Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo fin de grado

Grado en Ingeniería Informática
(Mención en Tecnologías de la Información)

Análisis comparativo de generadores automáticos de pies de foto

Gonzalo de la Fuente Pelaz

Tutor:
Valentín Cardeñoso Payo

Resumen

El etiquetado de imagen consiste en la generación de etiquetas que sirvan para describir el contenido de imágenes. Para esto, es necesario detectar y reconocer los objetos o escenas presentes en una imagen e identificar aquellos relevantes.

En sus orígenes, el reconocimiento de imagen se limitaba al reconocimiento de patrones codificados manualmente, pero en los últimos años se ha evolucionado notablemente debido al empleo de aprendizaje automático profundo.

Actualmente, los etiquetadores de imagen reconocen automáticamente los objetos presentes en una imagen, así como sus propiedades y las relaciones entre estos; incluso permiten generar descripciones en lenguaje natural de bastante calidad semántica y sintáctica. Aunque el etiquetado de imagen siga siendo objeto de investigación activa, ya se ofrecen soluciones comerciales.

El objetivo de este trabajo es la realización de un experimento para evaluar la eficacia de cinco soluciones comerciales de etiquetado de imagen sobre un mismo conjunto de datos; lo que requiere emplear técnicas de procesamiento de lenguaje natural para comparar sus etiquetas.

Dicho experimento podrá servir de base para la construcción de un sistema que combine el etiquetado realizado por dichos sistemas y trate de mejorar las tasas de acierto y cobertura.

Agradecimientos

Me gustaría dedicar unas palabras a agradecer su presencia a las personas que me han acompañado y apoyado durante mi paso por la universidad y que, en definitiva, han hecho esto posible.

En primer lugar, a mis padres, Carlos y Mariola, y a mi hermano, Marcos, por apoyarme en los momentos más duros y compartir conmigo los buenos.

En segundo lugar, a mis amigos y compañeros de Grado, por su apoyo y presencia durante estos años, que gracias a ellos recordaré con alegría siempre.

A mis compañeros de Nielsen, por acompañarme durante la realización de este trabajo y haberme ayudado en aquello resultaba posible.

Y por último, a los profesores que se han involucrado en mi formación; y en especial a Valentín, por comprender y adaptarse a mi situación personal, por su labor durante la realización de este trabajo y por haber conseguido que obtenga unos conocimientos que valoro mucho.

A todos ellos, gracias.

Índice general

Capítulo 1

Introducción

1.1. Motivación

Actualmente estamos viviendo un intenso proceso de digitalización en el que la tecnología está aumentando su importancia en muchos ámbitos, lugares y acciones y automatizando funciones que estaban reservadas a los humanos gracias a los avances en inteligencia artificial.

La aplicación de Inteligencia Artificial (IA) en la actualidad se orienta hacia técnicas de Aprendizaje Automático basadas o dirigidas por datos, lo que requiere de una cantidad ingente de datos etiquetados a partir de los cuales se entrenan *modelos* capaz de generalizar las características de dichos datos.

Debido al coste y complejidad de generar datos etiquetados fiables, sobre todo en ámbitos tradicionales o casos de uso muy concretos, ha surgido una gran necesidad de métodos para la obtención de datos etiquetados:

- Empresas dedicadas al etiquetado de datos.
- Plataformas de *crowdsourcing*.
- Plataformas de intercambio de datos y descubrimiento de fuentes de datos.
- Acuerdos entre empresas poseedoras de datos e interesadas en explotarlo.

Una de las áreas de investigación en inteligencia artificial que más prometedora parece es la visión computacional, pues cuenta con aplicaciones tan llamativas como:

- Detección de enfermedades.
- Sistemas de realidad aumentada.
- Vehículos autónomos.

- Ayuda a personas con visibilidad reducida.
- Indexación de imágenes en base a su contenido.

La visión computacional sigue dos aproximaciones: la biológica, que persigue la obtención de modelos computacionales del sistema de visión humano, cuyo funcionamiento es aún objeto de múltiples investigaciones; y la tecnológica, que aborda la construcción de sistemas capaces de realizar funciones propias del sistema de visión humano, que será la que aborde en este trabajo.

Debido a la variedad de aplicaciones, las principales empresas del sector están dedicando grandes cantidades de recursos a la mejora de sus sistemas de visión computacional y resulta útil comprender el funcionamiento de las principales soluciones de etiquetado de imagen y comparar su eficacia.

1.2. Objetivos

El objetivo principal de este trabajo es comprender el funcionamiento y evaluar y comparar la eficacia de distintas soluciones de etiquetado de imagen. Para conseguirlo, se plantean los siguientes subobjetivos:

1. Estudio de los fundamentos aplicables a la cuestión.
2. Estudio y selección de soluciones de etiquetado.
3. Estudio y selección de herramientas de análisis semántico.
4. Obtención de los datos necesarios para realizar la comparativa, lo que supone varias tareas:
    - Selección, obtención y preparación del conjunto de imágenes a etiquetar.
    - Etiquetado de dicho conjunto de imágenes.
5. Análisis de las etiquetas obtenidas y obtención de resultados.
6. Análisis de los resultados obtenidos.

1.3. Estado de la cuestión

Desde que se realizase la primera fotografía en 1826, la forma de obtención, almacenamiento y procesamiento de las imágenes ha evolucionado notablemente; siendo en 1957 cuando se digitaliza por primera vez una imagen, haciendo viables las actuales técnicas de visión artificial.

En 1963, Lawrence Roberts [1] publica una tesis en torno a la percepción de objetos sólidos tridimensionales en imágenes que es ampliamente aceptada como la precursora de la investigación en visión computacional. Como se observa en la Figura 1.1, Roberts

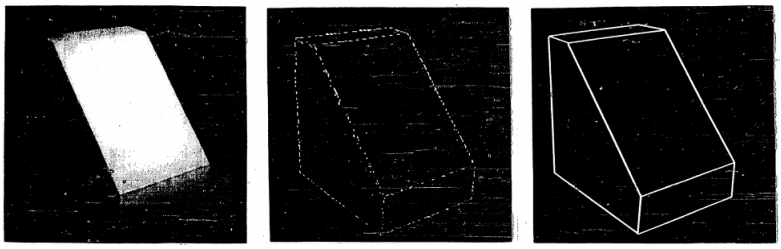


Figura 1.1: Figuras tomadas de [1]

implementa una heurística que analizaba los pixeles de una imagen en busca de cambios de color, con lo que reconocía bordes y, por lo tanto, cuerpos tridimensionales sencillos.

En 1966, se propuso como proyecto de verano un sistema de «reconocimiento de patrones» para la detección de objetos no solapados [2], cuyo desarrollo requirió bastante más tiempo y es ampliamente considerado como el nacimiento de la visión computacional como disciplina científica.

Entonces, el reconocimiento se realizaba mediante algoritmos codificados manualmente a partir de cambios de color, bordes y esquinas, lo que complicaba el reconocimiento en imágenes con baja claridad o variaciones, por lo que la investigación se centraba en la mejora y desarrollo de dichos algoritmos [3].

Durante los años sesenta y setenta, David Hubel y Torsten Wiesel [4] realizaron una investigación que demostraba que el sistema de percepción visual de un gato estaba compuesto por neuronas simples pero especializadas. Mientras que algunas neuronas se activaban al percibirse formas sencillas -como líneas, bordes o manchas-, otras se activaban ante ciertos patrones de activación de las anteriores.

Dicha investigación sirvió de base para que David Marr, a finales de los años setenta, propusiese un paradigma de trabajo *bottom-up* basado en una jerarquía de características [5]. Las más sencillas eran bordes, curvas o esquinas, que combinadas entre sí generaban características más complejas, que combinadas entre sí generaban características aún más complejas, y así sucesivamente hasta reconocer el contenido de la imagen. Dicho paradigma sigue vigente actualmente.

A partir de ese momento se empiezan a desarrollar modelos de redes neuronales profundas, en las que la propia red extrae la información que emplea posteriormente para la clasificación del contenido de la imagen, en vez de codificarse manualmente dichas características.

En 1980, se finaliza el desarrollo de Neocognitron[6], una red neuronal robusta a variaciones de posición de los objetos. Dos años después, en 1982, se aplicó el algoritmo *backpropagation* al entrenamiento de dicha red, dando lugar a LeNet-5[7], considerada

como la primera red neuronal convolutiva -el tipo de red neuronal más empleado en visión artificial- moderna.

Desde entonces, las arquitecturas de la redes neuronales empleadas han sufrido una evolución continua y, con ellas, su eficacia, por lo que ciertos *datasets* como *Pascal VOC* o *ImageNet* se han convertido en un estándar *de facto* para evaluar su eficacia, como se observa en la Figura 1.2

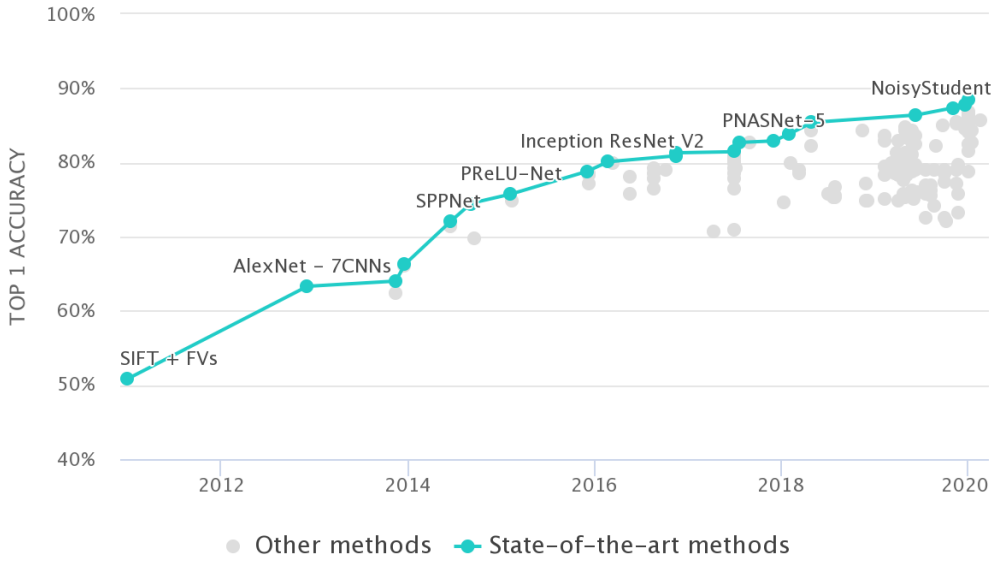


Figura 1.2: Evolución del porcentaje de aciertos (*accuracy*) en clasificación de imágenes sobre ImageNet. Tomado de [8].

En paralelo con los últimos avances en visión computacional, también se ha avanzado en etiquetado de imagen, lo que resulta más complejo, pues requiere una comprensión completa de la imagen, desde los objetos presentes y sus atributos hasta las relaciones entre estos o incluso inferir objetos que no aparecen en ella[9].

Las primeras soluciones basadas en aprendizaje profundo, capaces de generar descripciones originales en lenguaje natural, recurrieron a una estructura *encoder-decoder*: una red neuronal convolucional extrae las características de la imagen, que sirven de entrada para una red neuronal recurrente que genera la descripción en lenguaje natural de la imagen [10][11].

Posteriormente, se propusieron diversas mejoras a esta arquitectura como: alimentar la red recurrente con características de más bajo nivel [12], emplear información semántica contenida en la imagen como una entrada adicional de la red recurrente [13] o emplear varios *encoders* [14]. Más recientemente, se han desarrollado sistemas de etiquetado de imagen implementados completamente mediante redes convolucionales [15][16].

Actualmente, la investigación académica continúa en varias direcciones: desde el

reconocimiento de objetos que no están presentes en los datos empleados para el entrenamiento [17] o el entrenamiento sin emplear imágenes etiquetadas [18], hasta la mejora de la interpretabilidad de las redes neuronales [19].

1.4. Organización de la memoria

El presente trabajo está compuesto por 9 capítulos:

1. Introducción: en esta se motiva el tema que se tratará durante el trabajo, se exponen los objetivos a alcanzar, se describe el estado actual de la cuestión y se explica la organización de la memoria.
2. Estudio de los fundamentos aplicables al experimento a realizar.
3. Estudio y selección de las soluciones de etiquetado de imagen a evaluar.
4. Estudio de herramientas de análisis semántico, necesario para el análisis de etiquetas.
5. Selección, obtención y análisis del conjunto de imágenes empleado.
6. Etiquetado de dicho conjunto de imágenes.
7. Análisis de resultados.
8. Conclusiones obtenidas y trabajo futuro.
9. Bibliografía empleada.

Finalmente se recogen los apéndices cuya consulta pueda resultar necesaria:

- Planificación.
- Tecnologías empleadas para la realización del trabajo.
- Detalles sobre la implementación del experimento.
- Procedimiento a seguir para la reproducción del experimento.
- Conjuntos de datos estudiados.
- Estudio semántico del conjunto de datos empleado.

Capítulo 2

Fundamentos teóricos

El objetivo de este capítulo es presentar los fundamentos teóricos necesarios para comprender el resto de la memoria. Abarca tanto etiquetado de imagen como el análisis semántico de palabras, que, como más adelante se verá, resulta necesario para realizar el experimento comparativo.

2.1. Etiquetado de imagen

El etiquetado de imagen consiste en la generación de etiquetas que describan el contenido de imágenes digitales.

2.1.1. Imagen digital

Las imágenes pueden representarse digitalmente de dos formas:

- Imagen vectorial: conjunto de puntos bidimensionales relacionados mediante líneas con distintas propiedades como pueden ser su color, forma, curvatura o ancho. Esto causa que el tamaño y la relación de aspecto de estas imágenes se pueda modificar sin que su calidad se vea afectada.

- Imagen de mapa de bits: matriz de numerosos elementos -pixeles-, cada uno de los cuales está asociado a un color. El color de cada píxel se codifica como la combinación de unos determinados colores primarios, mediante un número determinado de bits -profundidad de color- para cada color primario. Habitualmente se emplea el modelo RGB (*Red Green Blue*) con una profundidad de color de 8 bits; de esta forma, una imagen de 320 pixeles de ancho y 240 pixeles de alto en RGB será codificada mediante una matriz de 320x240x3x8 *bits*. Este es el método de representación empleado en las técnicas explicadas a continuación.

2.1.2. Evolución

Aunque actualmente las redes neuronales convolucionales sean un estándar *de facto* en etiquetado de imagen, conviene estudiar la evolución de los sistemas de reconocimiento de imagen para comprender dicha dominancia.

Reconocimiento de patrones

Las primeras soluciones al problema del reconocimiento de imagen empleaban heurísticas de detección y clasificación de patrones [1][2], cuya eficacia era muy limitada.

Aprendizaje automático tradicional

El siguiente paso fue lo que se conoce como aprendizaje automático tradicional, un proceso formado por distintas fases [3]:

1. Clasificación de imágenes: se agrupan manualmente un cierto número de imágenes según su contenido. Aquí ya se definen dos conceptos clave en aprendizaje automático: se conoce como clase cada una de las distintas entidades que un sistema de reconocimiento es capaz de reconocer y como *ground truth* la información que se asume como válida. Por ejemplo, en etiquetado de imagen el *ground truth* de una imagen de un avión sería «avión», mientras que la clase de dicha imagen sería la etiqueta predicha por sistema.

2. Codificación de características: una característica puede verse como un «filtro» con el que se recorre toda la imagen para comprobar qué secciones de la imagen presentan una determinada peculiaridad. En la práctica, dado que las imágenes se codifican como matrices de pixeles, estos filtros consisten en una matriz que es multiplicada por múltiples submatrices de la imagen, cuantificando así la similaridad de cada sección de la imagen con los cambio de color, bordes o esquinas codificados.

3. Extracción de características: se obtienen todas las características presentes en las imágenes que se clasificaron manualmente, que sirven como definición de las distintas clases. Más adelante, esta fase recibió mejoras como agilizarla o hacer que incorporase relaciones espaciales entre las características detectadas [20].

4. Clasificación de imagen: consistente en la extracción de las características presentes en la imagen a clasificar y su comparación con las presentes en la definición de las distintas clases.

En un principio, la técnica empleada para realizar la clasificación era la de *bag-of-words*, que simplemente analizaba si una imagen contenía las características presentes en la definición de cada clase. Para mejorar la precisión, cada clase requería de un ajuste manual de la importancia de cada una de sus características, lo que suponía un pesado proceso de prueba y error guiado por la intuición del desarrollador y se volvía muy complejo con un número elevado de clases.

Más adelante se implantaron clasificadores neuronales, que debían ser entrenados proporcionándoles las características presentes en múltiples imágenes y la salida esperada (la clase de dichas imágenes), para que luego aplicase su «conocimiento» en otras imágenes. En la Sección 2.1.3 abordo con más detalle las redes neuronales.

Aprendizaje automático profundo

El último gran avance fue lo que se conoce como aprendizaje profundo, en el que tanto la extracción de características como la clasificación las realiza la misma red neuronal, como se observa en la Figura 2.1

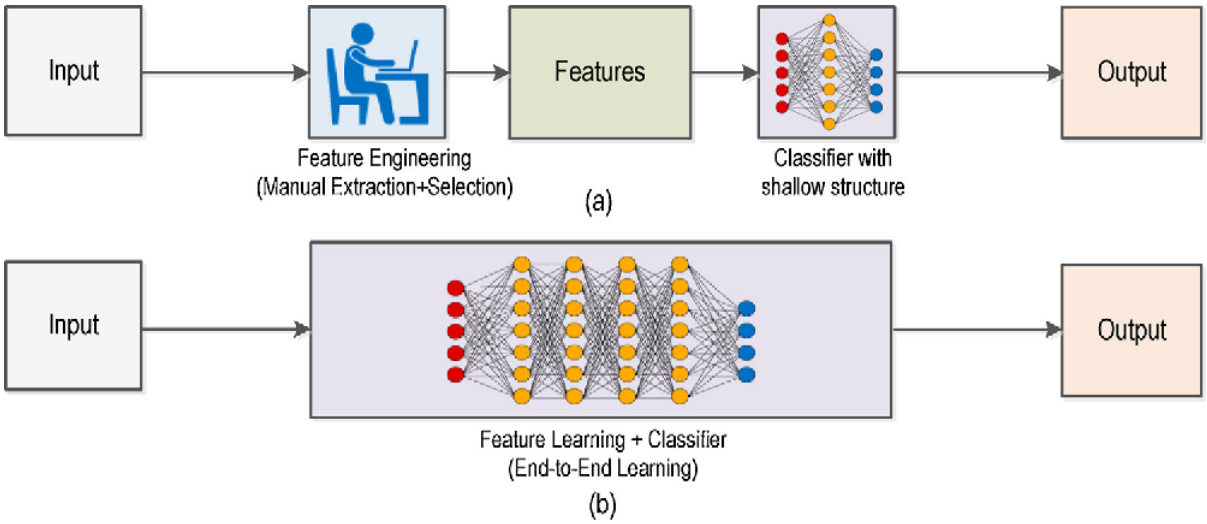


Figura 2.1: Diferencia entre aprendizaje automático tradicional(a) y profundo(b). Tomado de [21].

Dado que la red neuronal realiza tanto la extracción de características como la clasificación, es entrenada directamente con un conjunto de imágenes clasificadas -*dataset*-, en cuyas imágenes la propia red descubre las características presentes, y la importancia de cada una de estas para cada clase, «aprendiendo» así a clasificar imágenes.

Este enfoque, aparte de eliminar la complejidad de tener que codificar y ponderar manualmente las características, también resulta más robusto a la variabilidad de las imágenes a clasificar (debido a transformaciones, deformaciones o ruido), a cambio de necesitar muchos más datos y una gran potencia de cálculo para entrenarse.

Mientras que el aprendizaje automático tradicional permite clasificar imágenes cuando se cuenta con pocos datos (debido a la intervención humana), cuando aumenta la cantidad de dato o se producen variaciones en este, su eficacia se vuelve claramente inferior a la del aprendizaje profundo [21][3], como se observa en la Figura 2.2.

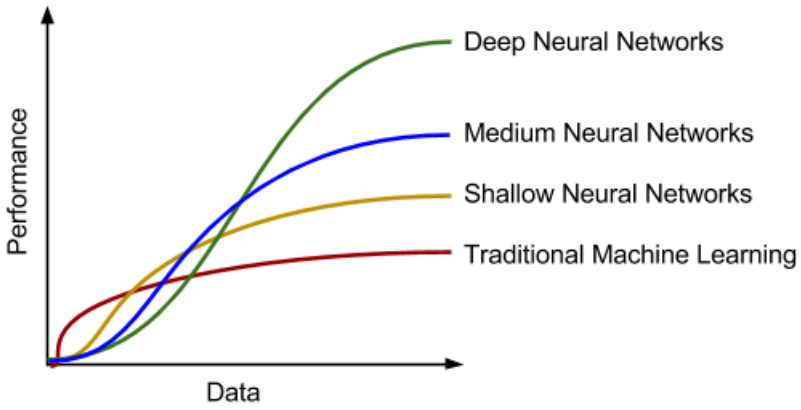


Figura 2.2: Eficiencia de las distintas técnicas. Tomado de [22].

2.1.3. Redes neuronales artificiales

Las redes neuronales artificiales son un modelo computacional inspirado en su homólogo biológico, compuesto por un gran número de neuronas que realizan operaciones sencillas y se comunican entre sí.

