



Universidad
Zaragoza

**Trabajo Fin de Grado
Grado en Ingeniería Informática**

Diseno Computacional de Iconos

Autor

Isabel Gaspar Sánchez

Directora

Elena Garcés García

Ponente

Diego Gutiérrez Pérez

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

2016



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Isabel Gaspar Sánchez

con nº de DNI 17457002-W en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) **Grado** _____, (Título del Trabajo) **Diseño Computacional de Iconos**

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 27 de Agosto de 2016



Fdo: Isabel Gaspar

Resumen

Diseno Computacional de Iconos

Elegir el mejor ícono para una interfaz de usuario es una tarea muy compleja. En el proceso normal de decidir el conjunto óptimo de íconos, un diseñador es el encargado de proponer unos diseños iniciales. A continuación, siguen una serie de iteraciones sobre dicho conjunto donde el cliente y un grupo de trabajo evalúan los íconos y proponen cambios que implementa el diseñador. Este proceso puede llegar a ser muy costoso de tiempo y presupuesto. Actualmente existen bases de datos de iconografía genérica¹ que proporcionan multitud de íconos para interfaces a precios mucho más asequibles. Sin embargo, elegir el conjunto óptimo presenta varios retos. 1) El conjunto debe ser coherente en estilo. 2) Los íconos deben ser distinguibles entre sí, es decir, no deben plantear confusión o ambigüedad entre ellos, es lo que se conoce como "identificabilidad". Este proyecto tiene como objetivo realizar la selección óptima de íconos en base a estas dos propiedades. Para ello, se comienza obteniendo un vector de características de bajo nivel sobre las imágenes. Seguidamente, a partir de estas características se calcula una métrica que medirá la distancia entre las distintas imágenes. Por último, según estas distancias se obtendrá el conjunto óptimo de imágenes para un ícono.

¹<https://thenounproject.com/>

Índice general

1. Introducción	6
1.1. Motivación	6
1.2. Objetivos	7
1.3. Contexto del proyecto	8
1.4. Estructura del documento	8
2. Contexto tecnológico	9
2.1. Conceptos básicos	9
2.2. Trabajo relacionado	10
3. Obtención de datos	13
3.1. Obtención de datos	13
3.2. Base de datos	16
3.2.1. Esquema E/R	17
4. Aprendizaje de la métrica de estilo	21
4.1. Definición de características de estilo	21
4.2. Algoritmo de aprendizaje: <i>Metric Learning to Rank</i>	22
4.3. Entrenamiento y test	22
4.4. Evaluación perceptual	23
5. Aplicaciones	30
5.1. Clustering	30
5.2. Selección de conjuntos óptimos	31
6. Conclusiones	38
6.1. Trabajo realizado	38
6.1.1. Planificación	38
6.2. Trabajo futuro	39
6.3. Conclusión personal	40
Bibliografía	41
A. Anexo I	42
A.1. Tablas de la Base de datos	42

ÍNDICE GENERAL

B. Anexo III	44
B.1. Métricas	44
B.1.1. AUC	44
B.1.2. Precision-at-k y KNN	46
B.1.3. MAP	47
B.1.4. MRR	47
B.1.5. NDCG	48
C. Anexo II	49
C.1. Resultados del aprendizaje de la métrica	49
D. Anexo III	57
D.1. Resultados del Clustering	57

Capítulo 1

Introducción

1.1. Motivación

Elegir el mejor ícono para una interfaz de usuario es una tarea muy compleja. En el proceso normal de decidir el conjunto óptimo de íconos, un diseñador es el encargado de proponer unos diseños iniciales. A continuación, siguen una serie de iteraciones sobre dicho conjunto donde el cliente y un grupo de trabajo evalúan los íconos y proponen cambios que implementa el diseñador. Sin embargo, los íconos son utilizados por muchas personas diferentes, cada una de ellas con diferentes preferencias y necesidades. Esto hace que elegir el ícono ideal para una interfaz sea muy difícil para un único diseñador. Este proceso puede llegar a ser muy costoso de tiempo y presupuesto, si se consigue automatizar, se conseguiría abaratar este coste. Actualmente existen bases de datos de íconos¹ que pueden abaratar costes en el diseño. Sin embargo, la cantidad de íconos presentes en dichas bases de datos es muy grande, del orden de 10^5 , y por tanto, seleccionar el conjunto óptimo sigue siendo una tarea muy costosa.

Cuando se habla de seleccionar el conjunto óptimo de un grupo de íconos, se está persiguiendo obtener un grupo de imágenes **coherentes en estilo** y cuyo sentido semántico coincide con la consulta realizada. Además, el usuario debe poder comprender lo que cada ícono intenta representar, esto es lo que se denomina **comprendibilidad**. Así como ha de poder identificar un ícono entre el conjunto que se utiliza en la interfaz, esto es la propiedad de **identificabilidad**. En resumen, un conjunto óptimo debe cumplir los requisitos de estilo, comprendibilidad e identificabilidad. En este proyecto nos vamos a centrar únicamente en el primer punto, el estilo.

A cualquier persona le gusta ver una interfaz “bonita” y una de las claves para ello es lograr que los elementos que la componen sean *coherentes en estilo*. Imaginad que queremos crear una interfaz sobre signos del zodiaco y tenemos que elegir un ícono para cada uno de ellos. De los 12 signos del zodiaco, cuatro están representados en la Figura 1.1, de izquierda a derecha en ambas filas se pueden ver Aries, Tauro, Libra y Capricornio representados de distinta forma, entre los cuales en total se pueden apreciar tres estilos diferentes:

- Los dos primeros íconos de la línea superior, constan de una única línea con cierto

¹<https://thenounproject.com/>

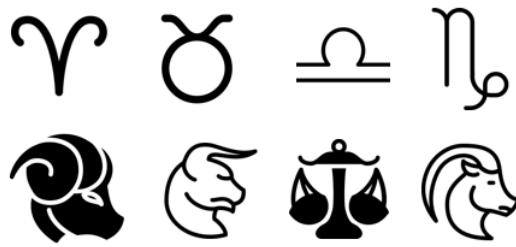


Figura 1.1: Iconos candidatos para una interfaz de usuario sobre signos del zodiaco. Se pueden apreciar distintos estilos visuales. Por ejemplo, el grosor de la línea no es igual en todos los casos, y no todos los estilos tienen relleno sólido negro.

grosor, mientras que los dos siguientes tienen un trazo más fino.

- El primero y tercero de los iconos de la línea inferior, son iconos dibujados como un sólido negro, mientras que el segundo y el cuarto son imágenes sin relleno.
- El resto de los iconos tienen un trazo similar; la diferencia que en la línea inferior están representados como animales, mientras que en la línea superior son símbolos griegos, pero esta característica no influye en el estilo.

Definir el estilo visual de un ícono es por tanto una tarea compleja que requiere análisis de bajo nivel de las características de la imagen. Por tanto, la pregunta que se intenta resolver en este trabajo es la siguiente: *Dada una base de datos de íconos existente, ¿somos capaces de elegir un conjunto óptimo de íconos coherente en estilo visual?* Para ello, definiremos un vector de características de bajo nivel que usará los píxeles de la imagen de entrada, utilizaremos datos etiquetados para entrenar una métrica de distancia que nos permitirá realizar búsquedas por similaridad en estilo, y finalmente presentaremos varias aplicaciones prácticas donde mostramos la utilidad de la métrica.

1.2. Objetivos

En resumen, los objetivos del proyecto son los siguientes:

- Diseñar y programar el conjunto de características de bajo nivel con las que se representará a cada ícono. Los íconos estarán representados por bitmaps en blanco y negro de tamaño fijo.
- Obtener una métrica de distancia que mida la similaridad en estilo para dos íconos dados, de modo que dos imágenes del mismo estilo tendrán una distancia muy pequeña, mientras que dos imágenes de distinto estilo tendrán una distancia mucho mayor. La métrica se aprenderá con técnicas de aprendizaje automático y bases de datos existentes.
- Obtener una función que para un conjunto de palabras clave dadas, busque en una base de datos el conjunto óptimo de íconos en base a la métrica de similaridad.

1. Introducción

1.3. Contexto del proyecto

El proyecto ha sido desarrollado en el laboratorio del Grupo de Informática Gráfica Avanzada (GIGA), dentro del Graphics and Imaging Lab (GILab) perteneciente al Departamento de Informática e Ingeniería de Sistemas (DIIS) en la Escuela de Ingeniería y Arquitectura (EINA) de la Universidad de Zaragoza.

Este proyecto se enmarca en el contexto de una colaboración previa de Elena Garcés y Diego Gutierrez con investigadores de la Universidad de Tokio. En dicho proyecto se utilizaron técnicas de *crowdsourcing* para obtener conjuntos óptimos de iconos en base a las propiedades de comprensibilidad, e identificabilidad mencionadas anteriormente. Sin embargo, no es estudió la parte correspondiente al estilo, que es de lo que trata el presente proyecto fin de grado.

1.4. Estructura del documento

Este documento está estructurado en seis capítulos que corresponden con las fases principales del proyecto, además de la bibliografía.

En el Capítulo 2 se habla del contexto tecnológico. Se explican los conceptos básicos necesarios para comprender el resto de la memoria.

El Capítulo 3 tiene que ver con la obtención de datos necesarios para trabajar, así como la forma en que son almacenados en la base de datos.

El Capítulo 4 contiene toda la información relevante al análisis de datos, esto es, el entrenamiento y testing de los mismos, así como las métricas que se han utilizado para ello y resultados obtenidos.

El Capítulo 5 muestra aplicaciones prácticas donde se demuestra la utilidad de la métrica propuesta. Se realiza un clustering de los datos y se proponen conjuntos de iconos para aplicaciones sobre invitaciones de boda, aplicaciones meteorológicas, una caja autoservicio de un supermercado o una reserva vacacional.

Por último, el Capítulo 6 presenta las conclusiones del proyecto, así como la línea de investigación que se podría contemplar para un trabajo futuro.

Capítulo 2

Contexto tecnológico

2.1. Conceptos básicos

La idea principal que envuelve este proyecto son los iconos. Se denomina **ícono** a un pictograma que representa un objeto o un concepto con los que guarda una relación de identidad o semejanza formal. En este proyecto únicamente se trabaja con iconos en blanco y negro como los que se muestran en la Figura 2.1.



Figura 2.1: Ejemplos de íconos

Estas imágenes (ver Figura 2.1) son muy diferentes entre sí, además de representar distintas ideas o conceptos, la apariencia visual, la forma en que han sido dibujadas es muy distinta. Entre ellas varía el grosor del trazo, contornos redondeados o rectos, zonas más o menos sombreadas, texturas, etc. Cada uno de estos rasgos que componen la imagen, se denomina **característica** o **feature** del ícono y el conjunto de todas las que pertenecen al mismo ícono definen su **estilo**. En la Figura 2.2 se pueden apreciar dos grupos de íconos, uno por cada una de las filas, que cumplen con la propiedad de **similaridad**¹ en estilo.

En la Figura 2.2, primera fila, se pueden apreciar imágenes sin relleno, con un trazo redondeado y un grosor similar. Sin embargo, en la Figura 2.2, segunda fila, se ven imágenes con un relleno sólido negro y contorno redondeado.

En el momento que un conjunto de íconos están relacionados por la idea que representan, se dice que pertenecen a la misma **categoría**. Por ejemplo, en la Figura 2.3 se pueden apreciar dos categorías de íconos: en la primera fila, íconos relacionados con la comida y en la segunda fila, íconos relacionados con el transporte.

¹Similaridad - Propiedad que mide la coherencia visual de los íconos



Figura 2.2: Los grupos de iconos de cada fila son similares en estilo.

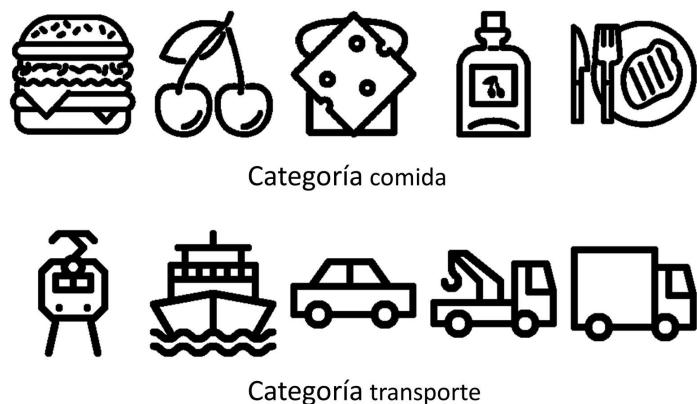


Figura 2.3: Conjunto de iconos pertenecientes a las categorías de comida, primera fila, y transporte, segunda fila.

2.2. Trabajo relacionado

Existen trabajos anteriores relacionados con la similaridad en estilo de imágenes. A continuación se van a mostrar algunos de ellos.

Iconos En el trabajo de Laursen et al. [LKC⁺16], se presentaba un nuevo método, en el cual dados algunos iconos candidatos para una funcionalidad, se selecciona un conjunto optimizado comprensible e identificable para un usuario. Estas dos propiedades se miden usando computación humana.

El escenario propuesto en este trabajo parte de un escenario tradicional en el que el cliente especifica lo que desea o necesita al diseñador. Es entonces, cuando el diseñador intenta reflejar lo mejor posible la petición del usuario en los iconos. Tradicionalmente, el trabajo del diseñador es evaluado por el cliente, un grupo de personas, un diseñador o una combinación de estos; seguidamente, se le notifican las críticas de nuevo al diseñador para realizar los cambios necesarios hasta que se obtiene la aprobación completa de su trabajo y se selecciona el conjunto que se va entregar al cliente. Sin embargo, en este proyecto se utiliza computación humana para la evaluación de los iconos que son evaluados a través

2. Contexto tecnológico

de un *crowd*² y el proceso de selección se automatiza por medio de software. Esto tiene las ventajas de una menor subjetividad que en el escenario tradicional y la obtención de resultados de forma descentralizada de los usuarios, con lo que se puede comenzar la selección una vez que los iconos necesarios estén disponibles.

En este trabajo, no miden la similaridad en estilo; sin embargo, la falta de consistencia en estilo influirá negativamente en la evaluación de la identificabilidad, el *crowd* encargado de evaluar los iconos, se centrará en las diferencias en estilo, en lugar de fijarse en el significado de los iconos.

Ilustración y clip-art Garcés et al. [GAGH14] presenta un método para medir la similaridad en estilo entre dos imágenes *clip art*, independientemente del contenido. Los *clip art* se encuentran en una enorme variedad de formatos, desde los formatos de imágenes de mapas de bits a los de los gráficos vectoriales. Los *clip art* están compuestos exclusivamente de ilustraciones (creadas a mano o bien mediante el uso de ordenadores) y no incluyen stocks de fotografías.

El modelo está basado en un conjunto de características que describen cada imagen, en total 169 características divididas en cuatro categorías: color, sombreado, textura y trazo. En la Figura 2.4 se pueden ver algunos ejemplos de cada una de estas categorías. En cada categoría (ver Figura 2.4) se muestran dos imágenes cuyo estilo es muy diferente.



Figura 2.4: Ejemplos de las distintas categorías de características que describen los iconos

En el ejemplo de color se ve una imagen con un gran contraste de colores, frente a una monocroma. En el sombreado se muestra una imagen con un gradiente de color frente a una imagen con regiones de color constante. En el ejemplo de la textura, se muestra una imagen con patrones artísticos frente a una imagen grabada en madera. Finalmente, el ejemplo del trazo muestra una imagen dibujada en pluma con líneas de diferente grosor, frente a una imagen con un grosor de trazo constante. En nuestro proyecto se va a tratar de medir la similaridad en estilo de iconos, en lugar de imágenes clip-art. Por tanto, se obtendrá un modelo similar al de dicho proyecto que se adapte a nuestras imágenes.

Buscar conjuntos de imágenes por estilo es un problema particular de recuperación de información. Uno de los principales problemas es que la percepción de estilo es altamente subjetiva, lo que hace que el etiquetado de estilo sea una tarea complicada. Garcés et al. [GAGH16] propone varias contribuciones hacia una mejor comprensión del estilo de una ilustración y su utilidad para la exploración y recuperación de datos. En primer lugar, proporcionan nuevos conocimientos sobre como percibir el estilo en una imagen. En segundo lugar, evalúan un conjunto de imágenes clip art dibujadas a mano con una métrica de estilo existente para analizar como funciona ésta métrica con conocimiento experto.

