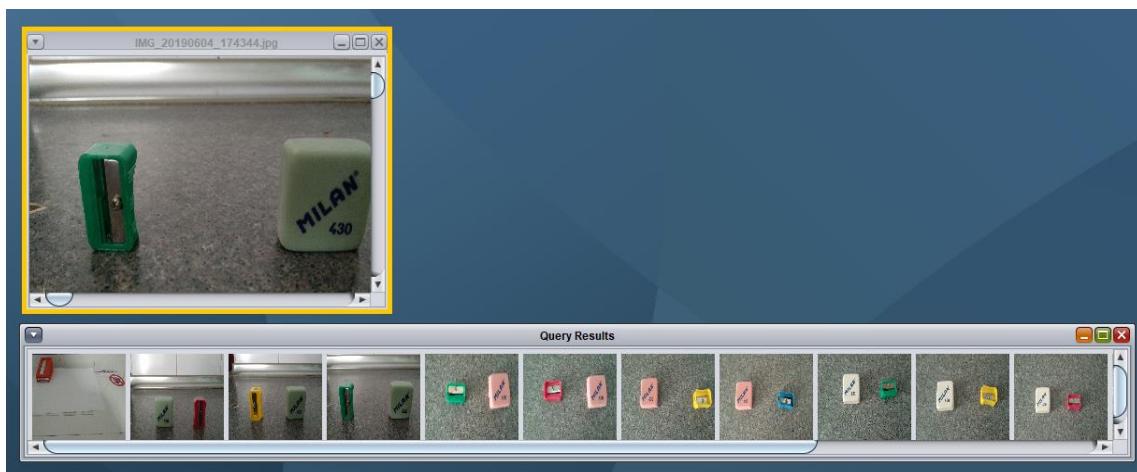


Figura 27: Primer experimento de la primera familia de comparadores – En promedio.

Tal y como podemos apreciar en la gran mayoría de imágenes aparecen ambos objetos, aunque en distintas posiciones y de distintas tonalidades. No obstante, si nos fijamos en la primera imagen podemos comprobar que se clasifica como la fotografía más parecida a la imagen consulta. Podemos ver que esto no es cierto, y por ello una vez más queda constancia de que aplicar la **inclusión simple** a ciertos ejemplos no es, quizás, la mejor forma de obtener unos resultados precisos. Pero este es un caso particular en el que la imagen destacada en primer lugar cuenta con las dos etiquetas pertenecientes a los dos elementos que aparecen en la imagen consulta, por lo que aplicar la **doble inclusión** sería inútil. Nos proporcionaría los mismos resultados puesto que los términos lingüísticos se encuentran en los descriptores de ambas imágenes.

Un **segundo experimento** aplicando esta métrica es el que se muestra a continuación. En él se encuentra una imagen consulta en la que aparecen tres objetos de diversa naturaleza: una goma de borrar, un mechero y un labial. Las fotografías esperadas son aquellas que contengan, en su mayoría, a todos los elementos de la imagen consulta.

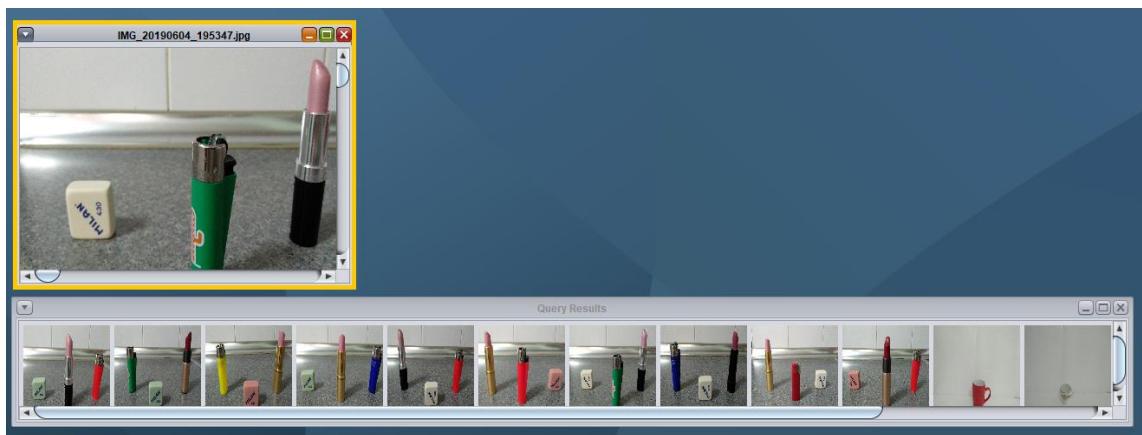


Figura 28: Segundo experimento de la primera familia de comparadores – En promedio.

Apreciando la lista de imágenes resultante podemos comprobar que aquellas fotos que contienen a los tres elementos se colocan en las primeras posiciones, mientras que las otras que no cumplen el patrón de etiquetas establecido se sitúan al final de la lista.

4.1.2. Sin repetidos

Esta segunda familia de métricas comparte dos aspectos con la anterior. El primero de ellos consiste en seguir buscando los objetos de la imagen consulta en distintas regiones de las otras imágenes. Y el segundo implica a todos los comparadores puesto que disponen de las tres casuísticas explicadas anteriormente: que al menos aparezca uno de los elementos, que se encuentren la mayoría de ellos o que, en promedio, se puedan observar todos los objetos. No obstante, dependiendo del funcionamiento del comparador, estas tres operaciones ofrecerán resultados muy distintos y sufrirán notables variaciones.

Si bien es cierto que la métrica que nos ocupa en cuestión comparte ciertos rasgos similares a la anterior, este comparador ofrece unos resultados más precisos. La razón de ello se puede explicar con el siguiente ejemplo. Supongamos que dadas dos imágenes, una de ellas se

corresponde con un único elemento que ocupa toda la fotografía, como puede ser el caso de la primera imagen que se muestra a continuación. Si realizamos la consulta con el anterior comparador, a consecuencia de que ambas imágenes tienen un elemento común, el césped, serían clasificadas como parecidas. No obstante esto no es cierto puesto que la primera es una imagen de un trozo de césped y en la segunda se muestra un conjunto de personas practicando un deporte en un parque con césped.



Figura 29: Imagen de un trozo de césped.

Figura 30: Imagen que representa un grupo de personas practicando deporte sobre un parque con césped.

Con el objetivo de evitar que todas las regiones de la imagen consulta se asemejen a una misma región de otra imagen, esta métrica incluye una novedosa restricción. Consiste en forzar que solo una región de una imagen A comparta un grado de similitud con una única región de la imagen B , si y solo si ninguna de las dos regiones tenga ya asignada otra celda. De esta manera evitaremos que dos imágenes, que tienen un elemento en común pero que no representan lo mismo, se califiquen como parecidas.

4.1.2.1. *Al menos un elemento*

Si bien es cierto que el principio fundamental de esta operación se mantiene constante como en la anterior familia, en este caso debemos aplicarle la restricción comentada anteriormente. Por ello dadas dos imágenes A y B se las considerará como similares si al menos uno de los elementos de A se encuentran en cualquier región de B , siempre y cuando ninguna de las dos regiones haya sido ya emparejada con una celda de una tercera imagen. Su fórmula matemática aproximada sería la siguiente:

$$d = \min \left(\min \left(|A_i - B_j|, \forall i, \forall j \leftrightarrow i \notin (i, k) \wedge j \notin (l, j) \right) \right)$$

En el caso en el que una de las dos regiones ya tenga asignada una pareja se compararía la diferencia entre las regiones que la forman actualmente y la otra celda que pretende asociarse con un miembro de la pareja. Si el par de celdas actuales proporcionan una diferencia mayor entonces se actualizan sus miembros y se busca una nueva región compañera a la celda de la imagen A que haya sido dividida.

El siguiente **ejemplo** ilustrativo de esta métrica aplicándole la restricción de que al menos uno de los objetos esté en la imagen B se puede observar a continuación. Si bien es cierto que la peculiar restricción que distingue a esta métrica se puede aplicar a las tres casuísticas, en

esta en concreto, sus efectos no son notorios. Para demostrarlo el siguiente experimento parte de una imagen consulta cuyos principales elementos son un labial, una goma y un sacapuntas. Con el objetivo de comparar los resultados de esta métrica con los que proporciona la anterior, se llevarán a cabo dos consultas para analizar las diferencias entre los dos resultados.

En este primer experimento que nos ocupa los resultados esperados en ambos casos consisten en mostrar imágenes en las que, al menos, aparezca uno de los objetos de la imagen consulta.

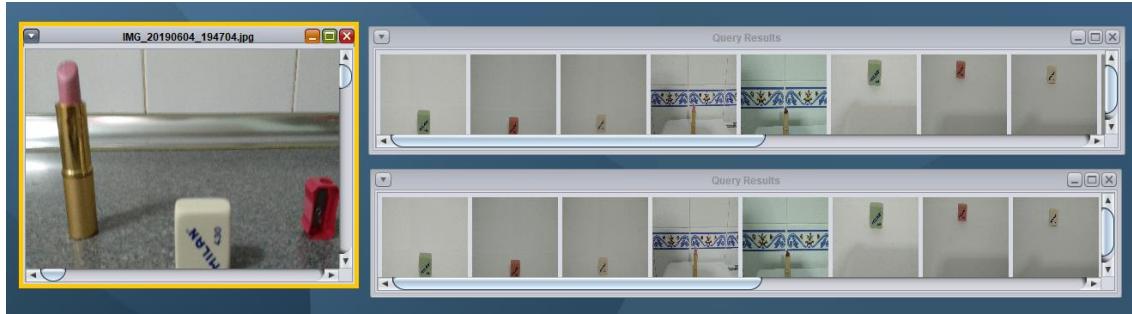


Figura 31: Primer experimento de la segunda familia de comparadores – Al menos un objeto.

Tal y como podemos observar en ambas listas resultantes se encuentran fotografías con al menos uno de los elementos de la imagen consulta. No obstante, pese a que las consultas han sido realizadas con una métrica distinta, los resultados son los mismos. Y es que, aunque queramos buscar imágenes en la que se encuentren las tres etiquetas referentes a los tres elementos, si al menos existe una de ellas es motivo suficiente como para clasificarlas como parecidas. Esta es una de las desventajas que tiene esta casuística, que si bien te permite buscar cualquier objeto de los que aparecen en la imagen consulta, no te garantiza que se encuentren todos.

4.1.2.2. *La mayoría de los elementos*

A diferencia de la casuística anterior en esta sí que se aprecian diferencias significativas con respecto a su homólogo que sí permite enlazar una celda con varias regiones de una misma imagen. En este caso podemos definir la métrica como aquella que, dadas dos imágenes A y B se considerarán como similares si aparecen la mayoría de los objetos de A en B , emparejando dichas celdas si ninguna de ellas ya ha sido asignada a una tercera región. La formulación matemática correspondiente a este comparador se puede observar a continuación:

$$d = \max \left(\min \left(|A_i - B_j|, \forall i, \forall j \leftrightarrow i \notin (i, k) \wedge j \notin (l, j) \right) \right)$$

Como en el caso anterior, si dos regiones que se quieren asociar contienen uno de sus miembros en otra pareja, se actualizará la susodicha siempre y cuando la diferencia que proporcione sea mayor que la que resulta del nuevo par de celdas. Posteriormente, si se da el caso, se realizará una nueva búsqueda para emparejar a la celda de la imagen consulta que se haya quedado sola.

El **primer ejemplo** ilustrativo del potencial que puede llegar a desarrollar esta casuística aplicada a la familia actual es el siguiente que se puede apreciar. En este experimento partimos de una imagen consulta formada por dos labiales, por lo que aunque la etiqueta para ambos elementos sea la misma, los resultados proporcionados por esta métrica y su homóloga con

repetidos no serán parecidos.

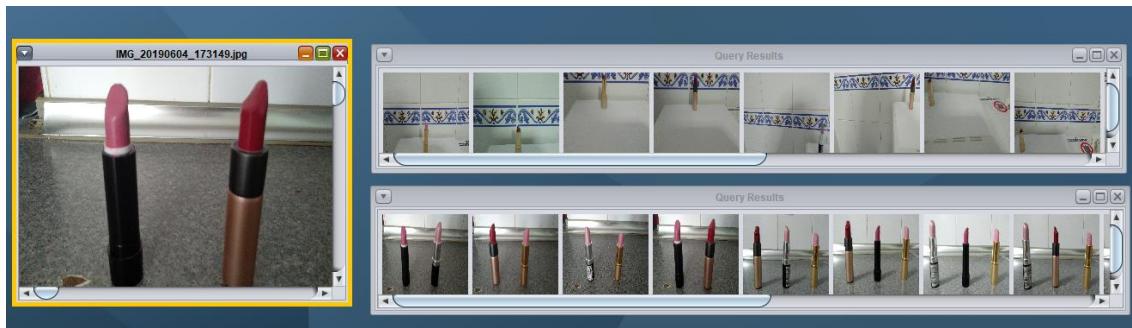


Figura 32: Primer experimento de la segunda familia de comparadores – La mayoría de objetos.

La lista de imágenes superior se corresponde con la métrica que sí permite emparejar las etiquetas de la imagen consulta con una misma celda de la otra fotografía con la que se compara. Por ello el resultado mostrado es el de un conjunto de imágenes en las que en todas aparece un pintalabios. No obstante, siendo medianamente estrictos, estas fotografías no son del todo similares a la imagen consulta.

Por el contrario si observamos la lista inferior producida por la consulta realizada con el comparador que no permite emparejar una misma celda de la imagen consulta con varias celdas de la imagen con la que se compara, podemos comprobar como las fotos que aparecen tienen un gran parecido con la imagen consulta. No solo contienen la etiqueta correspondiente al labial sino que además en los primeros puestos aparecen aquellas imágenes que contienen exactamente dos labiales, independientemente de sus características físicas. La razón de este comportamiento consiste en que, al aplicar la operación en la que la mayoría de las etiquetas de una imagen *A* deben aparecer en una imagen *B* de modo que no haya una celda de *A* emparejada con varias de *B*, provoca que una vez se ha asignado la primera etiqueta del labial si la imagen *B* no dispone de otra etiqueta igual, esta comparación producirá un valor muy grande que resultará en que dichas imágenes no se parecen. Este es el caso que se origina con las fotografías en las que solo hay un labial y es por ello por lo que estas no aparecen en la lista resultante inferior.

Un **segundo ejemplo** que represente el caso con el que hemos comenzado este capítulo, en el que se ha podido apreciar una fotografía en la que solo aparece un único elemento pero de manera generalizada, como el caso de la imagen del césped, y otra foto que contiene este objeto en ciertas regiones pero que no representa la misma escena que la primera imagen. Para demostrar esta teoría vamos a extrapolar ambos tipos de fotografías a las siguientes cuatro imágenes. En las dos primeras se muestran un conjunto de fresas que ocupan la mayor parte de la imagen. Y en las dos últimas podemos apreciar que, si bien también aparecen algunas fresas, lo que en realidad representan son dos postres dulces decorados con esta fruta.



Figura 33: Imagen repleta de fresas.



Figura 34: Cuenco de fresas.



Figura 36: Tarta de queso con dos fresas decorativas.



Figura 35: Tarta de fresa con dos fresas decorativas.

Para realizar este experimento vamos a establecer como imagen consulta la primera fotografía que se ha mostrado cuya escena está repleta de fresas. Tal y como ya es tendencia se realizarán dos consultas en base al comparador que sí permite parejas con miembros pertenecientes a otras parejas y la métrica que no lo permite, respectivamente. Los resultados esperados en el primer caso se corresponden con todas aquellas imágenes en las que aparezca esta fruta, sin importar su localización o su número. Y las fotografías que se espera que sean recuperadas por el otro comparador son aquellas en las que, además de aparecer la etiqueta asociada a la fresa, este elemento sea tan numeroso como en la imagen consulta.



Figura 37: Segundo experimento de la segunda familia de comparadores – La mayoría de los objetos.

Si nos fijamos en la primera lista sus imágenes muestran fotografías de fresas pero sin tener en cuenta si hay un mayor número de ellas o si son las principales protagonistas o no de la escena de las imágenes. No ocurre lo mismo en la segunda lista en la que, gracias a las restricciones impuestas por el comparador que actualmente estamos explicando, las dos únicas fotos en las que la fruta aparece de forma generalizada en la imagen son las que ha clasificado como las más similares. El resto al no cumplir los patrones concretados aparecen en el orden en el que se consultaron.

4.1.2.3. En promedio

De forma similar al caso anterior en esta ocasión también se pueden apreciar las discrepancias entre los resultados del comparador que permite emparejamientos con elementos que ya forman parte de otras parejas y los resultados de esta métrica en cuestión. En ella se aplica la misma restricción que en las dos anteriores solo que clasificará dos imágenes A y B como similares si existe un parecido general entre ellas. Esto se traduce a que la imagen B contenga, de forma general, los objetos que aparecen en la imagen A . Su fórmula matemática aproximada se puede observar a continuación:

$$d = \text{mean} \left(\min \left(|A_i - B_j|, \forall i, \forall j \leftrightarrow i \notin (i, k) \wedge j \notin (l, j) \right) \right)$$

En este comparador se aplican las mismas técnicas descritas en los anteriores.

Un **ejemplo** ilustrativo de este caso en particular se corresponde con el siguiente experimento, en el cual existe una imagen consulta cuyos tres objetos están representados por tres sacapuntas. Al igual que en el experimento anterior realizaremos dos consultas, la primera con el comparador homólogo al actual que sí permite parejas cuyos miembros estén en otras y la segunda en la que se aplique el comparador que estamos describiendo actualmente. Los resultados esperados por la primera consulta mostrarán aquellas imágenes en las que, en general, se encuentren los elementos de la imagen consulta. Sin embargo la lista de imágenes resultantes producida por la métrica que actualmente estamos explicando mostrará un conjunto de fotografías más acordes a la escena de la imagen consulta.

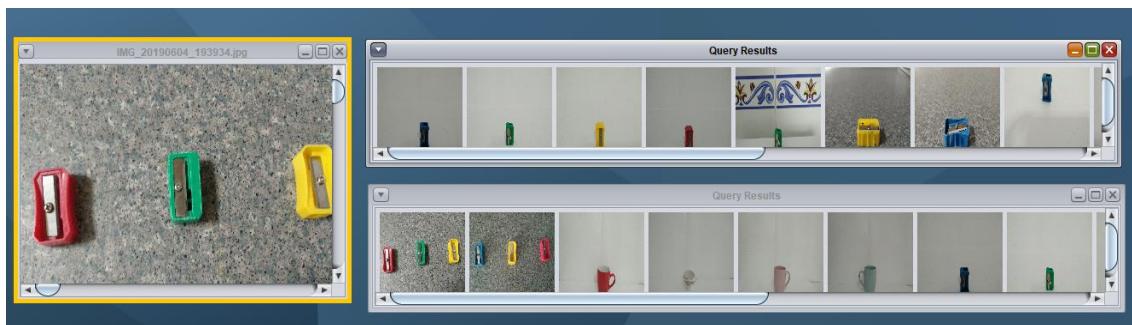


Figura 38: Primer experimento de la segunda familia de comparadores – En promedio.

Tal y como podemos observar, el comparador de la familia anterior solo es capaz de mostrar fotografías en las que aparece un sacapuntas. Si bien este hecho es correcto de acuerdo a su objetivo de recuperar aquellas imágenes que tengan un parecido general con la imagen consulta, estas fotografías no reflejan una escena similar a la de la imagen consulta.

No ocurre lo mismo con la segunda lista de imágenes, la cual ha sido producida por el comparador que explicamos actualmente. Este, además de buscar las etiquetas de la imagen consulta también toma en cuenta el número de ellas. Así las dos imágenes clasificadas como las más parecidas a la imagen consulta contienen tres sacapuntas. El resto de las fotografías se muestran en el orden en el que se consultaron, debido a que no existen más imágenes que cumplan con los criterios especificados.

4.1.3. Misma posición

Esta tercera familia de comparadores no comparte la filosofía de las dos anteriores con

respecto a la posición de los objetos. De hecho este tipo de métricas destaca por realizar la consulta entre dos imágenes comparando las regiones, de una y otra imagen, que se encuentren en la misma posición. Este nuevo procedimiento de consulta origina una consecuencia importante y es que, para aplicar alguno de los comparadores de esta familia, ambas imágenes deben estar divididas en el mismo número de regiones, es decir, sus descriptores deberán tener el mismo tamaño.

Este grupo de comparadores resulta muy útil en aquellos casos en los que queremos recuperar imágenes en las que los objetos que aparezcan estén situados en la misma posición en la que se encuentran los elementos de la imagen consulta. Además, tal y como sucede con las otras dos familias, en este tipo de comparadores se pueden aplicar las mismas tres operaciones. A continuación explicaremos su funcionamiento adaptado a este tipo de comparadores y los ilustraremos con ejemplos de distintas consultas.

4.1.3.1. *Al menos un elemento*

Aplicando este comparador en particular obtendremos que, dadas dos imágenes A y B se considerarán como similares si al menos aparece un elemento de A en B ocupando la misma posición. Así se recuperarán todas aquellas imágenes que contengan uno de los objetos que aparecen en la escena de la imagen consulta pero teniendo en cuenta su posición. La fórmula matemática asociada a esta métrica es la siguiente:

$$d = \min (|A_i - B_i| \forall i)$$

El **primer experimento** asociado a este comparador es el que se puede observar a continuación. En él realizamos una consulta cuya imagen consulta está compuesta por una taza situada en la esquina superior derecha de la imagen. Los resultados que esperamos tras aplicar esta métrica consisten en un conjunto de imágenes en las que aparezca una taza, de distintas características y fotografiada en distintos lugares, en la misma posición anterior.

