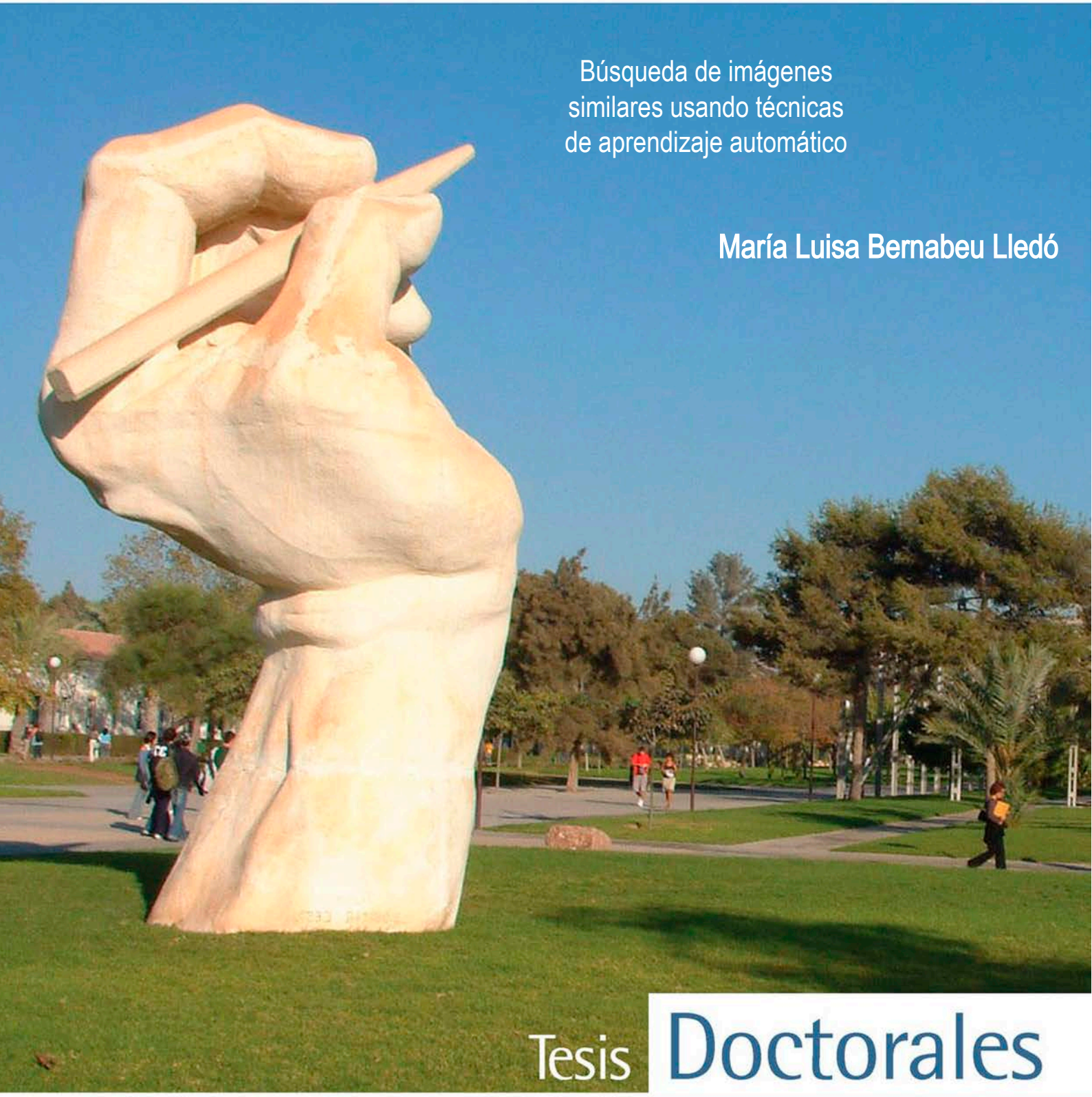
Universitat d'Alacant
Universidad de Alicante

Búsqueda de imágenes similares usando técnicas de aprendizaje automático

María Luisa Bernabeu Lledó



Tesis Doctorales
UNIVERSIDAD de ALICANTE

Unitat de Digitalització UA
Unidad de Digitalización UA

Universitat d'Alacant
Universidad de Alicante

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS

BÚSQUEDA DE IMÁGENES SIMILARES USANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

María Luisa Bernabeu Lledó

DISERTACIÓN PRESENTADA PARA ASPIRAR AL GRADO DE DOCTOR POR LA UNIVERSIDAD DE ALICANTE (DOCTOR OF PHILOSOPHY) DOCTORADO EN INFORMÁTICA

Directores

Dr. Antonio Javier Gallego Sánchez

Dr. Antonio Pertusa Ibáñez

ALICANTE, 2022

Si consigo ver más lejos
es porque he conseguido auparme
a hombros de gigantes

Isaac Newton.

Agradecimientos

A todas las personas que confiaron en mí y me apoyaron durante este largo y apasionante trayecto y, de alguna forma, hicieron posible que este trabajo se realice con éxito. En especial a mis directores Antonio Pertusa y Antonio Javier Gallego por su inestimable ayuda y buenos consejos, poder observar de cerca su trabajo y su gran profesionalidad ha sido muy motivador.

También quiero agradecer el soporte proporcionado al Grupo de Reconocimiento de Formas e Inteligencia Artificial (GRFIA) del Departamento de Lenguajes y Sistemas Informáticos (DLSI) de la Universidad de Alicante.

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1

Introducción

Las imágenes son representaciones visuales de objetos reales o imaginarios. Desde la prehistoria, el ser humano las ha utilizado como medio de comunicación gráfica, siendo un elemento fundamental del arte y muchas de sus disciplinas como la pintura, la fotografía, el cine o el diseño. Los mecanismos de percepción de la imagen han sido, y siguen siendo, ampliamente estudiados, ya que se trata de procesos complejos que dependen de múltiples factores como nuestra formación o experiencia, entre otros. Desde siempre el ser humano se ha esforzado por comprender estos mecanismos y, de hecho, en los años recientes se ha realizado un gran esfuerzo por simularlo en el campo de la visión por computador.

En la década de los 90 surgieron los primeros sistemas CBIR (*Content-Based Image Retrieval*) para buscar imágenes similares basándose en sus características como colores, formas, texturas o cualquier otra información que pueda extraerse del contenido de la imagen. Este sigue siendo un campo ampliamente estudiado debido a la dificultad de expresar mediante palabras las cualidades gráficas de una imagen. Algunos ejemplos de su aplicación los podemos encontrar en áreas tan variadas como el campo del diagnóstico en medicina, sistemas de información geográfica (GIS) o bancos de imágenes de museos para localizar obras de arte.

Un caso particular de los sistemas CBIR es la recuperación de imagen de marcas registradas (*Trademark Image Retrieval* o TIR), que consiste en la búsqueda de logotipos similares. Las imágenes de marca, comúnmente conocidas como logotipos o logos, se diferencian de las reales porque están creadas artificialmente y se han diseñado para que tengan impacto visual. Están formadas por una serie de elementos que pueden representar objetos conocidos, pasando por distintos grados de simplificación hasta llegar a formas abstractas, pudiendo contener texto o no.

La forma es probablemente la característica más utilizada por el ser humano para caracterizar una imagen de marca, pero categorizar estas imágenes automáticamente es una tarea compleja. Por ejemplo, un logo que contiene un triángulo con un circulo en medio será más parecido a otro con los mismos elementos y con igual distribución, aunque deformados, que otro logo con las mismas formas exactas pero en una distribución diferente (Schietse et al., 2007). Además de la estructura de los elementos que lo componen y su organización, se debe considerar la interpretación semántica que determina los objetos reconocibles que contiene, lo cual es una tarea muy compleja, como demuestran los estudios realizados sobre cómo los humanos perciben e interpretan las imágenes.

Además de la marca registrada a la que pertenecen, los logos también se pueden clasificar usando diferentes criterios de diseño como son el color, la forma, los símbolos o elementos reconocibles que contienen (y definen su semántica), o el sector de la actividad a la que pertenece la marca (y que por tanto representan).

Como profesora en escuelas de Arte y Superior de Diseño de la Comunidad Valenciana, estoy interesada en el análisis y reconocimiento de imagen, y en especial de logos. Por ello, en esta tesis he estudiado los procesos relacionados con tareas de similitud de imágenes, empezando por los métodos basados en descriptores locales usando vecinos más cercanos. Para llevar a cabo esta tarea, inicialmente se han usado los datos recopilados en MirBot (Pertusa et al., 2018), una aplicación móvil para etiquetar imágenes de forma colaborativa. También se ha desarrollado un algoritmo de verificación geométrica, *Segment Intersection of Interest Points* (SIIP), con el objetivo de reordenar los resultados de la búsqueda por similitud favoreciendo aquellas imágenes que tienen una distribución geométrica similar. A continuación, con la incorporación de las redes neuronales convolucionales, se han probado y evaluado distintas arquitecturas de aprendizaje profundo para la extracción de características neuronales y la recuperación de imágenes similares usando los datos de MirBot.

Como resultado final de la tesis se presenta la que considero mi principal contribución, un sistema de búsqueda de similitud de logos. Este sistema tiene en cuenta los diferentes aspectos que componen una imagen, tanto la parte de estructura (definida principalmente por la forma, color y presencia o no de texto) como la semántica para determinar los objetos reconocibles o no que contiene. En esta tarea normalmente la imagen tiene más de una etiqueta simultánea (por ejemplo, puede contener varios colores u objetos reconocibles), por lo que se trata de un problema de clasificación multi-etiqueta (*Multi-Label Classification* o MLC).

1.1. Propuesta y objetivos

El objetivo principal de la tesis gira en torno a la búsqueda de imágenes por similitud y, en concreto, a la evaluación y propuesta de técnicas para la recuperación y clasificación de imágenes de marcas similares.

Hasta llegar al desarrollo del sistema presentado para evaluar la similitud de logos, durante mi etapa de investigación he realizado un estudio de las técnicas existentes para la búsqueda de imágenes similares, implementando y comparando varias técnicas basadas en descriptores tradicionales para comprobar su efectividad que se describen en las Secciones 4.2 y 4.3 de esta tesis. Adicionalmente se ha desarrollado un método de verificación geométrica sobre puntos de interés que se detalla en la Sección 4.3.3.

Como puede verse en la literatura revisada en el Capítulo 3, las técnicas de visión artificial han ido evolucionado hasta utilizar características neuronales, por lo que mi investigación también ha girado en torno a ellas. En la Sección 4.4 se presenta el cambio de las características locales inicialmente evaluadas en MirBot al uso de aproximaciones neuronales, lo cual sirve de ejemplo para ilustrar el recorrido realizado por las diferentes técnicas existentes y su evolución.

Por último, como resultado final y aportación principal de esta tesis, en el Capítulo 5 se describe un sistema de búsqueda de similitud de logos. Para ello, se usa un conjunto de datos de la EUIPO (Oficina de Propiedad Intelectual de la Unión Europea) llamado EUTM (*European Union Trademark*) que, además de las imágenes, contiene metadatos con información sobre colores, formas, sectores y elementos figurativos.

En base a este tipo de datos, que son representativos del etiquetado habitualmente utilizado, se propone un método de búsqueda por similitud multi-etiqueta de imagen de marca. Para ello se combinan técnicas de pre-procesamiento, para por ejemplo detectar el texto, con redes neuronales convolucionales especializadas en la detección de características concretas de logotipos. Para esto se han estudiado topologías aplicables a la imagen de marca y su relación con los metadatos de la base de datos utilizada. También se ha utilizado una arquitectura tipo Auto-Encoder (AE) para extraer una representación compacta del logo que contiene su apariencia general, y que también puede usarse para buscar logotipos similares. Por último, toda esta información se combina en una función de distancia que permite ajustar el criterio seguido en la recuperación de logotipos similares.

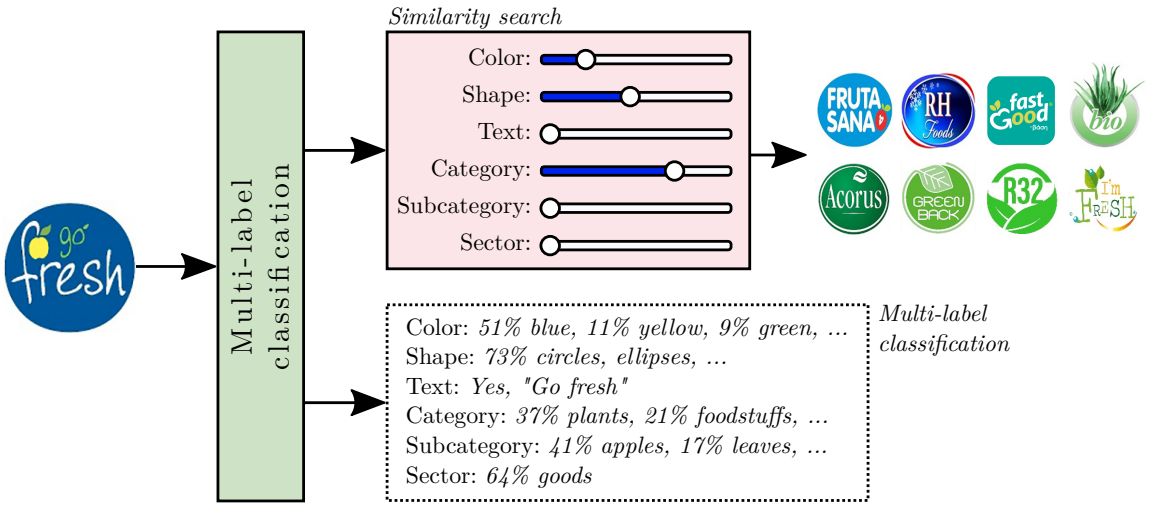


Figura 1.1.: *Interfaz de la aplicación diseñada para búsqueda de logos por similitud atendiendo a las preferencias del usuario.*

En la Figura 1.1 se representa el sistema propuesto, mostrando la interfaz de entrada y las salidas obtenidas. Como se puede ver, este sistema permite al usuario determinar las características que desea priorizar en la búsqueda por similitud de un logo para, de este modo, obtener como resultado un ranking de imágenes similares tras la aplicación de las características ponderadas establecidas. Hasta la fecha no se conoce ningún método de clasificación multi-etiqueta para logos de este tipo que permita obtener un ranking de imágenes similares con aplicación de diferentes características ponderadas y, además, configurable por los usuarios.

Para evaluar el sistema, y puesto que la semántica de marcas puede resultar muchas veces subjetiva, se han verificado los resultados mediante encuestas a estudiantes y profesionales del diseño, demostrando que el sistema propuesto mejora los resultados de los sistemas manuales incluso entre personas con conocimientos de diseño gráfico y composición de imágenes.

El sistema desarrollado puede contribuir a mejorar el proceso de etiquetado para la clasificación de una imagen, ya que ofrece una propuesta de clasificación con la probabilidad de etiquetado para cada una de las clases (hay que considerar que para esto se utiliza un etiquetado con cientos de clases posibles). La clasificación y búsqueda por similitud realizada, además de ayudar a detectar similitudes o plagios entre imágenes de marca, también resulta de utilidad para estudiantes y profesionales del diseño, ya que permitirá buscar referentes, ideas o estilos.

1.2. Estructura de la tesis

Esta tesis tiene la siguiente estructura:

- En el Capítulo 1 se han presentado los objetivos de la tesis, definiendo su contexto y enumerando las contribuciones realizadas durante su desarrollo.
- En el Capítulo 2 se describen los fundamentos de las técnicas de aprendizaje automático y aprendizaje profundo utilizadas.
- En el Capítulo 3 se revisa el estado del arte para los sistemas de búsqueda de similitud entre imágenes, con especial atención a la imagen de marca, y diferenciando entre técnicas tradicionales y neuronales. También se incluye un apartado con las bases de datos utilizadas en la literatura para búsqueda y recuperación de imagen de marca.
- El Capítulo 4 describe las técnicas desarrolladas para búsqueda de similitud entre imágenes. Presentamos la aplicación MirBot desarrollada con la implementación de descriptores locales y su adaptación posterior a descriptores neuronales. También se incluye una contribución (SIIP) para realizar verificación geométrica sobre descriptores locales.
- En el Capítulo 5 se detalla el sistema de búsqueda por similitud de logos que se ha desarrollado, el cual tiene en cuenta múltiples aspectos que definen una imagen. Inicialmente se presenta el estudio de topologías realizado de la imagen de marca y su relación con la codificación de Viena presente en la base de datos. A continuación se presenta la metodología utilizada, su evaluación y el análisis visual de los resultados, comparándolos con las encuestas realizadas.
- El Capítulo 6 contiene las conclusiones finales.

1.3. Contribuciones

Durante este trabajo se ha contribuido a desarrollar la aplicación para dispositivos móviles llamada MirBot que consiste en un sistema multimodal interactivo para reconocimiento de imagen.

Además, se han realizado una serie de publicaciones que se enumeran a continuación.

- Congresos:

  - Pertusa, A.; Gallego, A.J.; Bernabeu, M. "MirBot: A multimodal interactive image retrieval system". Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA, Lecture Notes in Computer Science, vol. 7887, pp. 197–204 (2013)

  - Bernabeu, M.; Pertusa, A.; Gallego, A.J. "Image spatial verification using Segment Intersection of Interest Points". Proc. of the 24 Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), ISBN: 2464-4614 (2016)

  - Gallego, A.J.; Pertusa, A.; Bernabeu, M. "Multi-Label Logo Classification using Convolutional Neural Networks". Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA, ISBN: 978-3-030-31332-6, pp. 485–497, Madrid, Spain (2019)

- Revistas:

  - Pertusa, A.; Gallego, A.J.; Bernabeu, M. "MirBot: A collaborative object recognition system for smartphones using convolutional neural networks". Neurocomputing, vol. 293, pp. 87–99 (2018)

  - En proceso de revisión: Bernabeu, M.; Gallego, A.J.; Pertusa, A. "Multi-Label Logo Recognition and Retrieval based on Weighted Fusion of Neural Features". Preprint: https://arxiv.org/abs/2205.05419

CAPÍTULO 2

Fundamentos

2.1. Inteligencia Artificial

La inteligencia artificial (IA) es un campo con muchas aplicaciones prácticas como la automatización de trabajos rutinarios, el reconocimiento del habla, el tratamiento de música o imágenes, la asistencia a diagnósticos médicos, etc.

En 1956 se utilizó por primera vez este término, aunque no fue hasta la década de los 80 cuando comenzó a desarrollarse. En sus inicios se resolvieron problemas que son intelectualmente difíciles para los seres humanos pero relativamente sencillos para los ordenadores. Estos son los problemas que pueden describirse mediante una lista de reglas matemáticas formales. El verdadero desafío para la IA resultaron ser las tareas fáciles de realizar para las personas pero difíciles de describir formalmente, es decir, problemas que resolvemos intuitivamente, como reconocer el habla o identificar rostros en imágenes (Goodfellow et al., 2016).

Para solucionar este tipo de problemas más complejos se propusieron una serie de técnicas que permiten generar modelos a partir de los datos que se les proporcionan. A este conjunto de técnicas se les conoce como aprendizaje automático o *Machine Learning* (ML) y se considera una rama de la IA. No obstante, estas técnicas también presentaban sus limitaciones, como son las dificultades para que los modelos aprendieran ciertos conceptos, de alto nivel, a partir de los datos en bruto. Una solución a este problema consiste en el uso de técnicas que permitan a los modelos representar el mundo en términos de jerarquía de conceptos, de forma que los conceptos complejos se puedan descomponer en otros más simples. De este modo, el modelo aprende características para el problema concreto a resolver introduciendo representaciones que se expresan en términos de otras representaciones más simples. Si mostrásemos

mediante una gráfica esta jerarquía de conceptos, podríamos ver que tiene muchos niveles construidos unos sobre otros, por esta razón a este último enfoque se le ha denominado aprendizaje profundo o *Deep Learning* (DL), tal y como se describe en Goodfellow et al. (2016).

En la Figura 2.1 se muestra cómo se relacionan y agrupan las metodologías mencionadas. En la parte superior está la IA, que cubre una amplia variedad de métodos, entre los que se encuentran el aprendizaje automático y el aprendizaje profundo, los cuales vamos a ver con más detalle a continuación.

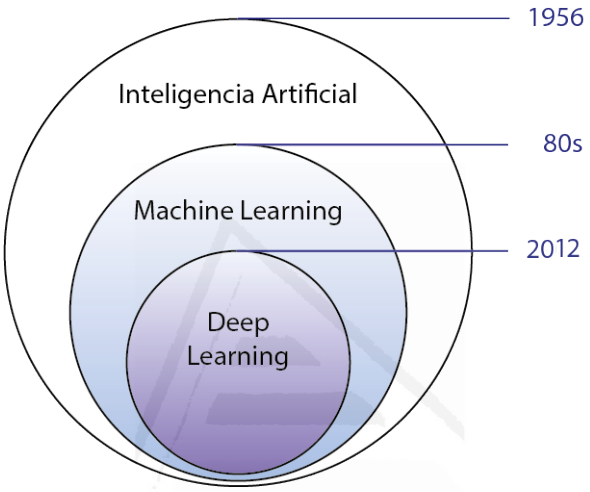


Figura 2.1.: *Esquema de la relación entre la inteligencia artificial, aprendizaje automático y aprendizaje profundo. Imagen adaptada de Taulli (2019)*

2.2. Aprendizaje automático

Se conoce como aprendizaje automático a la rama de la IA que permite generar modelos para resolver problemas a partir del uso de información y sin que sean explícitamente programados para ello. En esta aproximación se realiza un proceso de aprendizaje utilizando un conjunto de datos con información sobre la tarea (denominado conjunto de entrenamiento) y se genera una hipótesis a partir de estos datos que posteriormente puede ser utilizada para realizar predicciones ante nuevos datos del dominio de la tarea (denominado conjunto de evaluación o test). Es de vital importancia el conjunto de datos suministrados para el entrenamiento de los modelos, pues de su naturaleza dependerá en gran medida el éxito del modelo generado. Es importante contar con un alto número de muestras en este conjunto de datos y que

además sean variadas, representativas de la tarea y que consideren todas las posibles casuísticas sobre las que se quiera aplicar posteriormente el modelo.

El aprendizaje automático puede ser de tipo supervisado o no supervisado. En un algoritmo de aprendizaje supervisado, en el conjunto de entrenamiento disponemos tanto de los datos de entrada como de la salida esperada (es decir, los datos están etiquetados con la predicción asociada que se espera obtener). En un algoritmo de aprendizaje no supervisado solo disponemos de los datos de entrada, por lo que desconocemos las etiquetas (o predicciones esperadas como salida). En términos generales, el aprendizaje no supervisado implica observar varios ejemplos o muestras de entrada (representadas como un vector de datos que denominamos x) e intentar aprender implícita o explícitamente su distribución de probabilidad $p(x)$, o algunas propiedades interesantes de esa distribución. Por otro lado, el aprendizaje supervisado implica la observación de los ejemplos de entrada x y de un valor o vector asociado y con la etiqueta o salida esperada, y, en base a estos, aprender a predecir y a partir de x, generalmente mediante la estimación de $p(y|x)$.

Haciendo un símil, el término aprendizaje supervisado representaría el caso de un instructor o profesor que muestra a los alumnos lo que deben hacer, proporcionando un objetivo a aprender y dando una serie de ejemplos para que los alumnos aprendan. En el aprendizaje no supervisado no hay instructor o profesor, por lo que el algoritmo debe aprender a dar sentido a los datos sin esta guía (Goodfellow et al., 2016).

El aprendizaje supervisado se utiliza normalmente para predecir una clase (clasificación) o un valor numérico (regresión) a partir de unos datos de entrada, mientras que un algoritmo de aprendizaje no supervisado se utiliza habitualmente para detectar patrones o aprender estructuras relevantes en los datos de entrada, como agrupaciones. Algunos algoritmos de aprendizaje supervisado bien conocidos son, entre otros, *Support Vector Machines* (SVM) propuesto por Cortes and Vapnik (1995), *k-Nearest Neighbors* (k vecinos más cercanos), redes neuronales (LeCun et al., 2015), árboles de decisión, inicialmente propuestos por Breiman et al. (1983) y que han evolucionado, entre otras, a una metodología conocida como CART (*Classification and Regression Trees*) que proporciona un marco para generar diferentes árboles de decisión (Duda et al., 2001). Ejemplos de algoritmos de aprendizaje no supervisado son *Principal Components Analysis* (PCA) (Jolliffe, 2011) y *k-Means clustering* (Jin and Han, 2010).

A continuación se describen con más detalle las técnicas de aprendizaje automático más relevantes para esta tesis.

2.2.1. k-Nearest Neighbors

En términos generales, los k-vecinos más cercanos o kNN es una técnica de clasificación supervisada de tipo no paramétrica. Por tanto, este algoritmo no tiene parámetros a aprender, de hecho ni siquiera existe realmente una etapa de entrenamiento o un proceso de aprendizaje (de aquí que también se denomine como *lazy learning* o aprendizaje vago) (Cover and Hart, 1967). En cambio, en la fase de test, para clasificar una muestra q se buscan sus k-vecinos más cercanos en el espacio del conjunto de datos de entrenamiento X, devolviendo como predicción la etiqueta más común entre los vecinos encontrados. Formalmente, se puede definir kNN de la siguiente forma:

$$k\text{NN}(q) = \text{mode}\left(Y_k\left(\arg\min_{x_i \in T}{}_{\text{k}}\{d(q, x_i)\}\right)\right) \tag{2.1}$$

donde k es el número de vecinos considerados, $d(q, x_i)$ denota la distancia entre la consulta q y el prototipo x_i de X, e Y_k es el conjunto de etiquetas recuperadas de los k elementos más cercanos a la consulta q. Para comparar los prototipos se pueden utilizar diversas medidas de distancia, siendo la distancia Euclídea la más común, aunque también podemos aplicar métricas de Minkowski, distancia de edición entre cadenas, etc.

Siempre que el número de muestras de entrenamiento sea lo suficientemente grande, esta regla simple generalmente obtiene un buen resultado. Esto también se fundamenta en resultados teóricos. Cuando $N \to \infty$, siendo N el tamaño del conjunto de entrenamiento, la probabilidad de error de clasificación P_{NN} para el vecino más cercano (representando NN al método kNN cuando $k = 1$) está delimitada por:

$$P_B \leq P_{NN} \leq P_B\left(2 - \frac{M}{M-1}P_B\right) \leq 2P_B \tag{2.2}$$

donde P_B es el error de Bayes y M el número de clases[1]. Así pues, el error cometido por el clasificador NN es, como mucho, el doble que el del error de Bayes. El análisis teórico de kNN es mejor que el de NN. Sin embargo, en la práctica existen casos donde kNN obtiene peores resultados que NN.

[1] https://www.prhlt.upv.es/~evidal/students/app/tema4/t4app2p.pdf

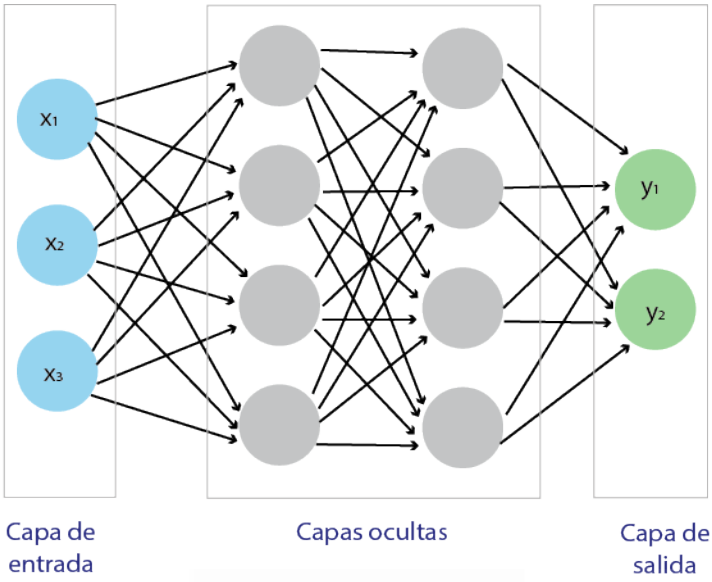


Figura 2.2.: *Estructura de una red neuronal con tres neuronas de entrada, dos capas ocultas con cuatro neuronas cada una y una capa de salida con dos neuronas.*

En general, teniendo en cuenta que en la práctica los valores de k y el número de muestras con los que se trabaja son finitos, se puede afirmar que el método del vecino más cercano se encuentra entre los candidatos a ser adoptado como clasificador en una amplia serie de aplicaciones.

2.2.2. Redes Neuronales

Otro tipo de algoritmo de aprendizaje automático son las redes neuronales artificiales, las cuales se inspiran en el cerebro humano y su estructura, formada por millones de neuronas interconectadas entre sí. Con este fin, McCulloch and Pitts (1943) propusieron un modelo matemático en términos de un proceso computacional que emula la actividad nerviosa que nuestro cerebro utiliza para transmitir señales entre las neuronas. En la Figura 2.2 se ejemplifica el diseño de una red neuronal artificial formada por la interconexión de una serie de neuronas sintéticas. Como se puede ver, en esta red las neuronas se organizan en capas, desde una capa de entrada (que recibe los datos), pasando por una serie de capas ocultas (que procesan la información) hasta llegar a una capa de salida (que devuelve la predicción realizada). El número de capas y el número de neuronas en cada capa depende de la naturaleza del problema, del tamaño del conjunto de datos y de la tarea a realizar.

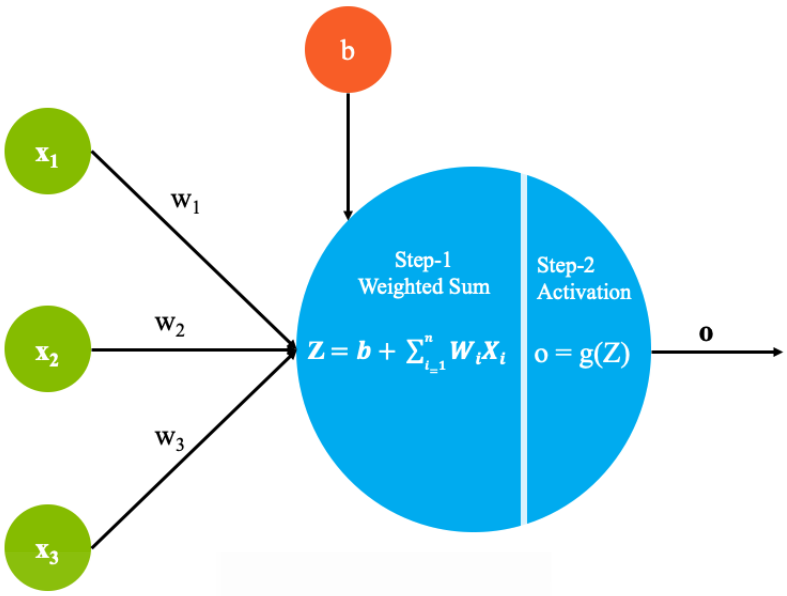


Figura 2.3.: *Estructura de una neurona artificial. Imagen obtenida de Madan and Madhavan (2020).*

Para las neuronas de estas capas se utiliza el modelo propuesto por Rosenblatt (1958), al que denominó "perceptrón". Cada una de estas neuronas artificiales se conecta con todas las neuronas de la capa anterior, de las cuales recibe su entrada, y con todas las de la siguiente capa, a las cuales envía su resultado. Estas conexiones emulan las sinapsis de las neuronas biológicas y la transmisión de señales electroquímicas para estimular a otras neuronas.

En la Figura 2.3 se muestra una representación de una de estas neuronas artificiales o perceptrones. Como se puede ver, la neurona recibe una serie de entradas x_i, las multiplica por un peso w_i, a continuación suma los resultados de la ponderación y le añade un bias b. Este resultado finalmente se transforma mediante una función de activación g, la cual determina el valor de salida. Es decir, la operación que realiza cada una de estas neuronas es $o = g\left(\sum_{i=1}^{n} w_i x_i + b\right)$, donde los pesos w_i y el bias b son los parámetros a ajustar durante el proceso de entrenamiento de la red.

Siguiendo el esquema con la arquitectura de la red que hemos visto previamente en la Figura 2.2, el dato recibido como entrada de la red se envía a cada una de estas neuronas, las cuales realizan el proceso descrito y devuelven un resultado que dependerá de los pesos aprendidos durante el entrenamiento. Este resultado se conecta con todas las neuronas de la siguiente capa. Por este motivo también se denominan

a estas capas como totalmente conectadas o *fully connected layers*. De esta forma, la entrada se va procesando capa tras capa, hasta transformarla en una representación linealmente relacionada con la tarea a resolver pudiendo, por tanto, calcular a partir de este valor la predicción devuelta como salida de la red.

2.2.2.1. Backpropagation

Antes de utilizar la red neuronal para realizar predicciones es necesario realizar un proceso de entrenamiento para ajustar los parámetros de la misma. En este proceso, básicamente se trata de ir progresivamente modificando el valor de los pesos de la red para minimizar el error cometido en las predicciones realizadas.

Para esto se emplea el algoritmo de *backpropagation* o retro-propagación, el cual consta de dos pasos (Goodfellow et al., 2016). En primer lugar se realiza una pasada hacia adelante (paso *forward*) en la que la red procesa las muestras de entrenamiento. Las predicciones realizadas por la red se comparan con el etiquetado de los datos, el cual indica el resultado esperado, y en base a este se calcula el error cometido usando una función de pérdida como, por ejemplo, el error medio al cuadrado.

Posteriormente, se realiza una pasada hacia atrás (paso *backward*) en el que se ajustan los pesos de las neuronas para reducir este error, empezando desde la capa de salida y propagando el error hasta la capa de entrada. En cada neurona se ajustan sus pesos en función de su contribución al error. Para calcular esto se emplea el método de optimización de descenso por gradiente basado en derivadas parciales y usando la regla de la cadena. De esta forma se determina el ajuste a realizar y se modifica el valor de los parámetros siguiendo la dirección opuesta al gradiente calculado (LeCun et al., 2015).