²multitud de personas

2. Contexto tecnológico

Finalmente, proponen un método eficiente de navegación y exploración en un gran conjunto de datos clip art que tiene en cuenta tanto el etiquetado semántico de los datos, como de su estilo. El enfoque que proponen combina clustering jerárquico con técnicas de reducción de dimensionalidad, y muestreo estratégico para obtener visualizaciones intuitivas y útiles. En nuestro proyecto seleccionaremos algunas etiquetas y utilizaremos el método propuesto en dicho trabajo sobre navegación y exploración de un conjunto de datos para obtener grupos de iconos similares en estilo cuyo sentido semántico sea coherente con dichas etiquetas.

Otras métricas Recientemente se han desarrollado numerosos métodos que siguen un proceso similar a Garces et al. [GAGH14] para aprender métricas de similaridad. Por ejemplo, Bell el at. [BB15] utilizan redes convolucionales para aprender similaridad visual entre productos de decoración; Saleh et al. [SDHL15] aprenden una métrica para comparar infografías; o Liu et al. [LHLF15] y Lun et al. [LKS15] se basan también en la percepción humana para capturar la similaridad entre objetos 3D.

Capítulo 3

Obtención de datos

Este capítulo se centra en explicar cómo se han obtenido los iconos con los que se ha trabajado en el proyecto, así cómo la forma en que se han almacenado en una base de datos creada específicamente para este proyecto.

3.1. Obtención de datos

Actualmente, existen bases de datos de iconografía genérica disponibles en la web. En este proyecto, se recurrió a *Noun Project*¹, que proporciona multitud de iconos para una gran variedad de interfaces y es de uso gratuito. A través de una API proporcionada por Noun Project se realizaron peticiones para obtener iconos de su web. Noun Project ofrece documentación sobre como realizar dichas peticiones. Haciendo uso de la API, se implementó desde cero el código python necesario para realizar las peticiones, teniendo en cuenta que únicamente se puede realizar una petición cada 5 segundos o se producirá una denegación de servicio en el sistema por la gran cantidad de información solicitada.

Observamos que cada una de las **colecciones** del dataset contenía iconos coherentes en estilo (ver Figura 3.1), información que queríamos capturar. Por tanto, nuestro objetivo era realizar las peticiones necesarias para descargarnos los iconos de cada una de las colecciones y almacenar dicha información en una base de datos. Noun Project identifica las colecciones numéricamente comenzando en 1 y otorgando nuevos identificadores a medida que el número de colecciones aumenta, de modo que nosotros lo haremos de la misma manera. Sin embargo, no sabemos exactamente si existe una colección por cada número, de modo que se utilizará un identificador auto-incrementado y se realizarán peticiones que podrán tener o no éxito. En el caso de que la colección exista, necesitamos hacer dos peticiones ya que así lo especifica la API. En la primera petición, hacemos la petición a la colección y en caso de que exista obtendremos su información general, es decir, el tema al que hace referencia o el autor (ver Figura 3.2(a)). Entonces realizaremos la segunda petición que nos proporcionará información sobre el conjunto de iconos que componen la colección, en los que se pueden ver datos como el identificador del ícono, las etiquetas que lo referencian, o su URL, con la que podremos descargar la imagen (ver Figura 3.2(b)).

A partir de ambos ficheros, hubo que realizar una extracción de datos, para guardar

¹<https://thenounproject.com/>

3. Obtención de datos



Figura 3.1: Colección cuyos iconos son coherentes en estilo.

aquellos que interesaban en el proyecto. Al mismo tiempo que se iban obteniendo los iconos y toda la información referente a ellos, se fueron almacenando en una base de datos previamente creada y que se explica en la siguiente sección.

Una vez obtuvimos el dataset completo, el cual contiene 26566 iconos divididos en 1084 colecciones, comprobamos si los datos eran consistentes en estilo. En principio, cada conjunto de iconos perteneciente a una misma colección posee el mismo estilo, sin embargo, observamos que esto no era así siempre (ver Figura 3.3), y en ocasiones había colecciones con datos de estilos diferentes. Por tanto, se llevó a cabo un **curado de los datos**. El proceso que se realizó para ello se resume en la Figura 3.4. Manualmente se inspecciona el dataset, en el caso de encontrar una colección cuyo estilo no es coherente se selecciona un subconjunto de iconos con el mismo estilo y se agrupan en un nuevo directorio. Entonces, mediante un script python, se inspecciona el dataset completo y por cada uno de los nuevos conjuntos creados manualmente se genera una nueva colección en el directorio raíz y se elimina el directorio creado a mano en el paso anterior. Este es un proceso delicado ya que, como en este caso hay varios autores que comparten el tema de una colección y hay que conservar la relación entre autores, temas de la colección e iconos. Para diferenciar una nueva colección de la anterior se le añade un número al final de la cadena de texto que lo representa y se modifica la base de datos existente con los cambios realizados.

Además, se realizó una **normalización de los iconos**; a pesar de que todos los ficheros eran bitmaps *.png* y tenían el mismo tamaño, había una gran diferencia en el espacio que ocupaba el ícono dentro de la imagen. Este espacio que quedaba alrededor del ícono podía generar un error innecesario en el cálculo de características de bajo nivel, como veremos en la Sección 4.1. En la Figura 3.6 se pueden ver dos ejemplos de íconos antes y después del normalizado. En las imágenes de la derecha (ver Figura 3.6) se puede ver como el ícono después de la normalización aprovecha el espacio del fichero mejor que en las imágenes de la izquierda. El proceso de normalización puede verse en la Figura 3.5 y se resume en 6 pasos: 1) Obtener las propiedades que definen el ícono; 2) Calcular los límites del ícono en la imagen, en realidad lo que se pretende conseguir es únicamente la superficie que ocupa el ícono y para ello, se calcula la bounding box más pequeña que lo encierra. 3) Se extrae la imagen de la bounding box calculada en el paso anterior. 4) Se calcula el factor para redimensionar la imagen, buscamos conseguir una imagen de tamaño 200x200px, en la que se respete un borde de 20px. 5) Se realiza la redimensión de la imagen. 6) Se genera la

3. Obtención de datos

```
{  
  "collection": {  
    "author": {  
      "location": "Los Angeles, US",  
      "name": "Edward Boatman",  
      "permalink": "/edward",  
      "username": "edward"  
    },  
    "author_id": "6",  
    "date_created": "2012-01-27 19:15:07",  
    "date_updated": "2012-10-16 15:37:06",  
    "description": "",  
    "icon_count": "1462",  
    "id": "1",  
    "is_collaborative": "",  
    "is_featured": "0",  
    "is_published": "0",  
    "is_store_item": "0",  
    "name": "Noun Collection",  
    "permalink": "/edward/collection/noun-collection",  
    "slug": "noun-collection",  
    "sponsor": {},  
    "sponsor_campaign_link": "",  
    "sponsor_id": "",  
    "tags": [],  
    "template": "24"  
  },  
}  
}
```

(a) Colección

```
{  
  "generated_at": "Mon, 22 Feb 2016 16:17:40 GMT",  
  "icons": [  
    {  
      "attribution": "Trash from Noun Project",  
      "date_uploaded": "",  
      "icon_url": "https://d30y9cdsu7xlg0.cloudfront.net/noun-svg/1.svg?Expires=14!",  
      "id": "1",  
      "is_active": "1",  
      "license_description": "public-domain",  
      "permalink": "/term/trash/1",  
      "preview_url": "https://d30y9cdsu7xlg0.cloudfront.net/png/1-200.png",  
      "preview_url_42": "https://d30y9cdsu7xlg0.cloudfront.net/png/1-42.png",  
      "preview_url_84": "https://d30y9cdsu7xlg0.cloudfront.net/png/1-84.png",  
      "sponsor": {},  
      "sponsor_campaign_link": null,  
      "sponsor_id": "",  
      "tags": [  
        {  
          "id": 19,  
          "slug": "trash"  
        },  
        {  
          "id": 20,  
          "slug": "garbage"  
        },  
        {  
          "id": 21,  
          "slug": "litter"  
        }  
      ],  
      "term": "Trash",  
    }  
  ]  
}
```

(b) Lista

Figura 3.2: (a) Respuesta de petición de Noun Project referente a una colección. (b) Respuesta de petición de Noun Project referente a la lista de iconos de la colección. En (b) sólo se muestra la información referente a un único ícono, en realidad el fichero contendría esta información para cada uno de los íconos del grupo.

3. Obtención de datos

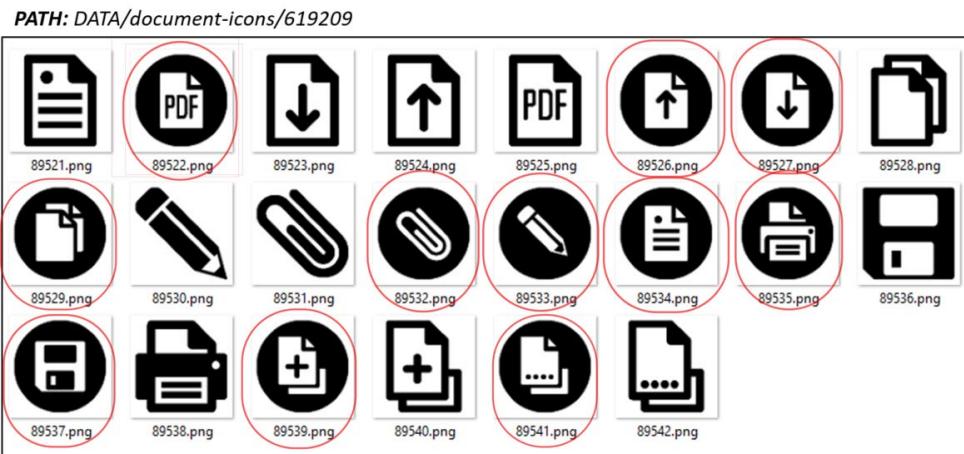


Figura 3.3: Los iconos de la colección marcados en rojo no siguen una coherencia en el estilo con el resto.

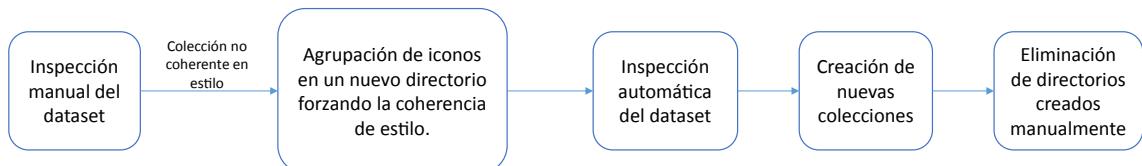


Figura 3.4: Esquema resumen del reordenamiento del dataset

imagen final.

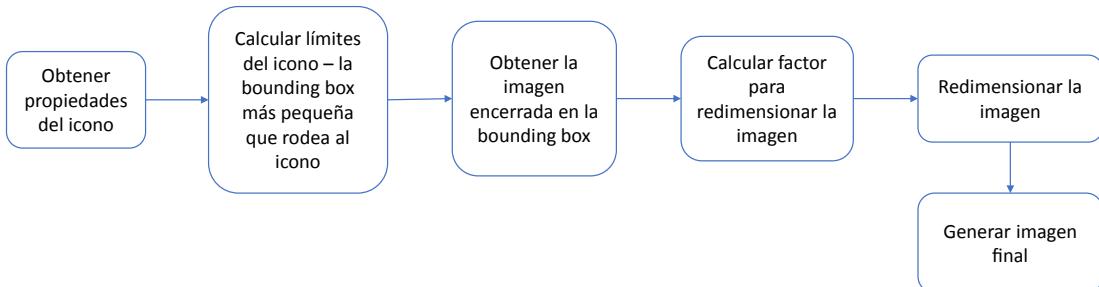


Figura 3.5: Proceso de normalización de los iconos.

3.2. Base de datos

La base de datos con la que se ha decidido trabajar es SQLite en su versión 3. Este tipo de bases de datos no requieren una gran configuración, no tienen un servidor ejecutándose en un proceso separado, y por tanto su uso es muy sencillo y proporciona gran flexibilidad.

3. Obtención de datos

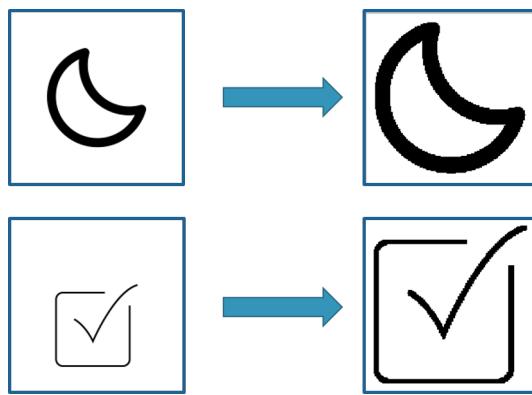


Figura 3.6: Ejemplos de normalización de dos iconos.

3.2.1. Esquema E/R

El esquema entidad-relación (E/R) sobre el que se ha construido la base de datos que almacena toda la información relacionada con los iconos se puede ver en la Figura 3.7. Como se puede apreciar en el esquema, la BD está compuesta por cuatro entidades, cada una de ellas definida por un conjunto de atributos. El conjunto de datos al completo, exceptuando los atributos *features* de *Image* y *SIDS* de *Collection* son originales de Noun Project y se recuperaron de las peticiones a la web.

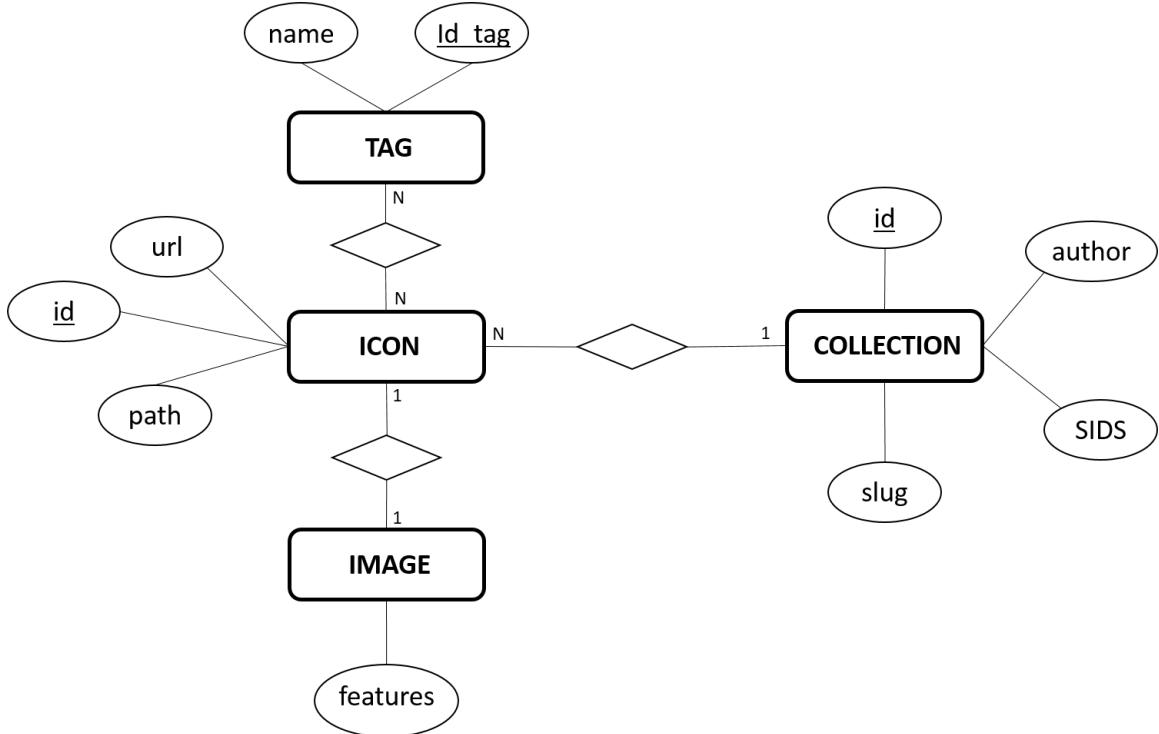


Figura 3.7: Esquema E/R de la base de datos

3. Obtención de datos

Collection La entidad *Collection* representa un conjunto de iconos relacionados por el momento en que fueron creados. Un autor específico decide crear un conjunto de imágenes coherentes en estilo y relacionadas, por ejemplo, con la comida, el conjunto de esos iconos etiquetados bajo el tema “comida” y el identificador de la persona que los ha creado es lo que se representa en la entidad *Collection*. Los atributos que la definen son los siguientes:

- **id** - Identificador de la colección. Este identificador es único para cada colección y por ello es la clave primaria de la entidad.
- **slug** - Este atributo es una cadena de texto que representa el tema sobre el que trata la colección. Ejemplos de slug son: *fishes*, *flags of the world*, *yoga*, o *zodiac*, de modo que entre sus iconos podremos encontrar imágenes relacionadas con peces, banderas del mundo, yoga o los signos del zodiaco.
- **author** - Identificador del autor que ha creado la colección. En nuestro dataset las colecciones son almacenadas bajo un directorio identificado por su *slug*; sin embargo, varias colecciones pueden identificarse por el mismo slug, por lo tanto, para no mezclar los elementos de las distintas colecciones es necesario crear un subdirectorio bajo el que determina el slug de la colección, y que representará su autor. De esta manera logramos ordenar las colecciones de iconos coherentemente.
- **SIDS** - *Same Icon Different Style* - Este atributo puede estar vacío, y lo que representa es aquella colección que está formada por el mismo ícono pero con diferente estilo. Este tipo de íconos fueron detectados al realizar la inspección visual durante el curado del dataset. En este proyecto no llega a usarse, pero podría ser útil en una línea futura de investigación. Hace referencia a datos como los que se ven en la Figura 3.8.