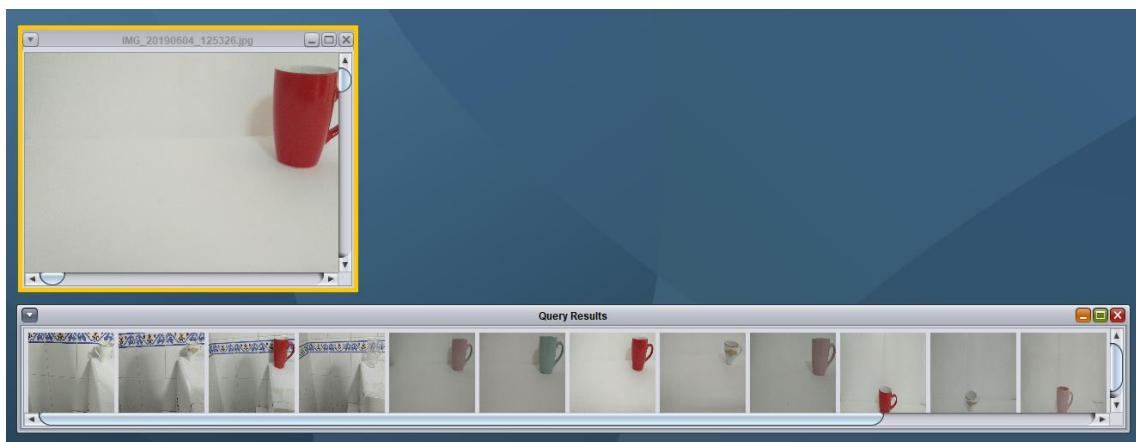


Figura 39: Primer experimento de la tercera familia – Al menos uno de los elementos.

Tal y como podemos comprobar en la figura anterior los resultados obtenidos son acordes a los que esperábamos. En las primeras posiciones de la lista se encuentran aquellas fotografías en las que aparece una taza situada en la esquina superior derecha de diferentes colores y con fondos distintos.

El **segundo ejemplo** de esta métrica lo adjunto a continuación. En este caso la imagen consulta está compuesta por tres objetos, clasificados bajo la misma etiqueta pero con

diferentes propiedades gráficas, que se corresponden con tres labiales. Si aplicamos esta métrica los resultados que debemos esperar consisten en un grupo de imágenes en las que, al menos, debe aparecer un labial en la misma posición.

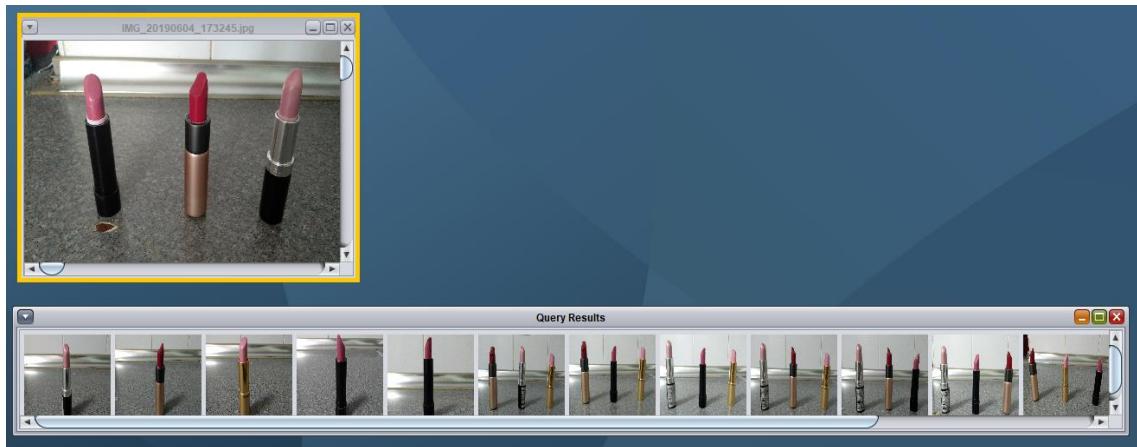


Figura 40: Segundo experimento de la tercera familia – Al menos uno de los objetos.

Si apreciamos la lista de imágenes resultante podemos comprobar como en todas aparece, al menos un pintalabios, situado en una de las posiciones en las que se encuentran los tres labiales en la imagen consulta. Es por ello por lo que las primeras fotografías muestran un labial situado en la parte central de la imagen. De este mismo modo las siguientes imágenes muestran tres labiales colocados en las mismas posiciones que en la imagen consulta.

4.1.3.2. *La mayoría de los elementos*

Este segundo comparador perteneciente a esta familia también tiene en cuenta la posición en la que se encuentran los objetos en la imagen consulta. No obstante, a diferencia de la métrica anterior, esta clasificará dos imágenes como parecidas si en ambas se encuentran los mismos objetos. De este modo dadas dos imágenes A y B se considerarán como similares si los elementos que aparecen en A los tiene B situados, además, en la misma posición. La fórmula matemática asociada a esta métrica es la siguiente:

$$d = \max (|A_i - B_i| \forall i)$$

Tal y como podemos intuir esta es la métrica más estricta dentro de esta familia de comparadores puesto que exige un enorme grado de similitud para considerar que dos imágenes se parezcan. En particular ambas imágenes deben tener los mismos objetos situados en las mismas posiciones. Lo único que puede variar son sus características gráficas.

El **primer experimento** que ilustra el funcionamiento de este comparador es el que se puede comprobar a continuación. En él tomaremos como imagen consulta aquella fotografía en la que aparece un mechero a la izquierda y una goma de borrar a la derecha. Los resultados esperados consisten en obtener un listado de imágenes en las que en todas aparezcan estos dos objetos situados en las mismas posiciones anteriores.

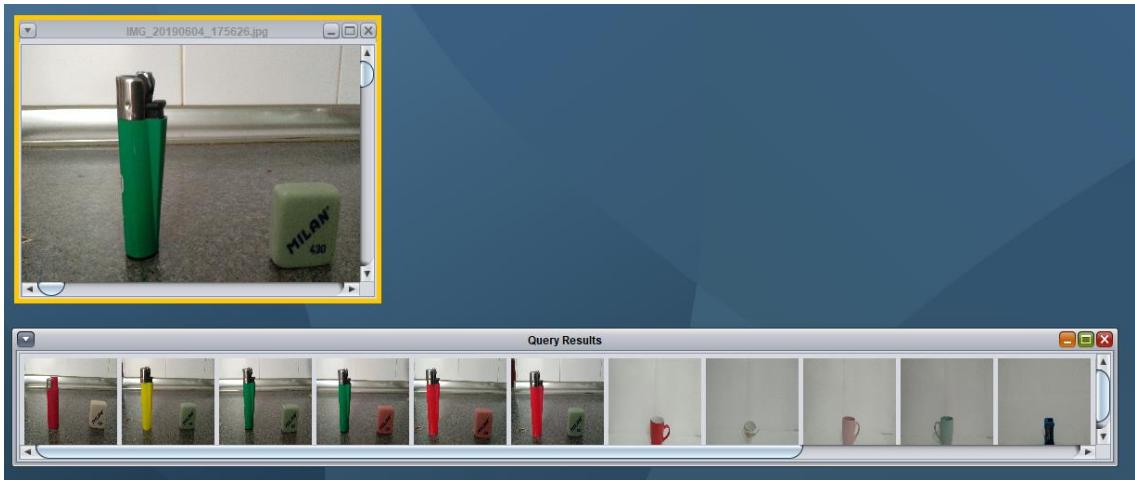


Figura 41: Primer experimento de la tercera familia – La mayoría de objetos.

Tal y como podemos observar aquellas imágenes que ocupan los primeros puestos en la lista son las únicas que cumplen todos los requisitos. Es decir, en todas ellas aparecen un mechero a la izquierda y una goma de borrar a la derecha, independientemente de sus características visuales. El resto de imágenes que aparecen, como no cumplen las restricciones descritas anteriormente, se ordenan en función de cuándo han sido consultadas.

Un **segundo experimento** aplicando este tipo de métrica es el que se ilustra en la siguiente figura. En él la imagen consulta estará compuesta por tres objetos, en este caso, se trata de tres mecheros de distintos colores. En teoría, las imágenes resultantes tras aplicar este comparador estarían compuestas por exactamente tres mecheros y ocupando las mismas posiciones que en la imagen consulta.

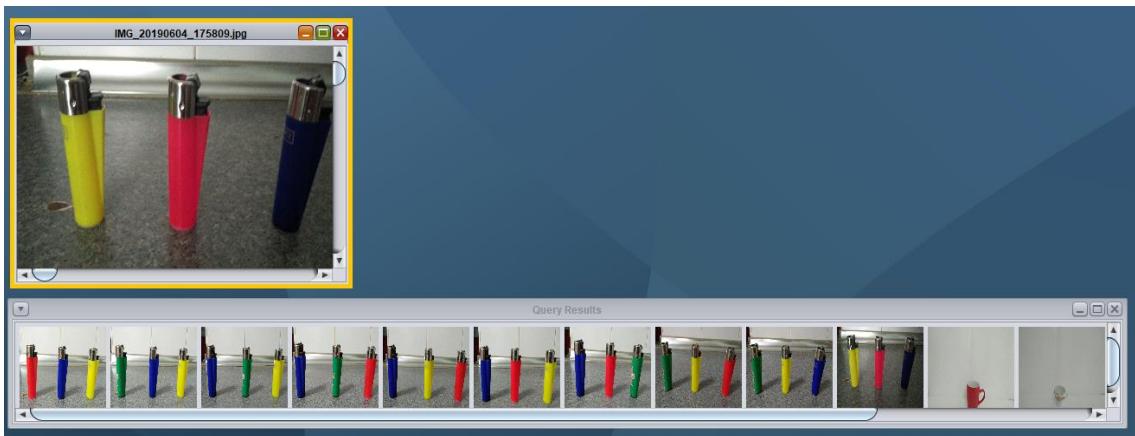


Figura 42: Segundo experimento de la tercera familia – La mayoría de los elementos.

Si observamos las fotografías obtenidas podemos comprobar que en todas ellas aparecen tres mecheros ocupando las mismas posiciones que en la imagen consulta pero de colores diferentes. De este modo podemos confirmar la teoría que expusimos anteriormente.

4.1.3.3. *En promedio*

En este último comparador de la tercera familia de métricas, como en los dos casos

anteriores, se considerará la posición de los objetos situados en la imagen consulta. No obstante este comparador, dadas dos imágenes A y B las clasificará como similares si los elementos de A aparecen, en media, en la imagen B y situados, además, en la misma posición. Si bien esta métrica también es estricta no lo es tanto comparado con la anterior. La fórmula matemática que ilustra el funcionamiento de este comparador se puede ver a continuación:

$$d = \text{mean} (|A_i - B_i| \quad \forall i)$$

Un **primer ejemplo** que demuestre el comportamiento de esta métrica es el siguiente que se puede observar. Para ello utilizaremos como imagen consulta una fotografía en la que aparecen un labial a la izquierda y una taza la derecha. Los resultados esperados estarán formados por un conjunto de imágenes en las que aparecerán estos mismos dos objetos de manera generalizada. Así mismo deberán ocupar la misma posición que en la imagen consulta.

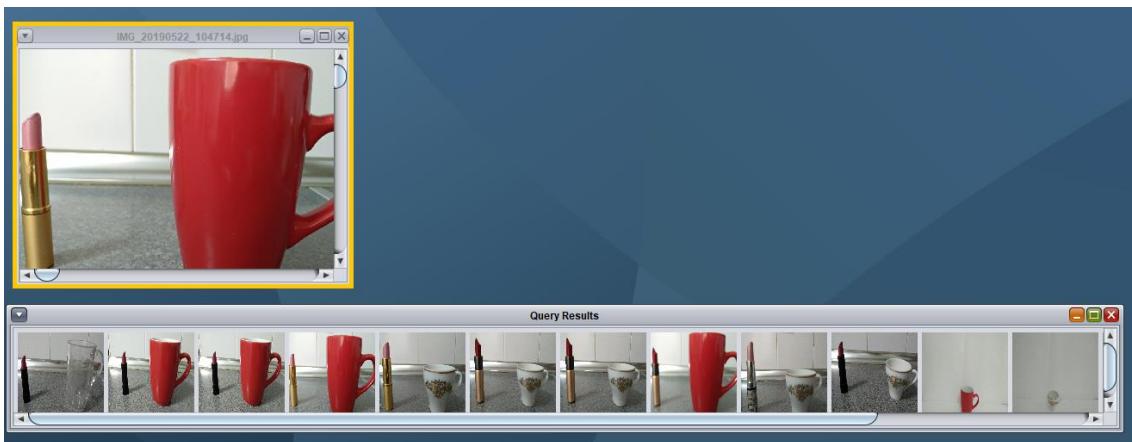


Figura 43: Primer experimento de la tercera familia de comparadores – En promedio.

Observando la lista de imágenes resultante podemos comprobar como en los primeros puestos de esta aparecen fotografías que comparten el mismo patrón que muestra la imagen consulta. Es decir, en todas, de manera general, aparecen un labial a la izquierda y una taza a la derecha, independientemente de su aspecto físico y del fondo de la imagen, ocupando las mismas posiciones que en la imagen consulta.

Para finalizar este capítulo ilustraremos el funcionamiento de este comparador con un **segundo ejemplo** que se muestra a continuación. En él establecemos como imagen consulta aquella en la que aparecen dos pelotas de tenis, cada una situada en un lateral de la imagen. El pronóstico para este experimento es el consistente en obtener imágenes en las que, de manera general, aparezcan dos pelotas de tenis cada una posicionada en un lado distinto de la imagen.

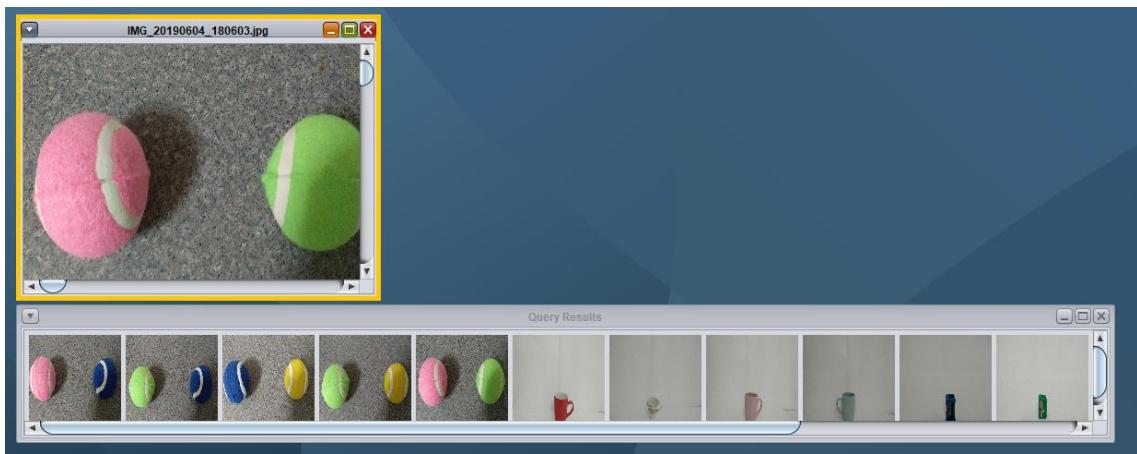


Figura 44: Segundo experimento de la tercera familia – En promedio.

Tal y como podemos comprobar las cuatro primeras imágenes son las que cumplen con los requisitos descritos anteriormente, y por ende, ocupan los cuatro primeros puestos en la lista. El resto de imágenes que no los satisfacen se encontrarán ordenadas en función de cuándo han sido consultadas.

Capítulo 5

Consultas en base a términos lingüísticos

Tras haber explotado el potencial atribuido al software que he desarrollado en base a la identificación de los elementos de una imagen de forma local a través de etiquetas lingüísticas, así como las distintas métricas explicadas e ilustradas para realizar diversas consultas en base a una imagen, en este capítulo me dispongo a detallar una novedosa mejora relacionada con este mismo ámbito. Esta consiste en brindar la posibilidad al usuario de realizar una búsqueda de imágenes especificando, para ello, un determinado término lingüístico. Existen dos formas de llevar a cabo este procedimiento: tomando la imagen al completo o de forma local a cada una de sus regiones.

5.1. Consulta global a la imagen.

Si bien el hecho de establecer una imagen, en concreto, como la protagonista de la consulta puede llegar a aportar numerosas ventajas, también puede resultar ser una tarea engorrosa el tener que buscar una fotografía para ejecutar una consulta. Por ello, con el objetivo de proporcionar una mayor flexibilidad a la hora de realizar consultas sobre imágenes, se ha introducido en este proyecto la posibilidad de recuperar fotografías en base a una etiqueta lingüística. Este procedimiento fue el que Google aplicó cuando introdujo en sus motores de búsqueda el proceso de etiquetado mediante el uso de la red neuronal convolucionada *ResNet*. Tal y como se comentó anteriormente, un modelo de este tipo de CNN es el que también estamos utilizando en este proyecto.

No obstante el tipo de búsqueda que desarrolla la gran compañía no es lo suficientemente precisa como para añadir parámetros adicionales referentes, por ejemplo, a la posición en la que deseamos que se encuentre el elemento buscado. Un ejemplo representativo de esta teoría se muestra en la siguiente figura, en la cual se ha realizado una búsqueda de una pelota de tenis situada a la derecha de la imagen.

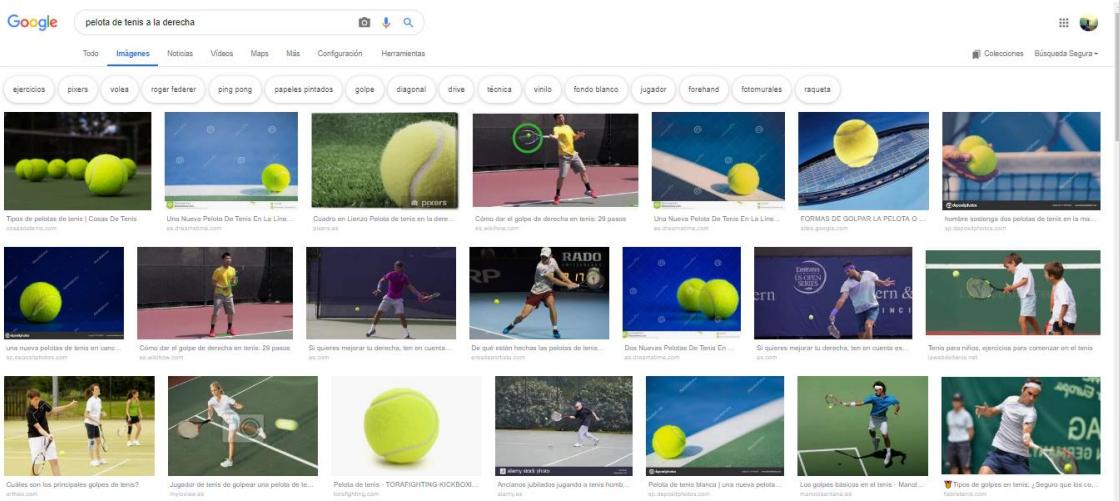


Figura 45: Búsqueda en Google de una pelota de tenis situada a la derecha de la imagen.

Tal y como podemos comprobar los resultados coinciden con la aparición de imágenes que contienen, al menos, una pelota de tenis, pero estos no demuestran que se esté tomando en cuenta la posición especificada. Cada una de las imágenes muestra este elemento en una ubicación distinta, que puede coincidir o no, con la que hemos concretado. Para llevar a la práctica este tipo de búsquedas, en las que se pueda especificar la localización de un determinado elemento, se necesita realizar un análisis local a la imagen para comprobar en qué posición se encuentra el objeto y si coincide o no con la establecida. Este ámbito será el que mi proyecto desarrolle e ilustre con ejemplos en el siguiente apartado.

5.2. Consulta local a la imagen.

En este modelo de consulta se lleva a cabo un análisis de las distintas regiones que componen una imagen con el fin de identificar los elementos que hay en ellas. Gracias a la división que se les aplica a las imágenes podemos tener en consideración una posición concreta en la que el objeto buscado debe encontrarse. Para ello basta con identificar las celdas correspondientes a la dimensión de la rejilla de la imagen y a la localización establecida para comprobar si en alguna de ellas existe, al menos, una etiqueta vinculada al término lingüístico buscado.

Con el fin de demostrar la versatilidad que proporciona este tipo de consultas procedo a adjuntar diversos ejemplos ilustrativos en los que participan distintos objetos en diferentes posiciones. En el **primero de ellos** se va a repetir la consulta que se había realizado con anterioridad en el motor de búsqueda de Google. Para ello seleccionamos, de entre la lista de 1000 términos que la CNN puede reconocer, la etiqueta asociada a la pelota de tenis y le especificamos que debe encontrarse a la derecha.



Figura 46: Consulta de una pelota de tenis situada a la derecha.

Tal y como podemos apreciar en la captura anterior, en este caso, mi proyecto sí que ha sido capaz de realizar la consulta de forma correcta, puesto que todas las imágenes muestran pelotas de tenis, de diversos colores, situadas a la derecha de la imagen. Cabe destacar que, como por ahora solo le estamos indicando una única posición, el hecho de especificar que esté a la derecha de la imagen implica que el elemento se puede encontrar tanto arriba a la derecha, abajo a la derecha o en la parte central de dicho lado.

Un **segundo ejemplo** de una búsqueda basada en un elemento restringido a ocupar una determinada posición se representa con el siguiente experimento. En ella se realizará una consulta en base a la etiqueta asociada a una copa, independientemente de su uso y sus características. La posición en la que deberá de encontrarse estará restringida a la parte de arriba de la imagen.