Cada neurona toma la salida de otras neuronas, las multiplica por el *peso* asociado a cada conexión neuronal, calcula su suma y le aplica a esta una función no lineal para obtener su valor de salida, como se observa en la Figura 2.3.

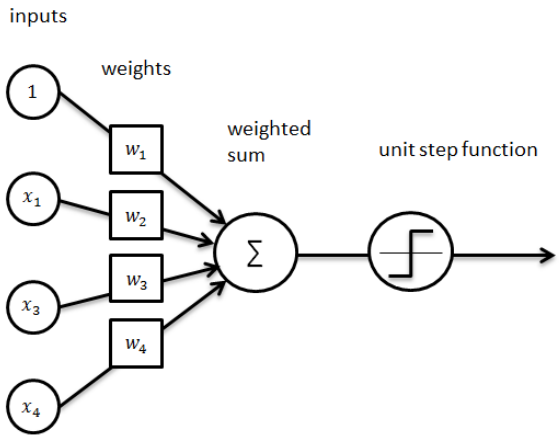


Figura 2.3: Funcionamiento básico de una neurona. Tomado de [23].

Estructura

Las redes neuronales cuentan con una estructura compuesta por tres partes:

- Capa de entrada: recibe el dato a procesar en el resto de la red.
- Capas intermedias: realizan el procesamiento del dato. Se considera que una red neuronal es profunda si tiene dos o más capas intermedias, también conocidas como ocultas.
- Capa de salida: generan el resultado final de la red.

En la Figura 2.4 se observa la arquitectura de una red neuronal profunda básica.

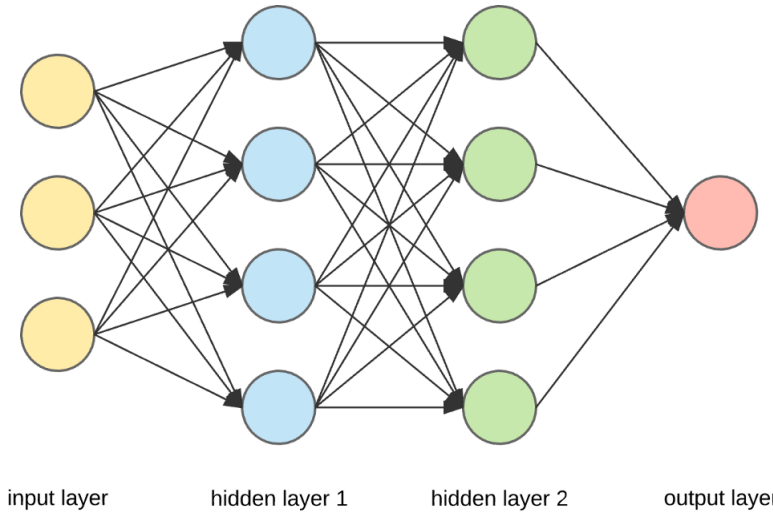


Figura 2.4: Arquitectura de una red neuronal profunda básica. Tomado de [23].

Entrenamiento

El entrenamiento de una red neuronal consiste en minimizar la diferencia entre los resultados deseados para unas entradas determinadas -*ground truth*- y los resultados generados por la red para dichas entradas.

Esto se consigue alimentando la red neuronal con un mismo conjunto de datos repetidas veces -*epochs*- y ajustando los pesos de las conexiones neuronales de la red. Generalmente, se emplea el algoritmo *backpropagation*: para cada entrada, se compara la salida de la red con la salida esperada, obteniendo así una señal de error. Esta señal de error se propaga hacia atrás, a todas las capas neuronales y cada una de sus neuronas, para las que se calcula su aportación al total del error. Posteriormente, se ajusta cada peso en función de dicha aportación, especializándose así cada neurona en la detección de una característica concreta del espacio de entrada.

Una vez que los pesos apenas varían para las distintas entradas -han convergido-, se registran estos y la asociación entre la activación de las neuronas de la capa de salida y las clases presentes en el dato de entrada, generando lo que se conoce como *modelo*, que es el artefacto empleado para realizar inferencias.

En el caso de aprendizaje supervisado, el empleado en etiquetado de imagen, se emplea un conjunto de datos generados manualmente -*dataset*-, el cual es muy importante que tenga un tamaño suficiente y un *ground truth* fiable. Para cuantificar de forma fiable la eficacia de los modelos generados se divide -*split*- el conjunto de datos empleado en varios subconjuntos disjuntos:

- *Training set*, empleado para realizar el entrenamiento. Es el más grande, en torno a un 80 % del dato.
- *Validation set*. Si el *training set* presenta ciertas particularidades, el modelo se sobreajustará a estas particularidades perdiendo eficacia en los casos no presentes en dicho conjunto. Para evitar esto, una vez entrenado el modelo se ajustan ciertos hiperparámetros empleando este conjunto.
- *Testing set*, empleado para cuantificar la eficacia del modelo entrenado una vez sus hiperparámetros ya han sido ajustados.

Resulta muy importante mantener estos conjuntos disjuntos durante todo el ciclo de vida del modelo, para asegurar que la evaluación de su eficacia sea correcta.

Tipos

Existen multitud de tipos de redes neuronales, pero generalmente se pueden dividir en dos tipos:

- *Recurrent Neural Network (RNN)*
  - Entrada y salida como secuencias de longitud arbitraria.
  - Presentan ciclos internos que provocan «memoria» a corto plazo en la red, pues lo que pasa por la red tendrá efecto en el procesamiento posterior del resto de la secuencia.
  - Adecuadas para el análisis de texto y voz.
- *Feedforward Neural Network (FNN)*
  - Entrada y salida de un tamaño fijo.
  - Carecen de ciclos, la salida de una neurona sólo se emplea en neuronas pertenecientes a una capa posterior.

- Adecuadas para el procesamiento de imagen.

Concretamente, para etiquetado de imagen se emplean redes neuronales convolucionales, un subtipo de estas últimas.

2.1.4. Redes neuronales convolucionales

Desde que se empezaron a aplicar redes neuronales profundas para el reconocimiento de imagen, se fue avanzando en las arquitecturas de estas, hasta llegar a las redes neuronales convolucionales, que cuentan con una serie de ventajas respecto a otras redes neuronales profundas:

- Detectan las características en la imagen independientemente de su ubicación.
- Eficiencia del entrenamiento, debido a:
  - Pesos compartidos.
  - Bajo número de conexiones neuronales al no estar cada neurona completamente conectada con la capa anterior, como se observa en la Figura 2.5.

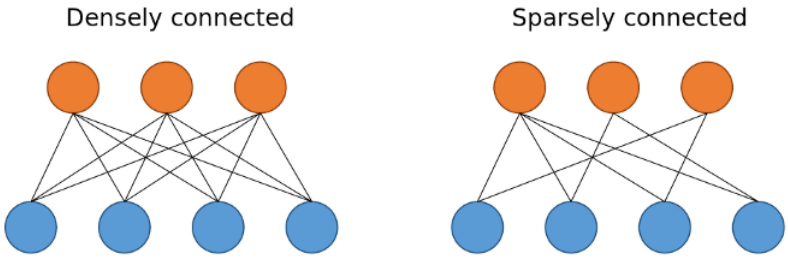


Figura 2.5: Densidad de las conexiones interneuronales. Tomado de [24].

  Esto permite agilizar los entrenamientos y entrenar arquitecturas más profundas, lo que permite la detección de características más complejas.

- Robustez a variaciones espaciales.

Funcionamiento

Estas toman su nombre de las convoluciones que realizan algunas de sus neuronas, pues recorren la imagen tomando grupos de pixeles cercanos y operando sobre estos. Como se observa en la Figura 2.6, están formadas por múltiples capas neuronales con distintos cometidos:

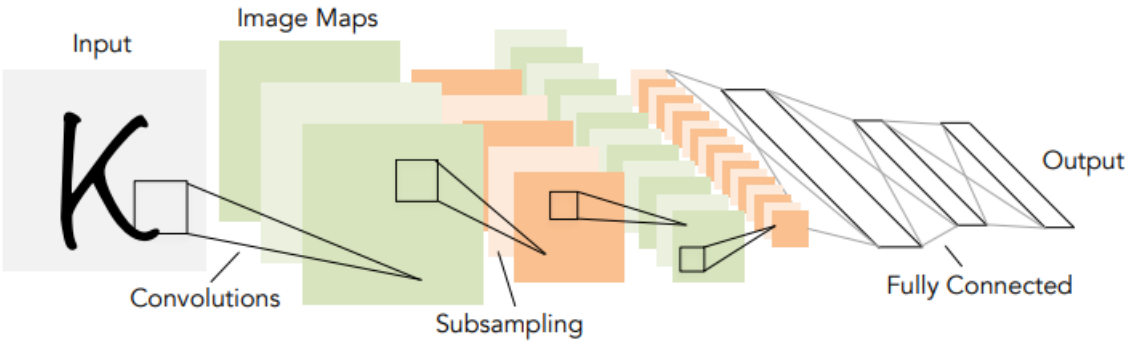


Figura 2.6: Estructura básica de una *CNN*. Tomado de [25].

1. Extracción de características:

   - Capas convolucionales: cada neurona recorre la imagen tomando submatrices de un tamaño y solapamiento determinados, como se observa en la Figura 2.7, y calculando su producto escalar con una matriz llamada *kernel*.

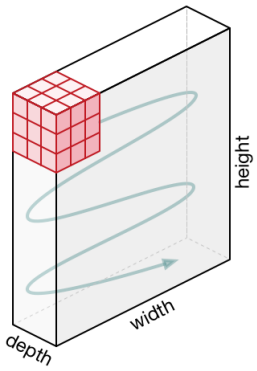


Figura 2.7: Recorrido del *kernel*. Tomado de [26].

Tras la convolución, es decir, haber recorrido la totalidad de la imagen y haber calculado todos los productos, se obtiene un mapa de características, como se observa en la Figura 2.8.

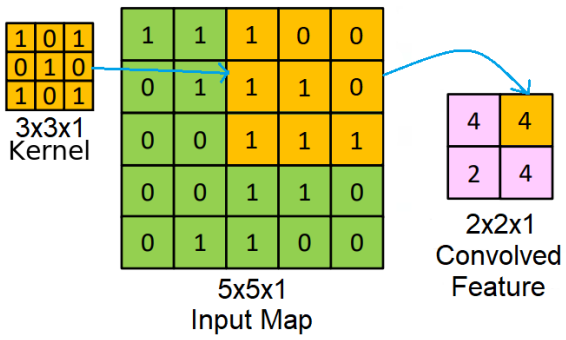


Figura 2.8: Ejemplo de una convolución. Tomado de [27].

Cada *kernel* detecta una característica determinada; por ejemplo, en el caso de la Figura 2.9, detecta bordes.

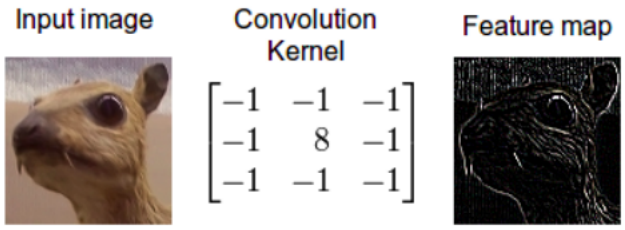


Figura 2.9: Ejemplo de *kernel* y *feature map* resultado. Tomado de [28].

Cada capa opera con la salida de la capa anterior; por lo que según se profundiza las capas se vuelven menos sensibles a la variación de los datos de entrada y detectan características más complejas.

- Capas de submuestreo: reducen la dimensión de los mapas de características que las capas anteriores han generado. Normalmente se intercalan con varias capas convolucionales y no es necesario entrenarlas; por ejemplo, en la Figura 2.10 se selecciona el valor máximo.

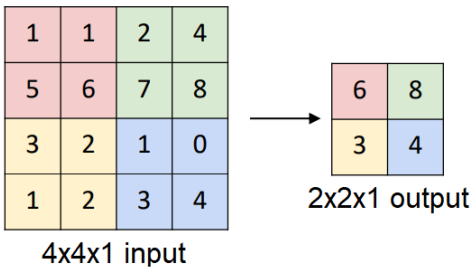


Figura 2.10: Ejemplo de submuestreo por valor máximo. Tomado de [27].

2. Clasificación en función de las características extraidas:
   - Clasificador totalmente conectado, que combina todas las características extraidas de la imagen.
   - Capa final. Para cada clase, esta capa cuenta con una neurona que se activa en mayor o menor grado según lo acorde que sean las características detectadas con dicha clase. La valoración de esta concordancia se denomina confianza o *score*.

2.2. Análisis semántico de palabras

Como se verá más adelante, para el cometido de este trabajo resulta necesario evaluar la similitud semántica de distintas palabras; lo que se puede realizar de distintas formas:

- **Valoración de la similitud ortográfica**. No tiene en cuenta la semántica de las palabras; por lo que dos palabras con semánticas completamente distintas pueden tener una ortografía muy similar y considerarse equivalentes, al igual que dos palabras con semánticas idénticas y distinta ortografía no se considerarían equivalentes.

- **Empleo de una base de datos léxica**. Estas permiten tanto conocer los distintos significados que puede tomar una misma palabra, como consultar relaciones jerárquicas de hiperónimos e hipónimos (tesauro), como se observa en la Figura 2.11.

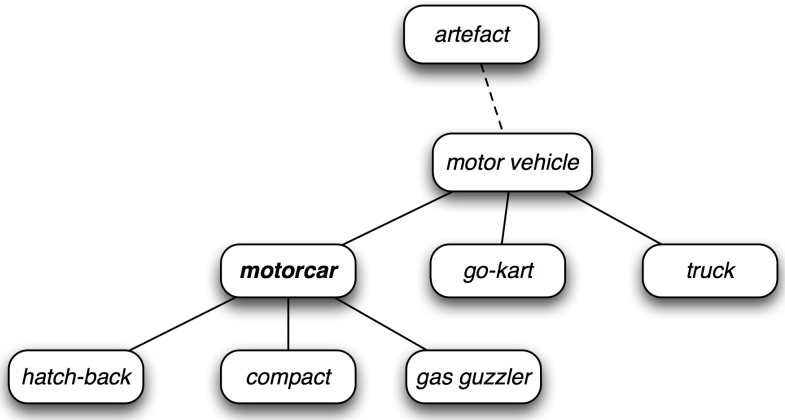


Figura 2.11: Ejemplo de tesauro. Tomado de [29].

- **Modelado vectorial de palabras** (*word embedding*): las palabras se representan en un espacio vectorial multidimensional en base a su contexto; es decir, las palabras que aparezcan escritas juntas frecuentemente aparecerán próximas en dicho espacio vectorial.

  Teniendo en cuenta que el significado de una palabra se caracteriza mediante el contexto de dicha palabra, para cuantificar la sinonimia de dos palabras, basta con calcular la distancia entre sus representaciones vectoriales.

  Este enfoque no soporta bien la polisemia, pues asume que cada palabra tiene un único significado, por lo que han surgido distintas técnicas para solventar este inconveniente, como emplear también información procedente de bases de datos léxicas [30].

Capítulo 3

Soluciones de etiquetado

Para realizar el experimento comparativo resulta imprescindible emplear distintas soluciones de etiquetado. Estas soluciones deben recibir una imagen y generar un conjunto de pares formados por una etiqueta y una confianza, en el que la primera describa el contenido de dicha imagen y la segunda cuantifique la confianza del sistema en que dicha etiqueta sea la correcta para la imagen procesada.

3.1. Estudio de alternativas

Como soluciones de etiquetado aplicables al experimento comparativo, se dispone de varias alternativas:

- **Generar modelos propios**: para ello, no tengo que implementar mi propia red, pues existen múltiples implementaciones públicas de redes neuronales en distintas bibliotecas de aprendizaje automático, como `TensorFlow`, `PyTorch` o `Keras`.

  Pese a que de esta manera podría evaluar soluciones muy cercanas al estado del arte, requiere de unos conocimientos que quedan fuera del alcance de este trabajo y de una gran potencia de cálculo.

- **Emplear modelos pre-entrenados**: al igual que con las implementaciones de redes neuronales, tambien existen modelos previamente entrenados disponibles públicamente.

  Pero dichos modelos no son específicos para etiquetado de imagen, por lo que habría que aplicar *Transfer Learning* y modificar una red de reconocimiento de objetos o clasificación de imagen para que realizase etiquetado, lo que también queda fuera del alcance de este trabajo.

- **Emplear soluciones comerciales**: múltiples empresas ofrecen, junto a otras soluciones de reconocimiento de imagen, `APIs` *cloud* de etiquetado de imagen.

Entonces; realizaré la comparativa empleando `APIs` *cloud*, que son, con diferencia, la alternativa más sencilla de implantación en el experimento y más próxima a la realidad.

3.2. Selección

Para realizar la selección, me baso en aquellas presentes en artículos científicos [31][32][33], parte de las cuales ya no están disponibles, o solamente detectan objetos, o no disponen de planes de prueba gratuitos.

Dichas soluciones de etiquetado forman parte de plataformas de reconocimiento de imagen, pero la información disponible acerca de su implementación resulta casi nula, como también se constata en [32]. Las empresas las tratan como soluciones de gran precisión sin entrar en detalles más allá de que usan redes neuronales profundas [34][35], convolucionales [36][37] o que entrenan sus modelos de forma periódica añadiendo nuevas etiquetas.

Figura 3.1: Importancia de las plataformas *cloud* de reconocimiento de imagen en el último cuarto del año 2019, según Forester. Tomado de [38].

A falta de información relevante sobre cómo obtienen el dato empleado para el entrenamiento, sobre la gestión del entrenamiento de los modelos y sobre el conjunto de etiquetas empleado; selecciono las cinco soluciones desarrolladas por empresas relevantes en el sector de computación en la nube o especializadas en reconocimiento de imagen. Esta selección coincide con la efectuada en los artículos consultados [31][32][33] y es acorde con la importancia de las plataformas *cloud* de reconocimiento de imagen en el ultimo cuarto del año 2019, como se observa en la Figura 3.1

A continuación presento las características básicas de las cinco plataformas de reconocimiento de imagen escogidas para constatar sus funcionalidades y madurez y comprobar que puedo emplearlas de forma gratuita.

3.2.1. Amazon Web Services Rekognition

Rekognition es el servicio de reconocimiento visual de la plataforma de computación en la nube de Amazon Web Services. Este fue anunciado a finales del año 2016, implementando solamente detección de objetos, una pequeña parte de lo que ofrece actualmente [39], aplicable tanto en imágenes como en vídeos:

- Reconocimiento de objetos y escenas.
- Etiquetado de imágenes con un modelo personalizado (SageMaker).
- Detección de contenido explícito o adulto.
- Detección de rostros.
- Reconocimiento facial.
- Reconocimiento de personas famosas.
- Análisis facial .
- Reconocimiento de texto.

AWS ofrece un modo de prueba de Rekognition, que permite analizar 5.000 imágenes de forma gratuita durante 12 meses.

3.2.2. Clarifai

Fundada en 2013, esta empresa está especializada en reconocimiento visual, más concretamente en clasificación de imágenes y vídeos. Estas son las funcionalidades que implementa [40]:

- Reconocimiento de objetos y escenas.

- Etiquetado de imágenes con un modelo personalizado.
- Detección de contenido específico de un dominio.

Clarifai ofrece un modo de prueba de duración ilimitada que permite analizar hasta 5.000 imágenes al mes de forma gratuita.

3.2.3. Microsoft Azure Computer Vision

Computer Vision agrupa los servicios de reconocimiento visual desarrollados por Microsoft para su plataforma Azure. Está fue lanzada al mercado en marzo del 2018 y actualmente es la plataforma más completa, soportando las siguientes operaciones [41]:

- Reconocimiento de objetos y escenas.
- Etiquetado de imágenes con un modelo personalizado (Custom Vision).
- Descripción en lenguaje natural.
- Detección de contenido explícito o adulto.
- Detección de rostros.
- Reconocimiento facial.
- Reconocimiento de personas famosas.
- Análisis facial.
- Reconocimiento de texto.
- Reconocimiento de escritura manual.
- Reconocimiento de formularios manuscritos.
- Reconocimiento de logotipos de marcas comerciales.
- Detección de contenido específico de un dominio.

Microsoft ofrece un modo de prueba gratuito de duración ilimitada que permite analizar un máximo de 5.000 imágenes al mes.

3.2.4. Google Cloud Vision

Cloud Vision es el servicio de reconocimiento visual ofrecido por Google en su plataforma Cloud desde mayo del 2017. Implementa las siguientes funciones [42]:

- Reconocimiento de objetos y escenas.
- Etiquetado de imágenes con un modelo personalizado.
- Detección de contenido explícito o adulto.
- Detección de rostros.
- Reconocimiento de texto.
- Reconocimiento de escritura manual.
- Detección de logotipos de marcas comerciales.
- Reconocimiento de elementos geográficos famosos.

Google ofrece un modo de prueba de duración ilimitada que permite analizar un máximo de 1.000 imágenes al mes de forma gratuita, pero también ofrece un crédito inicial de 300$, con lo que puedo emplear dicho crédito.

3.2.5. IBM Watson Visual Recognition

Visual Recognition es el servicio de reconocimiento visual presente en la plataforma IBM Watson. Fue lanzado al mercado en mayo del 2016 y actualmente implementa las siguientes funcionalidades [43]:

- Reconocimiento de objetos y escenas.
- Etiquetado de imágenes con un modelo personalizado.
- Detección de rostros.
- Análisis facial.

IBM ofrece un modo de prueba gratuito de duración ilimitada pero limitado a 1.000 análisis de imágenes al mes, pero también ofrece un crédito inicial de 200$, con lo que puedo emplear dicho crédito.