Estos dos pasos se repiten durante un número determinado de iteraciones (denominadas épocas), para así ir ajustando poco a poco los parámetros de la red. En cada iteración los pesos se ajustan solo un factor del error cometido determinado por el valor del *learning rate* o ratio de aprendizaje, para de esta forma ir acercándose progresivamente a un mínimo en la función de error.

Una vez finalizado el proceso de entrenamiento se fijan los mejores parámetros encontrados y se genera un modelo de red con dicha configuración, el cual ya puede ser utilizado para realizar predicciones ante nuevas muestras.

2.2.2.2. Funciones de activación

La función de activación que aplica cada neurona permite modificar su resultado y de esta forma minimizar o maximizar determinados valores. Esto nos permite utilizar diferentes tipos de funciones de activación dependiendo de la capa y de la tarea a realizar. Por ejemplo, podemos utilizar funciones de activación de paso binario, que umbralizan la salida y devuelven, por tanto, solo valores discretos (1 o 0, indicando si se activa o no la salida, respectivamente). También podemos aplicar funciones de activación lineales, las cuales no realizan ningún proceso, simplemente devuelven el mismo valor, resultando adecuadas para tareas de regresión en las que se quieran predecir valores continuos. Sin embargo, las funciones de activación más utilizadas son las no lineales, ya que permiten el aprendizaje de distribuciones de datos complejas (no separables linealmente). Algunas de las más comunes son:

- La función sigmoidea, la cual se se define como:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

  Esta ha sido una de las funciones más utilizadas en las primeras propuestas de redes neuronales.

- La función ReLU (*Rectified Linear Unit*) es una de las funciones más usadas en las redes neuronales profundas (que se describirán en la Sección 2.3), ya que resulta más eficiente computacionalmente y además evita problemas con los valores del gradiente durante el entrenamiento (*vanishing and exploding gradients*). Esta función se define como:

$$\sigma(z) = \max(0, z) \tag{2.4}$$

Además de las funciones mencionadas, cuando se trata de problemas de clasificación se suele utilizar la función de activación Softmax para la capa de salida de la red, la

cual se define como:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \text{ para } i = 1, \ldots, C \tag{2.5}$$

donde C representa el número de clases y z el vector de salida de la red. Esta función normaliza los valores de salida entre 0 y 1, cuya suma siempre será uno, representando una distribución de probabilidad sobre las C clases a clasificar.

2.2.2.3. Regularización

El objetivo principal en aprendizaje automático consiste en que el modelo funcione adecuadamente ante entradas nunca antes vistas, y no solo en aquellas con las que se entrenó. Durante el proceso de *training* se minimiza el error usando el conjunto de entrenamiento y, una vez finalizada esta fase, se espera que al aplicar el modelo generado sobre el conjunto de test este error también sea mínimo. Cuando hay una gran diferencia entre el error de entrenamiento y el error de test se produce lo que conocemos como *overfitting* o sobreajuste, y se debe a que la red memoriza los datos de entrenamiento pero no es capaz de reconocer nuevos datos correctamente. Es decir, la red no es capaz de generalizar.

Para evitar este problema se suelen aplicar métodos de regularización, entre los cuales destacan los siguientes:

- **Aumentado de datos** (Krizhevsky et al., 2012) es una técnica que consiste en generar artificialmente variaciones en los datos utilizados durante el entrenamiento. Para esto, en el caso del uso de imágenes se puede, por ejemplo, aplicar transformaciones como traslaciones, rotaciones, cambios de iluminación o color, etc. Con esto se consigue aumentar la variabilidad de representación de los datos y, por tanto, forzar a que el modelo tenga que generalizar.

- **Early stopping** es una forma de regularización muy utilizada en aprendizaje automático. La idea es detener automáticamente el proceso de entrenamiento cuando el resultado sobre el conjunto de validación[2] no mejora durante una serie

[2]La partición de validación se extrae del conjunto de entrenamiento y se utiliza solamente para evaluar los hiperparámetros del modelo durante el entrenamiento.

de iteraciones.

- **Dropout** (Krizhevsky et al., 2012) es un tipo de capa que podemos añadir a una arquitectura de red y que, durante el proceso de entrenamiento, desactivará aleatoriamente (con una probabilidad dada, normalmente entre 0,2 y 0,5) algunas conexiones de la red. De esta forma se previene que la red sea demasiado dependiente de conexiones particulares, forzando de esta forma a que no pueda aprender características específicas y que deba generalizar.

- **Batch Normalizacion** (Ioffe and Szegedy, 2015) es también un tipo especial de capa de regularización que normaliza los datos de cada batch de entrenamiento. De esta forma se consigue normalizar los datos también en capas intermedias de la red, no solo a la entrada de la misma, lo que ayuda al proceso de entrenamiento y además suele mejorar su tasa de acierto.

2.3. Aprendizaje profundo

Con el paso de los años las redes neuronales fueron evolucionando, no solamente gracias a la aparición de nuevas técnicas, sino también al incremento en la capacidad de procesamiento y de almacenaje de los ordenadores, y al notable aumento en las cantidades de datos disponibles con los que entrenar y evaluar estos modelos. La combinación de todos estos avances produjo una revolución en este tipo de metodologías, mejorando notablemente sus resultados y llegando en ocasiones a superar a expertos humanos en la realización de algunas tareas.

Las redes neuronales tradicionales presentaban problemas para procesar bases de datos complejas, cuando por ejemplo la tarea a resolver tenía una alta variabilidad, como podría ser el caso de clasificación de imágenes en general, en la que un mismo objeto puede tener múltiples representaciones, aparecer con diferentes formas, colores, iluminaciones, posiciones, etc. Además también se encontraban problemas para entrenar redes neuronales grandes, con muchas capas o neuronas, debido a que o bien no llegaban a converger o aprendían características muy concretas de la tarea que no generalizaban bien ante nuevos datos. A raíz de una serie de publicaciones realizadas en 2006 (Hinton et al., 2006; Bengio et al., 2006) se demostró que era posible entrenar redes neuronales profundas, con muchas capas, y solucionar estos problemas mencionados. Todos estos motivos llevaron finalmente a la aparición de lo que actualmente

conocemos como aprendizaje profundo o *Deep Learning* (Goodfellow et al., 2016).

Estas técnicas aprenden representaciones de características que se expresan en términos de otras más simples, es decir, aprenden conceptos complejos a partir de conceptos más sencillos. Lo más relevante es que esta jerarquía de características se aprende automáticamente, es decir, los modelos aprenden la representación más adecuada para un conjunto de datos dado. Tal como indican LeCun et al. (2015), la capacidad para aprender las representaciones directamente de los datos sin procesar ha supuesto una revolución en muchas áreas de investigación, como la visión por computador o el reconocimiento de voz, entre otras.

En el aprendizaje profundo también se aplica el método de *backpropagation* y las técnicas de regularización que hemos visto previamente. Además, debido al avance de las técnicas también han surgido nuevos tipos de capas y operaciones a realizar, gracias a lo cual es posible construir arquitecturas de redes para resolver distintos tipos de tareas. Entre las más utilizadas para el procesamiento de imagen encontramos las redes neuronales convolucionales (*Convolutional Neural Networks* o CNN) y también las arquitecturas tipo Auto-Encoder Convolucionales, las cuales, debido a su uso en esa tesis doctoral, detallaremos a continuación.

2.3.1. Redes Neuronales Convolucionales

Las CNN son un tipo especial de redes neuronales diseñadas para procesar datos de entrada en forma de matrices bidimensionales, como por ejemplo una imagen en color que contiene niveles de intensidad de píxeles en 3 canales RGB (LeCun et al., 2015). Estas redes, por tanto, están enfocadas principalmente para su uso en tareas de visión por computador, como la clasificación de imágenes, la segmentación o la detección de objetos, para las que obtienen resultados mucho mejores que las aproximaciones previas.

Las neuronas utilizadas, denominadas filtros convolucionales o capas de convolución, se inspiran en las neuronas biológicas de la corteza visual primaria del cerebro, en las que sus campos receptivos se activan al detectar determinados patrones. En las primeras capas de las CNN se aplican este tipo de neuronas para la extracción de características de las imágenes. Estas capas se suelen alternar con capas de submuestreo (o *pooling*), para de esta forma ir reduciendo progresivamente el tamaño de la entrada. Esto permite formar la jerarquía de características comentada, ya que los

primeros filtros tendrán un campo receptivo menor y solo podrán fijarse en detalles, y al ir reduciendo el tamaño de la imagen el campo receptivo irá abarcando una mayor proporción de la entrada, por lo que cada vez se podrán ir fijando en características de mayor nivel, las cuales, a su vez, se basarán en las características previamente extraídas. Por último, en este tipo de estructuras, se suelen añadir una o más capas completamente conectadas (*fully-connected*) para obtener el resultado final.

La primera red CNN se introdujo en 1989 por LeCun et al. (1989), aunque no fue realmente hasta 2012, con la aparición de AlexNet (Krizhevsky et al., 2012), cuando se centró la atención en este tipo de arquitecturas. Esto se debió a que AlexNet ganó por una notable diferencia (superior al 10 %) la competición de clasificación de imágenes ImageNet (Deng et al., 2009). Desde 2012 las propuestas para procesamiento de imagen se han centrado en el uso de CNN, planteándose nuevas arquitecturas que han mejorado los resultados año tras año. Algunas de estas arquitecturas se revisarán en detalle en el Capítulo 3.

A continuación se describen los conceptos básicos de las redes neuronales convolucionales.

2.3.1.1. Convolución

El término CNN viene del tipo de operación que utilizan, la convolución discreta, que consiste en aplicar un filtro o *kernel* (una matriz) a una entrada (por ejemplo, una imagen). Para calcular la convolución se multiplican los valores de su *kernel* por los de la imagen y después se suma este resultado como puede verse en la Figura 2.4. El filtro se va desplazando por la imagen realizando esta operación, finalmente devolviendo como resultado otra matriz del mismo tamaño que la entrada que se denomina mapa de activaciones (o mapa de características, *feature map*). El desplazamiento del *kernel* sobre la imagen suele ser de 1 píxel. Sin embargo también se puede incrementar el valor del parámetro *stride* para que el filtro se desplace con un espaciado mayor.

Las convoluciones se han usado tradicionalmente en visión por computador para detectar bordes, suavizar imágenes, resaltar colores, etc. La diferencia que añaden las CNN es que aprenden los coeficientes de los filtros que resultan más adecuados para resolver la tarea en cuestión. En las CNN, después de calcular la convolución se suele aplicar una función de activación al igual que en las redes neuronales tradicionales. En concreto, esta función se puede definir como $o = g\left(\sum_{i=1}^{n} K \otimes x_i + b\right)$, donde K es

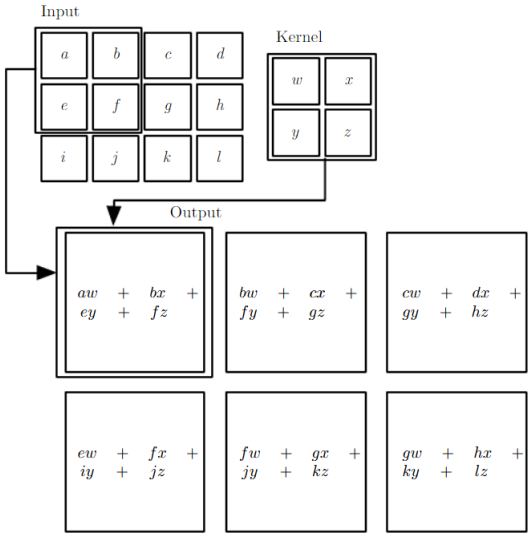


Figura 2.4.: *Ejemplo de convolución 2D extraído de Goodfellow et al. (2016).*

el *kernel* que se aplica mediante la operación de convolución $\otimes$ sobre las entradas x_i, mientras que b y g representan el *bias* y la función de activación, respectivamente. Por tanto, en el diseño de una CNN se debe determinar el tamaño y número de filtros junto con las funciones de activación, y durante el proceso de entrenamiento se aprenderá el *bias* y los valores del *kernel*.

2.3.1.2. Submuestreo

Las CNN aprovechan la propiedad de que muchas señales naturales tienen una composición jerárquica en las que características de nivel superior se obtienen mediante composición de las de nivel inferior. En imágenes, por ejemplo, combinaciones locales de bordes forman motivos, los motivos se agrupan en partes y las partes forman objetos. La operación de submuestreo (*pooling*) permite que las representaciones varíen muy poco cuando los elementos de la capa anterior cambian de posición y apariencia (LeCun et al., 2015).

Normalmente, en una CNN las capas de submuestreo se alternan con las capas convolucionales. El objetivo del submuestreo es reducir la dimensionalidad del mapa de características, bien sea calculando el promedio (*average pooling*) o el valor máximo (*max pooling*) de una región cuyo tamaño se determina en el diseño de la red. La

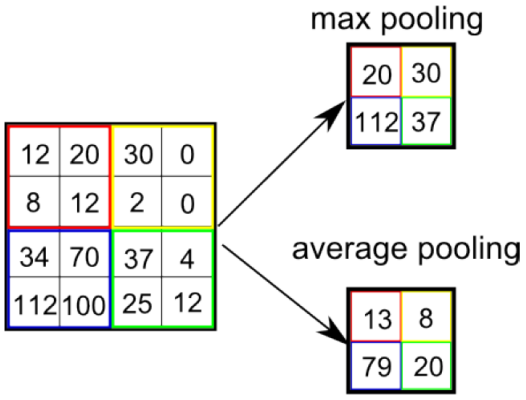


Figura 2.5.: *Ejemplo de submuestreo con dos resultados distintos usando promedio y máximo. Imagen extraída de* https://ichi.pro/es/agrupacion-maxima-36351685447441

Figura 2.5 ilustra el resultado de la operación usando submuestro promedio y máximo. Con el máximo cada filtro toma el valor máximo de la región y con el promedio el valor medio.

2.3.2. Auto-Encoders

Otro tipo de redes neuronales que se utilizan habitualmente para el procesamiento de imágenes y que consideraremos también en el desarrollo de esta tesis, son los Auto-Encoders (AE), propuestos inicialmente por Hinton and Zemel (1994). Se trata de una arquitectura cuyo objetivo es obtener a la salida una reconstrucción de la imagen de entrada, particularidad que nos permite realizar un entrenamiento no supervisado y que, debido al diseño de la red, consigue una representación o codificación reducida de los datos.

Por lo general, un AE se divide en dos partes: el codificador $h = f(x)$, que recibe la imagen de entrada x y crea una representación codificada, y el decodificador, que realiza la reconstrucción $r = g(h)$. En el codificador normalmente se reduce progresivamente la dimensión de los datos de entrada hasta llegar a la capa intermedia h, a partir de la cual se realiza el proceso de decodificación hasta obtener la salida. Esta reducción de dimensionalidad de la entrada fuerza a que en la representación intermedia se extraigan las características más representativas de los datos, ya que a partir de esta codificación se ha de reconstruir de nuevo la entrada con el menor error

posible.

Durante el proceso de entrenamiento se minimiza el error de reconstrucción según la siguiente función de pérdida:

$$L(x, g(f(x))) \tag{2.6}$$

donde L es una función de error, como por ejemplo *Mean Squared Error* (MSE), que calcula la diferencia entre la entrada y la reconstrucción realizada por la red.

Cuando el decodificador es lineal y L es el MSE, un AE aprende a abarcar el mismo subespacio que PCA. Según Goodfellow et al. (2016), los Auto-Encoders con funciones de codificación y decodificación no lineales pueden aprender una generalización no lineal más robusta que PCA.

En el caso de aprendizaje con imágenes se suelen usar Auto-Encoders Convolucionales (CAE) (Masci et al., 2011), en los cuales las capas del codificador son capas convolucionales y las del decodificador son capas convolucionales traspuestas, que revierten las operaciones de convolución realizadas. En estas arquitecturas se suelen emplear capas de submuestreo combinadas con las capas convolucionales para reducir la dimensionalidad en la etapa de codificación.

CAPÍTULO 3

Estado del Arte

Se conoce como CBIR (*Content-Based Image Retrieval*) al conjunto de técnicas orientadas a recuperar automáticamente imágenes basándose en características como color, formas, texturas u otro tipo de información que pueda extraerse del contenido de la imagen, como pueden ser descriptores semánticos para determinar el tipo de objeto que contiene (Eakins, 1999). Los sistemas CBIR surgieron en la década de los 90 para solucionar los problemas de la indexación manual, consistente en la asociación de términos a las imágenes. Estos problemas suponían (y siguen suponiendo) una gran inversión en tiempo debido a diversos factores, como la dificultad de expresar mediante palabras las cualidades gráficas de una representación visual, las limitaciones en los sistemas de etiquetado u otras causas que influyen en la percepción visual de las personas encargadas en realizar este proceso, como pueden ser factores sociales o culturales. Parte de este problema se soluciona con el uso de sistemas de etiquetado de imágenes que eviten ambigüedades semánticas, por ejemplo aquellos que usan diccionarios como Wordnet basados en organización jerárquica (Oram, 2001).

En este capítulo se describe el estado actual de las técnicas de búsqueda por similitud de imágenes (Sección 3.1), tanto las basadas en descriptores locales como en neuronales. La imagen de marca es un caso particular de imagen y tiene sus propias características, por lo que se diferencia su estado del arte, que se revisa en detalle en la Sección 3.2.

3.1. Búsqueda de imágenes similares

Según Jégou et al. (2008), el término imágenes similares se refiere a las representaciones de los mismos tipos de objetos o escenas que se aprecian bajo diferentes

condiciones. La búsqueda automática de similitud sobre un conjunto de imágenes se realiza estudiando elementos como la forma, color, textura o ubicación espacial de su contenido. Esta tarea ha supuesto durante largo tiempo un área de investigación activa que aún hoy sigue centrando gran interés, ya que resulta de especial utilidad cuando se trata de buscar imágenes similares en grandes bases de datos (Jégou et al., 2008).

El funcionamiento general de los sistemas CBIR suele tener dos fases: en primer lugar se extraen las características que definen las imágenes y, a continuación, estas se utilizan en funciones de similitud o *matching* para comparar las imágenes. Como veremos, dichas características se extraen y se representan usando estructuras de datos numéricas.

Desde principios de los años 2000 hasta 2015 se usaron mayoritariamente sistemas de reconocimiento de imagen basados en descriptores invariantes locales (Mikolajczyk and Schmid, 2004; Lowe, 2004). Para extraer las características que definen una imagen (o una región de interés de una imagen) estos sistemas detectan en una primera fase una serie de puntos distintivos de interés. Estos puntos deben ir acompañados de un descriptor que codifica la información local alrededor del mismo. La idea es que los puntos de interés que corresponden a la misma zona de la imagen (con objetos o elementos similares) deben tener descriptores similares, por lo que el cálculo de similitud de puntos de interés puede ser tratado como el cálculo de distancias entre ellos.

Estos descriptores idealmente deben ser invariantes a cambios en la imagen, como cambios de escala, rotaciones, presencia de ruido, cambios de iluminación u oclusiones, motivo por el cual reciben el nombre de descriptores invariantes locales. En la literatura relacionada podemos encontrar multitud de propuestas para el cálculo de descriptores de características locales, los cuales los describiremos en detalle en la Sección 3.1.1. Sin embargo, estas técnicas tienen limitaciones, ya que requieren de la intervención humana para diseñar cómo se extraen las características relevantes que son adecuadas para ese tipo de imagen y, a la vez, para descartar las poco importantes.

Desde 2015 la mayoría de propuestas para la búsqueda de imágenes similares se han centrado en el uso de aprendizaje profundo y, en concreto, en las CNN. A diferencia de los métodos clásicos, estas redes neuronales aprenden durante el entrenamiento a extraer por sí mismas (en base a ejemplos) las características más relevantes de forma específica para la tarea propuesta.

Aunque las CNN se entrenan normalmente para tareas de clasificación y detección, también pueden emplearse para la búsqueda de imágenes similares usándolas como extractores de características. Una red ya entrenada para reconocer ciertas clases (como, por ejemplo, perros y gatos) puede utilizarse como extractor de características para otro problema distinto (como podrían ser vehículos). Una de las técnicas más usadas para realizar este proceso es el *transfer learning* o transferencia de aprendizaje, el cual consiste en pasar una imagen como entrada a la red y extraer los mapas de características de una de sus últimas capas. Estos vectores numéricos, que se conocen como descriptores neuronales (o *neural codes*), pueden emplearse para compararlos (usando cualquier función de distancia, como la distancia Euclídea) con los de otras imágenes con el objetivo de realizar búsquedas por similitud. Asimismo, también pueden usarse para entrenar otro clasificador (como SVM) de forma supervisada con clases de otra tarea distinta.

En esta sección vamos a hacer un recorrido por los métodos de extracción de características para reconocimiento de imágenes. Empezaremos por los descriptores locales, para revisar a continuación los trabajos más recientes basados en técnicas de aprendizaje profundo basadas en arquitecturas como CNN o AE.

3.1.1. Descriptores de características locales

En la literatura pueden encontrarse multitud de propuestas para extraer descriptores locales, entre los que destacan SIFT (*Scale-invariant feature transform*) (Lowe, 2004) y SURF (*Speeded up robust features*) (Bay et al., 2008) por ser ampliamente usados. Para que sean robustas, estas características deben ser invariantes a transformaciones de la imagen, como su escala o rotación. Idealmente también deberían ser robustas ante cambios de iluminación, ruido, oclusión de parte del objeto en la imagen o pequeños cambios del punto de vista.

Los descriptores SIFT calculan puntos de interés de la imagen buscando zonas con valores máximos locales mediante una pirámide de diferencias gaussianas calculada para diferentes escalas de la imagen. Para cada uno de los puntos detectados se asigna un descriptor que incluye, entre otros, información sobre la orientación, calculada usando la magnitud y dirección del gradiente del entorno de vecindad del punto (Lowe, 2004). Este tipo de descriptores resultan invariantes a escala y rotación, y parcialmente invariantes a cambios de iluminación y del punto de vista.

En su versión original se calculaban sobre imágenes en escala de grises y se almacenaban como un vector de características de dimensión 128. Una vez realizado este proceso, permiten buscar similitudes entre puntos de interés calculando sus distancias entre vectores de dimensión 128. Un valor pequeño de distancia entre dos descriptores denota que los dos puntos de interés son coincidentes con alta probabilidad.

SURF es una alternativa a SIFT que se impuso a este por ser bastante más robusto y más rápido de calcular. También es invariante a escala y rotación. Se basa en una aproximación de la matriz Hessiana para la detección de puntos de interés y un cálculo de las respuestas de wavelets Haar dentro del entorno de vecindad de estos puntos para conformar los descriptores (Thomee et al., 2010). Estos descriptores pueden ser vectores de distinto tamaño, aunque el más habitual es 64.

En algunos casos, los descriptores locales se agrupan en una bolsa de características (*Bag of Features* o BOF), también llamadas palabras, para mejorar el rendimiento de la búsqueda por similitud. Uno de estos enfoques es TOP-SURF (Thomee et al., 2010), el cual calcula las características SURF, las agrupa en un histograma según su similitud con un diccionario pre-calculado de "palabras visuales" (*visual words*) y las pondera usando la técnica tf-idf (*term frequency–inverse document frequency*) (Salton and McGill, 1983).

Una alternativa a estos descriptores son los descriptores binarios, que codifican la información mediante cadenas binarias mejorando la eficiencia de la búsqueda mediante distancia Hamming, mucho más rápida que la Euclídea. Ejemplos de estos descriptores son FAST (Rosten and Drummond, 2006), BRIEF (Calonder et al., 2010) y ORB (Rublee et al., 2011). ORB es un descriptor binario muy rápido basado en FAST y BRIEF, que además es invariante a rotaciones y tolerante al ruido, aunque no invariante a escala. Existen ligeras diferencias entre estos descriptores binarios, como por ejemplo que BRIEF no es invariante a rotación y FAST no es invariante al ruido. FREAK (Alahi et al., 2012) es otro descriptor binario que está inspirado en el sistema de visión humano y más precisamente la retina. Sin embargo, para tareas de búsqueda por similitud los descriptores binarios suelen obtener peores resultados que los descriptores que describen los puntos de interés mediante vectores de números reales, tales como SIFT o SURF.

Otro tipo de características utilizadas para imagen son los descriptores globales, como por ejemplo los histogramas de color (van de Sande et al., 2010; Juang et al., 2009; Lei et al., 1999) y textura, como LBP (*Local Binary Pattern*) (Manjunath and Ma, 1996; Ojala et al., 2000), que en algunos conjuntos de datos han demostrado

también ser una elección adecuada. Suelen ser más rápidos que los descriptores locales, aunque habitualmente ofrecen un porcentaje de acierto inferior en tareas de reconocimiento de imagen.

También podemos encontrar sistemas que combinan diferentes características, como el color con descriptores locales (Fernando et al., 2012), o la forma con color y textura (Banerji et al., 2013). Cuando se combinan distintos tipos de características se debe ponderar la influencia de las mismas.

Es importante resaltar que los modelos basados en descriptores de características locales no proporcionan información geométrica, esto es, la posición del punto de interés en el mapa de píxeles de la imagen. Por tanto, para mejorar el resultado del sistema se puede añadir una etapa de verificación geométrica eficiente para reordenar los resultados devueltos por el modelo y mejorar así la calidad de la búsqueda.

Durante mi etapa de investigación he estudiado la aplicación de algunos de estos descriptores locales sobre el sistema MirBot en los inicios de su desarrollo. Este trabajo fue publicado en 2013 (Pertusa et al., 2013), aunque el sistema evolucionó con el tiempo hasta la versión publicada en 2018 (Pertusa et al., 2018), donde se utilizan descriptores neuronales. La versión inicial utilizaba descriptores locales TOP-SURF e histogramas de color y se realizaron pruebas para incorporar una fase de verificación geométrica. Todos estos trabajos se detallan en el Capítulo 4.

3.1.1.1. Verificación geométrica

La verificación geométrica consiste en comprobar la concordancia espacial entre puntos de interés de dos imágenes. En la Figura 3.1 se puede ver un ejemplo de este proceso ilustrando la correspondencia obtenida entre puntos de interés de dos muestras de la base de datos MirBot (Pertusa et al., 2018). Como se puede observar, dos vistas de una figura pueden relacionarse mediante una homografía que se puede calcular a partir de las correspondencias de las regiones prominentes entre las dos imágenes.

El problema de la verificación geométrica ha sido bien estudiado por Hartley and Zisserman (2003), tanto respecto a las transformaciones necesarias como a su estimación. Normalmente, en los algoritmos de estimación de datos visuales deben considerarse dos tipos de errores: errores en la posición y forma de una característica

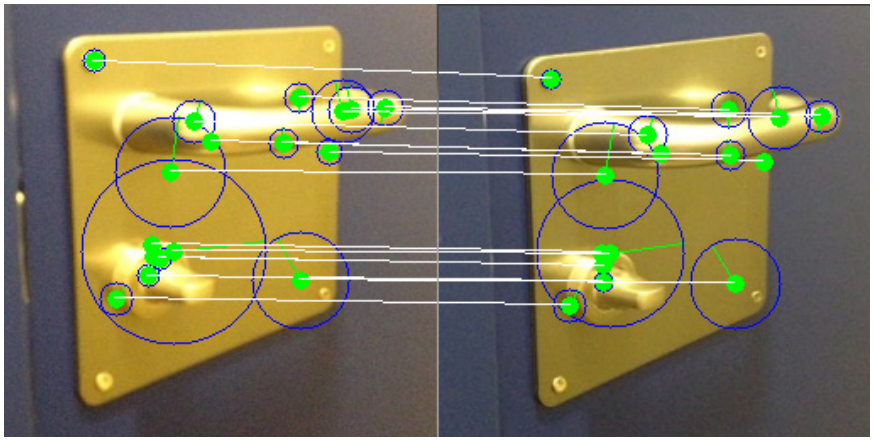


Figura 3.1.: *Ejemplo de correspondencia entre puntos de interés de dos imágenes. El punto de interés se representa en color verde y su escala en azul.*

detectada, así como errores debidos a valores atípicos de características que no encajan o no corresponden con el modelo seleccionado y que pueden ser provocados por un fallo del detector, oclusiones, etc.

El procedimiento de verificación geométrica RANSAC (*Random Sample Consensus*) (Fischler and Bolles, 1981) ha sido ampliamente usado para estimar la transformación entre dos imágenes. Este algoritmo se planteó como una solución que propone generar hipótesis de transformación usando un número mínimo de correspondencias para luego evaluar cada hipótesis en función del número de *inliers* (datos que pueden ser explicados por el modelo hipotético) entre todas las características bajo esa hipótesis. La verificación espacial con RANSAC se ha utilizado en varios trabajos, como los de Philbin et al. (2007); Chum et al. (2004). Sin embargo, su coste computacional es muy alto, lo que ha llevado a proponer otras alternativas en la literatura. Por ejemplo, Jégou et al. (2009) introdujo una alternativa de verificación espacial "débil" y Tsai et al. (2010) propuso otra técnica rápida para añadir información espacial con BOF.

Lazebnik et al. (2006) propusieron otro método de reconocimiento que calcula la correspondencia geométrica aproximada a escala global utilizando una técnica de aproximación eficiente adaptada del esquema de combinación de pirámides de Grauman y Darrell (Grauman and Darrell, 2005). Este método, llamado *Spatial Pyramid Matching* (SPM), implica la subdivisión repetida de la imagen y el cálculo de histogramas de características locales a resoluciones cada vez más pequeñas. Como se explica en el artículo, esta operación simple mejora el rendimiento sobre una representación básica de BOF, aunque sus resultados pueden variar en función de la categoría de las imágenes, su similitud y de su composición.

Durante la tesis he estudiado estas técnicas y he desarrollado un algoritmo de verificación geométrica simple y eficiente llamado SIIP (*Segment Intersection of Interest Points*) que mejora los resultados de aproximaciones previas (Bernabeu et al., 2016). En concreto, el método se compara con RANSAC, obteniendo mejores resultados y con un menor coste computacional. Como se verá en detalle en la Sección 4.3.3, la técnica desarrollada se basa en la comparación de las intersecciones formadas entre los segmentos construidos por pares de puntos de interés.

3.1.2. Descriptores Neuronales

La aparición de las técnicas de aprendizaje profundo y, en especial, de las CNN, ha supuesto un gran avance en las tareas de reconocimiento de imágenes. Estas redes, tal y como se ha descrito anteriormente, pueden usarse para extraer una representación numérica de la imagen a partir de la codificación obtenida en una de sus últimas capas. Estas representaciones, a las que podemos llamar descriptores neuronales, son las que nos permiten realizar búsquedas por similitud.

En la literatura podemos encontrar multitud de arquitecturas CNN entre las que destacamos las siguientes, bien por su importancia en la evolución de estas técnicas o bien por haberse usado en el desarrollo de esta tesis:

1. AlexNet (Krizhevsky et al., 2012) es una CNN con 8 capas considerada la primera arquitectura moderna de este tipo. Mejoró los resultados obtenidos hasta el momento en la competición ImageNet 2012 (Russakovsky et al., 2015), que consiste en clasificar cerca de 1,3 millones de imágenes categorizadas en 1000 clases.

2. GoogLeNet (Szegedy et al., 2015) es una red neuronal desarrollada por Google que consiguió los mejores resultados en ImageNet 2014. Está compuesta por 22 capas, pero usa 12 veces menos parámetros que AlexNet, y está basada en una concatenación de módulos de tipo Inception, los cuales procesan la misma información mediante una estructura paralela de filtros de distinto tamaño cuyos resultados posteriormente se concatenan.

3. VGG-16 y VGG-19 (Simonyan and Zisserman, 2015) se desarrollaron en la Universidad de Oxford. VGG-16 tiene 13 capas convolucionales y 3 completamente conectadas, mientras VGG-19 tiene 16 capas convolucionales y 3 completamen-

te conectadas. Ambas utilizan *dropout*, *maxpooling*, y funciones de activación ReLU.

4. Inception v2 (Szegedy et al., 2015) es una mejora de la arquitectura GoogLeNet en la que destaca la incorporación de capas *Batch Normalization* (Ioffe and Szegedy, 2015).

5. Inception v3 (Szegedy et al., 2015) es una mejora sobre la versión anterior consistente en 6 capas convolucionales seguidas de 3 módulos Inception y una última capa completamente conectada. En esta arquitectura se añade *Batch Normalization* tanto a las capas convolucionales como a las completamente conectadas.

6. REsNet (He et al., 2016) incluye funciones residuales con referencia a las capas anteriores. Esta técnica permite entrenar redes con un gran número de capas. En el trabajo citado se evalúan hasta 50, 101 y 152 capas.

7. Xception (Chollet, 2017) tiene 36 capas convolucionales con una versión rediseñada de los módulos Inception para realizar lo que denominan convolución separable en profundidad (*depthwise separable convolutions*). Esta arquitectura supera los resultados de Inception utilizando el mismo número de parámetros.

8. SENet (Hu et al., 2018) propone una unidad de bloques SE (*Squeeze-and-Excitation*) que permite reequilibrar adaptativamente cada nivel de capas mediante el modelado de interdependencias entre canales. Esta red ganó la competición ImageNet 2017, reduciendo significativamente el error al 2,25 % y logrando una mejora relativa de cerca del 25 % con respecto al mejor modelo del año anterior.

A partir de cualquiera de estas arquitecturas se puede aplicar la técnica de *transfer learning* explicada previamente, inicializando la red con los pesos pre-entrenados con algún conjunto de datos, como por ejemplo ImageNet (Russakovsky et al., 2015), y posteriormente procesar nuestras imágenes para extraer los descriptores neuronales de una de sus últimas capas. Por último, en base a estos descriptores, se puede usar algún método para realizar la comparación entre imágenes.

