UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Reconocimiento de estructuras visuales utilizando aprendizaje no supervisado en corpus de galaxias

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Ricardo García García

DIRECTOR DE TESIS

Dr. Ivan Vladimir Meza Ruiz

Ciudad Universitaria, Cd. Mx., 2023

"Espero la vida te ofrezca retos interesantes

y disfrutes resolverlos"

J.F.

Reconocimientos

A la Facultad de Ingeniería y sus profesores que incentivaron mi curiosidad, especialmente a mi amigo y asesor el Dr. Iván Vladimir por ayudarme a encontrar el camino cuando pensaba en cambiarme de carrera, por todo lo que he aprendido de él y su enorme pasión por la investigación.

Gracias a todos los amigos, amores y hermanos que conocí en la Facultad e hicieron esta travesía mucho más placentera. Y sobre todo a mis padres que con su cariño incondicional y su enorme esfuerzo financiaron mis estudios.

Sinceramente, gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El avance tecnológico de las últimas décadas ha incrementado drásticamente la cantidad de información que generamos día con día, disciplinas como el cómputo paralelo, *big data* e inteligencia artificial cobran mayor relevancia y se convierten en el nuevo estándar de cualquier empresa con intenciones de explotar el potencial que ofrecen los datos, en este sentido, la demanda de algoritmos más eficientes se hace una necesidad para procesar los datos en tiempo récord.

Este trabajo presenta un método eficiente para el análisis de imágenes de forma masiva, a través de la extracción de características simples creamos un vocabulario de palabras visuales para agrupar características comunes que puedan ser útiles en el descubrimiento de estructuras visuales más complejas ocultas en una colección de imágenes.

1.1. Motivación

Uno de los problemas más comunes en el campo de la visión computacional es la detección de objetos dónde se han desarrollado numerosas técnicas con un alto nivel de

desempeño y fiabilidad en aplicaciones de reconocimiento en tiempo real (16).

La mayoría de estas herramientas tiene por objetivo la clasificación de objetos conocidos, es decir, a partir de una colección de imágenes etiquetadas se crean los patrones para reconocer los objetos, sin embargo, han sido pocos los esfuerzos por crear métodos capaces de inferir las estructuras embebidas en una colección de imágenes. La identificación de estructuras resulta útil para muchos campos de estudio ya que proporcionan información acerca del comportamiento y características de un objeto, en la astronomía estas estructuras pueden proporcionar información valiosa acerca de la composición, evolución y formación de galaxias.

1.2. Planteamiento del problema

La inmensidad del cosmos y el desarrollo de telescopios cada vez más sofisticado ha hecho impráctico el análisis manual de grandes colecciones de imágenes, de tal forma que es necesario recurrir a métodos de procesamiento automatizados capaces de proporcionar información en tiempos razonables. Los primeros métodos empleados para la detección de objetos se basan en la descripción local de puntos, algoritmos como SIFT (11) realizan la extracción de características basándose en los cambios de contraste, por otra parte, las unidades de procesamiento modernas han dado paso a la creación de modelos de aprendizaje profundo que describen de mejor manera aquello que queremos detectar centrándose en las características de interés del objeto y no solamente en los cambios de contraste. La extracción de características por sí sola resulta poco útil para el estudio astronómico, sin embargo, estas pueden ser empleadas para descubrir los

patrones morfológicos que exhiben las galaxias.

La implementación de técnicas de *Machine Learning* en la segmentación de grandes corpus de galaxias han demostrado buenos resultados (5) (1) (19) (2), sin embargo, son pocos los estudios que tienen como objetivo identificar las estructuras comunes entre diferentes grupos de galaxias sin necesidad de realizar categorizaciones.

El presente trabajo pretende crear un método escalable para la detección de estructuras en galaxias a través de la agrupación de características visuales extraídas a partir del algoritmo SIFT y un modelo de red neuronal conocido como DELF para hacer un análisis comparativo de las estructuras obtenidas empleando distintas técnicas de extracción.

1.3. Objetivos

Desarrollar un método eficiente para la identificación de estructuras visuales en grandes colecciones de imágenes que contribuyan al estudio morfológico de galaxias.

- Ajustar un modelo de red neuronal con atención para la detección, extracción y descripción de características en imágenes de galaxias.
- Construir un *pipeline* para extraer características de imágenes, generar un vocabulario de palabras visuales y minar estructuras utilizando la técnica de "Sampled Min Hashing".
- Describir de forma cuantitativa las estructuras encontradas variando los hiperparámetros del minado.

- Realizar un análisis comparativo de las estructuras obtenidas a través de extractores de características tradicionales y extractores de características basados en aprendizaje profundo.

1.4. Estructura de la tesis

- El primer capítulo explica las motivaciones y una breve introducción al trabajo realizado.
- El segundo capítulo se compone del sustento teórico sobre el cual se diseña el proyecto, un breve repaso por literatura en el campo del procesamiento digital de imágenes, el aprendizaje profundo, la extracción de características en imágenes y la similitud de conjuntos.
- El tercer capítulo esta enfocado en mostrar el flujo de desarrollo del proyecto: el preprocesamiento del corpus, la asignación de categorías el entrenamiento del modelo de red neuronal y el pipeline del minado de estructuras. Así mismo se comentan cuales fueron los retos presentados y la manera en la que fueron solventados para mejorar el desempeño de cada etapa.
- El cuarto capítulo muestra los resultados del procesamiento del corpus, las diferencias entre las dos técnicas de extracción de características y los conjuntos de estructuras en función de los hiperparámetros del minado.
- El quinto capítulo analiza los modelos obtenidos del procesamiento y expone las conclusiones del trabajo realizado así como las oportunidades de trabajo a futuro.

Capítulo 2

Marco teórico

2.1. Procesamiento digital de imágenes

Muchas áreas de estudio se benefician de manera directa o indirecta por el procesamiento digital de imágenes a través de técnicas como adquisición, mejora, restauración, segmentación y extracción de características por mencionar algunas.

Una imagen digital se define como una función bidimensional $I(x, y)$ de valores discretos donde x y y representan coordenadas de un píxel (unidad más pequeña dentro de una imagen) y el valor de la función I corresponde al nivel de intensidad, siendo el modelo de colores (también conocido como sistema de colores) un estándar que especifica la representación de cada color en un píxel.

Algunos de los modelos de colores más comunes en el procesamiento digital de imágenes son:

1. *Escala de grises.* Interpreta los valores en niveles de grises donde el valor más pequeño es interpretado como negro y el valor más alto como blanco. En general se utilizan valores enteros de 0 a 255 representados en 8 bits de memoria.

2. *Modelo RGB.* Representa los colores como una combinación aditiva (la suma de todos es luz blanca) de los componentes espectrales primarios rojo, verde y azul con niveles de intensidad de 8 bits de memoria por cada color [1] pudiendo obtener hasta 16581375 colores distintos.

3. *Modelo CMY.* Representa los colores como una combinación sustractiva (la suma de todos es el color negro) de los componentes espectrales secundarios cian, magenta y amarillo, muy utilizado en la imprenta.

4. *Modelo HSI.* Representa los colores mediante el tono que está relacionado a la longitud de onda del color, la intensidad que mide la sensación de reflectancia de un objeto y la saturación que refiere a la cantidad de luz blanca combinada con el color principal; siendo este modelo el que se asemeja más la manera en la que los humanos interpretan los colores.

Durante el procesamiento digital de imágenes es muy conveniente interpretar una imagen de color dividiéndola en canales. En el modelo RGB cada canal representa la intensidad en rojo, azul y verde separada en distintas imágenes 2.2.

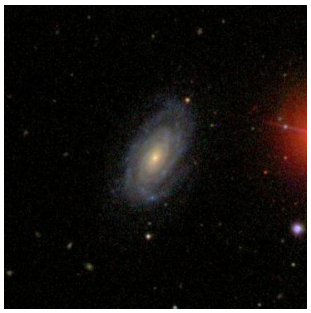
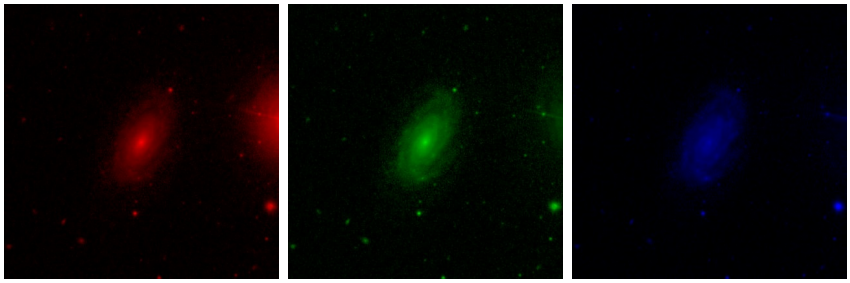


Figura 2.1: Imagen de una galaxia.

[1] Existen modelos RGB que implementan niveles distintos para cada color



(a) Canal rojo. **(b)** Canal Verde. **(c)** Canal azul.

Figura 2.2: División por canales de una imagen.

2.1.1. Filtros espaciales

El filtrado de imágenes es una de la herramientas más útiles en el procesamiento de imágenes ya que permite mejorar secciones específicas resaltando o eliminando componentes de una imagen. Los filtros espaciales modifican una imagen remplazando el valor de un píxel en función de los píxeles vecinos.