3.2.6. Comparativa

Tratando de resumir las características de las plataformas *cloud* de reconocimiento de imagen escogidas, recojo estas en la Tabla 3.1

	Amazon	Microsoft	Google	IBM	Clarifai
Reconocimiento de objetos	✓	✓	✓	✓	✓
Reconocimiento de escenas	✓	✓	✓	✓	✓
Etiquetado con modelo personalizado	✓	✓	✓	✓	✓
Descripción en lenguaje natural		✓			
Detección de contenido inadecuado	✓	✓	✓		
Detección de rostros	✓	✓	✓	✓	
Reconocimiento facial	✓	✓			
Reconocimiento de personas famosas	✓	✓			
Análisis facial	✓	✓		✓	
Reconocimiento de texto	✓	✓	✓		
Reconocimiento de escritura manual		✓	✓		
Reconocimiento de formularios		✓			
Reconocimiento de marcas comerciales		✓	✓		
Reconocimiento de elementos geográficos			✓		
Detección de contenido específico		✓			✓
Salida al mercado	2016	2018	2017	2016	2013

Tabla 3.1: Comparativa de las plataformas escogidas

En el Apéndice C recojo lo referido a la implementación de los cinco etiquetadores escogidos.

Capítulo 4

Análisis semántico

Debido a la falta de información sobre el conjunto de etiquetas empleado por cada solución de etiquetado se hace necesario desarrollar una herramienta que permita armonizar y comparar las etiquetas generadas por las distintas soluciones de etiquetado para una misma imagen y el *ground truth*. En este capítulo, estudio las herramientas aplicables y la aplicación de una de ellas.

4.1. Estudio de alternativas

Como expuse en la Sección 2.2, existen distintas formas de valorar la similitud semántica entre distintas palabras. Para la realización del experimento comparativo optaré por emplear una base de datos léxica o *word embeddings* pre-entrenados.

Debido a que desconozco como soportan los *word embeddings* disponibles públicamente la polisemia de las palabras y que valorarlo requeriría de un estudio adicional fuera del alcance del trabajo, decido emplear una base de datos léxica, que resulta suficiente para el cometido del trabajo.

Concretamente emplearé `WordNet`[44], una base de datos léxica de gran aceptación fácilmente utilizable a través la librería de `Python` *Natural Language Toolkit*[29] (`NLTK`). Esta también cuenta con una gran popularidad y proporciona múltiples métricas de similaridad entre palabras, que incluso incorporan información de distintos *text corpus*.

4.2. Natural Language Toolkit (NLTK)

El funcionamiento de esta librería se basa en *synsets*, que son conjuntos de sinónimos, a partir de los cuales podemos obtener sus hiperónimos e hipónimos y entre los cuales podemos cuantificar su similaridad.

Cada *synset* de una misma palabra representa un significado distinto de esa palabra; como ejemplo, en la Figura 4.1 recojo los *synsets* de la palabra *dog*.

```
>>> wn.synsets("dog")
    [Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'),
       Synset('cad.n.01'), Synset('frank.n.02'),
       Synset('pawl.n.01'), Synset('andiron.n.01'),
       Synset('chase.v.01')]
```

Figura 4.1: *Synsets* de la palabra *dog*

Se observa como *dog* puede significar desde el animal, hasta perrito caliente o hasta perseguir. Cada *synset* se identifica por su palabra más representativa, su categoría gramatical, y un identificador numérico; de esta forma, `Synset('dog.n.01')` se refiere al animal, que es el primer significado sustantivo de *dog*[45].

Cada uno de las palabras asociadas a un mismo *synset* se denomina *lemma*; a modo de ejemplo, en la Figura 4.2 recojo los *lemmas* de `Synset('dog.n.01')`.

```
>>> wn.synset("dog.n.1").lemmas()
    [Lemma('dog.n.01.dog'), Lemma('dog.n.01.domestic_dog'),
       Lemma('dog.n.01.Canis_familiaris')]
```

Figura 4.2: *Lemmas* de `Synset('dog.n.01')`

4.2.1. Métricas de similaridad semántica

Puesto que en vez de trabajar con las etiquetas directamente, trabajaré con los synsets asociados a estas, necesito escoger una métrica adecuada para cuantificar la similaridad entre estos.

`NLTK` ofrece distintas métricas de similaridad [46], las cuales se pueden clasificar en dos grupos:

- Métricas que solamente emplean información procedente de la jerarquía del tesauro (*Structure-based measures*). A este grupo pertenecen *Path Distance Similarity*, *Leacock-Chodorow Similarity* y *Wu-Palmer Similarity*.

- Métricas que combinan información del tesauro con información semántica (*Information Content Measures*). Esta información semántica proviene de un *text corpus*[47], por lo que los resultados de estas métricas varían en función del *corpus* empleado. En esta categoría están encuadrados *Resnik Similarity*, *Jiang-Conrath Similarity* y *Lin Similarity*.

A continuación presento las características de cada métrica [45] y sus resultados para un ejemplo tomado de [29]. Como el segundo segundo grupo de métricas necesita de

un *text corpus*; de entre los proporcionados por `NLTK` [48], emplearé *Brown Corpus*, un corpus generado por la Universidad de Brown con 500 textos en inglés procedentes de noticias de los Estados Unidos publicadas en el año 1961.

Path Distance Similarity

Analiza la similaridad entre dos *synsets* basándose únicamente en la longitud del camino más corto que los une en la estructura jerárquica de hiperónimos de `WordNet` a través de su antecesor -hiperónimo- común más específico (más profundo).

```
>>> seahorse = wn.synset('seahorse.n.01')
>>> shark = wn.synset('shark.n.01')
>>> tiger = wn.synset('tiger.n.01')
>>> plant = wn.synset('plant.n.01')
>>> book = wn.synset('book.n.01')
>>>
>>> wn.path_similarity(seahorse, seahorse)
1.0
>>> wn.path_similarity(seahorse, shark)
0.09090909090909091
>>> wn.path_similarity(seahorse, tiger)
0.09090909090909091
>>> wn.path_similarity(seahorse, plant)
0.06666666666666667
>>> wn.path_similarity(seahorse, book)
0.058823529411764705
```

Figura 4.3: Similaridad de *seahorse* con distintos *synsets* empleando *Path Similarity*

Como se observa en la Figura 4.3, la similaridad con caballito de mar disminuye según nos movemos del espacio semántico de animales, al de plantas y al de inanimados pero no varía entre los espacios semánticos de animales terrestres y acuáticos, pues presumiblemente ambos tendrán como hiperónimo común más próximo a animal. También se observa que sus resultados son poco lineales, pues la similaridad con caballito de mar disminuye de manera drástica de caballito de mar a tiburón pero decrece mucho menos de tiburón a libro.

Leacock-Chodorow Similarity

Extiende *Path Distance Dimilarity* normalizándola al dividirla por la máxima profundidad en la jerarquía de ambos conceptos:

$$Sim_{\mathrm{LC}}(S1, S2) = -log\frac{L}{2D} \tag{4.1}$$

Siendo $S1$ y $S2$ los *synsets* a comparar, L la longitud del camino más corto que los une y D la profundidad máxima de estos.

```
>>> wn.lch_similarity(seahorse, seahorse)
3.6375861597263857
>>> wn.lch_similarity(seahorse, shark)
1.2396908869280152
>>> wn.lch_similarity(seahorse, tiger)
1.2396908869280152
>>> wn.lch_similarity(seahorse, plant)
0.9295359586241757
>>> wn.lch_similarity(seahorse, book)
0.8043728156701697
```

Figura 4.4: Similaridad de *seahorse* con distintos *synsets* empleando *Leacock-Chodorow Similarity*

Como se observa en la Figura 4.4; la similaridad con caballito de mar no varía entre los espacios semánticos de animales acuáticos y terrestres y tiene un comportamiento poco lineal. Al simplemente normalizar el resultado de *Path Distance Similarity*, era de esperar que se comportase de una manera similar.

Wu-Palmer Similarity

Similar a *Path Distance Dimilarity*, pero asigna pesos a los saltos en función de su profundidad en la jerarquía, siendo más costosos los saltos según se reduce la profundidad de sus extremos:

$$Sim_{\mathrm{WP}}(S1, S2) = \frac{2N}{N1 + N2 + 2N} \tag{4.2}$$

Siendo $S1$ y $S2$ los *synsets* a comparar, N la profundidad del antecesor común a $S1$ y $S2$ más específico y $N1$ y $N2$ las distancias a dicho antecesor desde $S1$ y $S2$ respectivamente.

Como se observa en la Figura 4.5, esta métrica si devuelve distintas similaridades con caballito de mar para los espacios semánticos de animales terrestres y acuáticos, pero con una diferencia muy pequeña. También se aprecia que es una métrica poco lineal, pues la similaridad con caballito de mar apenas varía de tiburón a libro, resultando la primera similaridad demasiado baja y la última demasiado alta.

```
>>> wn.wup_similarity(seahorse, seahorse)
1.0
>>> wn.wup_similarity(seahorse, shark)
0.6428571428571429
>>> wn.wup_similarity(seahorse, tiger)
0.5454545454545454
>>> wn.wup_similarity(seahorse, plant)
0.36363636363636365
>>> wn.wup_similarity(seahorse, book)
0.3333333333333333
```

Figura 4.5: Similaridad de *seahorse* con distintos *synsets* empleando *Wu-Palmer Similarity*

Resnik Similarity

Calcula la similaridad entre dos conceptos basándose únicamente en la información semántica de su antecesor común más específico:

$$Sim_{\mathrm{R}}(S1, S2) = -lnP_{\mathrm{LCS}}(S1, S2) \tag{4.3}$$

Siendo $S1$ y $S2$ los *synsets* a comparar, LCS el antecesor común a $S1$ y $S2$ más específico (*Least Common Subsumer*) y $P_{\mathrm{A}}(B, C)$ la frecuencia de aparición conjunta de los *synsets* B y C en la información semántica del *synset* A.

```
>>> wn.res_similarity(seahorse, seahorse, brown_ic)
13.772452732502442
>>> wn.res_similarity(seahorse, shark, brown_ic)
5.2175784741185165
>>> wn.res_similarity(seahorse, tiger, brown_ic)
2.2241504712318556
>>> wn.res_similarity(seahorse, plant, brown_ic)
1.5318337432196856
>>> wn.res_similarity(seahorse, book, brown_ic)
1.5318337432196856
```

Figura 4.6: Similaridad de *seahorse* con distintos *synsets* empleando *Resnik Similarity*

Como se observa en la Figura 4.6, esta métrica tampoco devuelve distintas similaridades con caballito de mar para los espacios semánticos de animales terrestres y acuáticos, como ocurre con *Path Distance Similarity* y *Leacock-Chodorow Similarity*. También se aprecia que es una métrica poco lineal, pues la similaridad con caballito de mar varía mucho de caballito de mar a tiburón, pero apenas de tigre a libro.

Jiang-Conrath Similarity

Basada en *Resnik Similarity*, pero también tiene en cuenta la información semántica de los dos *synsets* a comparar:

$$Sim_{\mathrm{JC}}(S1, S2) = 1/(P_{\mathrm{S1}}(LCS, S2) + P_{\mathrm{S2}}(LCS, S1) - 2 * P_{\mathrm{LCS}}(S1, S2)) \tag{4.4}$$

Siendo $S1$ y $S2$ los *synsets* a comparar, LCS el antecesor común a $S1$ y $S2$ más específico (*Least Common Subsumer*) y $P_{\mathrm{A}}(B, C)$ la frecuencia de aparición conjunta de los *synsets* B y C en la información semántica del *synset* A.

```
>>> wn.jcn_similarity(seahorse, seahorse, brown_ic)
1e+300
>>> wn.jcn_similarity(seahorse, shark, brown_ic)
0.061753333555040014
>>> wn.jcn_similarity(seahorse, tiger, brown_ic)
0.04631231880249754
>>> wn.jcn_similarity(seahorse, plant, brown_ic)
0.05539204836589075
>>> wn.jcn_similarity(seahorse, book, brown_ic)
0.05537714572831336
```

Figura 4.7: Similaridad de *seahorse* con distintos *synsets* empleando *Jiang-Conrath Similarity*

Como se observa en la Figura 4.7, esta métrica devuelve resultados poco lineales, pues la similaridad con caballito de mar varía muchísimo de caballito de mar a tiburón, pero apenas de tiburón a libro.

Lin Similarity

Bastante similar a *Jiang-Conrath Similarity*:

$$Sim_{\mathrm{L}}(S1, S2) = 2 * P_{\mathrm{LCS}}(S1, S2)\ (P_{\mathrm{S1}}(LCS, S2) + P_{\mathrm{S2}}(LCS, S1)) \tag{4.5}$$

Siendo $S1$ y $S2$ los *synsets* a comparar, LCS el antecesor común a $S1$ y $S2$ más específico (*Least Common Subsumer*) y $P_{\mathrm{A}}(B, C)$ la frecuencia de aparición conjunta de los *synsets* B y C en la información semántica del *synset* A.

Como se observa en la Figura 4.8, esta métrica es la más lineal de las distintas métricas estudiadas.

```
>>> wn.lin_similarity(seahorse, seahorse, brown_ic)
1.0
>>> wn.lin_similarity(seahorse, shark, brown_ic)
0.39187757428680065
>>> wn.lin_similarity(seahorse, tiger, brown_ic)
0.1708202569752729
>>> wn.lin_similarity(seahorse, book, brown_ic)
0.14504862325209697
>>> wn.lin_similarity(seahorse, plant, brown_ic)
0.14508199436170063
```

Figura 4.8: Similaridad de *seahorse* con distintos *synsets* empleando *Lin Similarity*

Comparativa

En la Tabla 4.1 resumo los resultados y las características de las seis métricas evaluadas.

Métrica	Intervalo	Linealidad
Path Distance Similarity	[0,1]	-
Leacock-Chodorow Similarity	?	-
Wu-Palmer Similarity	[0,1]	-
Resnik Similarity	?	-
Jiang-Conrath Similarity	[0,1]	-
Lin Similarity	[0,1]	✓

Tabla 4.1: Comparativa de las distintas métricas

Debido a que los *scores* devueltos por las `APIs` *cloud* se sitúan en el intervalo [0,1], pero sobre todo debido a la linealidad que presenta, emplearé *Lin Similarity* para la realización del experimento.

4.2.2. Aplicación

Una vez decidida la métrica de similaridad a emplear, queda definir como emplearla en el experimento comparativo:

- Si alguna etiqueta no cuenta con *synsets* al no estar presente en `WordNet`, la descarto.
- Debido a que una etiqueta puede contar con múltiples *synsets*, establezco que los dos *synsets* empleados -uno para cada etiqueta- sean aquellos entre los que exista una mayor similaridad de entre todos los pares posibles.

  Si tuviese en cuenta otras similaridades menores se perjudicarían los resultados en caso de similaridad plena, por ejemplo, si comparásemos dos etiquetas *dog*,

la similaridad no sería plena al tener también en cuenta la similaridad entre un perrito caliente y el animal.

Capítulo 5

Selección y obtención del *dataset*

Para realizar el experimento comparativo resulta necesario contar con un *dataset* adecuado y fiable para etiquetar sus imágenes con cada *vendor* y comparar estas etiquetas con el *ground truth* de dicho *dataset*.

5.1. Selección

Primeramente, obtengo una amplia lista de *datasets* candidatos de distintas fuentes [49][50][51][52][53], que recojo en el Apéndice E, para luego descartar de ésta aquellos que no sean adecuados.

Como el objetivo es la comparativa entre sistemas de etiquetado de imagen, descarto aquellos *datasets* que no etiqueten escenas (la mayor parte están orientados a la detección y clasificación de objetos), reduciendo así la lista a SUN Database, Places 2, Indoor Scene Recognition, MSCOCO y Flickr30k.

Compruebo que éstos son utilizables para el cometido del trabajo, según los términos de uso que hayan establecido sus creadores.

Tanto MSCOCO como Flickr30k etiquetan escenas en forma de descripciones en lenguaje natural, lo que complica bastante su uso en la comparativa. Además, en el caso de Flickr30k, estas descripciones son generadas por los usuarios de Flickr, a los que en ningún momento se les pide describir concisamente la imagen, por lo que no resultan fiables. Por lo tanto, restan SUN Database, Indoor Scene Recognition y Places2, para los que realizaré una revisión manual de calidad.

5.1.1. Revisión manual de calidad

Para comprobar que la calidad de los *datasets* escogidos es adecuada, compruebo que el *ground truth* de las imágenes realmente describe la escena que estas imágenes representan. Para ello recorro una muestra pequeña de sus imágenes, les asigno una

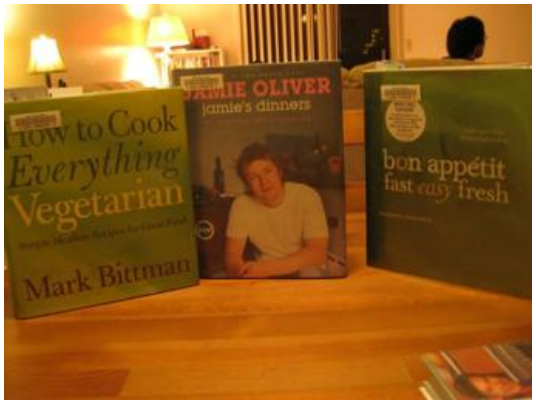
categoría y compruebo que esta es similar a su *ground truth*.

Los resultados de Indoor y SUN resultan bastante satisfactorios; pero los de Places resultan bastante llamativos en algún caso, como se observa en la Figura 5.1:

(a) Aquarium

(b) Conference room

(c) Forest



(d) Library

Figura 5.1: Ejemplos de imágenes del *dataset* Places y su categoría

Aún así, decido no descartar este *dataset* para comprobar si sus resultados resultan peores que los de los demás *datasets*. Así, emplearé los *datasets* SUN Database, Indoor Scene Recognition y Places2.

5.1.2. Características de los *datasets* escogidos

En la Tabla 5.1 resumo las características de los *datasets* escogidos

	Indoor	Places	SUN
Número de imágenes	15.620	>2.000.000	108.754
Número de categorías	67	365	397
Imágenes por categoría	>100	>5000	>100
Resolución	>200x200	256x256	>200x200

Tabla 5.1: Tabla comparativa de los *datasets* empleados

5.1.3. Preparación del *dataset* a emplear

Cada *dataset* está organizado de una forma distinta:

- Indoor está organizado de una forma sencilla: las imágenes de cada categoría están contenidas en un directorio con el *ground truth* de dicha categoría como nombre.
- En Places, todas las imágenes están contenidas en un mismo directorio y la relación entre estas y el identificador de su categoría y la relación entre los identificadores de las categorías y su *ground truth* están contenidas en dos ficheros distintos.
- En SUN, las imágenes de cada categoría están contenidas en un directorio llamado como el *ground truth* de dicha categoría y estos, a su vez, están organizados en carpetas según su letra inicial; también existen categorías cuyas imágenes están organizadas en dos subdirectorios: *indoor* y *outdoor*.

Como la estructura de Indoor hace bastante sencillo el procesamiento a la hora de realizar la comparativa, organizo los tres *datasets* de la misma forma que está organizado Indoor.

5.2. Análisis semántico de los *datasets* escogidos

Resulta útil realizar un análisis cualitativo de los *datasets* escogidos, comprobando si existen categorías semánticamente equivalentes o si algunas de ellas pueden agruparse en otras más genéricas, pues esto permite:

- Evaluar la composición y variedad de los datasets.
- Comprobar la viabilidad de establecer un conjunto común de etiquetas para los tres *datasets*.

Para ello; calculo el hiperónimo común más específico de cada par de categorías. Empleo la función `lowest_common_hypernyms` de `NLTK`, que dado un par de *synsets* calcula su antecesor común que más profundo se encuentre en la jerarquía de `WordNet`.

Para realizar una recategorización relevante, selecciono las categorías con al menos tres subcategorías asociadas. Estas categorías se corresponden con aquellos hiperónimos que se repiten por lo menos 9 veces, pues el hiperónimo común más profundo de una palabra y ella misma no siempre resulta ser la propia palabra, como se observa en la Tabla 5.2.

Frecuencia	Subcategorías
1	1
[4,6]	2
[9,12]	3
[16,20]	4

Tabla 5.2: Relación entre la frecuencia de un hiperónimo y su número de hipónimos

Una vez calculada la recategorización, compruebo que algunas categorías demasiado genéricas (como *artifact*, *object*, *instrumentality* o *entity*) agrupan la mayoría de categorías originales, por lo que vuelvo a calcular la categorización descartando estas categorías.

Mientras que en el Apéndice F recojo los resultados al completo, a continuación analizo los de cada *dataset*.

5.2.1. Indoor

24 categorías no tenían *synsets* sustantivos asociados al tener un *ground truth* mal compuesto, por lo que modifico éste manualmente (Apéndice F.1.1).

Tras ese renombrado, se descartan 17 categorías (Apéndice F.1.2) por no tener *synsets* sustantivos asociados, entre ellas:

- Un número relativamente elevado de categorías que representan escenas comunes, como *book store*, *hair salon* o *jewellery shop*.
- Algunas categorías demasiado específicas, como *children room* o *restaurant kitchen*.