Figura 3.8: Conjunto de íconos que representan el mismo concepto, con estilo diferente.

Icon La entidad *Icon* representa la imagen en sí. *Icon* está compuesta por tres atributos:

- **id** - Identificador del ícono. Único para cada una de las imágenes y por ello se ha definido como clave primaria de la entidad.
- **url** - Localizador del ícono en la web.
- **path** - Los íconos están almacenados en un directorio local; este atributo representa la ruta donde encontrarlo en ese conjunto de ficheros.

Icon se relaciona con la entidad *Collection* a través del identificador de la colección; mientras que lo hace con su propio identificador para el resto de entidades.

3. Obtención de datos

Tag La entidad *Tag* representa un conjunto posible de palabras con que diferentes usuarios pueden referenciar al mismo ícono. Los atributos que la definen son:

- **name** - Etiqueta del ícono que se está representando. Por ejemplo si queremos buscar una imagen sobre un perro, podemos buscar por la palabra perro, pero también dog, animal, mascota o bulldog entre otras. Cada ícono tiene asociadas una serie de palabras por las que un usuario podría hacer referencia a él y esto son las etiquetas.
- **id_tag** - Identificador de la etiqueta. Valor numérico asociado a cada etiqueta y que es único para cada una de ellas. Por ejemplo la etiqueta dog en todas sus apariciones tendrá el identificador 399 y ninguna otra etiqueta tiene el mismo.

En la Figura 3.9 se muestran dos íconos acompañados de una lista de etiquetas junto con el identificador que las representa. Ambos íconos comparten algunas etiquetas como bubble, chat, dialog o speech entre otras, y por supuesto, el identificador del ícono se va a repetir para toda su lista.

Id_tag	name
1652	Talking
2286	Design
371	Conversation
26760	Commentary
364	Cloud
11650	Chatting
2147	chat
2055	Bubble
1157	Box
2836	Dialog
2660	Speaking
1414	Speech
2142	Message
3337	dialogue

Id_icon: 70008

Id_tag	name
1652	Talking
2286	Design
461	Communication
26760	Commentary
364	Cloud
11650	Chatting
2147	chat
2055	Bubble
24352	Dialog-box
2836	Dialog
2660	Speaking
1414	Speech
13970	Messages
3337	dialogue

Id_icon: 69997

Figura 3.9: Entidad *Tag*. Se muestran dos íconos acompañados de su identificador numérico y una lista de etiquetas que les hacen referencia junto con su identificador de etiqueta.

Image Por último, la entidad *Image* almacena el vector de características de cada ícono. El atributo *features* es una cadena de texto que representa el vector de características de las imágenes; en realidad lo que se almacena es un vector numérico convertido a cadena de caracteres usando JSON. Más detalles sobre las características de estilo y la métrica se explicarán en la Sección 4.1 y 4.2, respectivamente.

Las entidades se relacionan mediante claves. Como muestran las relaciones del esquema, un ícono puede pertenecer a una única colección, y como es lógico una colección contiene un número indefinido de íconos. Las entidades *Icon* e *Image* tienen relación 1:1 porque un ícono está definido por un único vector de características, así como ese vector pertenece únicamente a un ícono. Y por último, un ícono puede tener varias etiquetas (*Tag*) que lo

3. Obtención de datos

referencien, así como una etiqueta puede estar referenciada por un número indefinido de iconos. En el Apéndice A se pueden consultar algunos ejemplos del almacenamiento de la información en la base de datos.

Capítulo 4

Aprendizaje de la métrica de estilo

Este apartado contiene la información relevante al análisis de datos. Se empezará explicando la definición de las características de estilo con las que se ha trabajado y seguidamente se verá el entrenamiento y la evaluación de los datos, así como los resultados obtenidos.

4.1. Definición de características de estilo

Para poder medir la distancia entre la similaridad de estilo de los iconos, necesitamos obtener las características de bajo nivel que definen cada uno de ellos. Garcés et al. [GAGH14], define un vector de 169 características computables a partir del bitmap de las imágenes, que capturan propiedades de *color*, *sombreado*, *textura* y *trazo*. Éstas, son las características de las que se va a partir en este proyecto; sin embargo, el estilo de dichas imágenes -imágenes en color de más complejidad visual (ver Figura 2.4)- no es igual que el estilo de nuestros iconos -en blanco y negro sin propiedades de sombreado- y por ello se va a realizar un análisis de la varianza sobre ellas para eliminar las que no aportan información relevante. Como resultado de este análisis, se eliminan características como el nivel de gris o la saturación de la imagen, ya que únicamente se trata de iconos en blanco y negro puro. Por lo tanto, el número de características con las que se va a trabajar se reduce de 169 a 117.

Normalización de las características Una vez se obtiene el vector de características final, es necesario normalizar los datos y para ello se utiliza *feature scaling* o lo que es lo mismo, escalado de características; este método escala los datos en el rango [-1,1] de la siguiente manera:

$$x' = \frac{x - \min}{\max - \min} \quad (4.1)$$

donde x es el vector de características de una imagen, x' el vector de características normalizado y \min y \max son los límites del rango entre los cuales están comprendidos los valores de las características de dicho vector. En la normalización de los datos los valores que pertenezcan a una misma categoría se tratarán de forma conjunta e independientemente del resto de categorías, así se realizará el escalado para cada una de las características de las imágenes por separado.

4.2. Algoritmo de aprendizaje: *Metric Learning to Rank*

El estudio del aprendizaje de la métrica se ha tratado como un problema de recuperación de información. En particular hemos utilizado el algoritmo Metric Learning to Rank (MLR) [ML10] para el entrenamiento de nuestra métrica de estilo. Este algoritmo aprende una métrica de distancia partiendo de conjuntos de datos etiquetados. Para ello se basa en entrenamiento de Support Vector Machines (SVM) optimizadas para diferentes medidas de ranking (AUC, Precision-at-k, MRR, MAP o NDCG). En este proyecto se utilizó el código proporcionado por los autores y su completa comprensión está fuera de los objetivos de este proyecto. No obstante, sí que se estudiaron las diferentes medidas de ranking para evaluar el impacto que podían tener con nuestros datos. Una explicación detallada de estas medidas se encuentra en el Apéndice B.

El resultado del aprendizaje es una métrica de distancia que relaciona las características de bajo nivel definidas previamente. En particular, se aprende la distancia Manhattan definida a continuación:

$$d(x, y) = \sqrt{(x - y)^T W (x - y)} \quad (4.2)$$

donde x e y son los vectores de los dos iconos entre los que se mide la distancia y W es la matriz que representa la métrica que aprende el algoritmo.

Se considera que una métrica es buena si, la clasificación del conjunto de elementos resultado de su entrenamiento queda ordenado por incremento de la distancia de forma ascendente al ítem evaluado, de forma que los elementos que más se asemejen en estilo al que se está evaluando aparecerán al principio y los menos coherentes en estilo al final de la lista. La tasa de error, por tanto, se puede tratar como una función de pérdida sobre la clasificación de los elementos. En su esencia, el problema de aprendizaje de la métrica es un caso especial de recuperación de información.

4.3. Entrenamiento y test

Se realiza el entrenamiento de la métrica de estilo para encontrar aquella que mejor se adapta a nuestro problema, es decir, aquella que mejor mide la coherencia en estilo de los iconos. Los datos que se utilizan para el entrenamiento son un conjunto 26566 iconos agrupados en distintas colecciones según la similaridad en estilo, los cuales han sido obtenidos de Noun Project como se explica en la Sección 3.1. La correspondencia entre datos para la fase de entrenamiento y test, es de 80 % - 20 %, respectivamente.

En este punto, ya se puede dar comienzo al entrenamiento de los datos, y para ello se va a utilizar código de McFee y Lim [ML10], a partir del cual se va a aprender la métrica. En este caso como se ha explicado en la Sección 4.2, obtendremos una métrica para cada tipo de función de ranking usada (AUC, MAP, MRR, Prec@k, KNN y NDCG), de modo que es necesario entrenar para cada una de ellas por separado y para diferentes valores de k . k es un factor que únicamente es necesario en las métricas Prec@k, KNN y NDCG y que determina el número de iconos que se van a tener en cuenta al realizar el cálculo de la métrica. Con los pesos de la métrica aprendida y el factor k para las métricas Prec@k,

4. Aprendizaje de la métrica de estilo

KNN y NDCG, además de las características y etiquetas de los iconos de las fases de entrenamiento y test por separado se puede realizar la fase de test de los datos.

Los resultados que muestran el error del test se pueden ver en la Figura 4.1, donde hay columnas etiquetadas con un número, el valor de k descrito anteriormente, y una letra que indica si el error de test es medido antes o después del aprendizaje, 'A' y 'B', respectivamente; las columnas 'DIF' miden la diferencia entre ambos y, por tanto indican si el algoritmo mejora y cuánto mejora.

Una vez analizados los datos, se observa que la mejor opción es entrenar para MAP (el mejor valor obtenido se puede ver resaltado en amarillo en la tabla MAP) y utilizar AUC como medida de evaluación. En primer lugar, MAP es una métrica que contempla la colección completa; mientras que medidas como KNN, Prec@ k o NDCG únicamente trabajan sobre los k primeros elementos. Además, MAP da más importancia a encontrar items relevantes en las primeras posiciones de la colección; cuánto más cerca estén los items relevantes del primer resultado recuperado mejor será el MAP obtenido. Si a esto le añadimos que AUC encuentra items relevantes más rápido que el resto de métricas, es decir, en índices más bajos de la lista de items recuperados obtenemos una buena combinación. De modo que, entrenamos la colección para obtener mayor beneficio al encontrar items relevantes en las primeras posiciones de la lista de items recuperados, es decir, MAP; y seleccionamos AUC que es la métrica que mejor se adapta a ello. AUC encuentra verdaderos positivos a partir de las posiciones del primer item relevante para MAP, independientemente de que sea el primer elemento recuperado o no.

También se puede ver en las Figuras 4.2 (a) y (b) cómo destaca la métrica AUC frente a las demás cuando el sistema ha sido entrenado para MAP. Ambas gráficas representan la misma información desde dos ángulos diferentes. El eje de ordenadas de ambas gráficas representa la precisión del test; el eje de abcisas de la primera de ellas (Figura 4.2(a)), representa las distintas métricas; mientras que en la segunda gráfica (Figura 4.2(b)), están representados los valores $k = 1,2,4,8$ y 16 antes y después del entrenamiento. Destaca en las gráficas que la métrica que menos mejora con el sistema entrenado para MAP, es precisamente MAP, esto podría ser porque el etiquetado que poseemos no es bueno; existen etiquetas que pueden representar el mismo estilo y en el dataset con el que trabajamos se están tratando como estilos diferentes, para resolver este problema sería necesario hacer un clustering de los datos, o bien recoger nueva información de etiquetado.

4.4. Evaluación perceptual

Completado el entrenamiento de la métrica, como los resultados de la evaluación numérica no son demasiado concluyentes (la mejora de error antes y después del entrenamiento es muy pequeña), es necesario una inspección visual de los resultados. Para ello vamos a hacer búsquedas sobre el dataset utilizando una imagen de consulta, y esperando obtener el conjunto de imágenes más parecidas en estilo visual. Idealmente deberíamos obtener en las primeras posiciones de la búsqueda las imágenes de la misma colección. Para el mismo conjunto de iconos de consulta compararemos el resultado de la búsqueda con la métrica de Garces et al. [GAGH14], y con nuestras propias características antes y después de aprender la métrica.

4. Aprendizaje de la métrica de estilo

AUC															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,83502	0,00436	0,83938	0,83502	0,00436	0,8394	0,835	0,00436	0,83938	0,83502	0,00436	0,83938	0,83502	0,00436
KNN	0,39545	0,39827	-0,0028	0,35065	0,35855	-0,0079	0,3386	0,3512	-0,01261	0,32053	0,33427	-0,01374	0,28628	0,31131	-0,02503
MAP	0,39545	0,39827	-0,0028	0,35206	0,35507	-0,00301	0,3026	0,3062	-0,00367	0,24671	0,24962	-0,00291	0,18814	0,19279	-0,00465
MRR	0,15437	0,15813	-0,0038	0,15437	0,15813	-0,00376	0,1544	0,1581	-0,00376	0,15437	0,15813	-0,00376	0,15437	0,15813	-0,00376
NDCG	0,45769	0,46331	-0,0056	0,45769	0,46331	-0,00562	0,4577	0,4633	-0,00562	0,45769	0,46331	-0,00562	0,45769	0,46331	-0,00562
PREC	0,35206	0,35507	-0,003	0,35206	0,35507	-0,00301	0,3168	0,3203	-0,0035	0,27635	0,27939	-0,00304	0,22916	0,2333	-0,00414

(a) AUC

KNN															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814
KNN	0,39545	0,33465	0,0608	0,35065	0,29927	0,05138	0,3386	0,29851	0,04009	0,32053	0,28948	0,03105	0,28628	0,26558	0,0207
MAP	0,39545	0,33465	0,0608	0,35206	0,2955	0,05656	0,30256	0,25301	0,04955	0,24671	0,20709	0,03962	0,18814	0,15975	0,02839
MRR	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883
NDCG	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542
PREC	0,35206	0,2955	0,05656	0,35206	0,2955	0,05656	0,3168	0,26541	0,05139	0,27635	0,23194	0,04441	0,22916	0,19352	0,03564

(b) KNN

PREC															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814
KNN	0,39545	0,33465	0,0608	0,35065	0,29927	0,05138	0,3386	0,29851	0,04009	0,32053	0,28948	0,03105	0,28628	0,26558	0,0207
MAP	0,39545	0,33465	0,0608	0,35206	0,2955	0,05656	0,30256	0,25301	0,04955	0,24671	0,20709	0,03962	0,18814	0,15975	0,02839
MRR	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883	0,15437	0,12554	0,02883
NDCG	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542	0,45769	0,40227	0,05542
PREC	0,35206	0,2955	0,05656	0,35206	0,2955	0,05656	0,3168	0,26541	0,05139	0,27635	0,23194	0,04441	0,22916	0,19352	0,03564

(c) PREC@k

MAP															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,85414	-0,01476	0,83938	0,85414	-0,01476	0,83938	0,85414	-0,01476	0,83938	0,85414	-0,01476	0,83938	0,85414	-0,01476
KNN	0,39545	0,31357	0,08188	0,35065	0,278	0,07265	0,3386	0,2748	0,0638	0,32053	0,26746	0,05307	0,28628	0,25033	0,03595
MAP	0,39545	0,31357	0,08188	0,35206	0,27819	0,07387	0,30256	0,23932	0,06324	0,24671	0,19824	0,04847	0,18814	0,15536	0,03278
MRR	0,15437	0,12501	0,02936	0,15437	0,12501	0,02936	0,15437	0,12501	0,02936	0,15437	0,12501	0,02936	0,15437	0,12501	0,02936
NDCG	0,45769	0,38367	0,07402	0,45769	0,38367	0,07402	0,45769	0,38367	0,07402	0,45769	0,38367	0,07402	0,45769	0,38367	0,07402
PREC	0,35206	0,27819	0,07387	0,35206	0,27819	0,07387	0,3168	0,25081	0,06599	0,27635	0,22078	0,05557	0,22916	0,18598	0,04318