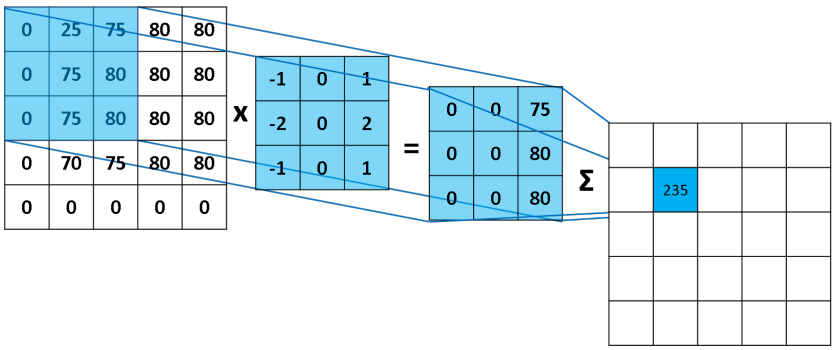


Figura 2.3: Convolución

Matemáticamente el proceso consiste en realizar la convolución en un espacio bidimensional entre la imagen y una matriz denominada *kernel* o filtro. El *kernel* define el

tamaño de la localidad de píxeles considerados para realizar la modificación mientras que los coeficientes de la matriz determinan las características del filtro.

(a) Original

(b) Filtro Gaussiano

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

(c) Kernel

(d) Original

(e) Filtro Prewitt

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

(f) Kernel

(g) Original

(h) Filtro Mediana

$$\begin{bmatrix} 196 & 196 & 185 \\ 196 & 7 & 189 \\ 198 & 189 & 191 \end{bmatrix}$$

$$191$$

(i) Región pixeles y su mediana

Figura 2.4: Algunos filtros comunes en el procesamiento digital de imágenes.

2.1.2. Transformaciones morfológicas

Las transformaciones morfológicas son un conjunto de operaciones que permiten extraer las formas simples embebidas en los objetos a través de la interpretación de imágenes como conjuntos.

Los objetos se definen mediante un grupo de conjuntos asociados a un píxel de la imagen cuyos los elementos corresponden a los píxeles adyacentes, así mismo, las formas simples se definen como conjuntos también denominados elementos estructurantes. Ambos conjuntos pueden verse como regiones cuadradas de tamaño arbitrario que diferencian el primer plano del fondo.

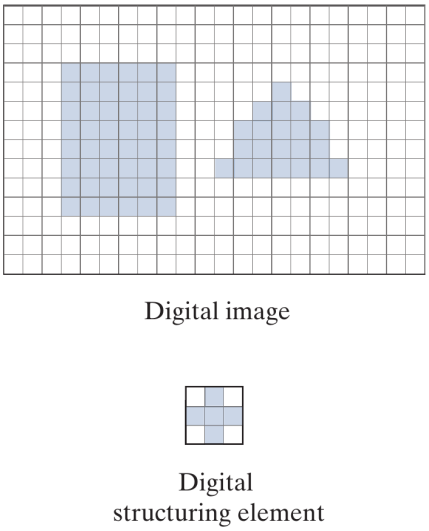


Figura 2.5: Imagen binaria (arriba) y un elemento estructurante (abajo). Imagen tomada de *Digital Image Processing, 2017* (6)

Algunos de los operadores morfológicos más simples son la erosión que consiste en la eliminación de píxeles en los límites del objeto 2.1 y la dilatación que agrega píxeles

a los límites del objeto 2.2. Sea B el uno de los conjuntos que representan a la imagen y E el conjunto que representa el elemento estructurante estas operaciones pueden ser definidas mediante:

$$E \ominus B = \{z|(E)_z \subseteq B\} \tag{2.1}$$

$$E \oplus B = \{z|(E)_z \cap B \neq \varnothing\} \tag{2.2}$$

Donde z son los valores de los píxeles en primer plano de la imagen binaria. En otras palabras la erosión conserva un pixel cuando la superposisión de E sobre B es completa, mientras que la dilatación añade un pixel cuando existe una superposisión parcial de algún elemento de E sobre B.

2.2. Aprendizaje automático

El aprendizaje es el proceso por el cual adquirimos los conocimientos y habilidades necesarios para realizar una tarea. La adquisición de conocimiento puede darse de diferentes formas, por ejemplo, a través de la enseñanza (transmisión de un individuo a otro) o la experiencia (experimentación y análisis causa-efecto).

La manera tradicional en la que una computadora aprende a realizar una tarea es mediante la programación, donde a priori se conoce el algoritmo que resuelve el problema, no obstante, esto no es posible si desconocemos la solución al problema o la complejidad algorítmica impide la implementación en un tiempo razonable, para

ello existe otra metodología denominada aprendizaje automático que programa a las máquinas para que puedan aprender por sí mismas a realizar tareas.

En términos generales, los modelos de aprendizaje automático requieren de una fuente de experiencia para aprender, que comúnmente es especificada como un conjunto de entradas y salidas lo suficientemente representativas como para abarcar todos los escenarios posibles del problema.

Muchos de estos modelos utilizan un conjunto de parámetros para calcular la respuesta, los parámetros son inicializados de forma aleatoria y van siendo actualizados a lo largo de un proceso de entrenamiento. La fase de entrenamiento consiste en pasarle una entrada al modelo cuya salida es conocida para medir el error de cálculo permitiendo modificar los parámetros con el objetivo de minimizar el error.

El aprendizaje automático puede clasificarse en diversas categorías en función de cómo adquiere el conocimiento, entre las cuales tenemos:

1. *Aprendizaje supervisado.* Requiere un conjunto de entrenamiento con entradas y respuestas, comúnmente llamados datos etiquetados, es la técnica más empleada por cualquier tipo de algoritmo pues solo necesita los datos de entrada y el valor esperado del sistema. Algunas de la tareas más usuales en esta área es la clasificación que utiliza un conjunto de datos como imágenes, vectores numéricos para asignar una categoría o la regresión que busca calcular valores numéricos a partir de un conjunto de características.

2. *Aprendizaje no supervisado.* Emplea conjuntos de datos sin etiquetas donde no

se conoce ningún valor objetivo categórico o numérico permitiendo al algoritmo aprender de los datos sin supervisión aprendiendo de las estructuras intrínsecas de los datos. Algunas de las tareas más comunes en este tipo de aprendizaje es la segmentación de grupos que consiste en dividir poblaciones de datos para describir características comunes entre los elementos de cada grupo, la reducción de dimensionalidad para simplificar grandes cantidades de datos agrupando características con un alto nivel de correlación y la detección de anomalías que permite eliminar instancias poco comunes al resto del conjunto de datos.

El conjunto de tareas que es posible realizar empleando aprendizaje automático va desde tareas simples como la clasificación o regresión hasta más sofisticadas como la conducción automática o el reconocimiento de voz, en este sentido, el aprendizaje automático permite resolver problemas que pueden ser demasiado complejos para implementar mediante programación tradicional.

2.2.1. K-Medias

El proceso de segmentar datos conocido como *clustering* permite clasificar y encontrar patrones ocultos en los conjuntos de datos. Los grupos o *clusters* reúnen datos con características similares diferenciados a su vez entre los demás clusters. El algoritmo K-Medias (o *K-Means* por su nombre en inglés) es un método numérico, iterativo y no determinista (13) para segmentar conjuntos de datos. Propuesto en 1967 por MacQueen el algoritmo de aprendizaje no supervisado permite clasificar un conjunto de datos especificando el número k de clusters deseados.

Inicialmente se seleccionan k centroides aleatorios, seguido se calcula la distancia euclidiana asignado cada miembro del conjunto de datos al centroide más cercano. Una vez que cada miembro del conjuntos fue asignado a un centroide se recalcula el valor de los centroides utilizando los elementos asignados a él. Este proceso se repite de forma iterativa hasta que los centroides no se muevan entre cada iteración o se alcance un cierto número de iteraciones.

2.3. Aprendizaje profundo

El aprendizaje profundo corresponde a una categoría del aprendizaje automático capaz de representar funciones de increíble complejidad. Se basa en el comportamiento del cerebro humano modelando redes neuronales artificiales. Gran parte de este ramo aprovecha, aunque no le limita, a utilizar conjuntos de datos etiquetados para calcular funciones de mapeo entre un vector de entrada y un vector de salida.

2.3.1. Redes Neuronales

El origen de las redes neuronales se inspira en el funcionamiento del cerebro, un conjunto de neuronas que intercambian información mediante impulsos eléctricos. La primera aproximación introducida por Warren McCulloc y Wakter Pitts en 1943 utilizaba entradas y salidas binarias para realizar operaciones lógicas simples que podían ser combinadas para calcular expresiones lógicas más complejas (12).

Años más tarde en 1957 Frank Rosenblatt revoluciona el modelo con el *perceptron* cambiando las entradas binarias por valores numéricos (17). El pereptron es la forma

más simple de una red neuronal que sirve para la clasificación de problemas linealmente separables conformado por un conjunto de neuronas denominadas *theshold logic units* (TLU por sus siglas en inglés), cada una conectadas a un vector numérico que representa la entrada 2.6. Una TLU calcula una combinación lineal de los valores de entrada incorporando un bias externo, a este resultado se le aplica una función de activación para obtener la salida. Sea x el vector de entrada, w el vector de pesos de la suma ponderada, b el bias y a la función de activación, el cálculo de una TLU esta dado por:

$$h_{\mathbf{w}}(\mathbf{x}) = a(\mathbf{x}^T\mathbf{w} + b) \tag{2.3}$$

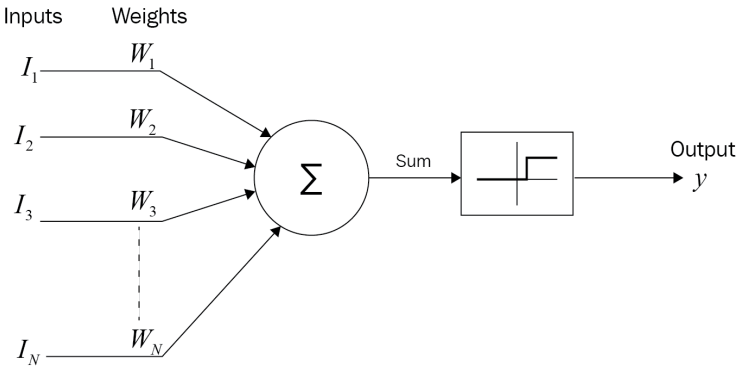


Figura 2.6: Theshold logic unit. Imagen tomada de The McCulloch-Pitts neuron

Un perceptron con una única TLU se encuentra limitado a la clasificación binaria, sin embargo es posible utilizar más de una neurona para clasificar entre más de dos clases, este tipo de redes reciben el nombre de *feedforward neural networks* o *multilayer perceptrons* que consisten en el uso de múltiples capas de TLU's interconectadas 2.7. Todas la conexiones de las neuronas pasan de una capa a otra sin formar ciclos con las capas previas, las capas intermedias también denominadas capas ocultas utilizan los

cálculos de las capas anteriores permitiendo añadir múltiples niveles de profundidad.