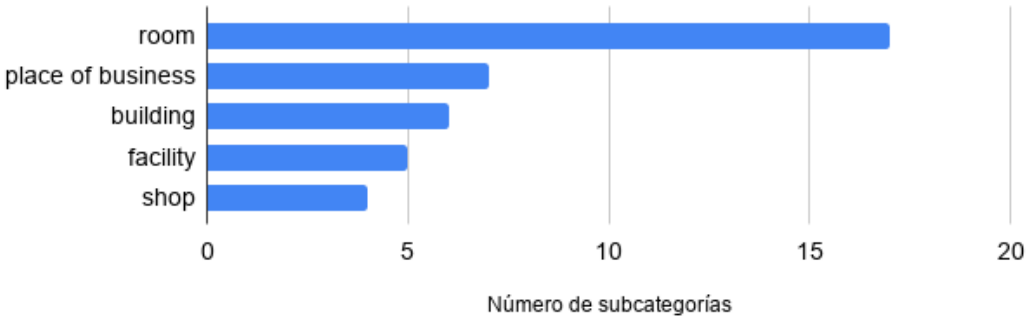


Figura 5.2: Categorización del *dataset* Indoor

Como se observa en la Figura 5.2, la categorización de este *dataset* -pequeño y con un *scope* reducido- no aporta demasiada información, pues existe un gran solapamiento entre las nuevas categorías (Apéndice F.1.3), de forma que las categorías originales podrían pertenecer a varias de éstas:

- *place of business* y *shop* se solapan en gran medida.
- *room* resulta obvia, teniendo en cuenta que el *dataset* está formado por imágenes de interiores.

5.2.2. Places

Descarto 53 categorías (Apéndice F.2.1) por no tener *synsets* sustantivos asociados, entre ellas:

- Algunas categorías que representan escenas comunes, como *soccer field*, *water park* o *construction site*.
- Algunas categorías demasiado específicas, como *artists loft*, *legislative chamber* o *natural history museum*.

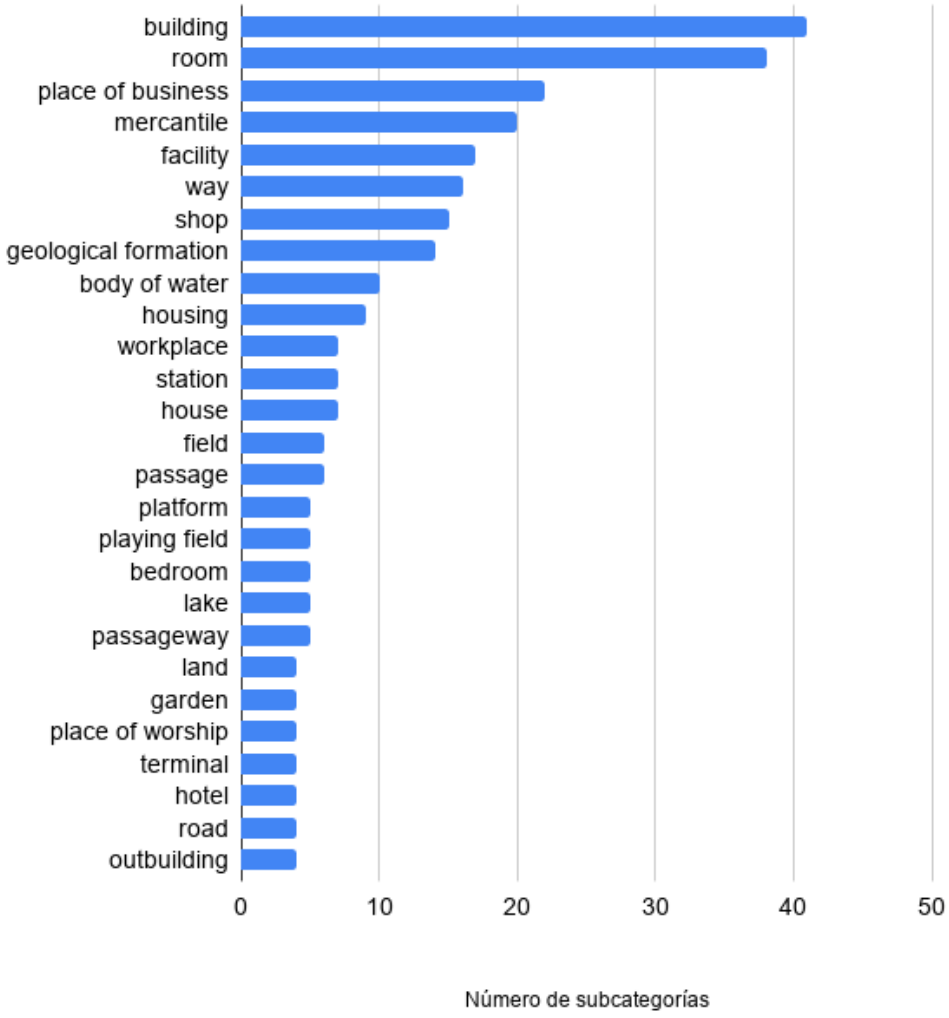


Figura 5.3: Categorización del *dataset* Places

Como se observa en la Figura 5.3, para este *dataset* también se produce un solapamiento notable entre algunas de las categorías obtenidas (Apéndice F.2.2), pero hay categorías lo suficientemente específicas y con un número suficiente de subcategorías como para aportar información relevante:

- La importancia de la categoría *room* resulta relevante debido a que el *scope* de este *dataset* es principalmente imágenes de exteriores.
- *building* y *facility* resultan demasiado genéricas.
- *place of business*, *shop* y *workplace* se solapan en gran medida.
- Otras categorías también se solapan, pero de una forma menos relevante; como *housing*, *house* y *hotel* o *body of water* y *lake*.

5.2.3. SUN

Descarto 85 categorías (Apéndice F.3.1) por no tener *synsets* sustantivos asociados, entre ellas:

- Un número mayor que en Places de categorías que representan escenas comunes, como *skatepark*, *jacuzzi* o *sea cliff*.
- Algunas categorías demasiado específicas, como *dinette vehicle*, *cheese factory* o *restaurant patio*.
- Bastantes categorías que no describen de forma concisa el contenido de la imagen, como *wine cellar barrel storage*, *temple south asia* o *car interior frontseat*.
- Bastantes categorías que no describen de forma concisa el contenido de la imagen, como *wine cellar barrel storage*, *temple south asia* o *car interior frontseat*.
- Bastantes categorías que no describen de forma concisa el contenido de la imagen, como *wine cellar barrel storage*, *temple south asia* o *car interior frontseat*.
- Unas pocas categorías con el *ground truth* mal formado, como *desert sand*, *canal natural* o *parking garage*.

Para este *dataset* se produce un solapamiento notable entre algunas de las categorías obtenidas (Apéndice F.3.2), pero hay categorías lo suficientemente específicas y con un número suficiente de subcategorías como para aportar información relevante, como se observa en la Figura 5.4:

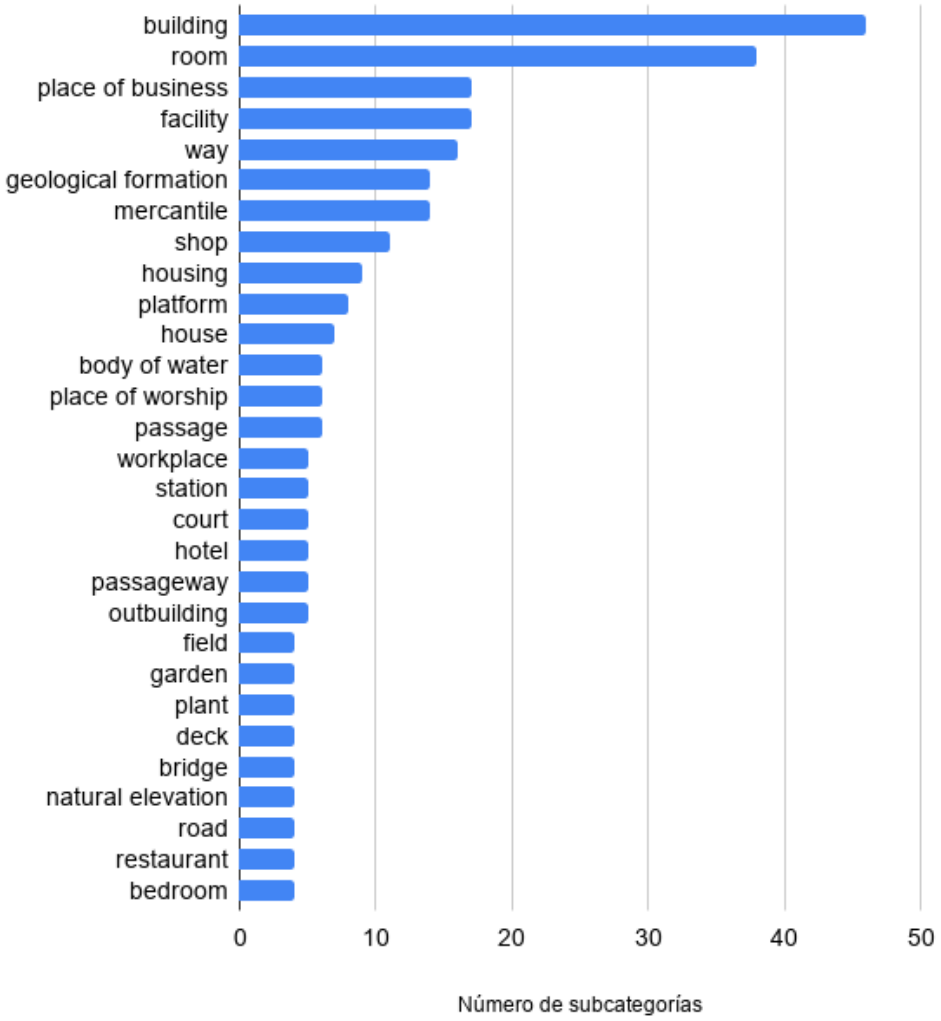


Figura 5.4: Categorización del *dataset* SUN

- Las categorías *building* y *facility* resultan demasiado genéricas. Además *facility* se solapa con otras categorías como *station*.
- *Place of business*, *shop*, *mercantile* y *workplace* se solapan en gran medida.
- Otras categorías también se solapan, pero de una forma menos relevante; como *geographical formation* y *natural elevation* o *housing* y *hotel* o *platform* y *deck*.

5.2.4. Conclusión

Aunque en todos los *datasets* exista un solapamiento importante entre algunas de las categorías obtenidas y algunas de ellas resulten demasiado genéricas, la caracterización realizada permite obtener una visión general los *datasets*.

Los tres *datasets* tienen dominios similares, siendo SUN el más heterogéneo, seguido de Places y de Indoor, que se podría ver como un subconjunto del anterior, pues tiene categorías similares pero con menor frecuencia.

Por otro lado, descarto la posibilidad de basar la comparativa en esta recategorización:

- El solapamiento entre categorías no supone un problema; pues al calcular la similaridad de las etiquetas generadas para una imagen con sus dos *ground truth* posibles -el de las dos categorías que se solapan-, `NLTK` devolvería resultados parecidos al tener las dos categorías significados similares.
- Que las categorías resultantes sean demasiado generales si supone un problema, pues se valoraría la eficacia de los sistemas de etiquetado a un nivel de detalle demasiado bajo.

Capítulo 6

Etiquetado del *dataset*

Una vez obtenidos y preparados los *datasets* a emplear, los etiqueto con cada una de las soluciones de etiquetado seleccionadas.

Debido a que solamente la primera etiqueta o la segunda de las generadas para cada imagen resulta relevante y el procesamiento resulta pesado, limito las etiquetas obtenidas a 5.

Debido al límite de 5.000 imágenes de algunas `APIs` de etiquetado, limito el etiquetado a 5 imágenes por categoría, lo que supone un total de 4.145 imágenes.

6.1. Análisis de las etiquetas

Una vez etiquetado la totalidad del *dataset*, resulta interesante obtener algunas métricas acerca de las etiquetas obtenidas.

Número de etiquetas por *vendor*

En la Tabla 6.1 recojo el número medio de etiquetas por imagen que cada *vendor* ha generado. Estos datos resultan esperables, pues el número de etiquetas por imagen

Vendor	Amazon	Azure	Clarifai	Google	Watson
Etiquetas por imagen	4,83	4,86	5	4,86	4,87

Tabla 6.1: Número medio de etiquetas por imagen de cada *vendor*

estaba limitado a 5, pero destaca que Clarifai haya devuelto 5 etiquetas para cada una de las 4.145 imágenes del *dataset*.

Etiquetas distintas obtenidas

En la Tabla 6.2 recojo el número de etiquetas distintas que cada *vendor* ha generado para cada *dataset*. Lo más destacable resulta que para el *dataset* Places se han generado

Vendors	Número de etiquetas			
	Indoor	Places	SUN	Promedio
Amazon	776	526	788	696
Azure	631	440	649	573
Clarifai	798	548	814	720
Google	1313	870	1338	1173
Watson	1943	1169	1976	1695
Promedio	1092	710	1172	991

Tabla 6.2: Etiquetas distintas generadas por cada *vendor* para cada *dataset*

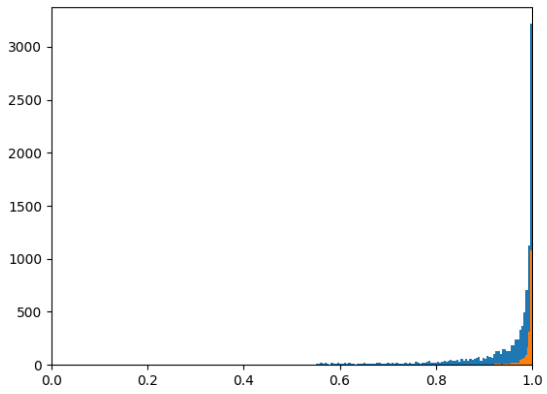
notablemente menos etiquetas que para los otros *datasets*, para los que se han generado un número similar de etiquetas. Por otro lado, destaca que Watson haya generado casi el triple de etiquetas distintas que Azure.

6.2. Análisis de las confianzas

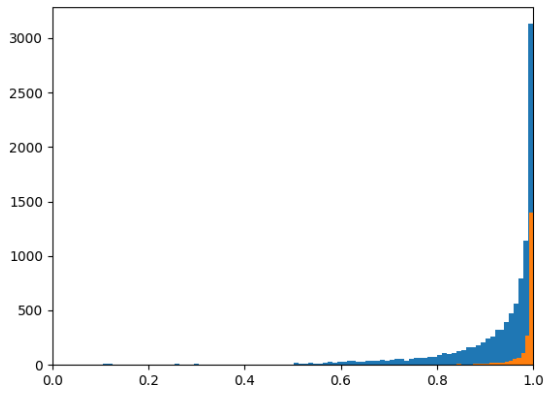
Sería interesante ponderar la similaridad entre el *ground truth* y las etiquetas que genera cada *vendor* según la confianza de dichas etiquetas, para así cuantificar la precisión de los *vendors* teniendo en cuenta tanto sus etiquetas como las confianzas asociadas a estas. Pero esto requiere conocer la distribución de las confianzas, información que los *vendors* no proporcionan, por lo que procedo al estudio de sus distribuciones.

6.2.1. Estudio de su distribución

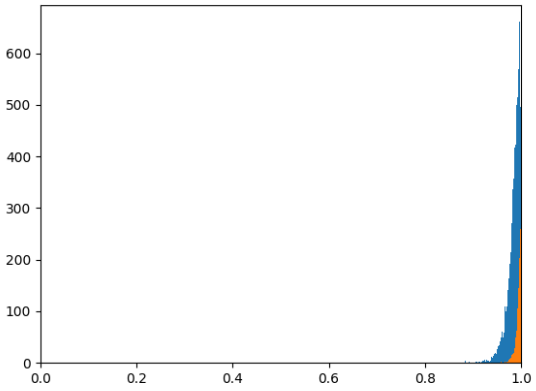
En la Figura 6.1, recojo los histogramas de las confianzas obtenidas de cada *vendor*. El color azul representa la totalidad de las confianzas mientras que el naranja representa sólo la máxima de entre todas las obtenidas para una misma imagen.



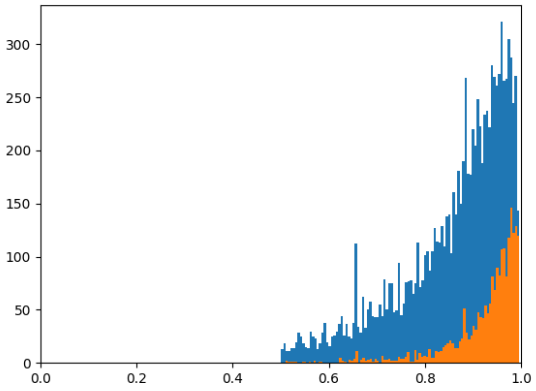
(a) *Scores* de Amazon



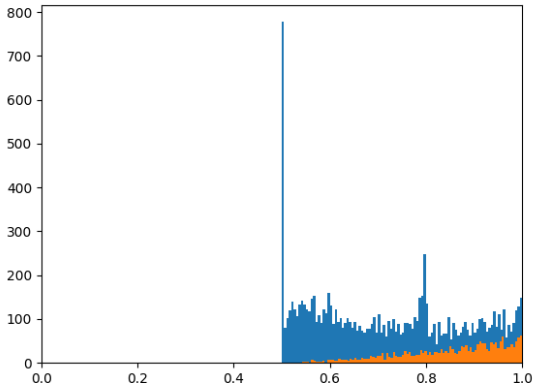
(b) *Scores* de Azure



(c) *Scores* de Clarifai



(d) *Scores* de Google



(e) *Scores* de Watson

Figura 6.1: Distribución de las confianzas generadas por cada *vendor*

En estos se aprecia lo siguiente:

- Amazon, Azure y Clarifai presentan distribuciones parecidas, mientras que Google y Watson presentan distribuciones muy distintas a estos.
- Las confianzas menores de 0,5 resultan inapreciables para todos los *vendors*.
- Las distribuciones para las confianzas máximas y la totalidad de estas resultan similares, salvo en el caso de Watson.
- Tanto para Google como para Watson no he registrado ningún *score* menor de 0,5. En el caso de Watson parece que los *scores* menores de 0,5 se aumentan a esta cifra, que presenta una frecuencia enorme, pero en el caso de Google puede deberse al límite de cinco etiquetas por imagen.

Para cuantificar lo observado en los histogramas, en la Tabla 6.3 recojo algunos percentiles de las confianzas generadas por cada *vendor*. En esta se observa lo siguiente:

Vendor	Confianzas	Percentiles					
		20	40	60	70	80	90
Amazon	Máximas	0.983	0.994	0.997	0.998	0.999	0.999
	Todas	0.913	0.974	0.992	0.995	0.998	0.999
Azure	Máximas	0.981	0.994	0.998	0.999	0.999	0.999
	Todas	0.867	0.948	0.981	0.990	0.995	0.998
Clarifai	Máximas	0.990	0.994	0.996	0.997	0.998	0.998
	Todas	0.976	0.985	0.991	0.993	0.995	0.997
Google	Máximas	0.884	0.937	0.961	0.972	0.980	0.988
	Todas	0.769	0.864	0.916	0.938	0.957	0.975
Watson	Máximas	0.755	0.842	0.908	0.932	0.956	0.980
	Todas	0.558	0.652	0.778	0.820	0.888	0.948

Tabla 6.3: Percentiles de las confianzas generadas de cada *vendor*

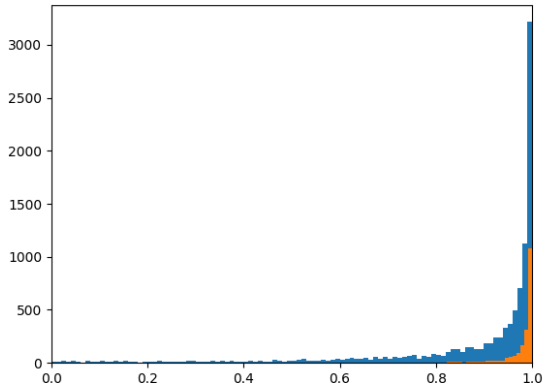
- En torno al 40 % de las confianzas generadas por Amazon, Azure y Clarifai son mayores de 0.99 mientras que para Google y Watson suponen menos del 10 %.
- Los percentiles de las confianzas máximas y la totalidad de estas resultan muy similares, pues sus distribuciones son similares.