(d) MAP

MRR															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,82966	0,00972	0,83938	0,82966	0,00972	0,83938	0,82966	0,00972	0,83938	0,82966	0,00972	0,83938	0,82966	0,00972
KNN	0,39545	0,33917	0,05628	0,35065	0,30604	0,04461	0,3386	0,29381	0,04479	0,32053	0,28628	0,03425	0,28628	0,26558	0,0207
MAP	0,39545	0,33917	0,05628	0,35206	0,29804	0,05402	0,30256	0,25475	0,04781	0,24671	0,20855	0,03816	0,18814	0,16189	0,02625
MRR	0,15437	0,12803	0,02634	0,15437	0,12803	0,02634	0,15437	0,12803	0,02634	0,15437	0,12803	0,02634	0,15437	0,12803	0,02634
NDCG	0,45769	0,40655	0,05114	0,45769	0,40655	0,05114	0,45769	0,40655	0,05114	0,45769	0,40655	0,05114	0,45769	0,40655	0,05114
PREC	0,35206	0,29804	0,05402	0,35206	0,29804	0,05402	0,3168	0,2673	0,0495	0,27635	0,23357	0,04278	0,22916	0,19551	0,03365

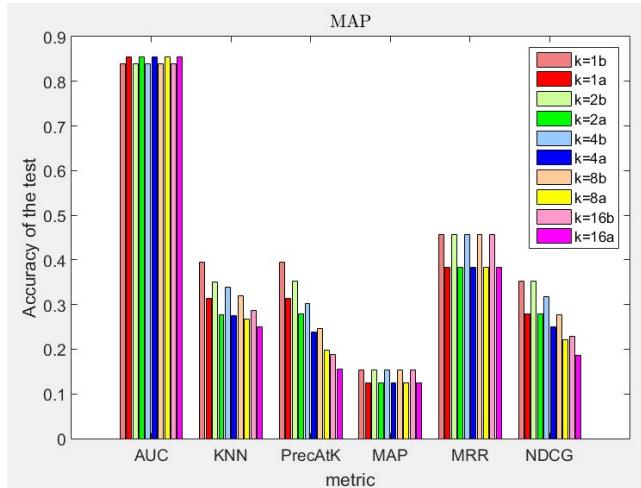
(e) MRR

NDCG															
K=	1B	1A	DIF1	2B	2A	DIF2	4B	4A	DIF4	8B	8A	DIF8	16B	16A	DIF16
AUC	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814	0,83938	0,83124	0,00814
KNN	0,39545	0,33465	0,0608	0,35065	0,29889	0,05176	0,3386	0,29832	0,04028	0,32053	0,28948	0,03105	0,28628	0,26558	0,0207
MAP	0,39545	0,33465	0,0608	0,35206	0,29531	0,05675	0,30256	0,25301	0,04955	0,24671	0,20711	0,0396	0,18814	0,15975	0,02839
MRR	0,15437	0,12555	0,02882	0,15437	0,12555	0,02882	0,15437	0,12555	0,02882	0,15437	0,12555	0,02882	0,15437	0,12555	0,02882
NDCG	0,45769	0,40215	0,05554	0,45769	0,40215	0,05554	0,45769	0,40215	0,05554	0,45769	0,40215	0,05554	0,45769	0,40215	0,05554
PREC	0,35206	0,29531	0,05675	0,35206	0,29531	0,05675	0,3168	0,26536	0,05144	0,27635	0,23192	0,04443	0,22916	0,1935	0,03566

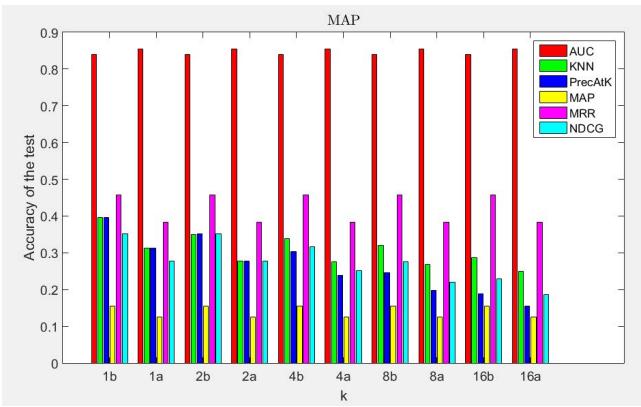
(f) NDCG

Figura 4.1: Error del test para las funciones de ranking aprendidas. En cada una de las tablas se muestra el error de test para cada una de las funciones de ranking de la columna de la izquierda con la métrica aprendida que encabeza la tabla. Por ejemplo, la tabla (f) muestra el error en el test obtenido entrenando para la métrica NDCG.

4. Aprendizaje de la métrica de estilo



(a) EjeX: precisión del test; EjeY: métricas



(b) EjeX: precisión del test; EjeY: k

Figura 4.2: (a) Precisión del test para cada una de las métricas con el sistema entrenado para MAP. (b) Precisión del test para cada uno de los valores de k antes y después del entrenamiento con el sistema entrenado para MAP.

En las Figuras 4.3, 4.4, 4.5 y en el Apéndice C se pueden ver algunos de los resultados obtenidos. En las distintas figuras se ven iconos enmarcados en morado, los cuales corresponden al ícono consulta, e iconos enmarcados en verde, son íconos de la misma colección que el ícono consulta, es decir, han sido diseñados por el mismo autor, y etiquetados en el mismo grupo. En cada una de las figuras se pueden ver tres búsquedas diferentes, de arriba a abajo (top-down) se pueden ver el emparejamiento de los datos con los pesos de las características de Garcés et al. [GAGH14] en primer lugar; seguidamente se ven los emparejamientos previamente a entrenar la métrica; y por último, encontramos los emparejamientos finales con la métrica aprendida. Visualizando las imágenes se puede apreciar una clara mejora en el estilo del conjunto de íconos con la medida de clasificación que usamos respecto a la de Garcés et al. [GAGH14]. En los distintos resultados apenas aparecen imágenes de la misma colección que la consulta, esto es porque dichos íconos no se han tratado de forma especial en el entrenamiento de la métrica; el dataset contiene una

4. Aprendizaje de la métrica de estilo

gran cantidad de iconos y el ícono consulta es similar al resto de íconos de su colección, pero también a los de otras colecciones, cuyo número en la mayoría de las veces es más elevado.

4. Aprendizaje de la métrica de estilo

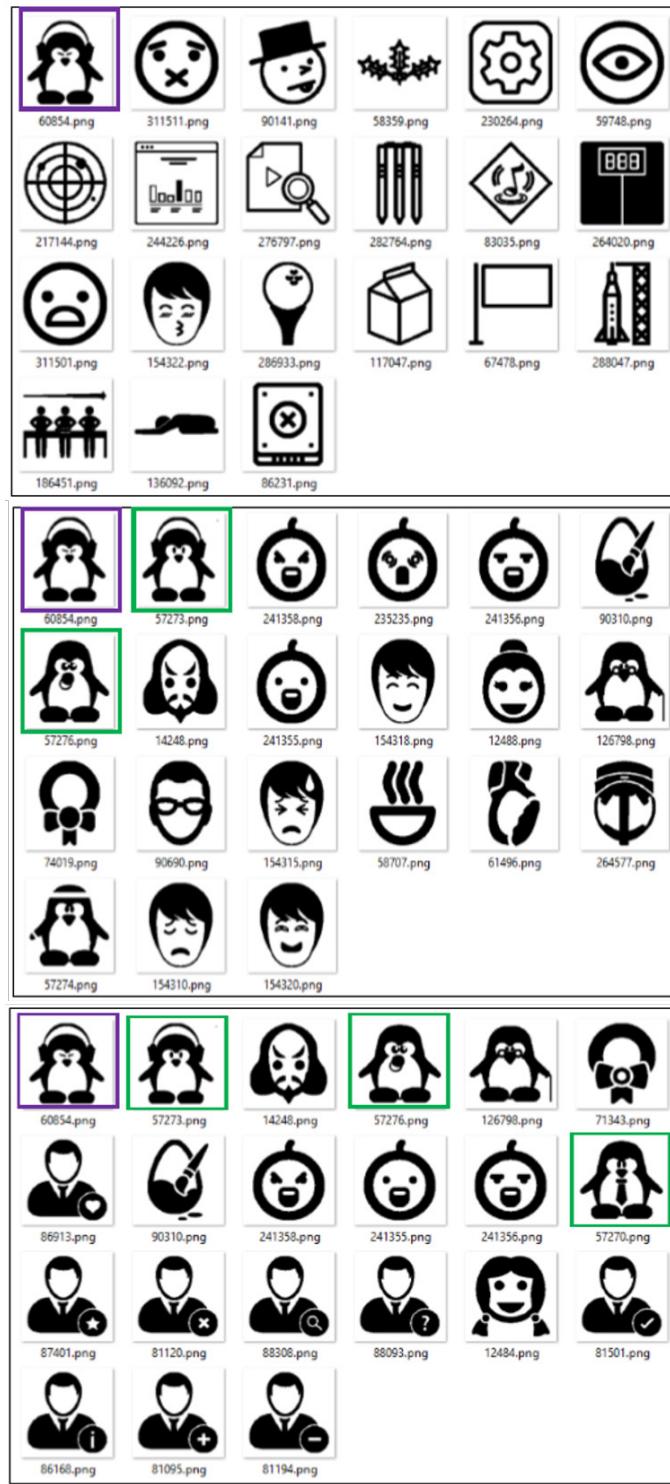


Figura 4.3: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos íconos que tienen el mismo autor que el de la búsqueda.

4. Aprendizaje de la métrica de estilo



Figura 4.4: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

4. Aprendizaje de la métrica de estilo



Figura 4.5: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

Capítulo 5

Aplicaciones

5.1. Clustering

Una de las posibles aplicaciones que tiene aprender una métrica de distancia es poder realizar clustering de los datos. Para realizar el clustering de los datos se utiliza *K-means*, un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n elementos en k grupos en el que cada elemento pertenece al grupo más cercano. Como resultado de este algoritmo se obtiene un nuevo etiquetado de los datos, en los que se agrupan mediante similaridad en estilo, ya que como medida de distancia al centroide se utilizan las propias características de los iconos aplicando la métrica aprendida en la Sección 4.2.

Es de gran importancia evaluar el resultado del clustering, sin embargo, es difícil definir cuando el resultado de un agrupamiento es aceptable. En nuestro caso queremos agrupar iconos similares en estilo en el mismo clúster y ubicar aquellos cuyo estilo sea distinto en diferentes clúster. Para ello calculamos la métrica de error para cada k como sigue:

$$E_k = \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^{N_i} \frac{d(x_{ij}, c_i)^2}{N_i} \quad (5.1)$$

donde k es el número de clústers, x_{ij} es el vector de características de cada elemento del cluster, N_i es el número total de elementos del cluster i , c_i es el valor del centroide del clúster correspondiente, y d es la distancia definida en la Ecuación 4.2.

En la Figura 5.1 se representa en el eje de abcisas el número de clusters y en el eje de ordenadas la tasa de error correspondiente E_k . Se puede ver que a partir de 30 clusters no hay mejora perceptible en la estimación del error de modo que se escoge 30 como el número óptimo de clusters. Probamos a sacar resultados con menos clusters pero visualmente no se percibían diferencias significativas entre los clusters. Por lo tanto los iconos serán divididos en 30 grupos según su coherencia en estilo. En la Figura 5.2 (a), (b) y (c) se muestran tres de los 30 conjuntos obtenidos en el clustering y en el Apéndice D se puede consultar el resto. Visualizando los distintos conjuntos se puede ver como los iconos agrupados en ellos mantienen coherencia en el estilo. Los iconos pertenecientes al grupo representado en la Figura 5.2(a) son iconos sin relleno, con un trazo similar en grosor y en estilo. En el segundo grupo representado en la Figura 5.2(b) los iconos tienden a ser circulares con

5. Aplicaciones

una gran parte de la imagen sombreada; sin embargo, el grupo de iconos representado en la Figura 5.2(c) tiende a mantener trazos más rectos.

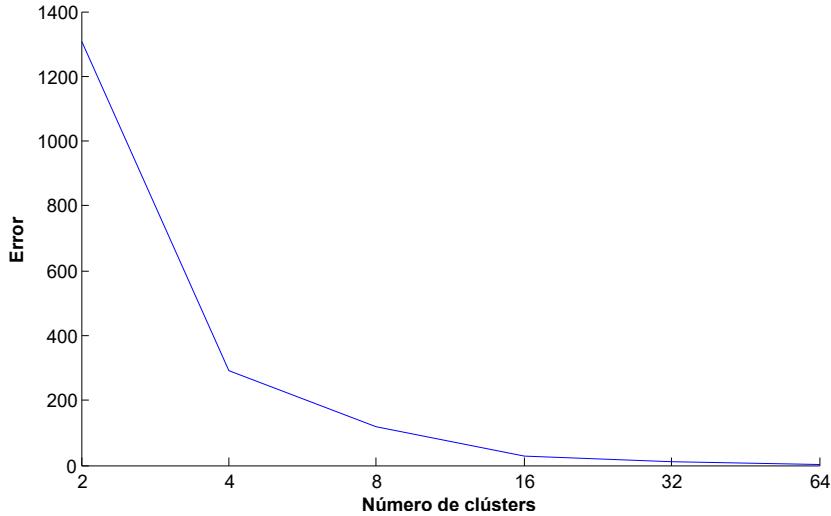


Figura 5.1: Gráfica que representa la selección del número óptimo de clusters.

5.2. Selección de conjuntos óptimos

Seleccionar el conjunto óptimo de un grupo de iconos es lo mismo que obtener un grupo de imágenes coherentes en estilo y cuyo sentido semántico coincide con la consulta realizada. Con el objetivo de obtener varios conjuntos óptimos para algunas consultas nos hemos basado en el trabajo de Garces et al. [GAGH16] donde se propone una herramienta de visualización de imágenes basada en estilo que tiene en cuenta el etiquetado semántico de las mismas.

El método citado se basa en un algoritmo de clustering jerárquico, que parte de un conjunto dado de imágenes etiquetadas y una métrica de similaridad de estilo. En el artículo original utilizan la métrica de similaridad de imágenes para clip art, sin embargo, en este proyecto utilizamos la métrica definida en la Sección 4.2, que como se ha visto funciona mejor en el dominio de los iconos. Siguiendo un enfoque de abajo hacia arriba (bottom-up), inicialmente, cada elemento comienza definiendo su propio clúster; y, en sucesivas iteraciones los clusters se fusionan en base a sus valores de mínima varianza dando lugar a una completa jerarquía. Los nodos de mínima varianza definirán clusters de estilo similar. Además, para que un cluster sea válido tiene que tener varias imágenes de las categorías semánticas definidas al comienzo. Para más información sobre este proceso, se ruega consultar el artículo original [GAGH16].

Como resultado de este proceso, obtenemos un conjunto de clústers. El tamaño de mínimo cada cluster y el número de imágenes de cada etiqueta que tiene que haber por cluster viene dado por dos parámetros, κ y τ respectivamente, donde para este proyecto se han definido con los valores $\kappa = 5$ y $\tau = 1$. Lo cual significa que este algoritmo denominado HClus generará grupos de elementos de como mínimo κ iconos donde habrá

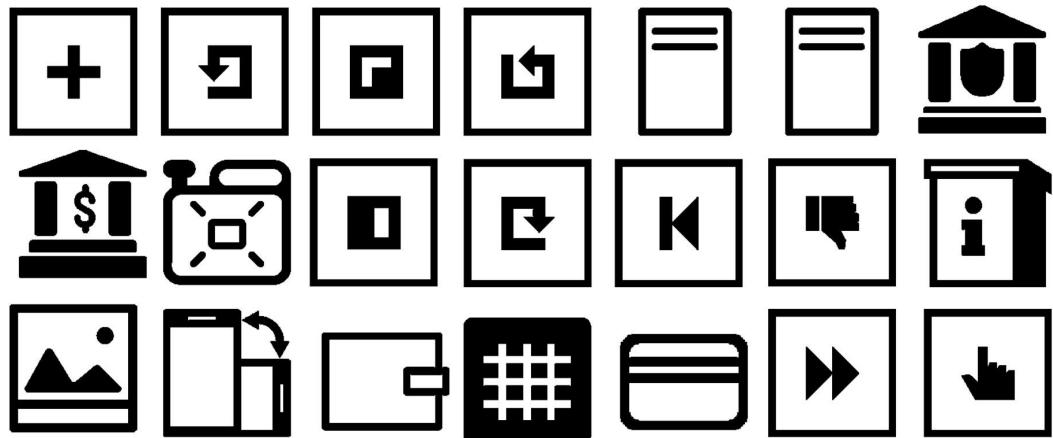
5. Aplicaciones



(a) Grupo 1



(b) Grupo 2



(c) Grupo 3

Figura 5.2: Ejemplos de tres conjuntos de iconos obtenidos en el clustering

5. Aplicaciones

al menos τ imágenes de cada etiqueta. Una vez el algoritmo termine habrá que realizar y una inspección visual para crear manualmente conjuntos óptimos para cada grupo.