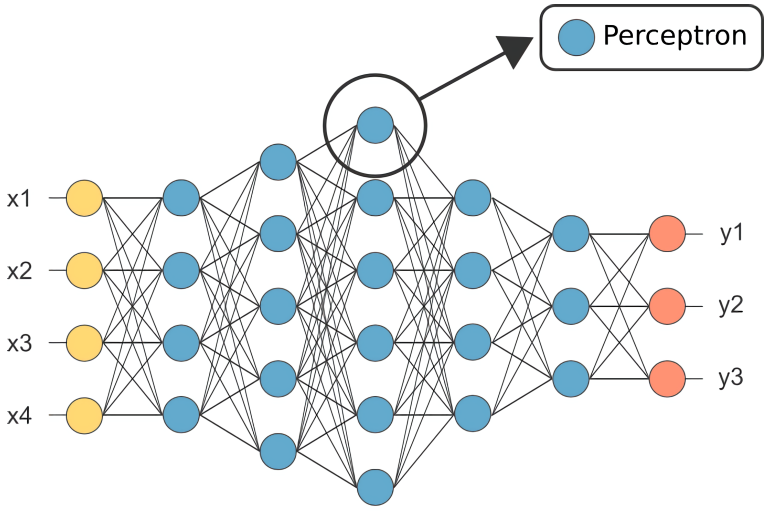


Figura 2.7: Feedforward Neural Network. Imagen tomada de Deep Learning: Feed Forward Neural Networks

El entrenamiento de un red neuronal consiste en aproximar una función $y = f(x)$ mediante el ajuste de los pesos y bias de todas las capas de la red neuronal de tal manera que la salida sea lo más parecida a el valor y. Desde la propuesta del perceptron tardaron algunos años en encontrar un método eficiente para calcular los parámetros óptimos y no fue hasta 1986 que David Rumelhart, Goeyffrey Hinton y Ronald Willims publican el paper "Learning Internal Representacion by Error Propagation"donde introdujeron el método que hoy conocemos como *backpropagation* (18).

El algoritmo de *backpropagation* utiliza un *mini-batch* (conjunto de instancias) para predecir la salida de la red neuronal preservando los resultados intermedios de las capas ocultas, calcula el gradiente de error de cada una de las neuronas a partir del error de la salida aplicando la regla de la cadena sobre las funciones de activación hasta la capa de

entrada. Una vez se obtienen los gradientes de la red se realiza un paso de actualización para ajustar cada uno de los pesos aplicando el método descenso del gradiente.

2.3.2. Redes Neuronales Convoluciones

Las redes neuronales convolucionales (Convolutional Neural Networks, CNN por sus siglas en inglés) son un tipo de redes que afrontan las principales dificultades de procesar objetos en imágenes: la escala, rotación y traslación, desarrolladas por Yann LeCun para el el reconocimiento de letras manuscritas (9), las CNN aplican filtros a través de convoluciones bidimensionales que separan los componentes de las imágenes en bordes y texturas permitiendo abstraer el contenido de una imagen, lo que minimiza la variabilidad a tal punto que es posible clasificar, detectar y segmentar objetos independientemente del contexto en el que se encuentren.

Una CNN se compone de capas especializadas en realizar una operación de convolución y otra de subsampling. En las capas convolucionales se define el tamaño y número de filtros deberán ser aprendidos durante el entrenamiento, mientras que en las capas de subsampling se realiza reducción de las imágenes obtenidas del paso anterior con la finalidad de preservar las características más importantes detectadas por cada filtro y disminuir el procesamiento en las capas subsecuentes.

La arquitectura completa de una CNN se compone pares de capas de convolución y subsampling seguidas de una o más capas fully-conected, las primeras capas convolucionales se encagarán de reconocer características simples como líneas horizontales, verticales o diagonales mientras que las capas más profundas serán capaces de detectar

características más complejas.

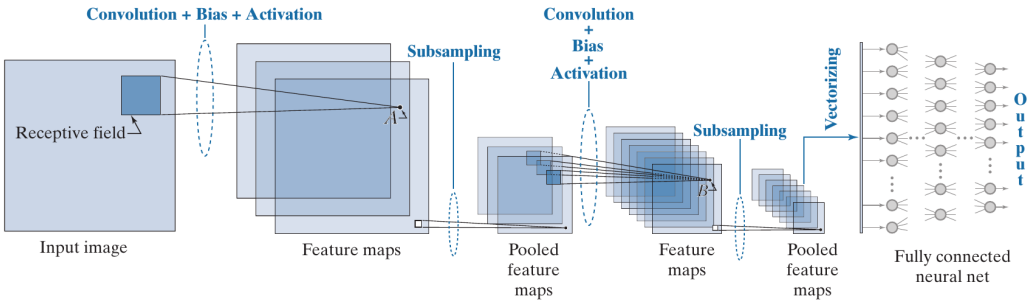


Figura 2.8: Arquitectura de una red neuronal convolucional. Imagen tomada de *Digital Image Processing, 2017* (6)

2.3.3. ResNet-50

Uno de los modelos más innovadores durante *ILSVRC 2015 classification challenge* fue ResNet-50 obteniendo un excelente desempeño en múltiples tareas de detección de imágenes. Introducido en 2015 por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en el artículo de investigación titulado *Deep Residual Learning for Image Recognition* propone una nueva forma de interconectar neuronas capaz de lidiar con el problema del desvanecimiento del gradiente (8).

El desvanecimiento del gradiente es un problema común en el campo del aprendizaje profundo derivado del incremento de capas ocultas en una red neuronal que implementan ciertas funciones de activación que mapean la entrada en un rango reducido de valores, por ejemplo, la función sigmoide cuyos valores de salida varían entre 0 y 1. Dado que un cambio en la entrada produce un pequeño cambio en la salida la derivada se hace todavía más pequeña haciendo que las primeras capas se ajusten lentamente y sean difíciles de entrenar.

La arquitectura de ResNet-50 introduce el aprendizaje residual para facilitar el entrenamiento de las redes permitiendo que sean más profundas. Esta aproximación propone calcular la función residual $F(x) = H(x) - x$ en lugar de la función de mapeo $H(x)$ creando conexiones entre bloques de capas simplemente añadiendo la identidad 2.9, estas conexiones además de demostrar una amplia mejora en el rendimiento no añaden ninguna complejidad computacional adicional o parámetros extras.

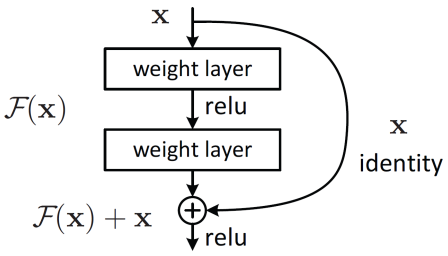


Figura 2.9: Arquitectura de un bloque residual. Imagen tomada de *Deep Residual Learning for Image Recognition, 2015* (8)

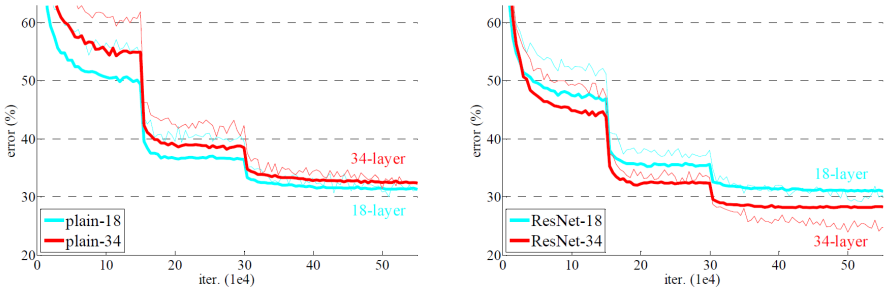


Figura 2.10: Entrenamiento sobre ImageNet: La curva mide el error de entrenamiento, a la izquierda se muestra el desempeño de dos redes con 18 y 34 capas, a la derecha dos redes residuales con 18 y 34 capas. Imagen tomada de *Deep Residual Learning for Image Recognition, 2015* (8)

2.4. Extracción de características

Una característica en el contexto del procesamiento digital de imágenes es un atributo que permite diferenciar entre distintos objetos presentes en una imagen, estas características deben ser independientes de la localización, orientación o escala así como cambios de iluminación, perspectiva o el contexto en el que se encuentre el objeto. La extracción de características se compone de dos pasos: la detección que corresponde al proceso de encontrar los puntos de interés en la imagen y la descripción que se refiere a la asignación cuantitativa de los puntos de interés encontrados. Las características pueden clasificarse como locales o globales en función de su aplicación, es decir, si los descriptores hacen referencia a una región o a la totalidad de la imagen.