6.2.2. Normalización de sus distribuciones

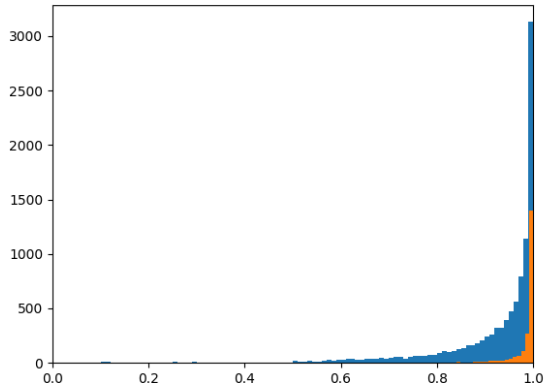
Para poder comparar estas distribuciones de una manera más precisa, efectúo una normalización de escala mediante la siguiente fórmula:

$$S' = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \tag{6.1}$$

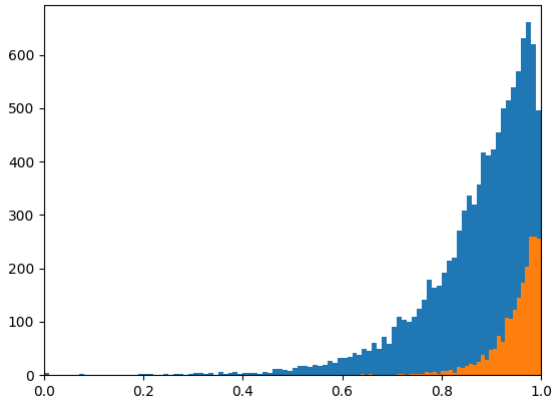
En la Figura 6.2 recojo los histogramas una vez realizada la normalización:



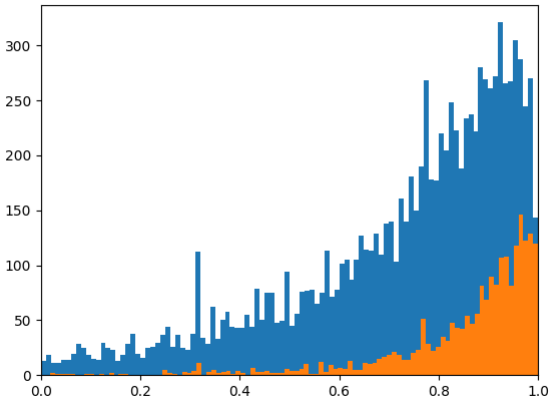
(a) *Scores* de Amazon



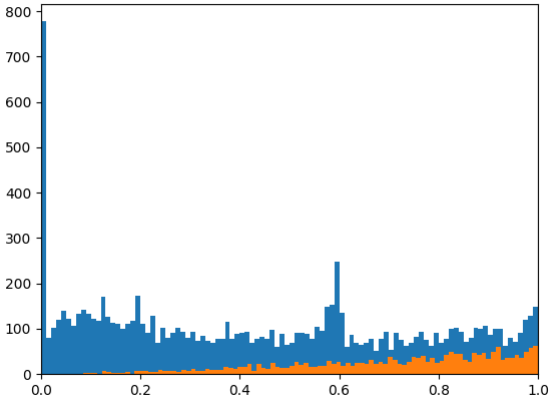
(b) *Scores* de Azure



(c) *Scores* de Clarifai



(d) *Scores* de Google



(e) *Scores* de Watson

Figura 6.2: Distribución normalizada de las confianzas de cada *vendor*

En estos histogramas se aprecia lo siguiente:

- Amazon y Azure siguen presentando distribuciones similares, mientras que Clarifai se distingue de ellos; aunque en los tres casos, las confianzas más bajas siguen siendo inapreciables.
- Confirmo que las distribuciones para los *scores* máximos y la totalidad de ellas resultan similares, salvo en el caso de Watson.

Como explico al comienzo de esta sección, sería interestante ponderar la similaridad entre el *ground truth* y las etiquetas que genera cada *vendor* según la confianza de dichas etiquetas.

Debido a las distribuciones tan distintas que presentan los *vendors* en sus confianzas y la dificultad de normalizarlos de una forma adecuada que no distorsione la métrica

de similaridad -que considero más relevante-, no pondero la similaridad entre etiquetas y *ground truth* con las confianza de dichas etiquetas.

Capítulo 7

Análisis de resultados

Una vez etiquetados los tres *datasets* empleando los cinco *vendors*, procedo a analizar las etiquetas obtenidas, junto a sus confianzas asociadas.

Primeramente analizo la similaridad de dichas etiquetas con el *ground truth*; a continuación, la similaridad entre las etiquetas generadas por los distintos *vendors* para las mismas imágenes; posteriormente, la distribución de la similaridad entre las etiquetas obtenidas y el *ground truth* para los distintos *vendors* y *datasets*; y finalmente analizo las imágenes que mejores y peores etiquetas han obtenido.

7.1. Precisión de cada *vendor*

El dato más relevante resultante del experimento es la precisión de cada *vendor*, es decir, qué *vendor* genera etiquetas más similares al *ground truth*.

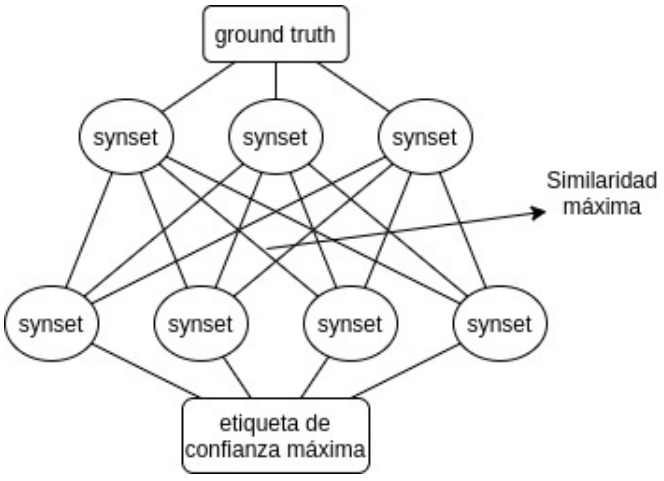


Figura 7.1: Cálculo de la similaridad entre las etiquetas de confianza máxima y el *ground truth*.

Como ilustro en la Figura 7.1, calculo dicha similaridad de la siguiente forma: para

una imagen y un *vendor* determinados, obtengo todos los *synsets* asociados a su *ground truth* y a la etiqueta generada con la máxima confianza y calculo la similaridad de cada par resultante de cruzar ambos conjuntos de *synsets* y escojo la máxima entre estas.

En la Tabla 7.1 recojo la similaridad promedio entre el *ground truth* y las etiquetas de confianza máxima generadas por cada *vendor*, diferenciando cada *dataset*.

Vendors	Similaridad (%)			
	Indoor	Places	SUN	Promedio
Amazon	34,41	23,58	25,88	27,96
Azure	20,03	16,16	14,36	16,85
Clarifai	33,26	21,98	25,23	26,82
Google	48,77	22,06	29,19	33,34
Watson	40,59	24,96	32,46	32,67
Promedio	35,41	21,74	25,42	27,52

Tabla 7.1: Similaridades promedio entre las etiquetas de confianza máxima y el *ground truth*.

En ésta destacan varios hechos:

- Son resultados bajos, pues la similaridad promedio es de un 27,52 % y el mejor caso (Google en Indoor) solamente alcanza el 48,77 %.
- En Indoor se obtienen resultados notablemente mejores (aunque siguen siendo malos, con un 35,11 % de similaridad promedio), mientras que en Places y SUN se obtienen resultados similares.

Atendiendo al promedio de los tres *datasets*, se podrían ordenar los *vendors* en función de su precisión:

1. Google, con una similaridad media del 33,34 %.
2. Watson, con una similaridad media del 32,67 %, bastante cercana a la de Google.
3. Amazon, con una similaridad media del 27,96 %.
4. Clarifai, con una similaridad media del 26,82 %, bastante cercana a la de Amazon.
5. Azure, con una similaridad media del 16,85 %, casi la mitad de la Google.

7.2. Coincidencia entre *vendors*

Una vez conocida la precisión de cada *vendor*, resulta interesante analizar la coincidencia entre estos, es decir, la similaridad entre las etiquetas que generan para las mismas imágenes.

Como ilustro en la Figura 7.2, calculo dicha similaridad de la siguiente forma: para una imagen y un par de *vendor* determinados, obtengo todos los *synsets* asociados a ambas etiquetas de máxima confianza y calculo la similaridad de cada par resultante de cruzar ambos conjuntos de *synsets* y escojo la máxima entre estas.

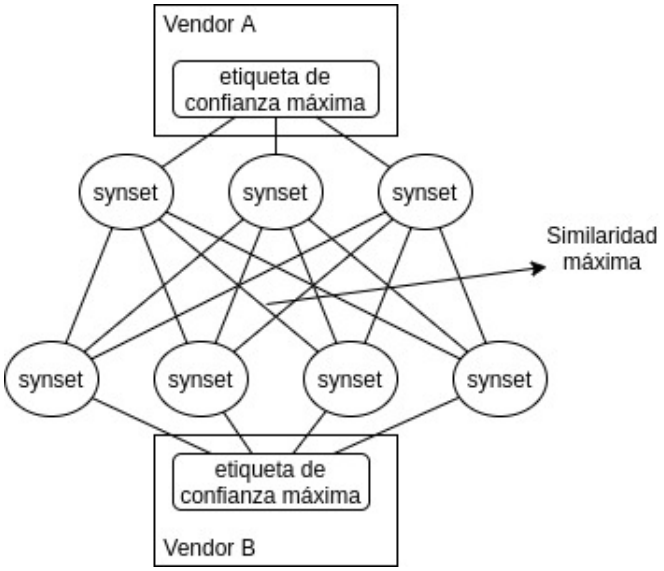


Figura 7.2: Cálculo de la similaridad entre etiquetas de una misma imagen.

En la Tabla 7.2 recojo la similaridad promedio entre las etiquetas generadas por cada par de *vendors* para las mismas imágenes, diferenciando cada *dataset*.

Vendors		Similaridad (%)			
		Indoor	Places	SUN	Promedio
Azure	Amazon	24,26	27,44	25,25	25,65
Azure	Clarifai	24,55	27,61	24,89	25,68
Azure	Google	24,43	25,85	23,02	24,43
Azure	Watson	14,50	22,13	17,77	18,13
Amazon	Clarifai	39,89	34,35	37,49	37,24
Amazon	Google	37,68	31,15	33,83	34,22
Amazon	Watson	27,43	34,94	30,21	30,86
Clarifai	Google	38,98	34,27	37,07	36,78
Clarifai	Watson	26,72	29,30	26,91	27,64
Google	Watson	36,08	27,92	27,15	30,38
Promedio		29,45	29,50	28,36	29,10

Tabla 7.2: Similaridades promedio entre las etiquetas generadas para las mismas imágenes.

En esta se observa que son valores bajos, pues la coincidencia media es del 29,10 % y la máxima es del 39,89 %, pero resulta complicado valorar las diferencias entre los

distintos pares; por lo que en la Tabla 7.3 recojo la desviación respecto a la coincidencia media.

Vendors		Variación de la similaridad (%)			
		Indoor	Places	SUN	Promedio
Azure	Amazon	-4,84	-1.66	-3.85	-3.45
Azure	Clarifai	-4.55	-1.49	-4.21	-3.42
Azure	Google	-4.67	-3.25	-6.08	-4.67
Azure	Watson	-14.6	-6.97	-11.33	-10.97
Amazon	Clarifai	10.79	5.25	8.39	8.14
Amazon	Google	8.58	2.05	4.73	5.12
Amazon	Watson	-1.67	5.84	1.11	1.76
Clarifai	Google	9.88	5.17	7.97	7.68
Clarifai	Watson	-2.38	0.2	-2.19	-1.46
Google	Watson	6.95	-1.18	-1.95	1.28
Promedio		0.35	0.5	-0.74	0

Tabla 7.3: Variación respecto a la media de las similaridades promedio entre las etiquetas generadas para las mismas imágenes.

En esta se observan varios hechos:

- La coincidencia varía poco entre los distintos *datasets*.
- Las combinaciones que incluyen a Azure son, con diferencia, las que menor similaridad presentan. Esto concuerda con el estudio de la precisión de los *vendors* (7.1), donde Azure también resultaba el peor etiquetador.
- Amazon, Clarifai, Google y Watson no presentan grandes variaciones; salvo para algunos *datasets*, en los que Watson obtiene peores resultados.

7.3. Distribución de la similaridad

En esta sección analizo cómo se distribuye la similaridad entre el *ground truth* de una imagen y las etiquetas obtenidas para esta, agrupando tanto por *vendor* como por *dataset*.

Como ilustro en la Figura 7.3, calculo dicha similaridad de la siguiente forma: obtengo todos los *synsets* asociados a una determinada etiqueta de una imagen concreta y al *ground truth* de dicha imagen; posteriormente, calculo la similaridad de cada par resultante de cruzar ambos conjuntos de *synsets* y escojo la máxima entre estas.

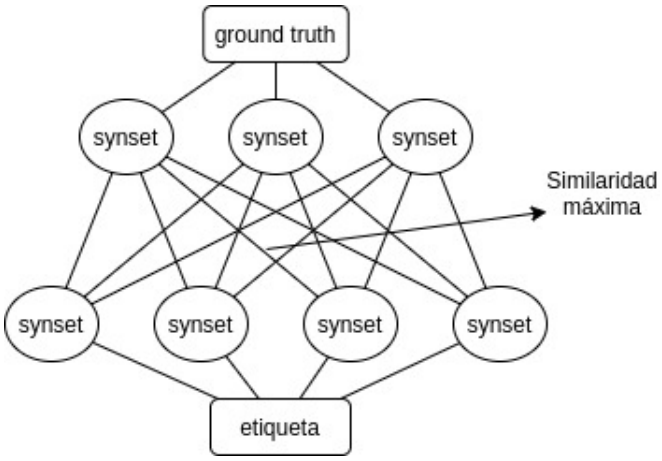
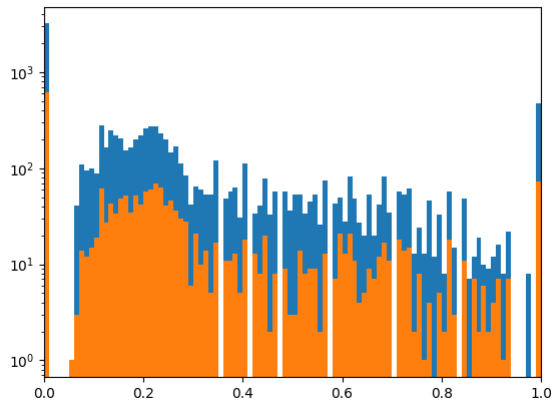


Figura 7.3: Cálculo de la similaridad entre etiquetas y *ground truth*

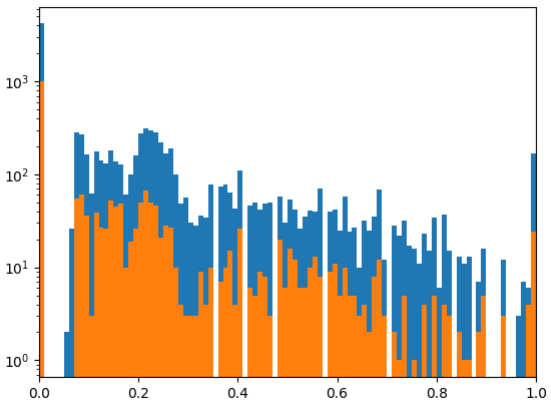
7.3.1. Por *vendors*

En la Figura 7.4 se muestran el histograma para cada uno de los *vendors* y para la combinación de ellos.

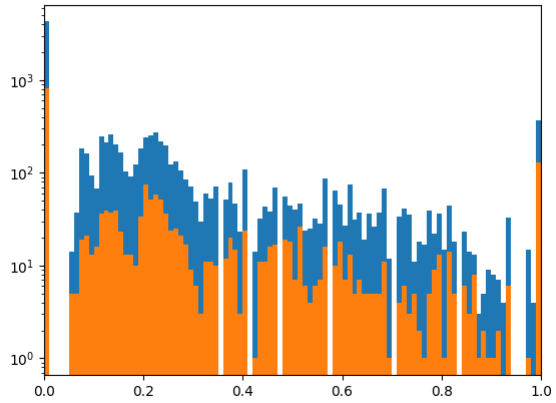
Debido a que la similaridad nula y plena presentan una frecuencia mucho mayor que el resto, aplico a la frecuencia una escala logarítmica. El color azul recoge la totalidad de las etiquetas y el naranja sólo aquella de confianza máxima para cada imagen.



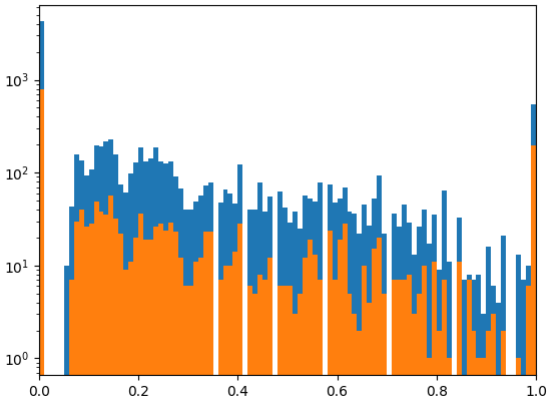
(a) Similaridades de Amazon



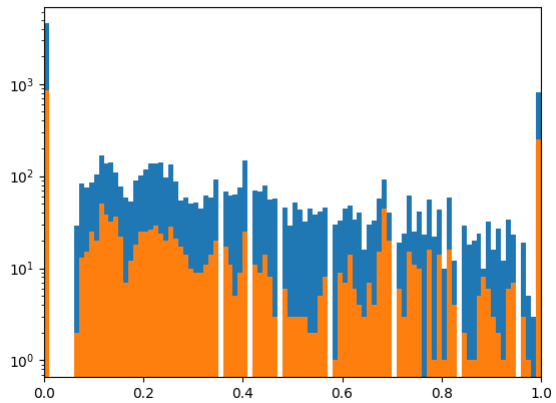
(b) Similaridades de Azure



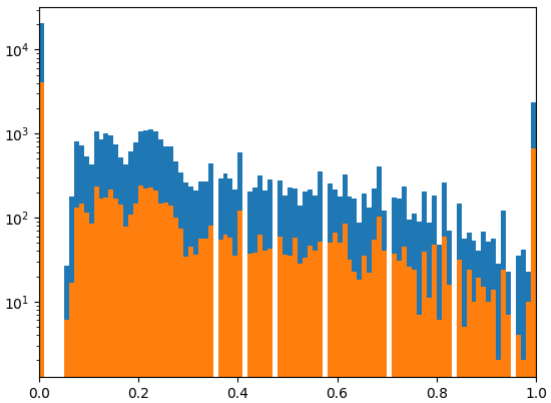
(c) Similaridades de Clarifai



(d) Similaridades de Google



(e) Similaridades de Watson



(f) Similaridades combinadas

Figura 7.4: Distribución de la similaridad entre las etiquetas generadas y el *ground truth*.

En estos histogramas se observan varios hechos:

- La frecuencia de las etiquetas con similaridad nula es muy alta, tanto si se trata de etiquetas con confianza máxima o no; al igual que se aprecia una cantidad destacable de etiquetas con similaridad plena.
- Las similaridades bajas son más frecuentes que las altas.
- Entre el 20 % y 25 % de las etiquetas generadas obtienen una similaridad entre el 5 % y 30 %.
- La distribución de las similaridades para la totalidad de etiquetas y sólo para aquellas con confianza máxima resulta muy similar.

- En el caso de Azure, para las etiquetas con confianza máxima, se aprecia un repunte en las similaridades menores del 10 %.

En la Tabla 7.4 cuantifico las frecuencias que suponen las etiquetas con similaridad nula o plena.

Vendor	Similaridad nula (%)		Similaridad plena (%)	
	Total de etiquetas	*Score* máximo	Total de etiquetas	*Score* máximo
Amazon	31,55	29,42	4,65	3,45
Azure	40,81	47,13	1,64	1,14
Clarifai	40,41	38,93	3,42	6,15
Google	41,18	37,13	5,2	9,37
Watson	44,18	40,11	7,95	11,78
Promedio	39,62	38,53	4,51	6,37

Tabla 7.4: Etiquetas con similaridad con el *ground truth* nula o plena

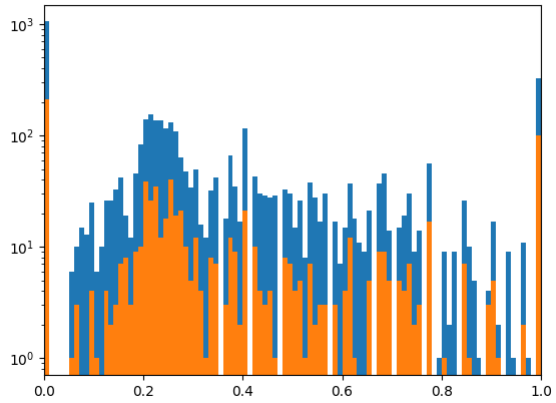
Estos porcentajes reflejan varios hechos:

- Los porcentajes de etiquetas con similaridad nula y plena son bastante similares para todas las etiquetas y solo para aquellas con confianza máxima.
- Las etiquetas de confianza máxima de todos los vendors están equivocadas para el 38,53 % de las imágenes y en el caso de Azure esta cifra aumenta al 47,13 %.
- Amazon tiene un porcentaje de etiquetas con similaridad nula bastante más bajo que el resto de *vendors*.
- Watson es a la vez el *vendor* con más etiquetas de similaridad nula y plena, destacando bastante su porcentaje de etiquetas con similaridad plena.

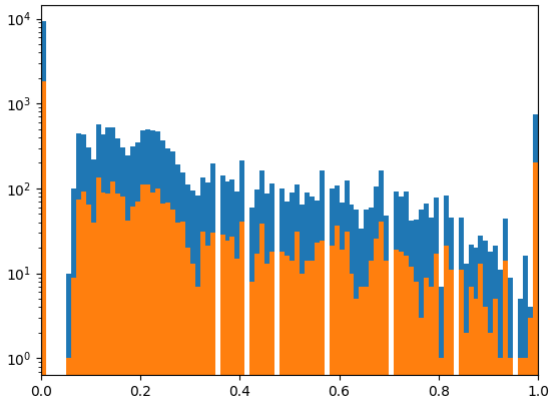
7.3.2. Por *datasets*

En la Figura 7.5 recojo los histogramas de la similaridad entre el *ground truth* y las etiquetas obtenidas para cada uno de los *datasets* etiquetados.