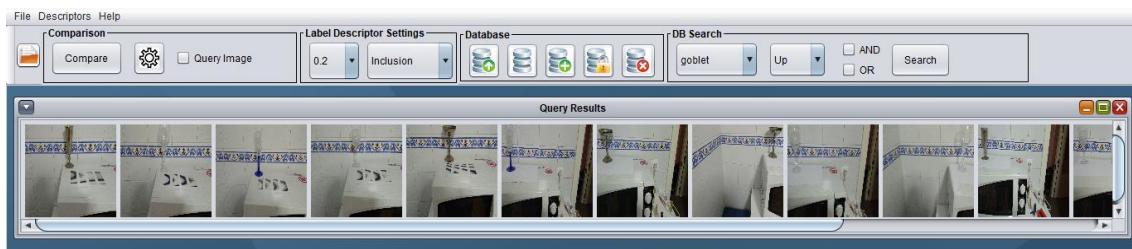


Figura 47: Consulta de una copa de beber situada en la parte superior de la imagen.

De nuevo volvemos a obtener un resultado satisfactorio acorde a los parámetros de búsqueda especificados. En todas las imágenes aparece una copa, independientemente de sus características físicas, y además está situada en la parte superior. Tal y como sucedía en el anterior experimento hay copas que aparecen en el centro de la parte superior de la imagen, otras a la derecha o a la izquierda de la localización establecida.

Como he comentado anteriormente, estos experimentos se han realizado en base a una única posición, sin embargo mi proyecto permite que el usuario establezca una combinación en base a dos localizaciones. Para ello utilizaremos dos principales operadores: uno en el que ambas posiciones se fusionarán para convertirse en una sola, como por ejemplo, que el objeto se encuentre abajo y a la derecha. Y un segundo operador más flexible que permitirá recuperar imágenes en las que el elemento buscado se encuentre en la primera o en la segunda posición.

5.2.1. Operador AND

El primer operador que procedo a explicar es capaz de buscar un determinado elemento en la combinación resultante de mezclar dos posiciones. Por supuesto ambas posiciones deben ser complementarias y razonables, es decir, no se podría realizar una consulta en la que el objeto estuviese “arriba y abajo” puesto que va en contra de las leyes de la física.

Para llevar a cabo este tipo de consulta basta con aplicar la operación *AND* a las regiones pertenecientes a una posición y a las asociadas a la otra con el objetivo de realizar la búsqueda en aquellas celdas comunes a ambas localizaciones. Esta operación también es propia de los conjuntos algebraicos, y por ende, puede representarse con la siguiente principio matemático:

Dados dos conjuntos A y B el resultado de aplicar la intersección se describiría como un conjunto $A \cap B = \{(a \in A) \wedge (a \in B), (b \in A) \wedge (b \in B)\}$.

Con el propósito de ilustrar su funcionamiento procedo a adjuntar, a continuación, diversos ejemplos. En el **primer ejemplo** se realiza una consulta en función del elemento conocido como sacapuntas, el cual estará restringido a la localización establecida como abajo y a la izquierda.

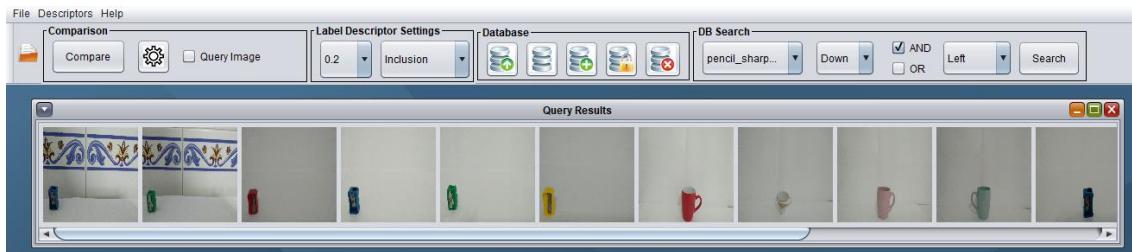


Figura 48: Consulta de un sacapuntas situado en la esquina inferior izquierda de la imagen.

Tal y como podemos apreciar en la captura las primeras fotografías situadas en la lista resultante son todas aquellas que cumplen con los requisitos especificados. La razón de ello consiste en que cada una de ellas muestra un sacapuntas, de diversos colores, en la esquina inferior izquierda tal y como se ha establecido. Una vez que no se hayan encontrado más imágenes que cumplan este patrón se muestran el resto de fotografías en el orden en el que se consultaron.

El **segundo ejemplo** se corresponde con un experimento en el que se va a realizar una búsqueda distinta. En este caso se consulta por una taza en la esquina superior derecha y su resultado se puede apreciar en la siguiente captura.



Figura 49: Consulta de una taza situada en la esquina superior derecha de la imagen.

De nuevo conseguimos un resultado satisfactorio en el que las primeras imágenes contienen una taza, de distintas cualidades gráficas y situadas en fondos diferentes, pero cuya posición se ajusta a la especificada como “a la derecha y arriba”. Posteriormente, como ya es habitual, se encuentran el resto de imágenes que no cumplen con los requisitos.

Como **tercer ejemplo** desarrollaremos el último experimento anterior a la explicación de este operador para proporcionar una posición más exacta. De este modo buscaremos el mismo objeto que en el ejemplo mencionado, una copa, cuya primera posición estará restringida al centro de la imagen. Sin embargo, gracias a la precisión que nos brinda las combinaciones que realiza este operador en base a dos localizaciones, realizaremos dos consultas basadas en una posición centrada pero en la parte inferior y otra en la parte superior. Los resultados de ambas consultas se pueden apreciar en la siguiente captura.

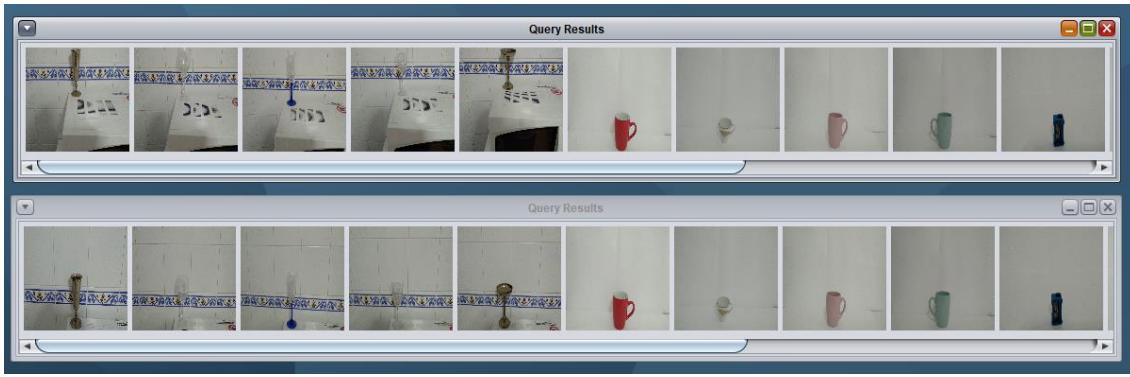


Figura 50: Consulta doble de una copa situada en la parte superior central de la imagen y otra situada en la parte inferior central.

Tal y como podemos observar la primera lista de imágenes se corresponde con la posición central superior ya que en las primeras fotografías podemos apreciar este elemento centrado en la parte de arriba de la imagen. Por el contrario, en la lista que se encuentra debajo las primeras imágenes se corresponden con la consulta en base al mismo objeto pero centrado en la parte inferior de la imagen. En base a esta posibilidad se puede determinar que la peculiaridad más relevante de este operador es la capacidad de precisión que brinda al usuario a la hora de escoger una posición combinada muy concreta.

5.2.2. Operador OR

En este otro operador la filosofía aplicada es parecida, puesto que se tiene en cuenta tanto la etiqueta del elemento buscado como las dos posiciones seleccionadas. No obstante, la combinación que se realiza con ambas es bien distinta. En este caso el objeto puede ajustarse a una posición u a otra para que la imagen forme parte del resultado. En base a ello podemos determinar que este operador es el más flexible de los dos, aunque también pertenece a una de las muchas operaciones que se pueden realizar con conjuntos algebraicos. Se trata de la operación comúnmente conocida como la *unión* entre conjuntos. En este operador se buscará el término seleccionado entre todas las celdas pertenecientes tanto a la primera como a la segunda localización definida. De nuevo, su funcionamiento puede describirse en base al siguiente principio matemático:

Dados dos conjuntos A y B el resultado de aplicar la unión se describiría como un conjunto $A \cup B = \{(a \in A) \vee (b \in B)\}$.

A continuación procedo a mostrar diversos ejemplos en los cuales se realizarán búsquedas en función de elementos distintos situados en localizaciones diferentes a partir de la combinación de las dos posiciones que se especifiquen. El **primer ejemplo** tiene como objetivo principal buscar un labial situado en la parte superior o inferior de la imagen. Los resultados que se consiguen se pueden apreciar en la siguiente captura.

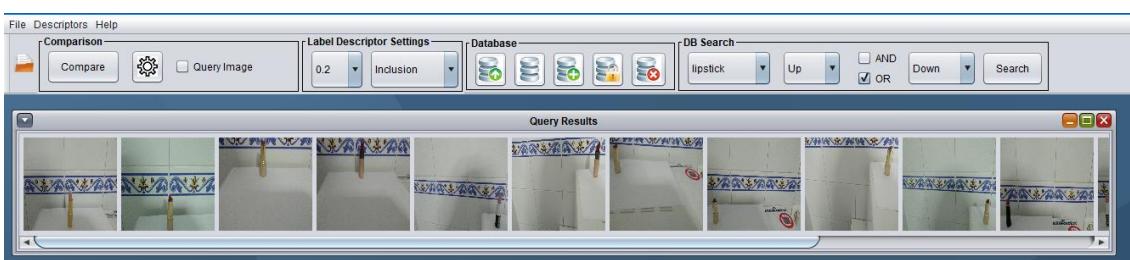


Figura 51: Consulta de un labial situado en la parte superior o inferior de la imagen.

Tal y como podemos comprobar en todas las imágenes resultantes se encuentra un pintalabios situado, bien en la parte de arriba o en la de debajo de la imagen. Este ejemplo, además de ser un experimento satisfactorio que demuestra el funcionamiento de este operador, también indica que a diferencia del operador anterior, todas las combinaciones entre ambas posiciones son válidas. Es decir, mientras que en el otro operador era imposible encontrar un objeto arriba y abajo, en este caso sí es posible manejar sendos términos gracias a que realiza la búsqueda en ambos sitios.

Continuamos con un **segundo ejemplo** ilustrativo que representa la ya mencionada flexibilidad que caracteriza a este operador. En este caso el elemento a consultar es un pimiento, independientemente de su color, el cual va a estar condicionado a que se encuentre en la parte derecha o izquierda de la imagen. En la siguiente captura podemos comprobar los resultados obtenidos de la búsqueda realizada.

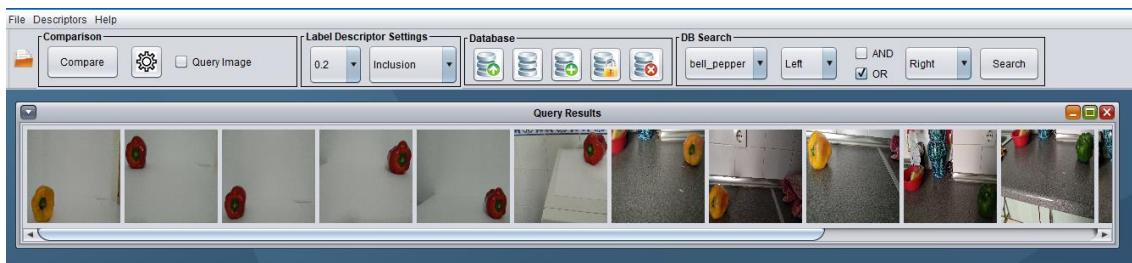


Figura 52: Consulta de un pimiento situado a la derecha o a la izquierda de la imagen.

En cada una de las imágenes aparece un pimiento, de diverso color y con distinto fondo, pero todos se encuentran en la parte derecha o izquierda de la imagen. Si bien es cierto que algunos de los elementos se encuentran en la esquina superior o inferior de uno de los laterales de la foto, este suceso no se puede controlar con este operador. A diferencia del anterior en el que se podían combinar ciertas posiciones para obtener una sola, como por ejemplo “abajo y a la izquierda”, a razón de la operación de conjuntos que utiliza el operador **OR**, esta posibilidad no puede ser realizada.

No obstante, gracias a su flexibilidad se pueden realizar un conjunto de consultas que con el anterior operador no se podían llevar a cabo. Recordemos que en el operador **AND** dependiendo de la primera posición que escogísemos las segundas localizaciones se ajustan con el fin de construir una posición correcta. Es por ello por lo que si en primer lugar se escoge, por ejemplo, una ubicación central luego en el segundo rango de posiciones las únicas posibilidades son arriba o abajo. La razón de ser de este criterio se fundamenta en que un objeto no puede estar centrado y a la vez situado en un lateral de la imagen, ya que va en contra de los principios básicos de la física. Sin embargo sí que podemos realizar una consulta en la que el objeto buscado se encuentre en alguna de las dos posiciones. Este hecho nos da pie al **tercer ejemplo** en el que dicho elemento a encontrar será un mechero, el cual deberá posicionarse centrado en la imagen o en el lateral derecho. La lista resultante se puede observar a continuación.

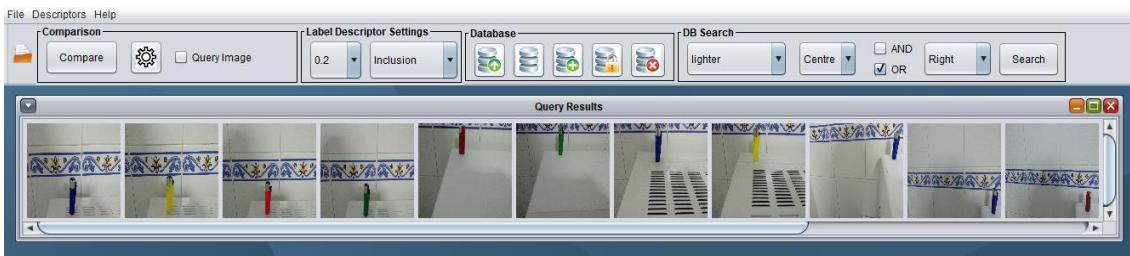


Figura 53: Consulta de un mechero situado en la parte central o en el lateral derecho de la imagen.

En este caso todas las imágenes que se muestran contienen el elemento buscado. En las primeras el objeto se sitúa en la parte central de la imagen, no pudiendo especificar si se encuentran en la parte central inferior o superior. Y en las restantes se puede comprobar que el mechero se sitúa a lo largo del lateral derecho de la imagen, de nuevo sin poderle concretar al operador si debe colocarse en su parte superior o inferior, puesto que para ello debemos utilizar el anterior operador.

Capítulo 6

Base de datos

Con el objetivo de demostrar el funcionamiento de todas las anteriores funcionalidades desarrolladas ha sido un requisito indispensable el hecho de tener que formar mi propia base de datos de imágenes. Por ello este capítulo se dedicará a explicar los principales aspectos que se han tenido en cuenta para llevar a cabo esta tarea.

14.1. Requisitos considerados

Existen un amplio y diverso número de condiciones que se han debido estimar a la hora de plantear la confección de la base de datos. A continuación se explican las que han sido más relevantes:

- **Distintos objetos reconocibles por la CNN.** Tal y como se podía esperar el principal dilema que me he encontrado ha sido el de realizar una lista de todos aquellos objetos que la red neuronal era capaz de reconocer y cuáles de ellos estaban a mi alcance. Este último aspecto lo comento puesto que, si bien es cierto que existen 1000 etiquetas lingüísticas relacionadas con las 1000 cosas que puede identificar, no todas ellas son accesibles ni cumplen el siguiente requisito.
- **Objetos de distintas propiedades gráficas.** Con el fin de demostrar las distintas mejoras y diversas ventajas que proporciona el uso de las etiquetas lingüísticas frente a la tarea consistente en extraer las características visuales de las imágenes, he tenido que buscar objetos que, además de cumplir el requisito anterior, también hubiese disponibles varios modelos de diferentes aspectos. La razón de ello, tal y como se ha explicado en este proyecto, es que la gran desventaja de realizar consultas en base a las cualidades gráficas es que si un objeto dispone de diversos modelos con distintos colores, texturas, entre otras propiedades, las consultas no son capaces de adaptarse. No obstante este hecho no ocurre con los términos lingüísticos.
- **Diferentes localizaciones.** Para demostrar el funcionamiento tanto de aquellas métricas que toman en cuenta la posición de los objetos como de la consulta en base a una etiqueta y una localización, he tenido que realizar diversas fotos a cada uno de los objetos situándolo en cada una de las posibles posiciones.

- **Combinaciones del mismo u otro tipo de objetos.** Con el propósito de demostrar el funcionamiento de ciertos comparadores he tenido que combinar elementos relacionados con la misma o distinta etiqueta lingüística. Por ello he tenido que ir variando y fotografiando las distintas combinaciones posibles a la vez que cambiaba los modelos de cada uno de los objetos fotografiados.
- **Búsqueda de imágenes en Internet.** Por último, de manera complementaria, he realizado diversas búsquedas en Internet para adjuntar fotos a mi base de datos que me permitiesen, por ejemplo, redactar la motivación de este proyecto. Para ello he llevado a cabo consultas en función de las propiedades gráficas y no de los objetos en sí. Así mismo también he incorporado algunas fotografías de alimentos buscadas en Internet con el objetivo de explicar el funcionamiento de uno de los comparadores pertenecientes a la última familia.

6.2. Creación de la base de datos de imágenes

Tras realizar esta progresiva reflexión de los aspectos a considerar la estructura de carpetas en la que se encuentran las fotografías tiene como primera referencia el número de objetos que hay. De este modo existen tres principales categorías:

- *1 objeto.* En ella se almacenan todas aquellas fotografías en las que solamente aparece un único elemento. A su vez, dentro de ella, existe una segunda clasificación para ordenar las imágenes en función de la posición que ocupe el objeto que aparece en ellas. Así podemos encontrarnos las siguientes clases:
 - *Abajo.* Aquí se albergan aquellas imágenes cuyos únicos objetos se encuentran en la parte inferior central de la imagen.



Figura 55: Mechero situado en la parte de debajo de la imagen.



Figura 54: Pimiento situado debajo de la imagen.

- *Arriba.* En esta segunda clase se almacenan todas las fotografías cuyos únicos objetos se sitúen en la parte superior central de la foto. Como en el caso anterior, no se realiza una distinción en base al tipo de objeto que es.



Figura 56: Taza situada arriba de la imagen.



Figura 57: Pelota de tenis situada arriba de la imagen.

- *Centro*. La tercera categoría de esta segunda clasificación guarda todas aquellas imágenes que ocupen el espacio central de las fotografías.



Figura 59: Labial situado en el centro de la imagen.



Figura 58: Sacapuntas situado en la parte central de la imagen.

- *Laterales*. En esta clase se almacenan aquellas fotografías en las que aparezca un único objeto en la parte derecha o izquierda de la imagen.



Figura 60: Copa situada a la derecha.



Figura 61: Goma de borrar situada la izquierda.

- *Esquinas*. En esta última carpeta se encuentran todas aquellas imágenes en las que aparezca un único objeto situado en alguna de las esquinas de la foto.



Figura 62: Taza situada a la derecha de la imagen.



Figura 63: Taza situada a la derecha de la imagen.

- *2 objetos*. En esta categoría se almacenan todas aquellas fotografías en cuyas escenas aparecen dos objetos de cualquier tipo.



Figura 64: Dos labiales.



Figura 65: Imagen de una pelota y una taza.

- *3 objetos.* Aquí se guardan todas las imágenes en las que aparezcan tres objetos, sean de distinto tipo o no.



Figura 67: Imagen de una goma de borrar y dos sacapuntas.



Figura 66: Imagen de tres mecheros.

- *Internet.* En esta carpeta se almacenan todas aquellas imágenes que han sido buscadas en Internet para completar esta base de datos.



Figura 69: Campo de margaritas.



Figura 68: Cuenco de fresas.

Parte II

Desarrollo e implementación del software.

Capítulo 7

Organización y presupuesto

En este capítulo se comentarán todos los aspectos relacionados con la planificación temporal asociada a este proyecto así como las tareas realizadas mediante un diagrama de *Gantt*. En él se verá reflejado el tiempo invertido en cada una de las actividades que se han ido integrando hasta conformar el proyecto actual.

Para desarrollar e implementar cada uno de los ámbitos relacionados con este proyecto se ha escogido la metodología de desarrollo software tradicional por sus numerosas ventajas relacionadas con su concreta estructuración en etapas. Esta metodología es la que mejor se adapta a mi proyecto puesto que cada una de sus tareas se ha ido desarrollando en diferentes períodos. En base a esto podemos esquematizar este segundo bloque en las siguientes fases que conforman esta metodología elegida:

1. Requisitos.
2. Casos de uso.
3. Análisis.
4. Diseño.
5. Implementación.

Así mismo se incluirá un presupuesto acerca del tiempo invertido tanto para desarrollar este proyecto como para realizar las distintas fotos que componen la base de datos de imágenes. También se ha incluido un manual de usuario en el que se expliquen las distintas funcionalidades del software que pueden ser probadas a través de los distintos elementos que conforman la interfaz. Para ello, además, se han incluido tres ejemplos en los que se detallan los pasos a seguir para realizar los tres tipos de consultas.

7.1. Diagrama de Gantt

El diagrama de Gantt es una herramienta que representa el conjunto de actividades que deben realizarse y el tiempo invertido en cada una de ellas para organizar el desarrollo de un proyecto. En la parte izquierda del diagrama se situarán las diversas tareas a realizar y en la parte superior se encontrará una línea del tiempo que marcará el inicio y el final del desarrollo del proyecto. Mediante barras horizontales se reflejará el espacio temporal invertido en cada una de las actividades que han debido realizarse [15]. El diagrama de Gantt asociado a este proyecto se puede observar en la siguiente figura.