Una práctica también habitual de *transfer learning* es realizar un proceso de *fine-tune* (ajuste fino) de los pesos para mejorar los resultados con nuestro conjunto de datos (Gallego et al., 2018b). Esta operación consiste en, una vez inicializada la red con unos pesos pre-entrenados, sustituir la última capa de la misma por una adecuada

al número de clases (o a la tarea) de nuestro conjunto de datos y entrenar los pesos de la red para ajustarlos ligeramente a nuestros datos. Este proceso, que suele implicar pocas épocas de entrenamiento al partir de una buena inicialización, suele conseguir mejores resultados ya que ajusta los descriptores a una representación más adecuada a la tarea a resolver Gallego et al. (2018a, 2020).

En esta tesis se describen dos trabajos relacionados con el uso de CNN para la búsqueda de imágenes similares. El primero, detallado en el Capítulo 4, se aplica sobre la base de datos de MirBot y realiza la búsqueda de similitudes usando kNN con los descriptores neuronales. El segundo trabajo, que se detalla en el Capítulo 5, se centra en la búsqueda por similitud de logotipos e imágenes de marca. En este caso, se combinan los descriptores obtenidos de varias CNN especializadas en diferentes características junto con la codificación devuelta por un Auto-Encoder (Masci et al., 2011; Turchenko et al., 2017) para realizar una búsqueda ajustable al criterio del usuario.

3.2. Recuperación de imagen de marca

Los sistemas TIR (*Trademark Image Retrieval*) se centran en la detección y clasificación de imagen de marca. Este problema se considera un caso especial de reconocimiento de imágenes, ya que las imágenes de marca tienen sus propias características porque están creadas por un humano y son diseñadas para representar una marca o empresa que requiere ser fácilmente identificable y recordada, por lo que deben tener impacto visual.

Las imágenes de marca están formadas por una serie de elementos que pueden representar objetos conocidos, pasando por distintos grados de simplificación hasta llegar a formas abstractas, y pueden contener o no texto. Además de la estructura de los elementos que la componen y su organización, se debe considerar la interpretación semántica para determinar los objetos reconocibles que contiene, lo cual es una tarea muy compleja, tal como demuestran los estudios realizados sobre cómo los humanos perciben e interpretan las imágenes (Schietse et al., 2007). Por ejemplo, si buscamos logos que contengan un zorro podemos encontrar imágenes con diseños completamente diferentes, como se puede ver en la Figura 3.2. Los colores también representan un problema, ya que estos no reflejan la realidad sino que se usan para crear impacto visual, además de que podemos encontrar versiones en blanco y negro o en escala de

Figura 3.2.: *Ejemplo de logos con zorros en el conjunto de datos EUTM, clasificados con el código Vienna 03.01.08 (Perros, lobos, zorros). Se puede ver una representación muy diferente de los diseños, que se han simplificado y estilizado en diferentes grados.*

grises. También está el problema del formato de representación y la resolución, pues a menudo encontramos los logos en imágenes de baja resolución.

La detección y reconocimiento de logos se ha convertido en una tarea interesante a nivel industrial debido a la necesidad que tienen las empresas para detectar el uso de sus logos en imágenes, páginas web, verificar su visibilidad en eventos deportivos (Köstinger et al., 2013) y detectar plagios o uso de logotipos sin autorización. Sin embargo, en los últimos años se han publicado relativamente pocos artículos sobre reconocimiento de logos (Iandola et al., 2015) y, además, la mayoría de los conjuntos de datos utilizados no están públicamente disponibles (Ghosh and Parekh, 2015; Rusiñol et al., 2011). Además de revisar las metodologías existentes, durante esta tesis también se han analizado las bases de datos recientes de imagen de marca que están disponibles para investigación, las cuales se relacionan en la siguiente sección.

3.2.1. Conjuntos de datos de logos

La detección y el reconocimiento de logos es una tarea que plantea una serie de desafíos. Uno de ellos es que para obtener buenos resultados de entrenamiento usando métodos de aprendizaje automático se necesitan grandes bases de datos de logos. La mayoría de trabajos existentes utilizan bancos de imágenes que contienen logos, pero, o bien no contemplan bases de datos compuestas exclusivamente por logos o bien no están disponibles públicamente. No obstante, recientemente se han publicado algunos conjuntos de datos de logos que además son de acceso público.

La Tabla 3.1 muestra una lista de las bases de datos habitualmente utilizadas en la literatura que están relacionadas con la detección y clasificación de logos. En la tabla se especifica el nombre de la base de datos, el número de clases e imágenes que la componen y su tipo: imágenes en general que contienen logos, denotadas como

Tabla 3.1.: *Bases de datos en la literatura relacionadas con la detección y clasificación de logos.*

Nombre	Tipo de imágenes	Nº clases	Nº imágenes
FlickrLogos-27 (Kalantidis et al., 2011)	Logos en imagen	27	1.080
FlickrLogos-32 (Romberg et al., 2011)	Logos en imagen	32	8.240
MICC-Logos (Sahbi et al., 2013)	Logos en imagen	13	720
BelgaLogos dataset (Joly and Buisson, 2009)	Logos en imagen	26	10.000
Logos in the Wild (Tüzkö et al., 2017)	Logos en imagen	871 (marcas)	11.054
TraidMarks (Gu, 2014)	Solo logos	–	999
METU dataset (Tursun et al., 2017)	Solo logos	35	920.000
LLD (Sage et al., 2018)	Solo logos	–	600.000
NPU-TM (Lan et al., 2017)	Solo logos	317	7.139
EUTM (EUTM)	Solo logos	–	1.270.000
USPTO (USPTO)	Solo logos	–	millones

"Logos en imagen", o conjuntos de datos compuestos exclusivamente por logos ("Solo logos"). En esta tesis nos centramos en estos últimos, ya que son el objeto de nuestra propuesta. La Figura 3.2 que hemos visto previamente sería un ejemplo del caso "Solo logos", mientras que en la Figura 3.3 se puede ver un ejemplo de "Logos en imagen", con imágenes generales que contienen logos y que se usan habitualmente para la tarea de detección y reconocimiento de logos.

Entre los conjuntos de datos compuestos exclusivamente por logos, METU (Tursun et al., 2017) es probablemente el más usado en la literatura. Contiene 920k imágenes no etiquetadas de unas 410.000 marcas aproximadamente y un subconjunto de consulta con 417 imágenes etiquetadas, que se organiza en 35 clases con 10-15 imágenes cada una, en la que los logos del mismo grupo son similares entre sí. Dentro de este tipo de conjuntos también cabe destacar LLD (*Large Logo Dataset*) de Sage et al. (2018), con 600k logos descargados de Internet, *TradeMarks Image Database* (TraidMarks) de Gu (2014) con 999 imágenes de marca en blanco y negro proporcionadas por el Centro de Investigación Myron Flickner IBM-Almaden, y la base de datos NPU-TM de (Lan et al., 2017) que incluye 317 grupos de marcas comerciales similares, donde cada grupo contiene al menos dos marcas.

Como se ha indicado, entre las bases de datos publicadas, la más utilizada es METU debido al gran número de imágenes que contiene. Sin embargo, estas imágenes están sin etiquetar, excepto el subconjunto de query, que solo tiene anotadas las marcas a las que pertenecen y sin llegar a profundizar en más detalle. Para desarrollar un sistema TIR completo como el que se plantea en los objetivos de esta tesis, son necesarias bases de datos con información sobre el color, la forma, los objetos representados, etc. con los que entrenar y evaluar cuantitativamente la propuesta.

Figura 3.3.: *Ejemplo de imágenes denotadas como "Logos en imagen", obtenidas de la base de datos MICC-Logos.*

Para organizar toda esta información en una base de datos, una de las opciones es utilizar la clasificación de Viena[1]. Esta ontología, establecida por la Organización Mundial de la Propiedad Intelectual (WIPO, *World Intellectual Property Organization*) en el Acuerdo de Viena de 1973, es la más usada en las oficinas de patentes y marcas de todo el mundo para asignar códigos que describen el contenido de un logo.

Estos códigos se organizan en un sistema jerárquico de tres niveles con el que se etiquetan los elementos siguiendo el patrón "`xx.yy.zz`", donde `xx` representa el primer nivel (con 29 posibles categorías), `yy` el segundo nivel (con decenas de posibles opciones) y `zz` el tercer nivel (con cientos de opciones). Por ejemplo, el código 5.5.1 representa la categoría "rosas", la cual pertenece a la subcategía 5.5 "flores y brotes", que a su vez pertenece a 5 "plantas". En la Tabla 3.2 se muestran algunos ejemplos de códigos de Viena.

Actualmente podemos encontrar dos grandes bases de datos etiquetadas siguiendo la codificación Viena: USPTO (*United States Patent and Trademark Office*) y EUTM (*European Union Trademark*). Esta última es la base de datos de marcas de la Oficina de Propiedad Intelectual de la Unión Europea, conocida como EUIPO (*European Union Intellectual Property Office*), y encargada de gestionar las patentes y marcas de la Unión Europea (en 2020 tenía registradas más de 1,27 millones de marcas[2]). La base de datos contiene imágenes de marcas con metadatos asociados etiquetados mediante los códigos Viena con información sobre colores, formas, texto o diseños figurativos. También ofrece información sobre los sectores a los que pertenece la marca utilizando la codificación de Niza[3] que divide Bienes y Servicios en 45 sub-categorías (ver Apéndice C). Por ejemplo, "Bienes" incluye productos químicos, medicamentos,

[1] https://www.wipo.int/classifications/vienna/es/
[2] https://euipo.europa.eu/tunnel-web/secure/webdav/guest/document_library/contentPdfs/news/EUIPO_TM_Focus_Report_2010-2019_Evolution_en.pdf
[3] https://euipo.europa.eu/ohimportal/en/nice-classification

Tabla 3.2.: *Extracto de clasificación de Viena establecida por la WIPO en 1973*

CÓDIGO	**Descripción**
1.1	STARS, COMETS
1.1.1	Stars
1.1.15	Comets, stars with tail
1.1.17	Compass cards
1.3	SUN
1.3.1	Sun rising or setting
1.3.2	Other representations of the sun
1.3.19	Several suns
	...
26.1	CIRCLES, ELLIPSES
26.1.1	Circles
26.1.2	Ellipses
26.1.4	Two circles, two ellipses, one inside the other
26.1.5	More than two circles or ellipses, inside one another, spirals
	...
26.3	TRIANGLES, LINES FORMING AN ANGLE
26.3.1	One triangle
26.3.2	Two triangles, one inside the other
26.3.3	More than two triangles, inside one another
	...
29.1	COLOURS
29.1.1	Red, pink, orange
29.1.2	Yellow, gold
29.1.3	Green
29.1.4	Blue
	...

metales, materiales, máquinas, herramientas, vehículos, instrumentos, etc., mientras que los "Servicios" incluyen publicidad, seguros, telecomunicaciones, transporte y educación, entre otros.

La clasificación Viena, aún siendo la más utilizada, también tiene algunos inconvenientes inherentes a los sistemas de etiquetado manual: 1) consume mucho tiempo, ya que hay que seleccionar entre cientos de posibles opciones, 2) se trata de un etiquetado subjetivo en el que puede influir la formación, factores sociales o culturales, 3) debido a esta subjetividad y a la cantidad de etiquetas es relativamente sencillo cometer errores (lo cual hace que se requiera de operadores capacitados y entrenados) y 4) los códigos no siempre permiten describir adecuadamente los contenidos de la imagen y en ocasiones resultan un tanto ambiguos.

En esta tesis se ha analizado esta codificación y se ha establecido una relación con las topologías de imagen de marca adoptadas en el mundo del diseño, las cuales se

describen en la Sección 5.1.3. Además, con el objetivo de que el sistema desarrollado permita establecer criterios de búsqueda ajustados a estas topologías y que, por tanto, sean de utilidad para los potenciales usuarios del sistema, se ha propuesto un etiquetado que intenta eliminar algunos de los problemas mencionados mediante la agrupación de códigos, como veremos en el Capítulo 5.

3.2.2. Métodos TIR

En general, los sistemas TIR pueden enfocar el problema de la recuperación de información de marca de dos formas: extraer y comparar características de imágenes tomadas como un conjunto, o considerar la imagen como una suma de componentes discretos, que bien se identifican de forma individual o bien se combinan para determinar la similitud de la imagen calculando las formas a partir de la similitud de sus componentes. Entre las propuestas basadas en este segundo enfoque encontramos STAR (*System for Trademark Archival and Registration*) (Wu, 1996) y su posterior versión ARTISAN (*Automatic Retrieval of Trademark Images by Shape ANalysis*), desarrollado en 1998 por Eakins et al. (1998) de la Universidad de Northumbria en Newcastle.

Este método permite realizar la comparación de similitud en tres niveles: imágenes completas, familias de componentes, y componentes individuales de imágenes. Se basa en las leyes de Gestalt para inferir reglas que permitan agrupar componentes individuales de la imagen en familias de componentes. Estas leyes describen cómo los humanos percibimos las imágenes: no como la suma de sus partes, sino que percibimos los elementos por su proximidad, por su similitud (color, tamaño o forma), continuidad de líneas o cierres (el cerebro tiende a completar patrones reconocibles), o como grupos en lugar de como objetos aislados.

No obstante, la mayoría de los sistemas TIR se suelen basar en el primer enfoque comentado, la extracción y comparación de características que definen las imágenes para luego utilizar funciones de similitud. Algunas de las características usadas en la literatura son las formas (Kato, 1992; Ghosh and Parekh, 2015; Rusiñol and Lladós, 2010), curvatura y distancia al centroide (Wei et al., 2009), modelos basados en BOF para agrupar características como SIFT (Köstinger et al., 2013; Kalantidis et al., 2011; Sahbi et al., 2013), histogramas de color (Ghosh and Parekh, 2015), o combinaciones de descriptores (Rusiñol et al., 2011). Para la fase de búsqueda por similitud, los vectores de características suelen compararse utilizando métricas como la

distancia Euclídea o mediante métodos más complejos, como los basados en plantillas (Pornpanomchai et al., 2015).

Sin embargo, todas estas aproximaciones que se proponen en los sistemas TIR suelen obtener unos resultados bastante limitados, principalmente debido a la complejidad de la tarea. Hay que tener en cuenta que, tal como indican Schietse et al. (2007), un sistema adecuado para comparar imagen de marca debería cumplir las siguientes restricciones:

1. Se deben tener en cuenta todas las posibles interpretaciones de una imagen de marca.

2. Debería ser posible buscar en grandes conjuntos de imágenes con una velocidad aceptable.

3. Las imágenes muy similares a la imagen de la consulta en la base de datos siempre deberían encontrarse (tolerancia cero).

4. Las imágenes de marcas registradas deberían compararse en detalle (en cuanto a forma, contorno y estructura) teniendo en cuenta transformaciones como rotación, escalado, inversión y desenfoque.

Schietse et al. (2007) plantean que los principales retos a los que se enfrentan los sistemas automatizados actuales radican en el análisis inicial de la imagen. A menos que todas las características cruciales de las imágenes se hayan calculado y almacenado de manera efectiva, es poco probable que la comparación posterior identifique todas las similitudes relevantes. Un sistema ideal debería poder reconocer similitudes de forma, estructura y semántica, y ser capaz de manejar texto (posiblemente estilizado). Hay que tener en cuenta que la mayoría de los estudios se centran en la estructura y/o apariencia de los elementos que componen la imagen. Sin embargo, para cumplir con el primer punto de la lista de restricciones anterior, es muy importante analizar la semántica de los logos, es decir, detectar e identificar los objetos reconocibles que contiene. En este sentido, los sistemas clásicos mencionados que están basados en la comparación de descriptores tradicionales se ven limitados a la hora de realizar este tipo de análisis.

Exceptuando algún trabajo encontrado en la literatura que incorpora el análisis de la semántica de logos, como los de Perez et al. (2018) o Rusiñol et al. (2011), la mayoría de los métodos TIR utilizan conjuntos de datos con logos etiquetados exclusivamente

por marca, por ejemplo METU de Tursun et al. (2017), asumiendo que las imágenes de una misma marca deberían ser similares. Sin embargo, una marca suele tener diferentes versiones de logos a lo largo del tiempo con cambios en fondo, color, textura o forma. Por ejemplo, como menciona Iandola et al. (2015), Disney ha cambiado su logotipo más de 30 veces desde 1988 a 2015 y, de hecho, ha seguido cambiando hasta la actualidad[4]. Por tanto, se debe tener en cuenta que a veces hay fuertes cambios en los diseños de la misma marca, haciéndolos muy diferentes en apariencia, lo que dificulta aun más la detección de similitud entre logos pertenecientes a la misma marca.

Métodos de aprendizaje profundo para TIR

La aparición del aprendizaje profundo ha revolucionado las tareas de reconocimiento de imágenes y, por consiguiente, también la de logos. Como ya se ha desarrollado previamente, usando esta tecnología se puede inferir una jerarquía de características representativas de un tipo de imagen para la que, a alto nivel, genera descriptores que identifican los objetos que contiene (LeCun et al., 2015). Es por este motivo que al utilizar los descriptores neuronales se consigue mejorar notablemente los resultados de las técnicas previas empleadas para el reconocimiento de logos, como se puede ver en la comparación realizada por Iandola et al. (2015).

Las propuestas más sencillas basadas en aprendizaje profundo simplemente emplean redes neuronales pre-entrenadas para la extracción de características, como es el caso de Chiam (2015). Otras propuestas más elaboradas se basan en la combinación de descriptores, como en Perez et al. (2018) que combina las características extraídas de dos redes VGG-19, una entrenada utilizando USPTO (usando la codificación de Viena) y otra con imágenes obtenidas de Internet y organizada en clases de percepción por un experto. Este trabajo utiliza la base de datos METU para comparar los resultados. También encontramos propuestas más metodológicas, como el trabajo de Lan et al. (2017) que aplica Uniform LBP para extraer características de cada capa convolucional de una CNN pre-entrenada. Para realizar los experimentos utilizan la base de datos NPU-TM, que se presenta por primera vez en ese artículo y, además, usan METU para comparar resultados. Xia et al. (2019) también utilizan NPU-TM y METU para presentar un método llamado *Transform-invariant Deep hashing* en el que se propone una estructura unificada para aprender códigos binarios de imagen de marca, mejorando los métodos de *deep hashing* existentes.

[4] https://www.closinglogos.com/page/Logo_Variations_-_Walt_Disney_Pictures

En trabajos más recientes podemos encontrar distintas aproximaciones para trabajar con imagen de marca. Por ejemplo, Sage et al. (2018) emplean redes adversarias generativas o GAN (*Generative Adversarial Network*) para la síntesis de logos con el objetivo de facilitar la tarea de diseño gráfico. Wang et al. (2019) emplean una red Faster R-CNN (Ren et al., 2015) de detección de objetos, de la que se extraen tanto descriptores globales como locales usando para estos últimos las regiones propuestas por la red RPN (*Region Proposal Networks*) de la propia arquitectura. Las características globales se utilizan para buscar inicialmente entre todas las imágenes de la base de datos y, a continuación, las características de la región se utilizan para la reordenación. Tao et al. (2019) proponen un modelo llamado *Spatial Feature Collaborative Network* (SFCN) que también extrae características globales y locales fusionando en este caso dos arquitecturas: AlexNet y Faster R-CNN. Los autores realizan experimentos exhaustivos usando un conjunto de datos específico de marcas comerciales. Otro ejemplo reciente es la propuesta de Cao et al. (2021) en la que se introduce un mecanismo de *lightweight attention* para aprender la zona de la imagen con mayor peso para la extracción de características. Este método puede obtener de forma no supervisada buenas representaciones y mejorar el rendimiento en la recuperación de marcas.

Todos estos trabajos revisados se centran únicamente en la recuperación de logotipos similares, encontrando solo en algún caso propuestas para clasificarlos por sus características. Si bien la búsqueda de logotipos similares es una tarea muy útil actualmente, la clasificación automática también lo es, ya que podría asistir en el proceso de etiquetado, ayudar a diseñadores, etc. A este respecto hemos de tener en cuenta que un logotipo puede tener más de una etiqueta simultánea para una misma categoría, por ejemplo, varios colores, varias formas, etc. Se trata por tanto de un problema de clasificación multi-etiqueta (*multi-label* o MLC). Esta tarea es diferente de la clasificación multi-clase utilizada tradicionalmente para logos, en la cual las etiquetas se tratan como variables independientes y no se permite asignar a una misma imagen más de una etiqueta (confiando en que son mutuamente excluyentes), lo cual resulta claramente sub-óptimo con respecto a MLC puesto que las dependencias entre clases no pueden aprovecharse.

Podemos encontrar aplicaciones de MLC en tareas tan diversas como la categorización de texto (Dong et al., 2020), de música (Trohidis, 2011), o la clasificación semántica de escenas (Boutell et al., 2004), entre otras. En Zhang and Zhou (2014) puede consultarse una revisión de algoritmos de aprendizaje multi-etiqueta. Sin embargo, hasta la fecha de esta tesis no se ha aplicado MLC a problemas de clasificación

de logos. Por tanto, dadas las características particulares de esta tarea, consideramos de especial interés desarrollar sistemas que permitan la clasificación y búsqueda multi-etiqueta de logos en base a diferentes criterios que puedan ser establecidos por el usuario, el cual será uno de los objetivos principales a abordar en el Capítulo 5.

CAPÍTULO 4

Búsqueda de imágenes por similitud

4.1. Introducción

En este capítulo se revisan las contribuciones realizadas en la tesis dentro del marco de la búsqueda de imágenes por similitud. Estas contribuciones tienen en común la base de datos utilizada, la cual ha sido generada mediante la aplicación MirBot, un sistema de reconocimiento de imágenes desarrollado como una aplicación colaborativa para dispositivos móviles.

Durante mi etapa de investigación he colaborado en la implementación de este sistema, lo cual me ha permitido estudiar diferentes técnicas existentes hasta el momento para detección de similitud entre imágenes y además desarrollar algunas técnicas nuevas para tareas relacionadas.

MirBot es una app disponible gratuitamente en las tiendas Apple y Google Play cuya primera versión data del 2013 (Pertusa et al., 2013). La aplicación está diseñada como un juego en el que los usuarios tienen que enseñar a un robot a reconocer el mundo mediante la toma de fotografías y la validación de los objetos detectados. Además incluye herramientas de gamificación para incentivar a que los usuarios compitan e intenten mejorar su robot.

En la Figura 4.1 se puede ver un esquema del funcionamiento de la aplicación. Inicialmente el usuario toma una fotografía y selecciona la región de interés (ROI, *Region of Interest*) con el objeto a reconocer. Esta información se envía a un servidor junto a una serie de metadatos del dispositivo móvil (como por ejemplo GPS, acelerómetro, giroscopio, etc.). La imagen se clasifica y se devuelve el resultado al usuario para su validación. Si la clase propuesta no es correcta el usuario puede corregirla usando

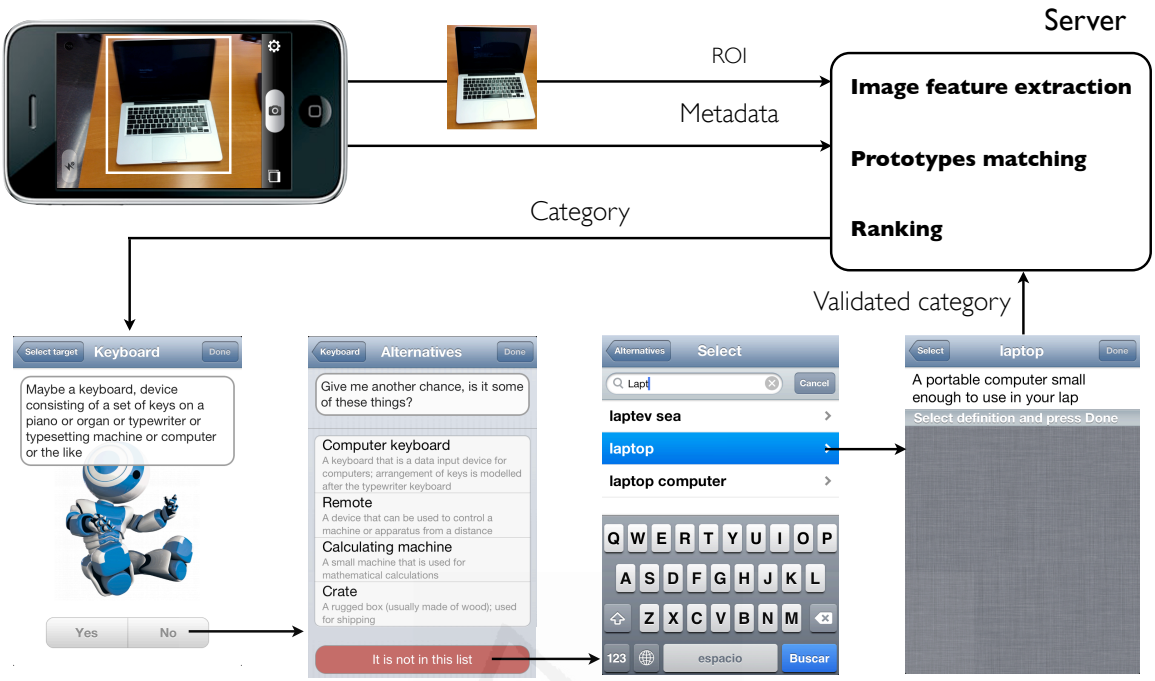


Figura 4.1.: *Arquitectura de la aplicación MirBot.*

una lista de sugerencias o buscando directamente en un diccionario, para el que se ha utilizado WordNet (Oram, 2001). Finalmente, la instancia validada se agrega a la base de datos de MirBot, por lo que se trata de un conjunto de datos dinámico que crece continuamente gracias a los usuarios. Esto permite que sus resultados también vayan mejorando al ir aprendiendo a reconocer nuevos objetos y tener más ejemplos de las clases.

El uso de WordNet en MirBot permite evitar ambigüedades en las etiquetas gracias a su organización jerárquica. Los *synsets* de WordNet (conjuntos de sinónimos) son identificadores únicos para conceptos semánticos. Cada uno de ellos está vinculado a una definición aunque puede estar relacionado con diferentes palabras. MirBot utiliza los *synsets* como identificadores de clase en el conjunto de datos. En MirBot solo se pueden asignar objetos (no abstractos) de la jerarquía WordNet, incluidas las siguientes categorías raíz: animales (código lexicográfico número 5), alimentos/bebidas (13), plantas (20) y objetos, tanto naturales (6) como artificiales (17). WordNet considera que los objetos artificiales están hechos por el hombre, mientras que los objetos naturales no. Ambas categorías se fusionaron en MirBot para evitar confusiones entre los usuarios. Además de indicar la clase, los usuarios también pueden etiquetar sus

Figura 4.2.: *Imágenes de ejemplo de la base de datos MirBot.*

imágenes de forma personalizada (por ejemplo indicando "Mi perro Tobby") y administrarlas (verlas, eliminarlas, etc.) en la sección "Mis imágenes" de la aplicación.

Una característica de los datos en MirBot es que no están balanceados, ya que los objetos más comunes aparecen con más frecuencia (pues dependen de los envíos de los usuarios). Por ejemplo, en noviembre de 2021, observando las categorías principales de WordNet, se podía ver que la mayoría de las imágenes son objetos (25.272), seguidas de animales (6.396), alimentos y bebidas (1.470), y por último plantas (771). Otra característica es que evoluciona con el tiempo y el número de imágenes que contiene aumenta continuamente gracias a la interacción de los usuarios. Por ejemplo, las cifras anteriores las podríamos comparar con las de octubre de 2016, cuando contenía 18.685 objetos, 4.928 animales, 1.113 alimentos y bebidas y 546 plantas. En la Figura 4.2 se pueden ver algunas imágenes de ejemplo de la base de datos.

En definitiva, MirBot ha permitido recopilar una gran base de datos con imágenes de objetos clasificados que sigue creciendo a día de hoy gracias a sus usuarios. Además, una de sus principales aportaciones con respecto a otras bases de datos de objetos similares, es la inclusión de información usando geocodificación inversa y metadatos recopilados de los sensores de los teléfonos, que pueden usarse para restringir el espacio de búsqueda en un escenario multimodal. Por ejemplo, si un usuario hace una fotografía de un elefante, será más probable que esté en un zoológico que en una playa.

Además de almacenar todos los metadatos del dispositivo (versión, modelo, red, etc.), de sus sensores (acelerómetro, giroscopio, GPS, etc.) y de la propia cámara (datos EXIF), se utiliza la latitud y longitud obtenida para invertir la codificación geográfica en el servidor mediante Gisgraphy[1], que utiliza la base de datos geográfica GeoNames. Esto permite obtener más datos valiosos a considerar, como el tipo de localización, que puede ser, por ejemplo, parque, colegio, montaña, etc. El listado completo de tipos de localizaciones puede consultarse en Geonames.

La aproximación seguida en MirBot para realizar el reconocimiento de las imágenes consta de dos fases: en primer lugar se extraen las características de la imagen de la región de interés (ROI) proporcionada por el usuario y en segundo lugar se utiliza la técnica k-vecinos más cercanos (kNN) para buscar las imágenes más similares. Las primeras versiones de MirBot usaban como características histogramas de color y descriptores locales, como SIFT o TOP-SURF (ver Sección 4.2). Posteriormente evolucionó al uso de descriptores neuronales, como veremos en la Sección 4.4. En MirBot se han realizado diferentes experimentos que incluyen el uso de metadatos y de técnicas de verificación geométrica, lo que ha llevado al desarrollo de un nuevo algoritmo llamado SIIP (*Segment Intersection of Interest Points*) detallado en la Sección 4.3.3.

4.2. Descriptores locales

La primera versión de MirBot usaba descriptores locales TOP-SURF (Thomee et al., 2010) e histogramas de color para extraer características de las imágenes. Posteriormente, la búsqueda por similitud se realizaba usando kNN para devolver al usuario la clase de la imagen más similar.

Como se ha mencionado en el capítulo anterior, TOP-SURF extrae descriptores SURF y los agrupa en palabras visuales usando *Bag of Features*. En MirBot inicialmente se utilizó un diccionario genérico con 100.000 palabras visuales[2] como base para calcular los histogramas tf-idf. De cada uno de estos histogramas nos quedamos con las 100 palabras visuales más relevantes para una imagen dada a la hora de realizar la comparación.

[1] http://www.gisgraphy.com/
[2] http://press.liacs.nl/researchdownloads/topsurf/

Las características SURF no consideran el color y este puede ser importante para la identificación de ciertas clases (Jeong, 2001). Por esta razón, los descriptores TOP-SURF se complementaron con histogramas de color para mejorar los resultados. Se llevaron a cabo experimentos con los que evaluar diferentes espacios de color (RGB, HSV, CIE-LUV e YCrCb), y los mejores resultados se obtuvieron utilizando YCrCb, que ha demostrado ser un espacio relativamente robusto ante cambios de iluminación. Adicionalmente, el valor de color de cada píxel se ponderó utilizando una función gaussiana bidimensional para obtener un histograma de color ponderado. El objetivo de esta ponderación es dar menos relevancia a los colores que aparecen en los bordes de la región de interés y más peso a los centrales, que es donde probablemente se ubiquen los objetos a reconocer. La función gaussiana bidimensional utilizada se define como:

$$f(x,y) = Ae^{-\left(\frac{(x-x_o)^2}{2\sigma_x^2}+\frac{(y-y_o)^2}{2\sigma_y^2}\right)} \tag{4.1}$$

donde A es la altura del pico de la curva, x_o, y_o es la posición central del pico, y σ_x, σ_y define el ancho de la forma de la campana. En este trabajo se utilizaron los valores $A = 1$ para definir la función entre 0 y 1, se situó x_o, y_o en el centro de la imagen, y se estableció el ancho de la campana a un quinto del ancho y la altura de la imagen, respectivamente.

Para comparar los descriptores TOP-SURF de dos imágenes a y b, se utiliza la distancia del coseno normalizada d_t de sus histogramas tf-idf T y T':

$$d_t(a,b) = 1 - \frac{T \cdot T'}{|T||T'|} \tag{4.2}$$

En paralelo, los histogramas de color se comparan mediante la divergencia de Jensen-Shannon (JSD) (Lin, 1991), definida como:

$$d_c(a,b) = \sum_{m=1}^{M} H_m log\frac{2H_m}{H_m + H'_m} + H'_m log\frac{2H'_m}{H'_m + H_m} \tag{4.3}$$

donde H y H' son los histogramas de color de las imágenes a y b, respectivamente, y M es el tamaño del histograma.

Por último, las distancias TOP-SURF y de color se combinan para obtener la distancia final entre dos imágenes:

$$d(a,b) = w \cdot d_t(a,b) + (1-w) \cdot d_c(a,b) \tag{4.4}$$

donde w es un parámetro para ponderar su contribución. Dada una imagen de consulta a, sus K vecinos más cercanos del conjunto de prototipos se recuperan de acuerdo con $d(a,b)$. Los parámetros del sistema se establecieron a M= 64, w= 0,8 y K= 10.

El resultado de la clasificación se envía al usuario de la siguiente forma para su validación: una vez recibida la consulta, si no hay un número mínimo de características SURF ($s \geq 4$) en la ROI, el usuario recibe un mensaje indicando que la región seleccionada está vacía. De lo contrario, se recuperan las clases de las K imágenes más similares utilizando la metodología anterior y se muestra al usuario la primera clase en este ranking para su validación, siempre y cuando su distancia $d(a,b) < z$ (donde $z = 0,8$ es un umbral fijo). En caso de que el usuario confirme la respuesta del servidor, la imagen se etiqueta con esa clase y se almacena en el conjunto de datos. De lo contrario, se muestra una lista alternativa con las clases de las primeras imágenes top-K en la clasificación. Si ninguna de ellas se corresponde con la clase real, el usuario puede asignar otra clase seleccionándola manualmente de WordNet.