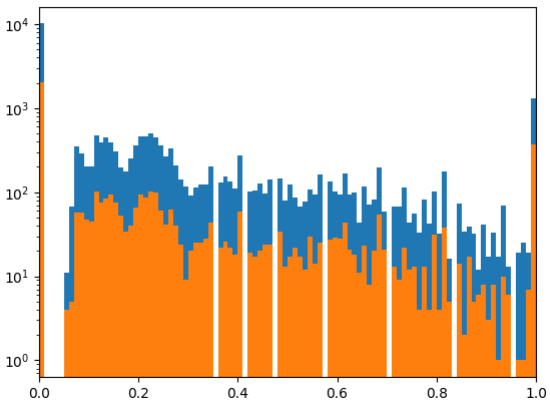
Debido a que la similaridad nula y plena presentan una frecuencia mucho mayor que el resto, aplico a la frecuencia una escala logarítmica. El color azul recoge la totalidad de las etiquetas y el naranja sólo aquella de confianza máxima para cada imagen.



(a) Similaridades de Indoor



(b) Similaridades de Places



(c) Similaridades de SUN

Figura 7.5: Distribución de la similaridad por *datasets*

En estos se observa que:

- La distribución de las similaridades para la totalidad de etiquetas y sólo para aquellas con confianza máxima vuelve a resultar bastante similar
- Para Indoor se generan etiquetas más similares al *ground truth*:
  - Al contrario que Places y SUN, no presenta un pico en la frecuencia de las similaridades menores al 20 %.
  - La similaridad plena es más frecuente que para Places y SUN.

En la Tabla 7.5 cuantifico las frecuencias que suponen las etiquetas con similaridad nula o plena.

Dataset	Similaridad nula (%)		Similaridad plena (%)	
	Total de etiquetas	*Score* máximo	Total de etiquetas	*Score* máximo
Indoor	25,76	25,24	7,77	12,02
Places	39,93	38,76	3,2	4,24
SUN	41,69	40,58	5,33	7,46
Promedio	35,79	34,86	5,43	7,9

Tabla 7.5: Etiquetas con similaridad con el *ground truth* nula o plena

Estos porcentajes reflejan varios hechos:

- Al igual que en el estudio por *vendors*, los porcentajes de etiquetas con similaridad nula y plena resultan similares para la totalidad de etiquetas y solo para aquellas con confianza máxima.
- Indoor cuenta notablemente con los mejores porcentajes de etiquetas con similaridad nula y plena.
- SUN cuenta con el porcentaje más alto de etiquetas con similaridad nula, pero tiene un mejor porcentaje que Places en cuanto a similaridad plena.

7.4. Imágenes con mejores y peores similaridades

En este apartado recojo, para cada *dataset*, las imágenes con mayor y menor similaridad media entre su *ground truth* y las etiquetas generadas por todos los *vendors*. Esto puede resultar útil para comprender casos concretos en los que los etiquetadores se comportan de una forma peculiar.

7.4.1. Indoor

Mejor caso

La imagen con mejores resultados, que cuenta con «*grocery store*» como *ground truth*, es la siguiente:

Como se observa en la Tabla 7.6, los resultados son generalmente buenos y todos los *vendors* proporcionan al menos una etiqueta equivalente al *ground truth*:

Amazon	Etiqueta	*grocery store*	*shop*	*supermarket*	*market*	*indoors*
	Score (%)	*99,36*	*99,36*	*98,68*	*98,68*	*71,58*
	Similaridad (%)	*100,0*	*84,0*	*84,0*	*100,0*	*0*
Azure	Etiqueta	*text*	*convenience store*	*ceiling*	*indoor*	*shelf*
	Score (%)	*99,34*	*98,01*	*95,71*	*95,46*	*83,67*
	Similaridad (%)	*25,0*	*0,0*	*23,0*	*0*	*24,0*
Clarifai	Etiqueta	*shelf*	*stock*	*supermarket*	*commerce*	*market*
	Score (%)	*99,69*	*99,51*	*99,44*	*99,08*	*98,61*
	Similaridad (%)	*24,0*	*25,0*	*84,0*	*0*	*100,0*
Google	Etiqueta	*supermarket*	*retail*	*product*	*grocery store*	*building*
	Score (%)	*94,63*	*92,52*	*91,03*	*90,98*	*90,82*
	Similaridad (%)	*84,0*	*0*	*31,0*	*100,0*	*53,0*
Watson	Etiqueta	*retail store*	*building*	*marketplace*	*grocery store*	*supermarket*
	Score (%)	*96,3*	*96,3*	*88,0*	*88,0*	*77,1*
	Similaridad (%)	*89,0*	*53,0*	*96,0*	*100,0*	*84,0*

Tabla 7.6: Resultados para la mejor imagen de *Indoor*

También se observa como `NLTK` arroja una similaridad nula entre *convenience store* y *grocery store*, lo que es incorrecto; pues la primera significa "*a shop that sells food, drinks, etc. and is usually open until late*", mientras que la segunda significa "*a store where food and small items for the house are sold*", según [54]. Este comportamiento resulta inherente a `WordNet` y causará la obtención de similaridades bajas, afectando a todos los *vendors* por igual.

Peor caso

La imagen con peores resultados, que cuenta con *ground truth* «*deli*», es la siguiente:

Como se observa en la Tabla 7.7, la similaridad entre el *ground truth* y las etiquetas generadas es siempre nula; esto lo achaco a que *deli* es una palabra muy específica, pero resulta extraño que su similaridad con *shop* resulta nula.

Amazon	Etiqueta	*human*	*person*	*shop*	*flyer*	*advertisement*
	Score (%)	*98,82*	*98,82*	*97,26*	*68,07*	*68,07*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	*0*
Azure	Etiqueta	*text*	*person*	*clothing*	*newspaper*	*man*
	Score (%)	*99,92*	*91,28*	*90,59*	*88,9*	*87,06*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	*0,0*
Clarifai	Etiqueta	*stock*	*adult*	*people*	*portrait*	*shelf*
	Score (%)	*98,76*	*98,39*	*98,13*	*97,48*	*96,65*
	Similaridad (%)	*0,0*	*0,0*	*0*	*0,0*	*0,0*
Google	Etiqueta	*games*				
	Score (%)	*52,12*				
	Similaridad (%)	*0,0*				
Watson	Etiqueta	*retail store*	*building*	*shop*	*convenience store*	*retail merchant*
	Score (%)	*83,1*	*83,1*	*82,8*	*73,7*	*63,4*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	*0,0*

Tabla 7.7: Resultados para la peor imagen de *Indoor*

7.4.2. Places

Mejor caso

La imagen con mejores resultados, que cuenta con *ground truth* «*house*», es la siguiente:

Como se observa en la Tabla 7.8, todos los *vendors* proporcionan al menos una etiqueta equivalente al *ground truth* y sólo uno (Azure) no la etiqueta como residencia sino como *building*.

Amazon	Etiqueta	*housing*	*building*	*house*	*cottage*	*villa*
	Score (%)	*99,78*	*99,78*	*99,63*	*99,63*	*99,0*
	Similaridad (%)	*94,0*	*88,0*	*100,0*	*78,0*	*72,0*
Azure	Etiqueta	*outdoor*	*building*	*sky*	*architecture*	*porch*
	Score (%)	*96,81*	*94,97*	*77,78*	*61,57*	*61,12*
	Similaridad (%)	*0*	*88,0*	*10,0*	*59,0*	*47,0*
Clarifai	Etiqueta	*house*	*home*	*architecture*	*no person*	*bungalow*
	Score (%)	*99,71*	*99,61*	*98,91*	*98,28*	*98,09*
	Similaridad (%)	*100,0*	*100,0*	*59,0*	*0*	*78,0*
Google	Etiqueta	*home*	*house*	*property*	*lighting*	*light*
	Score (%)	*97,88*	*96,94*	*95,94*	*88,54*	*87,91*
	Similaridad (%)	*100,0*	*100,0*	*40,0*	*32,0*	*37,0*
Watson	Etiqueta	*building*	*residence*	*religious building*	*detached house*	*duplex house*
	Score (%)	*86,9*	*63,4*	*58,2*	*57,3*	*51,5*
	Similaridad (%)	*88,0*	*92,0*	*0*	*67,0*	*67,0*

Tabla 7.8: Resultados para la mejor imagen de *Places*

Peor caso

La imagen con peores resultados, que cuenta con *ground truth* «*heliport*», es la siguiente:

Como se observa en la Tabla 7.9, el problema reside en que no se ha etiquetado *heliport* literalmente, pues todos los *vendors* la han etiquetado como *helicopter*, pero esta cuenta con una similaridad con *heliport* nula.

Amazon	Etiqueta	*aircraft*	*transportation*	*vehicle*	*helicopter*	
	Score (%)	*99,9*	*99,9*	*99,9*	*99,9*	
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	
Azure	Etiqueta	*outdoor*	*sky*	*helicopter*	*aircraft*	*transport*
	Score (%)	*99,36*	*98,36*	*93,39*	*92,32*	*83,64*
	Similaridad (%)	*0*	*0,0*	*0,0*	*0,0*	*0,0*
Clarifai	Etiqueta	*helicopter*	*aircraft*	*transportation*	*vehicle*	*military*
	Score (%)	*99,89*	*99,2*	*99,06*	*98,65*	*98,34*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	*0*
Google	Etiqueta	*helicopter*	*helicopter rotor*	*rotorcraft*	*vehicle*	*aircraft*
	Score (%)	*99,15*	*98,26*	*97,49*	*96,32*	*95,66*
	Similaridad (%)	*0,0*	*0*	*0*	*0,0*	*0,0*
Watson	Etiqueta	*aircraft*	*craft*	*tail rotor*	*rotating mechanism*	*helicopter*
	Score (%)	*87,7*	*87,7*	*73,8*	*73,6*	*73,6*
	Similaridad (%)	*0*	*0,0*	*0,0*	*0,0*	*0,0*

Tabla 7.9: Resultados para la peor imagen de *Places*

7.4.3. SUN

Mejor caso

La imagen con mejores resultados, que cuenta con *ground truth* «*house*», es la siguiente:

Como se observa en la Tabla 7.10, todos los *vendors* generan una etiqueta sinónima o hipónima de *house*.

Amazon	Etiqueta	*building*	*housing*	*urban*	*neighborhood*	*house*
	Score (%)	*99,39*	*99,2*	*98,97*	*98,97*	*97,36*
	Similaridad (%)	*88,0*	*94,0*	*0*	*86,0*	*100,0*
Azure	Etiqueta	*building*	*outdoor*	*house*	*car*	*vehicle*
	Score (%)	*99,49*	*99,35*	*95,86*	*94,75*	*92,76*
	Similaridad (%)	*88,0*	*0*	*100,0*	*50,0*	*35,0*
Clarifai	Etiqueta	*house*	*architecture*	*street*	*roof*	*home*
	Score (%)	*99,14*	*97,52*	*95,47*	*95,09*	*95,07*
	Similaridad (%)	*100,0*	*59,0*	*84,0*	*27,0*	*100,0*
Google	Etiqueta	*property*	*house*	*cottage*	*building*	*home*
	Score (%)	*97,7*	*94,44*	*92,62*	*90,82*	*85,09*
	Similaridad (%)	*40,0*	*100,0*	*78,0*	*88,0*	*100,0*
Watson	Etiqueta	*housing*	*dwelling*	*semi-detached house*	*building*	*duplex house*
	Score (%)	*86,1*	*84,3*	*84,1*	*76,4*	*58,9*
	Similaridad (%)	*94,0*	*98,0*	*0,0*	*88,0*	*67,0*

Tabla 7.10: Resultados para la mejor imagen de *SUN*

Peor caso

La imagen con peores resultados, que cuenta con *ground truth* «*art school*», es la siguiente:

Como se observa en la Tabla 7.11, tres de cinco *vendors* reconocen a los niños y que estos están pintando o incluso arte. Pero `NLTK` proporciona similaridades siempre nulas.

Amazon	Etiqueta	*human*	*person*	*clothing*	*apparel*	*footwear*
	Score (%)	*99,81*	*99,81*	*98,45*	*98,45*	*64,5*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0,0*	*0,0*
Azure	Etiqueta	*person*	*indoor*	*drawing*	*clothing*	*painting*
	Score (%)	*93,4*	*86,9*	*84,1*	*82,61*	*72,11*
	Similaridad (%)	*0,0*	*0*	*0,0*	*0,0*	*0,0*
Clarifai	Etiqueta	*people*	*painting*	*child*	*group*	*room*
	Score (%)	*99,2*	*98,64*	*98,43*	*96,88*	*96,81*
	Similaridad (%)	*0*	*0,0*	*0,0*	*0,0*	*0,0*
Google	Etiqueta	*child*	*play*	*adaptation*	*visual arts*	*art*
	Score (%)	*72,77*	*71,4*	*66,97*	*59,32*	*50,22*
	Similaridad (%)	*0,0*	*0*	*0,0*	*0*	*0,0*
Watson	Etiqueta	*indoors*	*day care center*	*bedroom*	*creche*	*person*
	Score (%)	*73,3*	*73,2*	*73,2*	*63,6*	*59,6*
	Similaridad (%)	*0,0*	*0,0*	*0,0*	*0*	*0,0*

Tabla 7.11: Resultados para la peor imagen de *SUN*

7.4.4. Conclusión

Una vez conocidas las imágenes con resultados extremos, analizo las posibles causas de los peores resultados:

- Limitación inherente a `NLTK`, como en el caso de *heliport* y *helicopter* (Sección 7.4.2), el caso de *convenience store* y *grocery store* (Sección 7.4.1) o el caso de *art school* y *art* (Sección 7.4.3).
- *Ground truth* demasiado específico, como en el caso de *deli* (Sección 7.4.1).
- *Ground truth* poco relacionado con el contenido de la imagen, como en el caso de *art school* (Sección 7.4.3).

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

Una vez obtenidos los resultados de los distintos análisis realizados; recojo los hechos que resultan más relevantes o generales:

- Los resultados generales son poco esperanzadores, pues la similaridad entre las etiquetas generadas con la mejor confianza y el *ground truth* es del 27,52 % de media (Tabla 7.1) y nula para el 38 % de las imágenes (Tabla 7.4). Esto puede deberse en parte a una limitación inherente al uso de `WordNet` y `NLTK`, como se observa claramente en los distintos casos de la Sección 7.4.

- La etiqueta de confianza máxima para cada imagen apenas resulta más correcta que el resto, pues los porcentaje de etiquetas con similaridad con el *ground truth* nula y plena sólo varian un 1 % y un 2 % respectivamente entre dichos conjuntos (Tablas 7.4 y 7.5).

- El *dataset* empleado influye bastante en las etiquetas generadas:

  - Para Indoor, la similaridad de dichas etiquetas con el *ground truth* resulta un 150 % de la media para los otros *datasets* (Tabla 7.1).
  - Para Indoor, las etiquetas de similaridad con el *ground truth* nula suponen el 64 % de la media para los otros *datasets* (Tabla 7.5).
  - Para Places, el número de etiquetas distintas generadas es un 62 % de la media para los otros *datasets* (Tabla 6.2).

- Atendiendo a la similaridad entre las etiquetas generadas por cada *vendor* y el *ground truth*, resulta el siguiente *ranking* (Tabla 7.1):

  1. Google (33,34 %).

2. Watson (32,67 %). Genera un 44 % de etiquetas distintas más que Google (Tabla 6.2) y sus etiquetas de similaridad con el *ground truth* plena son un 234 % de la media del resto de *vendors* (Tabla 7.4).
3. Amazon (27,96 %). Sus etiquetas de similaridad con el *ground truth* nula resultan un 23 % de la media del resto de *vendors* (Tabla 7.4).
4. Clarifai (26,82 %).
5. Azure (16,85 %). La similaridad de las etiquetas que genera con las generadas por otros *vendors* es un 78 % de la media de los otros *vendors* (Tabla 7.2).

8.2. Trabajo futuro

El objetivo del trabajo era realizar una comparativa de distintos sistemas de etiquetado de imagen; para lo cual sólo restaría estudiar soluciones a la falta de precisión que `NLTK` muestra en algunas ocasiones, pues devuelve similaridades nulas para ciertas palabras que son similares.

Pero si se quisiese desarrollar un sistema de etiquetado basándose en las etiquetas generadas por los distintos *vendors* -no devolver simplemente las etiquetas generadas por Google- habría varias direcciones en las que trabajar:

8.2.1. Normalización de los *scores*

Debido a que las distribuciones de los *scores* de los *vendor* no son semejantes, no podemos comparar un determinado *score* de un *vendor* con uno de otro *vendor* distinto. Si los normalizase, podría escoger la etiqueta con mayor *score* normalizado de entre todas las generadas por los distintos *vendors* para una misma imagen.

Como los *vendors* pueden incluir o modificar sin advertir de ello algoritmos para «retocar» los *scores* que devuelven; habría que revisar periódicamente las distribuciones de los *scores* obtenidos.

8.2.2. Análisis semántico de las etiquetas obtenidas

Mediante la información semántica de las etiquetas obtenidas para una imagen se pueden generar nuevas etiquetas o seleccionar cual se devuelve. Como ejemplo; si tres *vendors* han reconocido «león» o «tigre» con *scores* del 80 % y un *vendor* ha reconocido «naturaleza» con un *score* del 90 %, lo correcto sería devolver «felino» (el hipónimo común a «tigre» y «león» más específico) y no «naturaleza».

8.2.3. Análisis de la precisión de los *vendors* según el tipo de escena

Combinando dichos datos con la información semántica de cada una de las etiquetas obtenidas para una imagen, se pueden infravalorar o sobrevalorar las etiquetas de ciertos *vendors*. Como ejemplo; si un determinado *vendor* genera etiquetas muy precisas para imágenes de negocios y en una imagen varios *vendors* reconocen «negocio», habría que sobrevalorar las etiquetas generadas por dicho *vendor*, o infravalorarlas en casos contrarios.

Habría que tener en cuenta que los *vendors* entrenan constantemente sus modelos y no advierten de ello; por lo que habría que revisar periódicamente su precisión para las distintas escenas.

Bibliografía

[1] Lawrence Roberts. «Machine perception of three-dimensional solids». En: *Massachusetts Institute of Technology* (ene. de 1963), https://dspace.mit.edu/handle/1721.1/11589.

[2] Seymour Papert. «The Summer Vision Project». En: *MIT AI Memos* (jul. de 1966), http://hdl.handle.net/1721.1/6125.

[3] Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco-Hernandez, Lenka Krpalkova, Daniel Riordan y Joseph Walsh. «Deep Learning vs. Traditional Computer Vision». En: *arXiv e-prints* (oct. de 2019), arXiv:1910.13796.

[4] *The Neural Basis of Visual Perception*. Accedido el 9/6/2020. URL: `http://centennial.rucares.org/index.php?page=Neural_Basis_Visual_Perception`.

[5] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H.Freeman, 1982.

[6] Kunihiko Fukushima. «Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position (traslated from Japanese)». En: (abr. de 1980), https://www.rctn.org/bruno/public/papers/Fukushima19

[7] *LeNet-5, convolutional neural networks*. Accedido el 14/3/2020. URL: `http://yann.lecun.com/exdb/lenet/`.

[8] *ImageNet Leaderboard*. Accedido el 14/3/2020. URL: `https://paperswithcode.com/sota/image-classification-on-imagenet`.

[9] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat y Barbara Plank. «Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures». En: *arXiv e-prints* (ene. de 2016), arXiv:1601.03896.

[10] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko y Trevor Darrell. «Long-term Recurrent Convolutional Networks for Visual Recognition and Description». En: *arXiv e-prints* (nov. de 2014), arXiv:1411.4389.

[11] Oriol Vinyals, Alexander Toshev, Samy Bengio y Dumitru Erhan. «Show and Tell: A Neural Image Caption Generator». En: *arXiv e-prints* (nov. de 2014), arXiv:1411.4555.

[12] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel y Yoshua Bengio. «Show, Attend and Tell: Neural Image Caption Generation with Visual Attention». En: *arXiv e-prints* (feb. de 2015), arXiv:1502.03044.

[13] Xu Jia, Efstratios Gavves, Fernando Basura y Tinne Tuytelaars. «Guiding Long-Short Term Memory for Image Caption Generation». En: *arXiv e-prints* (sep. de 2015), arXiv:1509.04942.

[14] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu y Tong Zhang. «Recurrent Fusion Network for Image Captioning». En: *arXiv e-prints* (jul. de 2018), arXiv:1807.09986.

[15] Jyoti Aneja, Aditya Deshpande y Alexander Schwing. «Convolutional Image Captioning». En: *arXiv e-prints* (nov. de 2017), arXiv:1711.09151.

[16] Qingzhong Wang y Antoni B. Chan. «CNN+CNN: Convolutional Decoders for Image Captioning». En: *arXiv e-prints* (mayo de 2018), arXiv:1805.09019.

[17] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao y Tao Mei. «Pointing Novel Objects in Image Captioning». En: *arXiv e-prints* (abr. de 2019), arXiv:1904.11251.

[18] Yang Feng, Lin Ma, Wei Liu y Jiebo Luo. «Unsupervised Image Captioning». En: *arXiv e-prints* (nov. de 2018), arXiv:1811.10787.

[19] Xiang Ma, Zhaohong Deng, Peng Xu, Kup-Sze Choi y Shitong Wang. «Deep Image Feature Learning with Fuzzy Rules». En: *arXiv e-prints* (mayo de 2019), arXiv:1905.10575.