2.4.1. SIFT

El algoritmo *Scale Invariant Features Transform*, SIFT desarrollado por David Lowe en 1999 permite transformar una imagen en características locales robustas e invariantes ante un amplio tipo de distorsiones (11). SIFT identifica las coordenadas de los puntos de interés (características) y crea un vector descriptor eliminado la variabilidad en la orientación y escala de cada punto encontrado. La idea fundamental es encontrar en la imagen puntos de interés invariantes a la traslación, escala y rotación que a su vez sean mínimamente afectados por el ruido, esto se realiza mediante la búsqueda características estables a través de todas escalas posibles donde puede aparecer el objeto utilizando filtros de suavizado que simulan la pérdida de detalle al disminuir la resolución.

El algoritmo consiste en aplicar filtros gaussianos sobre la imagen con un valor σ

incremental. Primero se establece un valor inicial de σ que aumentará gradualmente de forma k^n donde k corresponde a un valor constante y n el número de filtro de la iteración hasta que $k^n\sigma \approx 2\sigma$, después se se realiza un submuestreo tomando 1 de cada 2 píxeles en filas y columnas de la última imagen filtrada para repetir el proceso estableciendo el valor de $\sigma = 2\sigma$ (el sigma de la iteración anterior). A cada uno de los grupos de imágenes filtradas se les denomina octavas haciendo una analogía a la teoría musical donde cada nota se repite al duplicar la frecuencia.

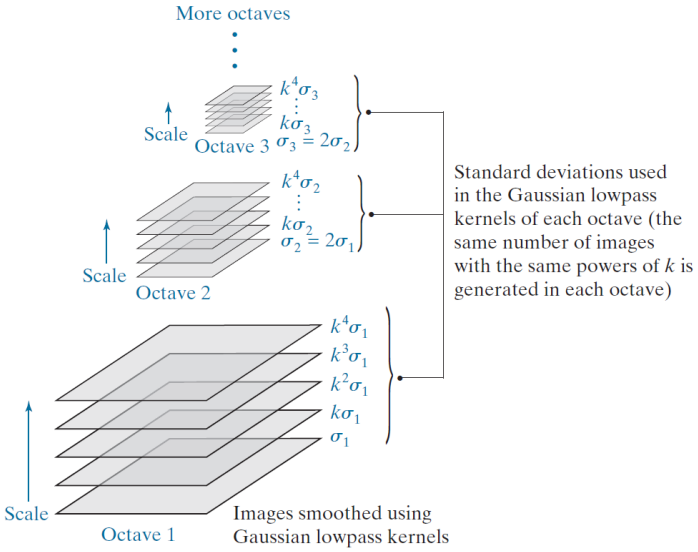


Figura 2.11: Generación de octavas. Imagen tomada de *Digital Image Processing, 2017* (6)

Las imágenes suavizadas sirven para calcular la diferencia gaussiana, empleando dos imágenes adyacentes en una octava se realiza la resta y se convoluciona con la imagen correspondiente a esa octava, este procesamiento no es otra cosa que una aproximación al Laplaciano del Gaussiano empleado para la detección de bordes.

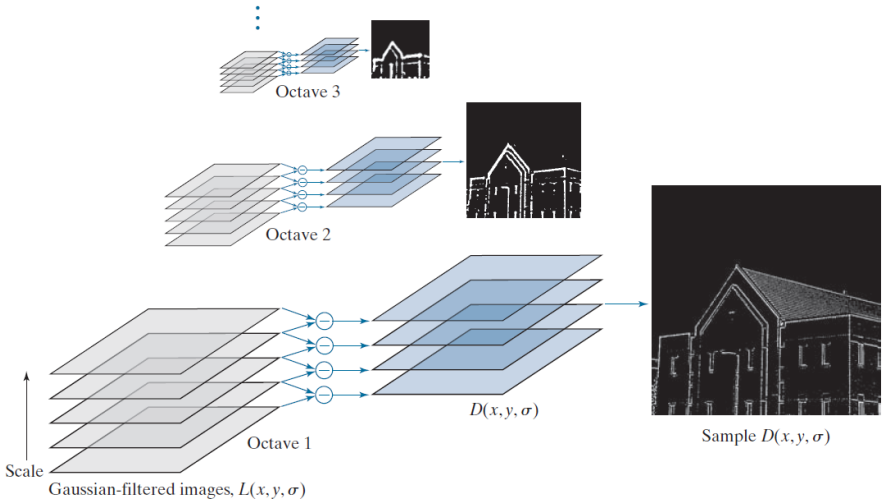


Figura 2.12: Diferencias gaussianas entre distintas octavas. Imagen tomada de *Digital Image Processing, 2017* (6)

La localización de los puntos de interés se realiza detectando los puntos críticos de las diferencias gaussianas, para ello, se comparan los 8 píxeles adjuntos a cada punto así como los 9 píxeles de la diferencia gaussina superior e inferior, un punto será considerado como crítico si su valor es un máximo o mínimo dentro de la vecindad de comparación 2.13.

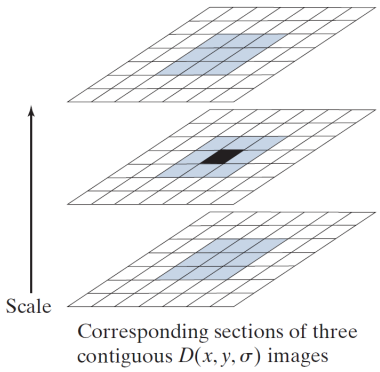


Figura 2.13: Puntos críticos: comparación con diferencias gaussianas. Imagen tomada de *Digital Image Processing, 2017* (6)

El resultado de este proceso produce un gran número de puntos críticos que no serán de utilidad, por esta razón es necesario mejorar la precisión implementando una función de interpolación en cada punto crítico utilizando una expansión en serie de Taylor para descartar aquellos puntos que se encuentren por debajo de un umbral definido.

La aproximación de SIFT se centra en analizar esquinas en lugar de bordes y dado que muchos de los puntos encontrados se localizan en bordes es necesario descartar algunos de ellos, para ello, SIFT implementa un método similar a el detector de esquinas de Harris (7) que calcula un radio de curvatura en cada punto y establece un rango de aceptación para considerar si el punto evaluado es un punto de interés.

La invarianza del método ante cambios de escala deriva de la búsqueda realizada en todo el espacio de escalas posibles, el siguiente paso consiste en asignar una orientación relativa a cada punto de interés basándose en las características locales de la imagen permitiendo que las características sean invariantes a la rotación, para ello se calcula la magnitud M(x, y) y ángulo θ(x, y) del gradiente en cada punto de las imágenes utilizando las diferencias de píxeles 2.4.

$$\begin{aligned} d_x &= L(x+1, y) - L(x-1, y) \\ d_y &= L(x, y+1) - L(x, y-1) \\ M(x, y) &= \sqrt{d_x^2 + d_y^2} \\ \theta(x, y) &= arctan(\frac{d_y}{d_x}) \end{aligned} \tag{2.4}$$

Utilizando las orientaciones se forman histogramas conformados por a la vecindad de cada punto de interés, utilizando un rango de 36 particiones de 10 grados cada uno.

Cada muestra del histograma se pondera por la magnitud del gradiente y una función Gaussiana circular centrada en el punto de interés con desviación estándar igual a 1.5; el máximo del histograma se denomina la dirección dominante o gradiente local del punto de interés y en caso de existir otro máximo con al menos el 80 % del valor del pico más alto se crea otro punto de interés con esa orientación.

En cada punto de interés se utiliza una región de 16x16 para calcular la orientación y magnitud, se crean subregiones de 4x4 y para cada una de estas un histograma utilizando 8 particiones separadas cada 45 grados. Finalmente el descriptor se forma por la concatenación de las 16 histogramas de cada subregión.

2.4.2. Recuperación de imágenes con DELF

Deep Local Feature, DELF es un descriptor de características locales basado en redes neuronales convolucionales con atención creado para la recuperación y reconocimiento de imágenes a gran escala. El modelo desarrollado por Google en 2016 (14) fue entrenado utilizando un conjunto de 1,060,709 imágenes distribuidas entre 12,894 lugares de emblemáticos a través de diferentes locaciones en todo el mundo (20).

La arquitectura de este modelo está conformada por tres etapas, primero la detección y extracción de características que se realiza mediante una red neuronal convolucional que reutiliza parte de la arquitectura y parámetros de ResNet-50, segundo la selección de las características representativas basadas en el contexto y finalmente la reducción de dimensionalidad y normalización de los descriptores empleando PCA 2.15.

Figura 2.14: Ejemplos de lugares emblemáticos empleados para entrenar DELF. Imagen tomada de *Large-Scale Image Retrieval with Attentive Deep Local Features, 2016* (20)

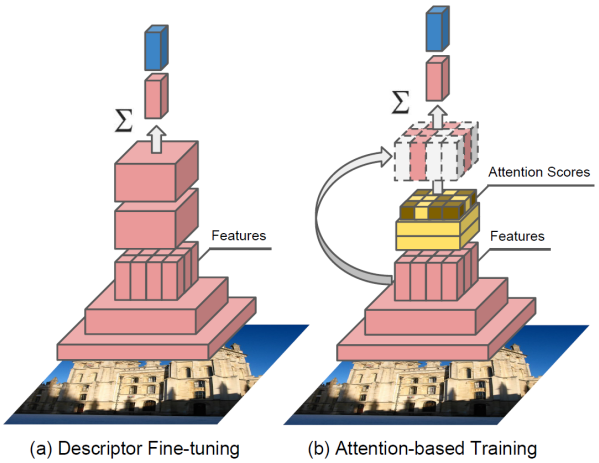


Figura 2.15: Arquitectura del modelo. Imagen tomada de *Large-Scale Image Retrieval with Attentive Deep Local Features, 2016* (20)

Las descriptores se obtienen de los mapas de características en las capas convolucionales, específicamente, DELF utiliza la salida del cuarto bloque convolucional de ResNet-50 mientras que la localización de cada característica se calcula a partir de los campos receptivos de la red. DELF utiliza como base el modelo ResNet-50 reali-

zando un ajuste de parámetros para mejorar la discriminabilidad de los descriptores locales, durante el entrenamiento se emplea una pirámide de imágenes que permite detectar características a diferentes escalas con un conjunto de imágenes etiquetadas correspondientes a lugares emblemáticos. Las principales ventajas de emplear una red convolucional es que los descriptores aprenden de forma implícita cuáles son las representaciones más importantes para el problema planteado.