En la Figura 5.3 se puede ver el resumen del proceso seguido para obtener los conjuntos óptimos en similaridad de estilo para una consulta. En primer lugar, se decide la consulta a realizar, seguidamente a partir de las características se crean subconjuntos de posibles emparejamientos, se realiza el clustering descrito anteriormente y tras inspeccionar el resultado de HClus donde puede haber múltiples conjuntos en los cuales aparezcan varias imágenes de una misma etiqueta, se seleccionan manualmente aquellos que mejor concuerden visualmente en estilo para obtener los conjuntos óptimos buscados. Este último proceso que por el momento es manual, se espera automatizar de manera que el algoritmo proponga múltiples conjuntos candidatos a partir de un clúster para cada consulta.

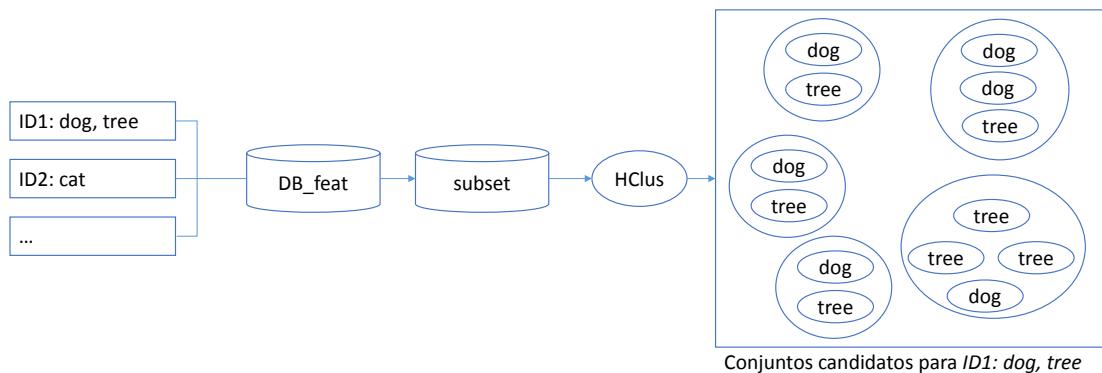


Figura 5.3: Esquema sobre la selección de conjuntos. El conjunto de iconos candidatos puede ser formado por varios grupos y contener un número variable de etiquetas, pero es indispensable que haya al menos una imagen de cada tipo. En este caso se obtienen cinco grupos de iconos candidatos en los que tres de ellos tienen una aparición cada etiqueta, dog y tree, pero sin embargo; hay dos grupos con más de una aparición por etiqueta.

A continuación se muestran seis consultas con sus respectivos conjuntos óptimos que podrían aplicarse a situaciones reales:

- Consulta 1 - '*house hearth*'

En la Figura 5.4 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una aplicación de invitaciones de boda. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.

- Consulta 2 - '*house hearth man*'

En la Figura 5.5 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una aplicación de invitaciones de boda. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.

- Consulta 3 - '*house hearth woman*'

En la Figura 5.6 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una aplicación de invitaciones de boda. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.

5. Aplicaciones

- Consulta 4 - *'map hotel transport'*

En la Figura 5.7 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una aplicación de reserva de vacaciones. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.

- Consulta 5 - *'sun moon rain'*

En la Figura 5.8 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una aplicación de meteorológica. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.

- Consulta 6 - *'tools fruit clothes'*

En la Figura 5.9 se pueden ver varios conjuntos, separados en filas, que se podrían usar en una caja autoservicio de un supermercado o unos grandes almacenes, donde hay distintas categorías de productos. Se puede apreciar que los iconos del mismo conjunto son similares en estilo.



Figura 5.4: Cuatro distintos conjuntos óptimos para la consulta *'house hearth'*, uno por fila.



Figura 5.5: Cuatro distintos conjuntos óptimos para la consulta 'house hearth man', uno por fila.

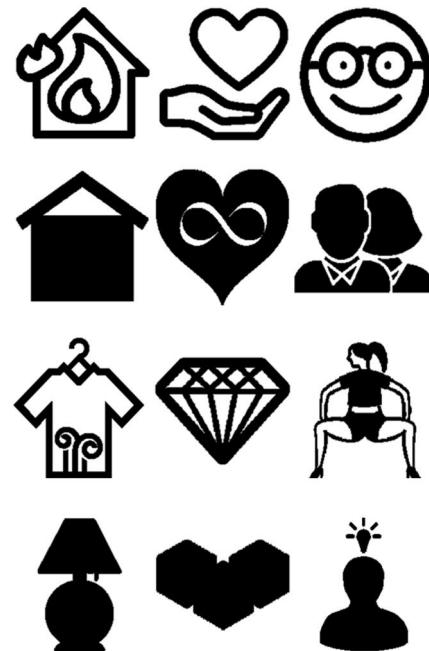


Figura 5.6: Cuatro distintos conjuntos óptimos para la consulta 'house hearth man', uno por fila.



Figura 5.7: Dos distintos conjuntos óptimos para la consulta 'map hotel transport', uno por fila.



Figura 5.8: Tres distintos conjuntos óptimos para la consulta 'sun moon rain', uno por fila.

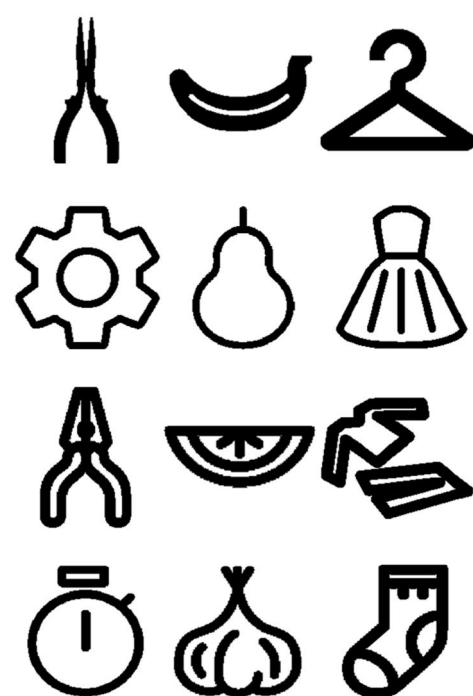


Figura 5.9: Cuatro distintos conjuntos óptimos para la consulta 'tools fruit clothes', uno por fila.

Capítulo 6

Conclusiones

6.1. Trabajo realizado

Elegir el mejor ícono para una interfaz es una tarea muy compleja, ya que de alguna manera es algo subjetivo, diferentes personas pueden tener diferentes impresiones sobre la idea que un ícono representa. En este proyecto, se ha presentado un problema que obtiene conjuntos óptimos de datos coherentes en estilo y cuyo sentido semántico coincide con un ícono de consulta previamente dado. Para llegar a este objetivo final ha sido necesario estudiar problemas anteriores relacionados, analizar las imágenes con las que se ha trabajado, obtener una función de distancia para medir la similaridad de los íconos y pasar por una serie de etapas de trabajo para tratarlos de forma adecuada. Finalmente se han logrado los objetivos propuestos de un problema relacionado con la similaridad en estilo de imágenes.

6.1.1. Planificación

En la Figura 6.1 se muestra la planificación temporal del proyecto y que se resume en:

- **Estudio inicial** - En la primera fase del proyecto tuve que analizar lo que queríamos conseguir, conocer la línea de investigación de la que partía el proyecto, así como pensar y analizar los pasos que debíamos seguir para conseguir el objetivo que se perseguía. Además, necesité un rápido proceso de aprendizaje en algunas herramientas y lenguajes de programación con los que íbamos a trabajar, los cuales desconocía.
- **Análisis de los datos** - El proceso seguido desde el diseño de la base de datos hasta tener conseguir un dataset robusto se realizó en varias fases y tuvo una duración aproximada de un mes. En primer lugar hubo que investigar como conseguir los íconos y conocer las herramientas necesarias para ello, para después poder almacenarlos en una base de datos. Seguidamente, se realizó una inspección visual del dataset, el cual se decidió reordenar para obtener un conjunto de datos coherente en estilo y por último se normalizaron los íconos para evitar errores innecesarios.
- **Aprendizaje de la métrica** - Una vez obtuvimos una colección de datos con la que trabajar nos dispusimos a aprender la métrica que mediría la similaridad de

6. Conclusiones

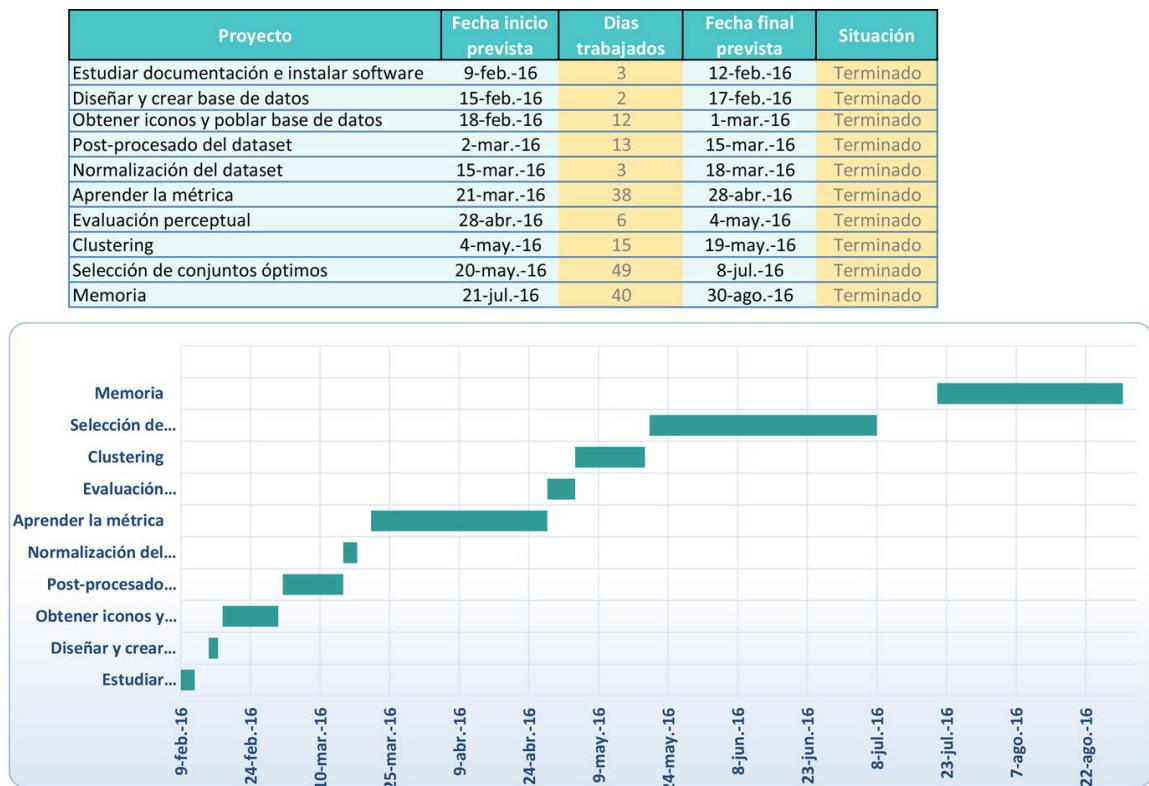


Figura 6.1: Diagrama de Gant de la planificación del proyecto

los iconos. Obtener dicha métrica y evaluar los resultados obtenidos con ella fue un proceso costoso que duró aproximadamente 40 días.

- **Aplicaciones** - Con la métrica aprendida se pensó en aplicaciones donde poder demostrar su utilidad. Este fue un largo proceso que incluye un clustering de los datos y la selección de conjuntos de iconos similares en estilo que se pudieran utilizar en distintas aplicaciones.
- **Memoria** - El presente documento es el resultado de esta etapa de la planificación y refleja el proyecto en general, intentando que el lector comprenda la necesidad y objetivos del proyecto, así como las diferentes etapas del proceso.

6.2. Trabajo futuro

El proyecto realizado da pie a nuevas líneas de investigación y a continuación se van a proponer algunas opciones.

En primer lugar, cómo ya se ha podido apreciar a lo largo del trabajo la métrica obtenida no es demasiado buena debido a que el etiquetado del dataset con el que se ha trabajado no es un buen etiquetado. Hay demasiados grupos pequeños de iconos que posiblemente comparten un determinado estilo y se están tratando como estilos distintos. En una futura línea de trabajo que resolviera este problema estaría bien lanzar test de

6. Conclusiones

usuario con el objetivo de crear conjuntos de datos de mayor tamaño uniendo aquellos cuyo estilo es similar.

Una segunda opción de trabajo podría ser diseñar nuevas características para medir la identificabilidad de un ícono del mismo modo que en este proyecto se ha tratado la similaridad en estilo.

Por último, y a partir de este proyecto se han visualizado colecciones de datos formadas por el mismo ícono, pero cuyo estilo es distinto, un ejemplo de estos datos puede verse en la Figura 3.8. Quizá sería interesante analizar las características de estos conjuntos de datos y la forma en que los usuarios los identifican más rápidamente.

6.3. Conclusión personal

Tras la realización de este trabajo y el contacto mantenido con el resto de personas en el laboratorio del Grupo de Informática Gráfica Avanzada (GIGA), dentro del Graphics and Imaging Lab (GILab), he logrado conocer varios temas y proyectos interesantes y novedosos de la informática, a parte de los conocimientos necesarios para este proyecto.

Personalmente, me ha aportado beneficios en cuanto a conocimientos, así como a aprender a superar retos complicados y que hay que afrontar de forma positiva, si es necesario compartiéndolos con otras personas que pueden proporcionar un punto de vista diferente y que puede ayudar en la solución del problema.

Al principio, seguramente como el resto de compañeros, me encontraba perdida o bloqueada en algunos momentos debido a que ha sido una nueva forma de trabajo, afrontando un problema completo, en el que quizás se necesita tener la mente más abierta que en el resto de proyectos realizados hasta el momento. La falta de familiarización previa con algunos instrumentos de trabajo han sido causa de frustración en algunos instantes; sin embargo, el trabajo constante y el esfuerzo invertido en el proyecto me han ayudado a superar estos retos con satisfacción.

Creo que he sabido adaptarme bastante bien al cambio de metodología de trabajo y que los resultados conseguidos tras la elaboración del trabajo son satisfactorios.

Bibliografía

- [BB15] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.
- [GAGH14] Elena Garces, Aseem Agarwala, Diego Gutierrez, and Aaron Hertzmann. A similarity measure for illustration style. *ACM Transactions on Graphics*, 33, 2014.
- [GAGH16] Elena Garces, Aseem Agarwala, Diego Gutierrez, and Aaron Hertzmann. Style-based exploration of illustration datasets. *Multimedia Tools and Applications*, 2016.
- [LHLF15] Tianqiang Liu, Aaron Hertzmann, Wilmot Li, and Thomas Funkhouser. Style compatibility for 3d furniture models. *ACM Transactions on Graphics (TOG)*, 34(4):85, 2015.
- [LKC⁺16] Lasse Farnung Laursen, Yuki Koyama, Hsiang-Ting Chen, Elena Garces, Diego Gutierrez, Richard Harper, and Takeo Igarashi. Icon set selection via human computation. *Pacific Graphics Short Papers*, 2016.
- [LKS15] Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. Elements of style: learning perceptual shape style similarity. *ACM Transactions on Graphics (TOG)*, 34(4):84, 2015.
- [ML10] Brian McFee and Gert Lanckriet. Metric learning to rank. *Proceedings of the 27th annual International Conference on Machine Learning (ICML)*, 2010.
- [SDHL15] Babak Saleh, Mira Dontcheva, Aaron Hertzmann, and Zhicheng Liu. Learning style similarity for searching infographics. In *Proceedings of the 41st Graphics Interface Conference*, pages 59–64. Canadian Information Processing Society, 2015.

Apéndice A

Anexo I

A.1. Tablas de la Base de datos

Collection El Cuadro A.1 hace referencia a los datos correspondientes a la entidad *Collection* de la base de datos.

ID	SLUG	AUTHOR	SIDS
1	noun-collection	6	"NIL"
5	modern-pictograms	2113	"NIL"
6	health-care-symbols	6	"NIL"
35	pocket-pictograms	101911	"NIL"
37	othericons	97345	"NIL"
38	automoticons	181519	"NIL"

Cuadro A.1: Ejemplo de datos almacenados en la entidad Collection.