Diagrama de Gantt

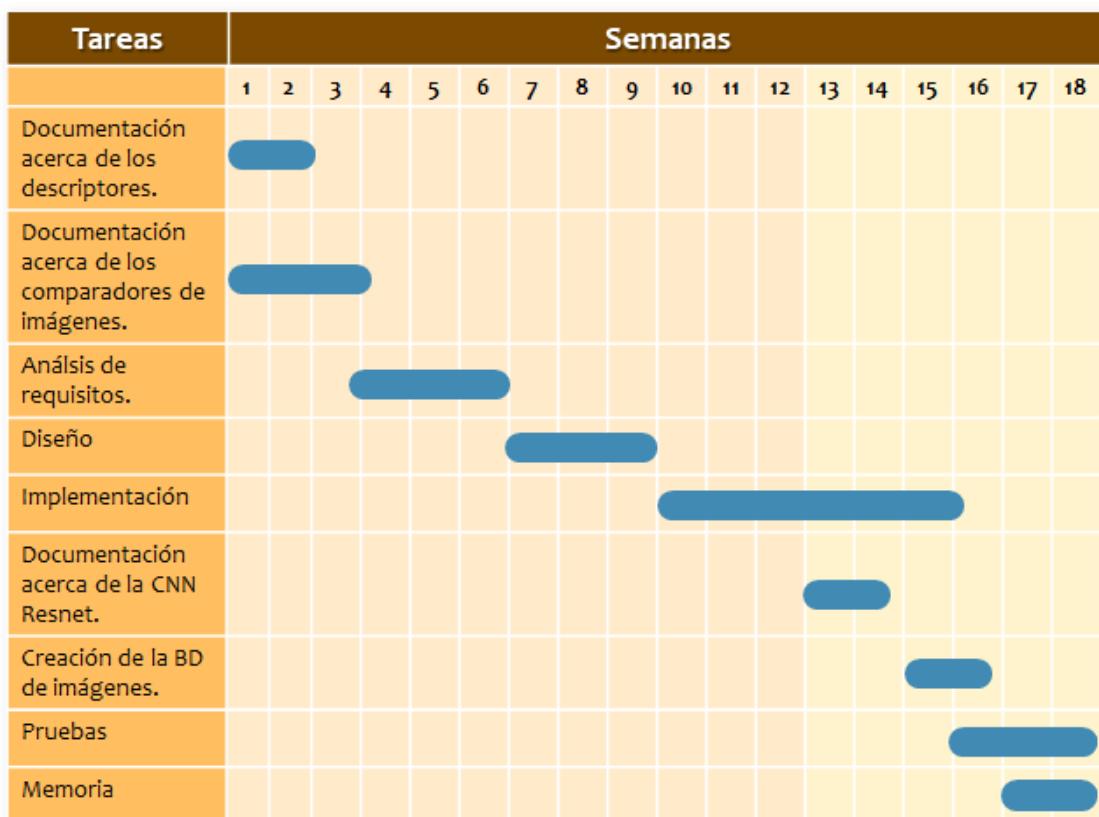


Figura 70: Diagrama de Gantt.

Los dos primeros pasos que se realizaron al comienzo del proyecto consistían en documentarse acerca de la estructura y el funcionamiento de los descriptores existentes. A continuación se comenzó a plantear las funcionalidades que mi proyecto debía contener para, posteriormente, realizar un diseño que permitiese desarrollarlas. Una vez tanto los requisitos como la estructura del proyecto estaban claros se comenzó a implementar una primera versión del sistema. Posteriormente, con la introducción de la CNN Resnet y del descriptor asociado a la generación de las etiquetas lingüísticas se inició una segunda fase enfocada al desarrollo de consultas en base a conceptos lingüísticos. Así mismo, también empezó una nueva fase paralela consistente en formar mi propia base de datos de imágenes con el propósito de demostrar la bondad de las operaciones que se iban implementando de manera progresiva.

Para terminar, comenzó la última fase consistente en redactar esta memoria invirtiendo, para ello, 15 días aproximadamente.

7.2. Presupuesto

Con el propósito de calcular el presupuesto estimado de este proyecto se tomarán en consideración diversos aspectos tales como el tiempo invertido, el personal implicado y la tecnología utilizada para su desarrollo.

En referencia al tiempo utilizado para el desarrollo de este proyecto podemos determinar, que en términos generales, se han invertido un total de 4 meses. En particular, de manera aproximada, se han trabajado unas 4 horas diariamente durante el espacio temporal especificado. Por lo tanto, podemos realizar el cálculo de las horas trabajadas en total, el cual podría representarse de la siguiente forma: 4 horas/día * 30 días/mes * 4 meses = 480 horas. Este es el tiempo total que una persona, con unos conocimientos suficientemente amplios como para comprender la temática del proyecto, ha invertido en llevar a cabo todas las tareas relacionadas con el desarrollo de este, tales como la investigación de los distintos ámbitos que abarca y el tiempo empleado tanto para el análisis, diseño, desarrollo y testeo del software.

Estimando, aproximadamente, que el sueldo de un ingeniero *junior* ronda los 1000 euros al mes, el gasto dedicado solamente a la parte de desarrollo es de 4000 euros. A este se le deberá sumar la inversión monetaria que se ha realizado en contratar un fotógrafo para tomar todas las fotografías necesarias y confeccionar, así, la base de datos de imágenes utilizada. Para realizar el cálculo pertinente deberemos considerar el número de horas invertidas en tomar las imágenes, las cuales han sido alrededor de 40 horas, además de la tarifa del fotógrafo, el cual puede llegar a cobrar hasta 40 euros por hora. Por lo tanto el gasto invertido solamente en la realización de las fotografías es de 800 euros. En términos generales, el desembolso invertido en el personal mencionado anteriormente asciende a 4800 euros.

A este presupuesto estimado hay que sumarle el coste de la tecnología usada, la cual en este caso ha sido un portátil cuyo precio aproximado es de 650 euros. Por lo tanto, sumando los costes relacionados con los trabajadores que han sido necesarios para realizar el software y tomar las fotografías, además de los dispositivos informáticos utilizados, **el presupuesto total del proyecto ronda los 5450 euros.**

Capítulo 8

Requisitos

En este capítulo se detallará la lista de funcionalidades que el sistema es capaz de realizar. Para ello llevaremos a cabo un análisis de requisitos con el fin de estudiar las necesidades de los usuarios finales a los que va destinado este sistema. Estos requisitos se clasifican en tres categorías: requisitos de datos, requisitos funcionales y no funcionales.

8.1. Requisitos de datos

En este apartado se explicarán los requerimientos asociados a los datos que nuestro sistema CBIR tendrá que gestionar. Este apartado tiene una notoria relevancia puesto que este tipo de información se corresponde con la base principal para el correcto funcionamiento del sistema. Sin unos requisitos correctos de información el CBIR no será capaz de llevar a cabo ninguna de las funcionalidades de las que dispone. Por ello, a continuación se procede a explicar cada uno de estos datos adjuntando, además, el papel que desempeña:

1. Una estructura de datos que represente la información que almacena una imagen.
2. Una base de datos que sea capaz de almacenar el descriptor global de cada una de las imágenes además de su dirección URL para poder acceder a la fotografía.
3. Una estructura de datos que guarde el objeto relacionado con el clasificador que se va a utilizar para identificar a los objetos que aparecen en una imagen.
4. Un descriptor capaz de almacenar un conjunto de términos lingüísticos asociados a una región de la imagen.
5. Una estructura de datos que sea capaz de almacenar tantos descriptores como regiones tenga una imagen.

8.2. Requisitos funcionales

En esta segunda categoría se explicará el conjunto de actividades que nuestro sistema es capaz de llevar a cabo. Su principal pilar se fundamenta en la serie de requisitos de datos descritos anteriormente puesto que se hará uso de ellos en cada una de las tareas que desarrolle nuestro prototipo CBIR. De este modo procedo a detallar la lista de este tipo de requerimientos:

1. Abrir y cerrar una imagen o conjunto de imágenes.
2. Mostrar una imagen en una ventana interna así como presentar la lista de imágenes resultante tras realizar una consulta en otro tipo de ventana interna.
3. Posibilidad de seleccionar uno de los descriptores disponibles para calcularlo a cada imagen.
4. Realizar consultas en base a una imagen consulta, seleccionada previamente, comparándola con el resto de imágenes abiertas en la aplicación. Para ello se deberán calcular los descriptores de cada una de ellas.
5. Cambiar los parámetros asociados a la ejecución de una consulta, tales como la dimensión del descriptor o el tipo de comparador.
6. Crear una nueva base de datos de descriptores, abrir una ya existente, guardarla en un fichero y cerrarla.
7. Añadir uno o varios registros a la base de datos formados por el descriptor de la imagen y su dirección URL.
8. Realizar consultas en base al descriptor de una imagen consulta, seleccionada previamente, comparándolo con el resto de descriptores almacenados en una base de datos.
9. Importar y cargar el fichero de un clasificador leyendo las etiquetas lingüísticas disponibles.
10. Modificar el umbral del clasificador.
11. Generar un descriptor compuesto por los términos lingüísticos asociados a los objetos reconocidos de una imagen.
12. Mostrar el descriptor de conceptos lingüísticos asociado a una única imagen en una ventana o a una fotografía dentro de una lista de imágenes.
13. Seleccionar un concepto lingüístico asociado a un objeto para realizar una consulta en la base de datos de imágenes.
14. Posibilidad de especificar una sola posición o una combinación de dos para restringir la localización en la que debe situarse el objeto buscado.

8.3. Requisitos no funcionales

En esta última categoría procedo a analizar aquellos requerimientos que están relacionados

con las restricciones a las que está sujeto el sistema y que no están vinculadas directamente con el funcionamiento de este. No obstante gozan de una notoria importancia puesto que, también, de estos requerimientos depende el buen funcionamiento del sistema. A continuación se detallan los más relevantes:

1. Una base de datos debe estar siempre compuesta por descriptores del mismo tipo aunque no con el mismo tamaño.
2. Para realizar una consulta se debe seleccionar un descriptor de entre los existentes en la lista de descriptores, además de una imagen consulta o un concepto lingüístico de la lista de etiquetas válidas.
3. La lista de imágenes resultante tras realizar una consulta debe mostrar las fotografías en un orden descendente en base a su grado de similitud con la imagen consulta o la etiqueta buscada.
4. El sistema debe de proporcionar una respuesta en un tiempo prudencial.
5. El sistema debe estar disponible en todos aquellos momentos en los que los usuarios abran la aplicación para hacer uso de él.
6. La fiabilidad del CBIR debe ser lo suficientemente buena como para proporcionar unos resultados adecuados a las actividades que se haya realizado.
7. El sistema debe ser capaz de adecuar las opciones disponibles en función de las que se han seleccionado previamente, dentro de un mismo ámbito.
8. La capacidad de almacenamiento ya ha sido especificada en el apartado relacionado con los requisitos de datos.

Capítulo 9

Casos de uso

En este capítulo representaremos gráficamente el conjunto de las funciones principales que se pueden llevar a cabo en este sistema. Para ello se adjuntarán, en primer lugar, un *diagrama de casos de uso* para ilustrar la interacción entre el usuario y las actividades que le proporciona el sistema. Así mismo se detallan los *casos de uso* que representan el flujo de interacción que se produce cuando un usuario activa una de las funcionalidades del sistema.

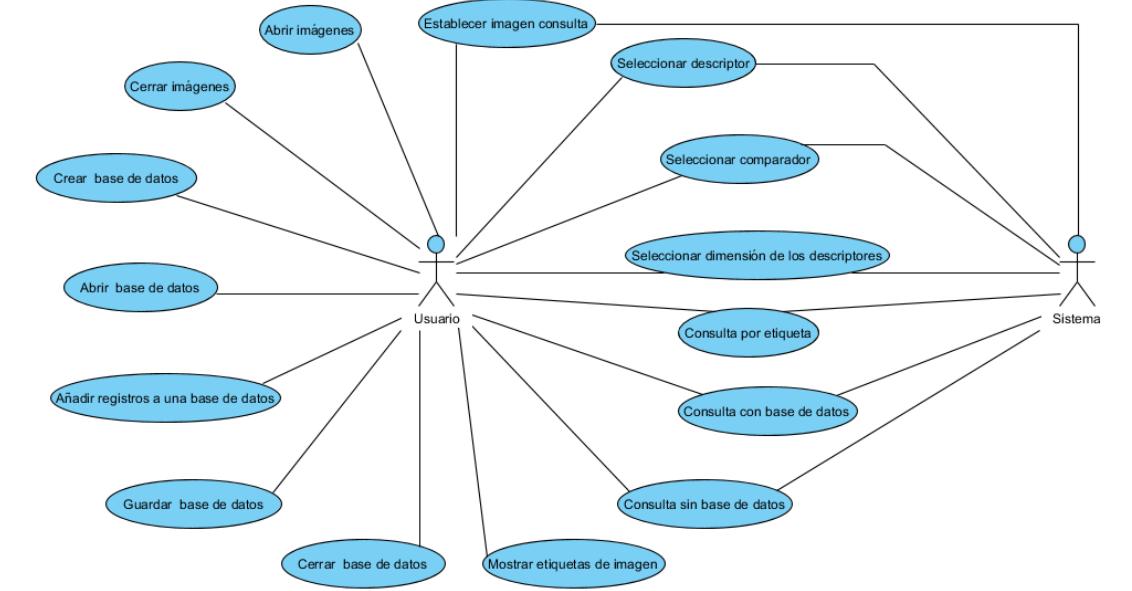
9.1. Actores

Comenzamos explicando que el único tipo de actor que interactúa con este sistema son los usuarios que lo utilizan. No obstante, tal y como podemos comprobar en la siguiente descripción relacionada con este tipo de actor, los usuarios no necesitarán conocimientos previos para usar el CBIR.

Actor	Usuario	AC-1
Descripción	Persona que hace uso del sistema.	
Características	No tiene por qué poseer características particulares de ningún ámbito. Este sistema puede usarlo cualquier tipo de persona.	
Relaciones		
Referencias	CU-1, CU-2, CU-3, CU-4, CU-5, CU-6, CU-7, CU-8, CU-9, CU-10, CU-11, CU-12, CU-13, CU-14	
Comentarios	Para usar el sistema ningún usuario deberá introducir datos personales o realizar ningún tipo de registro.	

9.2. Diagrama de casos de uso

A continuación presentamos el diagrama que muestra los diferentes casos de uso con los que los usuarios pueden interactuar. Además también se muestran aquellos en los que el sistema interviene de manera directa para llevarlos a cabo. Este tipo de casos de uso están relacionados con las actividades automáticas que realiza el sistema, como por ejemplo, los tipos de consultas.



9.3. Descripciones de los casos de uso

En base al diagrama anterior a continuación adjuntamos las descripciones de cada uno de los anteriores casos de uso.

Caso de uso	<i>Abrir imágenes.</i>	
Nº de caso de uso	<i>CU-1.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Primario, real.</i>	
Referencias	<i>RD-1, RF-1, RF-2.</i>	
Descripción	<i>El usuario podrá elegir una o varias imágenes de su sistema mediante un cuadro de diálogo con el fin de abrir las en la aplicación.</i>	
Precondición		
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para abrir una o varias imágenes.</i>
	2	<i>El sistema le muestra un cuadro de diálogo para que el usuario navegue por su sistema.</i>
	3	<i>El usuario selecciona una imagen o un conjunto de ellas.</i>
	4	<i>El sistema verifica que el fichero contenga una extensión válida.</i>
	5	<i>El sistema abre las imágenes seleccionadas.</i>
Postcondición	<i>Cada imagen se abrirá en una ventana interna individual.</i>	
Excepciones	Paso	Acción
	5a	<i>La imagen seleccionada no se puede abrir puesto que no dispone de una extensión válida.</i> <ul style="list-style-type: none"> • <i>El sistema informa sobre el suceso.</i>
	5b	<i>La imagen no se puede abrir debido a un error interno del fichero.</i> <ul style="list-style-type: none"> • <i>El sistema informa acerca del incidente.</i>
Rendimiento		
Frecuencia esperada		
Importancia	<i>Vital.</i>	
Urgencia	<i>Alta.</i>	
Comentarios		

Caso de uso	<i>Cerrar imágenes.</i>	
Nº de caso de uso	<i>CU-2.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Primario, real.</i>	
Referencias	<i>RF-1, CU-1.</i>	
Descripción	<i>El usuario podrá cerrar una imagen en concreto o todas las imágenes que estén abiertas en la aplicación.</i>	
Precondición	<i>Deben haberse abierto las imágenes previamente.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón de cerrar de una imagen en concreto.</i>
	2	<i>El sistema cierra una imagen determinada.</i>
Postcondición	<i>La imagen o imágenes cerradas ya no se encuentran visibles en la aplicación.</i>	
Excepciones	Paso	Acción
	1b	<i>El usuario pulsa sobre el botón integrado en la aplicación para cerrar todas las imágenes de la aplicación.</i>
	2b	<i>El sistema cierra todas las imágenes abiertas en la aplicación.</i>
Rendimiento		
Frecuencia esperada		
Importancia	<i>Baja.</i>	
Urgencia	<i>Baja.</i>	
Comentarios		

Caso de uso	<i>Seleccionar un descriptor.</i>	
Nº de caso de uso	<i>CU-3.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Primario, real.</i>	
Referencias	<i>RF-3, RNF-2, RNF-7.</i>	
Descripción	<i>El usuario podrá seleccionar uno de los descriptores disponibles.</i>	
Precondición		
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el menú que contiene la lista de descriptores.</i>
	2	<i>El sistema le muestra la lista de descriptores disponibles.</i>
	3	<i>El usuario selecciona uno de los descriptores.</i>
Postcondición	<i>El descriptor quedará seleccionado.</i>	
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	<i>Vital.</i>	
Urgencia	<i>Alta.</i>	
Comentarios	<i>Solo un descriptor podrá estar seleccionado a la vez.</i>	

Caso de uso	<i>Crear una nueva base de datos.</i>	
Nº de caso de uso	<i>CU-4.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Secundario, esencial.</i>	
Referencias	<i>RD-2, RF-6.</i>	
Descripción	<i>El usuario puede crear una nueva base de datos vacía para almacenar la dirección URL de las imágenes junto con sus descriptores asociados.</i>	
Precondición	<i>No puede existir ninguna base de datos abierta.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para crear una nueva base de datos.</i>
	2	<i>El sistema comprueba que no exista ya una base de datos disponible.</i>
	3	<i>El sistema crea la base de datos vacía.</i>
Postcondición	<i>Se creará una nueva base de datos vacía.</i>	
Excepciones	Paso	Acción
	3a	<i>Ya existe una base de datos abierta.</i> <ul style="list-style-type: none"> • <i>El sistema avisa al usuario acerca de ello.</i>
Rendimiento		
Frecuencia esperada		
Importancia	<i>Vital.</i>	
Urgencia	<i>Alta.</i>	
Comentarios	<i>Este es el paso previo a crear los descriptores de las imágenes para poder almacenarlos.</i>	

Caso de uso	<i>Abrir una base de datos existente.</i>	
Nº de caso de uso	CU-5.	
Actores	<i>Usuario.</i>	
Tipo	<i>Primario, real.</i>	
Referencias	RD-2, RD-4, RD-5, RF-6, CU-4.	
Descripción	<i>El usuario podrá escoger un fichero de base de datos de su sistema mediante un cuadro de diálogo.</i>	
Precondición	<i>El archivo debe existir y tener una extensión asociada al almacenaje de una base de datos.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa el botón correspondiente para escoger un fichero de base de datos.</i>
	2	<i>El sistema comprueba que no exista ya ninguna base de datos abierta.</i>
	3	<i>El sistema le muestra un cuadro de diálogo para que el usuario escoja un archivo de su sistema.</i>
	4	<i>El usuario selecciona un fichero.</i>
	5	<i>El sistema comprueba que la extensión del fichero es válida.</i>
	6	<i>El sistema abre la base de datos cargando todos sus elementos.</i>
Postcondición	<i>La base de datos abierta estará disponible para su uso.</i>	
Excepciones	Paso	Acción
	3a	<i>Ya existe una base de datos disponible.</i> <ul style="list-style-type: none"> • <i>El sistema avisa al usuario de ello.</i>
	6a	<i>El fichero seleccionado no dispone de una extensión relacionada con una base de datos.</i> <ul style="list-style-type: none"> • <i>El sistema informa sobre el incidente.</i>
	6b	<i>El fichero que almacena una base de datos no ha podido abrirse.</i> <ul style="list-style-type: none"> • <i>El sistema informa acerca del resultado de la operación.</i>
Rendimiento		
Frecuencia esperada		
Importancia	<i>Alta.</i>	
Urgencia	<i>Media.</i>	
Comentarios		

Caso de uso	Añadir registros a una base de datos.	
Nº de caso de uso	CU-6.	
Actores	Usuario.	
Tipo	Primario, real.	
Referencias	RD-2, RD-3, RD-4, RD-5, RF-7, RF-9, RF-10, RF-11, RNF-1, CU-4, CU-5.	
Descripción	<i>El usuario podrá añadir uno o varios registros compuestos por la dirección URL de la imagen y su descriptor.</i>	
Precondición	<i>Debe haberse abierto o creado una base de datos.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para añadir nuevos registros a la base de datos.</i>
	2	<i>El sistema genera el descriptor de cada imagen abierta en la aplicación en función del tamaño asociado a este y a la clase de descriptores que la base de datos almacene.</i>
	3	<i>El sistema almacena cada uno de los descriptores creados en la base de datos junto con la URL de la imagen asociada a ellos.</i>
Postcondición	<i>Los nuevos registros se habrán añadido a la base de datos.</i>	
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	<i>Alta.</i>	
Urgencia	<i>Alta.</i>	
Comentarios	<i>Se deberá guardar la base de datos tras añadir los registros para que estos elementos sean almacenados en el fichero que contiene la base de datos.</i>	