[20] Ebrahim Karami, Siva Prasad y Mohamed Shehata. «Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images». En: *arXiv e-prints* (oct. de 2017), arXiv:1710.02726.

[21] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X. Gao y Dazhong Wu. «Deep learning for smart manufacturing: Methods and applications». En: *Journal of Manufacturing Systems* (jul. de 2018). URL: `https://docs.wixstatic.com/ugd/af3c33_e49547dc92ad408eaef72da5dad6a001.pdf`.

[22] Alejandro Correa Bahnsen. *Building AI Applications Using Deep Learning*. Accedido el 5/9/2019. URL: `https://blog.easysol.net/building-ai-applications`.

[23] *Everything you need to know about Neural Networks and Backpropagation*. Accedido el 25/6/2020. URL: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a.

[24] *Using a VNN architecture*. Accedido el 29/6/2020. URL: https://amiralavi.net/blog/2018/07/29/vnn-implementation.

[25] Fei-Fei Li, Justin Johnson y Serena Yeung. *CS231n: Convolutional Neural Networks for Visual Recognition (Lecture 1)*. Accedido el 5/9/2019. URL: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture1.pdf.

[26] *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Accedido el 13/6/2020. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[27] Bhiksha Raj. *Deep Neural Networks (Lecture 8)*. Accedido el 5/9/2019. URL: https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Fall.2015/slides/lec8.CNN.pdf.

[28] *Deep Learning in a Nutshell: Core Concepts*. Accedido el 29/6/2020. URL: https://developer.nvidia.com/blog/deep-learning-nutshell-core-concepts/.

[29] Steven Bird, Ewan Klein y Edward Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. URL: https://www.nltk.org/book/ch02.html.

[30] Loic Vial, Benjamin Lecouteux y Didier Schwab. «Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation». En: *arXiv e-prints* (mayo de 2019), arXiv:1905.05677.

[31] Dou Goodman y Tao Wei. «Cloud-based Image Classification Service Is Not Robust To Simple Transformations: A Forgotten Battlefield». En: *arXiv e-prints* (jun. de 2019), arXiv:1906.07997.

[32] Alex Cummaudo, Rajesh Vasa, John Grundy, Mohamed Abdelrazek y Andrew Cain. «Losing Confidence in Quality: Unspoken Evolution of Computer Vision Services». En: *arXiv e-prints* (jun. de 2019), arXiv:1906.07328.

[33] Terrance DeVries, Ishan Misra, Changhan Wang y Laurens van der Maaten. «Does Object Recognition Work for Everyone?» En: *arXiv e-prints* (jun. de 2019), arXiv:1906.02659.

[34] *Amazon Rekognition FAQs*. Accedido el 13/6/2020. URL: https://aws.amazon.com/rekognition/faqs/?nc1=h_ls.

[35] *Overview: Visual Recognition in Watson Studio*. Accedido el 13/6/2020. URL: https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/visual-recognition-overview.html.

[36] *HOW DOES CLARIFAI'S COMPUTER VISION TECHNOLOGY WORK?* Accedido el 13/6/2020. URL: https://www.clarifai.com/technology.

[37] *AutoML API for Vision Classification*. Accedido el 13/6/2020. URL: https://aihub.cloud.google.com/u/0/p/products%2Ffd607928-12f3-4aa1-a523-ad4431a96ed6.

[38] *The Forrester New Wave^{TM}: Computer Vision Platforms, Q4 2019*. Accedido el 4/7/2020. URL: https://reprints.forrester.com/#/assets/2/108/RES144576/reports.

[39] *Amazon Rekognition Developer Guide*. Accedido el 5/9/2019. URL: https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html.

[40] *Clarifai Developer Guide*. Accedido el 5/9/2019. URL: http://developer-dev.clarifai.com/developer/guide.

[41] *Computer Vision API Documentation*. Accedido el 5/9/2019. URL: https://docs.microsoft.com/bs-latn-ba/azure/cognitive-services/computer-vision.

[42] *Google Cloud Vision Documentation*. Accedido el 5/9/2019. URL: https://cloud.google.com/vision/docs.

[43] *IBM Watson Visual Recognition API Documentation*. Accedido el 5/9/2019. URL: https://cloud.ibm.com/apidocs/visual-recognition.

[44] Princeton University. *WordNet. A Lexical Database for English*. Accedido el 2/11/2019. URL: https://wordnet.princeton.edu/.

[45] *NLTK WordNet Interface How To*. Accedido el 30/10/2019. URL: http://www.nltk.org/howto/wordnet.html.

[46] *NLTK 3.4.5 documentation (nltk.corpus.reader.wordnet module)*. Accedido el 30/10/2019. URL: http://www.nltk.org/api/nltk.corpus.reader.html#module-nltk.corpus.reader.wordnet.

[47] Zili Zhou, Yanna Wang y Junzhong Gu. «A New Model of Information Content for Semantic Similarity in WordNet». En: *International Conference on Future Generation Communication and Networking Symposia* (dic. de 2008), arXiv:1905.10575.

[48] *NLTK Corpora*. Accedido el 2/11/2019. URL: http://www.nltk.org/nltk_data.

[49] *Computer vision image datasets*. Accedido el 15/6/2020. URL: https://www.cs.utexas.edu/~grauman/courses/spring2008/datasets.htm.

[50] *Image Datasets for Computer Vision Training*. Accedido el 15/6/2020. URL: https://lionbridge.ai/datasets/20-best-image-datasets-for-computer-vision/.

[51] *50 free Machine Learning Datasets: Image Datasets*. Accedido el 15/6/2020. URL: https://blog.cambridgespark.com/50-free-machine-learning-datasets-image-datasets-241852b03b49.

[52] *List of datasets for machine-learning research*. Accedido el 15/6/2020. URL: https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research.

[53] *Top 10 Open Image Datasets for Machine Learning Research*. Accedido el 15/6/2020. URL: https://medium.com/playment/top-10-open-image-datasets-for-machine-learning-research-93ab9c18bed1.

[54] *Cambridge English Dictionary*. Accedido el 6/7/2020. URL: https://dictionary.cambridge.org/dictionary/english.

[55] *Google Cloud Compute Engine pricing*. Accedido el 8/7/2020. URL: https://cloud.google.com/compute/all-pricing.

[56] Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994. URL: https://www.nltk.org/book/ch02.html.

[57] *Adapter Pattern*. Accedido el 4/7/2020. URL: https://www.gofpatterns.com/structural-design-patterns/structural-patterns/adapter-pattern.php.

Apéndices

Apéndice A

Planificación

En esta sección se exponen las tareas que supone la realización de este trabajo; se analizan los costes, riesgos y recursos necesarios y se realiza la planificación y seguimiento de la ejecución del proyecto.

A.1. Recursos necesarios

En este apartado recojo los recursos que resultan necesarios para la realización de este trabajo, que agrupo en tres categorías:

- Humanos, siendo yo el único desarrollador.
- Máquinas para el desarrollo y la ejecución del experimento. Mientras que para el desarrollo emplearé mi ordenador personal; la implementación de las soluciones de etiquetado posiblemente requiera una mayor potencia de cálculo y un tiempo de ejecución alargado, por lo que emplearé una máquina virtual.
- Herramientas necesarias para la realización del experimento:
    - Lenguaje de programación. Debe soportar herramientas de cálculo científico y generación de gráficos.
    - Soluciones de etiquetado de imagen.
    - Herramienta de análisis semántico.
    - Entorno de desarrollo.
    - Repositorio de código, tanto para el código como para la documentación.
    - Herramienta de composición de textos.
    - Herramienta de creación de diagramas.

A.2. Análisis de costes

Una vez conocidos los recursos que este trabajo requiere, queda analizar el coste asociado a estos.

A.2.1. Herramientas necesarias

Debido a que para todas las herramientas necesarias existen alternativas gratuitas, emplearé estas (Apéndice B), por lo que el coste asociado es nulo.

A.2.2. Entorno de ejecución

Si la implementación de las soluciones de etiquetado requiriese reentrenar modelos o realizar inferencias con modelos ya entrenados, resultaría necesaria una potencia de cálculo grande.

Múltiples plataformas de computación en la nube proporcionan cŕeditos iniciales gratuitos, por lo que compruebo si estos permiten el uso de una máquina de suficiente suficiente.

En concreto, calculo el coste en Google Cloud de una máquina virtual `c2-standard-16`, con las siguientes características:

- 16 vCPUs.
- 64GB de RAM.
- 2 GPUs NVIDIA TESLA T4 (16 GB GDDR6 y núcleos `Tensor`).

Dicha máquina es de una potencia muy notable y su coste por hora de ejecución es de 6,67$ [55]. Dado que Google Cloud proporciona un cŕedito de 300$, esto me permite emplear está maquina durante 44 horas, lo que seguramente resulte suficiente para el desarrollo del presente trabajo.

A.2.3. Salario del desarrollador

A continuación calculo el coste que supondría el desarrollo del presente trabajo por un desarrollador por cuenta propia con menos de un año de experiencia laboral, el cual recibe una retribución bruta de 18 euros por hora trabajada.

Teniendo en cuenta que para el Grado en Ingeniería Informática de la Universidad de Valladolid, el Trabajo de Fin de Grado supone 12 créditos ECTS y que cada uno se corresponde con 25 horas de dedicación, la dedicación total requerida por dicho trabajo son 300 horas.

Por lo tanto, dichas 300 horas de dedicación supondrían una retribución total de 5.400 euros.

A.2.4. Total

Debido a que el único coste de este trabajo es la retribución del desarrollador, el coste total del presente proyecto sería de 5.400 euros.

A.3. Análisis de riesgos

A continuación analizo los riesgos que pueden surgir durante la realización de este trabajo; identificándolos, describiéndolos y estableciendo para cada uno de ellos:

- Probabilidad de que suceda.
- Impacto en caso de producirse.
- Plan de mitigación para prevenir que ocurra o reducir su impacto.
- Plan de contingencia en caso de que se produjese.

A.3.1. Fallo en la planificación

Identificador	R1 - Fallo en la planificación
Descripción	El tiempo restante no resulta suficiente para el desarrollo de las tareas pendientes
Probabilidad	Media
Impacto	Alto
Plan de mitigación	Valorar el progreso de las tareas en desarrollo para detectar lo antes posible desvios sobre la planificación
Plan de contingencia	Revisar y reevaluar las tareas pendientes, priorizarlas y planificar de nuevo

Tabla A.1: Riesgo 1: Fallo en la planificación

A.3.2. Indisponibilidad del desarrollador

Identificador	R2 - Indisponibilidad del desarrollador
Descripción	La situación laboral del desarrollador cambia y la cantidad de tiempo que puede dedicar al desarrollo del trabajo se reduce
Probabilidad	Media
Impacto	Alto
Plan de mitigación	Dedicar los días no laborables a la realización el trabajo
Plan de contingencia	Revisar las tareas pendientes y planificar sobre la nueva franja de tiempo disponible

Tabla A.2: Riesgo 2: Indisponibilidad del desarrollador

A.3.3. Conocimientos insuficientes

Identificador	R3 - Conocimientos insuficientes
Descripción	Las tecnologías necesarias para el desarrollo del trabajo requieren de conocimientos cuya obtención escapa de la planificación
Probabilidad	Baja
Impacto	Medio
Plan de mitigación	Comprobar previamente la complejidad de dichos conocimientos
Plan de contingencia	Emplear tecnologías alternativas y planificar de nuevo

Tabla A.3: Riesgo 3: Conocimientos insuficientes

A.3.4. Sobrecostes del entorno de ejecución

Identificador	R4 - Sobrecostes del entorno de ejecución
Descripción	Se producen sobrecostes asociados al entorno de ejecución
Probabilidad	Media
Impacto	Media
Plan de mitigación	Comprobar el coste de forma periódica y frecuente
Plan de contingencia	Buscar entornos de ejcución o tecnologías alternativas y planificar de nuevo

Tabla A.4: Riesgo 4: Sobrecostes del entorno de ejecución

A.3.5. Falta de calidad de los resultados

Identificador	R5 - Falta de calidad de los resultados
Descripción	Los resultados generados no cuentan con la calidad esperada
Probabilidad	Media
Impacto	Medio
Plan de mitigación	Comprobar previamente a su generación si son adecuados
Plan de contingencia	Volver a obtener estos y planificar de nuevo

Tabla A.5: Riesgo 5: Falta de calidad en los resultados

A.3.6. Pérdida de código o documentación

Identificador	R6 - Pérdida de código o documentación
Descripción	Se produce la pérdida de los artefactos resultado de las tareas ya completadas
Probabilidad	Baja
Impacto	Muy alto
Plan de mitigación	Subir los artefactos al servidor de código frecuentemente y realizar copias de seguridad periódicas
Plan de contingencia	Planificar de nuevo

Tabla A.6: Riesgo 6: Pérdida de código o documentación

A.4. Desglose de tareas

A continuación recojo las distintas tareas que supone la realización de este trabajo:

1. Preparación de la memoria: estudio de los requisitos de formato y creación de la base para la posterior documentación.

2. Planificación:

   a) Análisis de los recursos necesarios.

   b) Análisis de costes.

   c) Análisis de riesgos.

   d) Desglose de tareas.

   e) Planificación y elaboración de tabla de seguimiento.

3. Documentación de la motivación y los objetivos.
4. Estudio del estado de la cuestión.
5. Estudio de los fundamentos teóricos aplicables al etiquetado de imagen.
6. Estudio de los fundamentos teóricos aplicables al análisis semántico.
7. Estudio, selección e implementación de soluciones de etiquetado de imagen.
8. Estudio y selección de herramientas para el análisis semántico.
9. Estudio, selección y obtención del conjunto de datos a etiquetar.
10. Etiquetado de dicho conjunto de datos con las soluciones de etiquetados seleccionadas.
11. Obtención de los resultados.
12. Representación de los resultados obtenidos, tanto como tablas o gŕaficas.
13. Análisis de los resultados obtenidos.
14. Documentación de la conclusión y el trabajo futuro.
15. Revisión final de la memoria.

A.5. Seguimiento

En la Tabla A.7 registro la estimación inicial de la duración de las tareas descritas en la Sección A.4 junto a su duración real.

Tarea	Estimación (h)	Duración (h)
Preparación de la memoria	1	1
Análisis de recursos	2	1
Análisis de costes	3	2
Análisis de riesgos	4	3
Desglose de tareas	3	2
Planificación	3	3
Motivación y objetivos	2	2
Estudio del estado de la cuestión	15	30
Estudio de fundamentos de etiquetado de imagen	40	30
Estudio de fundamentos de análisis semántico	25	30
Estudio, selección e implementación de soluciones de etiquetado	30	35
Estudio y selección de herramientas para el análisis semántico	25	25
Selección, estudio y obtención del conjunto de datos	30	20
Etiquetado	25	35
Obtención de resultados	45	55
Representación de resultados	15	10
Análisis de resultados	15	25
Documentación de la conclusión y el trabajo futuro	6	5
Revisión final de la memoria	15	15

Tabla A.7: Estimación inicial y duración real de las tareas

Apéndice B

Tecnologías y herramientas empleadas

- Como sistema de control de versiones de tanto el código como la documentación he empleado `Git`. Como servidor he empleado el servidor de **GitLab** alojado por la Escuela de Ingeniería Informática.
- Como entorno de desarrollo he empleado **Visual Studio Code**, desarrollado por Microsoft, libre y de gran popularidad.
- Para desarrollar esta memoria he empleado **Latex**, un sistema de composición de textos ampliamente empleado entre la comunidad científica.
- Para generar los distintos diagramas presentes en esta memoria he empleado **draw.io**.
- Como soluciones de etiquetado de imagen he empleado **Amazon Web Services Rekognition**, **Clarifai**, **Microsoft Azure Computer Vision**, **Google Cloud Vision** e **IBM Watson Visual Recognition**.
- Como lenguaje de programación principal he empleado **Python** por ser un lenguaje en el que tengo experiencia y que dispone de gran cantidad de bibliotecas:
    - **NumPy** (*Numerical Python*), biblioteca de cálculo científico basada en arrays multidimensionales de uso muy extendido.
    - **Matplotlib**, biblioteca para la generación de gráficos muy popular basada en `NumPy`.
    - **Pandas**, biblioteca de manipulación y análisis de datos muy popular que también está basada en `NumPy`. Es muy rápida, pues opera sobre tablas alojadas en memoria y cuenta con secciones críticas implementadas en C,

y potente, pues permite operaciones como uniones entre tablas o consultas sobre varias columnas. Permite importar y exportar fácilmente datos en formato `csv`.

  - **NLTK**, biblioteca de procesamiento de lenguaje natural compuesta por gran cantidad de herramientas que proporciona una forma sencilla de emplear bases de datos léxicas como `WordNet`, así como multitud de *text corpus*.
  - `SDKs` de las herramientas de etiquetado. Tanto Amazon, como Clarifai o IBM disponen de bibliotecas clientes en `Python` que facilitan su uso; para el resto de *vendors*, empleo `APIs REST`.

- El script de descarga de los distintos *datasets* lo he desarrollado en **Bash**.
- Debido a que tanto la descarga de los *datasets* como la obtención de las etiquetas consumía bastante tiempo, empleé una máquina virtual proporcionada por **Google Compute Engine**, también dentro de su crédito gratuto inicial.

Apéndice C

Implementación del experimento

C.1. Arquitectura

Como se observa en la Figura C.1, la arquitectura propuesta consta de cinco elementos principales:

- El *dataset* empleado.
- Un recolector de etiquetas, que etiquete el *dataset* con cada uno de los *vendors*.
- Sistemas *cloud* de los *vendors*.
- Las etiquetas obtenidas.
- Analizador de etiquetas, que recorra las etiquetas y genere los resultados.

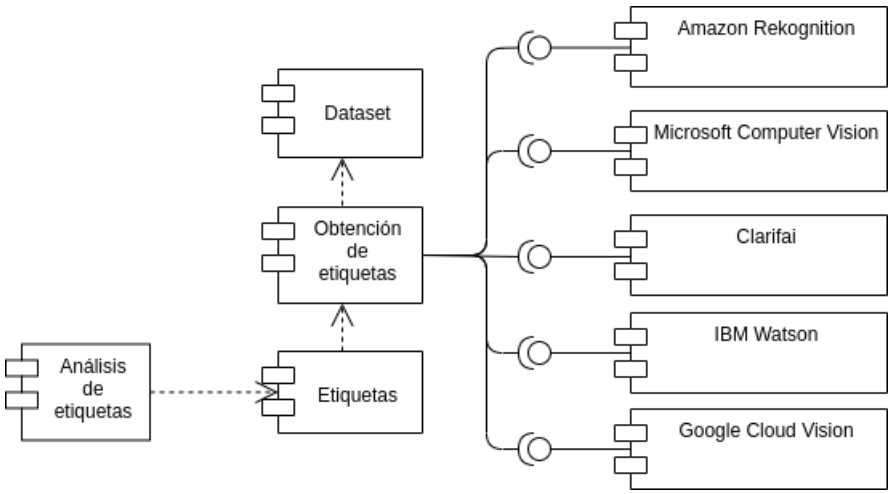


Figura C.1: Diagrama de componentes

C.2. Aplicación de etiquetado

A continuación recojo lo referido a la aplicación de las soluciones de etiquetado al experimento comparativo.

Como este requiere etiquetar el *dataset* con cada API *cloud*, implemento una interfaz común para estas, lo que se conoce como patrón adaptador[56] (Figura C.2). De

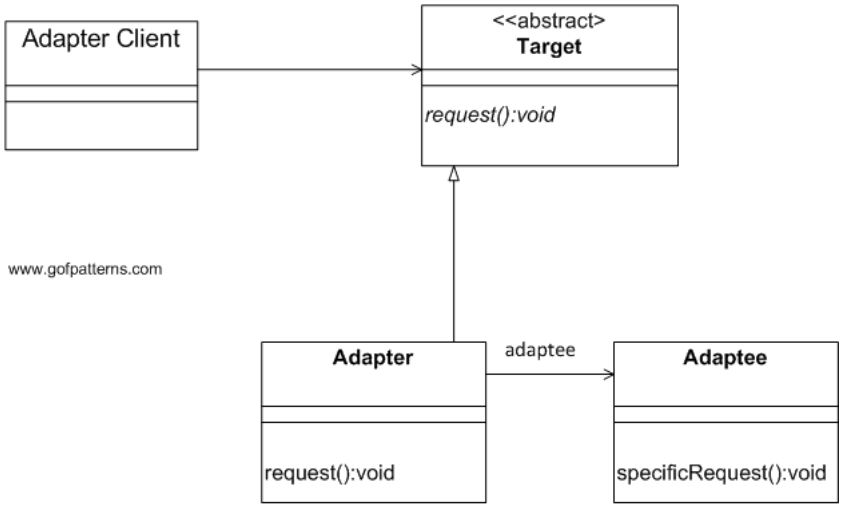


Figura C.2: Patrón adaptador. Tomado de [57].

esta forma, implemento una clase abstracta que recibe la ruta a una imagen en disco y el número de etiquetas a obtener y devuelva un conjunto de pares etiqueta y confianza y para cada *vendor*, un adaptador que implemente dicha clase y realice llamadas a la API correspondiente.

C.3. `APIs` empleadas

Como expongo en el Apéndice B, emplearé el lenguaje de programación `Python`, por lo que dispongo de múltiples alternativas para implementar cada adaptador, de entre las cuales escojo la de implementación más sencilla:

- Amazon Web Services Rekognition: empleo `Boto3`, el `SDK` oficial de AWS para `Python` y su método `detect_labels`, que recibe la imagen en bytes como argumento.
- Microsoft Azure Computer Vision: empleo su `API REST`, de forma que realizo peticiones `POST` contra un endpoint especifico para el método `Tag Image` con las credenciales en el header y la imagen en bytes en el cuerpo de la petición.
- Clarifai: empleo la API para `Python`, y su método `predict`, que toma la imagen en bytes como argumento.