Icon El Cuadro A.2 hace referencia a los datos correspondientes a la entidad *Icon* de la base de datos.

ID	COLLECTION	URL	PATH
567	6	https://d30y9cdsu7xlg0.cloudfront.net/png/567-200.png	/DATA/health-care-symbols/6/567.png
569	6	https://d30y9cdsu7xlg0.cloudfront.net/png/569-200.png	/DATA/health-care-symbols/6/569.png
573	6	https://d30y9cdsu7xlg0.cloudfront.net/png/573-200.png	/DATA/health-care-symbols/6/573.png
20949	37	https://d30y9cdsu7xlg0.cloudfront.net/png/20949-200.png	/DATA/othericons/97345/20949.png
20950	37	https://d30y9cdsu7xlg0.cloudfront.net/png/20950-200.png	/DATA/othericons/97345/20950.png
20951	37	https://d30y9cdsu7xlg0.cloudfront.net/png/20951-200.png	/DATA/othericons/97345/20951.png

Cuadro A.2: Ejemplo de datos almacenados en la entidad Icon.

A. Anexo I

Image El Cuadro A.3 hace referencia a los datos correspondientes a la entidad *Image* de la base de datos.

ID_ICON	FEATURES
579	[0,08885300089796122, 0,5526348523625589, ... 0,9866840541782287]
580	[0,08590247245711549, 0,05441961780952063, ... 0,9894736467562757]
2601	[0,3961752448377659, 0,9662956407776481, ... 0,9965783903534893]
2602	[0,08135553691861021, 0,5308286696563436, ... 0,9948566886842258]

Cuadro A.3: Ejemplo de datos almacenados en la entidad Image.

Tag El Cuadro A.4 hace referencia a los datos correspondientes a la entidad *Tag* de la base de datos.

ID_TAG	ID_ICON	NAME
23	630	human
410	630	medicine
602	630	person
7279	6250	french-bulldog
3118	6250	bulldog
399	6250	dog
1309	61393	christmas_tree
213	61393	tree

Cuadro A.4: Ejemplo de datos almacenados en la entidad Tag.

Apéndice B

Anexo III

B.1. Métricas

B.1.1. AUC

La métrica AUC (Area Under the ROC curve) se utiliza comúnmente en problemas de clasificación binaria.

Una curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

En un problema de predicción de clases binario, como el propuesto, en la que los resultados se etiquetan positivos (p) o negativos (n), hay cuatro posibles resultados. Si el resultado de una exploración es p y el valor dado también es p, entonces se conoce como un Verdadero Positivo (VP); sin embargo, si el valor real es n, entonces se conoce como un Falso Positivo (FP). De igual modo, tenemos un Verdadero Negativo (VN) cuando tanto la exploración como el valor dado son n, y un Falso Negativo (FN) cuando el resultado de la exploración es n pero el valor real es p. Esto se resume en la Figura B.1.

		Realidad	
		p	n
Predicción	p'	VP	FP
	n'	FN	VN

Figura B.1: Matriz de confusión de positivos y negativos.

En una curva ROC sólo son necesarias las razones de verdaderos positivos (VPR) y falsos positivos (FPR). La VPR mide hasta qué punto un clasificador es capaz de detectar o clasificar los casos positivos correctamente, de entre todos los casos positivos disponibles durante la prueba. La FPR define cuántos resultados positivos son incorrectos de entre todos los casos negativos disponibles durante la prueba.

Un espacio ROC (ver Figura B.2) se define por FPR y VPR como ejes x e y respectivamente, y representa los intercambios entre verdaderos positivos y falsos positivos. Dado que VPR es equivalente a sensibilidad

$$VPR = VP/P = VP/(VP + FN) \quad (B.1)$$

y FPR es igual a 1-especificidad,

$$FPR = VP/P = VP/(VP + FN) \quad (B.2)$$

$$SPC = VN/N = VN/(FP + VN) = 1 - FPR \quad (B.3)$$

$$1 - FPR = 1 - FP/N = 1 - FP/(FP + VN) = VN/(FP + VN) = SPC \quad (B.4)$$

el gráfico ROC también es conocido como la representación de sensibilidad frente a (1-especificidad). Cada resultado de predicción o instancia de la matriz de confusión representa un punto en el espacio ROC.

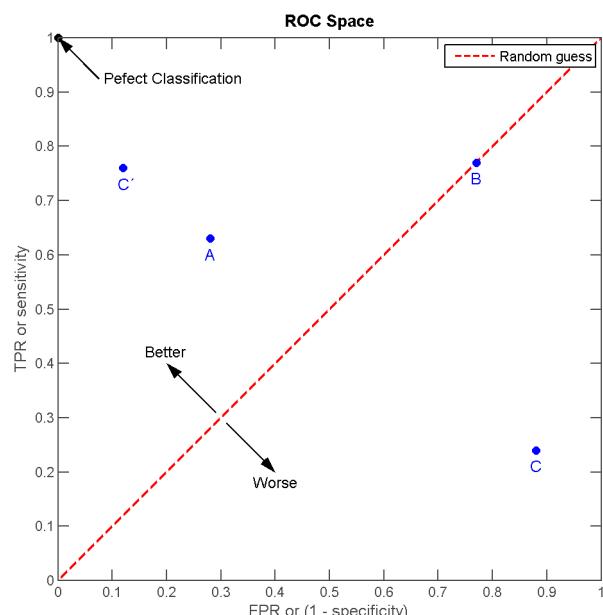


Figura B.2: El espacio ROC y cuatro ejemplos de predicción A, B, C y C'. (Fuente: Wikipedia)

El mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100 % de sensibilidad (ningún falso negativo) y un 100 % también de especificidad (ningún falso positivo). A este punto (0,1) también se le llama una clasificación perfecta. Por el contrario, una clasificación totalmente aleatoria daría un punto a lo largo de la línea diagonal, que se llama también línea de no-discriminación, desde el extremo inferior izquierdo hasta la esquina superior derecha. Un ejemplo típico de adivinación aleatoria sería decidir a partir de los resultados de lanzar una moneda al aire, a medida que el tamaño de la muestra aumenta, el punto de un clasificador aleatorio de ROC se desplazará hacia la posición (0.5, 0.5).

B. Anexo III

La diagonal divide el espacio ROC. Los puntos por encima de la diagonal representan los buenos resultados de clasificación (mejor que el azar), puntos por debajo de la línea de los resultados pobres (peor que al azar). Nótese que la salida de un predictor consistentemente pobre simplemente podría ser invertida para obtener un buen predictor.

En la Figura B.2 se muestran cuatro puntos de ejemplo en el espacio ROC. El resultado del método A muestra claramente ser el mejor de entre los métodos A, B Y C. El resultado de B se encuentra sobre la línea de estimación aleatoria (diagonal); mientras que el método C aparece como el peor de los tres, con un resultado muy pobre.

Sin embargo, consideremos ahora la construcción de un cuarto método de predicción C' que simplemente invierte los resultados predichos por el método C y su punto en el espacio ROC estará ahora por encima de la diagonal, y más próximo al punto de clasificación perfecta que el método A. Mientras C presentaba un pobre poder de predicción, a partir de él se ha construido un predictor mejor que todos los demás. Cuando el método C predice 'n' o 'p', el método C' predice 'p' o 'n' respectivamente. Siempre que un método presente un punto en el espacio ROC por debajo de la diagonal habrá que invertir sus predicciones para aprovechar su capacidad de predicción.

Cuanto más cerca esté un método de la esquina superior izquierda (clasificación perfecta) mejor será, pero lo que en realidad marca el poder predictivo de un método es la distancia de este a la línea de estimación aleatoria, da igual si por arriba o por abajo.

B.1.2. Precision-at-k y KNN

En recuperación de información hay dos conceptos clave que miden la relevancia de los datos, el recall y la precisión.

Se define como recall o exhaustividad, la fracción de documentos relevantes para una consulta que fueron recuperados.

$$Recall = \frac{\text{documentosrelevantes} \cap \text{documentosrecuperados}}{\text{documentosrelevantes}} \quad (B.5)$$

Resulta trivial obtener un 100 % de exhaustividad si se toman como respuesta para cualquier consulta todos los documentos de la colección. Por lo tanto, la exhaustividad sola no es suficiente, sino que se necesita también medir el número de documentos no relevantes, por ejemplo con el cálculo de la precisión.

La precisión se define cómo la fracción de documentos recuperados que son relevantes para la necesidad de información del usuario. Ésta, tiene en cuenta todos los documentos recuperados; sin embargo, también puede ser evaluada en un corte determinado del ranking, considerando solamente los primeros resultados obtenidos del sistema y esta es conocida como **precision-at-k** (Prec@k).

$$Precision = \frac{\text{documentosrelevantes} \cap \text{documentosrecuperados}}{\text{documentosrecuperados}} \quad (B.6)$$

Prec@k y **KNN** están estrechamente vinculadas. En clasificación binaria, las dos medidas de evaluación son relacionadas por

$$KNN(q, y; k) = 1[Prec@k(q, y) > 0,5] \quad (B.7)$$

,

de modo que la Prec@k puede ser fácilmente adaptada a los k-vecinos más cercanos. Sin embargo, en clasificación multiclas, esta técnica falla porque ya no sólo depende de la pertenencia o no de cada uno de los puntos al conjunto evaluado, sino que depende de las propias etiquetas de cada uno de ellos.

B.1.3. MAP

El MAP (Mean Average Precision, o Promedio de precisión media) de un conjunto de consultas es la media de las puntuaciones medias de precisión para cada consulta.

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q} \quad (B.8)$$

donde Q es el número de consultas.

La precisión media (AveP) de una lista de elementos tiene en cuenta el orden de los items recuperados. Mediante el cálculo de una precisión y la recuperación en cada posición en la secuencia ordenada de items, se puede trazar una curva de precisión-recall, calculando la precision $p(r)$ como una función de recall r . La precisión media se calcula como valor medio de $p(r)$ en el intervalo desde $r=0$ a $r=1$.

$$AveP = \int_0^1 p(r)dr \quad (B.9)$$

Este es el área bajo la curva precisión-recall. En la práctica, esta integral es sustituida por una suma finita sobre cada posición en la secuencia ordenada de los elementos.

$$AveP = \sum_{k=1}^n P(k)\Delta r(k) \quad (B.10)$$

donde k es la posición en la secuencia de items recuperados, n es el número de elementos recuperados, $P(k)$ es la precisión en la posición k de la lista y $\Delta r(k)$ es la variación en el recall entre los items $k-1$ y k .

Esta suma finita es equivalente a:

$$AveP = \frac{\sum_{k=1}^n (P(k)rel(k))}{NR} \quad (B.11)$$

donde $rel(k)$ es una función que vale 1 si el elemento en la posición k es relevante, 0 en caso contrario y NR es el número de elementos relevantes. Téngase en cuenta que la media es sobre todos los elementos relevantes, y los elementos relevantes no recuperados obtienen una precisión de 0.

B.1.4. MRR

El MRR (Mean Reciprocal Rank, o Rango Medio de Reprocidad) es una estadística de medida para la evaluación de cualquier proceso que produce una lista de posibles respuestas a una muestra de consultas, ordenada por la probabilidad de corrección. El rango recíproco de una respuesta de la consulta es el inverso de la multiplicación de la fila de la primera

B. Anexo III

respuesta correcta. El rango de reciprocidad media es el promedio de los rangos recíprocos de resultados para una muestra de consultas Q:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (B.12)$$

donde $rank_i$ se refiere a la posición del primer elemento relevante de la consulta i .

B.1.5. NDCG

La métrica NDCG (Normalized Discounted Cumulative Gain) es similar a MRR, con la diferencia de que no sólo tiene en cuenta el primer ítem relevante de la colección, sino los primeros k elementos. Éstos, son ponderados con un valor que varía de 0.0 a 1.0, siendo 1.0 la clasificación ideal de las entidades.

En el presente trabajo, con niveles de relevancia binaria, la formulación adoptada es

$$NDCG(q, y; k) = \frac{\sum_{i=1}^k D(i) 1[i \in \chi_q^+]}{\sum_{i=1}^k D(i)} \quad (B.13)$$

$$D(i) = \begin{cases} 1 & i = 1 \\ 1/\log_2(i) & 2 \leq i \leq k \\ 0 & i > k \end{cases} \quad (B.14)$$

NDCG puede no ser adecuado para medir el rendimiento de consultas que pueden tener varios resultados igualmente buenos. Esto es especialmente cierto cuando esta métrica se limita a sólo los primeros resultados como se hace en la práctica. Esto es, porque no se penalizan resultados malos o cuyos datos no se conocen. Por ejemplo, si una consulta devuelve dos resultados con los valores $\{1,1,1\}$ y $\{1,1,1,0\}$ respectivamente, ambos serán considerados igual de buenos, a pesar de que el segundo conjunto contenga un fallo en el resultado.

Apéndice C

Anexo II

C.1. Resultados del aprendizaje de la métrica

Completados el entrenamiento y test de los datos, como los resultados de la evaluación numérica no son concluyentes, es necesario una inspección visual a través de los datos y a continuación se pueden ver algunos de estos resultados.

En las distintas figuras se ven iconos enmarcados en morado, los cuales corresponden al ícono consulta, e iconos enmarcados en verde, son íconos de la misma colección que el ícono consulta, es decir, han sido diseñados por el mismo autor, y etiquetados en el mismo grupo. En cada una de las figuras se pueden ver tres búsquedas diferentes, de arriba a abajo se pueden ver el emparejamiento de los datos con los pesos de las características de Garces et al. [GAGH14] en primer lugar; seguidamente se ven los emparejamientos previamente a entrenar la métrica; y por último, encontramos los emparejamientos finales con la métrica aprendida.