Caso de uso	Guardar una base de datos.	
Nº de caso de uso	CU-7.	
Actores	Usuario.	
Tipo	Secundario, real.	
Referencias	RD-2, RD-4, RD-5, RF-6, CU-4, CU-5.	
Descripción	<i>El usuario podrá guardar la base de datos actual en un fichero que se genera en una ruta específica con un nombre predefinido.</i>	
Precondición	Debe existir una base de datos abierta.	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para guardar una base de datos.</i>
	2	<i>El sistema crea el fichero en el que se va a almacenar la base de datos.</i>
	3	<i>El sistema guarda en el fichero todos los elementos de la base de datos.</i>
Postcondición	<i>Se creará un nuevo fichero que contenga todos los registros de la base de datos actual.</i>	
Excepciones	Paso	Acción
	3a	<i>No se han podido almacenar los registros de la base de datos en el fichero.</i> <ul style="list-style-type: none"> • <i>El sistema informa sobre el error que ha ocurrido.</i>
Rendimiento		
Frecuencia esperada		
Importancia	Media.	
Urgencia	Baja.	
Comentarios		

Caso de uso	<i>Cerrar una base de datos.</i>	
Nº de caso de uso	<i>CU-8.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Secundario, real.</i>	
Referencias	<i>RD-2, RD-4, RD-5, RF-6, CU-4, CU-5</i>	
Descripción	<i>El usuario podrá cerrar la base de datos actual.</i>	
Precondición	<i>Debe existir una base de datos previamente creada o abierta.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para cerrar una base de datos.</i>
	2	<i>El sistema elimina el contenido de la estructura de datos de la aplicación que representa a la base de datos actual.</i>
Postcondición	<i>La base de datos deja de estar disponible.</i>	
Excepciones	Paso	Acción
	2a	<i>No existe una base de datos abierta.</i> <ul style="list-style-type: none">• <i>El sistema no realiza ninguna acción.</i>
Rendimiento		
Frecuencia esperada		
Importancia	<i>Media.</i>	
Urgencia	<i>Media.</i>	
Comentarios		

Caso de uso	<i>Seleccionar una imagen consulta.</i>	
Nº de caso de uso	<i>CU-9.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Primario, real.</i>	
Referencias	<i>RD-1, RF-1, RF-2, RF-4, CU-1.</i>	
Descripción	<i>El usuario elige una imagen consulta de entre todas las imágenes que haya abiertas en la aplicación.</i>	
Precondición	<i>La imagen consulta debe haber sido abierta previamente.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón correspondiente para seleccionar una imagen consulta.</i>
	2	<i>El sistema resalta el borde de la ventana cuya imagen ha sido seleccionada como nueva imagen consulta.</i>
	3	<i>El sistema actualiza la nueva imagen consulta y almacena una referencia a la ventana interna en la que se encuentra.</i>
Postcondición	<i>La imagen destacada como imagen consulta será la que se utilice para las siguientes comparaciones.</i>	
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	<i>Alta.</i>	
Urgencia	<i>Alta.</i>	
Comentarios		

Caso de uso	Seleccionar un comparador.	
Nº de caso de uso	CU-10.	
Actores	Usuario.	
Tipo	Primario, real	
Referencias	RF-5, RNF-7.	
Descripción	<i>El usuario podrá seleccionar un comparador de entre los disponibles estableciendo los valores de los campos correspondientes.</i>	
Precondición		
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón asociado a los parámetros relacionados con las consultas.</i>
	2	<i>El sistema le muestra un cuadro de diálogo con los campos y sus valores disponibles.</i>
	3	<i>El usuario pulsa "Aceptar" cuando esté satisfecho con los valores seleccionados.</i>
	4	<i>El sistema cierra el cuadro de diálogo y actualiza los nuevos valores de los campos.</i>
Postcondición		
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	<i>Si el usuario selecciona un comparador que puede aplicar un tipo de inclusión, el usuario podrá escoger entre simple o doble inclusión.</i>	

Caso de uso	Seleccionar la dimensión de los descriptores.	
Nº de caso de uso	CU-11.	
Actores	Usuario.	
Tipo	Primario, real.	
Referencias	RD-4, RD-5, RF-5	
Descripción	<i>El usuario podrá seleccionar una dimensión de entre las disponibles para componer la rejilla que dividirá a las imágenes y que, posteriormente, se utilizará para establecer el tamaño de sus descriptores.</i>	
Precondición		
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre el botón asociado a los parámetros de configuración de las consultas.</i>
	2	<i>El sistema le muestra una ventana de diálogo con todos los campos disponibles junto con sus valores permitidos.</i>
	3	<i>El usuario selecciona un determinado valor para establecer la dimensión que se aplicará de manera generalizada a todas las imágenes.</i>
	4	<i>El usuario pulsa en el botón "Aceptar" cuando esté conforme con los valores escogidos.</i>
	5	<i>El sistema cierra la ventana de diálogo y establece los nuevos valores.</i>
Postcondición		
Excepciones	Paso	Acción
	3a	<i>El usuario selecciona una dimensión para la imagen consulta y otra distinta para el resto de las imágenes.</i>
Rendimiento		
Frecuencia esperada		
Importancia	Alta.	
Urgencia	Alta.	
Comentarios		

Caso de uso	Realizar consultas en una base de datos.	
Nº de caso de uso	CU-12.	
Actores	Usuario.	
Tipo	Primario, real.	
Referencias	RF-8, RF-9, RF-10, RF-11, RNF-3, RNF-4, RNF-6, CU-1, CU-5, CU-6, CU-9, CU-10, CU-11.	
Descripción	<i>El usuario realizará una consulta en una base de datos para comparar el descriptor de la imagen consulta con los descriptores almacenados en la base de datos.</i>	
Precondición	<i>Debe haber establecido una imagen consulta previamente. Así mismo deberá disponer de una base de datos abierta.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario establece la imagen consulta.</i>
	2	<i>El usuario pulsa sobre el botón que comienza la consulta.</i>
	3	<i>El sistema busca y recupera el descriptor de la imagen consulta.</i>
	4	<i>El sistema obtiene el comparador activado en el momento.</i>
	5	<i>El sistema realiza la consulta comparando el descriptor de la imagen consulta con cada descriptor almacenado en la base de datos, mediante el comparador establecido.</i>
Postcondición	<i>Aparecerá una lista de imágenes ordenadas de mayor a menor grado de similitud.</i>	
Excepciones	Paso	Acción
	3a	<i>El sistema no ha podido encontrar el descriptor de la imagen consulta, y por ende, lo genera a partir de la dimensión del descriptor que se encuentre establecida y de la clase de descriptor que la base de datos almacena. Luego lo añade a la base de datos.</i>
Rendimiento		
Frecuencia esperada		
Importancia	Vital.	
Urgencia	Alta.	
Comentarios		

Caso de uso	Realizar consultas sin base de datos.	
Nº de caso de uso	CU-13.	
Actores	Usuario.	
Tipo	Primario, real.	
Referencias	RF-4, RF-9, RF-10, RF-11, RNF-3, RNF-4, RNF-6, CU-1, CU-3, CU-9, CU-10, CU-11.	
Descripción	<i>El usuario realizará una consulta comparando el descriptor de la imagen consulta con los descriptores de las imágenes abiertas en la aplicación.</i>	
Precondición	<i>Deben existir imágenes abiertas en la aplicación y haberse establecido una imagen consulta previamente.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario establece la imagen consulta.</i>
	2	<i>El usuario pulsa sobre el botón que comienza la consulta.</i>
	3	<i>El sistema genera los descriptores de cada una de las imágenes abiertas en la aplicación en función del tipo de descriptor seleccionado y de la dimensión establecida para este.</i>
	4	<i>El sistema obtiene el comparador establecido en el momento actual.</i>
	5	<i>El sistema compara el descriptor de la imagen consulta con el resto de descriptores calculados mediante el comparador establecido.</i>
Postcondición	<i>Aparecerá una lista de imágenes ordenadas de mayor a menor grado de similitud.</i>	
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	<i>Vital.</i>	
Urgencia	<i>Alta.</i>	
Comentarios		

Caso de uso	Realizar consultas en base a una etiqueta lingüística.	
Nº de caso de uso	CU-14.	
Actores	Usuario.	
Tipo	Primario, real.	
Referencias	RD-3, RF-13, RF-14, RNF-2, RNF-3, RNF-4, RNF-6, CU-4, CU-5, CU-6.	
Descripción	<i>El usuario podrá seleccionar una etiqueta lingüística entre las disponibles además de la posición o la combinación de dos posiciones en la que debe encontrarse. Luego se realizará la consulta en una base de datos.</i>	
Precondición	<i>Debe haber abierto una base de datos previamente.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa sobre la lista de las etiquetas lingüísticas y selecciona una de ellas.</i>
	2	<i>El usuario escoge una posición.</i>
	3	<i>El usuario pulsa sobre el botón que inicializa la consulta en base a una etiqueta.</i>
	4	<i>El sistema genera el descriptor en base al concepto lingüístico seleccionado.</i>
	5	<i>El sistema obtiene las regiones de cada imagen que van a ser consultadas en función de la posición establecida y de la dimensión de su descriptor.</i>
	6	<i>El sistema realiza la consulta comparando el nuevo descriptor generado con los almacenados en la base de datos, teniendo en cuenta la posición final.</i>
Postcondición	<i>Aparecerá una lista de imágenes ordenadas de mayor a menor grado de similitud.</i>	
Excepciones	Paso	Acción
	2a	<i>El usuario establece dos posiciones y la operación para combinarlas.</i>
	5a	<i>El sistema obtiene las regiones de cada imagen que van a ser consultadas en función de la dimensión de su descriptor y de la operación, que haya escogido el usuario, para combinar dos posiciones.</i>
Rendimiento		
Frecuencia esperada		
Importancia	Vital.	
Urgencia	Alta.	
Comentarios		

Caso de uso	<i>Mostrar las etiquetas de una imagen.</i>	
Nº de caso de uso	<i>CU-15.</i>	
Actores	<i>Usuario.</i>	
Tipo	<i>Secundario, real.</i>	
Referencias	<i>RD-1, RD-3, RD-4, RD-5, RF-9, RF-10, RF-11, RF-12, RNF-4.</i>	
Descripción	<i>El usuario podrá hacer doble click sobre una imagen para crear y mostrar su descriptor con las etiquetas lingüísticas asociadas a cada una de sus regiones.</i>	
Precondición	<i>La imagen debe estar abierta previamente.</i>	
Secuencia Normal	Paso	Acción
	1	<i>El usuario pulsa dos veces sobre la imagen.</i>
	2	<i>El sistema genera el descriptor con tantas etiquetas como indique la dimensión del descriptor que se haya establecido.</i>
	3	<i>El sistema muestra las etiquetas a través del panel de salida situado en la parte inferior de la aplicación.</i>
Postcondición		
Excepciones	Paso	Acción
Rendimiento		
Frecuencia esperada		
Importancia	<i>Baja.</i>	
Urgencia	<i>Baja.</i>	
Comentarios		
Importancia	<i>Baja.</i>	
Urgencia	<i>Baja.</i>	
Comentarios		

Capítulo 10

Análisis

En este capítulo en cuestión se explicará la manera en la que se ha abordado el desarrollo del prototipo ligado a este proyecto basándonos en los requerimientos detallados en capítulos anteriores. Para ello procedo a analizar cada una de las funciones de las que dispone mi sistema, además de las herramientas que se han usado así como las implementaciones propias que se han realizado.

En relación a los requerimientos de información explicados la primera necesidad que surge a partir de ellos es disponer de una clase que almacene las características de una imagen. Para ello disponemos, en casi todos los lenguajes de programación incluyendo aquel con el que se ha realizado este proyecto que es Java, clases ya definidas que podemos reutilizar.

Para la estructura de datos que representa a un descriptor utilizaremos una biblioteca denominada **JMR** en la que existen diversas clases que representan distintos tipos de descriptores, incluyendo el que es capaz de generar las etiquetas lingüísticas. Para ello he hecho uso de un clasificador ya entrenado y de su respectiva clase para inicializar el objeto que lo representa.

Así mismo para trabajar de forma local con las imágenes he utilizado una de las clases contenidas en esta biblioteca para dividir la imagen en un conjunto de regiones con el objetivo de generar un descriptor por cada una de ellas. Este conjunto de descriptores puede almacenarse en una base de datos cuya clase también se encuentra en la **JMR**. No obstante se ha debido implementar un método cuya tarea consiste en realizar una consulta dado el descriptor de una imagen consulta o un descriptor creado directamente mediante una etiqueta lingüística. Para realizar esto último he implementado una clase propia.

Adicionalmente, con el objetivo de que el funcionamiento del sistema no dependiese únicamente de la existencia de una base de datos, se ha implementado un segundo método que realiza la consulta en base a las imágenes que haya abiertas en la aplicación. Para ello hará uso, a su vez, de un conjunto de métodos propios que se han implementado para crear sus respectivos descriptores en función de la configuración activa del sistema.

Continuando con las funcionalidades de las que dispone el sistema todas las métricas explicadas a lo largo de este documento han sido implementadas en clases individuales. Si bien este conjunto de comparadores suman nueve clases, se han implementado cuatro clases más. La primera de ellas se corresponde con una superclase global a todas las métricas. En ella se almacenan las propiedades y métodos comunes independientemente del funcionamiento asociado a cada comparador.

Existe una segunda superclase, que hereda de la anterior, la cual engloba a todas las métricas que no tienen en cuenta la posición de los objetos. En ella se almacenan los métodos asociados a establecer el tipo de inclusión que se va a aplicar.

A su vez cada una de las dos familias que comparten esta filosofía dispone de una superclase distinta. Aquellas métricas que forman parejas de regiones aplicando la restricción consistente en que ninguno de sus miembros haya sido emparejado con otra celda, heredarán de una superclase que almacena los métodos comunes. El mismo suceso le ocurre al conjunto de comparadores que no aplica esta restricción. La principal razón por la que se han añadido estas dos superclases reside en no repetir código incluyendo todos los métodos que se usan en cada una de las clases. Para ello se ha realizado un proceso de abstracción con el propósito de que solo en una clase aparezcan los métodos comunes.

Con el objetivo de permitir que se realicen consultas en base a un concepto lingüístico en lugar de una imagen, se ha llevado a cabo una ampliación de una de las métricas para realizar una búsqueda de un determinado objeto. Con el propósito de implementar esta funcionalidad se han incluido los métodos necesarios para calcular las regiones en las que debe buscarse el elemento para cada imagen. La razón de haberlo implementado de este modo reside en la posibilidad de almacenar descriptores de distintos tamaños en la base de datos. Así mismo, en este procedimiento también se han implementado las dos operaciones explicadas en capítulos anteriores que son capaces de combinar las dos posiciones especificadas de dos formas distintas. En la primera de ellas se podrá buscar el objeto en las regiones de la imagen que estén contenidas en ambas posiciones, mientras que con la segunda operación se permitirá buscar el elemento en ambas localizaciones concretadas.

Para terminar con este tipo de requisitos se ha diseñado una interfaz de usuario con la que se pudiese probar todas las funcionalidades del sistema. Con el objetivo de reutilizar código he adaptado todas las clases relacionadas con los diferentes tipos de ventanas que existen en el proyecto de un sistema multimedia que desarrollé el año pasado para la asignatura de *Sistemas Multimedia*. Del mismo modo también he recuperado las funcionalidades de ciertos botones comunes a ambos sistemas, como ha sido el caso del botón asociado a abrir imágenes. Siguiendo con la parte que se ha desarrollado para la interfaz de este sistema existe un botón también relacionado con las imágenes que es capaz de cerrar todas las ventanas abiertas de la aplicación. Esta función es muy útil cuando queremos despejar el escritorio de la aplicación tras realizar diversas actividades que han causado la apertura de muchas ventanas internas.

En relación a la selección de un descriptor se ha añadido un menú desplegable en el cual aparecen todos los descriptores disponibles para que el usuario escoja aquel que deseé. No obstante, por defecto está seleccionado aquel que genera las etiquetas lingüísticas. Cabe destacar que si esto sucede aparecerá una nueva sección en la barra de herramientas superior para ajustar la métrica interna que tiene este descriptor. Sin embargo, esto no es necesario a priori puesto que también se han establecido valores por defecto. Así mismo se ha configurado un rango relacionado con el umbral del clasificador para seleccionar uno de los valores disponibles, aunque, de nuevo, también hay uno preestablecido.

En el segundo panel disponible en la barra de herramientas podemos encontrar tres tipos de botones relacionados con la acción de ejecutar una consulta en base a una imagen. Para ello existe un *checkbox* que destaca la imagen consulta que hemos seleccionado, un botón que abre una ventana de diálogo donde se pueden cambiar los parámetros de la consulta y un segundo botón con el cual iniciamos la consulta. Un tercer panel vinculado a la consulta, pero mediante el concepto lingüístico en vez de la imagen, contiene una serie de elementos tales como un selector en donde se despliegan las etiquetas de los objetos que la CNN puede reconocer. Así como un primer desplegable para seleccionar la primera posición y dos alternativas para realizar la combinación entre la posición ya escogida y una segunda a establecer. En función de la primera localización y de la operación marcada se mostrarán en el segundo desplegable aquellas

posiciones que sean compatibles. Por último en este panel también podremos encontrar un único botón con el cual podremos iniciar la consulta una vez hayamos especificado nuestros términos de búsqueda.

Para terminar con el cumplimiento de estos requisitos existe un último panel asociado a las distintas operaciones que se pueden realizar con una base de datos. Para cada una de ellas existe su botón correspondiente que lleva a cabo la acción. Las principales operaciones consisten en: crear una nueva base de datos vacía o abrir una existente mediante un dialogo que permitirá al usuario escoger un fichero que contenga una base de datos. También se podrá añadir uno o varios registros a la base de datos que esté actualmente abierta, delegando en otros métodos las acciones vinculadas a crear los descriptores oportunos de las imágenes que haya abiertas en la aplicación. Así mismo se podrá guardar la base de datos en un fichero y por último, se podrá cerrar la base de datos actual.

Con el fin de finalizar este capítulo cabe destacar que los requisitos no funcionales anteriormente descritos se cumplirán, de manera indirecta, mediante la toma de decisiones que ha causado la introducción de todos los elementos explicados con anterioridad.

Capítulo 11

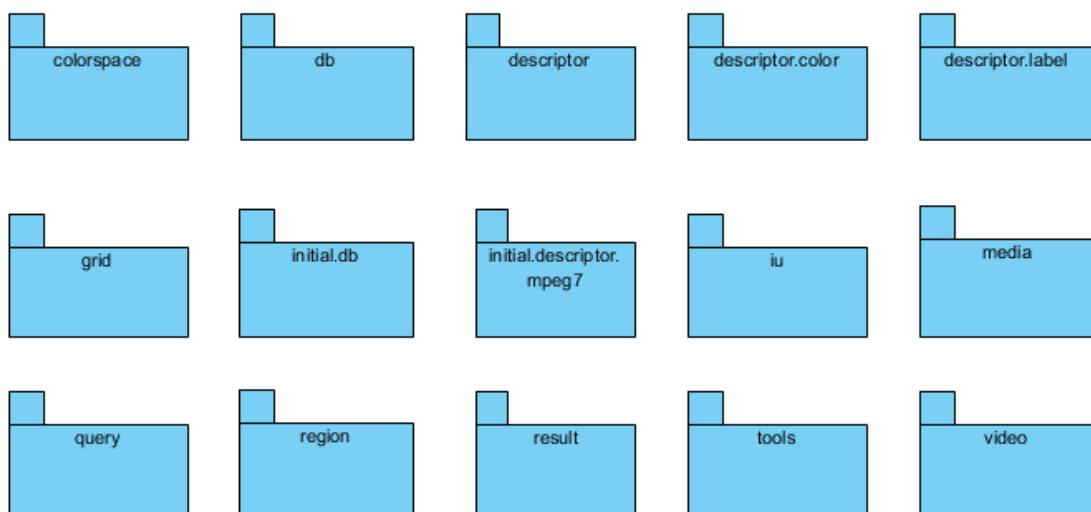
Diseño

A continuación detallaremos en este capítulo todos los aspectos relacionados con el diseño realizado para desarrollar el sistema prototipo. Para ello explicaremos la estructura de los paquetes cuyas clases han sido implementadas además de los paquetes situados en la biblioteca *JMR* de los cuales hemos utilizado diversas clases.

10.1. Java Multimedia Retrieval (JMR)

Esta librería está implementada con el objetivo de trabajar con cualquier tipo de archivo multimedia tales como imágenes, vídeo, audio, entre otros. Sus principales características es que se encuentra escrita en lenguaje Java y es de código abierto [16]. Dicha librería está desarrollada por el profesor de la Universidad de Granada Jesús Chamorro Martínez.

Con el propósito de ilustrar las funcionalidades que se han utilizado procedentes de esta librería y las que he desarrollado, a continuación se presenta el siguiente diagrama de paquetes que incluye y, posteriormente, detallaremos las clases de las que se ha hecho uso en mi software.



De entre todos ellos los más relevantes para mi proyecto han sido los cuatro siguientes:

1. **Paquete db.** En él se almacena la clase que es capaz de crear la base de datos así como añadirle sus registros y guardarla en un fichero. Si bien también incluye un método para realizar consultas en ella, no hemos podido usarlo debido a incompatibilidades de las estructuras utilizadas para la consulta en base a una etiqueta en vez de una imagen.
2. **Paquete descriptor.** En él se encuentra la interfaz *Comparator* la cual ha sido crucial para desarrollar todas las métricas. La razón de ser se fundamenta en que esta interfaz contiene el método que será implementado en cada una de las clases de las métricas de manera distinta puesto que es el encargado de comparar dos descriptores.

A su vez, en este paquete también se sitúa la clase *GriddedDescriptor*, la cual contiene los métodos y propiedades suficientes como para dividir a una imagen en un conjunto de regiones cuadradas. Esta clase representa el descriptor con el que se trabaja en mi sistema, el cual a su vez, contiene un conjunto de descriptores de un tipo determinado. Si bien es cierto que ha sido generalmente utilizado para crear los descriptores en función de una imagen, no he podido usarlo para realizar este mismo procedimiento en base a una etiqueta lingüística.