La selección de puntos de interés es sumamente importante para mejorar el desempeño y precisión en tareas de reconocimiento o clasificación de imágenes; dado que gran parte de las descriptores obtenidos de los mapas de características son poco relevantes dentro del contexto del problema que se quiere resolver es necesario asignar una puntuación a cada característica, para esto, DELF implementa un clasificador de lugares emblemáticos con atención que mide la relevancia de cada punto de interés encontrado.

2.5. Búsqueda por similitud

2.5.1. Min-Hashing

Comparar uno a uno los elementos de dos o más conjuntos requiere de un gran número de operaciones que crecen de forma exponencial con el tamaño de los conjuntos, muchas de estas comparaciones resultan innecesarias ya que no se encontrará similitud entre todos los elementos de los conjuntos. Min Hash es una técnica que permite estimar la similitud de dos conjuntos de manera eficiente utilizando la similitud de Jaccard para comparar los conjuntos.

Sean los conjuntos X_i y X_j la similitud de Jaccard esta dado por:

$$sim(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|} \in [0, 1] \tag{2.5}$$

La similitud de dos conjuntos será mayor en función de la diferencia entre la unión e intersección, sin embargo, realizar estas operaciones continua siendo costoso especialmente si el tamaño de los conjuntos es grande, por esta razón, Min Hash utiliza un algoritmo aleatorio para aproximar la similitud de Jaccard.

Es posible visualizar conjuntos es mediante una matriz de características donde cada columna corresponde a un conjunto y los renglones a los elementos del conjunto universo. En cada celda se escribe el valor 1 para indicar que ese elemento pertenece al conjunto y cero en caso contrario, por ejemplo, sean los conjuntos $X_i = \{x_1, x_2, x_3, x_4\}$, $X_j = \{x_3, x_4, x_5\}$ y $X_k = \{x_2, x_1, x_6\}$, la matriz de características que los representa queda de la siguiente manera:

Elemento	Xi	Xj	Xk
x_1	1	0	1
x_2	1	0	1
x_3	1	1	0
x_4	1	1	0
x_5	0	1	0
x_6	0	0	0

Tabla 2.1: Representación matricial de tres conjuntos

Una función Min-Hash consiste en aplicar una permutación aleatoria π sobre los elementos ordenados de un conjunto y seleccionar el valor mínimo de la permutación,

es decir, el elemento en la posición que contenga el primer uno en el conjunto.

$$h(X_i) = min(\pi(X_i)) \tag{2.6}$$

Elemento	Xi	Xj	Xk
x_3	1	1	0
x_1	1	0	1
x_6	0	0	1
x_5	0	1	0
x_4	1	1	0
x_2	1	0	1

Tabla 2.2: Permutación de los renglones

Una permutación sobre los conjuntos 2.1 es 2.2 calculando los valores mínimos de cada conjunto obtenemos:

$$\begin{aligned} min(\pi(X_i)) &= x_3 \\ min(\pi(X_j)) &= x_3 \\ min(\pi(X_k)) &= x_1 \end{aligned} \tag{2.7}$$

En la práctica aplicar permutaciones aleatorias resulta poco práctico, no obstante es posible obtener un comportamiento similar implementando funciones hash que mapeen los elementos del conjunto a un nuevo dominio que simule las permutaciones de los elementos.

La estimación de la similitud puede realizarse de distintas maneras, utilizando una única función hash π y un valor m se realiza una selección aleatoria de m elementos de

cada conjunto creando subconjuntos de los originales, de esta manera, se aproxima la similitud de Jaccard mediante la fórmula 2.5 aplicada a los subconjuntos. Es importante destacar que entre más grande sea el valor m mejor será la estimación.

En min-hashing la probabilidad de que dos conjuntos X_i y Xj tomen los mismos valores min-hash es igual a la similitud de Jaccard, esto es:

$$P[h(X_i) = h(X_j)] = sim(X_i, X_j) \tag{2.8}$$

Haciendo uso de la tabla 2.2 y la ecuación 2.8 podemos calcular la similitud de Jaccard para cada uno de los conjuntos de la siguiente manera:

$$sim(X_i, X_j) = \begin{cases} 1 \text{ si } h(X_i) = h(X_j) \\ 0 \text{ en cualquier otro caso} \end{cases} \tag{2.9}$$

Para calcular una mejor estimación es posible realizar este cálculo utilizando múltiples funciones hash para ello se aplican k distintas permutaciones para cada conjunto y se realiza el promedio de cada resultado.

2.5.2. Sampled Min-Hashing

Sampled Min-Hashing es un método de aprendizaje no supervisado que permite la extracción de tópicos (15) y la detección estructuras visuales (4) en colecciones de datos a gran escala.

El algoritmo se basa en la hipótesis de que las imágenes contienen características locales que pueden aparecer de forma co-ocurrente representando una misma estructura visual a lo largo de toda la colección, para ello, se construye un vocabulario de palabras

visuales $V = v_1, ..., v_k$ agrupando descriptores de características locales de una colección de imágenes, posteriormente, se representa cada imagen mediante la lista de palabras visuales contenidas en ella.

El descubrimiento de las estructuras visuales se realiza mediante un índice invertido construido a partir del vocabulario V, asociando cada palabra v_i a la lista de imágenes donde aparece $\hat{v_i}$. Este índice sirve para buscar palabras que tienden a estar de forma constante en una misma imagen.

La ocurrencia de dos palabras v_i y v_j en una misma imagen es calculada utilizando la Similitud de Jaccard a través de la técnica de Min-Hashing. El método Sampled Min-Hashing implementa l funciones min-hash elegidas aleatoriamente donde cada función calcula el valor hash concatenando r valores min-hash. A las palabras visuales que caen en el mismo bucket de las tablas hash se les denomina conjuntos de palabras co-ocurrentes ϕ.

Debido a la inestabilidad y polisemia de las palabras visuales los conjuntos ϕ corresponden a particularidades de las estructuras visuales (4), para descubrir objetos más representativos se agrupan los conjuntos ϕ con palabras visuales en común mediante:

$$ovr(\phi_i, \phi_j) = \frac{|\phi_i \cap \phi_j|}{min(|\phi_i|, |\phi_j|)} \in [0, 1] \tag{2.10}$$

Siendo necesario que $ovr(\phi_i, \phi_j) > \epsilon$, para poder unir los conjuntos ϕ_i y ϕ_j al mismo grupo, donde ϵ es un umbral aceptación definido por el algoritmo.

Capítulo 3

Metodología

3.1. Preprocesamiento del corpus

3.1.1. Galaxy Zoo 2

Los métodos tradicionales de clasificación morfológica de galaxias cuentan con un número limitado de datos debido a que la clasificación a manos de expertos resulta ineficiente para la gran cantidad de imágenes del universo que disponemos hoy en día, como respuesta a la demanda de un método práctico surge el proyecto Galaxy Zoo en 2007 con la finalidad de proveer una clasificación de casi un millón de galaxias (10). Debido al gran apoyo de la comunidad en 2013 se crea una segunda versión del proyecto que buscaba hacer un estudio más completo de la morfología añadiendo una mayor variedad de características que puedan ser correlacionadas con otras propiedades físicas de las galaxias así como una mejor estadística de los objetos del universo corrigiendo muchos de los errores de la primera versión. El conjunto de imágenes empleadas por Galaxy Zoo2 contiene 245609 galaxias provenientes de los sistemas más grandes, brillantes y cercanos que permiten la clasificación morfológica de características finas (21).

El método de clasificación consiste en mostrar a un grupo de voluntarios imágenes generadas a partir del catálogo SDSS Data Release 7 [1], acompañando cada imagen de una pregunta y un conjunto de posibles respuestas utilizando un árbol de decisiones de 11 preguntas de respuesta múltiple 3.1. Opcionalmente la fracción de votantes cada respuesta fue ajustada con la finalidad de evitar una clasificación sesgada producto del corrimiento al rojo que provoca cambios en la morfología observada independientes a la evolución de la galaxia (21).