Figura C.1: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

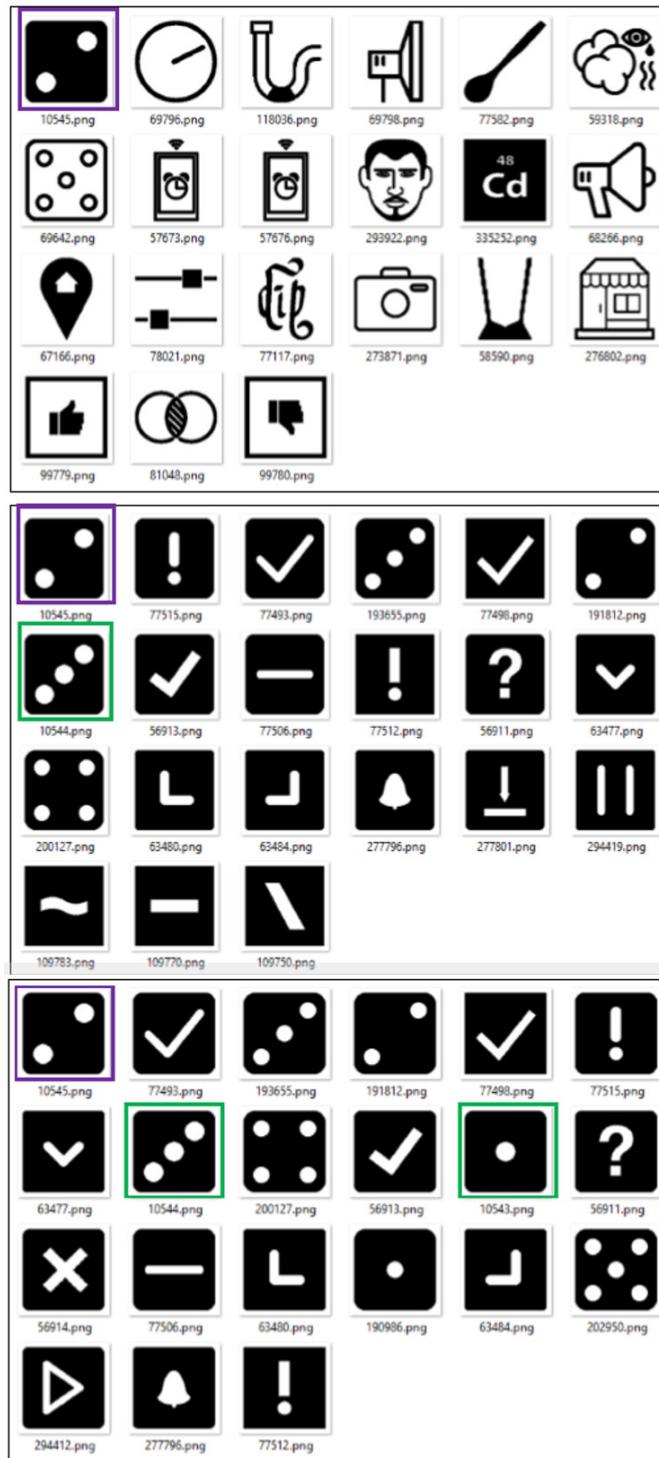


Figura C.2: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

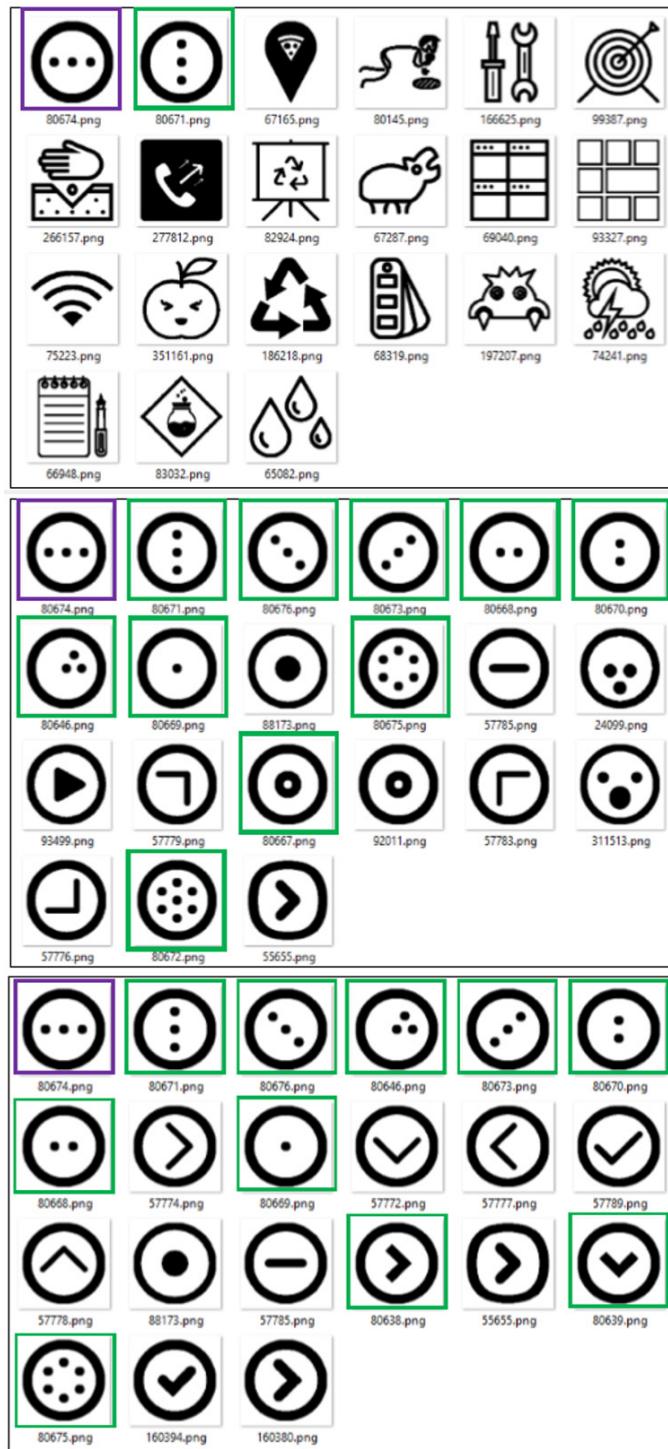


Figura C.3: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos íconos que tienen el mismo autor que el de la búsqueda.

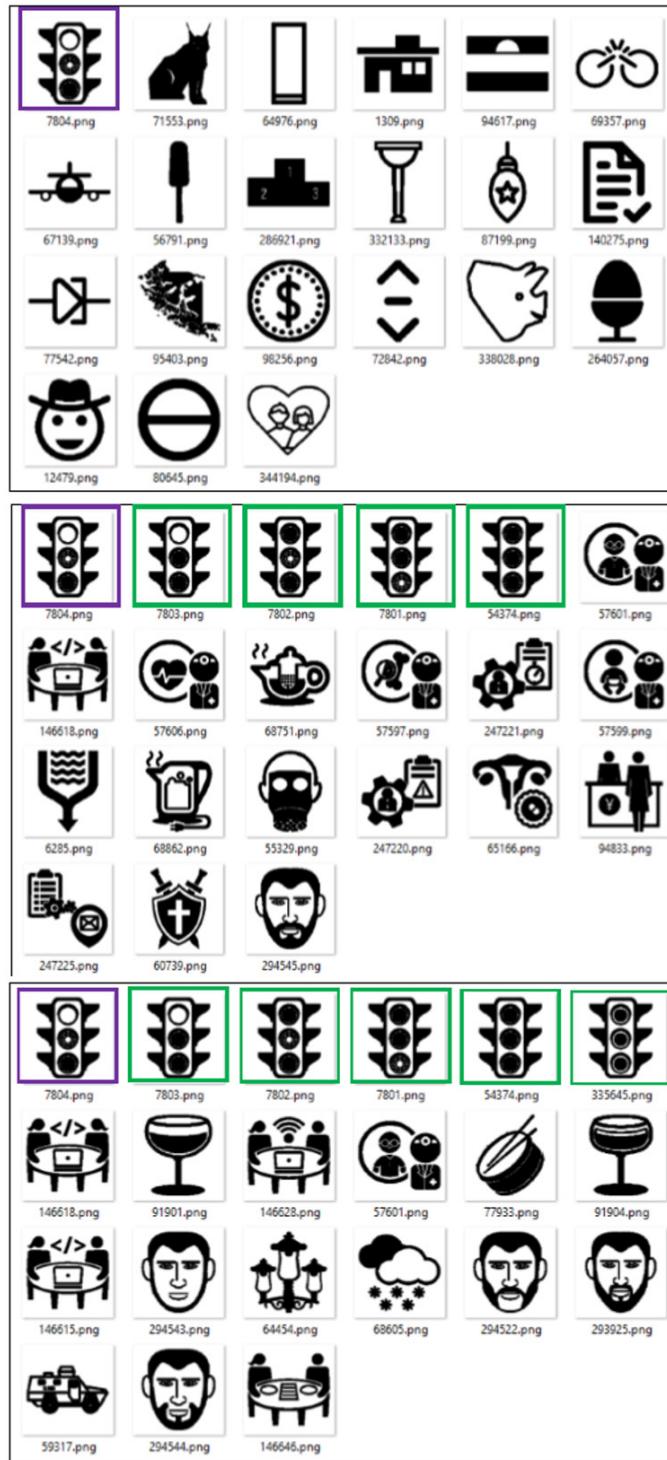


Figura C.4: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.



Figura C.5: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

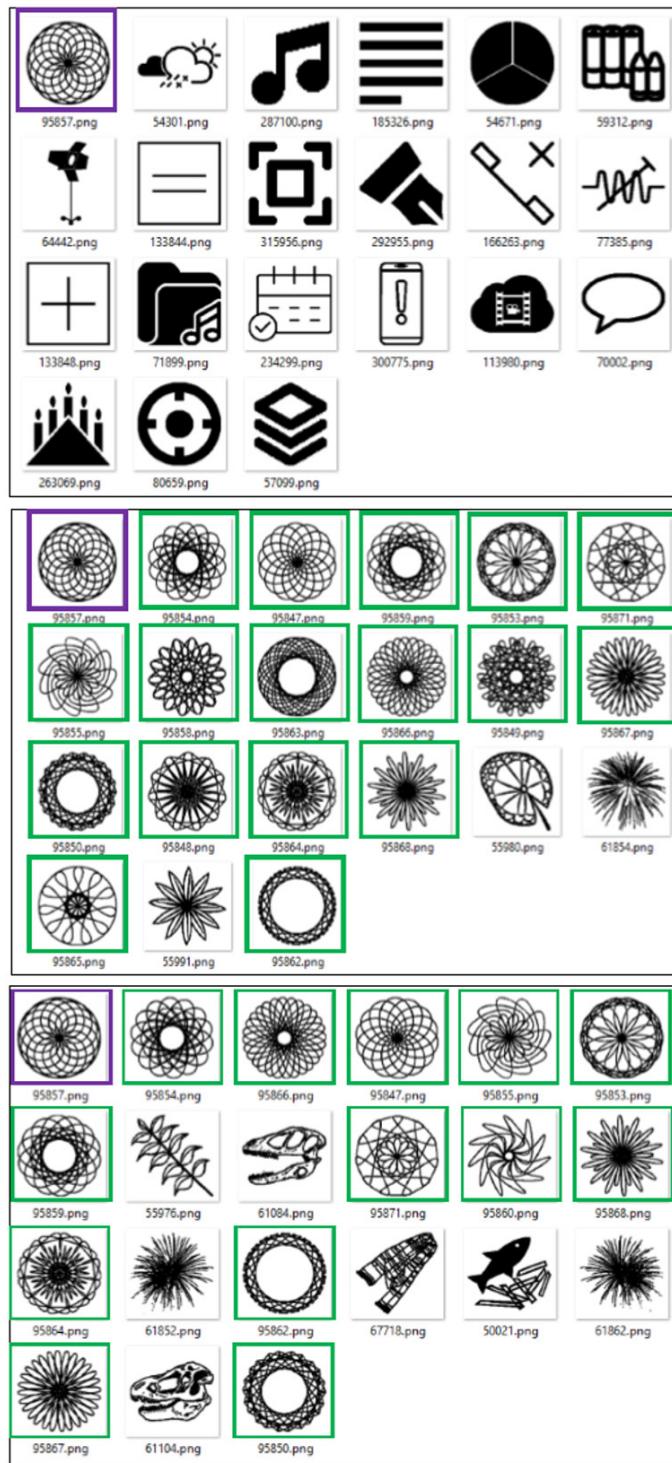


Figura C.6: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

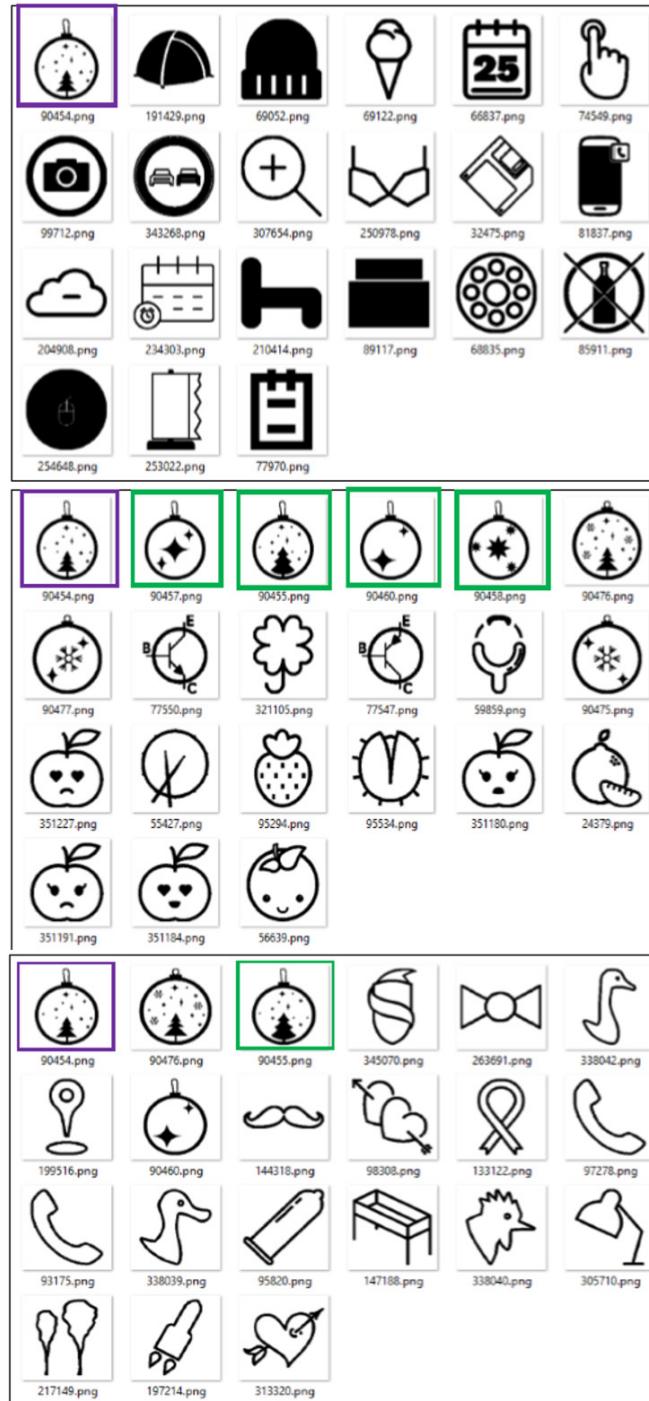


Figura C.7: Mejores emparejamientos de la similaridad de un ícono consulta con el resto de la colección. De arriba a abajo tenemos el resultado utilizando los pesos de características de Garces et al. [GAGH14], seguidamente el conjunto obtenido antes de entrenar la métrica y por último los resultados con la métrica aprendida. En verde están marcados aquellos iconos que tienen el mismo autor que el de la búsqueda.

Apéndice D

Anexo III

D.1. Resultados del Clustering

A continuación en las Figuras D.1- D.27 se muestran los resultados del clustering mencionados en la Sección 5.1.