3. **Paquetes color y label.** Del primer paquete se han utilizado aquellas clases cuyos descriptores contienen diversas propiedades gráficas de una imagen. Si bien su utilidad ha sido bastante escasa, se ha hecho uso de sus clases, principalmente, para realizar las consultas que motivan este proyecto. Todo lo contrario ha ocurrido con las clases del segundo paquete, las cuales han permitido que mi proyecto sea capaz de cargar un clasificador para generar las etiquetas asociadas a una imagen.
4. **Paquete iu.** En él se encuentra una determinada clase que contiene el panel en el que se muestra la lista resultante de imágenes tras realizar una consulta. Es por ello por lo que este componente ha sido añadido a la ventana encargada de mostrar este panel.

10.2. Novedades con respecto a la JMR

A pesar de haber utilizado la biblioteca *JMR* para incorporar algunas de sus funciones también se han desarrollado en este proyecto nuevas funcionalidades que no existen en esta librería. Todas ellas están directamente relacionadas con las diversas formas de realizar una consulta, ya sea en base a una imagen o en función de una etiqueta lingüística. Me refiero, en particular, al conjunto de comparadores y a una clase que representa un subtipo de descriptor.

10.2.1. Métricas

Comenzamos a describir la estructura de paquetes y clases que se ha llevado a cabo con el propósito de implementar los diversos comparadores. El primer paquete que podemos observar en la jerarquía de este sistema engloba a los sub-paquetes que contienen cada familia de comparadores. La razón principal de esta implementación es que en él se incluye la clase *Comparators* en la que se almacena un atributo que representa el comparador interno de la clase *LabelDescriptor* que esté activo. Así mismo también dispone del correspondiente método consultor y modificador para obtener y modificar el valor de dicho atributo en función de los valores seleccionados en la interfaz, respectivamente. Tanto la propiedad anterior como sus métodos son comunes a todos los comparadores existentes y, por ello, se ha realizado un

proceso de abstracción para que todas las métricas puedan aplicar este comparador. Por lo tanto todas las clases venideras heredan de esta superclase general.

Este primer paquete contiene, a su vez, dos paquetes más. El primero de ellos denominado *samePosition* es aquel que contiene las clases correspondientes a los tres comparadores que tienen en cuenta la posición de los objetos que aparecen en la imagen consulta.

El segundo se corresponde con el paquete denominado *differentPosition*, en el cual se encuentra una segunda superclase abstracta que almacena un atributo, además de su método consultor y modificador, que establece el tipo de inclusión que se va a aplicar. Como esta peculiaridad es común a todos los comparadores que no consideran la posición de los objetos al realizar consultas, esta será la superclase de la que hereden las clases de los siguientes paquetes. En particular existen dos más. Uno de ellos, denominado *noDuplicates*, que contiene a las tres métricas que integran la restricción, explicada en capítulos anteriores, relacionada con la formación de parejas de regiones sin que ninguno de sus miembros ya esté asociado con otra región. En este paquete, además de los tres tipos de comparadores, también se sitúa una cuarta superclase abstracta que los engloba. Su principal propósito es el de agrupar los métodos comunes a las métricas de esta familia de modo que todas ellas puedan utilizarlos sin la necesidad de incluir dichos métodos en cada clase repitiendo código.

El otro paquete denominado *withDuplicates* contiene a los tres tipos de comparadores que no aplican la restricción mencionada anteriormente. Tal y como ocurría en el caso anterior, en este paquete también se ha implementado una cuarta superclase abstracta de la que hereden las tres métricas asociadas a esta familia. Su objetivo es similar a la de la clase anterior, es decir, en ella se almacenan los métodos comunes a las tres métricas de modo que todas puedan utilizarlos sin repetir código.

Una de las métricas, en particular, ha sufrido una ampliación con respecto a las demás con el propósito de llevar a la práctica la idea de realizar una consulta en función del concepto asociado a un objeto, sin utilizar una imagen consulta, y además restringiendo su posición a una especificada. Se trata del comparador que busca un elemento determinado en cualquier posición y que devuelve una distancia cero cuando encuentra dicho término lingüístico en, al menos, una región perteneciente a una imagen. Para realizar esta tarea se han añadido, en primer lugar, dos métodos que aplican el operador *AND* o el *OR*, dependiendo del que el usuario haya escogido, siempre y cuando se hayan establecido dos posiciones. Posteriormente existe un tercer método que calcula las regiones que se deben consultar en busca de la etiqueta seleccionada, teniendo en cuenta el tamaño del descriptor de la imagen además de la posición final calculada. Por último a través de un cuarto método se establecen los índices de las celdas a consultar antes de comenzar a realizar la comparación entre los descriptores.

A su vez, tal y como comenté en el capítulo anterior también he tenido que desarrollar mi propio método *query*, el cual recorre todos los descriptores almacenados en la base de datos para llevar a cabo las consultas. Cabe destacar que se ha diseñado de forma general para que sea capaz de desarrollarlas tanto en base a una imagen como en función de una etiqueta lingüística. No obstante, con el objetivo de otorgarle una mayor flexibilidad al sistema de manera que no dependa de tener una base de datos para realizar una consulta, se ha implementado una segunda versión. Su principal propósito es la de permitir que el usuario también sea capaz de ejecutar consultas entre las imágenes abiertas en la aplicación. De este modo el procedimiento varía notablemente puesto que en este caso hay que calcular los descriptores para cada una de las imágenes abiertas en cada una de las consultas que se realicen.

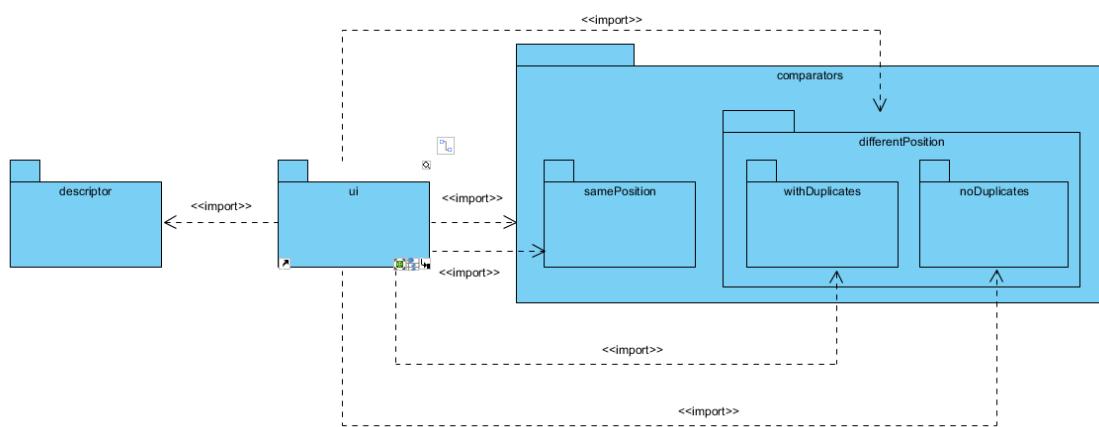
10.2.2. *LabelGridDescriptor*

Esta clase representa un nuevo descriptor implementado con el objetivo de desarrollar el

tipo de consultas basadas únicamente en una etiqueta lingüística. Para ello aplicamos la misma filosofía existente en la clase *GriddedDescriptor*, es decir, este descriptor contiene a su vez un conjunto de descriptores en los que se almacena una etiqueta lingüística. Debido a las diversas características que comparten ambos tipos de descriptores, la clase *LabelGriddedDescriptor* hereda las propiedades de *GriddedDescriptor*. No obstante, cuenta con su vector de descriptores de conceptos lingüísticos, además de un conjunto de métodos que han sido redefinidos para adaptarse al funcionamiento del nuevo descriptor.

10.3. Diagrama de paquetes

El software está compuesto por diversos paquetes en los que cada uno agrupa un determinado número de clases en función de la relación que las vincula entre sí. En la siguiente captura se puede apreciar el diagrama que representa la jerarquía de paquetes.



Tal y como podemos observar el paquete que contiene las clases que componen la interfaz de la aplicación, denominado *ui*, es el que está conectado con todos los demás. La razón de ello es que, en función de los parámetros que estén activos en la aplicación, se realizarán las correspondientes llamadas a los métodos de unas clases o de otras para establecer los valores seleccionados.

El paquete que se coloca en la parte izquierda del diagrama es el que almacena los descriptores que se han desarrollado, en este caso contiene la clase denominada *LabelGriddedDescriptor*, la cual ha sido explicada con anterioridad.

En relación a las clases que implementan las diversas métricas, tal y como podemos observar, se encuentran englobadas en el paquete *comparators*. A su vez existen dos paquetes más, el situado a la izquierda denominado *samePosition* es aquel que contiene las clases de los tres comparadores que consideran la posición de los objetos en la imagen consulta a la hora de realizar comparaciones entre imágenes.

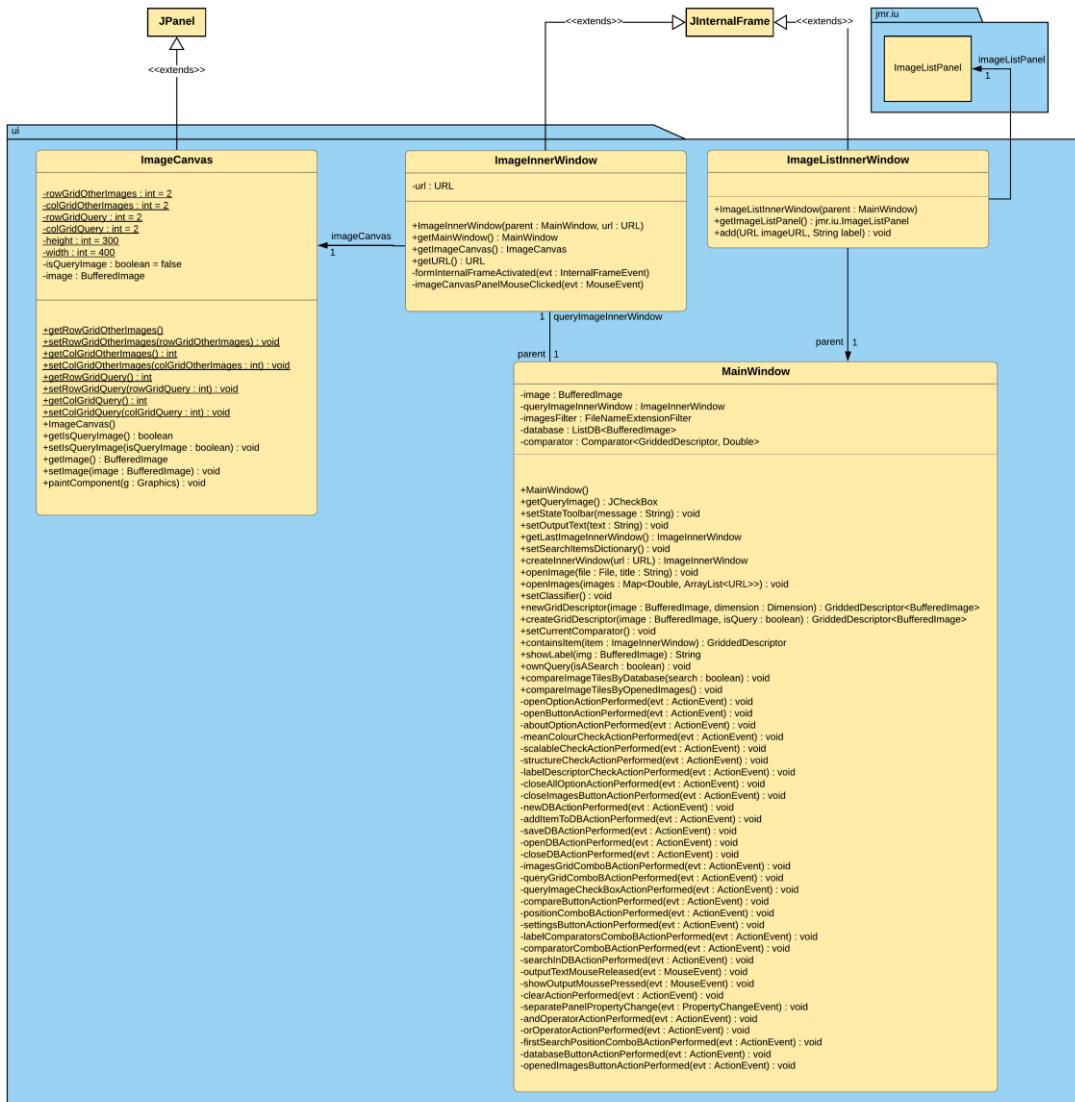
Aquellas métricas que aplican la filosofía opuesta se encuentran englobadas en el paquete denominado *differentPosition*. A su vez este paquete contiene otros dos más dependiendo de si los comparadores almacenados en él aplican o no la restricción explicada en capítulos anteriores. Con ella se consigue que dadas dos imágenes en las que en ambas aparezca el mismo elemento pero representen escenas muy distintas no se las clasifique como parecidas. Las métricas que integran esta restricción se encuentran en el paquete *noDuplicates* mientras que las restantes que no la incluyen se alojan en el paquete llamado *withDuplicates*.

10.4. Diagrama de clases

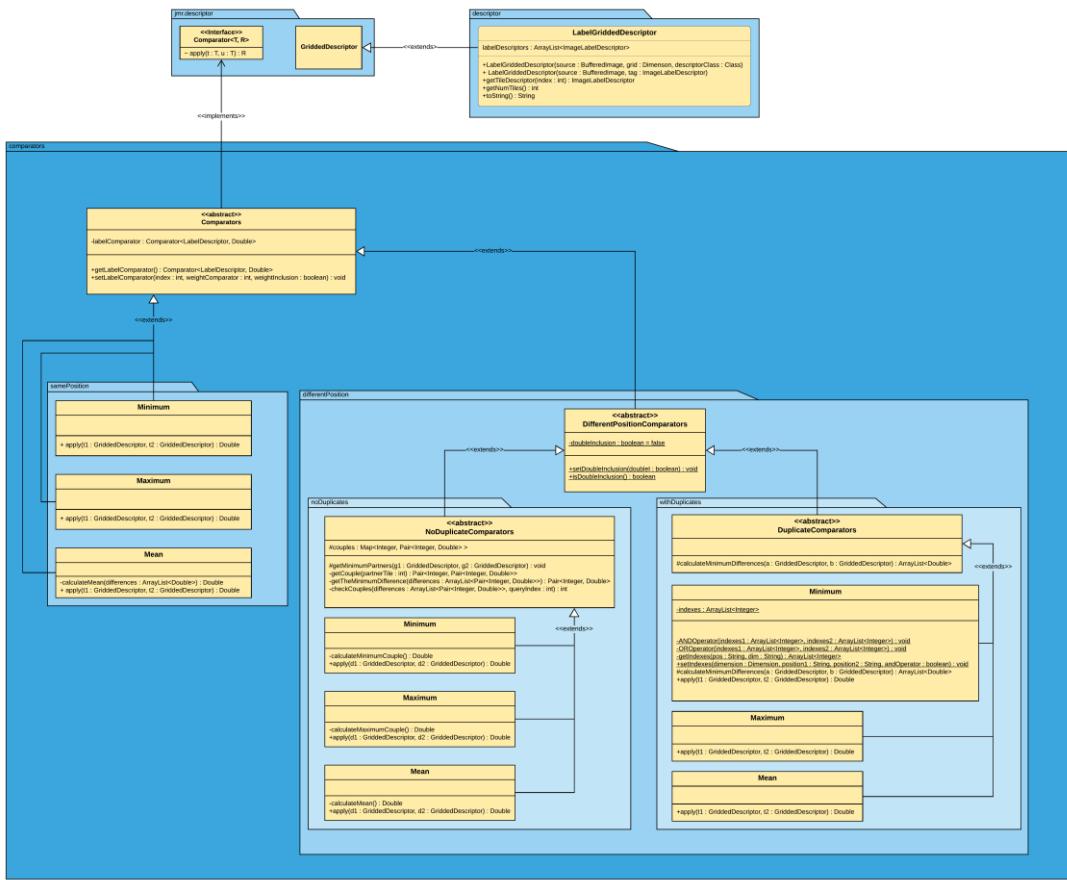
A continuación se procede a detallar la jerarquía del sistema explicando, para ello, la estructura de las clases implementadas incluyendo tanto sus atributos, métodos así como las relaciones que las vinculan con otras clases.

En esta jerarquía de clases se pueden distinguir tres categorías principales en función de la utilidad que cada una de las clases representa. La primera de ellas está relacionada con las clases asociadas a la implementación de los distintos comparadores, la segunda representa las características del descriptor que ha sido desarrollado, y por último, explicaremos el diagrama asociado a las clases que conforman la interfaz de usuario. Es por ello por lo que se van a adjuntar dos diagramas de clases distintos, uno que represente la estructura desarrollada para la interfaz y el otro que refleje la jerarquía de los comparadores implementados junto con el nuevo descriptor.

10.4.1. Diagrama de la interfaz



10.4.2. Diagrama de los comparadores y descriptor



Capítulo 12

Implementación

La implementación de este sistema CBIR se ha realizado en el lenguaje de programación Java y para ello se ha utilizado el entorno de desarrollo *NetBeans IDE 8.1* puesto que contiene un conjunto de herramientas que facilitan la tarea de desarrollo, como por ejemplo, un depurador para seguir paso a paso la ejecución del código. Así mismo también incluye un conjunto de componentes gráficos muy intuitivos y sencillos tanto de utilizar como de personalizar con el propósito de componer la interfaz del sistema prototipo.

La documentación acerca de las clases, sus atributos y métodos se encuentra en la API ya que ha sido desarrollada mediante *Javadoc*. La razón de ello se fundamenta, principalmente, en la facilidad para redactar la funcionalidad de cada uno de los componentes. Aunque por motivos de espacio no se ha incluido en este documento, a continuación adjunto una captura que muestra gráficamente el resultado de la documentación asociada a este proyecto.

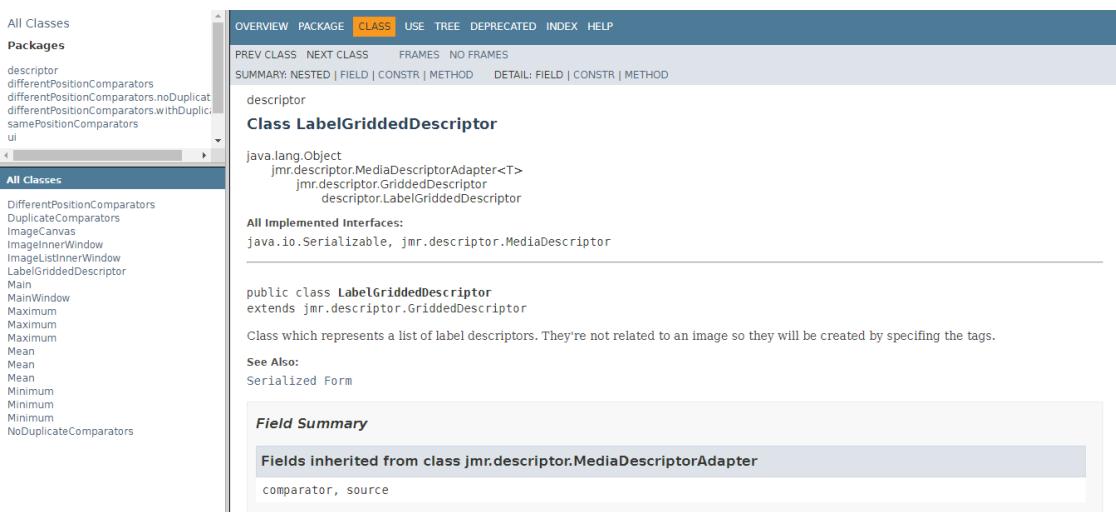


Figura 71: Ejemplo de una de las páginas del *javadoc* del proyecto.

Tal y como se puede comprobar ha sido desarrollada en inglés por continuar dicha tendencia procedente de la librería JMR. Así mismo, utilizando este idioma, incrementamos el número de

posibilidades de que terceras personas que estén situadas en cualquier parte del mundo puedan comprender el funcionamiento de cada uno de los componentes de las clases.

Tanto el código fuente como el *javadoc* además de los ficheros asociados a la CNN, las bases de datos utilizadas y la base de datos de fotografías que he realizado personalmente se encuentran en el siguiente repositorio: <https://github.com/lidiasm/TFGProject.git>

Capítulo 13

Manual de usuario

Pese a que se ha tratado de confeccionar una interfaz sencilla e intuitiva, a consecuencia de las diversas funcionalidades que el sistema integra, en este capítulo se redactará un manual de usuario para explicar las distintas operaciones asociadas a los botones que componen la interfaz.

Con el propósito de agrupar estos componentes en base a las funciones que desempeñan el sistema cuenta con hasta cinco paneles diferentes. A ellos hay que sumar los tres menús desplegables que podemos observar en la parte superior de la aplicación.

Antes de comenzar a explicar cada una de las categorías de componentes a continuación se adjunta una imagen de la visión general del sistema.