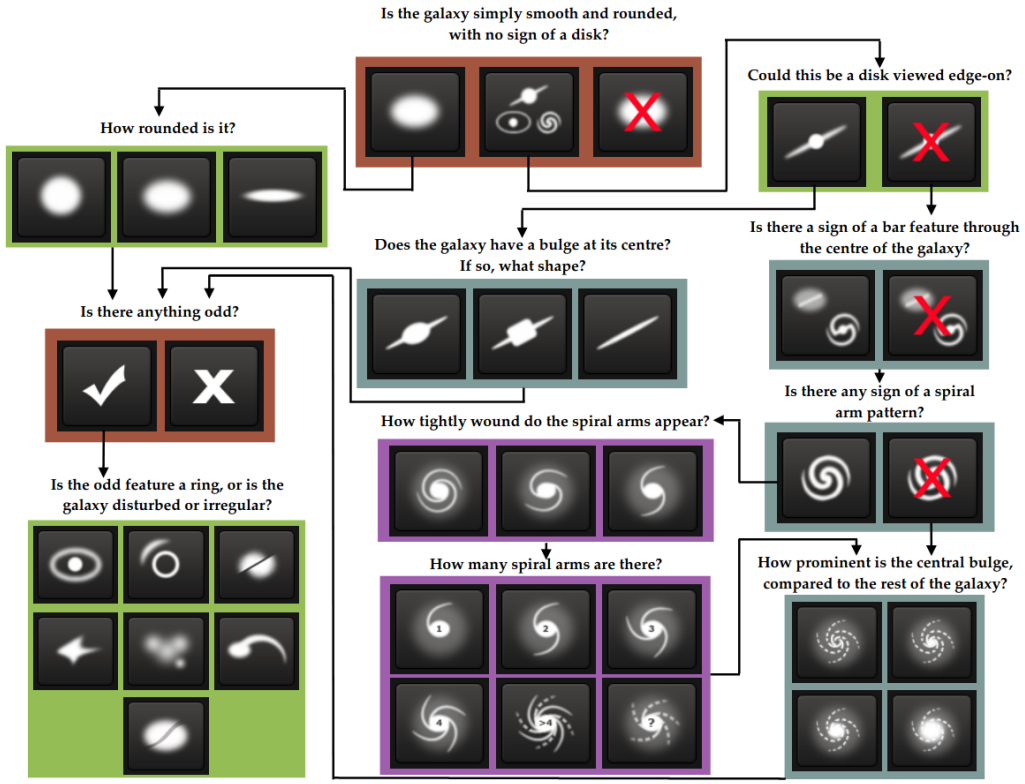


Figura 3.1: Árbol de preguntas Galaxy Zoo 2. Imagen tomada de *Galaxy Zoo 2: detailed morphological classifications for 304,122 galaxies from the Sloan Digital Sky Survey, 2013* (21)

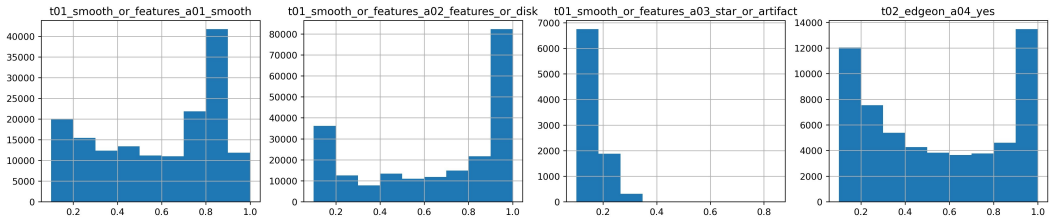
[1] *SDSS Data Release 7:* Es la séptima versión de la principal publicación de datos que provee de imágenes, espectros e información del corrimiento al rojo disponible para su descarga.

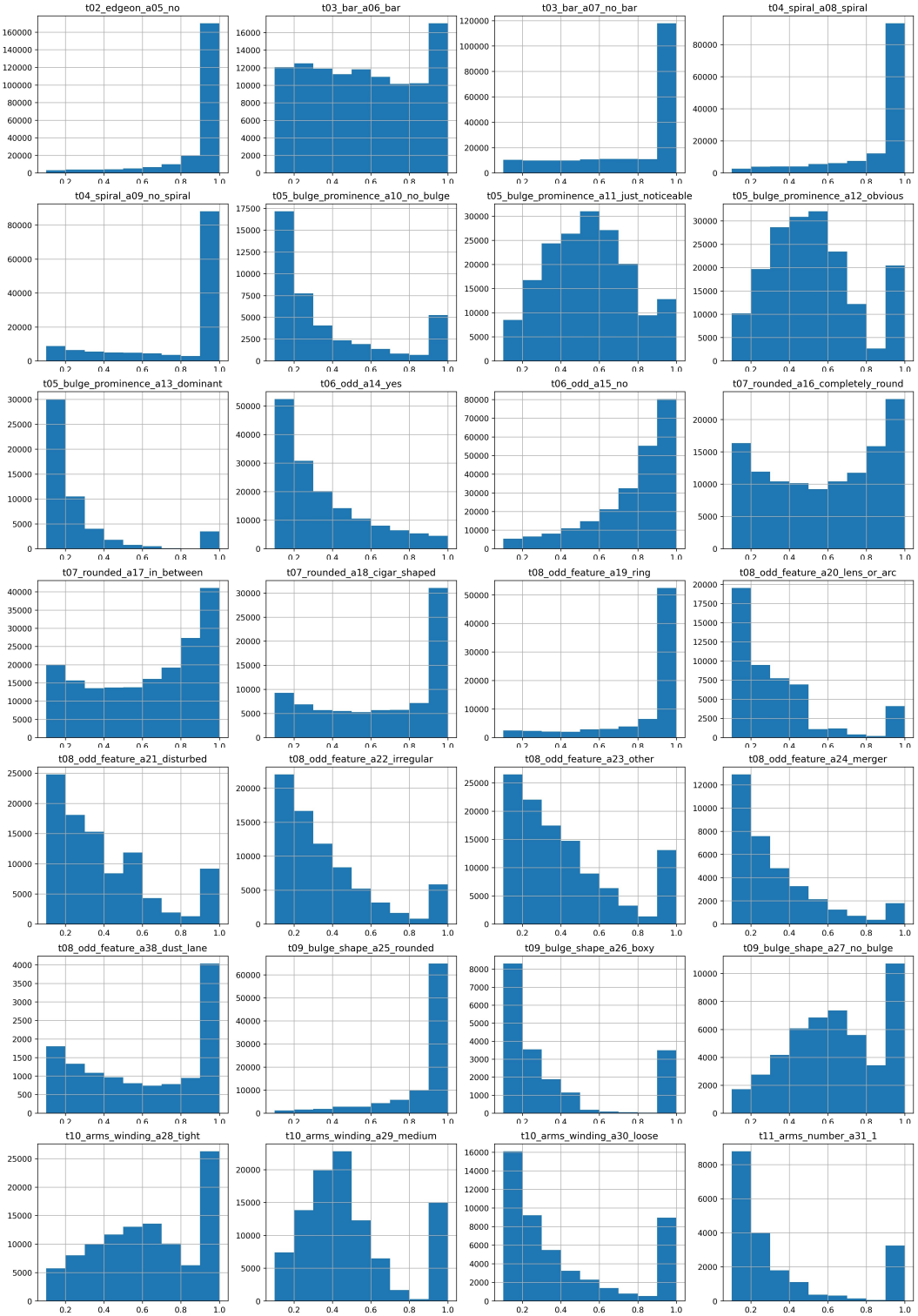
3.1.2. Generación del conjunto entrenamiento

El modelo original DELF fue entrenado para reconocer lugares emblemáticos presenta un pésimo desempeño cuando se utiliza para a la extracción de características en imágenes de galaxias, obteniendo un número bajo de características en zonas de la imagen con poco interés, este inconveniente se corrige mediante el ajuste del modelo para la detección, descripción y clasificación de puntos de interés en galaxias.

El ajuste se realiza mediante el entrenamiento con el conjunto de imágenes de Galaxy Zoo 2 en la tarea de clasificación, para esto el modelo precisa que cada imagen corresponda a una sola categoría por lo cual es necesario procesar los datos del corpus para crear un conjunto de entrenamiento adecuado.

La definición de categorías considera las respuestas del árbol de decisiones independientes entre sí, es decir, que el camino que conduzca a la respuesta propuesta como categoría sea único lo que se logra definiendo umbrales de aceptación en cada respuesta que se concatenan con las respuestas previas a la categoría propuesta. La asignación de categorías se basa en las fracciones de votos calculadas a partir del método insesgado, omitiendo aquellas preguntas donde la respuesta de los participantes no es concisa o ambigua. (Véase 3.2 t_05_bulge_pominance_a11_just_noticeable).




t02_edgeon_a05_no
t03_bar_a06_bar
t03_bar_a07_no_bar
t04_spiral_a08_spiral
t04_spiral_a09_no_spiral
t05_bulge_prominence_a10_no_bulge
t05_bulge_prominence_a11_just_noticeable
t05_bulge_prominence_a12_obvious
t05_bulge_prominence_a13_dominant
t06_odd_a14_yes
t06_odd_a15_no
t07_rounded_a16_completely_round
t07_rounded_a17_in_between
t07_rounded_a18_cigar_shaped
t08_odd_feature_a19_ring
t08_odd_feature_a20_lens_or_arc
t08_odd_feature_a21_disturbed
t08_odd_feature_a22_irregular
t08_odd_feature_a23_other
t08_odd_feature_a24_merger
t08_odd_feature_a38_dust_lane
t09_bulge_shape_a25_rounded
t09_bulge_shape_a26_boxy
t09_bulge_shape_a27_no_bulge
t10_arms_winding_a28_tight
t10_arms_winding_a29_medium
t10_arms_winding_a30_loose
t11_arms_number_a31_1

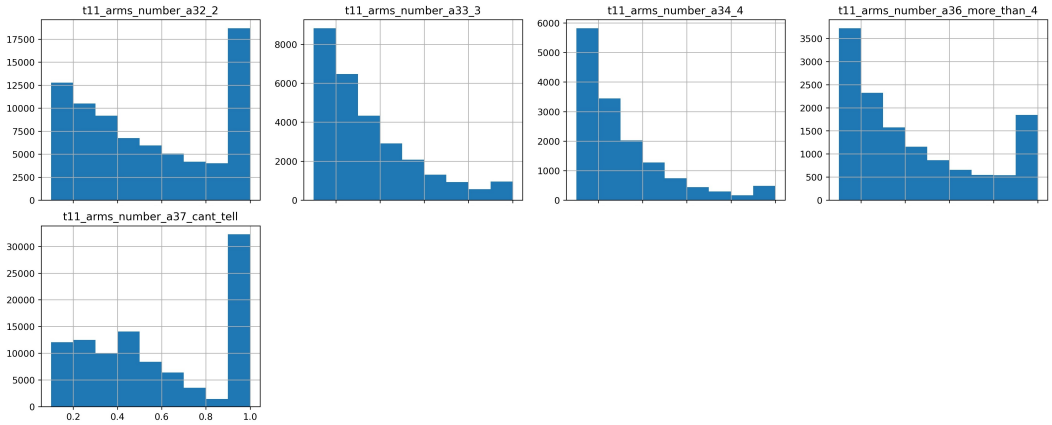


Figura 3.2: Histograma de cada una de las respuestas del árbol de decisiones de Galaxy Zoo 2

En la generación del conjunto de datos se establece umbral de aceptación superior a 0.70 para las respuestas previas y 0.75 para asignar la imagen a una de las 9 categorías propuestas obteniendo una distribución de categorías desbalanceada 3.3, no obstante, con la finalidad de mejorar el entrenamiento del modelo y reducir la carga de procesamiento se seleccionan aleatoriamente hasta 5000 imágenes de cada categoría 3.4.