Resultados. MirBot se evaluó inicialmente a nivel de usuario usando *leaving-one-out* con descriptores locales. Los resultados de esta evaluación preliminar fueron relativamente buenos (alrededor del 30 % para 1.240 clases a fecha de enero de 2014) considerando las limitaciones de las características utilizadas para la búsqueda por similitud.

4.3. Verificación geométrica

Durante mi participación en MirBot realicé un estudio de posibles métodos empleando descriptores alternativos e incorporando verificación geométrica. Finalmente me centré en sistemas de verificación geométrica que permitieran mejorar los resultados de la búsqueda sin comprometer demasiado su eficiencia. Los detalles de este trabajo se detallan en la Sección 4.3.3. Para comprender esta contribución, en la siguiente sección se introducen otros sistemas alternativos de verificación geométrica junto con sus resultados de evaluación.

4.3.1. RANSAC

RANSAC (Mikolajczyk and Schmid, 2004) es un algoritmo muy utilizado para verificación geométrica que obtiene buenos resultados aunque con un alto coste computacional. Por este motivo se suele usar como un proceso de re-ranking.

Este método calcula una homografía y una puntuación de repetibilidad que tiene en cuenta la ubicación de los puntos y la escala de detección. La tasa de repetibilidad entre dos imágenes está representada por el número de puntos correspondientes respecto al número de puntos detectados. Considerando dos puntos x_a y x_b, estos se corresponden si:

1. El error en la posición relativa del punto es menor de 1,5 píxeles: $\|x_a - H \cdot x_b\| < 1,5$, donde H es la homografía entre las imágenes.

2. El error en la superficie de la imagen cubierta por los puntos de vecindad es $\epsilon_s < 0,4$. En la literatura este se conoce como error de solapamiento (*overlap*).

   a) En el caso de puntos invariantes a escala el error de solapamiento se define como:

$$\epsilon_s = \left| 1 - s^2 \frac{min(\sigma_a^2, \sigma_b^2)}{max(\sigma_a^2, \sigma_b^2)} \right| \tag{4.5}$$

   donde σ_a y σ_b son las escalas de los puntos seleccionados y s es el factor

de escala actual obtenido de la homografía entre las imágenes ($s > 1$).

b) El error de solapamiento para regiones afines es:

$$\epsilon_s = 1 - \frac{\mu_a \cap \left(A^T \mu_b A\right)}{(\mu_a \cup A^T \mu_b A)} \tag{4.6}$$

donde μ_a y μ_b son regiones elípticas definidas por $x^T \mu x = 1$. La unión de las regiones es $\left(\mu_a \cup A^T \mu_b A\right)$ y su intersección es $\left(\mu_a \cap \left(A^T \mu_b A\right)\right)$. A es una linealización local de la homografía H en el punto x_b. Se rechaza el posible error de 1,5 píxeles al calcular ϵ_s porque tiene poca influencia y la homografía entre imágenes reales no es perfecta.

En nuestro caso se decidió aplicar RANSAC a las 100 primeras imágenes más similares devueltas por MirBot para reordenarlas. Para la generación de hipótesis, en la experimentación inicial comparamos RANSAC usando transformaciones con 4 y 6 grados de libertad (*Degrees of Freedom*, DOF). Esto se hace para evaluar si existe alguna diferencia de rendimiento significativa entre los tipos de transformación. La transformación de 4 DOF cubre aproximadamente la rotación 2D, la traslación 2D y el escalado (los ángulos no se ven afectados), equivalente a la cuarta columna en la Figura 4.4. La transformación de 6 DOF agrega un escalado anisotrópico, que implica dos grados más de libertad que la transformación anterior (última transformación en la Figura 4.4) y que en el artículo de Mikolajczyk and Schmid (2004) llaman *Affine*, aunque realmente las transformaciones de rotación, traslación y escalado (llamadas *Similarity* en ese trabajo) son también afines.

Resultados. Se evaluaron los resultados de clasificación con datos de MirBot a fecha de enero de 2013. En ese momento la base de datos estaba compuesta por 7.958 imágenes. En la Tabla 4.1 se muestran los resultados de las pruebas realizadas incluyendo el número y porcentaje de las imágenes correctamente devueltas como primera opción por el sistema. En las dos primeras columnas se ven los resultados con los diferentes DOF de la verificación: *Similarity* muestra el resultado calculando el error con la ecuación 4.5 y *Affine* con 4.6, en ambos casos considerando toda la base de datos (sin re-ranking). La tercera columna (Base) muestra el número y el porcentaje de imágenes correctamente devueltas como primera opción sin tener en cuenta la verificación geométrica, es decir, utilizando solamente el método descrito en la sección anterior.

Tabla 4.1.: *Resultado comparativo de aplicar RANSAC sobre los datos de MirBot. Se muestra el número y porcentaje de imágenes que se recuperan correctamente.*

Similarity	Affine	Base	Similarity (re-ranking)	Affine (re-ranking)
1.863 (23,4 %)	2.133 (26,77 %)	1.646 (20,7 %)	2.084 (26,1 %)	2043 (25,6 %)

Las columnas 4 y 5 muestran los resultados tras aplicar re-ranking sobre el resultado base. Para esto se realizó la reordenación mediante RANSAC de las primeras 100 imágenes de la lista obtenida por el método base.

Como se puede ver en la Tabla 4.1, la repetibilidad con transformaciones afines tiene la mejor tasa de clasificación (26,77 %), pero con un coste computacional muy alto. *Similarity* no es tan buena como *Affine*, pero mejora la clasificación base y es bastante más rápida que esta. En el caso de re-ranking, el resultado de aplicar *Similarity* es ligeramente mejor pero no hay diferencias significativas entre ambas hipótesis, lo cual es consistente con los datos reportados por Philbin et al. (2007), quienes obtuvieron aproximadamente los mismos resultados usando diferentes valores de DOF. Una posible explicación para esto es que las imágenes en estos conjuntos de datos generalmente tienen la orientación correcta (vertical) tal como indican Philbin et al. (2007).

Como las transformaciones de 4 DOF son más eficientes y obtienen resultados similares a las de 6 DOF, en adelante los métodos estudiados se han evaluado considerando solo estas transformaciones. Este es el caso del algoritmo llamado *Spatial Pyramid Matching* (SPM) que se detalla a continuación.

4.3.2. Spatial Pyramid Matching

Probamos sobre MirBot el método de verificación geométrica SPM alternativo a RANSAC basado en el artículo de Lazebnik et al. (2006), donde se utiliza una técnica adaptada del esquema piramidal de Grauman and Darrell (2005), que propone un método para encontrar las correspondencias aproximadas entre dos conjuntos de vectores en un espacio de características d-dimensionales X e Y.

Este método aplica una subdivisión repetida de la imagen en celdas y calcula histogramas de características locales a resoluciones cada vez más finas. Para cada nivel de resolución se extraen las características que caen en cada sección. Finalmente se

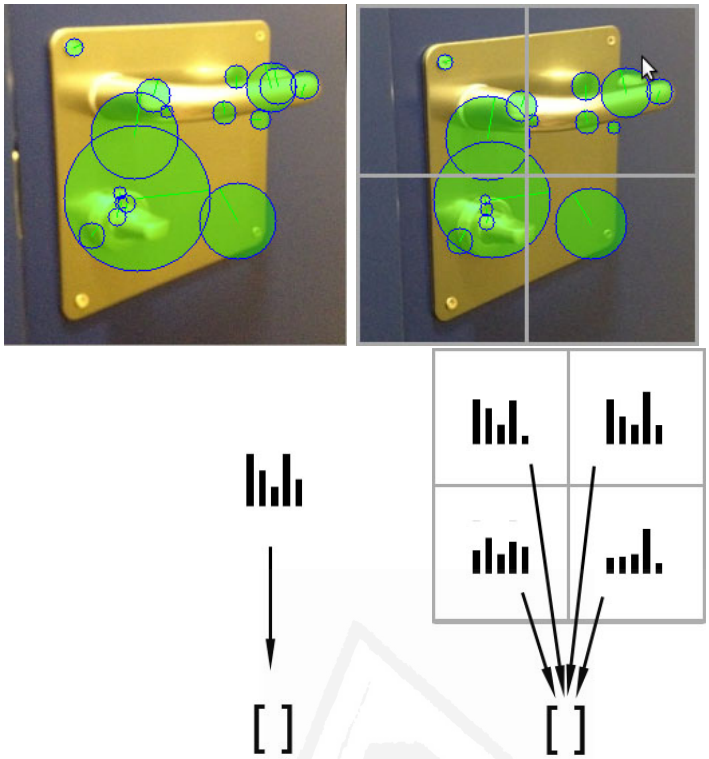


Figura 4.3.: *Subdivisión de una imagen con 2 niveles de resolución adaptada del esquema de combinación de pirámides de Grauman and Darrell (2005).*

pondera cada histograma espacial dando más peso a las cuadrículas más finas de acuerdo con la ecuación:

$$\kappa^L(X,Y) = \frac{1}{2^L}\mathcal{I}^0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}}\mathcal{I}^l \tag{4.7}$$

donde L el el número de niveles de resolución e $\mathcal{I}^l$ es el número de coincidencias en el nivel l.

En la Figura 4.3 se puede ver un ejemplo de división de una imagen extraída de la base de datos MirBot, en la que se muestran los puntos de interés y la división con 2 niveles de resolución siguiendo este esquema.

Tabla 4.2.: *Resultado comparativo de aplicar el método SPM con $L = 1$ sobre MirBot. Se muestra el número y porcentaje de imágenes que se recuperan correctamente y tiempo de proceso.*

	Base	SPM ($L = 1$)
Top-1	1.646 (20,7 %)	1.579 (19,84 %)
Tiempo	93min	137min

Tabla 4.3.: *Ejemplos de las clases en las que mejora el resultado al aplicar el método SPM con $L = 1$ sobre MirBot.*

Clases	Base	SPM ($L = 1$)
Mouse	9/88 (10,2 %)	16/88 (18 %)
Pencil	6/55 (10,9 %)	7/55 (12,7 %)
Glasses	6/75 (8 %)	9/75 (12 %)

Resultados. Este método se ha evaluado con los datos de MirBot considerando dos niveles de resolución ($L = 1$). En la Tabla 4.2 se pueden ver los resultados de este experimento. Aunque a nivel general no mejora, en algunas clases sí que lo hace. La Tabla 4.3 muestra resultados de las clases donde se observa esta mejora en comparación con la clasificación base.

Tras las pruebas realizadas podemos concluir que los resultados de aplicar SPM son dependientes de la categoría de las imágenes y de su similitud en diseño, pero que en general este método no reporta una mejora.

4.3.3. Segment Intersection of Interest Points

Durante el desarrollo de esta tesis se ha propuesto el algoritmo *Segment Intersection of Interest Points* (SIIP) publicado en Bernabeu et al. (2016). SIIP es un algoritmo de verificación geométrica simple y eficiente basado en la comparación de las intersecciones entre segmentos construidos por pares de puntos de interés de dos imágenes.

El proceso de SIIP es el siguiente. Dado un conjunto P_a de puntos de interés de una imagen a, obtenemos todos los segmentos posibles entre los puntos y calculamos sus intersecciones. Como puede verse en la Figura 4.4, las intersecciones de los segmentos permanecen invariables a las transformaciones geométricas comunes (traslación,

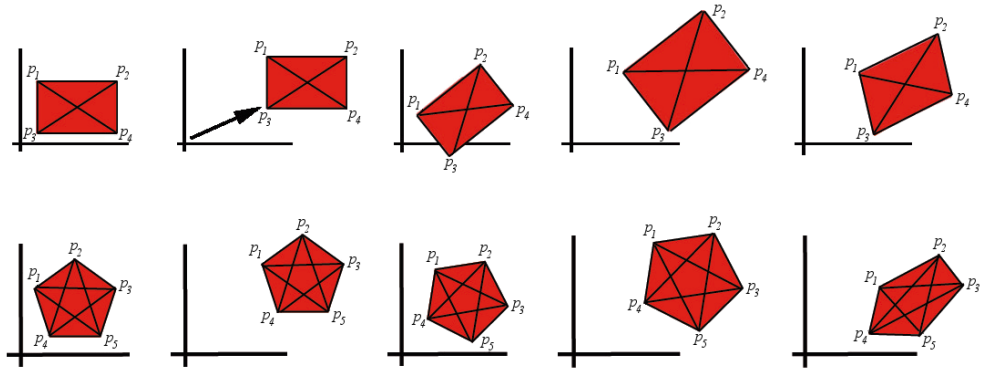


Figura 4.4.: *Las intersecciones de segmentos entre los puntos de intereses son invariantes a las transformaciones geométricas comunes.*

rotación, escala y sesgado). En el ejemplo de la primera fila de la figura, el segmento $\overline{p_1p_4}$ solo intersecta con el segmento $\overline{p_2p_3}$ en todas las transformaciones de la misma imagen. Este enfoque teórico se puede extrapolar a diferentes formas.

Usar el número de intersecciones coincidentes como medida de distancia entre dos imágenes es un método eficiente y, además, es robusto a las transformaciones 2D ya que el número de intersecciones permanece constante. La prueba de esto es sencilla, ya que los puntos de intersección pertenecen a los segmentos, por lo que la transformación aplicada a los segmentos también se aplicará a estos puntos.

4.3.3.1. Metodología

Consideramos un conjunto $P = \{p_1, p_2, \ldots, p_n\}$ de puntos de interés, donde P_a denota el conjunto de puntos de la imagen a, y $\overline{p_1p_2}$ el segmento entre los puntos p_1 y p_2. Llamamos $\mathcal{P}_2(P)$ al conjunto de todos los posibles segmentos entre dos puntos en P.

La intersección de los segmentos en un conjunto $\mathcal{P}_2(P)$ se puede calcular usando diferentes métodos. En este trabajo hemos usado un algoritmo de Chazelle and Edelsbrunner (1988) que tiene una complejidad de $\mathcal{O}(n \log n + k)$, donde n es el número de segmentos en el conjunto y k el número de intersecciones.

Dado un conjunto de segmentos $S_a = \{s_1, s_2, \ldots, s_n\}$ que pertenecen a una imagen a, definimos $I = s_1 \cap s_2$ como la función de intersección entre s_1 y s_2. Esta función devuelve si dos segmentos intersectan o no ($\emptyset$). También denotamos I_a como el conjunto

de todos los pares de segmentos que intersectan unos con otros:

$$I_a = \{(s_n, s_m) : s_n, s_m \in S_a, s_n \cap s_m \neq \emptyset, n \neq m\}$$

La verificación geométrica se realiza comparando el conjunto de intersecciones. Definimos la distancia $d(a, b)$ entre dos imágenes a y b como el número de intersecciones comunes dividido por el número máximo de intersecciones de ambas imágenes:

$$d(a, b) = 1 - (|I_a \cap I_b|)/(\text{máx}(|I_a|, |I_b|))$$

El Algoritmo 1 describe los pasos descritos para calcular la distancia entre dos imágenes a y b.

Algoritmo 1: Distancia entre dos imágenes a y b.

Data: Imágenes a, b
Result: Distancia $d_{a,b}$
$P_a = \text{SURF}(a)$; $P_b = \text{SURF}(b)$;
$M_{a,b} = \text{máx}_N\{(p_a, p_b) : p_a \in P_a, p_b \in P_b,$
$dist(p_a, p_b) < \epsilon\}$;
for ***each*** *imagen* i ***in*** a,b **do**
  $P'_i = \{p_i : p_i \in f_i(M_{a,b})\}$;
  $I_i = \{(s_n, s_m) : s_n, s_m \in \mathcal{P}_2(P'_i), s_n \cap s_m \neq \emptyset, n \neq m\}$;
end for
$d(a, b) = 1 - (|I_a \cap I_b|)/(\text{máx}(|I_a|, |I_b|))$;
return $d(a, b)$

Por tanto, en primer lugar se extraen los conjuntos de puntos de interés P_a y P_b de las imágenes de entrada. Luego, estos conjuntos se emparejan para obtener el subconjunto de puntos relacionados $M_{a,b}$ que son comunes a ambas imágenes. El emparejamiento se realiza mediante el distancia Euclídea entre los vectores de características de cada punto de interés, definido como $dist(p_a, p_b)$. Por eficiencia, los conjuntos se ordenan por relevancia usando la inversa de la distancia Euclídea entre cada par de puntos coincidentes para mantener como máximo solo los primeros N puntos (los más correlacionados) para construir segmentos entre ellos. Es importante tener en cuenta que el número de intersecciones puede ser muy elevado. En general, para un conjunto de n segmentos puede haber hasta n^2 intersecciones en el peor caso. Esta es la razón para mantener como máximo solo los N puntos de interés más similares de ambas imágenes.

Matemáticamente, definimos una función biyectiva $f_a : M_{a,b} \to P_a$ que dado un elemento del conjunto $M_{a,b}$ devuelve el punto correspondiente del conjunto P_a. Análogamente, definimos f_b para el conjunto de puntos clave P_b.

En resumen, a partir de los conjuntos iniciales de puntos de interés P_a y P_b, el algoritmo selecciona los subconjuntos de puntos P'_a y P'_b que están presentes en el conjunto de pares correspondientes $M_{a,b}$ dejando solo los puntos coincidentes para el próximo paso. A continuación se construyen los segmentos entre todos los puntos de interés filtrados P'_a y P'_b de forma independiente para las imágenes a y b respectivamente. Finalmente calculamos las intersecciones entre estos segmentos en ambas imágenes, y la distancia $d(a, b)$ que tiene en cuenta el número de intersecciones comunes.

Por ejemplo, consideremos el primer y último rectángulo de la primera fila en la Figura 4.4, denotados como imagen a e imagen b, respectivamente. Los puntos de interés p_1, p_2, p_3 y p_4 de la imagen a se corresponden con los puntos p_1, p_2, p_3 y p_4 de la imagen b. Las intersecciones de los segmentos en a son $\overline{p_1p_3} \cap \overline{p_2p_4}$, que son comunes a las intersecciones $\overline{p_1p_3} \cap \overline{p_2p_4}$ en b. Por tanto, como solo hay una intersección común y cada imagen solo contiene una intersección, su distancia es $d(a, b) = 0$.

La distancia propuesta $d(a, b)$ se ha usado para realizar una comparación con RANSAC, que solo considera los puntos clave coincidentes. No obstante, esta distancia no tiene en cuenta el número de puntos no emparejados, lo que puede ser un problema en algunos casos. Por ejemplo, en la Figura 4.4, si el rectángulo se compara con el pentágono, como su geometría es coherente, su distancia será 0.

Por esta razón se añade un término de regularización a la distancia original para considerar la proporción entre puntos clave coincidentes y no coincidentes. La distancia modificada $d'(a, b)$ se define como:

$$d'(a, b) = d(a, b) \frac{|M_{a,b}|}{\text{máx}(|P_a|, |P_b|)}$$

4.3.3.2. Evaluación

Al igual que RANSAC y SPM, SIIP se puede utilizar para reordenar un subconjunto de imágenes ya clasificadas con un algoritmo de búsqueda de similitud basado en descriptores locales con puntos de interés.

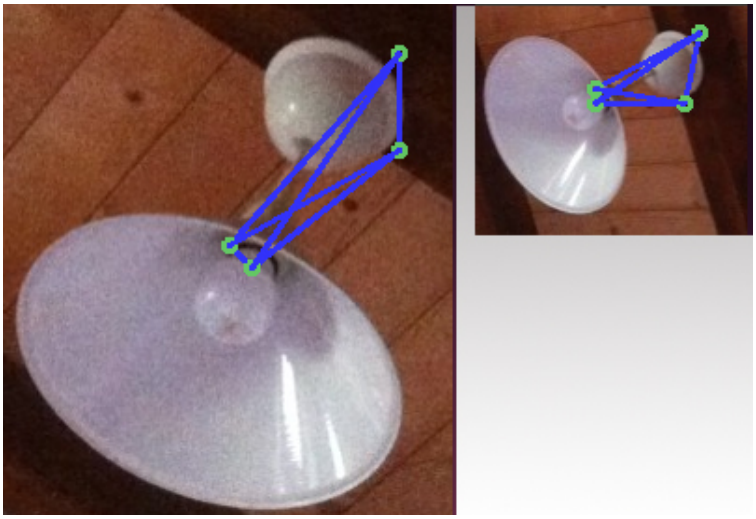


Figura 4.5.: *Puntos de interés principales y sus segmentos para dos imágenes de MirBot. Como se puede ver, sus intersecciones permanecen invariables a la rotación y escala.*

En la fase de reordenación, los puntos de interés se pueden obtener utilizando cualquier descriptor local. Hemos elegido SURF para evaluar tanto RANSAC como el método SIIP propuesto. La Figura 4.5 muestra un ejemplo de su aplicación en dos imágenes de la base de datos MirBot.

Tras el cálculo de las imágenes más similares, se reordenan las primeras K imágenes usando RANSAC y la técnica de verificación geométrica propuesta para comparar los resultados. No se ha comparado con SPM debido a que los resultados de este método con los datos de MirBot no mejoraban, como puede verse en la Tabla 4.2.

Para comparar dos imágenes a y b, primero se calcula con el descriptor SURF las N primeras coincidencias con la puntuación más alta en común entre ambas imágenes (más adelante veremos el valor óptimo de N). A continuación, se combinan pares de coincidencias para obtener segmentos (con las coordenadas x, y de cada descriptor del punto de interés de la imagen), y posteriormente se combinan los segmentos obtenidos para calcular los cruces.

La evaluación se ha realizado obteniendo las K primeras imágenes similares con TOP-SURF con valores de K entre 20 y 50, y luego reordenando los prototipos utilizando puntos de interés SURF. Además de MirBot, se han seleccionado las bases de datos Oxford 5K (Philbin et al., 2007) y Paris (Philbin et al., 2008) para comparar los resultados por ser ambas ampliamente utilizadas para técnicas de búsqueda por similitud de imágenes.

Oxford 5K contiene 5.062 imágenes de alta resolución (1024×768 píxeles) divididas en 11 clases. Se asignan las imágenes a cuatro posibles etiquetas: *Good, OK, Junk o Bad*. *Good* contiene imágenes claras (buenas) del objeto o edificio. En las imágenes *OK*, más del 25 % del objeto es claramente visible. En las imágenes *Junk*, menos del 25 % del objeto es visible o hay un nivel muy alto de oclusión o distorsión. En el conjunto *Bad* el objeto no está presente. Para la evaluación se han utilizado solo las imágenes etiquetadas con *Good* y *OK* como resultados positivos, descartando las imágenes *Junk* y utilizando *Bad* como resultados negativos.

Paris contiene 6.300 imágenes de alta resolución (1024×768 px) recopiladas de Flickr con zonas emblemáticas de París, las cuales se organizan de manera similar a Oxford.

MirBot contenía en el momento de estas pruebas 16.327 imágenes de objetos fotografiados con móviles, por lo que son de calidad media-baja y con dimensiones variables (máximo 640×640 px), a diferencia de Paris y Oxford que contienen el mismo objeto desde diferentes perspectivas. Además, en MirBot hay muchas más clases.

La tasa de acierto Top-1 (Acc) mide la relación entre los verdaderos positivos (TP) y la cantidad de imágenes en el conjunto de datos Q:

$$\text{Acc} = \frac{1}{|Q|} \sum_{q \in Q} \text{TP}(q)$$

Para calcular *Mean Average Precision* con k (MAP@k) primero calculamos Average Precision con k (AP@k) para la consulta q, y luego se obtiene MAP como la media de AP para todas las consultas:

$$\text{AP@}k(q) = \frac{1}{N_R} \sum_{n=1}^{N_R} P_k(q)$$

$$\text{MAP@}k(Q) = \frac{1}{|Q|} \sum_{q \in Q} \text{AP@}k(q)$$

donde N_R es el mínimo entre k y el número total de resultados devueltos y $P_k(q)$ es la precisión k en la lista de resultados.

Tabla 4.4.: *Resultados para los datos de MirBot reordenando las $K = 20$ primeras imágenes variando el número de puntos de interés N.*

		N=8	N=16	N=24	N=32
SIIP (d)	Accuracy	0,3140	0,3148	0,3149	**0,3150**
	MAP@10	0,1782	0,1786	**0,1786**	0,1781
RANSAC	Accuracy	0,3018	0,3017	0,3022	0,3023
	MAP@10	0,1770	0,1774	0,1776	0,1776

4.3.3.3. Resultados

La Tabla 4.4 muestra el resultado obtenido después de reordenar las $K = 20$ imágenes más similares de la base de datos MirBot. La reordenación se lleva a cabo tanto con RANSAC como con el método propuesto (SIIP). Se han evaluado diferentes valores de N para determinar cómo afecta al sistema el número de puntos de interés seleccionados para calcular la distancia, pero los cambios en N no alteran significativamente el resultado de la función de distancia d. Por tanto, para las siguientes pruebas se ha elegido $N = 24$, ya que obtiene un buen resultado para MAP@10 y Top-1 (Acc).

La Tabla 4.5 muestra los resultados obtenidos en las bases de datos MirBot, Oxford 5K y Paris. Los resultados con RANSAC y SIIP (ambos usando características SURF) se calculan con $N = 24$. Se muestran los resultados de SIIP con la función de distancia d y también añadiendo el termino de regularización d'. Como se puede ver, SIIP supera a RANSAC en todos los experimentos.

Con los datos de MirBot la función d' no mejora los resultados de d, pero en las otras bases de datos sí son consistentemente mejores. A diferencia de Paris y Oxford, en MirBot cada clase contiene diferentes objetos del mismo tipo, en lugar de el mismo objeto desde diferentes perspectivas. Este hecho, junto con que el numero de clases es muy superior, puede explicar las diferencias en los resultados.

Respecto al coste computacional del algoritmo, en estos experimentos SIIP es aproximadamente 3 veces más rápido que RANSAC de media. Por tanto se puede concluir que SIIP es más eficiente que RANSAC y mejora sus resultados.

Tabla 4.5.: *Accuracy y MAP@10 con las bases de datos MirBot, Oxford 5K y Paris. El resultado de Accuracy base (es decir, solo TOP-SURF sin reordenar) es* 0,247 *en MirBot,* 0,896 *en Oxford 5K, y* 0,744 *en Paris.*

		Accuracy			MAP@10		
Base de datos	K	RANSAC	SIIP (d)	SIIP (d')	RANSAC	SIIP (d)	SIIP (d')
MirBot	20	0,302	0,315	0,305	0,178	**0,179**	0,177
	30	0,304	0,319	0,306	0,177	0,177	0,175
	40	0,305	**0,322**	0,305	0,176	0,176	0,174
	50	0,304	0,319	0,305	0,176	0,175	0,173
Oxford 5K	20	0,920	0,923	0,928	0,823	0,826	0,830
	30	0,922	0,925	0,931	0,826	0,829	0,835
	40	0,920	0,925	0,933	0,827	0,831	0,836
	50	0,917	0,925	**0,933**	0,825	0,830	**0,837**
Paris	20	0,796	0,806	0,812	0,621	0,631	0,633
	30	0,800	0,811	0,817	0,629	0,638	0,644
	40	0,803	0,814	0,826	0,631	0,640	0,649
	50	0,803	0,816	**0,828**	0,631	0,643	**0,652**

4.4. Descriptores Neuronales

El reconocimiento de imagen dio un salto cualitativo tras la introducción de las técnicas de aprendizaje profundo, lo que nos obligó a plantearnos cambiar la aplicación MirBot para utilizar CNN en el proceso de extracción de características en lugar de los descriptores locales de la primera versión. Por tanto, la versión de MirBot publicada en (Pertusa et al., 2018) se presentó utilizando redes neuronales como extractores de características.

Para esto se llevaron a cabo pruebas con diferentes topologías de CNN. Inicialmente se evaluaron implementaciones de AlexNet (Krizhevsky et al., 2012) y GoogLeNet (Szegedy et al., 2015) de la librería Caffe (Jia et al., 2014) usando como base modelos pre-entrenados con ImageNet ILSVRC12 (Russakovsky et al., 2015). También se evaluó la red Inception21k[3] para MXNet (Chen et al., 2015), aunque hubo que adaptarla a Caffe. Finalmente se evaluó el sistema con VGG (Simonyan and Zisserman, 2015), Xception (Chollet, 2017), ResNet (He et al., 2016) e Inception v3 (Szegedy et al., 2015), usando la implementación disponible en Keras[4] con sus parámetros por defecto.

[3] https://github.com/dmlc/mxnet-model-gallery/blob/master/imagenet-21k-inception.md

[4] https://github.com/keras-team/keras

Para hacer la búsqueda por similitud las imágenes se envían a través de la red pre-entrenada para obtener los descriptores neuronales (*neural codes*), que son vectores que contienen las activaciones de salida de las neuronas de la última capa oculta (excluyendo la capa de salida). Por ejemplo, en el caso de AlexNet, esta es la capa `fc7` con 4.096 valores, en GoogLeNet es `pool5/7x7_s1`, con una dimensionalidad de 1.024, y en Inception21k es la capa `global_pool`, que también tiene 1.024 valores. Tras extraer estos vectores se usa el vecino más cercano para comparar el descriptor de la imagen consultada con el de los prototipos almacenados en el conjunto de datos.

La principal ventaja de usar los códigos neuronales con kNN en lugar de entrenar el modelo directamente con las clases de MirBot es que el sistema puede usarse de manera incremental, de forma que no hay que reentrenar la red con cada nueva muestra o clase. De esta forma, las clases de salida pueden ser diferentes a las que se usan para el entrenamiento y, además, esta técnica también permite obtener un listado con las imágenes más similares (y no solo una clasificación).

Cuando el usuario de la aplicación MirBot envía la imagen espera la respuesta del sistema para su validación, de manera que todo el proceso se realiza en tiempo real, por lo que el tiempo de respuesta es crucial. Debido a esto (y al tamaño creciente de la base de datos), en lugar de usar el método kNN estándar se decidió utilizar una técnica de vecinos más cercanos aproximados, en concreto Spotify Annoy (Bernhardsson, 2016). Este método tiene dos parámetros principales para ajustarse: la cantidad de árboles y la cantidad de nodos a inspeccionar durante la búsqueda.

En el conjunto de datos de MirBot se encontró experimentalmente un buen compromiso entre la precisión y el rendimiento al usar un índice con 100 árboles y establecer el número máximo de nodos en 1.000. Como MirBot debe operar en tiempo real, el índice no se puede reconstruir (esto requiere aproximadamente 1 minuto) cada vez que se agrega una nueva imagen. Por tanto, los descriptores que no son almacenados se mantienen en una tabla de la base de datos, y los resultados de kNN se obtienen realizando la búsqueda tanto en los prototipos del índice como en la tabla de descriptores. El índice de Annoy se reconstruye semanalmente.

En 2016 la red GoogLeNet se cambió en producción por Inception21k[5]. Inception21k es una red Inception v2 (Szegedy et al., 2015) basada en GoogLeNet con Batch Normalization (Ioffe and Szegedy, 2015). Llamamos a esta red Inception21k porque fue entrenada con el conjunto de datos completo de ImageNet (14.197.087

[5] https://github.com/dmlc/mxnet-model-gallery/blob/master/imagenet-21k-inception.md

imágenes y 21.841 clases), a diferencia de AlexNet y GoogLeNet, que fueron entrenadas con las 1.000 clases de la competición ImageNet.

Otra diferencia con respecto a la primera versión de MirBot es que inicialmente se utilizó la similitud de coseno normalizada para comparar los descriptores TOP-SURF de dos imágenes y la divergencia de Jensen-Shannon para los histogramas de color. Sin embargo, en esta segunda versión los códigos neuronales de las CNN se compararon directamente utilizando la distancia Euclídea.

4.4.1. Evaluación

En la Tabla 4.6 se puede ver una evaluación comparativa de los resultados obtenidos con los distintos métodos estudiados usando el conjunto de datos MirBot a fecha de Octubre de 2016. En esta fecha había 25.292 imágenes distribuidas en 1.808 clases. Como ya se ha comentado, la base de datos no está balanceada, por tanto algunas clases aparecen con más frecuencia que otras. Al observar las categorías raíz de WordNet, se ve que la mayoría de las imágenes almacenadas en Mirbot son objetos (18.685), seguidas de animales (4.928), alimentos y bebidas (1.113), y plantas (546). La Figura 4.6 muestra las 40 clases principales ordenadas por el número de muestras.

La evaluación se realizó usando *5-fold cross validation*. En estos experimentos solo se utilizaron las imágenes pertenecientes a las clases con más de un prototipo (24.794 imágenes de 1.180 clases). Los resultados (*accuracy*) mostrados en la Tabla 4.6 se calcularon a dos niveles, Top-1 y Top-10. En Top-1 se considera un verdadero positivo cuando la clase del prototipo más cercano coincide con la clase de consulta. Para el Top-10, se considera un verdadero positivo cuando la clase de consulta coincide con la clase de cualquiera de sus 10 prototipos más cercanos. Nótese que esto no significa que se consideren 10 clases, sino las clases de los 10 prototipos más cercanos, y si comparten la misma clase solo se tiene en cuenta una clase.