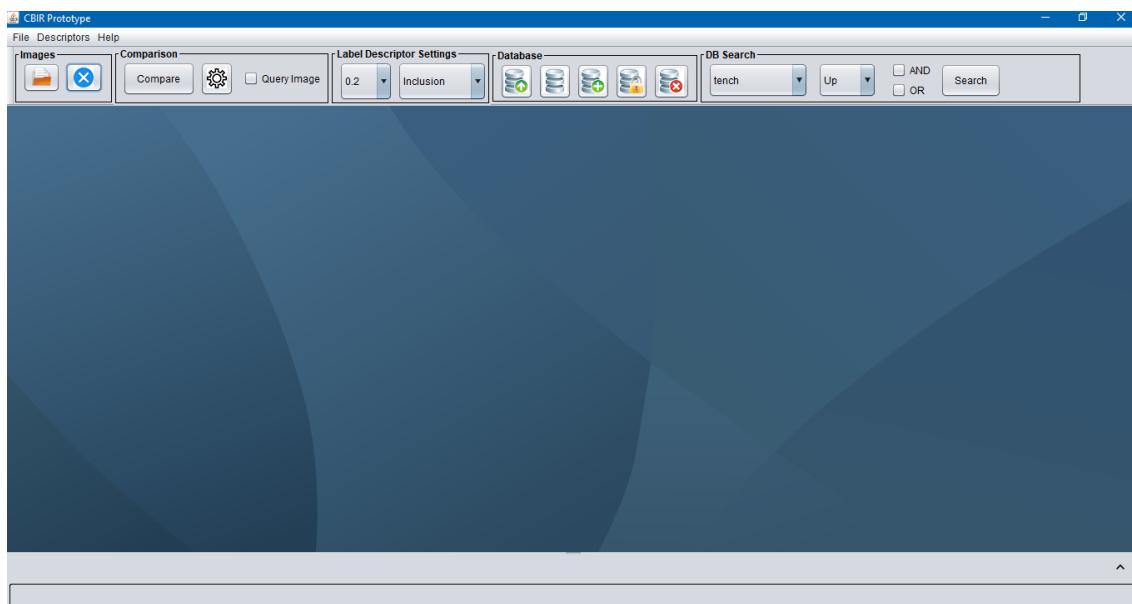


Figura 72: Imagen de la vista principal de la aplicación.

Tal y como se puede observar, es en la parte superior de la ventana de la aplicación donde se encuentran todos los componentes que nos permiten ejecutar las funcionalidades disponibles en el sistema. Así mismo podemos comprobar que dependiendo de la funcionalidad que desempeñen todos estos componentes están divididos en cinco paneles diferenciados. Es por ello por lo que procedo a explicarlos realizando la misma clasificación.

13.1. Primer bloque: imágenes.

En este primer panel denominado *Images* se encuentran, tal y como podemos observar en la siguiente captura, dos botones principales. Ambos realizan operaciones en las que las imágenes son las principales protagonistas. Con el primero de ellos, situado a la izquierda, el usuario podrá seleccionar una o varias imágenes de su sistema, mediante un cuadro de diálogo, con el propósito de abrirlos en la aplicación. Cada una de ellas se encontrará en una ventana interna independiente. En relación al segundo botón podemos determinar que su principal objetivo reside en cerrar todas las ventanas internas que se encuentren abiertas en la aplicación.

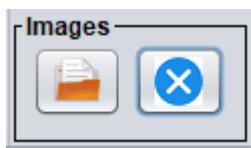


Figura 73: Primer panel asociado a las operaciones con imágenes.

13.2. Segundo bloque: consultas.

En el siguiente panel se encuentran todos aquellos elementos que están relacionados con las funcionalidades que llevan a cabo consultas en función de una imagen consulta. Tal y como se puede observar en la siguiente figura, en este panel se encuentran hasta un total de tres elementos, que procedemos a explicar.



Figura 74: Segundo panel relacionado con los parámetros de las consultas.

- La casilla *check* establecerá como imagen consulta aquella fotografía contenida en la ventana interna actualmente seleccionada. Para distinguirla del resto añadirá un borde naranja alrededor de ella.
- El primero de los botones es el responsable de iniciar una consulta a partir de los parámetros de consulta establecidos, tales como el descriptor y el comparador actualmente seleccionados.
- El segundo botón abre una ventana de diálogo en la cual se muestran los parámetros relacionados con las consultas además de sus posibles valores para

que el usuario los modifique a su gusto. La apariencia de dicha ventana se visualiza en la siguiente captura.

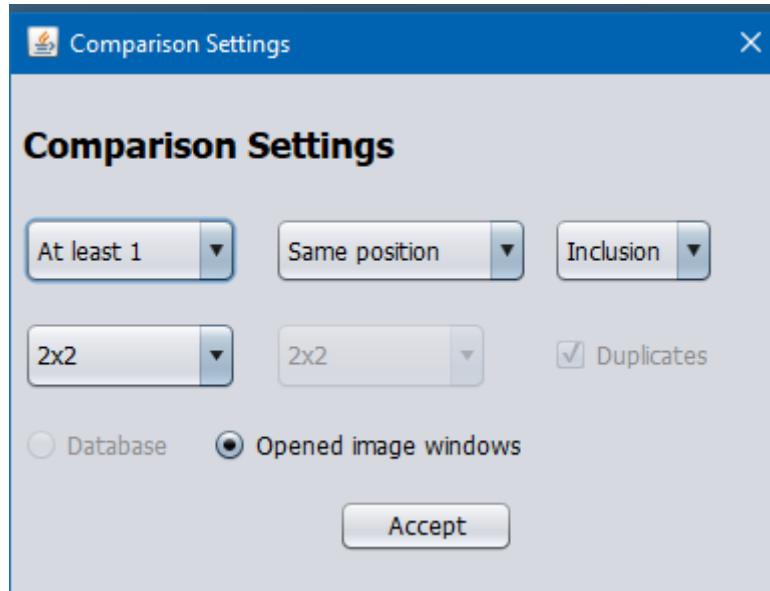


Figura 75: Vista de la ventana de diálogo que contiene los parámetros de las consultas

A continuación se procederá a explicar la funcionalidad que se esconde tras cada uno de los componentes que alberga esta ventana. En la parte superior, tal y como podemos ver, se encuentran los tres campos referentes a los distintos tipos de comparadores existentes.

- Con el primero podemos seleccionar cuántos elementos, de los que aparecen en una imagen consulta, se deben encontrar en las otras imágenes para clasificarlas como parecidas con respecto a esta. Sus posibles valores son: al menos un elemento, la mayoría de ellos o que aparezcan en general.
- El segundo campo nos permite seleccionar si el comparador debe considerar la posición de dichos objetos o no.
- Con el tercero podremos escoger el tipo de inclusión que la métrica actual debe aplicar para realizar una o dos veces la consulta. Este parámetro solo se podrá establecer si el tipo de comparador seleccionado no tiene en cuenta la posición de los objetos que aparecen en la imagen consulta.
- Un cuarto campo que se corresponde con una casilla de *check* se sitúa en la segunda fila a la derecha, denominado *Duplicates*. Si dicha casilla está seleccionada la restricción explicada en capítulos anteriores, que consigue que dadas dos imágenes que contienen el mismo elemento pero representan escenas distintas no se las clasifique como parecidas, no se aplicará. Si por el contrario esta casilla no está marcada se activará uno de los comparadores que aplica esta restricción.

En esa misma fila, al lado de la casilla de *check* anterior, hay dos listas con las dimensiones disponibles que se le pueden aplicar a una imagen.

- La primera de ellas establece el tamaño de los descriptores de todas las imágenes en general excepto si existe una imagen consulta establecida. En ese caso la dimensión

escogida se aplicará al descriptor asociado a dicha imagen.

- Por lo tanto, la segunda lista de dimensiones que se encuentra a su lado servirá para establecer el tamaño de los descriptores asociados al resto de imágenes, bajo la circunstancia descrita anteriormente.

Por último, en la tercera fila se encuentran dos botones que decidirán si la consulta se realiza en una base de datos o entre las imágenes abiertas en la aplicación. Por defecto, nada más abrir la aplicación el botón correspondiente a la segunda opción comentada es el que se encuentra marcado. La razón de ello es que al inicio de la aplicación no es posible que se encuentre una base de datos abierta. No obstante, cuando el usuario crea o abre una es cuando este botón se desmarca automáticamente para que el otro que está a su lado se seleccione. Este último indica que las siguientes consultas se realizarán en la base de datos actualmente disponible. Sin embargo el usuario podrá seleccionar uno de los dos botones, y así establecer, dónde se van a realizar las consultas futuras.

Para finalizar podemos observar que existe un botón en la parte inferior de la ventana cuya principal responsabilidad reside en cerrar esta ventana de diálogo.

13.3. Tercer bloque: métricas internas a *LabelDescriptor*.

El tercer panel situado en la aplicación alberga, como primer campo, aquel capaz de mostrar una lista con los umbrales disponibles para aplicarle al clasificador. Este parámetro establecerá el nivel de rigurosidad que tendrá el clasificador cuando proceda a identificar los elementos existentes en una imagen. Por defecto está establecido a 0.2. Tras él se encuentran los campos destinados a mostrar los parámetros asociados a los comparadores internos que tiene la clase *LabelDescriptor*. El primero de ellos activa uno de los comparadores disponibles, y si se trata del basado en pesos, entonces aparecerá un tercer campo para establecer la operación que se realiza con los susodichos. Así mismo también aparecerá un cuarto campo que permitirá al usuario escoger el tipo de inclusión que este comparador interno aplicará. Con el propósito de representar lo anteriormente explicado se adjunta la siguiente captura en la que se muestran los cuatro campos existentes.



Figura 76: Tercer panel relacionado con el comparador interno de *LabelDescriptor*.

13.4. Cuarto bloque: base de datos.

En el cuarto panel se encuentran los botones asociados a las distintas operaciones que se pueden realizar con una base de datos.

- El primer botón es el responsable de abrir un fichero de base de datos para cargarla en la aplicación. Para ello se abrirá un cuadro de diálogo permitiendo al usuario seleccionar dicho archivo. Esta operación se realizará si no existe ninguna otra base de datos abierta.

- El segundo botón está pensado para crear una base de datos vacía, siempre y cuando se cumpla el mismo requisito anterior.
- Con el tercer botón se pueden agregar nuevos registros a la base de datos actual. Con el fin de realizar esta operación se creará el descriptor asociado a cada imagen abierta en la aplicación y se añadirá a la base de datos junto con la dirección URL de dicha fotografía. Para ello se tendrá en cuenta el descriptor seleccionado y su dimensión establecida.
- El cuarto botón acciona la operación correspondiente a guardar la base de datos actual en un fichero. Para ello se creará un archivo en una ruta determinada con un nombre predefinido y, posteriormente, se almacenarán todos los registros de la base de datos actual.
- Por último, con el quinto botón se podrá cerrar la base de datos actual, dejando esta de estar disponible para su uso.

En la siguiente figura se muestra el panel con los botones explicados anteriormente.

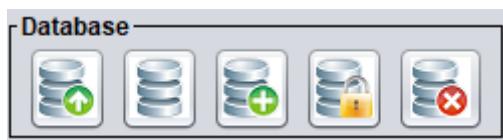


Figura 77: Cuarto panel relacionado con las operaciones con bases de datos.

13.5. Quinto bloque: consulta con una etiqueta.

En el quinto panel encontraremos todos aquellos elementos relacionados con la funcionalidad cuyo objetivo es realizar consultas en base a una etiqueta seleccionada y restringiendo su posición a una especificada o a una combinación de dos. El primer campo situado en el lateral izquierdo se corresponde con la lista de 1000 términos lingüísticos que la CNN es capaz de reconocer en una imagen.

En función de la necesidad del usuario este podrá especificar una sola posición, por lo que este quinto panel se visualizaría de la forma en la que aparece en la siguiente figura. En él, además del listado de etiquetas, solo disponemos de una lista para escoger la única posición en la que deberá encontrarse el objeto a buscar.



Figura 78: Quinto panel relacionado con las consultas en base a etiquetas con una sola posición.

No obstante, si el usuario desea realizar una combinación de dos posiciones distintas podrá marcar una de las dos casillas *check* que se encuentran a continuación. Dependiendo de la que seleccione la operación a aplicar sobre las dos posiciones será distinta. Con la primera casilla tendrá la posibilidad de restringir la posición del objeto a las regiones comunes a ambas localizaciones especificadas. Por ejemplo, podría buscar un elemento arriba y a la derecha, lo

que lo situaría en la esquina superior derecha de la imagen.

Sin embargo, si marca la segunda casilla entonces el elemento buscado deberá encontrarse en una posición o en otra. De esta forma podrá realizar búsquedas en las que el elemento buscado deba estar, por ejemplo, en la parte central de la imagen o a la derecha.

En función de la casilla seleccionada aparecerá un segundo desplegable para escoger la segunda posición, tal y como podemos comprobar en la siguiente figura. Este panel albergará aquellas posiciones que sean compatibles en función de la primera posición escogida y de la operación que se vaya a aplicar. De este modo, si por ejemplo ha escogido el primer operador como se muestra en la figura y como primera posición *arriba*, entonces podrá limitar la posición del objeto a *arriba y a la izquierda o arriba y a la derecha*.

Al final de este panel siempre se situará el botón que inicia este tipo de consulta obteniendo, para ello, los valores activos en el momento en el que ha sido pulsado.

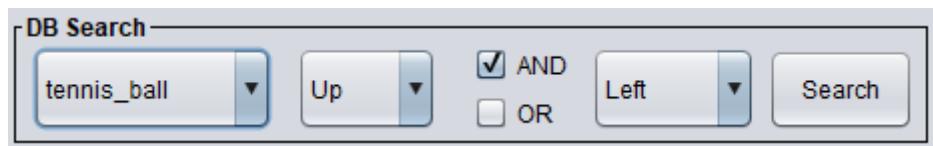


Figura 79: Quinto panel relacionado con las consultas en base a etiquetas con una combinación de dos posiciones.

13.6. Sexto bloque: descriptores.

En último lugar procederé a explicar el componente que despliega la lista de descriptores cuyos dos propósitos se basan en comprobar qué descriptor está seleccionado, y seguidamente, se le brinda la oportunidad al usuario de seleccionar otro que deseé. Cabe destacar que siempre habrá un descriptor seleccionado y que el descriptor marcado por defecto nada más iniciar la aplicación será aquel que está relacionado con las etiquetas lingüísticas. A su vez, siempre que haya una base de datos abierta este panel será inutilizado con el objetivo de que el usuario no mezcle descriptores de distintas clases en una misma base de datos. La razón de ello es que solo se pueden realizar consultas entre aquellos descriptores que pertenezcan a la misma clase. No obstante, tras cerrar la base de datos actual este panel volverá a habilitarse para su modificación.

A continuación podemos ver que la lista desplegable de descriptores se encuentra en la barra de herramientas superior, tal y como se muestra en la siguiente figura.

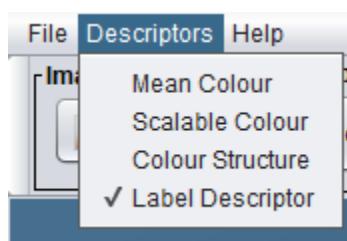


Figura 80: Lista desplegable de descriptores.

Tal y como podemos comprobar este sistema puede hacer uso de cuatro descriptores distintos. Los tres primeros extraen diversas propiedades gráficas de las imágenes, en particular, el primero

de ello obtiene su color medio y los dos siguientes calculan diferentes histogramas, también relacionados con la misma propiedad, el color. El último de ellos es el que genera las etiquetas lingüísticas a través del reconocimiento de objetos que puede realizar la CNN. Este es el principal protagonista en este proyecto, y por ende, aunque todo lo implementado también puede usarse con los otros tres descriptores, las funcionalidades desarrolladas están especialmente diseñadas para utilizar este último descriptor.

13.7. Ejemplos de uso.

Si bien todos los principales componentes de la interfaz han sido detallados, a continuación procedemos a ejemplificar los tres tipos de consultas que pueden realizarse en este sistema. El principal objetivo que perseguimos consiste en clarificar los pasos a seguir para que cualquier usuario sea capaz de reproducirlos y así hacer uso de estas tres principales funcionalidades.

13.7.1. Consulta en función de una imagen sin base de datos.

Comenzamos desarrollando la metodología propia del primer tipo de consulta que realizará una comparación entre una imagen consulta que seleccionemos y el resto de imágenes abiertas en la aplicación. Los pasos que se deben llevar a cabo son los siguientes:

1. En primer lugar se deberá abrir un conjunto de imágenes con el que realizar la consulta. Para ello pulsaremos sobre el primer botón situado en el panel *Images* para que podamos seleccionar aquellas fotografías de nuestro sistema que deseemos abrir mediante un cuadro de diálogo. Su apariencia se puede comprobar en la siguiente figura:

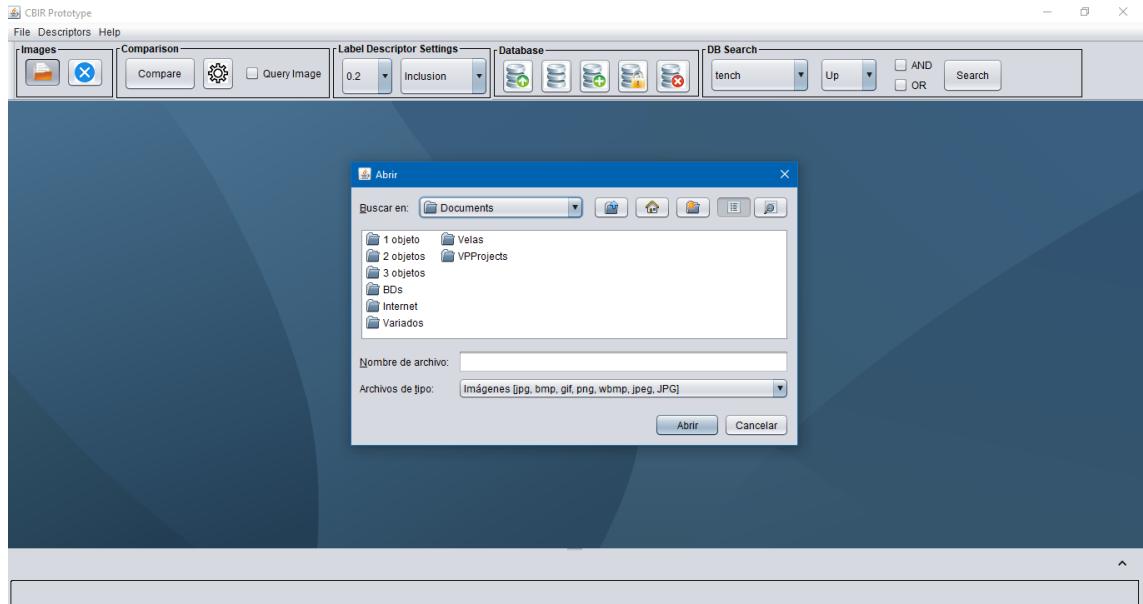


Figura 81: Cuadro de diálogo para seleccionar las imágenes a abrir.

2. Tras haber seleccionado las imágenes deseadas pulsaremos sobre el botón *Abrir* situado en el cuadro de diálogo anterior para que el sistema comience a abrir las y colocarlas, cada una de ellas, en una ventana interna individual. La siguiente figura muestra el aspecto de la aplicación tras haber abierto un conjunto de imágenes.

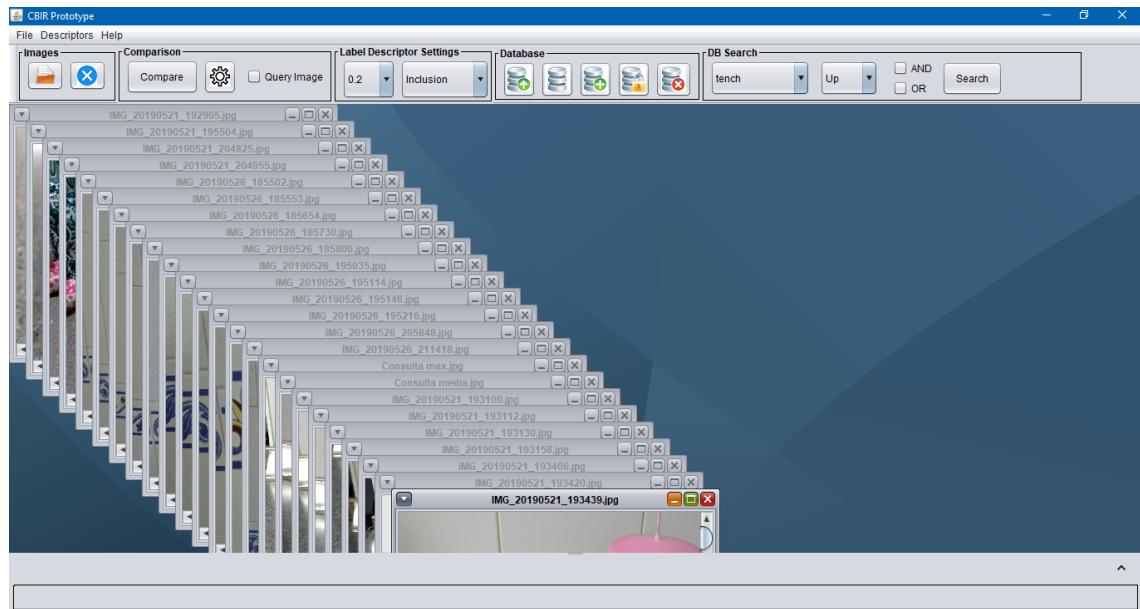


Figura 82: Vista de las imágenes abiertas en la aplicación.

3. A continuación seleccionaremos una de ellas y marcaremos la casilla *Query Image* para establecerla como imagen consulta y así poder realizar la consulta. En este ejemplo dicha imagen contiene un sacapuntas azul situado en la parte inferior de esta. Tanto ella como el resto de imágenes estarán divididas en seis regiones distribuidas en dos filas y tres columnas. Para establecer estos valores basta con pulsar sobre el segundo ícono del panel *Comparison* para que aparezca la ventana de diálogo, explicada y ejemplificada anteriormente, en la que se pueden modificar todos los parámetros relacionados con la consulta.

En base a dicha elección todos los descriptores almacenarán seis elementos en los cuales, utilizando el descriptor *LabelDescriptor*, se almacenarán las etiquetas correspondientes al contenido de cada región. Para utilizarlo no hace falta seleccionarlo puesto que es el descriptor que está marcado por defecto al iniciar la aplicación.

El comparador que he utilizado es aquel que considera que una foto es similar a la imagen consulta si encuentra en ambas, al menos, la misma etiqueta en la misma posición. De nuevo para establecer este tipo de comparador se deberá acudir a la ventana de diálogo que muestra los distintos parámetros asociados a una consulta.