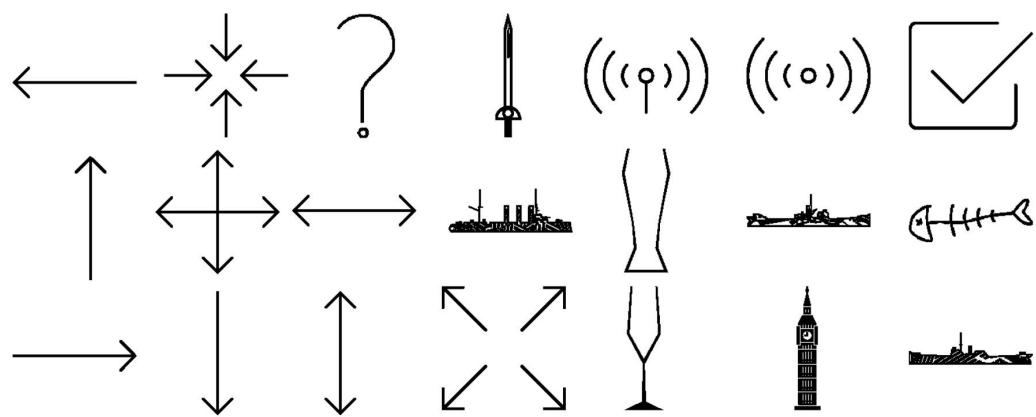


Figura D.1: Conjunto de iconos obtenido en el clustering



Figura D.2: Conjunto de iconos obtenido en el clustering



Figura D.3: Conjunto de iconos obtenido en el clustering



Figura D.4: Conjunto de iconos obtenido en el clustering

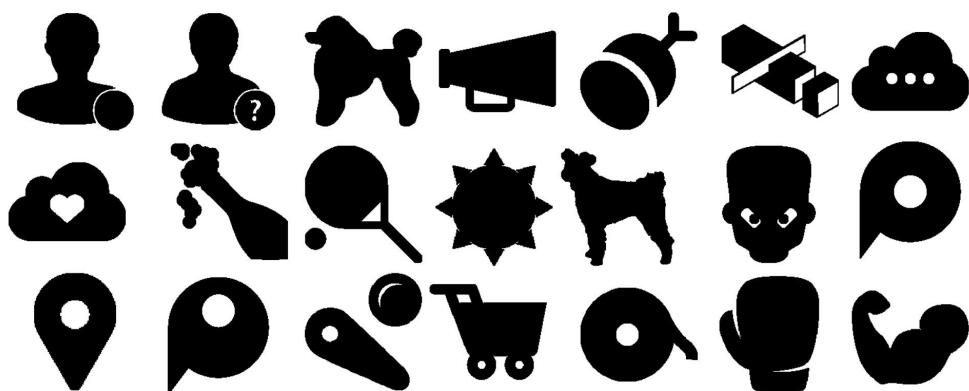


Figura D.5: Conjunto de iconos obtenido en el clustering



Figura D.6: Conjunto de iconos obtenido en el clustering



Figura D.7: Conjunto de iconos obtenido en el clustering



Figura D.8: Conjunto de iconos obtenido en el clustering

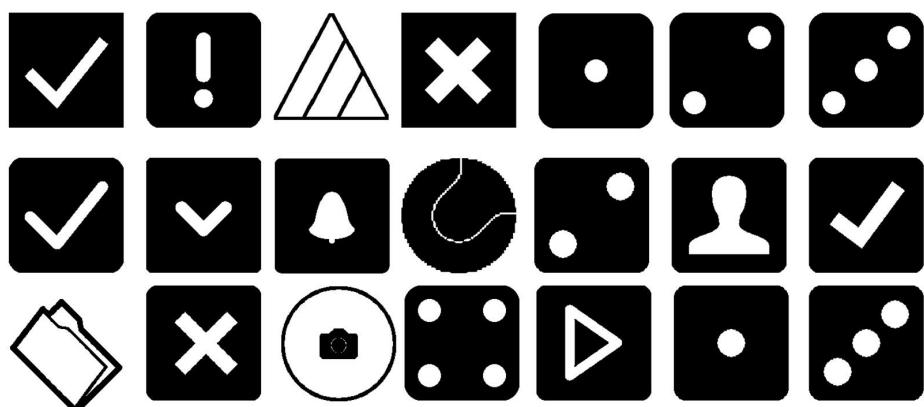


Figura D.9: Conjunto de iconos obtenido en el clustering

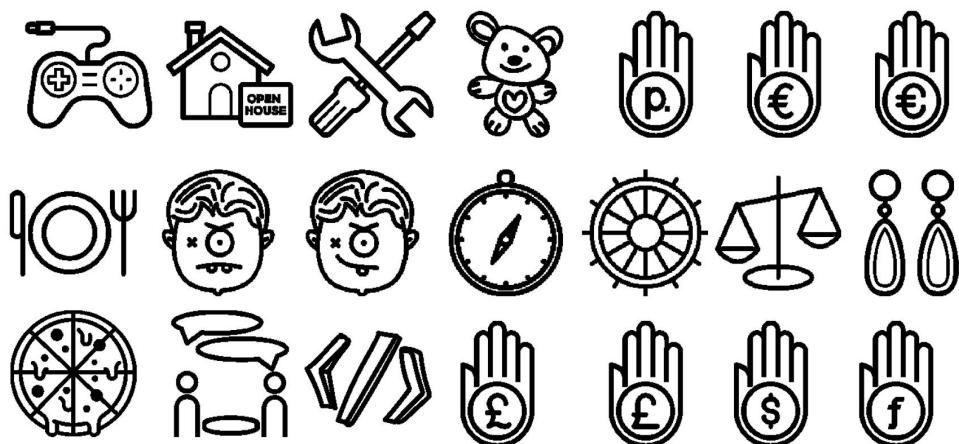


Figura D.10: Conjunto de iconos obtenido en el clustering



Figura D.11: Conjunto de iconos obtenido en el clustering



Figura D.12: Conjunto de iconos obtenido en el clustering

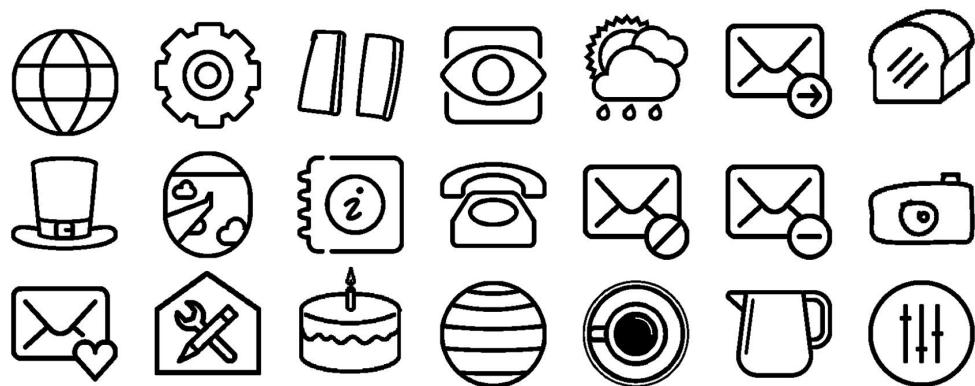


Figura D.13: Conjunto de iconos obtenido en el clustering



Figura D.14: Conjunto de iconos obtenido en el clustering

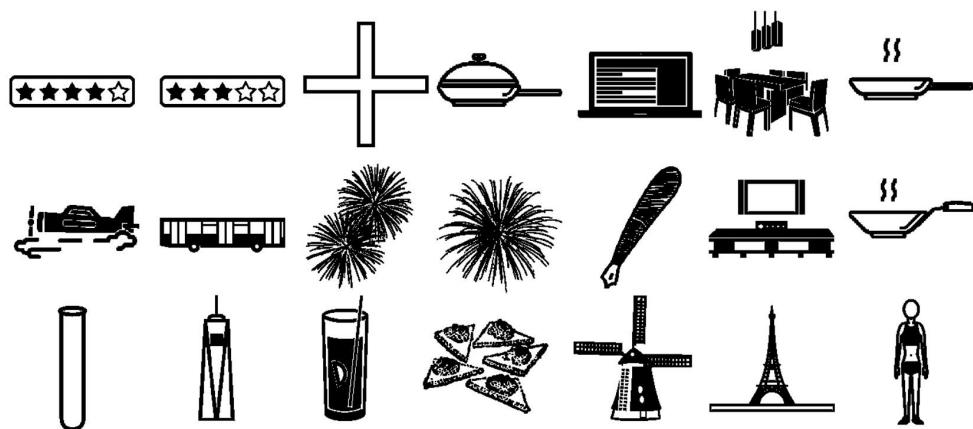


Figura D.15: Conjunto de iconos obtenido en el clustering

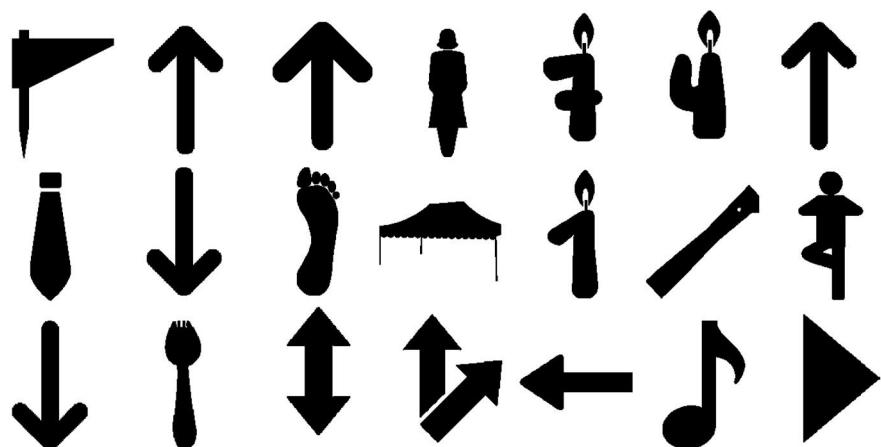


Figura D.16: Conjunto de iconos obtenido en el clustering

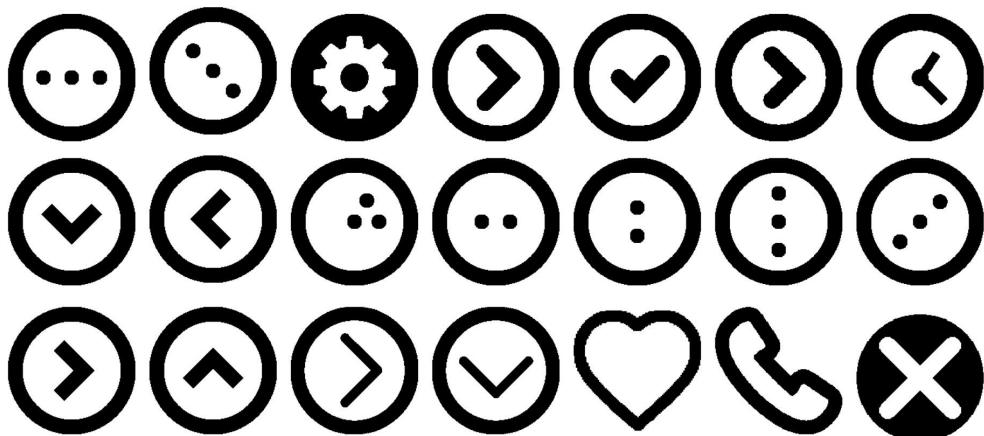


Figura D.17: Conjunto de iconos obtenido en el clustering

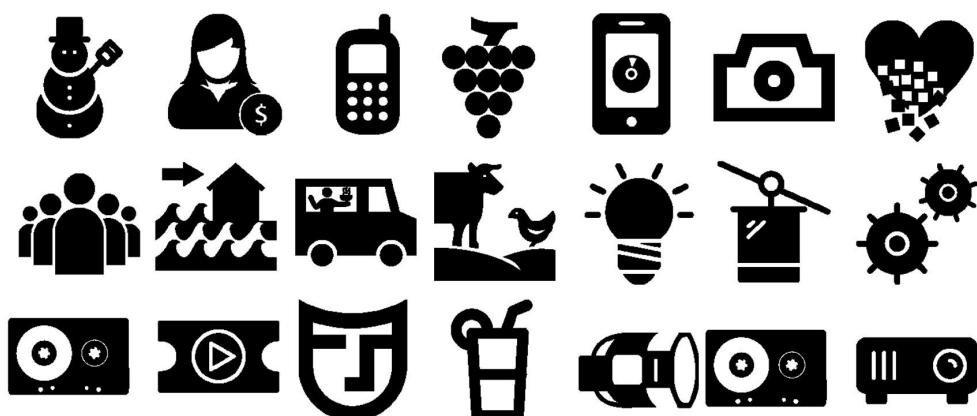


Figura D.18: Conjunto de iconos obtenido en el clustering



Figura D.19: Conjunto de iconos obtenido en el clustering

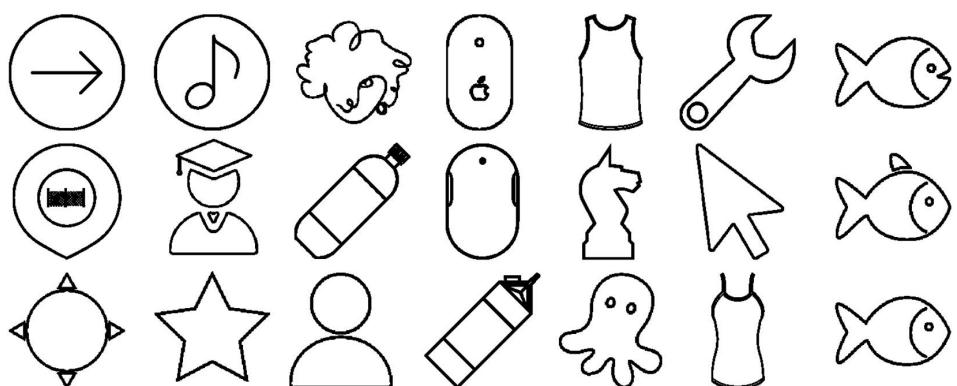


Figura D.20: Conjunto de iconos obtenido en el clustering



Figura D.21: Conjunto de iconos obtenido en el clustering

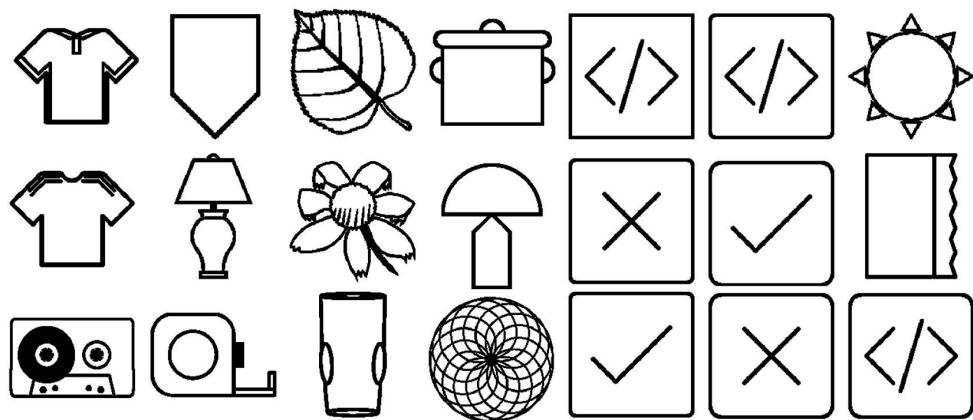


Figura D.22: Conjunto de iconos obtenido en el clustering

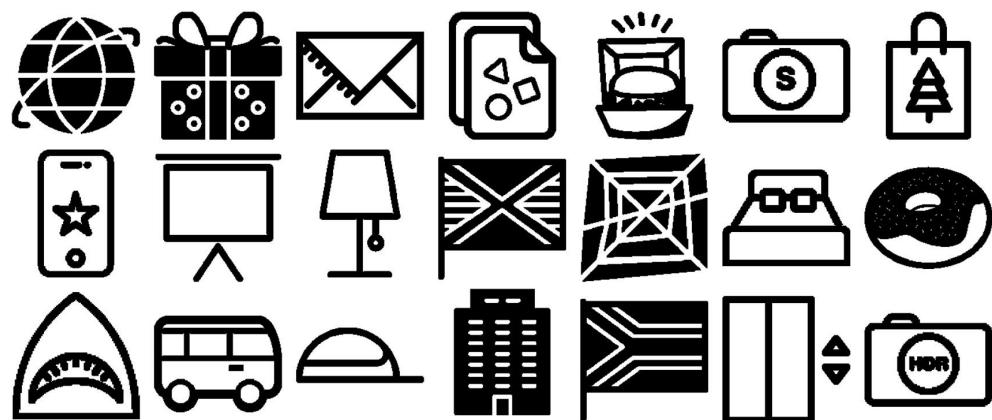


Figura D.23: Conjunto de iconos obtenido en el clustering

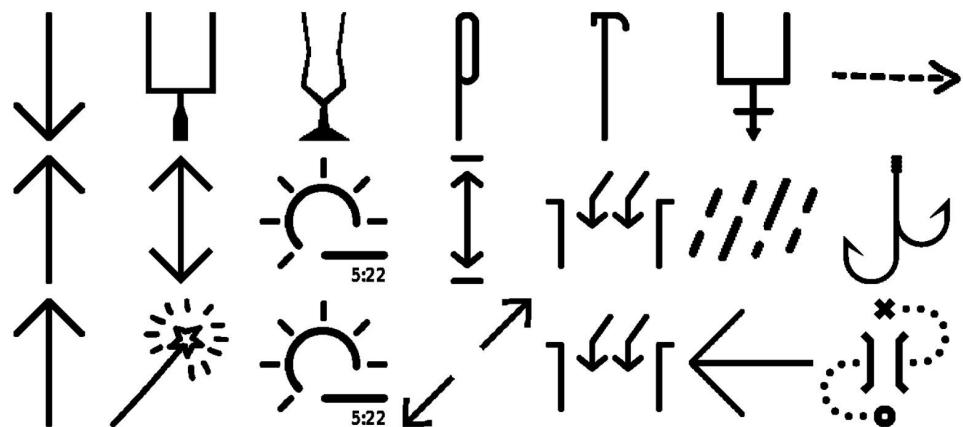


Figura D.24: Conjunto de iconos obtenido en el clustering

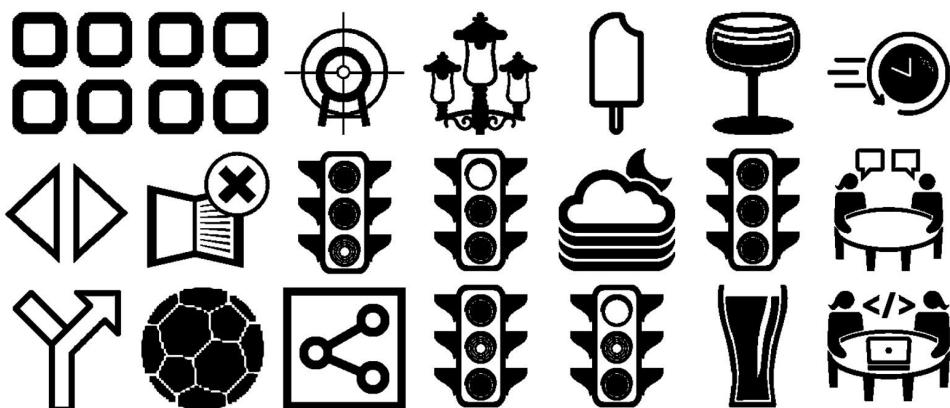


Figura D.25: Conjunto de iconos obtenido en el clustering



Figura D.26: Conjunto de iconos obtenido en el clustering



Figura D.27: Conjunto de iconos obtenido en el clustering