El sistema base de MirBot que usaba descriptores TOP-SURF e histogramas de color se evaluó usando los parámetros $M = 64$ (tamaño del histograma de color), $T = 100$ (número de palabras principales en TOP-SURF) y $D = 100000$ (tamaño del diccionario TOP-SURF). Se realizaron algunos experimentos variando el tamaño del histograma de color $M \in [32, 512]$, el número de palabras $T \in [50, 200]$ y el tamaño del diccionario $D \in [10000, 250000]$. En todas estas configuraciones la tasa de éxito varió en un máximo de 0,2 % y estos parámetros, por lo tanto, no tienen un impacto

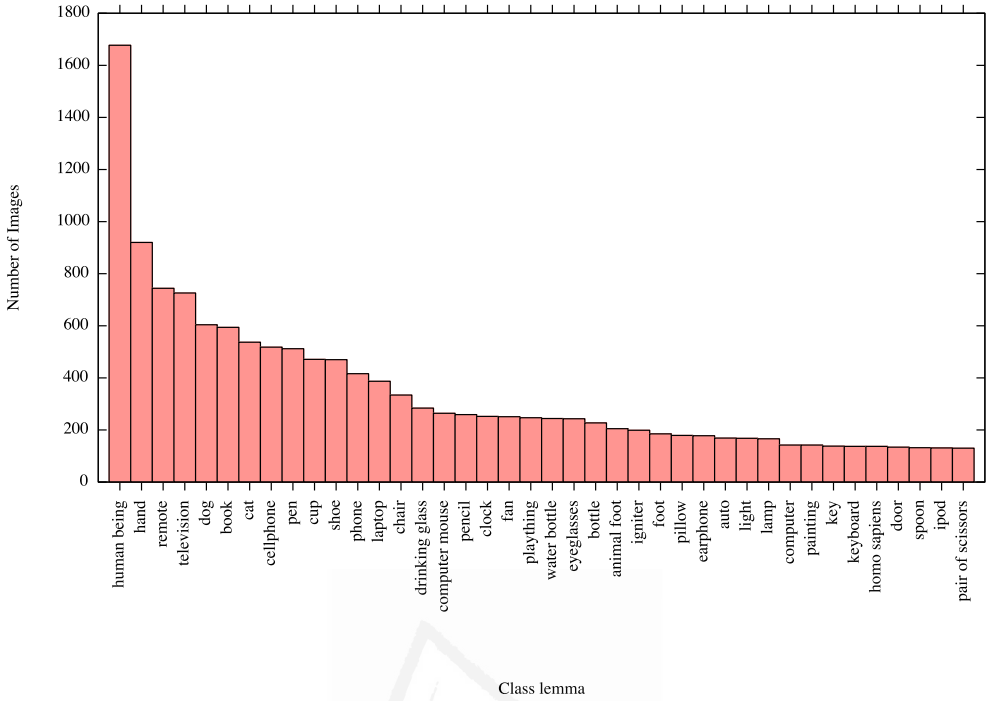


Figura 4.6.: *Número de imágenes de las 40 clases principales en MirBot.*

notable en los resultados.

Se realizaron experimentos con un diccionario TOP-SURF genérico de 100.000 palabras y con un diccionario entrenado con el conjunto de datos de MirBot para hacer los clusters más específicos a los datos objetivo. El porcentaje de acierto aumentó de 14,43 % a 19,63 % para Top-1 cuando se utilizó el diccionario entrenado, como se puede ver en la Tabla 4.6.

Además se evaluaron histogramas de color (en el espacio YCrCb) y un descriptor que combinaba color y TOP-SURF. Sorprendentemente, el descriptor de color global funcionó notablemente mejor que el descriptor local TOP-SURF. La razón de esto puede ser que algunas imágenes del mismo objeto tienen condiciones de iluminación similares, ya que los usuarios tienden a tomarlas consecutivamente cuando el sistema devuelve una respuesta incorrecta. También se evaluó la combinación de color y descriptores TOP-SURF entrenados (ver ecuación 4.4) usando el parámetro w para balancear su contribución. Curiosamente, aunque el color supera a TOP-SURF, la combinación de ambos descriptores aumenta la tasa de acierto individual para algunos valores de w. Los mejores resultados se obtuvieron con $w = 0,1$, que fue notablemente superior a los histogramas de color como puede verse en la Tabla 4.6.

Tabla 4.6.: *Resultados Top-1 y Top-10 (en porcentajes) usando 5-fold cross validation con distintas características.*

Aproximación	Top-1	Top-10
TOP-SURF genérico	14,43	35,97
TOP-SURF entrenado	19,63	41,10
Histograma de color	28,15	47,99
Combinación ($w = 0{,}1$)	36,61	55,95
VGG16	66,64	86,56
VGG19	67,02	86,81
ResNet	**71,13**	88,74
Inception v3	65,93	85,51
Xception	70,52	**88,88**
AlexNet	48,79	76,96
GoogLeNet	57,56	83,54
Inception21k	61,03	86,07
Inception21k-direct	30,31	75,54

En cuanto a los descriptores neuronales, las características se extrajeron de la última capa oculta de cada red y se normalizaron usando ℓ_2. Entre las aproximación evaluadas se comparó también la red Inception21k-direct, en la cual, en lugar de usar kNN sobre los descriptores se realiza la clasificación directamente usando la capa de salida, ya que esta red sí que contiene prácticamente todas las clases presentes en MirBot.

Como puede verse en la Tabla 4.6, los descriptores neuronales superaron claramente a los tradicionales. Como muestran los resultados, al usar modelos pre-entrenados para extraer descriptores que se comparan con kNN se obtienen altas tasas de acierto, especialmente teniendo en cuenta que las clases de consulta son diferentes a las utilizadas para el entrenamiento. Los mejores resultados se obtuvieron con ResNet y Xception, que son las redes que también obtuvieron los mejores resultados en la competición ImageNet.

Puesto que se pueden seleccionar diferentes parámetros para extraer los códigos neuronales de una imagen, además de las CNN mencionadas en este trabajo se evaluaron también técnicas de escalado de imágenes de entrada, el uso de activaciones neuronales de diferentes capas internas y la mejora obtenida al normalizar los códigos neuronales usando ℓ_2 antes de hacer la búsqueda de similitud con kNN. Se realizaron diferentes pruebas con y sin normalización y diferentes valores de k, como puede verse en la Tabla 4.7. Como se observa, el uso de normalización ℓ_2 generalmente mejora el resultado, obteniendo finalmente los mejores valores con $k = 1$ y códigos normalizados. El valor de k tan bajo podría deberse a que el conjunto de datos está muy

Tabla 4.7.: *Precisión Top-1 usando 5-fold cross-validation con y sin normalización ℓ_2 para diferentes valores de k.*

	k=1		k=5		k=11	
Aproximación	–	ℓ_2	–	ℓ_2	–	ℓ_2
VGG16	63,00	66,64	57,08	60,94	55,04	58,92
VGG19	63,56	67,02	57,53	61,07	54,93	59,12
ResNet	**69,77**	**71,13**	**62,70**	64,73	60,54	62,29
Inception v3	67,12	65,93	61,39	59,91	58,86	57,86
Xception	68,68	70,52	62,54	**65,15**	**60,88**	**63,33**
AlexNet	46,31	48,79	45,99	48,45	46,29	48,53
GoogLeNet	56,87	57,56	54,94	56,48	54,20	56,16
Inception21k	60,02	61,03	58,98	60,33	58,37	59,92

desbalanceado y algunas clases contienen muy pocas muestras.

Tras las diferentes pruebas realizadas las conclusiones obtenidas son las siguientes:

- El uso de descriptores neuronales mejora notablemente todas las aproximaciones consideradas previamente.
- La normalización de los códigos neuronales usando ℓ_2 mejoró consistentemente los resultados.
- Los mejores resultados se obtuvieron usando un valor de $k = 1$ para kNN.
- El tamaño de entrada y los parámetros de escalado más adecuados son los mismos que se consideraron originalmente para entrenar las redes.
- Los códigos neuronales de la última capa oculta obtuvieron los mejores resultados en todos los casos y, por lo tanto, esta capa fue la seleccionada para su uso en producción en MirBot.

4.4.2. Metadatos

Además de la experimentación anterior también se evaluó el sistema con los metadatos almacenados junto a las fotografías con el objetivo de complementar la información de la imagen. Para esto se llevaron a cabo experimentos preliminares para evaluar cuáles de estos metadatos contribuían a mejorar la búsqueda (eliminando to-

da información específica del usuario, como su identificador) y se comprobó que uno de los más representativos es el llamado *feature code* de Geonames, que almacena el tipo de lugar (por ejemplo zoo, universidad, playa, etc).

Para comprobar la mejora usando metadatos sin afectar demasiado a la eficiencia del sistema, en un primer experimento solo se usó el metadato *feature code* cuando la diferencia de las distancias entre la primera y la segunda clase devuelta era pequeña (es decir, para resolver situaciones de empate).

El algoritmo es el siguiente: primero se construye un histograma normalizado de *feature code* para cada clase del conjunto de entrenamiento. A continuación se clasifica una imagen de consulta usando la red ResNet. Si la diferencia entre las distancias de la clase más probable y la segunda clase devuelta es menor que un umbral ρ, el *feature code* de la imagen de consulta se comprueba en los histogramas de ambas clases. Si ese código tiene un valor mayor en el histograma de la segunda clase que en la primera, entonces la devolvemos como la clase más probable, cambiando así el orden de clase devuelto por el clasificador. Se ha evaluado esta configuración usando un umbral $\rho = 0,02$, que es la distancia promedio entre la primera y la segunda clase cuando el clasificador neuronal devuelve un resultado incorrecto.

Utilizando este valor como umbral, un 9,66 % de las muestras se reordenaron con los metadatos. El 70,1 % de estos cambios no surtieron efecto ya que no se devolvió la clase correcta debido a que esta no era ni la primera ni la segunda. En un 21,82 % de estos casos se cambió adecuadamente la clase incorrecta, y solo en un 4,29 % se cambió una clase correcta por una incorrecta. En general, la precisión Top-1 usando ResNet preentrenado mejoró del 71,13 % al 74,16 %.

Como puede comprobarse, una metodología simple que usa solo un metadato (*feature code*) consiguió mejorar la precisión en una configuración de dos etapas.

Se realizaron experimentos adicionales en el proyecto MirBot usando otros metadatos, cuyos resultados pueden verse en la publicación de Ortega-Bastida et al. (2019). En estos experimentos se añadieron los metadatos a los NC de entrada del clasificador kNN y se realizaron pruebas de clasificación a tres niveles: raíz (con las 5 categorías principales: animales, comida y bebida, objetos artificiales, objetos naturales y plantas), segundo nivel de la jerarquía WordNet (con 92 clases), y a nivel hoja (con las 1.180 clases).

Además, los resultados del clasificador kNN se compararon con los obtenidos me-

diante SVM (*Support Vector Machines*) y RF (*Random Forest*). De los experimentos realizados se puede concluir que la fusión de metadatos y características visuales solo resultó adecuada al nivel raíz (particularmente usando SVM), quizás debido a que los metadatos capturan información muy genérica por lo que no son beneficiosos para realizar una clasificación detallada de las imágenes.

CAPÍTULO 5

Búsqueda y clasificación de logos

En este capítulo se presenta un sistema desarrollado para realizar búsquedas por similitud de imágenes de marca (logos). Este método tiene en cuenta los diferentes aspectos que componen una imagen de este tipo, tanto su estructura como su semántica, y además permite que el usuario ajuste el criterio de búsqueda. El resultado de este trabajo cumple una doble funcionalidad: por un lado permite buscar similitudes entre logos basadas en distintas características de la imagen, lo cual es de gran ayuda para detectar usos no autorizados o plagios, y por otro lado facilita la tarea de etiquetado manual para la clasificación de un logo, ya que calcula la probabilidad de pertenencia a cada posible categoría, lo que también resulta muy útil para el registro de marcas y la asignación de metadatos.

Adicionalmente se han analizado las posibles topologías y clasificaciones de imagen de marca y su relación con los datos disponibles en la base de datos utilizada. Como resultado de este trabajo se han definido una serie de categorías y agrupaciones de clasificación de logos que usaremos como base para el entrenamiento y validación del sistema propuesto. La metodología para búsqueda por similitud se basa en el entrenamiento de un conjunto de CNN especializadas en cada una de las características definidas, como forma o color. Esto nos permite configurar la búsqueda por parte de los usuarios mediante un método ponderado que examina la similitud usando vecinos más cercanos.

Este capítulo está estructurado de la siguiente manera: en la Sección 5.1 se introduce el problema de clasificación de la imagen de marca y seguidamente (Sección 5.2) se detalla la base de datos usada (EUTM) y sus metadatos. La Sección 5.3 presenta los modelos neuronales usados para abordar esta tarea y seguidamente se incluye la experimentación realizada (Sección 5.4) y los resultados de la evaluación (Sección 5.5), comparando también el método propuesto con otros métodos del estado del arte y

presentando los resultados de encuestas realizadas a estudiantes y profesionales del diseño. Por último, la Sección 5.6 analiza las representaciones aprendidas por los modelos.

5.1. La imagen de marca

Tal como define Chaves (2015), "La marca no es sino la versión visual del nombre: entre ambos hay equivalencia exacta.[...] Esta sinonimia entre marca gráfica y nombre verbal se materializa empíricamente en las marcas 'bisígnicas': símbolo + logotipo. En ellas la contigüidad pautada de ambos signos instala su equivalencia en la memoria pública. Y esta sinonimia llega a su punto más alto cuando, gracias a la plena instalación del símbolo en el público, la marca puede prescindir del nombre verbal (logotipo) y funcionar eficazmente de un modo autónomo" .

Actualmente podemos encontrar una amplia variedad de logos, por ejemplo incluyendo solo una imagen, solo un texto, una letra, combinaciones de texto e imagen, etc. A continuación analizamos la categorización de imagen de marca utilizada en la literatura según su estilo, color y topología, lo cual nos servirá de base para estudiar en detalle los metadatos que se suelen emplear para etiquetar las imágenes y su posterior uso para la búsqueda y clasificación de logos.

5.1.1. El estilo

Un rasgo decisivo que establece Chaves (2015) a tener en cuenta en la creación de una imagen de marca es el estilo. Para ilustrar esto compara los estilos de marcas del sector de la tecnología informática, como por ejemplo IBM y APPLE (ver Figura 5.1), con estilos claramente diferenciados, uno basado únicamente en letras formado por líneas o bandas con un color uniforme y otro más simbólico, sin texto, solo con una imagen representativa de una manzana.

Analizando las características que definen los logos, nos podemos plantear determinar, en cierta manera, el estilo usando la información que disponemos de las formas que lo componen, por ejemplo si están formados por líneas y bandas, siluetas planas (cuadradas, redondeadas, triangulares, etc.) o con formas 3D, si tiene texto o no (y su tipo), por su semántica o significado figurativo, y por su color.

Figura 5.1.: *Ejemplo de diferentes estilos de logos de las empresas tecnológicas IBM y Apple en Chaves (2015).*

5.1.2. El color

El color es un identificador muy importante en una imagen de marca, ya que normalmente las empresas utilizan una gama de colores estable para diferenciarse de la competencia, dotarse de personalidad gráfica o visual, indicar un estilo, etc. Es tan potente el color en su dimensión identificadora y de atractivo visual que a veces la propia marca gráfica "pierde" su color y se lo cede a la superficie donde actúa para ganar en reconocimiento y llamar la atención. Por ejemplo, las letras del logotipo de Coca-Cola son rojas pero en los carteles, los envases y los camiones, el rojo pasa a la superficie y las letras quedan blancas (Chaves and Belluccia, 2003).

Las marcas pertenecientes a la misma área de negocio suelen utilizar colores similares debido a sus connotaciones culturales y sociales. Sin embargo, no siempre es así, ya que las organizaciones pueden utilizarlo para diferenciarse de la competencia. Por ejemplo, Capsule (2007) cita el caso de la empresa de tecnología Gear6 que utiliza el color verde para diferenciarse del conjunto competitivo, la categoría de sistemas *hardware* y el sector tecnológico en general, donde el azul es el color de marca más común y en el que tendría muchos competidores, incluyendo IBM, Intel, NEC, HP, EMC y Sun Microsystems.

El color se considera, por tanto, muy importante para el reconocimiento de una marca, aunque no debe entrar en conflicto con la funcionalidad del logo, por lo que a veces los logos pierden su color, como ya se ha comentado, y muestran su versión en blanco y negro. Por este motivo, la propuesta metodológica presentada en este trabajo permite clasificar el color por separado y comprobar su relación con otras características, como el sector al que pertenece la empresa.

5.1.3. Topologías

En la literatura podemos encontrar dos topologías principales para la clasificación de logos, la de Chaves and Belluccia (2003) y la de Wheeler (2013), que como veremos son coincidentes en gran medida y nos permiten establecer unas categorías genéricas para nuestro sistema con las que definir todos los elementos que definen un logotipo, como su estilo, color o significado.

En primer lugar detallamos la clasificación de marcas relativamente estandarizada establecida por Chaves and Belluccia (2003) y basada en los aspectos formales. Según Chaves, los signos identificadores gráficos presentan una gramática acabada, similar al lenguaje verbal (con posibilidades de emisión infinitas como el habla), y basada en una topología compleja pero cerrada, de manera que todo signo emergente se inscribirá en uno u otro tipo y sus respectivos subtipos, como se detalla a continuación (y que además se muestran gráficamente en la Figura 5.2):

- **Logotipos**: Solo tienen representación verbal con tipografía. Se puede establecer una subdivisión con tres tipos: logotipo solo, logotipo con fondo, logotipo con accesorio (los accesorios pueden ser incrustaciones de iconos, formas ambiguas, subrayados, etc.). Entre estos podemos diferenciar:

  - **Logotipo tipográfico**: Este puede clasificarse a su vez según la familia tipográfica utilizada: tipografía estándar, tipografía propia, tipografía singular o tipografía "retocada", en la cual el nombre se escribe con una tipografía regular pero se le aplican arreglos particulares, como modificación de tamaños y proporciones habituales de los cuerpos, ligaduras especiales, cortes o muescas en los caracteres, etc.

  - **Logotipo tipográfico iconizado**: Reemplaza alguna letra del logotipo por un icono formalmente compatible con dicha letra o con la actividad de la empresa (por ejemplo, en una hipotética marca "Sol" la "o" puede ser sustituida por el dibujo de un Sol).

- **Símbolos**: Contienen un signo no verbalizado. Se pueden dividir en tres subtipos: logo-símbolo que unifica imagen y logo en un mismo elemento (también llamado isologo), logo más símbolo (imagotipo) y símbolo solo (también llamado isotipo)[1]. Dentro de estos últimos podemos distinguir:

[1] https://es.wikipedia.org/wiki/Marca_corporativa

- **Símbolos icónicos**: El símbolo es diseñado con algún significado semántico o figurativo siguiendo un referente reconocible del mundo real o imaginario. Por ejemplo, el dibujo de un cocodrilo o el zigzag convencionalmente conocido como "rayo" (Lacoste, Goodyear).

- **Símbolos abstractos**: Son formas que no representan objetos o conceptos conocidos.

- **Símbolos alfabéticos**: Utilizan las iniciales del nombre o cualquier otra letra como motivo central, sin confundir con el modelo de "sigla" (por ejemplo la "M" de Motorola).

Estos tipos puros o extremos se complementan con formas intermedias o de transición: iconos abstraídos hasta el límite de ser irreconocibles, letras desfiguradas o iconizadas, etc. Los símbolos pueden materializarse a su vez en diversas retóricas gráficas, desde las más orgánicas hasta las más estilizadas (geométricas o normalizadas). Como se ha indicado previamente, en la Figura 5.2 se puede ver esta clasificación junto a algunos ejemplos de los mismos.

Alina Wheeler presenta otra clasificación muy similar en su libro (Wheeler, 2013), en el cual las marcas se asocian a una serie de categorías generales. En este caso, los límites entre las categorías son difusos, ya que las marcas pueden combinar elementos de más de una categoría. A continuación se detalla la topología propuesta manteniendo los nombres originales en inglés junto con su descripción para una mejor comprensión del término:

- **Wordmarks**: Un acrónimo, nombre de empresa o nombre de producto independiente que ha sido diseñado para transmitir un atributo o posicionamiento de marca. Las mejores marcas nominativas imbuyen una palabra o palabras legibles con características tipográficas distintivas que además pueden integrar elementos abstractos o elementos pictóricos. Algunos ejemplos son IKEA, IBM y Google.

- **Letterforms**: Un diseño único que utiliza una o más *letterforms* que actúan como un dispositivo nemotécnico para el nombre de una empresa. Esta letra está impregnada de una personalidad y un significado distintivo de la marca que es fácil de aplicar al icono de una aplicación. Algunos ejemplos son Univision y Unilever.

Figura 5.2.: *Clasificación de marcas establecida por Norberto Chaves. Imagen obtenida de Chaves (2015).*

- **Emblems**: Una marca en la que el nombre de la empresa está indisolublemente unido a un elemento pictórico. Los elementos nunca están aislados, como sucede por ejemplo en los logotipos de TiVo, OXO y LEED.

- **Pictorial marks**: Una imagen literal inmediatamente reconocible que se ha simplificado y estilizado. La imagen en sí misma puede aludir al nombre de la empresa o su misión, o puede ser un símbolo de algún atributo de la marca. Ejemplos: Apple, NBC y Lacoste.

- **Abstract/symbolic marks**: Un símbolo que transmite una idea y, a menudo, encarna una ambigüedad estratégica. Las marcas abstractas son especialmente eficaces para las empresas de tecnología y de servicios. Algunos ejemplos son Sprint, Nike y HSBC.

Estas dos topologías son las más utilizadas por los diseñadores y artistas. Sin embargo, como ya se ha indicado en la Sección 3.2.1, las agencias de registro de marcas de casi todo el mundo han aceptado como estándar la clasificación Viena (World Intellectual Property Organization, 2002) para etiquetar sus conjuntos de datos. Este sistema internacional permite etiquetar diferentes características de las marcas mediante una topología jerárquica que está ordenada desde los conceptos más generales a los más específicos. Para esto emplea una serie de metadatos que indican su significado figurativo (semántica), el color, la forma y si contienen o no texto.

En la Tabla 5.1 se muestra un resumen de las equivalencias entre las nomenclaturas y topologías propuestas por Wheeler y Chaves. Además se incluye una tercera columna donde se indica su relación con las etiquetas de la codificación de Viena utilizada en nuestro conjunto de datos (detallado en la Sección 5.2). Estas equivalencias nos permiten determinar las características más importantes de los logotipos a la hora de elaborar o analizar su diseño. Por ejemplo, el color y la forma son características que aparecen en todos los tipos de diseños y, por lo tanto, son de gran ayuda a la hora de distinguirlos. No ocurre lo mismo con el texto o los elementos figurativos, aunque resultan muy útiles para determinar algunas de las características de los logos. En resumen, existe una relación entre las diferentes topologías, lo que significa que los códigos de Viena utilizados en los que se basa este trabajo pueden describir el resto de topologías.

5.2. Conjunto de datos y clasificación propuesta

En base a las topologías descritas, aceptadas convencionalmente, se han propuestos unos criterios de clasificación que establecen unas reglas objetivas concretas que eliminan las posibles ambigüedades de estos etiquetados y que están orientados a facilitar su uso en sistemas de aprendizaje automático, como la metodología propuesta.

La base de datos utilizada en este trabajo es EUTM (*European Union TradeMark*), etiquetada siguiendo la codificación de Viena. En la Tabla 5.2 se muestran las 29

Tabla 5.1.: *Relación entre las topologías de logos mencionadas y la codificación de Viena en EUTM.*

			Viena			
	A. Wheeler	**N. Chaves**	**Figurative**	**Color**	**Shape**	**Text**
	Wordmark	**Logotype:**				
Nominal		◇ Logotype with background	–	✓	✓	✓
Identifier		◇ Pure Logotype	–	✓	✓	✓
		◇ Logotype with accessory	✓	✓	✓	✓
	Emblem	**Logo-symbol**	✓	✓	✓	✓
	–	**Logotype with symbol**	✓	✓	✓	✓
Symbolic	**Only symbol:**	**Only symbol:**				
identifier	◇ Pictorial mark	◇ Iconic symbol	✓	✓	–	–
	◇ Abstract/symbolic mark	◇ Abstract symbol	–	✓	✓	–
	◇ Letterform	◇ Alphabetic symbol	–	✓	✓	–

categorías principales que se utilizan para describir de manera codificada el contenido de las imágenes. Estas categorías, como ya vimos previamente, se dividen a su vez en categorías de segundo y tercer nivel, creando una clasificación con cientos de etiquetas posibles. Cada código se indica mediante el patrón `XX.YY.ZZ`. Por ejemplo, el código 5.9.1 se usaría para asignar la etiqueta "zanahorias" a un logo. La jerarquía de este código indica que pertenece a la categoría de segundo nivel 5.9 "hortalizas" y a la categoría principal 5 "plantas".

Esta organización jerárquica resulta muy útil, dado que nos permite agrupar los logos por distintos niveles de etiquetas y utilizar niveles jerárquicos superiores cuando el tercer o el segundo nivel presenten demasiado detalle, sean poco representativos, contengan pocas muestras o incluso sean ambiguos. Además, hay que tener en cuenta que el etiquetado que se realiza con esta clasificación la mayoría de veces no es exhaustivo, ya que se suelen anotar solo las características distintivas de la marca, por lo que podemos encontrar etiquetados incompletos o contradictorios (e.g. un logotipo con tres colores pero en el que solo se etiqueta uno de ellos).

Para solucionar estos problemas, en este trabajo proponemos agrupar los códigos según su significado y su relación con las topologías establecidas. La intención no es

Tabla 5.2.: *Codificación Viena (World Intellectual Property Organization, 2002). Para nuestra tarea se consideran figurativos los códigos desde 1 hasta 25. Del código 26 en adelante se utilizan para definir Forma, Texto y Color.*

Código	Descripción
1	Celestial Bodies, Natural Phenomena, Geographical Maps
2	Human Beings
3	Animals
4	Supernatural, Fabulous, Fantastic or Unidentifiable Beings
5	Plants
6	Landscapes
7	Constructions, Structures for Advertisements, Gates or Barriers
8	Foodstuffs
9	Textiles, Clothing, Sewing Accessories, Headwear, Footwear
10	Tobacco, Smokers' Requisites, Matches, Travel Goods, Fans, Toilet Articles
11	Household Utensils
12	Furniture, Sanitary Installations
13	Lighting, Wireless Valves, Heating, Cooking or Refrigerating Equipment, Washing Machines, Drying Equipment
14	Ironmongery, Tools, Ladders
15	Machinery, Motors, Engines
16	Telecommunications, Sound Recording or Reproduction, Computers, Photography, Cinematography, Optics
17	Horological Instruments, Jewelry, Weights and Measures
18	Transport, Equipment for Animals
19	Containers and Packing, Representations of Miscellaneous Products
20	Writing, Drawing Or Painting Materials, Office Requisites, Stationery and Booksellers' Goods
21	Games, Toys, Sporting Articles, Roundabouts
22	Musical Instruments and their Accessories, Music Accessories, Bells, Pictures, Sculptures
23	Arms, Ammunition, Armour
24	Heraldry, Coins, Emblems, Symbols
25	Ornamental motifs, Surfaces or backgrounds with ornaments
26	Geometrical figures and Solids
27	Forms of writing, Numerals
28	Inscriptions in various characters
29	Colors

sustituir la codificación de Viena, sino realizar un procesado de las etiquetas para quedarnos con aquellas que resultan más útiles para aplicarlas en métodos de aprendizaje automático como el propuesto. Para esto se definen las siguientes cuatro categorías, en las que se seleccionan o agrupan los códigos Viena que describen estas características:

- **Figurativo**: Llamamos diseños figurativos a los códigos entre el 1 y el 25 (ver Tabla 5.2), ya que están relacionados con símbolos icónicos (objetos reconocibles en el mundo real o imaginario que se encuentran en la imagen), es decir, la

Tabla 5.3.: *Codificación de Viena utilizada para color (código 29 en Tabla 5.2), y forma (código 26 en Tabla 5.2).*

Código	Color
29.01.01	Red
29.01.02	Yellow
29.01.03	Green
29.01.04	Blue
29.01.05	Violet
29.01.06	White
29.01.07	Brown
29.01.08	Black
29.01.95	Silver
29.01.96	Gray
29.01.97	Gold
29.01.98	Orange
29.01.99	Pink

Código	Shape
26.01	Circles, ellipses
26.02	Segments or sectors of circles or ellipses
26.03	Triangles, lines forming an angle
26.04	Quadrilaterals
26.05	Other polygons
26.07	Different geometrical figures, juxtaposed, joined or intersecting
26.11	Lines, bands
26.13	Other geometrical figures, indefinable designs
26.15	Geometrical solids

semántica de la imagen. Para esta categoría solo diferenciaremos entre el 1er y el 2^{o} nivel de la jerarquía, que denominaremos respectivamente categoría principal (que contiene los 25 códigos del 1er nivel) y subcategorías (con 123 clases posibles, cuyos códigos siguen el patrón XX.YY, por ejemplo, 5.9 vegetales o 5.7 grano, semillas, frutas). No consideramos el 3er nivel de división pues presenta una alta granularidad con cientos de posibilidades que en ocasiones pueden resultar ambiguos y además suelen contener pocos ejemplos, por lo que el entrenamiento no llega a obtener buenos resultados. Así pues, finalmente, se establecieron dos niveles de clasificación que llamaremos *main-category* y *sub-category*, respectivamente.

- **Colores**: En la Tabla 5.3 (izquierda) se muestran los colores utilizados en la agrupación propuesta. Viena también incluye otra información sobre el número de colores presentes en la imagen (por ejemplo 29.01.12 cuando hay dos colores predominantes), que no se utilizan en este trabajo puesto que no aportan información que pueda ser de utilidad para nuestro sistema.

- **Formas**: Para definir la forma de los logotipos se ha utilizado la categoría 26 de la clasificación Viena, en la cual se etiquetan diferentes tipos de formas, incluyendo círculos, triángulos, cuadriláteros, líneas, etc. En este caso, el 3er nivel de etiquetado es muy específico y, a veces, ambiguo (por ejemplo, líneas curvas

frente a líneas onduladas, o líneas punteadas frente a líneas discontinuas). Por lo tanto, proponemos usar solo hasta el 2º nivel. Además, las claves 26.07 y 26.13 se agrupan con la categoría 26.5 (Otros polígonos), puesto que visualmente no se identifica una forma definida y tras realizar una serie de pruebas preliminares se comprobó que esta agrupación obtiene mejores resultados. La lista completa de códigos propuestos se muestran en la Tabla 5.3 (derecha).

- **Texto**: La categoría 27 de la clasificación Viena define el texto y sus características. Sin embargo, esta categoría también resulta demasiado detallada (por ejemplo, hay 20 códigos diferentes para indicar la apariencia o la forma del texto y otros tantos para etiquetar el estilo de la fuente). Dado que el texto específico que aparece en el logotipo a menudo se compone de siglas, monogramas o nombres de marcas que no contribuyen mucho al cálculo de la similitud entre logotipos, proponemos etiquetar solo la presencia o ausencia de texto en la imagen.

Además de estas categorías, las imágenes en la base de datos EUTM también disponen de información relacionada con la actividad de la empresa. Para esto se utiliza la clasificación de Niza que divide esta actividad en 45 sub-categorías, distinguiendo Bienes y Servicios, los cuales serán referidos como sectores y sub-sectores en este trabajo. Las etiquetas utilizadas para Bienes incluyen productos químicos, medicamentos, metales, materiales, máquinas, herramientas, vehículos, instrumentos, etc., mientras que las etiquetas utilizadas para Servicios incluyen publicidad, seguros, telecomunicaciones, transporte y educación, entre otros. En el Apéndice C se puede consultar la lista completa de actividades recogidas en esta clasificación.

5.3. Metodología

La Figura 5.3 muestra el esquema del sistema propuesto. En este esquema se diferencian claramente tres bloques o etapas: una primera fase para el preprocesado de las imágenes de entrada, una segunda que realiza la clasificación multi-etiqueta a partir de la imagen preprocesada y un último paso que aprovecha las características aprendidas en la segunda etapa para realizar la búsqueda por similitud. En las siguientes secciones se explica cada una de estas etapas en detalle.

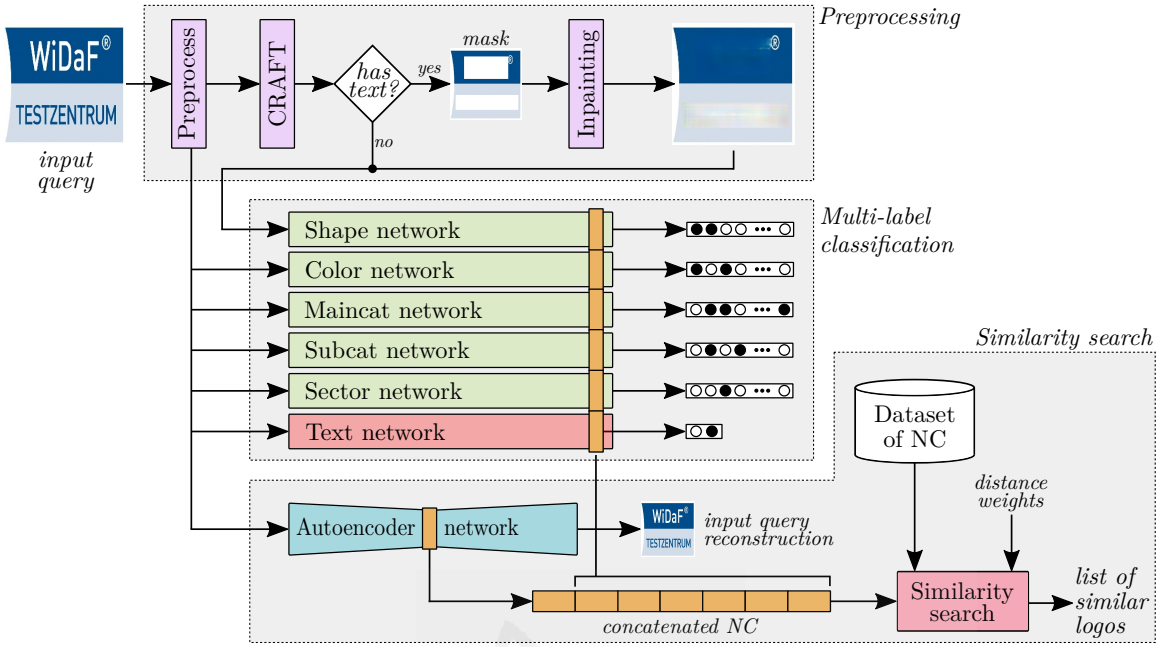


Figura 5.3.: *Esquema del método propuesto.*

5.3.1. Preprocesado de datos

El preprocesado de datos se realiza con el fin de preparar la imagen para los siguientes pasos. En primer lugar se recorta el logotipo para eliminar los bordes que contienen un fondo uniforme. Las imágenes de logotipos utilizadas para el registro de marcas o la búsqueda de similitudes generalmente tienen un fondo uniforme. Las imágenes se recortan, por tanto, eliminando los bordes con colores uniformes de manera que el logotipo ocupe todo el espacio disponible de la imagen. Esto permite homogeneizar su tamaño y, por tanto, facilita el proceso de comparación.