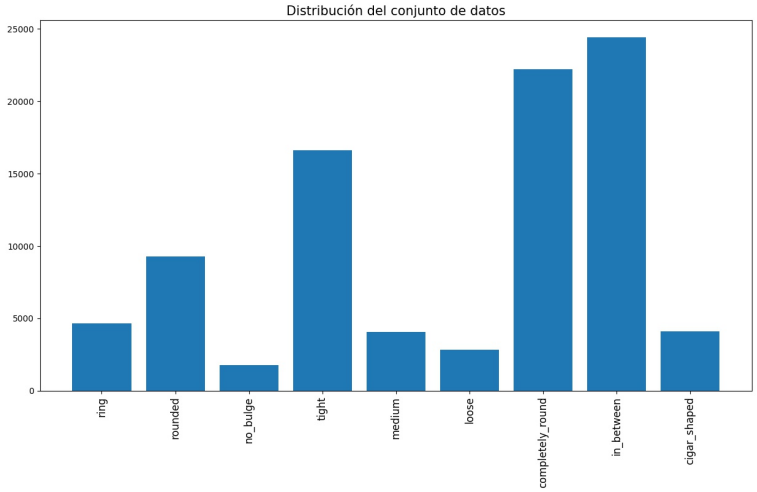


Figura 3.3: Distribución del conjunto de datos

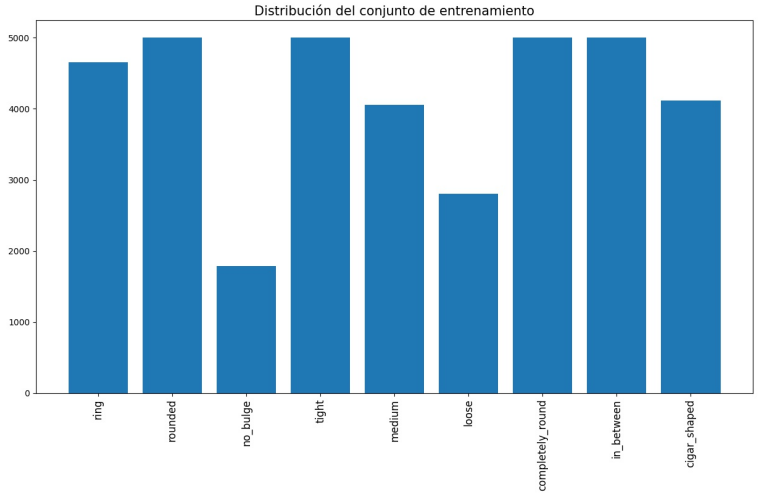


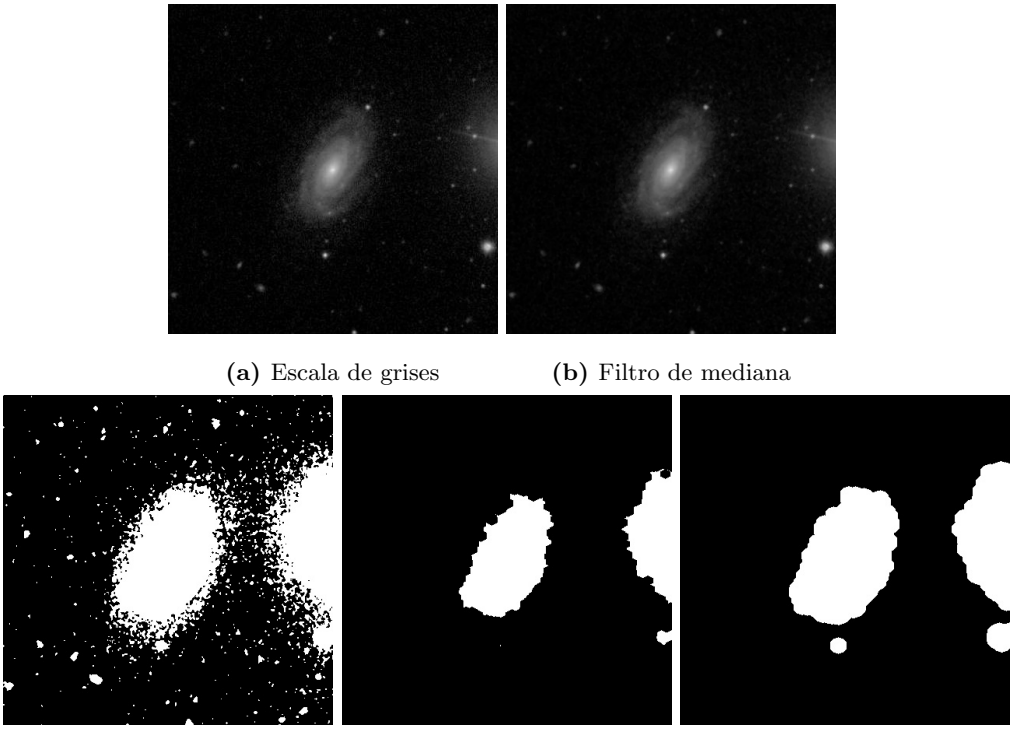
Figura 3.4: Distribución del conjunto de entrenamiento

3.1.3. Eliminación de ruido de fondo

La minimización de ruido de fondo presente en las imágenes de Galaxy Zoo 2 se realiza mediante la aplicación de mascaras de filtrado que eliminan el ruido sin modificar la morfología de la galaxia. Las imágenes seleccionadas del conjunto de entrenamiento se procesan cambiando el esquema de colores RGB a escala de grises 3.5a utilizando la relación:

$$I = 0.299R + 0.587G + 0.114B \tag{3.1}$$

Después se aplica un filtro de mediana utilizando un kernel 9x9 que reduce el ruido de fondo 3.5b, se realiza la binarización de la imagen 3.5c para aplicar el operador morfológico de erosión para eliminar estrellas pequeñas y ruido restante 3.5d seguido del operador de dilatación para compensar las modificaciones sufridas al borde de las galaxias 3.5e.



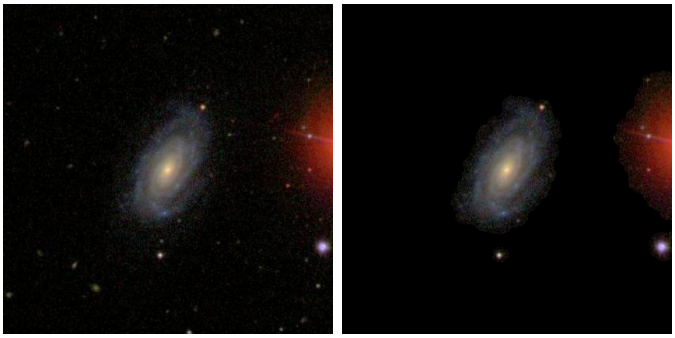
(a) Escala de grises **(b)** Filtro de mediana

(c) Binarización **(d)** Erosión **(e)** Dilatación

Figura 3.5: Resultados intermedios para la eliminación de ruido de fondo.

Este procesamiento crea una mascara utilizada para extraer el objeto original sin modificar la morfología de la galaxia, adicionalmente, se aplicaron transformaciones de rotación y traslación que permiten aumentar el número de muestras sin introducir problemas de sobreajuste [1] en el modelo.

[1] El sobreajuste u *overfitting* es una condición donde el modelo no logra generalizar el problema que trata de resolver obteniendo bajo desempeño en las pruebas de validación.



(a) Imagen original. **(b)** Imagen procesada.

Figura 3.6: Resultado del procesamiento de reducción de ruido y eliminación de estrellas.

3.2. Modelo DELF aplicado a galaxias

3.2.1. Ajuste del modelo

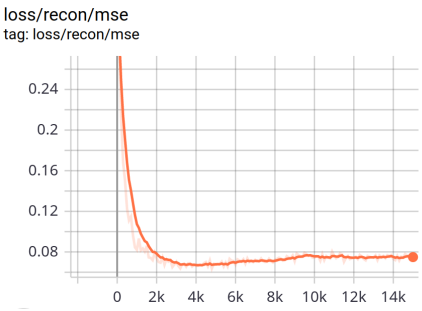
Tomando como base el código original [1]se adapta el modelo DELF para entrenar con las imágenes de Galaxy Zoo 2, para esto, se genera el conjunto de datos de entrenamiento implementando las reglas definidas anteriormente y se transforman a una estructura de datos propia de TensorFlow denominada *TFrecords* que optimiza el procesamiento en la GPU.

El modelo utiliza la trasferencia de conocimiento para inicializar los pesos de la red con los primeros 3 bloques convolucionales de ResNet50, cuya capa de salida se adapta para clasificar entre 9 categorías distintas. Una vez que el modelo ha aprendido a procesar imágenes de galaxias se exporta y se crean los scripts necesarios para la carga del modelo y su implementación como extractor de características.