En la siguiente figura se podrá observar la ventana mencionada anteriormente con los valores establecidos para el tamaño de los descriptores y el comparador utilizado. A su derecha se mostrará la fotografía del sacapuntas seleccionada como imagen consulta destacada en naranja. Y debajo de ellas se encuentra la lista de imágenes resultante tras haber realizado la consulta.

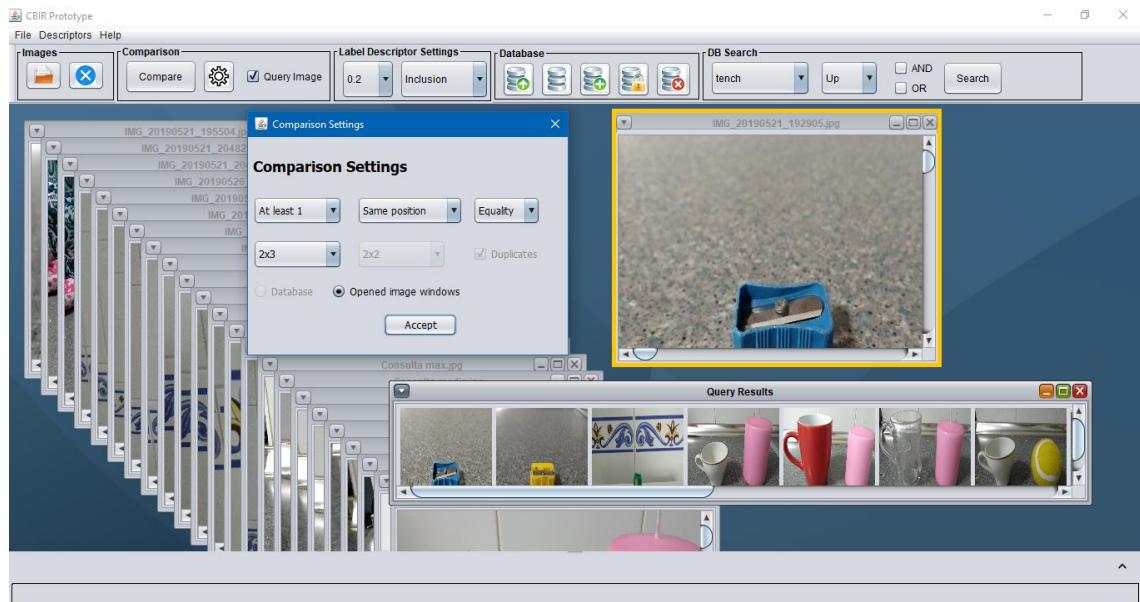


Figura 83: Vista del resultado del primer ejemplo de consulta.

Dicha lista muestra las imágenes ordenadas de mayor a menor grado de similitud. Las tres imágenes que cumplen los criterios anteriormente especificados son aquellas que ocupan las tres primeras posiciones. La razón de ello reside en que en todas aparece un sacapuntas, de diversas características, fotografiado de distintas formas y en diferentes lugares, situados en la parte inferior de estas. A partir de ellas el resto de imágenes que no cumplen estas restricciones serán mostradas en el orden en el que han sido consultadas.

13.7.2. Consulta en base a una imagen utilizando una base de datos.

En este tipo de consulta existen dos maneras de proceder. Si bien la más sencilla es utilizando una base de datos ya existente, como lo vamos a hacer en el siguiente ejemplo, también se puede crear una nueva. Para ello deberemos seguir los siguientes pasos:

1. En primer lugar abrimos un conjunto de imágenes en la aplicación pulsando, para ello, el primer botón situado en el panel *Images*.
2. Seleccionamos el tipo de descriptor que deseamos utilizar para generar los descriptores de las imágenes que serán almacenados en la base de datos.
3. Pulsamos sobre el segundo botón del panel *Database* para crear una nueva base de datos vacía.
4. En la ventana asociada a los parámetros de configuración seleccionamos la dimensión de la rejilla que será aplicada a las imágenes, y por ende el tamaño de sus descriptores, mediante el primer desplegable numérico. Si bien es cierto que la base de datos permite almacenar descriptores de distintos tamaños, se deberán abrir tantos grupos de imágenes como distintas dimensiones queremos que existan. Así podemos abrir un primer grupo de fotos con el propósito de dividirlas en seis regiones, y posteriormente, abrir un segundo grupo de imágenes para dividirlas en cuatro regiones.

5. Por último pulsamos sobre el tercer botón del panel *Database* para comenzar el proceso automático consistente en crear el descriptor de cada una de las imágenes abiertas.
6. En caso de que queramos guardar la base de datos en un fichero solo tendremos que pulsar el cuarto botón, situado en el mismo panel, una vez haya terminado el proceso del paso anterior.

No obstante, tal y como hemos comentado anteriormente, en esta consulta procedemos a utilizar la misma base de datos con la que se han realizado los ejemplos mostrados en capítulos anteriores. Para abrirla solo tenemos que pulsar sobre el primer botón del panel *Database* y a continuación seleccionar un fichero de base de datos de nuestro sistema, mediante el cuadro de diálogo que se puede observar en la siguiente figura.

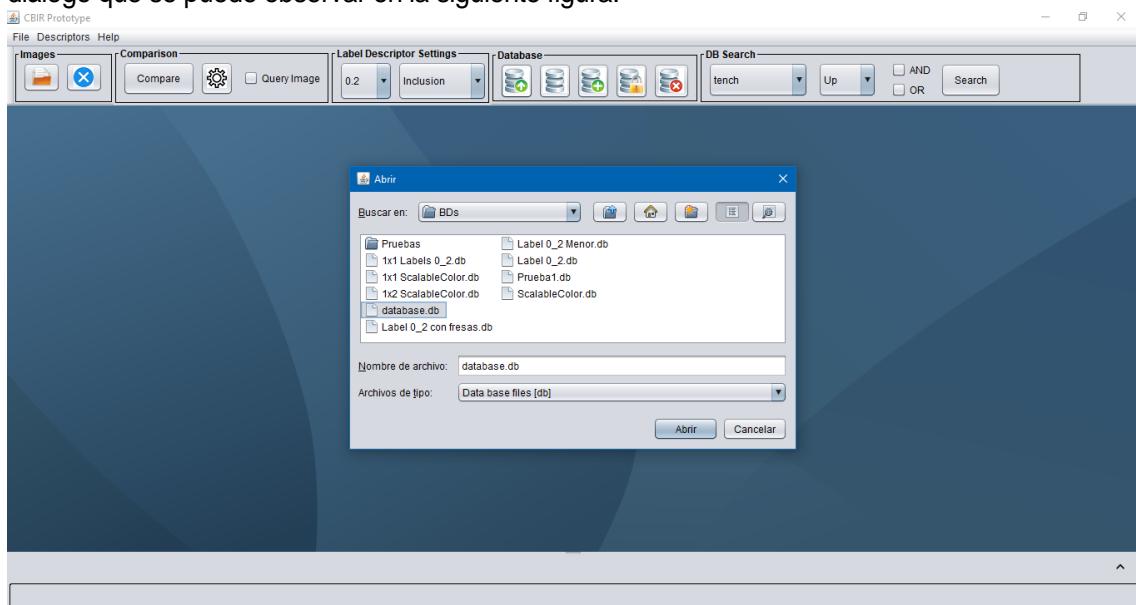


Figura 84: Vista del cuadro de diálogo para abrir un fichero de base de datos.

Siempre y cuando hayamos seleccionado un fichero de base de datos, el sistema procederá a cargar todos sus registros en la correspondiente estructura de datos de la aplicación. Si no es el caso se mostrará un mensaje de error avisando sobre el formato erróneo del archivo.

Una vez hemos cargado la base de datos deberemos abrir cualquier fotografía y seleccionarla como imagen consulta a través de la casilla *Query Image* situada en el panel *Comparison*. Por último configuraremos los parámetros asociados a la consulta que vamos a realizar y pulsamos sobre el botón *Compare* situado en el mismo panel. En este ejemplo, en particular, se ha utilizado como imagen consulta aquella en la que aparecen tres mecheros, dividida en tres regiones verticales y utilizando el comparador que busca la mayoría de estos objetos en la misma posición, tal y como se muestra en la siguiente captura. Además en ella también podremos visualizar en la parte inferior de sendas ventanas la lista de imágenes resultante tras haber realizado la consulta.

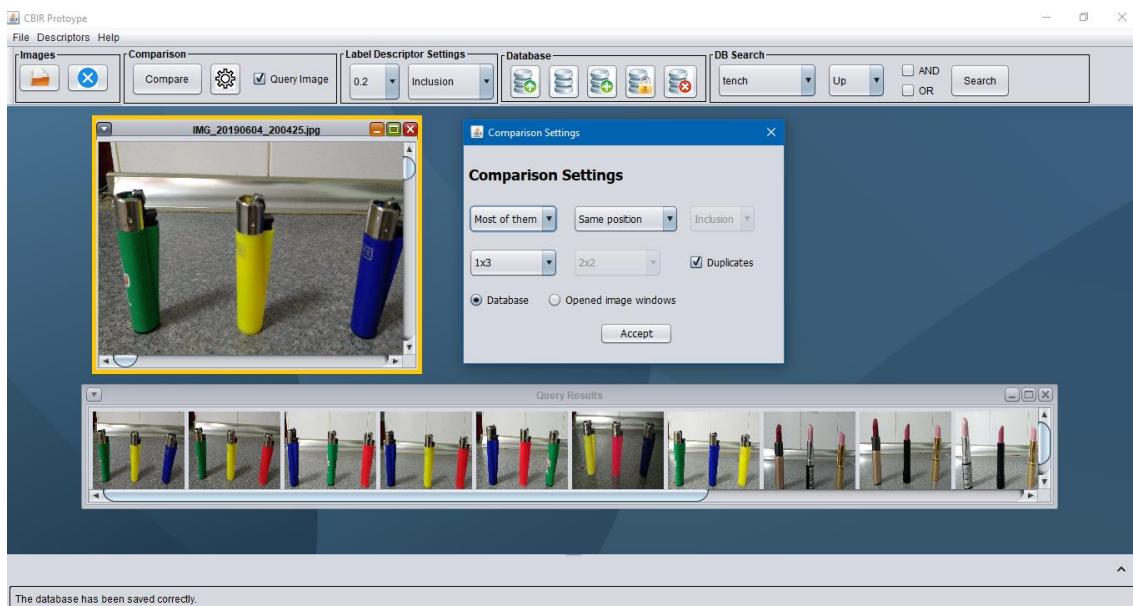


Figura 85: Vista del resultado del segundo ejemplo de consulta.

13.7.3. Consulta en base a una etiqueta.

En este tercer tipo de consulta se necesita, de nuevo, disponer de una base de datos. Así podemos utilizar una ya existente o crear una nueva siguiendo los pasos descritos anteriormente. En este último ejemplo optaremos por la primera opción, en la que seguiremos utilizando la misma base de datos. A diferencia de las dos consultas anteriores, en este caso solo se podrá trabajar con el descriptor que genera etiquetas lingüísticas. La razón de ello es que estamos utilizando como elemento consulta un concepto lingüístico y por tanto, para realizar una búsqueda de este objeto, es necesario que los descriptores de las imágenes también estén formados por etiquetas.

Los pasos a realizar para llevar a cabo este tipo de consulta se describen a continuación:

1. En primer lugar debemos escoger el objeto que deseamos buscar seleccionando su correspondiente etiqueta. Para ello desplegaremos la lista de los 1000 conceptos lingüísticos, situada en el panel *DB Search*, y elegimos uno de ellos. Cabe destacar que si el concepto escogido no se encuentra en ninguna de las imágenes cuyos descriptores se almacenan en la base de datos, la lista de imágenes resultante será mostrada en función del orden de consulta. En este ejemplo hemos seleccionado el término *cup* representando al objeto taza.
2. A continuación podemos escoger entre establecer una sola posición en la que el objeto buscado deberá situarse o especificar dos posiciones y la forma de combinarlas. En ambos casos deberemos escoger una de las posiciones disponibles en el primer desplegable que se encuentra al lado de la anterior lista de conceptos, en el mismo panel *DB Search*. Si nos ajustamos a la primera opción solo nos quedaría pulsar el botón *Search* para iniciar la consulta. Un ejemplo ilustrativo de este tipo de búsqueda se puede observar en la siguiente captura. En él, tal y como hemos comentado en el paso anterior, hemos seleccionado la etiqueta *cup* para buscar tazas situadas en la parte superior de la imagen.

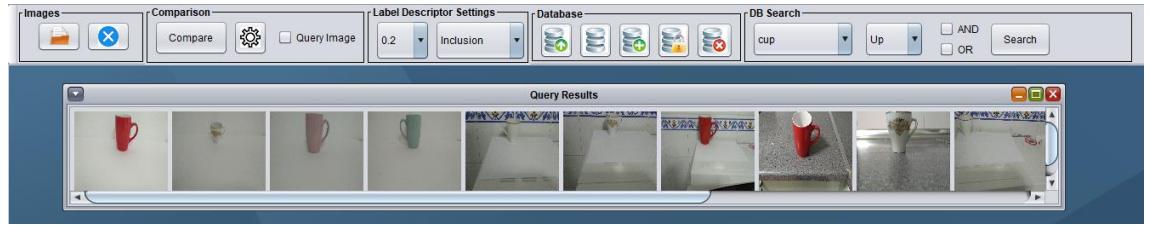


Figura 86: Vista del resultado del tercer ejemplo de consulta basado en etiquetas con una posición.

Si por el contrario deseamos utilizar dos posiciones entonces, tras haber seleccionado la primera, deberemos escoger uno de los dos operadores disponibles. Si marcamos la casilla *AND* estaremos combinando las dos posiciones para resultar en una sola. De esta forma podremos realizar consultas en las que un objeto se encuentre, por ejemplo, abajo a la derecha. Si optamos por seleccionar la casilla *OR* entonces se realizará una búsqueda en la que el objeto se encuentre en la primera o en la segunda posición. En este siguiente ejemplo hemos optado por el primer operador para buscar tazas que se encuentren abajo a la derecha.

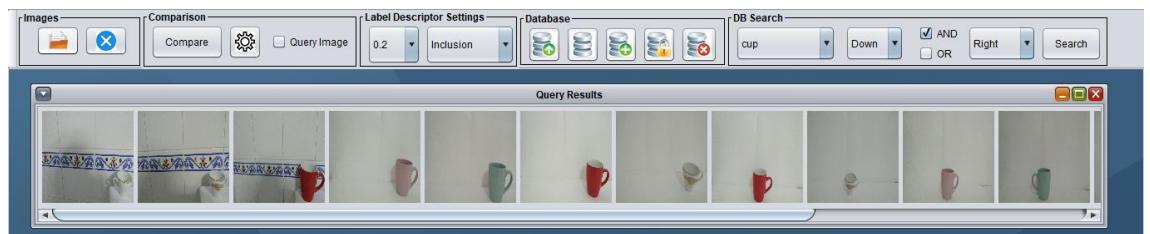


Figura 87: Vista del resultado del cuarto ejemplo de consulta basado en etiquetas con dos posiciones.

Tal y como podemos comprobar las imágenes situadas en las primeras posiciones son aquellas que cumplen con los criterios especificados. Posteriormente el resto de fotografías comienzan a aparecer en el orden en el que han sido consultadas.

Capítulo 14

Conclusiones y futuras investigaciones.

En este capítulo explicaremos las conclusiones finales a las que hemos podido llegar gracias al desarrollo de este proyecto. Por una parte debemos destacar la parte asociada al uso de las redes neuronales como herramienta para identificar los objetos de una imagen. En este ámbito podemos afirmar que su incorporación a los CBIR supone un gran avance. La razón de ello se basa en que, como hemos ido comprobando a lo largo de este documento, aquellos sistemas que realizan sus consultas en base a las propiedades gráficas de las imágenes no tienen en cuenta los objetos que aparecen. Pero además tampoco se consideran los diversos modelos existentes para la gran mayoría de objetos, en los que cada uno tiene diferentes propiedades visuales tales como el color o la textura. Así mismo este tipo de sistemas, a la hora de extraer las cualidades gráficas de una imagen, están influenciados por el fondo en la que esta se había tomado.

No obstante todas estas desventajas asociadas a los sistemas actuales de recuperación de imágenes no aparecen en el prototipo implementado gracias al nuevo procedimiento en el que se extraen las etiquetas correspondientes a los objetos de la imagen en lugar de sus características visuales.

Otro importante aspecto a destacar ha consistido en generar los conceptos lingüísticos asociados a una imagen de manera local. Así posibilitamos, tal y como hemos podido comprobar en bastantes experimentos, el hecho de que la CNN pueda ser capaz de reconocer varios objetos en la misma imagen que de otro modo no hubiese sido posible. Este hecho ya se demostró al comienzo de este proyecto cuando le mostrábamos una imagen con dos objetos y la CNN solo podía reconocer un único elemento generando las etiquetas de manera global. En el prototipo desarrollado, sin embargo, se aplica el reconocimiento de elementos llevado a cabo por la CNN en cada una de las regiones en las que se ha dividido la imagen.

Así mismo hemos desarrollado un conjunto de métricas que posibilitan obtener distintos resultados en función de las necesidades del usuario. Tal y como hemos podido comprobar en capítulos anteriores cada una de ellas tiene un comportamiento diferente, si bien todas son útiles en tanto en cuanto se deseen realizar unos tipos de consultas u otros. De este modo proporcionamos un amplio abanico de maneras en las que se puede medir el grado de similitud

entre dos imágenes.

Por último destacaremos el importante avance con respecto a las búsquedas de un objeto. Y es que tal y como hemos podido observar, los motores de búsqueda de Google aún no consideran la posición especificada a la hora de buscar un elemento. Es por ello por lo que en este proyecto se ha querido innovar e implementar una funcionalidad que no fuese tan común. De hecho la hemos ampliado hasta permitir dos tipos de combinaciones diferentes para integrar dos posiciones en una sola o para considerarlas a ambas a la hora de consultar por el elemento seleccionado.

En relación a las **futuras vías de trabajo** podemos concluir que en todos los aspectos se pueden realizar mejoras con respecto a lo implementado en este prototipo. Con respecto a la división de una imagen se puede integrar la posibilidad de que el usuario pueda seleccionar manualmente las regiones en las que le interesa dividir la imagen, en lugar de hacerlo de forma automática utilizando regiones cuadriculadas, tal y como se ha aplicado en este proyecto. De esta forma se le brinda la oportunidad al usuario de seleccionar qué partes de la imagen le interesa identificar mediante la CNN para posteriormente realizar una consulta.

Así mismo el último ámbito relacionado con la búsqueda de un elemento situado en una posición o en una combinación de dos también se puede ampliar. Para ello, por ejemplo, se podría añadir un segundo objeto a la búsqueda de manera que se pudiesen hacer consultas del siguiente tipo: “una taza a la derecha de una pelota”. De este modo el nivel de complejidad de la búsqueda aumentaría a la vez que lo hace su versatilidad.

También se podría apostar por buscar diversos objetos en lugar de uno teniendo en cuenta una posición para cada uno de los elementos. De ese modo podríamos realizar consultas en las que obtuviésemos un conjunto de imágenes en las que en unas apareciese el primer elemento en una posición, y otras que tuviesen el segundo objeto en una localización distinta a la del primero.

Bibliografía

1. David Kriesel. A Brief Introduction to Neural Networks. 2005
http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-1col-dkrieselcom.pdf
2. Pedro Isasi Vifiuela, Inés M. Galván León. Redes de Neuronas Artificiales, Un enfoque práctico. <https://www.dropbox.com/s/6rvbpt08hiztgve/274482888-Redes-Neuronales-Libro.pdf?dl=0>
3. Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando de Jesús. Neural Network Desing. 2014. <http://hagan.okstate.edu/NNDesign.pdf>
4. Simon Haykin. Neural Networks and Learning Machines. 2009.
<http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>
5. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning, 2016. Part 2, Chapter 9. <http://www.deeplearningbook.org/contents/convnets.html>
6. Christopher M. Bishop. Pattern Recognition and Machine Learning. 2006.
<http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf>
7. Priya Dwivedi. Understanding and Coding a ResNet in Keras.
<https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33>
8. Sara Pérez Álvarez. Universidad Complutense de Madrid. Aproximación al estudio de los sistemas de recuperación de imágenes “CBIR” desde el ámbito de la documentación.
<https://revistas.ucm.es/index.php/DCIN/article/download/DCIN0606110301A/19165>
9. John Eakins, Margaret Graham. University of Northumbria (Newcastle). Content-based Image Retrieval.
http://www.leeds.ac.uk/educol/documents/00001240.htm#_Toc442192675
10. Shutterstock Inc. ¿Qué son las fotos de stock?
<https://www.shutterstock.com/es/support/article/Qu%C3%A9-son-las-fotos-de-stock>

11. Santosh Bharti, Prof. Lalit Wadhwa. International Journal of Advanced Engineering and Global Technology. Content Based Image Retrieval Components. <http://ijaegt.com/wp-content/uploads/2013/12/IJAEGT-309156-SNT-303-308.pdf>
12. John Eakins. University of Northumbria (Newcastle). Automatic image content retrieval – are we getting anywhere?.
<https://pdfs.semanticscholar.org/5f4c/520ed514cf08355f1a3879fb9deeb694b45e.pdf>
13. Dave Marshall. QBIC (Query by Image Content). Query by example using color histograms. <https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node518.html>
14. Tinku Acharya, Ajoy K. Ray. Image Processing: Principles and Applications. 2005 <http://www.cs.ukzn.ac.za/~sviriri/Books/Image-Processing/book4.pdf>
15. What is a Gantt Chart? <https://www.gantt.com/>
16. Jesús Chamorro Martínez. Java Multimedia Retrieval.
<https://github.com/jesuschamorro/JMR>