El segundo paso del preprocesado es detectar si el logo de entrada contiene texto y, en caso afirmativo, generar una versión del mismo sin él. Muchos logos incluyen texto, pero esta información a veces puede resultar irrelevante o incluso confundir en la detección de determinadas características. Durante la experimentación se observó que al eliminar el texto se conseguía una mejora notable al clasificar las formas. No fue así para el resto de características, como el color o el significado figurativo, por lo que finalmente se decidió realizar este proceso solo para las formas y usar la imagen del logo completa para el resto de características (ver Figura 5.3).

Para realizar esta tarea se procesa la imagen utilizando el detector de texto CRAFT

(Baek et al., 2019), que detecta, de manera eficiente, el área de texto de una imagen explorando cada región y la afinidad entre caracteres de texto. Como resultado devuelve una caja (polígono) que contiene el texto detectado y que en nuestro caso utilizamos para generar una máscara con su posición. Esta máscara junto a la imagen original se procesan mediante una red de *inpainting* (Wang et al., 2018) para rellenar los huecos detectados con colores adecuados. Por optimización, si la máscara detectada está rodeada de píxeles de color blanco, se rellena el hueco directamente con ese color. La Figura 5.4 muestra algunos ejemplos de los pasos seguidos en este proceso.

Figura 5.4.: *Ejemplo de preproceso: la primera fila muestra el logo original, en la segunda se puede ver cómo se elimina el texto seleccionado en la imagen mediante el método CRAFT y, por último, en la tercera fila se ilustra cómo se rellena el hueco del texto detectado usando para esto una red neuronal de inpainting.*

5.3.2. Clasificación multi-etiqueta

En el segundo paso del método propuesto se usa un conjunto de redes neuronales especializadas en la clasificación multi-etiqueta de distintas características de la imagen de entrada. En concreto se decidió crear una red especializada en cada uno de los grupos de etiquetas propuestos para la clasificación (ver Sección 5.2), como son: forma, color, texto, categoría principal, subcategoría, sector y sub-sector al que pertenece el logo. De esta manera, cada una de estas redes devolverá una propuesta

de etiquetado para una característica distinta con el objetivo de asistir al operador en el proceso de clasificación.

En la Figura 5.3 se puede ver el esquema con la integración de estas redes en la metodología propuesta, las cuales parten del resultado preprocesado de la etapa anterior. En el caso de la red especializada en la forma, se parte de la versión de la imagen sin texto. Nótese que al ser independientes las redes se pueden ejecutar en paralelo, por lo que el rendimiento del algoritmo no se ve muy afectado.

Como se ha discutido en los capítulos de introducción, los métodos actuales que mejores resultados obtienen para el tratamiento de logos, o imágenes en general, usan CNN. Por este motivo hemos decidido basarnos en este tipo de arquitectura, pero adaptándola a una configuración multi-etiqueta (MLC).

La definición específica de las redes utilizadas se puede ver en la Figura 5.5 (diagrama superior). La arquitectura propuesta consta de cinco capas que alternan convoluciones, *Batch Normalization*, *Max-Pooling* y *Dropout*, más dos capas finales completamente conectadas (también con *Dropout*). *Batch Normalization* y *Dropout* se incluyen para reducir el sobreajuste (*overfitting*), para mejorar la generalización y precisión de las redes, y además para ayudar a entrenar de forma más rápida.

ReLU se ha usado como función de activación en todas las capas excepto para la capa de salida, la cual emplea la función de activación sigmoidea. Esta función modela la probabilidad de cada clase como una distribución de Bernoulli, donde cada clase es independiente del resto (a diferencia de lo que ocurriría en una activación tipo Softmax). Por lo tanto, como salida de estas redes obtenemos una clasificación multi-etiqueta para cada una de las características consideradas.

En el caso de la red especializada en la detección de texto solo es necesaria una salida, dado que esta red solo pretende detectar la presencia o ausencia de texto en la imagen. A diferencia de CRAFT, que busca caracteres de manera individual, en esta red se buscan características globales que permitan realizar dicha clasificación binaria. Esto permite que la siguiente etapa del método aproveche la representación intermedia aprendida (tanto de esta red como de las demás) para realizar la búsqueda por similitud entre logos en base a la característica que cada red se especializa en clasificar. De este modo, la comparación con las características detectadas por la red especializada en el texto devolverá otros logos similares, que también contengan texto, y no imágenes con textos que se parezcan.

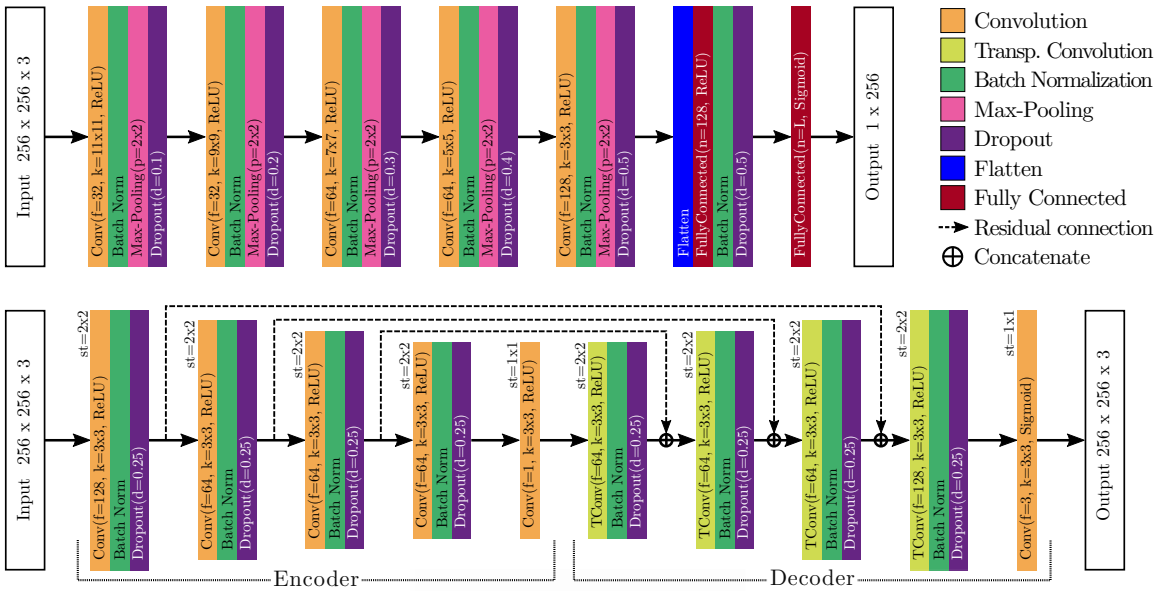


Figura 5.5.: *Esquema de las CNN (parte superior) y el Auto-Encoder (parte inferior). En esta figura, cada tipo de capa se representa con un color siguiendo la leyenda lateral. La configuración de cada capa se indica en el esquema, incluyendo la función de activación utilizada, el número de filtros (f) y el tamaño del kernel (k) para las convoluciones y las convoluciones transpuestas, el tamaño (p) usado para el Max-Pooling, la ratio d del Dropout, el valor de stride st aplicado para cada capa del Auto-Encoder, y el número de neuronas n usadas para las capas tipo fully-connected.*

5.3.3. Búsqueda por similitud

En el último paso del método se aprovechan las características aprendidas por las redes descritas en la sección anterior para realizar la búsqueda por similitud de logos. Para ello extraemos de la CNN los códigos neuronales (*Neural Codes* o NC) de la penúltima capa como un vector de características y los utilizamos como entrada del algoritmo de búsqueda por similitud (por ejemplo kNN). Para realizar una búsqueda durante la etapa de inferencia se procesa el nuevo logo y se compara con los distintos descriptores del conjunto de entrenamiento, devolviendo un listado de los más similares. Para acelerar este proceso de comparación, una vez se han entrenado las redes, se procesan todos los logos del conjunto de entrenamiento, extrayendo sus NC y almacenándolos para la comparación durante la etapa de inferencia.

Además de las CNN descritas en la sección anterior también se consideró el uso de una red tipo Auto-Encoder (ver Figura 5.3). Estas redes se entrenan de manera no supervisada para la reconstrucción de las imágenes suministradas como entrada. Su

arquitectura fuerza a tenga que aprenderse una representación comprimida de la imagen que permita realizar su reconstrucción con el menor error posible, obteniendo de este modo descriptores en los que se almacenan las características más representativas de la imagen. Esta representación en nuestro caso resulta muy útil, ya que nos permite realizar una comparación que considere de manera general todas las características del logo.

En la parte inferior de la Figura 5.5 se puede ver la topología concreta del Auto-Encoder implementado. El *encoder* contiene cuatro capas convolucionales combinadas con *Batch Normalization* y *Dropout*. El submuestreo se realiza en la misma convolución usando *stride* (en lugar de añadir una capa de *pooling*). En la parte del *decoder* se añaden otras cuatro capas con una arquitectura espejo que reconstruyen la imagen con el mismo tamaño de la entrada original. En este caso se utilizan capas convolucionales transpuestas, que realizan la operación inversa a la convolución. Además, con el objetivo de facilitar la convergencia y mejorar los resultados, se añadieron conexiones residuales desde cada capa de *encoder* con su capa análoga en el *decoder*.

El tamaño de los NC extraídos de las CNN es 128 y el del Auto-Encoder es 256. En experimentos preliminares se observó que con tamaños inferiores empeoraba el resultado y que tamaños superiores no reportaban ninguna mejora. Como puede verse en la Figura 5.3, los NC se combinan en un vector de características que se utiliza para realizar la búsqueda por similitud. Este vector se normaliza utilizando ℓ_2-norm, ya que esta técnica habitualmente mejora los resultados del proceso de comparación (Gallego et al., 2018b).

Para la búsqueda por similitud, los NC del conjunto de entrenamiento son extraídos y almacenados siguiendo el proceso descrito. Luego, durante la fase de inferencia, se obtiene la representación NC de la nueva imagen y se compara con los NC almacenados del conjunto de datos. En este proceso se utiliza kNN (Duda et al., 2001), pero además hemos comparado los resultados con los siguientes dos clasificadores:

- **Binary Relevance kNN (BRkNN)** (Eleftherios Spyromitros, 2008): Es un clasificador multi-label basado en kNN y en el problema de transformación mediante relevancia binaria (*Binary Relevance*). Para esto simplemente se entrena un clasificador binario por cada etiqueta diferente siguiendo una estrategia uno-contra-todos.

- **LabelPowerset** (Boutell et al., 2004): También sigue la aproximación de transformar el problema multi-etiqueta en uno multi-clase. En este caso se entrena

un clasificador *Random Forest* (Breiman, 2001) considerando todas las combinaciones de etiquetas encontradas en el conjunto de entrenamiento como si fueran clases distintas.

Además, para la búsqueda de los vecinos más cercanos se utiliza una función de distancia ponderada. De esta manera se permite al usuario ajustar los criterios de búsqueda modificando los pesos asignados a cada característica (por ejemplo para dar más peso al color que a la forma). En concreto usamos la siguiente ecuación para calcular la distancia entre dos vectores de características A y B:

$$d(A,B) = \frac{\sum_{c\in\mathcal{C}} w^c d(A^c, B^c)}{\sum_{c\in\mathcal{C}} w^c} \tag{5.1}$$

donde $\mathcal{C}$ es el conjunto de todas las características posibles a clasificar, A^c y B^c representan los subconjuntos de características correspondientes a la característica c, w^c es el peso asignado a esa característica, $\forall c \in \mathcal{C} : w^c \in [0,1]$, y d es la distancia Euclídea.

5.3.4. Entrenamiento de la red

El entrenamiento de las redes se realiza mediante retro-propagación estándar (*back-propagation*) usando *Stochastic Gradient Descent* (SGD) (Bottou, 2010) y actualizando la tasa de aprendizaje (*learning rate*) mediante el método adaptativo propuesto en Zeiler (2012). Como función de pérdida se usó *binary crossentropy* para calcular el error comparando la salida de la CNN y el resultado esperado etiquetado en el *ground truth*. En el caso de la red utilizada para el texto, al tratarse de clasificación binaria, se utilizó *categorical-cross-entrophy*. El entrenamiento se realizó durante un máximo de 100 épocas y aplicando *early stopping* cuando la pérdida no disminuía durante 15 épocas. El tamaño del mini-batch se estableció en 32 muestras.

Para la red CRAFT se utilizó un modelo pre-entrenado durante 25.000 iteraciones con más de 10.000 imágenes[2]. La red de inpainting se inicializó con los pesos pre-entrenados con ImageNet y posteriormente se entrenó durante 30.000 iteraciones

[2] https://github.com/clovaai/CRAFT-pytorch

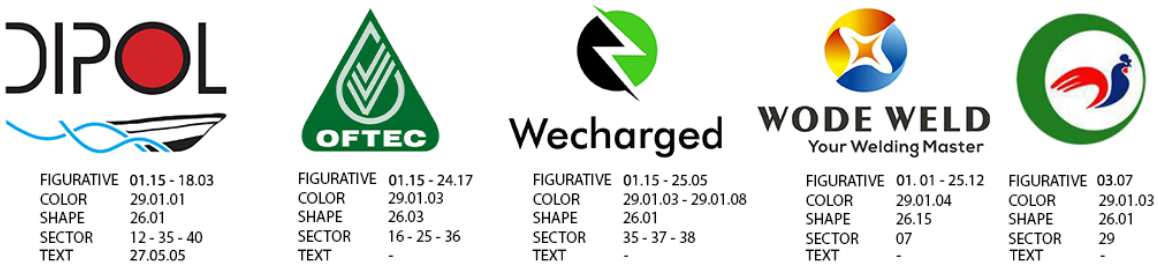


Figura 5.6.: *Ejemplos de la información etiquetada de la base de datos EUTM.*

utilizando 8.500 de nuestros logos[3].

5.4. Configuración de la experimentación

5.4.1. Conjunto de datos

Tal como se ha comentado anteriormente, la experimentación de este trabajo se ha realizado utilizando la base de datos *European Union Trademark* (EUTM) obtenida del sitio web de la Oficina de Propiedad Intelectual de la Unión Europea (EUIPO).

Este dataset está etiquetado usando la clasificación Viena. Sin embargo, hay que tener en cuenta que no todos los campos están etiquetados, como puede verse en la Figura 5.6. Esto se debe a que, por lo general, las características se etiquetan solamente cuando representan un elemento distintivo del logo. En esta figura podemos observar la inconsistencia al etiquetar los colores: En la primera imagen solo se ha anotado el color rojo, aunque también tiene negro y azul, en el tercer logo se ha anotado verde y negro, y en el cuarto y quinto logo, que tienen tres colores, solo se indica uno de ellos. Con respecto a las formas se observa algo similar. En la primera, tercera y quinta imagen solo se anota la etiqueta "círculos", aunque también contienen otras formas. El cuarto logo, que también es circular, se etiqueta como "sólido geométrico" ya que quizás se quiso destacar que tiene efecto 3D o degradado. También se observa cómo el texto solo se etiqueta en la primera imagen, cuando hay otros tres que también tienen texto. Esto se debe a que en los otros logos el texto no tiene ningún diseño figurativo o elemento distintivo.

[3] https://github.com/shepnerd/inpainting_gmcnn

Estas inconsistencias a veces se deben a otros factores como son que la semántica de marcas puede ser a veces subjetiva y presenta dificultades cuando el etiquetado se establece manualmente por un operador, bien por el propio proceso de etiquetado, que depende en gran medida de la destreza del usuario, o bien motivado porque la codificación de Viena es una categorización cerrada y hay ciertas características complicadas de definir.

La propuesta realizada puede facilitar el trabajo de etiquetado de las marcas puesto que las redes entrenadas ofrecen opciones de etiquetado con una probabilidad asociada, lo que puede ayudar al operador en el proceso de etiquetado al hacer constar una determinada característica que el usuario puede decidir si etiquetar o no.

Para la experimentación se ha añadido a los metadatos una etiqueta indicando si la imagen contiene texto o no. Esta etiqueta es afirmativa cuando la imagen contiene el código de Viena 27 (como se explica en la Sección 5.2). Para completar la información del texto se procesan todas las imágenes con CRAFT, etiquetando cuando detecta texto en la misma. De esta forma obtenemos 70.892 logos etiquetados con texto, que representa el 94 % del total, frente a los 22.845 que tienen originalmente esta etiqueta en la base de datos. De esta manera tendremos una información más completa y menos ruidosa sobre la presencia de texto en la imagen con la que entrenar y evaluar el modelo propuesto.

De las 76.000 imágenes de logos que disponemos en el conjunto de datos, el 80 % se han seleccionado para entrenamiento y el 20 % restante para la fase de test. Las imágenes se han escalado a una resolución de 256×256 píxeles. De esta manera se consigue homogeneizar su tamaño y se facilita el proceso de comparación.

5.4.2. Métricas

Tal como se ha comentado anteriormente, en una tarea MLC cada muestra puede tener asociadas varias etiquetas. Para evaluar cuantitativamente los resultados se otorga una puntuación alta y una mejor clasificación cuando se predicen más etiquetas correctas. En este trabajo utilizamos las siguientes métricas multi-etiqueta para realizar este proceso de evaluación (Tsoumakas et al., 2010):

5.4.2.1. Label Ranking Average Precision (LRAP)

Esta es una métrica vinculada a la precisión promedio pero basada en la noción de *Label Ranking* en lugar de *Precision* y *Recall*. LRAP promedia sobre las muestras la respuesta a la siguiente pregunta: para cada etiqueta, ¿qué fracción de las etiquetas mejor clasificadas son etiquetas verdaderas?

El valor de esta métrica será mayor si el método puede otorgar un mejor rango a las etiquetas asociadas con cada muestra. La puntuación obtenida es siempre mayor que 0, siendo 1 el mejor valor. Si hay exactamente una etiqueta relevante por muestra, la precisión promedio de clasificación de etiquetas es equivalente a calcular el *Mean Reciprocal Rank* (MRR).

Formalmente, dada una matriz de etiquetas $y \in \{0,1\}^{N\times L}$, donde N y L son el número de muestras y etiquetas, respectivamente, y dada la puntuación asociada a cada etiqueta $\hat{f} \in \mathbb{R}^{N\times L}$, LRAP se define como:

$$\mathrm{LRAP}(y,\hat{f}) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{\|y_i\|_0}\sum_{j:y_{ij}=1}\frac{|\mathcal{L}_{ij}|}{\mathrm{rank}_{ij}} \tag{5.2}$$

donde $\mathcal{L}_{ij} = \left\{k : y_{ik}=1, \hat{f}_{ik} \geq \hat{f}_{ij}\right\}$, $\mathrm{rank}_{ij} = \left|\left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}\right|$, $|\cdot|$ calcula la cardinalidad (número de elementos) del conjunto, y $\|\cdot\|_0$ es la ℓ_0-norm que calcula el número de elementos del vector que no son cero.

5.4.2.2. Label Ranking Loss (LRL)

La métrica LR calcula el *ranking* que promedia sobre las muestras el número de pares de etiquetas ordenadas incorrectamente (etiquetas verdaderas con una menor puntuación que las etiquetas falsas), ponderado por la inversa del número de pares ordenados de etiquetas falsas y verdaderas. El mejor valor posible es 0. La métrica se define formalmente como:

$$\mathrm{LRL}(y,\hat{f}) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{\|y_i\|_0(L-\|y_i\|_0)}\left|\left\{(k,l) : \hat{f}_{ik} \leq \hat{f}_{il}, y_{ik}=1, y_{il}=0\right\}\right| \tag{5.3}$$

donde $|\cdot|$ calcula la cardinalidad del conjunto, y $\|\cdot\|_0$ es el ℓ_0-norm que calcula el número de elementos del vector que no son cero.

5.5. Evaluación

En esta sección se detallan los resultados de evaluación del método propuesto. En primer lugar, la Sección 5.5.1 muestra la evaluación cuantitativa con los resultados obtenidos para la clasificación MLC usando el conjunto de datos descrito en la Sección 5.2. Posteriormente, en la Sección 5.5.2 se evalúa el resultado de la búsqueda por similitud y se compara con otros resultados del estado del arte usando la base de datos METU (Sección 5.5.4), la más utilizada en trabajos previos.

Además de los resultados cuantitativos, también se evalúa el método cualitativamente analizando las predicciones realizadas por el método, la representación interna aprendida por los modelos y las zonas de los logotipos en las que se centra para realizar las predicciones. También se compara la respuesta del sistema con la clasificación obtenida en encuestas realizadas a estudiantes y profesionales del diseño, todos ellos con conocimientos de diseño gráfico.

5.5.1. Clasificación MLC

En primer lugar se evalúa la clasificación multi-etiqueta del método para cada una de las características consideradas. La Tabla 5.4 muestra los resultados con LRAP y LRL para las diferentes CNN. Si analizamos el resultado reportado por ambas métricas podemos ver resultados similares con respecto a los mejores y peores resultados (con excepción de Sector), siendo Color, Main-category y Sub-category las características mejor detectadas. El clasificador de forma es el siguiente mejor y se puede ver que eliminar texto de la imagen mejora los resultados (fila Shape+) en comparación con utilizar la versión original del logotipo incluyendo el texto (Shape). Los peores resultados se obtienen para Sector y Sub-sector. Esto se debe a que no se detecta ningún patrón, característica, ni tipo de diseño específico que lo determine, ya que es bastante ambiguo y amplio respecto al concepto y posibilidades de diseño (aunque se vea un valor alto en LRAP para Sector, que puede deberse a que solo tiene dos opciones posibles). Debido a los malos resultados obtenidos para la clasificación del Sub-sector, no se considerará en los siguientes experimentos para evitar

Tabla 5.4.: *Resultados obtenidos por el método propuesto para la etapa de clasificación multi-etiqueta. Para la red Shape se distinguen dos casos: "Shape+" incluyendo el pre-procesado para eliminar el texto y "Shape" sin incluirlo.*

Network	LRAP	LRL
Color	0,8642	0,0561
Sub-category	0,7376	0,0561
Main-category	0,7979	0,0635
Shape+	0,7699	0,1169
Shape	0,6899	0,1534
Sector	0,8890	0,2220
Sub-sector	0,3165	0,2642

que introduzca ruido en la búsqueda por similitud.

Los resultados para la red de clasificación de texto no se incluyen en esta tabla dado que no se trata de una clasificación multi-etiqueta (puesto que solo se discrimina si la imagen contiene texto o no). Por esto, hemos recurrido a la métrica *accuracy*, obteniendo que el método propuesto clasifica correctamente el texto en el 96.06 % de los casos.

La Tabla 5.5 muestra algunos ejemplos de los resultados obtenidos para la clasificación multi-etiqueta, incluyendo solamente las predicciones realizadas con una confianza superior al 2 %. Si comparamos la predicción con el *Ground Truth* (GT) podemos ver cómo el método acierta en todos los casos con un porcentaje de confianza bastante alto. El único error es para la predicción de Main-category del segundo logotipo, para el que indica como primera opción la clase "plants". Sin embargo, este error resulta comprensible si analizamos los ejemplos etiquetados con esta clase, ya que suelen definir formas lineales con curvas y de color verde. Para el etiquetado de forma se puede ver cómo en algunos ejemplos, como el primero, tercero y cuarto, el método propone también otras clases que no estaban etiquetadas, pero que sin embargo describen características presentes en las imágenes.

5.5.2. Búsqueda por similitud

En esta sección vamos a evaluar los resultados de la búsqueda por similitud realizada utilizando los NC aprendidos por la redes neuronales de la etapa de MLC junto con los NC del *Auto-Encoder*. En este caso mostramos los resultados considerando solamente la métrica LRAP, puesto que, como se ha visto en el apartado anterior, la tendencia

Tabla 5.5.: *Ejemplos de clasificación MLC en el conjunto de datos EUTM, incluyendo el GT y la predicción realizada por las redes de main-category, forma, color y texto cuando el porcentaje de confianza de la predicción supera el 2 %. Se muestran dos ejemplos con texto y otros dos sin texto.*

					
GT	**Main-category:**	Ornamental motifs	Human beings	Plants	Ornamental motifs Games, toys
	Shape:	Quadrilaterals Lines, bands	Circles, ellipses	Lines, bands	Quadrilaterals
	Color:	Black; Orange	Green	Blue	Black; White
	Text:	Yes	Yes	No	No
Prediction	**Main-category:**	100 % ornamental motifs	47.10 % plants 17.70 % human beings 4 % arms, ammunition	46.47 % plants 31 % heraldry, coins 9.66 % celestial bodies	100 % ornamental motifs
	Shape:	94.85 % quadrilaterals 6.07 % lines, bands 4.08 % other polygons	99.81 % circles, ellipses	63.62 % circles, ellipses 61.87 % lines, bands 10.44 % quadrilaterals	99.96 % quadrilaterals 10.06 % circles, ellipses
	Color:	48.44 % black 94.41 % orange 58.45 % white	99.22 % green	100 % blue	87.54 % black 78.36 % white
	Text:	Yes	Yes	No	No

es la misma en ambas métricas (exceptuando el Sector).

Con la intención de ajustar el valor de k utilizado por los métodos kNN y BRkNN y a su vez analizar el ruido existente en el etiquetado, se evalúa el resultado obtenido al realizar la búsqueda por similitud para el caso *single-label*. Para esto, al procesar cada clase se han considerado solamente las muestras etiquetadas con una única etiqueta para esa característica.

La Tabla 5.6 muestra los resultados de este experimento (usando la métrica LRAP) con el método kNN y valores en el rango $k \in [1-11]$. Como se puede ver, los mejores resultados se obtienen usando valores de k altos, entre 7 y 11, lo que demuestra que el etiquetado proporcionado contiene ruido, dado que el método mejora al considerar más vecinos para realizar la inferencia. En base a estos resultados, se decidió establecer un valor intermedio de k (en concreto $k = 9$) para el resto de experimentos.

Como LabelPowerset está basado en *Random Forests*, también hemos realizado un experimento similar pero evaluando el número de árboles considerados en el rango $t \in [100, 500]$, obteniendo el mejor resultado con $t = 100$. Estos parámetros se han usado posteriormente para comparar los tres algoritmos de búsqueda de similitud multi-etiqueta considerados: kNN y BRkNN con $k = 9$, y LabelPowerset con $t = 100$. La Tabla 5.7 muestra los resultados de este experimento con la métrica LRAP. El mejor resultado por método se resalta en negrita. Como puede verse, LabelPowerset es el que mejor funciona con casi todas las características. La única excepción es el sector, que, como se ha argumentado anteriormente, es una característica muy subjetiva y, por lo tanto, puede contener un mayor nivel de ruido en las etiquetas.

En el caso del Auto-Encoder (AE) hay que considerar que se entrena de manera no supervisada para la reconstrucción de la entrada, para lo cual reporta un error medio al cuadrado (*Mean Squared Error* o MSE) de solo 0,0004. Para evaluar mejor sus resultados vamos a comparar el LRAP que obtendríamos al usar su codificación para realizar la búsqueda por similitud de las distintas características consideradas, comparando además estos resultados con los obtenidos por las redes especializadas. La Figura 5.7 muestra este análisis, en el que destacan buenos resultados para casi todas las características a excepción del color. Esto indica que el AE aprende una representación genérica de las características de forma combinada, que tiene principalmente en cuenta la forma frente al color. Como puede verse, el AE obtiene muy buenos resultados cuando clasifica la categoría principal, las sub-categorías y el sector. Destaca el hecho que el AE trabaja mejor que la CNN especializada en Shape, aunque no es así para el caso de Shape+, es decir, cuando eliminamos el texto de la imagen en la fase de pre-proceso.

El AE no resulta ser el mejor en ninguna característica particular a excepción del caso de forma sin pre-procesar. Por tanto, este método resulta especialmente útil para la búsqueda por similitud de manera genérica, teniendo en cuenta la apariencia general del logo sin centrarse en ninguna característica en particular.

5.5.3. Resultados cualitativos

En esta sección vamos a analizar de manera cualitativa los resultados obtenidos de la búsqueda por similitud. En las Figuras 5.8 a 5.14 se muestran una serie de ejemplos de los logos obtenidos al utilizar cada una de las redes especializadas por separado, es decir, asignando el 100 % del peso en la búsqueda a una sola característica. En todas

Tabla 5.6.: *Resultados obtenidos con el clasificador kNN para las diferentes tareas (single-label) usando la métrica LRAP.*

	CNN + kNN					
	k=1	**k=3**	**k=5**	**k=7**	**k=9**	**k=11**
Color	0,8322	0,8366	0,8369	**0,8394**	0,8378	0,8378
Main Category	0,7673	0,7842	0,7875	0,7885	**0,7886**	0,7880
Subcategory	0,7409	0,7611	0,7660	0,7682	0,7695	**0,7716**
Sector	0,8020	0,8027	0,8054	**0,8067**	0,8065	0,8060
Subsector	**0,0883**	0,0553	0,0474	0,0454	0,0423	0,0397
Shape	0,5489	0,5513	0,5503	0,5542	0,5544	**0,5552**
Shape+	0,6583	0,6717	0,6707	0,6689	0,6712	**0,6728**

Tabla 5.7.: *Resultados obtenidos con clasificadores multi-etiqueta para las diferentes tareas con la métrica LRAP.*

	KNeighbors	**BRkNNa**	**LaberPowerset**
Color	0,7042	0,7042	**0,7070**
Main Category	0,7015	0,7015	**0,7396**
Subcategory	0,6589	0,6589	**0,6850**
Sector	**0,8434**	**0,8434**	0,8001
Shape	0,5333	0,5333	**0,5594**
Shape+	0,6242	0,6242	**0,6579**

las figuras el primer logo de la fila es la consulta realizada y el resto son los 8 vecinos más cercanos recuperados.

La Figura 5.8 muestra dos ejemplos de búsquedas de logos similares aplicando todo el peso al color. Como se puede ver, en estos casos los colores recuperados son correctos, incluso cuando el logo tiene múltiples colores independientemente de otras características como la forma. En los ejemplos de la Figura 5.9 se muestra que las formas también se devuelven correctamente, en este caso sin tener en cuenta el color.

La semántica del logo viene determinada por las características main-category y sub-category, que son más complicadas de comparar visualmente debido a que muchas veces los elementos pueden representarse de manera icónica o abstracta. Por este motivo se han seleccionado etiquetas que visualmente resultan más sencillas de comparar. En la Figura 5.10 se muestran ejemplos de búsqueda por la categoría principal. La primera fila corresponde a la etiqueta “Plants” y la segunda a “Celestial Bodies, Natural Phenomena, Geographical Maps” (ver Tabla 5.2). Frecuentemente la categoría principal es muy general y es necesario bajar al nivel de subcategoría

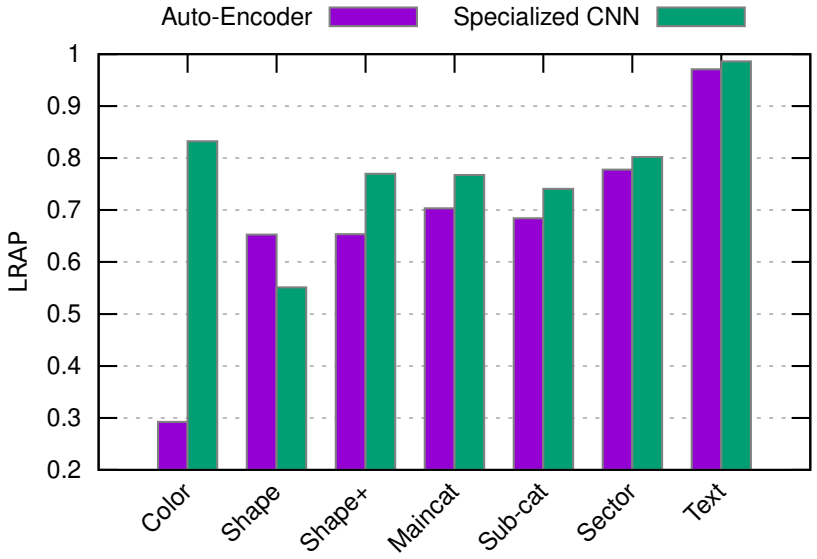


Figura 5.7.: *Resultados de la búsqueda por similitud para todas las características usando los NC del Auto-Encoder comparado con el resultado de las CNN. En la métrica LRAP los valores más altos indican mejores resultados.*

Figura 5.8.: *Ejemplo de 8 vecinos más cercanos usando solo la característica de color. El primer logo es el de consulta.*

Figura 5.9.: *Ejemplo de 8 vecinos más cercanos usando solo la característica de forma. El primer logo es el de consulta.*

para identificar mejor los elementos. En la Figura 5.11 se muestran subcategorías correspondientes a la categoría principal de la figura anterior, de esta forma: el primer

Figura 5.10.: *Ejemplo de 8 vecinos más cercanos usando solo la característica main-category. El primer logo es el de consulta.*

Figura 5.11.: *Ejemplo de 8 vecinos más cercanos usando solo la característica sub-category. El primer logo es el de consulta.*

Figura 5.12.: *Ejemplo de 8 vecinos más cercanos usando solo la característica del sector. El primer logo es el de consulta.*

Figura 5.13.: *Ejemplo de 8 vecinos más cercanos usando solo la característica de texto. El primer logo es el de consulta.*

Figura 5.14.: *Ejemplo de 8 vecinos más cercanos usando Auto-Encoder. El primer logo es el de consulta.*

ejemplo contiene la etiqueta "Leaves, Needles, Branches With Leaves Or Needles" (código Viena 05.03) como una subcategoría de "Plants" (primera línea en la figura anterior). La segunda línea muestra la etiqueta "Natural Phenomena" (código Viena 01.15) como una subcategoría de "Celestial Bodies, Natural Phenomena, Geographical Maps" en la figura anterior. En ambos casos se puede ver que se han recuperado logos similares pero al bajar a nivel de subcategoría los elementos son más fácilmente identificables. Por ejemplo, en el caso de "Plants" de la categoría principal los resultados devueltos contiene hojas o plantas, pero el diseño parece un poco más genérico, incluyendo otros elementos, como personas, mientras que para la subcategoría solo se muestran logos que incluyen hojas.