- Google Cloud Vision: empleo su `API REST`, realizo peticiones `POST` al *endpoint* del método `annotate` con la imagen en `Base64` en el cuerpo de la petición.
- IBM Watson Visual Recognition: empleo `Visual Recognition V3`, el `SDK` para `Python`, llamando al método `classify` con la ruta a la imagen como argumento.

C.4. Postprocesamiento de las etiquetas

Tras realizar el etiquetado del *dataset*, compruebo si las etiquetas obtenidas contienen algún caracter no alfabético, lo que sucede en algunas etiquetas generadas por *Watson*:

- *path near canal/river.*
- *refectory (dining hall).*
- *mull (promontories).*
- *alehouse (tavern).*
- *estaminet (cafe).*

Parece que en los casos que contienen paréntesis se generan dos etiquetas: una bastante específica y una más general, que se recoge entre paréntesis. Ante la falta de documentación sobre este comportamiento, empleo como etiqueta el contenido de los paréntesis.

Apéndice D

Manual de instalación y uso

D.1. Requisitos software

Para poder reproducir el experimento es necesario tener instalado:

- El lenguaje de programación `Python`, en concreto con una versión igual o posterior a la 3.6.
- `Pip`, un sistema de gestión de paquetes `Python`.

D.2. Requisitos de las `APIs`

Debido al empleo de `APIs` *cloud*, la reproducción del experimento resulta sencilla. A continuación recojo los pasos a seguir para comenzar a utilizar cada API:

- Amazon
  1. Registrar una cuenta de usuario raíz en su portal web.
  2. Dentro de la *consola*, activar el servicio Rekognition.
  3. Generar una API *key* para dicho servicio, compuesta por un identificador y la API *key* en sí.
- Azure
  1. Registrar una cuenta en su portal web
  2. Crear un nuevo recurso de Computer Vision.
  3. Generar una API *key* y obtener el *endpoint* de dicho recurso.
- Clarifai
  1. Registrar una cuenta en su portal web.

2. Crear una nueva aplicación.
3. Generar una API *key* para dicha aplicación.

- Google

1. Registrar una cuenta en su portal web.
2. Dentro de la *consola*, crear un nuevo proyecto.
3. Dentro de ese proyecto, generar una API *key* para Cloud Vision.

- IBM

1. Registrar una cuenta en su portal web
2. Crear un nuevo proyecto.
3. Obtener una API *key* para el servicio Visual Recognition.

D.3. Ejecución

A continuación recojo el procedimiento a seguir para reproducir el experimento. Por facilidad he incluido en el repositorio de código los *datasets* ya etiquetados, en la carpeta `results`.

1. Clonar el repositorio alojado en GitHub mediante el comando `git clone https://github.com/gdelafuente/tfg-gfp-code.git`.
2. Ubicarnos en la raiz de dicho repositorio, desde donde ejecutaremos el resto de comandos.
3. Ejecutar el comando `pip3 install -r requirements.txt` para instalar las dependencias.
4. Incluir las credenciales obtenidas en el diccionario `CREDENTIALS` ubicado en el fichero `labelling/settings.py`
5. Comprobar que las credenciales son correctas y que las `APIs` funcionan correctamente ejecutando los test de los wrappers, mediante el comando `python3 -m labelling.vendors.test`.
6. Descargar los *datasets*, mediante el comando `sh tools/download_datasets.sh`. Estos se descargan en la carpeta `datasets`.
7. Reorganizar los tres *datasets* y obtener la información presentada en la Sección 5.2 mediante el comando `python3 labelling/prepare_datasets.py`.

8. Etiquetar los tres *datasets* y obtener la información presentada en la Sección 6 mediante el comando `python3 labelling/get_labels.py`. Los ficheros `csv` con las etiquetas generadas se ubican en la carpeta `results`.

9. Obtener la información presentada en la Sección 7 mediante el comando `python3 labelling/compare_vendors.py`.

Apéndice E

Datasets valorados

A continuación recojo los *datasets* cuya utilización he valorado para el desarrollo de la comparativa:

- LabelMe: *dataset* creado por el MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) y disponible en su web. El *training set* está formado por 2.920 imágenes de calles de distintas ciudades españolas que contienen 32.164 instancias de 1.017 clases distintas; mientras que el *test set* lo forman 1.133 imágenes de calles de ciudades de otros países y 32.853 instancias de 838 clases. Tiene dos principales inconvenientes: que bastantes clases cuentan con muy pocas instancias y que cerca de 2.000 imágenes del *training set* no están completamente etiquetadas.

- Lego Bricks: *dataset* formado por 12.700 imágenes de 16 bloques Lego en 400 ángulos distintos obtenidas utilizando Blender. Está disponible en Kaggle.

- ImageNet: *dataset* muy numeroso, formado por 14.197.122 imágenes. Sus 32.326 clases están organizadas según la jerarquía *WordNet* y cada una está presente en al menos 1000 imágenes. Está disponible en su web.

- Microsoft COCO (Common Objects in Context): *dataset* de gran calidad formado por 330.000 imágenes. Este es empleado cada año por sus creadores para organizar competiciones de detección de objetos (un millón y medio de instancias de 80 objetos distintos), descripción de imágenes (5 descripciones por imagen), particionado de elementos (91 categorías) y detección de puntos clave en personas (250.000 personas). Está disponible en su web.

- COIL100 (Columbia University Image Library): *dataset* formado por 7.200 imágenes de 100 objetos distintos obtenidas contra un fondo negro rotándolos 5 grados (72 imágenes por objeto). Está disponible en Kaggle.

- Visual Genome: *dataset* formado por 108.077 imágenes, que contienen 1,3 millones de instancias de 75.729 clases de objetos, 1,6 millones de instancias de 40.513 atributos distintos de estos objetos y 1,5 millones de instancias de 40.480 relaciones distintas entre dichos objetos. Está disponible en su web.

- Google Open Images V5: *dataset* formado por 1,9 millones de imágenes, que contienen 16 millones de instancias pertenecientes a 600 categorías de objetos, 2,7 millones de particiones de 350 categorías de elementos y 391.073 instancias de 329 relaciones distintas entre objetos. Está disponible en su web.

- YouTube-8M: *dataset* formado por 350.000 horas de vídeo, en forma de 6 millones de vídeos almacenados en dicha plataforma asociados a 3.862 clasificadores, estando cada uno presente en 3.552 vídeos de media. Está disponible en su web.

- Labelled Faces in the Wild: *dataset* formado por 13.233 imágenes de caras asociadas al nombre de las 5.749 personas que aparecen en ellas. Está disponible en su web.

- Stanford Dogs Dataset: *dataset* formado 20.580 imágenes de perros procedentes de ImageNet asociadas a sus 120 diferentes razas y segmentadas. Está disponible en su web.

- Places 2: *dataset* orientado a comprensión de escenas a alto nivel y predicción de eventos formado por más de 2 millones de imágenes asociadas a 365 clasificadores de escenas en su version estándar. Está disponible en su web.

- CelebFaces: *dataset* formado por 202.599 imágenes de 10.177 distintas personas famosas, cada una con 40 atributos. Está disponible en su web.

- Flowers: *dataset* formado por 8.189 imágenes de flores y 102 especies distintas distintas, estando cada especie presente en un mínimo de 40 imágenes. su web.

- Home Objects: *dataset* formado por 224 imágenes que contienen distintos objetos del hogar. Está disponible en su web.

- CIFAR-10 y CIFAR-100: *datasets* formados por imágenes 32x32 procedentes de Tiny Images. CIFAR-10 contiene 60.000 imágenes clasificadas en 10 clases disjuntas y CIFAR-100 contiene 600 imágenes de cada una de sus 100 clases. Ambos están disponibles en su web.

- CompCars: *dataset* formado por 136.726 imágenes de 1.716 modelos de coches de 163 marcas distintas. Cada imagen también está etiquetada con cinco atributos como el número de puertas o el tipo de coche. Está disponible en su web.

- Indoor Scene Recognition: *dataset* creado debido a la diferencia de resultados de los modelos con imágenes de exteriores e interiores. Está formado por 15.620 imágenes y 67 clasificadores de escenas, con al menos 100 imágenes por cada uno. Está disponible en su web.

- Visual Question Answering: *dataset* formado por 265.016 imágenes procedentes principalmente de MSCOCO y preguntas sobre las escenas representadas en las imágenes (5,4 por imagen de media). Está disponible en su web.

- SUN (*Scene UNderstanding) Database* está formado por 908 categorías de escenas, tanto interiores como exteriores, y 130.519 imágenes. Pero solamente se encuentra disponible SUN397, un subconjunto formado por 397 categorías y 108.754 imágenes, contando cada categoría con al menos 100 imágenes por categoría. Dispone de una jerarquía de escenas con tres niveles en forma de grafo acíclico dirigido, por lo que una categoría puede estar contenida en más de una categoría. Está disponible en su web.

- Flickr30k: *dataset* formado a partir de 158.915 descripciones a nivel de imagen de 31.783 imágenes procedentes de Flickr. El acceso a este se puede solicitar en su web.

- Caltech-UCSD Birds 200: *dataset* formado por 11.788 imágenes de 200 especies de pájaros. Está disponible en su web.

- Caltech 256: *dataset* formado por 30.607 imágenes de 256 tipos de objetos distintos. Está disponible en su web.

- Cars Dataset: *dataset* formado por 16.185 imágenes de 196 distintos modelos de coches. Está disponible en su web.

Apéndice F

Estudio semántico de los *datasets*

A continuación recojo al completo los datos empleados en el análisis semántico de los *datasets* presentado en 5.2. Para cada *dataset*, recojo:

- Sus categorías sin *synsets* sustantivos asociados, que son descartadas.
- La relación entre las categorías originales y las categorías hiperónimas obtenidas.

Además de lo anterior, para Indoor también recojo el renombrado manual que algunas de sus categorías requerían.

F.1. Indoor Scene Recognition

F.1.1. Categorías renombradas manualmente

airport_inside: airport
artstudio: art_studio
bookstore: book_store
church_inside: church
clothingstore: clothing_store
computerroom: computer_room
dentaloffice: dental_office
gameroom: game_room
grocerystore: grocery_store
hairsalon: hair_salon
hospitalroom: hospital_room
inside_bus: bus
jewelleryshop: jewellery_shop
livingroom: living_room
movietheater: movie_theater
studiomusic: music_studio
poolinside: pool
prisoncell: cell
shoeshop: shoe_shop
stairscase: staircase
toystore: toy_store
trainstation: train_station
videostore: video_store
waitingroom: waiting_room
winecellar: wine_cellar

F.1.2. Categorías sin *synsets* sustantivos

music studio	kindergarden	video store
restaurant kitchen	meeting room	toy store
tv studio	jewellery shop	hair salon
dental office	fastfood restaurant	computer room
children room	book store	art studio
laboratorywet	inside subway	

F.1.3. Categorías obtenidas

Parent category: place of business

office	shoe shop	bakery
casino	deli	grocery store
clothing store		

Parent category: room

bar	locker room	library
living room	lobby	kitchen
bathroom	operating room	classroom
pantry	waiting room	game room
nursery	hospital room	concert hall
bedroom	dining	

Parent category: building

movie theater	cloister	greenhouse
casino	restaurant	garage

Parent category: facility

airport	warehouse	gym
museum	train station	

Parent category: shop

shoe shop	bakery	deli
clothing store		

F.2. Places

F.2.1. Categorías sin *synsets* sustantivos

office cubicles	computer room	bamboo forest
car interior	soccer field	kindergarden classroom
train interior	childs room	archaelogical excavation
forest path	japanese garden	field road
art studio	auto showroom	mountain snowy
parking garage	jewellery shop	desert road
butchers shop	artists loft	zen garden
fabric store	mountain path	topiary garden
fastfood restaurant	water park	restaurant kitchen
ball pit	restaurant patio	veterinarians office
ice cream parlor	legislative chamber	music studio
throne room	airplane cabin	forest road
utility room	martial arts gym	dining hall
ice skating rink	building facade	elevator lobby
television studio	residential neighborhood	jacuzzi
server room	bus interior	construction site
industrial area	natural history museum	underwater
rock arch	banquet hall	

F.2.2. Categorías obtenidas

Parent category: facility

golf course	airport terminal	army base
fire station	botanical garden	racecourse
airfield	gas station	bus station
archive	swimming pool	wind farm
subway station	heliport	train station
science museum	museum	

Parent category: geological formation

coast	grotto	butte
hot spring	iceberg	canyon
cliff	glacier	ski slope
landfill	ice floe	beach
mountain	valley	

Parent category: room

cockpit	bedchamber	playroom
discotheque	clean room	television room
bar	locker room	library
recreation room	beer hall	living room
ballroom	lobby	sauna
kitchen	lecture room	bathroom
conference room	operating room	jail cell
classroom	hotel room	bank vault
pantry	dressing room	waiting room
dorm room	cabin	engine room
phone booth	nursery	hospital room
bedroom	entrance hall	art gallery
storage room	dining room	

Parent category: place of business

office	shoe shop	flea market
bookstore	florist shop	candy store
clothing store	toyshop	general store
bakery	bazaar	pet shop
drugstore	gift shop	ticket booth
shopping mall	department store	supermarket
beauty salon	hardware store	pizzeria
repair shop		

Parent category: way

doorway	staircase	alley
arcade	elevator shaft	driveway
street	crosswalk	highway
boardwalk	catacomb	amusement arcade
raceway	fire escape	aqueduct
corridor		

Parent category: playing field

athletic field	basketball court	volleyball court
football field	baseball field	

Parent category: mercantile establishment

shoe shop	flea market	bookstore
florist shop	candy store	clothing store
toyshop	general store	bakery
bazaar	pet shop	drugstore
gift shop	shopping mall	department store
supermarket	beauty salon	hardware store
pizzeria	repair shop	

Parent category: shop

shoe shop	bookstore	florist shop
candy store	clothing store	toyshop
bakery	bazaar	pet shop
drugstore	gift shop	beauty salon
hardware store	pizzeria	repair shop

Parent category: field

field	corn field	lawn
wheat field	campus	rice paddy

Parent category: building

skyscraper	castle	pagoda
conference center	palace	house
barn	movie theater	synagogue
apartment building	art school	ski resort
cottage	greenhouse	gazebo
cafeteria	coffee shop	bowling alley
boathouse	motel	hospital
restaurant	schoolhouse	inn
beer garden	embassy	garage
stable	oast house	chalet
courthouse	kennel	shed
hotel	pub	temple
office building	beach house	mosque
hunting lodge	home theater	

Parent category: station

airport terminal	fire station	gas station
bus station	wind farm	subway station
train station		

Parent category: terminal

airport terminal	bus station	subway station
train station		

Parent category: bedroom

bedchamber	hotel room	dorm room
nursery	bedroom	

Parent category: house

castle	palace	house
cottage	chalet	beach house
hunting lodge		

Parent category: housing

castle	palace	house
cottage	youth hostel	chalet
beach house	manufactured home	hunting lodge

Parent category: body of water

creek	swimming hole	pond
fishpond	ocean	river
lake	waterfall	lagoon
canal		

Parent category: place of worship

pagoda	synagogue	temple
mosque		

Parent category: platform

loading dock	boat deck	landing deck
pier	boxing ring	

Parent category: garden

formal garden	orchard	roof garden
vegetable garden		

Parent category: lake

swimming hole	pond	fishpond
lake	lagoon	

Parent category: workplace

physics laboratory	chemistry lab	bakery
laundromat	farm	biology laboratory
vineyard		

Parent category: hotel

ski resort	motel	inn
hotel		

Parent category: road

alley	driveway	street
highway		

Parent category: passageway

arcade	elevator shaft	catacomb
amusement arcade	corridor	

Parent category: passage

arcade	elevator shaft	catacomb
amusement arcade	aqueduct	corridor

Parent category: outbuilding

boathouse	garage	kennel
shed		

Parent category: land

tree farm	tundra	islet
snowfield		

F.3. SUN Database

F.3.1. Categorías sin *synsets* sustantivos

batters box
poolroom home
waterfall fan
music store
videostore
forest path
dinette vehicle
cottage garden
covered bridge exterior
waterfall plunge
mountain snowy
jewelry shop
canal natural
elevator interior
fastfood restaurant
restaurant patio
wine cellar bottle storage
subway station platform
music studio
sea cliff
forest road
field wild
building facade
residential neighborhood
limousine interior
bus interior
moat water
skatepark
cubicle office
banquet hall
computer room
car interior backseat
lake natural
dentists office
childs room
theater seats
train station platform
stadium baseball
theater procenium
temple east asia
underwater coral reef
forest broadleaf
waterfall block
topiary garden
restaurant kitchen
wine cellar barrel storage
balcony interior
ice cream parlor
train railway
airplane cabin
dinette home
utility room
television studio
forest needleleaf
car interior frontseat
elevator door
ice skating rink
industrial area
balcony exterior
stadium football
bamboo forest
kindergarden classroom
riding arena
canal urban
temple south asia
gazebo exterior
nuclear power plant
art studio
electrical substation
poolroom establishment
butchers shop
parking garage
field cultivated
ball pit
veterinarians office
jacuzzi
desert vegetation
throne room
van interior
subway interior
martial arts gym
cheese factory
bakery shop
server room
construction site
atrium public
desert sand
rock arch

F.3.2. Categorías obtenidas

Parent category: facility

golf course	airport terminal	fire station
botanical garden	racecourse	swimming pool
gas station	museum	driving range
archive	warehouse	firing range
wind farm	heliport	power plant

Parent category: geological formation

coast	hill	butte
hot spring	iceberg	canyon
cliff	ski slope	sandbar
landfill	ice floe	beach
mountain	valley	

Parent category: room

cockpit	playroom	kitchenette
discotheque	clean room	bar
locker room	recreation room	living room
cabin	ballroom	lobby
sauna	kitchen	lecture room
bathroom	conference room	control room
operating room	anechoic chamber	jail cell
library	classroom	hotel room
pilothouse	pantry	parlor
waiting room	courtroom	dorm room
engine room	phone booth	nursery
hospital room	game room	bedroom
art gallery	dining room	

Parent category: building

cathedral	hunting lodge	skyscraper
castle	pagoda	conference center
cloister	palace	house
barn	inn	mosque
chicken coop	synagogue	art school
kennel	ski lodge	ski resort
pub	apartment building	observatory
cafeteria	coffee shop	bowling alley
boathouse	motel	hospital
bistro	greenhouse	restaurant
schoolhouse	planetarium	movie theater
abbey	basilica	garage
hotel	stable	monastery
oast house	chalet	courthouse
shed	outhouse	office building
casino		

Parent category: place of worship

cathedral	pagoda	mosque
synagogue	abbey	basilica

Parent category: place of business

office	shoe shop	shopping mall
bookstore	thriftshop	candy store
clothing store	florist shop	toyshop
drugstore	bazaar	gift shop
ticket booth	general store	supermarket
beauty salon	casino	

Parent category: house

hunting lodge	castle	cloister
palace	house	monastery
chalet		

Parent category: housing

hunting lodge	castle	cloister
palace	house	youth hostel
monastery	chalet	manufactured home

Parent category: mercantile establishment

shoe shop	shopping mall	bookstore
thriftshop	candy store	clothing store
florist shop	toyshop	drugstore
bazaar	gift shop	general store
supermarket	beauty salon	

Parent category: shop

shoe shop	bookstore	thriftshop
candy store	clothing store	florist shop
toyshop	drugstore	bazaar
gift shop	beauty salon	

Parent category: station

airport terminal	fire station	gas station
wind farm	power plant	

Parent category: body of water

creek	bayou	pond
fishpond	ocean	river

Parent category: plant

auto factory	oil refinery	brewery
factory		

Parent category: way

staircase	arrival gate	doorway
alley	elevator shaft	driveway
street	crosswalk	highway
boardwalk	catacomb	amusement arcade
raceway	fire escape	aqueduct
corridor		

Parent category: field

corn field	wheat field	campus
rice paddy		

Parent category: platform

lido deck	wrestling ring	boat deck
landing deck	pulpit	podium
promenade deck	boxing ring	

Parent category: deck

lido deck	boat deck	landing deck
promenade deck		

Parent category: court

badminton court	squash court	tennis court
basketball court	volleyball court	

Parent category: playing field

badminton court	squash court	tennis court
athletic field	basketball court	volleyball court
baseball field		

Parent category: bridge

bridge	rope bridge	lift bridge
viaduct		

Parent category: natural elevation

hill	butte	sandbar
mountain		

Parent category: hotel

inn	ski lodge	ski resort
motel	hotel	

Parent category: passageway

arrival gate	elevator shaft	catacomb
amusement arcade	corridor	

Parent category: passage

arrival gate	elevator shaft	catacomb
amusement arcade	aqueduct	corridor

Parent category: garden

formal garden	herb garden	orchard
vegetable garden		

Parent category: workplace

physics laboratory	chemistry lab	laundromat
biology laboratory	vineyard	

Parent category: outbuilding

kennel	boathouse	garage
shed	outhouse	

Parent category: road

alley	driveway	street
highway		

Parent category: restaurant

cafeteria	coffee shop	bistro
restaurant		

Parent category: bedroom

hotel room	dorm room	nursery
bedroom		