El sector es muy difícil de recuperar correctamente, más aún cuando solo hay dos clases (bienes y/o servicios) y no están relacionadas directamente con la información visual. La primera fila en la Figura 5.12 muestra un ejemplo de la clase "Bienes" en la que todas las imágenes recuperadas han sido de esa misma clase. La segunda contiene ambas etiquetas "Bienes" y "Servicios" y, de igual manera, todas las recuperadas son correctas.

En el caso de texto (Figura 5.13), se puede observar que además de recuperar logos con o sin texto, el sistema también tiene en cuenta la composición del mismo en la imagen de consulta. En la primera línea se han recuperado logos sin texto y, aunque en la quinta muestra se puede ver un logo con la letra "R", realmente no entraría en la clasificación de texto propiamente dicho sino que se considera *Letterforms* según la topología presentada en la Sección 5.1.3. En la segunda fila se realiza una búsqueda de logos con texto y, como hemos indicado, además de detectarlo correctamente también tiene en cuenta su composición, ya que todos los textos aparecen situados en la misma zona de la imagen

Por último analizamos los resultados obtenidos mediante el Auto-Encoder, los cua-

Figura 5.15.: *Resultados obtenidos utilizando la distancia ponderada con dos categorías diferentes. La primera imagen es la consulta y en la segunda columna se muestran los pesos aplicados a cada categoría.*

les, como se puede ver en la Figura 5.14, se centran principalmente en la composición espacial y, en algunos casos, también tiene en cuenta el color.

5.5.3.1. Combinación de características

A continuación vamos a analizar el efecto de combinar varias características mediante la distancia ponderada propuesta (ver ecuación 5.1) para permitir a los usuarios ajustar la búsqueda. La Figura 5.15 muestra algunos ejemplos de los resultados obtenidos al aplicar distintos pesos para combinar el color, la forma y los elementos figurativos.

En el primer ejemplo se utilizan las características de categoría principal y forma, comparando el resultado obtenido asignando el 70 % del peso a la categoría principal y el 30 % a la forma, y viceversa. Como se puede ver, al agregar peso a la forma, se recuperan imágenes con un elemento figurativo similar y formas más similares que los resultados mostrados en la Figura 5.10 (fila 1), en la que la forma varía mucho. Al invertir los pesos y dar más importancia a la forma, la Figura 5.15 muestra imágenes con formas más similares pero elementos figurativos ligeramente diferentes. Por ejemplo, en la fila 1 aparece una forma de cuadrilátero, que desaparece cuando le

damos más peso a la forma (fila 2) donde los resultados similares tienen la etiqueta "círculos, elipses".

En el siguiente ejemplo se han combinado las características de color y forma, comparando el resultado obtenido al asignar un 70 % del peso al color y 30 % a la forma, y al contrario, 30 % del peso al color y 70 % a la forma. Como puede verse, al dar más importancia al color se obtienen resultados en los que todas las imágenes tienen el mismo color pero con formas ligeramente distintas (a diferencia de los resultados mostrados en la Figura 5.9, en los que variaba mucho más el color). Al invertir los pesos y dar más importancia a la forma todos los resultados obtenidos presentan formas similares pero aparecen más cercanos los resultados que tienen un color similar. Si analizamos los resultados del último logo de consulta (con la etiqueta "Triangles, lines forming an angle") podemos ver que al dar más peso al color solo aparece alguna forma completamente triangular. Sin embargo, invirtiendo los pesos puede verse que se obtienen solo formas triangulares, como en la Figura 5.9 pero con una ordenación distinta, ya que se da cierta importancia al color.

5.5.4. Comparación con el estado del arte

En esta sección se compara el método propuesto con otros métodos del estado del arte para TIR. Dado que, como se ha argumentado previamente, no conocemos otras aproximaciones de MLC para logos que utilicen la clasificación de Viena, vamos a realizar esta comparación utilizando el conjunto de datos METU v2 descrito en la Sección 3.2.1. Para la evaluación utilizaremos la métrica *Normalized Average Rank* (NAR), ya que es la métrica común en los trabajos del estado del arte comparados. Para calcular esta métrica, el conjunto de imágenes de consulta es "inyectado" en el conjunto de datos principal y, para cada logo, el ranking obtenido por los logos del mismo grupo se calcula usando la siguiente fórmula:

$$\mathrm{NAR} = \frac{1}{N \times N_{rel}} \sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \tag{5.4}$$

donde N_{rel} es el número de imágenes relevantes para una imagen de consulta dada (el número de imágenes "inyectadas"), N es el tamaño del conjunto de imágenes, y R_i es el ranking de la i^{th} imagen inyectada. El valor 0 corresponde al mejor resultado y 0,5 corresponde a una ordenación aleatoria.

La Tabla 5.8 muestra el resultado de la comparación realizada en la que, como puede verse, se consideraron distintos tipos de aproximaciones, tanto basadas en descriptores tradicionales como en redes neuronales. Para las características tradicionales se muestran resultados extraídos de Tursun et al. (2017) usando histogramas de color (Lei et al., 1999), LBP (Ojala et al., 2002), SIFT (Lowe, 2004), SURF (Bay et al., 2008), TRI-SIFT y OR-SIFT (Kalantidis et al., 2011). También se han considerado dos propuestas más elaboradas: el uso de SIFT excluyendo las características de las áreas con texto (Perez et al., 2018), y una versión mejorada de SIFT (Feng et al., 2018) en la que se extraen características invariantes inversas de los bordes de los bloques segmentados que posteriormente se agregan para realizar la búsqueda por similitud.

Para las aproximaciones basadas en redes neuronales se ha comparado el uso de modelos pre-entrenados, en particular GoogLeNet (Szegedy et al., 2015), AlexNet (Krizhevsky et al., 2012) y VGG16 (Simonyan and Zisserman, 2015), extrayendo los NC de una de sus capas (`77S1`, `FC7` y `Pool5`, respectivamente). También se han considerado propuestas específicas para este dataset, como el trabajo de Tursun et al. (2017), que combina seis características tradicionales con NC extraídos de tres arquitecturas CNN distintas.

Asimismo, se ha evaluado la propuesta de Perez et al. (2018) que compara tres soluciones: los resultados de la arquitectura VGG19 entrenada de dos formas distintas (una para distinguir similitudes visuales y otra para similitudes conceptuales), y el resultado de fusionar las características de ambas. Por último también se muestran los resultados de un trabajo basado en mecanismos de atención (Tursun et al., 2020), que presta atención directa sobre información crítica, tales como elementos figurativos, y reduce la atención de elementos carentes de información aparente, como textos y fondo. Este proceso (que denominan ATRHA, *Automated Text Removal Hard Attention*) se combina con dos propuestas para la elaboración de las características comparadas, una basada en *Regional Maximum Activations of Convolutions* (R-MAC) y otra basada en el rasgo predominante de los mapas de activación convolucional (CAM) detectados mediante mecanismos de *Soft Attention* (CAMSA) y la agregación de *Maximum Activations of Convolutions* (MAC).

Como puede verse en los resultados de la Tabla 5.8, en general los métodos basados en redes neuronales mejoran notablemente a los basados en características tradicionales. Aunque con algunas excepciones, las redes pre-entrenadas, es decir, no preparadas específicamente para este tipo de datos, no consiguen buenos resultados, siendo superadas incluso por un método basado en características tradicionales ("Enhanced

Tabla 5.8.: *Comparación con los resultados anteriores del estado del arte con el conjunto de datos METU utilizando la métrica NAR. Un valor NAR más bajo indica mejores resultados.*

Aproximación	Método	NAR
Características tradicionales	Color histograms (Lei et al., 1999)	0,400
	SIFT (Lowe, 2004)	0,348
	TRI-SIFT (Kalantidis et al., 2011)	0,324
	LBP (Ojala et al., 2002)	0,276
	SURF (Bay et al., 2008)	0,207
	OR-SIFT (Kalantidis et al., 2011)	0,190
	SIFT without text (Perez et al., 2018)	0,154
	Enhanced SIFT (Feng et al., 2018)	0,083
Basadas en redes neuronales	GoogLeNet (Szegedy et al., 2015)	0,118
	AlexNet (Krizhevsky et al., 2012)	0,112
	VGG16 (Simonyan and Zisserman, 2015)	0,086
	Visual network (Perez et al., 2018)	0,066
	Conceptual network (Perez et al., 2018)	0,063
	ATRHA R-MAC (Tursun et al., 2020)	0,063
	Fusion of hand-crafted & CNN features (Tursun et al., 2017)	0,062
	Fusion of visual and conceptual networks (Perez et al., 2018)	0,047
	ATRHA CAMSA MAC (Tursun et al., 2020)	0,040
Método propuesto	Auto-Encoder	0,118
	Color	0,090
	Shape	0,044
	Weighted features (70 % color, 30 % shape)	0,034
	Weighted features (30 % color, 70 % shape)	**0,018**

SIFT" de Feng et al. (2018)). Es interesante ver cómo la propuesta de Tursun et al. (2017) para la combinación de descriptores tradicionales con los extraídos de CNN consigue una mejora notable en los resultados. Entre los métodos basados solamente en redes neuronales destaca la propuesta que utiliza mecanismos de atención de Tursun et al. (2020).

Al comparar estos resultados con nuestra propuesta podemos ver cómo el Auto-Encoder obtiene unos resultados muy bajos para esta tarea, similar a los obtenidos por las aproximaciones basadas en descriptores tradicionales. Esto posiblemente sea debido a que se entrena de manera no supervisada y aprende características muy genéricas. Al utilizar por separado las características aprendidas por las redes especializadas en el color y la forma los resultados mejoran, siendo mejor el resultado obtenido para la forma. Este último sí que consigue mejorar a casi todos los trabajos del estado del arte, a excepción de ATRHA CAMSA MAC (Tursun et al., 2020). Por último, al utilizar la propuesta para combinar de forma ponderada las características se consiguen mejorar todavía más los resultados, superando al resto de métodos del estado del arte. Destacar que, de nuevo, al asignar más peso a la forma (70 %) que al color (30 %), nuestra propuesta mejora el resultado de los trabajos anteriores con un

margen notable (0,018 frente a 0,040).

En la Figura 5.16 se puede ver un ejemplo de los resultados obtenidos para la base de datos METU al utilizar nuestra propuesta, combinando las características de forma al 70 % y las de color al 30 %. En esta figura el primer logo es la imagen consultada y a continuación se muestran los 10 logotipos más similares encontrados para dicha consulta, marcando con un asterisco (*) los resultados correctos encontrados. Para la consulta de la figura superior hay un total de 13 logos similares y nuestro método encuentra 8 de ellos entre los 10 primeros resultados, y el resto en las posiciones 11, 16, 17, 20 y 23 (de entre un total de 923.340 posibles logos). Para la consulta de la segunda fila son 9 los logotipos similares a encontrar. En este caso, el método devuelve 7 de ellos entre los 10 primeros resultados, y los otros dos en las posiciones 57 y 65.

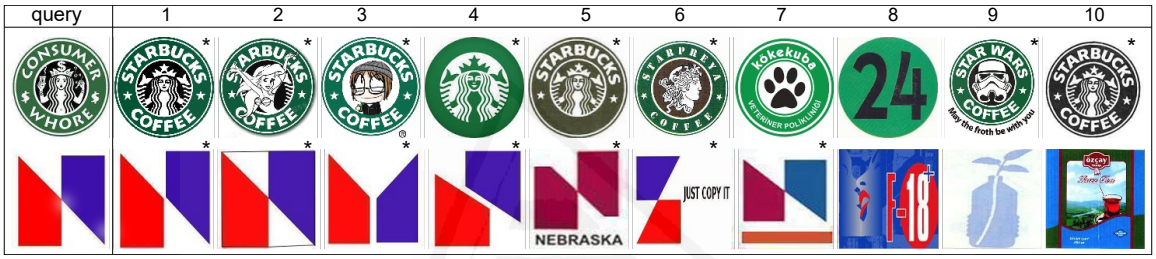


Figura 5.16.: *Ejemplo de los 10 vecinos más cercanos obtenidos en el conjunto de datos METU asignando el 30 % del peso al color y el 70 % a la forma. El primer logo es la consulta, los resultados correctos encontrados están marcados con un asterisco (*).*

5.5.5. Encuestas

Puesto que la semántica de marcas puede resultar subjetiva en muchos casos, se decidió evaluar los resultados obtenidos mediante encuestas realizadas a estudiantes y profesionales del diseño, los cuales, por tanto, están familiarizados con conceptos de diseño gráfico aplicado a imagen de marca. Para preparar las encuestas se seleccionaron al azar 3 logotipos con etiquetas de color, 3 de forma y 6 con elementos figurativos para cada participante. En total se realizaron 12 preguntas por participante y la encuesta la completaron un total de 107 personas. En las instrucciones se les indicaba que marcaran las etiquetas que observaban presentes en el logo de entre las etiquetas propuestas como respuestas, las cuales, además de las respuestas correctas, incluían otras no asignadas a la imagen.

En las encuestas sobre el color tenían que contestar a la pregunta: "Indica si observas

Criterio	Encuestas	Nuestra propuesta
Color	0,6735	0,7070
Shape	0,5467	0,6579
Subcategory	0,3673	0,6850

Tabla 5.9.: *Resultados obtenidos en la encuesta a estudiantes y profesionales del diseño usando la métrica LRAP comparados con los resultados obtenidos por nuestra propuesta. Valores más altos de LRAP indican mejores resultados.*

en este logotipo los colores siguientes (el fondo blanco no se considera color)". En este caso se mostraban como opciones los 13 colores posibles etiquetados en la base de datos. Para la forma se pidió a los encuestados responder a la pregunta: "Dada esta imagen indica si contiene las siguientes formas", mostrando todas las opciones utilizadas en las base de datos. Por último, en el caso de los elementos figurativos, dado que hay cientos de etiquetas posibles, se decidió mostrar las opciones correctas junto con otras respuestas incorrectas seleccionadas al azar y distintas para cada logo. En este caso debían responder a la pregunta: "Indica si observas en este logo los siguientes elementos figurativos."

La Tabla 5.9 muestra los resultados obtenidos calculados usando la métrica LRAP considerada anteriormente. Estos resultados se comparan con los de las redes CNN especializadas en la misma característica (mostrados previamente en la Tabla 5.7). Como puede verse, los resultados de las redes mejoran los obtenidos en las encuestas. Estos resultados confirman la dificultad que plantea esta tarea debido a la subjetividad existente cuando se interpreta el significado de los elementos que aparecen en un logo o las características que puedan ser consideradas representativas de la marca, incluso para personas familiarizadas con el diseño de logos.

A continuación se muestran algunos ejemplos de las preguntas realizadas en la encuesta, incluyendo datos estadísticos sobre el número de respuestas correctas proporcionadas por los participantes. Por ejemplo, si hay dos opciones posibles, se indica el número de participantes que acertaron ambas o solo una, y los casos en el que, además de tener una o dos respuestas correctas, también respondían otras opciones incorrectas. Los ejemplos incluyen las etiquetas asignadas en la base de datos y las opciones presentadas en la encuesta.

La Figura 5.17 muestra dos logos de ejemplo usados en la encuesta sobre color. En la figura de la izquierda destaca que ninguna persona ha marcado las respuestas correctas. Los colores azul y rojo fueron los más apreciados por los encuestados en la imagen, aunque la imagen no estaba etiquetada como azul sino como negra, y el color

(a) Etiquetas (correctas): Blanco; Negro

Respuestas	#	%
Two	0	0
One	0	0
Two (and others)	3	11,54
One (and others)	7	26,92
Others (red and/or blue)	16	61,54

(b) Etiquetas: Negro; Dorado

Respuestas	#	%
Two	19	67,86
One	0	0
Two (and others)	2	7,14
One (and others)	7	25,00
Others	0	0

Opciones disponibles (en ambos casos): Rojo, amarillo, verde, azul, violeta, blanco, marrón, negro, gris, plateado,dorado, naranja y rosa.

Figura 5.17.: *Ejemplos de preguntas y respuestas obtenidas en la encuesta sobre el color. Se muestra las etiquetas de cada imagen en la base de datos, un resumen de las respuestas proporcionadas y las opciones disponibles.*

rojo no está etiquetado en la base de datos. En la figura de la derecha la mayoría de los encuestados seleccionaron las dos opciones correctas (algunos de ellos incluyendo otras opciones) y 7 solo acertaron una correcta al confundir entre amarillo, dorado y marrón.

Como puede verse en estos ejemplos, las personas pueden apreciar el color de manera diferente, ya sea por la propia naturaleza del individuo, por el tono asignado al color (como en el caso de la figura de la derecha, donde se confunden amarillo, dorado y marrón), o por defectos relacionados con los medios de producción, como en el caso de la primera figura, donde el negro se puede confundir con azul oscuro. Otra fuente de errores está en el proceso de etiquetado, ya que a veces solo se etiquetan los colores que se consideran representativos de la marca.

La Figura 5.18 muestra dos logos de ejemplo presentes en la encuesta sobre forma. La figura de la izquierda tiene 3 etiquetas. De estas, 12 de los encuestados (de un total de 39) solo acertó una (8 de ellas con otras opciones incorrectas), 19 detectaron dos (la mayoría de ellos incluyeron también opciones incorrectas), y solo 5 seleccionaron las tres correctas (todas ellas con otras opciones incorrectas). El caso de la figura de la derecha es similar, ya que las respuestas son múltiples y con combinaciones muy diferentes. Cabe destacar que en este caso la imagen contiene “Sólidos geométricos

(a) Etiquetas: Círculos o elipses; Segmentos o partes de círculos, elipses; Líneas o bandas

Respuestas	#	%
Three	0	0
Two	6	15,38
One	4	10,26
Three (and others)	5	12,82
Two (and others)	13	33,33
One (and others)	8	20,51
Others	3	7,69

(b) Etiquetas: Círculos o elipses; Cuadriláteros; Sólidos geométricos

Respuestas	#	%
Three	0	0
Two	1	3,57
One	1	3,57
Three (and others)	3	10,71
Two (and others)	9	32,14
One (and others)	7	25,00
Others	7	25,00

Opciones: Círculos o elipses; Segmentos o partes de círculos, elipses; Triángulos, lineas formando ángulos; Cuadriláteros; Otros polígonos o figuras geométricas; Diferentes figuras geométricas superpuestas o unidas; Líneas o bandas; Sólidos geométricos (objetos 3D tipo esferas, cubos, cilindros, pirámides...)

Figura 5.18.: *Ejemplos de preguntas y respuestas obtenidas en la encuesta sobre forma. Se muestran las etiquetas de cada imagen en la base de datos, un resumen de las respuestas proporcionadas y las opciones disponibles.*

(objetos 3D)" y solo 5 de las 28 personas que evaluaron este logotipo marcaron esta respuesta.

En los ejemplos mostrados de forma puede apreciarse la complejidad del problema para detectar todas las formas presentes en una imagen. Se puede concluir que detectar la presencia de una forma en la imagen puede ser subjetiva. Además, en ocasiones se aprecia una forma predominante que llama nuestra atención e influye en que no nos fijemos en otras formas que hay en la imagen.

La Figura 5.19 muestra dos logos de ejemplo usados en la encuesta sobre semántica del logo. En la imagen de la izquierda, 9 de los encuestados marcaron las respuestas correctas (seleccionado otra respuesta adicional en dos casos). Analizado individualmente, el 95 % de los encuestados reconoce una de las dos etiquetas (algunos incluyeron otras respuestas además de la correcta), con la mayoría seleccionando otras opciones como "Mobiliario", "Equipo eléctrico" o "Equipos de calefacción, cocción o refrigeración, Máquinas de lavado, Equipos de secado".

(a) Etiquetas: Estrellas o Cometas; Esferas armilares, Planetarios...

Respuestas	#	%
Two	7	33,33
One	8	38,10
Two (and others)	2	9,52
One (and others)	3	14,29
Others	0	0
None	1	4,76

Opciones:
Estrellas o Cometas;
Esferas armilares, Planetarios...;
Sol;
Luna;
Mapas geográficos, planiesferas;
Hojas, ramas con hojas;
Fenomenos naturales (olas, rayos, fuego...) ;
Ninguno de los elementos indicados

(b) Etiquetas: Iluminación, válvulas inalámbricas

Respuestas	#	%
One	2	8,00
One (and others)	7	28,00
Others	14	56,00
None	2	8,00

Opciones:
Iluminación, válvulas inalámbricas;
Instalaciones Sanitarias;
Mobiliario;
Equipo eléctrico;
Equipos de calefacción, cocción...;
Fondos divididos en dos o cuatro;
Ninguno de los elementos indicados

Figura 5.19.: *Ejemplos de preguntas y respuestas obtenidas en la encuesta sobre elementos figurativos. Se muestran las etiquetas de cada imagen en la base de datos, un resumen de las respuestas proporcionadas y las opciones disponibles.*

Con estos ejemplos se muestra que no es trivial el reconocimiento de elementos semánticos en un logo, ya que, como se ha comentado anteriormente, en muchos casos estos elementos se simplifican hasta su máxima expresión y pueden sugerir otras interpretaciones, que con frecuencia dependen del individuo que las percibe y de sus antecedentes culturales y personales.

5.6. Análisis de la representación aprendida

En esta sección se analizan las representaciones aprendidas por los modelos propuestos utilizando para esto técnicas que nos permiten transformar esta información a una representación bidimensional para así analizarla visualmente.

5.6.1. Heatmaps

Los *heatmaps* o mapas de calor nos permiten ver qué zonas de la imagen de entrada son más importantes para la predicción realizada por el sistema aplicando para esto colores que marquen esta relevancia. La Figura 5.20 muestra algunos ejemplos de los mapas de calor obtenidos para distintas características utilizando para ello el algoritmo Grad-CAM (*Gradient-weighted Class Activation Mapping* de (Selvaraju et al., 2017)).

En la figura puede verse que el modelo se centra en las zonas de la imagen donde mejor se aprecian visualmente las características más relevantes. En la primera y última imagen se observa claramente cómo el modelo centra su atención en la zona donde se identifican el color y texto respectivamente. En el caso figurativo la red neuronal se fija en el centro de la imagen que contiene el elemento identificable principal, en este caso una gota, mientras que en la imagen que no tiene texto la red no identifica ninguna zona específica y por tanto no detecta texto.

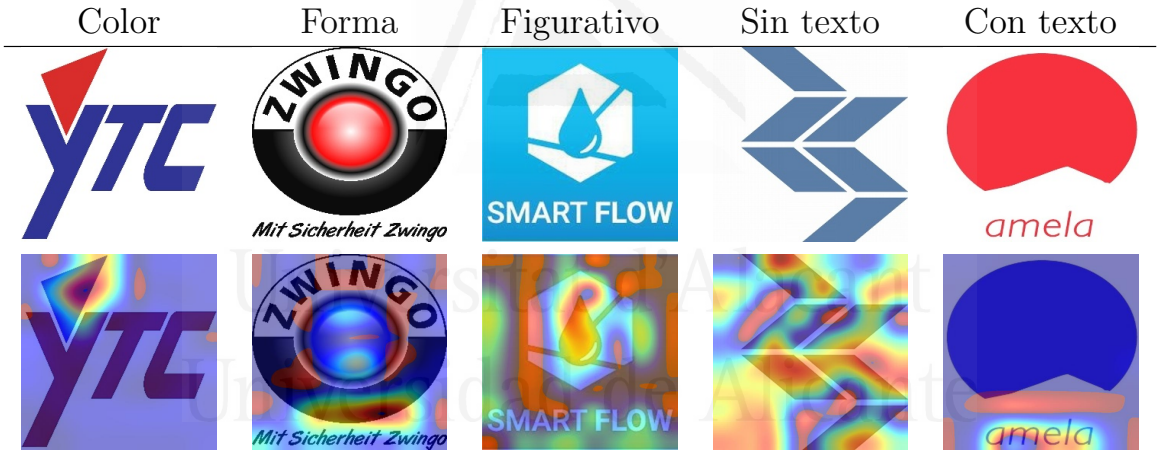


Figura 5.20.: *Heatmaps de imágenes de consulta para las características de color, forma, elementos figurativos y texto utilizando el algoritmo Grad-CAM sobre la capa* `dropout-4` *de la red especializada para cada una de estas características.*

5.6.2. t-SNE

Para analizar las representaciones aprendidas por los modelos se incluye una visualización de la agrupación formada por los NC para las características de color y forma utilizando la técnica t-SNE (*t-Distributed Stochastic Neighbor Embedding*) de van der

Maaten (2008). La Figura 5.21 muestra dos imágenes donde puede verse como, aun siendo un etiquetado tipo multi-etiqueta, los NC tienden a agrupar características similares. Por ejemplo, en el caso del color (imagen superior) puede verse cómo en la parte superior derecha se agrupan los colores grises o plateados, en la parte derecha los azules, en la izquierda los amarillos, marrones y naranjas, y en la parte superior izquierda el verde. Con la forma sucede algo similar (ver figura inferior donde se muestran ampliadas dos áreas de representación). En la parte inferior izquierda se ha realizado zoom en una zona donde se organizan las formas circulares y en la parte derecha los cuadriláteros. Se puede apreciar que los logos que incluyen texto junto a estas formas también se agrupan por separado.

Estos resultados muestran cómo las redes transforman las imágenes de entrada a un nuevo espacio dimensional (los NC extraídos) en el cual los logos similares están cerca. Esto nos permite realizar una búsqueda por similitud basada en la distancia entre las representaciones de los logos en este espacio dimensional y, de esta manera, analizar el espacio de vecindad de una imagen para recuperar imágenes similares.

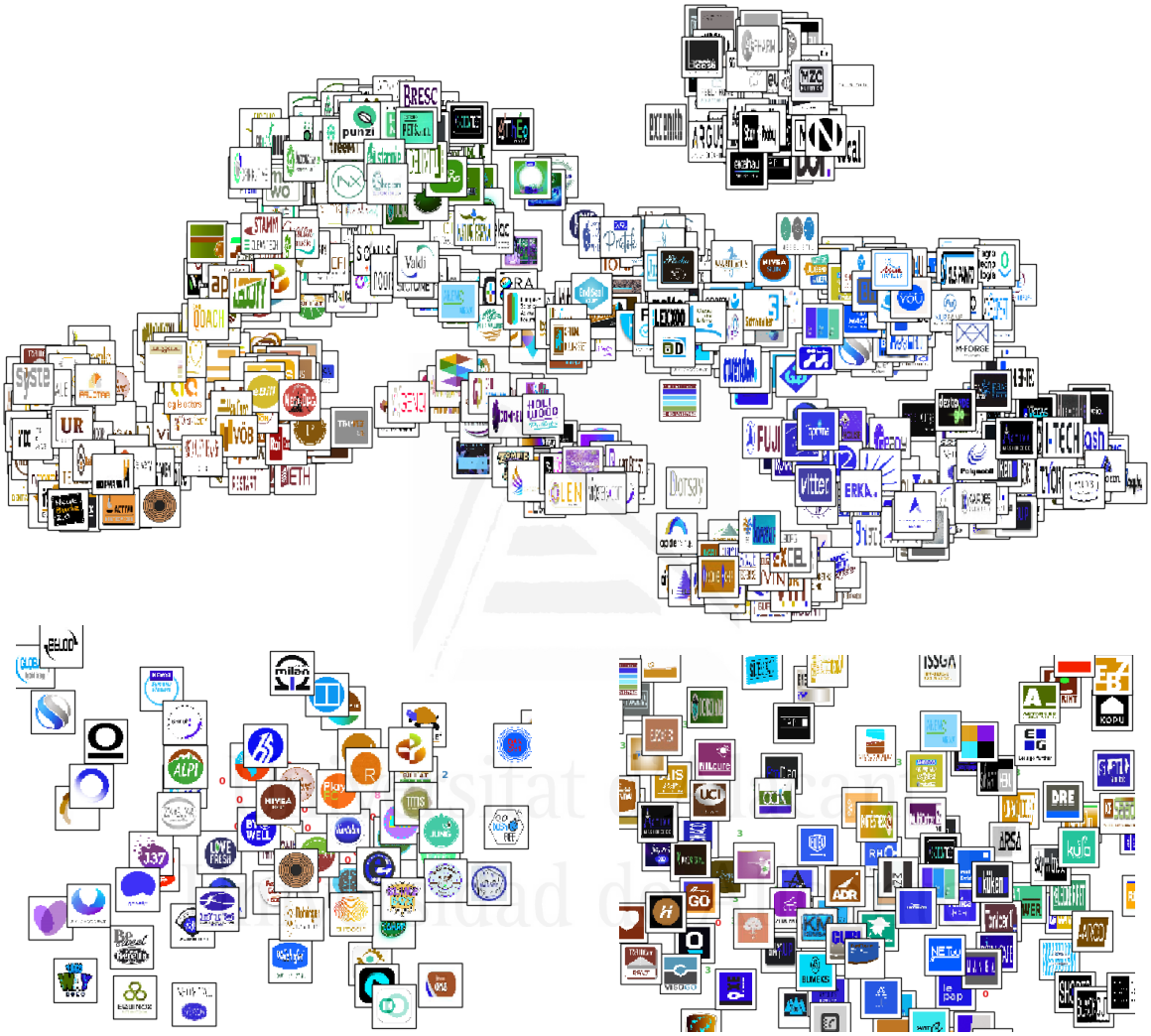


Figura 5.21.: *Agrupaciones formadas por los NC para las características de color (superior) y forma (inferior) utilizando el método t-SNE. En el caso de la forma se incluyen dos imágenes ampliando las áreas en las que se encuentran las formas circulares (izquierda) y cuadriláteras (derecha).*

CAPÍTULO 6

Conclusiones

En esta tesis se han estudiado sistemas para la búsqueda de similitud entre imágenes realizando un recorrido por las distintas técnicas existentes. Para ello hemos partido de los métodos más tradicionales y generales (para cualquier tipo de imágenes) hasta llegar a un sistema de búsqueda por similitud de imagen de marca utilizando redes neuronales.

El trabajo se ha organizado en dos partes principales: por un lado, el estudio y propuesta de técnicas aplicadas a la búsqueda de similitud entre imágenes, revisando la gran evolución que han tenido estas técnicas durante los últimos años, y, por otro lado, la metodología aplicada a la búsqueda y clasificación de logos en la que se han plasmado los conocimientos y la experiencia obtenida en los trabajos previos.

Finalmente se ha presentado una metodología para clasificación multi-etiqueta de logos, considerando las principales características utilizadas para el etiquetado de este tipo de imágenes, tales como son: color, forma, semántica y texto. Además de esta clasificación, el método propuesto permite obtener un ranking de los logos más similares y facilita que los usuarios seleccionen las características a considerar en el proceso de búsqueda. Hasta donde sabemos, no existen métodos en la literatura que aborden estos dos objetivos, por lo que consideramos que una propuesta de este tipo es de gran interés tanto metodológicamente como de forma práctica para ayudar en múltiples tareas, como pueden ser el etiquetado de logos, la detección de plagios o la búsqueda por similitud de imagen de marca.

6.1. Búsqueda de imágenes similares

En esta tesis, para desarrollar los trabajos orientados a la búsqueda por similitud de imágenes se ha usado la aplicación MirBot, un sistema de reconocimiento de imágenes que se presenta como una aplicación colaborativa para dispositivos móviles y que ha permitido generar una base de datos con miles de imágenes capturadas y etiquetadas por sus usuarios. Todos los trabajos presentados tienen en común el uso de esta base de datos, los cuales, además, se han introducido mostrando su desarrollo cronológicamente, lo que nos ha servido para ilustrar la evolución de las técnicas utilizadas en este tipo de tareas.

Inicialmente, en la aplicación MirBot se utilizaron descriptores invariantes locales, como TOP-SURF, en combinación con características globales como los histogramas de color. Con el objetivo de mejorar estos resultados se ha propuesto un nuevo sistema de verificación geométrica llamado SIIP (*Segment Intersection of Interest Points*), el cual permite comparar la geometría de descriptores locales calculados para dos imágenes sin comprometer demasiado la eficiencia del proceso de búsqueda. Estos resultados se han comparado con RANSAC, el algoritmo más utilizado en su momento, demostrando ser más eficiente con las bases de datos evaluadas. Por último, con la evolución de la tecnología y la aparición del aprendizaje profundo, se han propuesto también técnicas basadas en descriptores neuronales para mejorar los resultados de MirBot.

La evaluación utilizando descriptores tradicionales ha mostrado que la combinación de histogramas de color y características locales puede aumentar la tasa de acierto en comparación con su uso independiente. Asimismo, la inclusión de verificación geométrica también ha conseguido mejoras en los resultados. Sin embargo, el uso de CNN supera notablemente a estos descriptores y, además, su tasa de acierto no disminuye en función del número de imágenes cuando la base de datos crece, a diferencia de lo que ocurre con las características tradicionales.

Con respecto a esta última aproximación, se han evaluado varias arquitecturas CNN del estado del arte partiendo tanto de modelos pre-entrenados con ImageNet como realizando un proceso de ajuste a nuestro conjunto de datos antes de extraer los códigos neuronales. En este caso se han obtenido los mejores resultados al aplicar *fine-tune* sobre los modelos ResNet y Xception. Además, esta experimentación también nos ha permitido concluir que el uso de los códigos neuronales y kNN mejora la estrategia de predicción con SoftMax (es decir, usando directamente la salida de la red), que es comúnmente la más utilizada, y que la normalización de los códigos neuronales

usando ℓ_2 también es beneficiosa.

Las principales contribuciones de esta parte de la tesis han sido: la obtención de un conjunto de datos público (MirBot) que crece continuamente con las aportaciones de sus usuarios y que contiene imágenes etiquetadas con una serie de metadatos asociados; un análisis de diferentes descriptores visuales, tanto características tradicionales como códigos neuronales usando diferentes topologías de CNN; y la conclusión de que el uso de códigos neuronales normalizados de la última capa oculta con kNN en la etapa de predicción mejora los resultados de SoftMax en todos los casos evaluados.

La experiencia aprendida en estos trabajos ha servido de base para la siguiente tarea que nos planteamos, la búsqueda y clasificación de logos.

6.2. Búsqueda y clasificación de logos

En esta segunda parte se ha abordado la problemática de búsqueda y clasificación de imagen de marca como un caso particular de imagen, ya que tiene unas características propias y, por tanto, requiere un tratamiento específico. La mayoría de sistemas TIR se basan en la comparación de la imagen en su conjunto, asumiendo que los logos de la misma marca se parecen entre sí, pero esto no siempre es así dado que las marcas evolucionan con frecuencia y adaptan su imagen a las tendencias en diseño. Para solucionar este problema se ha propuesto un sistema que plantea la clasificación de los logos basándose en las características que lo definen, como son: el color, la forma o los objetos reconocibles que contiene. Dado que además un logo puede presentar de manera simultánea varias etiquetas para cada una de estas características, se ha tratado este problema como una tarea de clasificación multi-etiqueta.

La metodología propuesta combina, de manera ponderada, la representación aprendida por una serie de redes CNN de clasificación multi-etiqueta que se especializan en detectar las características más distintivas de los logos. Además, el método realiza una etapa de pre-procesado para eliminar fondos uniformes y texto de las imágenes de entrada. Según los experimentos realizados, la eliminación del texto del logo ayuda a la clasificación de la forma, aunque no con otro tipo de características. Esto puede deberse a que el texto a menudo incluye características representativas del logo, como el color, o puede combinarse con elementos figurativos, pero, sin embargo, por otro lado también dificulta el análisis general de su forma, puesto que la propia forma del

texto no se suele considerar en la anotación.

La experimentación realizada muestra que el enfoque propuesto obtiene buenos resultados, tanto para clasificación como para búsqueda por similitud. La comparación realizada con 17 métodos TIR existentes publicados recientemente demuestra cómo la propuesta mejora notablemente los enfoques anteriores, especialmente al considerar la combinación del color y la forma.

Otra cuestión importante que se ha tenido que abordar es la caracterización de bases de datos de este tipo. Las propuestas de etiquetado disponibles o bien resultaban muy simplistas e ineficaces (como las basadas en la marca), o bien demasiado densas, llegando a ser ambiguas, como la clasificación Viena. Dado que esta última es la comúnmente aceptada en las oficinas de marcas y patentes internacionales, se decidió estudiarla más en profundidad y realizar una propuesta para adaptarla a su uso con sistemas de aprendizaje automático.

A este respecto, se han encontrado varias problemáticas en el etiquetado de logos en las distintas bases de datos consideradas, ya que normalmente solo están etiquetadas las características más distintivas de la marca, lo que da como resultado un etiquetado incompleto y, a menudo, incoherente. Estos problemas vienen inducidos por el propio proceso de etiquetado o motivados por la propia codificación de Viena, ya que es una categorización cerrada y existen ciertas características que son difíciles de definir, considerando además que la semántica de las marcas puede ser subjetiva. Una de las ventajas que ofrece la metodología propuesta es que puede ayudar en esta tarea, sugiriendo una clasificación inicial con criterios homogéneos que, además de facilitar el trabajo, es más completa y exhaustiva.

Dado que en el proceso de etiquetado intervienen muchas personas diferentes, un método de clasificación automática como el propuesto podrá reducir las inconsistencias causadas por los diferentes factores que influyen en la percepción humana de una misma representación visual y también por la dificultad de expresar las cualidades gráficas mediante palabras.

Para comprobar esta afirmación y analizar la consistencia del etiquetado se realizó una encuesta a diseñadores y expertos en diseño. Este experimento mostró que la metodología propuesta proporciona un mejor etiquetado que el que un operador humano asignaría manualmente, incluso en el caso de personas familiarizadas con esta tarea, y por tanto se podría utilizar el etiquetado inicialmente propuesto por el sistema siendo solo necesaria la supervisión del operador. Además, estudiantes y profesionales

del diseño también podrán utilizar el sistema como una ayuda en las propias tareas de diseño o para la búsqueda de referencias, estilos, detección de logos similares o plagios.

CAPÍTULO 7

Bibliografía

Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 510–517, 06 2012. doi: 10.1109/CVPR.2012.6247715.

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

Sugata Banerji, Atreyee Sinha, and Chengjun Liu. New image descriptors based on color, texture, shape, and wavelets for object and scene image classification. *Neurocomputing*, 117:173 – 185, 2013. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2013.02.014. URL http://www.sciencedirect.com/science/article/pii/S0925231213001987.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014. URL http://dx.doi.org/10.1016/j.cviu.2007.09.014.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 153–160, Cambridge, MA, USA, 2006. MIT Press.

M. Bernabeu, A. Pertusa, and A.J. Gallego. Image spatial verification using segment intersection of interest points. In *Proc. of the 24 Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, May 2016. ISBN 2464-4614.

E. Bernhardsson. Annoy: Approximate nearest neighbors in c ++ /python op- timized for memory usage and loading/saving to disk, 2016. URL https://github.com/spotify/annoy.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition, 37(9)*, pages 1757–1771, 2004.

L. Breiman, Jerome H. Friedman, Richard A. Olshen, and C. J. Stone. Classification and regression trees. 1983.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15560-X, 978-3-642-15560-4. URL http://dl.acm.org/citation.cfm?id=1888089.1888148.

Jiangzhong Cao, Yunfei Huang, Qingyun Dai, and Wing-Kuen Ling. Unsupervised trademark retrieval method based on attention mechanism. *Sensors*, 21(5), 2021. ISSN 1424-8220. URL https://www.mdpi.com/1424-8220/21/5/1894.

Capsule. *Design Matters: Logos 01:An Essential Primer for Today's Competitive Market*. Rockport Publishers, 2007. ISBN 978-1592533411.

N. Chaves and R. Belluccia. *La Marca Corporativa: Gestión y Diseño de Símbolos y Logotipos*. Estudios de Comunicación Series. Paidós, 2003. ISBN 9789501227178.

Norberto Chaves. La marca: señal, nombre, identidad y blasón. *EME Experimental Illustration, Art —& Design*, 3(3):40–49, 2015. ISSN 2341-3018. doi: 10.4995/eme.2015.3432. URL https://polipapers.upv.es/index.php/eme/article/view/3432.

Bernard Chazelle and Herbert Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 590–600, 1988.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015. URL http://arxiv.org/abs/1512.01274.

Jia-Han Chiam. Brand logo classification. Technical report, Stanford University, 2015.

François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017. URL http://arxiv.org/abs/1610.02357.

Ondrej Chum, T. Werner, and J. Matas. Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 112–115 Vol.1, 2004. doi: 10.1109/ICPR.2004.1334020.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995. URL https://doi.org/10.1007/BF00994018.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Hang Dong, Wei Wang, Kaizhu Huang, and Frans Coenen. Automated social text annotation with joint multi-label attention networks. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–15, 06 2020. doi: 10.1109/TNNLS.2020.3002798.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification, 2nd Edition*. Wiley, 2001. ISBN 9780471056690.

J.P. Eakins, J.M. Boardman, and M.E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998. doi: 10.1109/93.682526.

Margaret Eakins, John; Graham. Content-based image retrieval. university of northumbria at newcastle, 1999. URL https://www.inf.fu-berlin.de/lehre/WS00/webIS/reader/WebIR/imageRetrievalOverview.pdf.

Ioannis Vlahavas Eleftherios Spyromitros, Grigorios Tsoumakas. An empirical study of lazy multilabel classification algorithms. In *Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008)*, 2008.

EUTM. European Union Trademark. https://euipo.europa.eu/ohimportal/en/open-data, 2020. Accessed: 20/5/2020.

Yitong Feng, Cunzhao Shi, Chengzuo Qi, Jian Xu, Baihua Xiao, and Chunheng Wang. Aggregation of reversal invariant features from edge images for large-scale trademark retrieval. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 384–388, 2018. doi: 10.1109/ICCAR.2018.8384705.

B. Fernando, E. Fromont, D. Muselet, and M. Sebban. Discriminative feature fusion for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3434–3441, June 2012. doi: 10.1109/CVPR.2012.6248084.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692.

Antonio-Javier Gallego, Jorge Calvo-Zaragoza, Jose J. Valero-Mas, and Juan R. Rico-Juan. Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation. *Pattern Recognition*, 74:531–543, 2018a. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.09.038.

Antonio-Javier Gallego, Antonio Pertusa, and Jorge Calvo-Zaragoza. Improving convolutional neural networks' accuracy in noisy environments using k-nearest neighbors. *Applied Sciences*, 8(11), 2018b. ISSN 2076-3417.

Antonio-Javier Gallego, Jorge Calvo-Zaragoza, and Juan Ramón Rico-Juan. Insights into efficient k-nearest neighbor classification with convolutional neural codes. *IEEE Access*, 8:99312–99326, 2020. doi: 10.1109/ACCESS.2020.2997387.

Geonames. Geonames feature codes. URL http://www.geonames.org/export/codes.html.

Souvik Ghosh and Ranjan Parekh. Article: Automated color logo recognition system based on shape and color features. *International Journal of Computer Applications*, 118(12):13–20, May 2015.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1458–1465 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.239.

Conny Gu. Trademarks image database (traidmarks). 2014. URL http://tc11.cvc.uab.es/datasets/TraidMarks_1.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Geoffrey E. Hinton and Richard S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems*, pages 3–10, 1994.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 07 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Forrest N. Iandola, Anting Shen, Peter Gao, and Kurt Keutzer. Deeplogo: Hitting logo recognition with the deep neural network hammer. *CoRR*, abs/1510.02131, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87:316–336, 2009.

S. Jeong. Histogram-based color image retrieval, psych221/ee362 project report. Stanford, 2001. URL http://pdfs.semanticscholar.org/e884/2a22aa486fd273606fcd5f8090619bbb468c.pdf.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, 06 2014. doi: 10.1145/2647868.2654889.

Xin Jin and Jiawei Han. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_425. URL https://doi.org/10.1007/978-0-387-30164-8_425.

Ian Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_455. URL https://doi.org/10.1007/978-3-642-04898-2_455.

Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 581–584, 2009.

Chia-Feng Juang, Wen-Kai Sun, and Guo-Cyuan Chen. Object detection by color histogram-based fuzzy classifier with support vector learning. *Neurocomputing*, 72(10):2464 – 2476, 2009. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2008.11.016. URL http://www.sciencedirect.com/science/article/pii/S092523120800533X. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Recent advances in large scale image search. pages 305–326, 01 2008. doi: 10.1007/978-3-642-00826-9_14.

Yannis Kalantidis, Lluis Garcia Pueyo, Michele Trevisiol, Roelof van Zwol, and Yannis Avrithis. Scalable triangulation-based logo recognition. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 20:1–20:7, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0336-1. doi: 10.1145/1991996.1992016. URL http://doi.acm.org/10.1145/1991996.1992016.

Toshikazu Kato. Database architecture for content-based image retrieval. In Albert A. Jamberdino and Carlton Wayne Niblack, editors, *Image Storage and Retrieval Systems*, volume 1662, pages 112 – 123. International Society for Optics and Photonics, SPIE, 1992. URL https://doi.org/10.1117/12.58497.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012. doi: 10.1145/3065386.

Martin Köstinger, Peter M. Roth, and Horst Bischof. Planar trademark and logo retrieval, 2013.

Tian Lan, Xiaoyi Feng, Zhaoqiang Xia, Shijie Pan, and Jinye Peng. Similar trademark image retrieval integrating lbp and convolutional neural network. In Yao Zhao, Xiangwei Kong, and David Taubman, editors, *Image and Graphics*, pages 231–242, Cham, 2017. Springer International Publishing.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.68. URL http://dx.doi.org/10.1109/CVPR.2006.68.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 12 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL https://doi.org/10.1162/neco.1989.1.4.541.

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

Zhang Lei, Lin Fuzong, and Zhang Bo. A cbir method based on color-spatial feature. In *Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99. 'Multimedia Technology for Asia-Pacific Information Infrastructure' (Cat. No.99CH37030)*, volume 1, pages 166–169 vol.1, 1999. doi: 10.1109/TENCON.1999.818376.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, January 1991. ISSN 0018-9448. doi: 10.1109/18.61115. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=61115.

David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–, 11 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

Piyush Madan and Samaya Madhavan. An introduction to deep learning. 2020. URL https://developer.ibm.com/articles/an-introduction-to-deep-learning/.

B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8): 837–842, Aug 1996. ISSN 0162-8828. doi: 10.1109/34.531803.

Jonathan Masci, Ueli Meier, Dan Ciresan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. pages 52–59, 06 2011. ISBN 978-3-642-21734-0. doi: 10.1007/978-3-642-21735-7_7.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943. ISSN 1522-9602. doi: 10.1007/BF02478259. URL https://doi.org/10.1007/BF02478259.

Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000027790.02288.f2. URL https://doi.org/10.1023/B:VISI.0000027790.02288.f2.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *Lecture Notes in Computer Science*, volume 1842, pages 404–420, 06 2000. ISBN 978-3-540-67685-0. doi: 10.1007/3-540-45054-8_27.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:971–987, 2002.

Peter Oram. Wordnet: An electronic lexical database. christiane fellbaum (ed.). cambridge, ma: Mit press, 1998. pp. 423. -. *Applied Psycholinguistics*, 22(1):131–134, 2001.

Javier Ortega-Bastida, Antonio-Javier Gallego, and Antonio Pertusa. Multimodal object recognition using deep learning representations extracted from images and smartphone sensors. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 521–529, 2019. ISBN 978-3-030-13469-3.

C. A. Perez, P. A. Estévez, F. J. Galdames, D. A. Schulz, J. P. Perez, D. Bastías, and D. R. Vilar. Trademark Image Retrieval Using a Combination of Deep Convolutional Neural Networks. In *Int. Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2018.

Antonio Pertusa, Antonio-Javier Gallego, and Marisa Bernabeu. Mirbot: A multimodal interactive image retrieval system. In João M. Sanches, Luisa Micó, and Jaime S. Cardoso, editors, *Pattern Recognition and Image Analysis*, pages 197–204, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Antonio Pertusa, Antonio-Javier Gallego, and Marisa Bernabeu. Mirbot: A collaborative object recognition system for smartphones using convolutional neural networks. *Neurocomputing*, 293:87–99, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.03.005. URL https://www.sciencedirect.com/science/article/pii/S0925231218302704.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

C. Pornpanomchai, P. Boonsripornchai, P. Puttong, and C. Rattananirundorn. Logo recognition system. In *2015 International Computer Science and Engineering Conference (ICSEC)*, pages 1–6, Nov 2015. doi: 10.1109/ICSEC.2015.7401394.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Stefan Romberg, Lluis Pueyo, Rainer Lienhart, and Roelof Zwol. Scalable logo recognition in real-world images. page 25, 01 2011. doi: 10.1145/1991996.1992021.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33832-2, 978-3-540-33832-1. doi: 10.1007/11744023_34. URL http://dx.doi.org/10.1007/11744023_34.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, Nov 2011. doi: 10.1109/ICCV.2011.6126544.

Marçal Rusiñol and Josep Lladós. Efficient logo retrieval through hashing shape context descriptors. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 215–222, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-773-8. doi: 10.1145/1815330.1815358. URL http://doi.acm.org/10.1145/1815330.1815358.

Marçal Rusiñol, David Aldavert, Dimosthenis Karatzas, Ricardo Toledo, and Josep Lladós. Interactive trademark image retrieval by fusing semantic and visual content. In Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, pages 314–325, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Alexander Sage, Radu Timofte, Eirikur Agustsson, and Luc Van Gool. Logo synthesis and manipulation with clustered generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. doi: 10.1109/cvpr.2018.00616. URL http://dx.doi.org/10.1109/CVPR.2018.00616.

H. Sahbi, L. Ballan, G. Serra, and A. Del Bimbo. Context-dependent logo matching and recognition. *IEEE Transactions on Image Processing*, 22(3):1018–1031, March 2013. ISSN 1057-7149. doi: 10.1109/TIP.2012.2226046.

Gerard Salton and Michael McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, NY, 1983.

Jan Schietse, John Eakins, and Remco Veltkamp. Practice and challenges in trademark image retrieval. pages 518–524, 07 2007. doi: 10.1145/1282280.1282355.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

Z. Tao, B. Wang, W. Wang, L. Yang, and Q. Zhou. Spatial feature collaborative network for trademark image retrieval. In *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 144–148, 2019. doi: 10.1109/CCIS48116.2019.9073730.

Tom Taulli. *Artificial Intelligence Basics: A Non-Technical Introduction*. 01 2019. ISBN 978-1-4842-5027-3. doi: 10.1007/978-1-4842-5028-0.

Bart Thomee, Erwin M. Bakker, and Michael S. Lew. Top-surf: A visual words toolkit. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1473–1476, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874250. URL http://doi.acm.org/10.1145/1873951.1874250.

Konstantinos et al Trohidis. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, 4, 2011. doi: 10.1186/1687-4722-2011-426793.

Sam S. Tsai, David Chen, Gabriel Takacs, Vijay Chandrasekhar, Ramakrishna Vedantham, Radek Grzeszczuk, and Bernd Girod. Fast geometric re-ranking for image-based retrieval. In *2010 IEEE International Conference on Image Processing*, pages 1029–1032, 2010. doi: 10.1109/ICIP.2010.5648942.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

Volodymyr Turchenko, Eric Chalmers, and Artur Luczak. A deep convolutional auto-encoder with pooling - unpooling layers in caffe. *International Journal of Computing*, 18, 01 2017.

Osman Tursun, Cemal Aker, and Sinan Kalkan. A large-scale dataset and benchmark for similar trademark retrieval. *CoRR*, abs/1701.05766, 2017.

Osman Tursun, Simon Denman, Sabesan Sivapalan, Sridha Sridharan, Clinton Fookes, and Sandra Mau. Component-based attention for large-scale trademark retrieval. *IEEE Transactions on Information Forensics and Security*, page 1–1, 2020. ISSN 1556-6021. doi: 10.1109/tifs.2019.2959921.

Andras Tüzkö, Christian Herrmann, Daniel Manger, and Jürgen Beyerer. Open set logo detection and retrieval. *CoRR*, abs/1710.10891, 2017. URL http://arxiv.org/abs/1710.10891.

USPTO. United States Patent and Trademark Office. https://www.uspto.gov/trademarks, 2020. Accessed: 20/5/2020.

K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, Sept 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.154.

Geoffrey van der Maaten, Laurens; Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9(nov):2579–2605, 2008.

Wenmei Wang, Xinxin Xu, Jianglong Zhang, LiFang Yang, Gege Song, and Xianglin Huang. Trademark image retrieval based on faster r-CNN. *Journal of Physics: Conference Series*, 1237:032042, jun 2019. doi: 10.1088/1742-6596/1237/3/032042. URL https://doi.org/10.1088/1742-6596/1237/3/032042.

Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 331–340, 2018.

Chia-Hung Wei, Yue Li, Wing-Yin Chau, and Chang-Tsun Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386 – 394, 2009. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2008.08.019. URL http://www.sciencedirect.com/science/article/pii/S0031320308003324.

A. Wheeler. *Designing Brand Identity: An Essential Guide for the Whole Branding Team*. Wiley, 2013. ISBN 9781118099209.

World Intellectual Property Organization. *International Classification of the Figurative Elements of Marks: (Vienna Classification)*. WIPO publication. World Intellectual Property Organization, 2002. ISBN 9789280510546.

J.K. Wu. Content-based retrieval for trademark registration. *Multimedia Tools and Applications*, 3:245–267, 1996. doi: 10.1007/BF00393940.

Zhaoqiang Xia, Jie Lin, and Xiaoyi Feng. Trademark image retrieval via transformation-invariant deep hashing. *Journal of Visual Communication and Image Representation*, 59:108–116, 2019. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2019.01.011. URL https://www.sciencedirect.com/science/article/pii/S1047320319300112.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 26:1819–1837, 08 2014. doi: 10.1109/TKDE.2013.39.

APÉNDICE A

Lista de Acrónimos

AA Aprendizaje Automático

AE Auto-Encoder

API Application Programming Interfaces

BOF Bag-Of-Features

BRIEF Binary Robust Independent Elementary Features

CART Classification and Regression Trees

CBIR Content-based Image Retrieval

CNN Convolutional Neural Network

CAE Convolutional Autoencoder

DAE Denoising Autoencoder

DL Deep Learning

DLSI Departamento de Lenguajes y Sistemas Informáticos

DOF Degrees Of Freedom

EUIPO European Union Intellectual Property Office

EUTM European Union Trademarks

FAST Features from Accelerated Segment Test

FREAK Fast Retina Keypoint

GAN Generative Adversarial Networks

GPU Graphics Processing Unit

GT Ground Truth

HOG Histogram of Oriented Gradients

IA Inteligencia Artificial

JSD Divergencia de Jensen-Shannon

KNN k-Nearest Neighbors

LBP Local Binary Pattern

LR Label Ranking

LSTM Long Short Term Memory

MAE Mean Absolute Error

ML Machine Learning

MLC Multi Label Classification

MSE Mean Squared Error

NAR Normalized Average Rank

NC Neural Codes

NN Neural Network

ORB Oriented fAST and Rotate BRIEF

PCA Principal Components Analysis

R-CNN Region Based Convolutional Neural Networks

RANSAC Random Sample Consensus

ReLU Rectified Linear Unit

RNN Recurent Neural Network

ROI Region Of Interest

SIFT Scale Invariant Feature Transform

SIIP Segment Intersection of Interest Points

SPM Spatial Pyramid Matching

SURF Speeded-Up Robust Features

SVM Support Vector Machine

TIR Trademark Image Retrieval

USPTO United States Patent and Trademark Office

WIPO World Intellectual Property Organization

APÉNDICE B

Clasificación de Viena

A continuación incluimos la lista de etiquetas principales en la clasificación de Viena (World Intellectual Property Organization, 2002).

1	Celestial bodies, Natural Phenomena, Geographical Maps.
2	Human beings.
3	Animals.
4	Supernatural, fabulous, fantastic or unidentifiable Beings.
5	Plants.
6	Landscapes.
7	Constructions, structures for advertisements, gates or Barriers.
8	Foodstuffs.
9	Textiles, clothing, sewing accessories, headwear, footwear.
10	Tobacco, smokers' requisites, matches, travel goods, fans, toilet articles.
11	Household utensils.
12	Furniture, sanitary installations.
13	Lighting, wireless valves, heating, cooking or refrigerating equipment, washing machines, drying equipment.
14	Ironmongery, tools, ladders.
15	Machinery, motors, engines
16	Telecommunications, sound recording or reproduction, computers, photography, cinematography, optics.
17	Horological instruments, jewelry, weights and measures.
18	Transport, equipment for animals.
19	Containers and packing, representations of miscellaneous products.
20	Writing, drawing or painting materials, office requisites, stationery and booksellers' goods.
21	Games, toys, sporting articles, roundabouts.
22	Musical instruments and their accessories, music accessories, bells, pictures, sculptures.

23	Arms, ammunition, armour.
24	Heraldry, coins, emblems, symbols.
25	Ornamental motifs, surfaces or backgrounds with ornaments.
26	Geometrical figures and solids.
27	Forms of writing, numerals.
28	Inscriptions in various characters.
29	Colors.

APÉNDICE C

Clasificación de Niza

La Clasificación de Niza propone un etiquetado que divide los Bienes y Servicios en las siguientes 45 subcategorías[1]:

C.1. Bienes

Clase 1 Productos químicos para la industria, la ciencia y la fotografía, así como para la agricultura, la horticultura y la silvicultura; resinas artificiales en bruto, materias plásticas en bruto; compuestos para la extinción de incendios y la prevención de incendios; preparaciones para templar y soldar metales; sustancias para curtir cueros y pieles de animales; adhesivos (pegamentos) para la industria; masillas y otras materias de relleno en pasta; compost, abonos, fertilizantes; preparaciones biológicas para la industria y la ciencia.

Clase 2 Pinturas, barnices, lacas; productos contra la herrumbre y el deterioro de la madera; colorantes, tintes; tintas de imprenta, tintas de marcado y tintas de grabado; resinas naturales en bruto; metales en hojas y en polvo para la pintura, la decoración, la imprenta y trabajos artísticos.

Clase 3 Productos cosméticos y preparaciones de tocador no medicinales; dentífricos no medicinales; productos de perfumería, aceites esenciales; preparaciones para blanquear y otras sustancias para lavar la ropa; preparaciones para limpiar, pulir, desengrasar y raspar.

Clase 4 Aceites y grasas para uso industrial, ceras; lubricantes; compuestos para absorber, rociar y asentar el polvo; combustibles y materiales de alumbrado; velas y mechas de iluminación.

Clase 5 Productos farmacéuticos, preparaciones para uso médico y veterinario; produc-

[1] https://www.wipo.int/classifications/nice/en/

tos higiénicos y sanitarios para uso médico; alimentos y sustancias dietéticas para uso médico o veterinario, alimentos para bebés; suplementos alimenticios para personas o animales; emplastos, material para apósitos; material para empastes e impresiones dentales; desinfectantes; productos para eliminar animales dañinos; fungicidas, herbicidas.

Clase 6 Metales comunes y sus aleaciones, menas; materiales de construcción y edificación metálicos; construcciones transportables metálicas; cables e hilos metálicos no eléctricos; pequeños artículos de ferretería metálicos; contenedores metálicos de almacenamiento y transporte; cajas de caudales.

Clase 7 Máquinas, máquinas herramientas y herramientas mecánicas; motores, excepto motores para vehículos terrestres; acoplamientos y elementos de transmisión, excepto para vehículos terrestres; instrumentos agrícolas que no sean herramientas de mano que funcionan manualmente; incubadoras de huevos; distribuidores automáticos.

Clase 8 Herramientas e instrumentos de mano que funcionan manualmente; artículos de cuchillería, tenedores y cucharas; armas blancas; maquinillas de afeitar.

Clase 9 Aparatos e instrumentos científicos, de investigación, de navegación, geodésicos, fotográficos, cinematográficos, audiovisuales, ópticos, de pesaje, de medición, de señalización, de detección, de pruebas, de inspección, de salvamento y de enseñanza; aparatos e instrumentos de conducción, distribución, transformación, acumulación, regulación o control de la distribución o del consumo de electricidad; aparatos e instrumentos de grabación, transmisión, reproducción o tratamiento de sonidos, imágenes o datos; soportes grabados o descargables, software, soportes de registro y almacenamiento digitales o análogos vírgenes; mecanismos para aparatos que funcionan con monedas; cajas registradoras, dispositivos de cálculo; ordenadores y periféricos de ordenador; trajes de buceo, máscaras de buceo, tapones auditivos para buceo, pinzas nasales para submarinistas y nadadores, guantes de buceo, aparatos de respiración para la natación subacuática; extintores.

Clase 10 Aparatos e instrumentos quirúrgicos, médicos, odontológicos y veterinarios; miembros, ojos y dientes artificiales; artículos ortopédicos; material de sutura; dispositivos terapéuticos y de asistencia para personas discapacitadas; aparatos de masaje; aparatos, dispositivos y artículos de puericultura; aparatos, dispositivos y artículos para actividades sexuales.

Clase 11 Aparatos e instalaciones de alumbrado, calefacción, enfriamiento, producción de vapor, cocción, secado, ventilación y distribución de agua, así como instalaciones sanitarias.

Clase 12 Vehículos; aparatos de locomoción terrestre, aérea o acuática.

Clase 13 Armas de fuego; municiones y proyectiles; explosivos; fuegos artificiales.

Clase 14 Metales preciosos y sus aleaciones; artículos de joyería, piedras preciosas y semipreciosas; artículos de relojería e instrumentos cronométricos.

Clase 15 Instrumentos musicales; atriles para partituras y soportes para instrumentos musicales; batutas.

Clase 16 Papel y cartón; productos de imprenta; material de encuadernación; fotografías; artículos de papelería y artículos de oficina, excepto muebles; adhesivos (pegamentos) de papelería o para uso doméstico; material de dibujo y material para artistas; pinceles; material de instrucción y material didáctico; hojas, películas y bolsas de materias plásticas para embalar y empaquetar; caracteres de imprenta, clichés de imprenta.

Clase 17 Caucho, gutapercha, goma, amianto y mica en bruto o semielaborados, así como sucedáneos de estos materiales; materias plásticas y resinas en forma extrudida utilizadas en procesos de fabricación; materiales para calafatear, estopar y aislar; tuberías, tubos y mangueras flexibles no metálicos.

Clase 18 Cuero y cuero de imitación; pieles de animales; artículos de equipaje y bolsas de transporte; paraguas y sombrillas; bastones; fustas, arneses y artículos de guarnicionería; collares, correas y ropa para animales.

Clase 19 Materiales de construcción no metálicos; tuberías rígidas no metálicas para la construcción; asfalto, pez, alquitrán y betún; construcciones transportables no metálicas; monumentos no metálicos.

Clase 20 Muebles, espejos, marcos; contenedores no metálicos de almacenamiento o transporte; hueso, cuerno, ballena o nácar, en bruto o semielaborados; conchas; espuma de mar; ámbar amarillo.

Clase 21 Utensilios y recipientes para uso doméstico y culinario; utensilios de cocina y vajilla, excepto tenedores, cuchillos y cucharas; peines y esponjas; cepillos; materiales para fabricar cepillos; material de limpieza; vidrio en bruto o semielaborado, excepto vidrio de construcción; artículos de cristalería, porcelana y loza.

Clase 22 Cuerdas y cordeles; redes; tiendas de campaña y lonas; toldos de materias textiles o sintéticas; velas de navegación; sacos para el transporte y almacenamiento de mercancías a granel; materiales de acolchado y relleno, excepto papel, cartón, caucho o materias plásticas; materias textiles fibrosas en bruto y sus sucedáneos.

Clase 23 Hilos e hilados para uso textil.

Clase 24 Tejidos y sus sucedáneos; ropa de hogar; cortinas de materias textiles o de materias plásticas.

Clase 25 Prendas de vestir, calzado, artículos de sombrerería.

Clase 26 Encajes, cordones y bordados, así como cintas y lazos de mercería; botones,

ganchos y ojetes, alfileres y agujas; flores artificiales; adornos para el cabello; cabello postizo.

Clase 27 Alfombras, felpudos, esteras y esterillas, linóleo y otros revestimientos de suelos; tapices murales que no sean de materias textiles.

Clase 28 Juegos y juguetes; aparatos de videojuegos; artículos de gimnasia y deporte; adornos para árboles de Navidad.

Clase 29 Carne, pescado, carne de ave y carne de caza; extractos de carne; frutas y verduras, hortalizas y legumbres en conserva, congeladas, secas y cocidas; jaleas, confituras, compotas; huevos; leche, quesos, mantequilla, yogur y otros productos lácteos; aceites y grasas para uso alimenticio.

Clase 30 Café, té, cacao y sus sucedáneos; arroz, pastas alimenticias y fideos; tapioca y sagú; harinas y preparaciones a base de cereales; pan, productos de pastelería y confitería; chocolate; helados cremosos, sorbetes y otros helados; azúcar, miel, jarabe de melaza; levadura, polvos de hornear; sal, productos para sazonar, especias, hierbas en conserva; vinagre, salsas y otros condimentos; hielo.

Clase 31 Productos agrícolas, acuícolas, hortícolas y forestales en bruto y sin procesar; granos y semillas en bruto o sin procesar; frutas y verduras, hortalizas y legumbres frescas, hierbas aromáticas frescas; plantas y flores naturales; bulbos, plantones y semillas para plantar; animales vivos; productos alimenticios y bebidas para animales; malta.

Clase 32 Cervezas; bebidas sin alcohol; aguas minerales y carbonatadas; bebidas a base de frutas y zumos de frutas; siropes y otras preparaciones para elaborar bebidas sin alcohol.

Clase 33 Bebidas alcohólicas, excepto cervezas; preparaciones alcohólicas para elaborar bebidas.

Clase 34 Tabaco y sucedáneos del tabaco; cigarrillos y puros; cigarrillos electrónicos y vaporizadores bucales para fumadores; artículos para fumadores; cerillas.

C.2. Servicios

Clase 35 Publicidad; gestión, organización y administración de negocios comerciales; trabajos de oficina.

Clase 36 Servicios financieros, monetarios y bancarios; servicios de seguros; negocios inmobiliarios.

Clase 37 Servicios de construcción; servicios de instalación y reparación; extracción minera, perforación de gas y de petróleo.

Clase 38 Servicios de telecomunicaciones.

Clase 39 Transporte; embalaje y almacenamiento de mercancías; organización de viajes.

Clase 40 Tratamiento de materiales; reciclaje de residuos y desechos; purificación del aire y tratamiento del agua; servicios de impresión; conservación de alimentos y bebidas.

Clase 41 Educación; formación; servicios de entretenimiento; actividades deportivas y culturales.

Clase 42 Servicios científicos y tecnológicos, así como servicios de investigación y diseño conexos; servicios de análisis industrial, investigación industrial y diseño industrial; control de calidad y servicios de autenticación; diseño y desarrollo de hardware y software.

Clase 43 Servicios de restauración (alimentación); hospedaje temporal.

Clase 44 Servicios médicos; servicios veterinarios; tratamientos de higiene y de belleza para personas o animales; servicios de agricultura, acuicultura, horticultura y silvicultura.

Clase 45 Legal Servicios jurídicos; servicios de seguridad para la protección física de bienes materiales y personas; servicios personales y sociales prestados por terceros para satisfacer necesidades individuales.