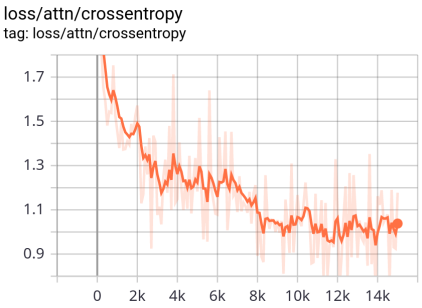
[1]Repositorio GitHub: Deep Local and Global Image Features

3.2.2. Entrenamiento

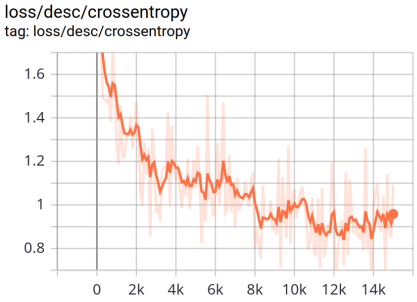
El conjunto de datos utilizado contiene 42097 imágenes generadas a partir de 3.1.2 junto con 42097 imágenes aplicando el método de eliminación de ruido 3.1.3 sobre las imágenes originales, esta combinación de muestras se elige con al finalidad de que el modelo aprenda de forma implícita la irrelevancia del ruido de fondo y asignara una puntuación baja a las características situadas en estas zonas.



(a) Función de pérdida del modelo en la clasificación de galaxias.



(b) Función de pérdida del modelo en la red de atención.



(c) Función de pérdida del modelo en la descripción de características.

Figura 3.7: Métricas de desempeño del modelo.

Los datos se particionan en dos conjuntos utilizando el 80 % para entrenamiento y 20 % para validación y se entrena al modelo durante 15,000 iteraciones utilizando

un batch size de 52. En este punto el modelo había alcanzado un nivel aceptable para utilizarlo como extractor de características.

3.3. Pipeline de procesamiento

3.3.1. Extracción de características

La extracción de características mediante el modelo DELF se realiza utilizando el corpus completo de 245609 imágenes, mientras que para SIFT se aplica el procesamiento para la eliminación de ruido 3.1.3 en corpus completo con la finalidad de reducir el número de características provenientes de estrellas y ruido.

Ambos algoritmos implementan una interfaz para homologar la estructura de datos y hacer que el post procesamiento sea independiente del algoritmo de extracción.

La extracción mediante SIFT se realiza utilizado la implementación del algoritmo por OpenCV [1] especificando los parámetros mediante la línea de comandos y en un archivo en el caso del modelo DELF. La configuración de parámetros especifica las cualidades que deben de cumplir los puntos de interés, el nivel de resolución y el número máximo de las características por imagen.

Las características extraídas se almacenan una estructura de datos con el nombre de la imagen, su localización, el tamaño y en el caso del extractor DELF la puntuación de las características encontradas, mientras que los descriptores se almacenan de forma independiente en un archivo *npy*.

[1] Documentación del algoritmo SIFT en OpenCV

location_x	location_y	size	score	image_name
226	135	1.4142271	190.65591	F1156
224	128	1.0	176.83842	F1156
226	135	0.7071135	173.92989	F1156
226	113	0.7071135	165.63936	F1144
240	128	0.5	132.9846	F1144

Tabla 3.1: Muestra del archivo de salida del proceso de extracción utilizando el algoritmo DELF.

3.3.2. Vocabulario de palabras visuales

Encontrar las relaciones existentes exige de la cuantificación de las características dentro de una imagen, sin embargo, las características extraídas de un mismo objeto varían en función de las propiedades en la imagen (cambios en la iluminación, tamaño o contexto del objeto) que provocan que los descriptores no sean exactamente iguales.

Haciendo una analogía al lenguaje cada uno de los descriptores de características puede representarse como una palabra, creando un vocabulario de palabras visuales mediante la agrupación de características semejantes y la asignación de un identificador a cada grupo.

La creación de palabras visuales se hace mediante Mini Batch K-Medias definiendo diferentes tamaños de vocabulario procesando todos los descriptores encontrados por el paso anterior. Se escogió Mini Batch K-Medias sobre K-Medias tradicional debido a que se ha demostrado un uso más eficiente de los recursos especialmente cuando se trata con conjuntos de datos muy grandes (3).

Este proceso asigna un identificador numérico que representa una palabra visual del vocabulario, que es concatenado al los datos del extractor.

location_x	location_y	size	score	image_name	visual_word_id
226	135	1.4142271	190.65591	F1156	322
224	128	1.0	176.83842	F1156	232
226	135	0.7071135	173.92989	F1156	105
226	113	0.7071135	165.63936	F1144	279
240	128	0.5	132.9846	F1144	596

Tabla 3.2: Muestra del archivo de salida del proceso de generación del vocabulario

3.3.3. Minado de estructuras con Sampled-MinHashing

Haciendo uso del vocabulario se representa cada una de las imágenes como un conjunto de palabras visuales generando un archivo que contiene el número de palabras visuales por imagen, el conjunto de palabras visuales y la frecuencia con la que aparecen dentro de la imagen.

3	339:1 750:1 898:1
11	166:1 312:1 352:1 432:3 465:2 544:2 855:1 860:1 881:2 980:1 990:1
3	100:3 198:1 283:1
7	402:1 488:1 693:1 780:2 834:1 843:1 958:1
6	170:1 279:1 383:1 429:2 455:1 925:1

Tabla 3.3: Imágenes representadas como conjuntos de palabras visuales: cada renglón representa una imagen, el primer número indica el total de palabras que la componen, el resto de valores corresponde a el identificador de la palabra visual y su frecuencia.

En forma similar al procesamiento de textos se eliminan las palabras sobre repre-

sentadas y sub representadas del vocabulario que pudieran ser poco informativas. Se seleccionan exclusivamente aquellas palabras cuya frecuencia de aparición en el corpus este como máximo a 2 desviaciones estándar el promedio, esta adecuación es opcional y puede ser implementada modificando los parámetros del script que genera el archivo.

Con el archivo generado se construye un índice invertido que sirve para realizar el minado de estructuras visuales. El minado de estructuras se realiza con la implementación de G. Fuentes y I. V. Meza [1] utilizando distintas combinaciones de los parámetros l y r para comparar los modelos obtenidos.

1763	208:1 295:1 697:1 786:1 831:1 876:1 948:1 1005:1 1189:1 1610:1 ...
793	1222:1 1241:1 1867:1 1966:1 2190:1 2264:3 2503:1 3634:1 4047:1 ...
2172	264:1 549:1 804:1 973:1 990:1 1045:1 1305:1 1368:1 1403:1 1777:1 ...
1449	135:1 196:1 199:1 296:1 497:1 716:2 818:1 1077:1 1279:1 1358:1 ...
766	130:2 469:2 555:1 952:1 1495:1 1569:2 2312:1 2776:1 3216:1 3636:1 ...

Tabla 3.4: Índice invertido de palabras visuales: en cada renglón se especifican las imágenes en las que aparece la palabra visual, el primer renglón indica el número de apariciones a través de toda la colección.

3.4. Descubrimiento de estructuras

3.4.1. Interpretación de características y estructuras

Pensar en características y estructuras visuales puede ser poco intuitivo sin apoyo gráfico. Las características extraídas por métodos como SIFT y DELF corresponden a una región dentro de la imagen descrita mediante un vector numérico de tal manera que

[1] Repositorio GitHub: Sampled-MinHashing

podemos calcular la similitud entre regiones mediante la comparación de los vectores que las describen.

Las imágenes son representadas mediante todas las características mientras que los objetos contenidos en ellas corresponden a un subconjunto específico de características 3.8, en consecuencia, podemos interpretarlas como las partes más simples que conforman una imagen o objeto.

(a) Características de toda la imagen.

(b) Características del objeto cámara.

Figura 3.8: Segmentación de características extraídas con SIFT.

Por otro lado, las estructuras se definen mediante un conjunto de características que pueden formar parte de un objeto y a su vez conforman un patrón cuantificable a través de la colección de imágenes, en este sentido, las estructuras no representan una clasificación de las imágenes sino partes de objetos que se repiten a lo largo de toda la colección.

(a) Características **(b)** Estructuras **(c)** Objetos

Figura 3.9: Analogía a la extracción de características.

Visto de otra forma, una característica describe las formas simples que exhiben regiones de píxeles 3.9a mientras que las estructuras representan partes de objetos contenidos en una colección3.9b que en conjunto forman los objetos 3.9c.

3.4.2. Asignación de estructuras

La asociación de estructuras visuales a las imágenes se realiza midiendo superposición entre el conjunto de palabras visuales de una imagen y el conjunto de palabras que forman a la estructura:

$$\phi_s(imagen_i,\ estructura_j) = \begin{cases} \frac{|estructura_j \cap imagen_i|}{|estructura_j|} & \text{si } |imagen_i| > |estructura_j| \\ \frac{|estructura_j \cap imagen_i|}{|imagen_i|} & \text{en caso contrario} \end{cases} \tag{3.2}$$

Utilizando una interfaz sencilla se seleccionan los parámetros de búsqueda para una estructura, se filtran las imágenes y calcula la superposición de las características de la imagen y la estructura empleando la fórmula 3.2 para desplegar en pantalla las

coincidencias encontradas. Adicionalmente es posible visualizar las estructuras como un mapa de calor creando un arreglo del mismo tamaño que del vocabulario 3.10.

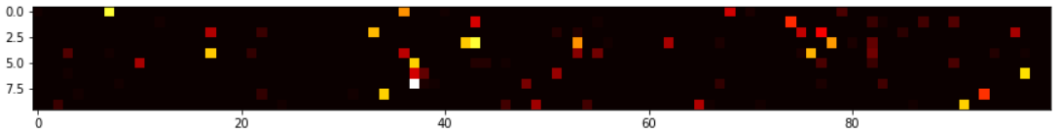


Figura 3.10: Estructura 0 del modelo DELF r2 l1000: Cada palabra visual esta representada por un pixel donde la intensidad representa la frecuencia de aparición en la estructura.

Capítulo 4

Análisis y Resultados

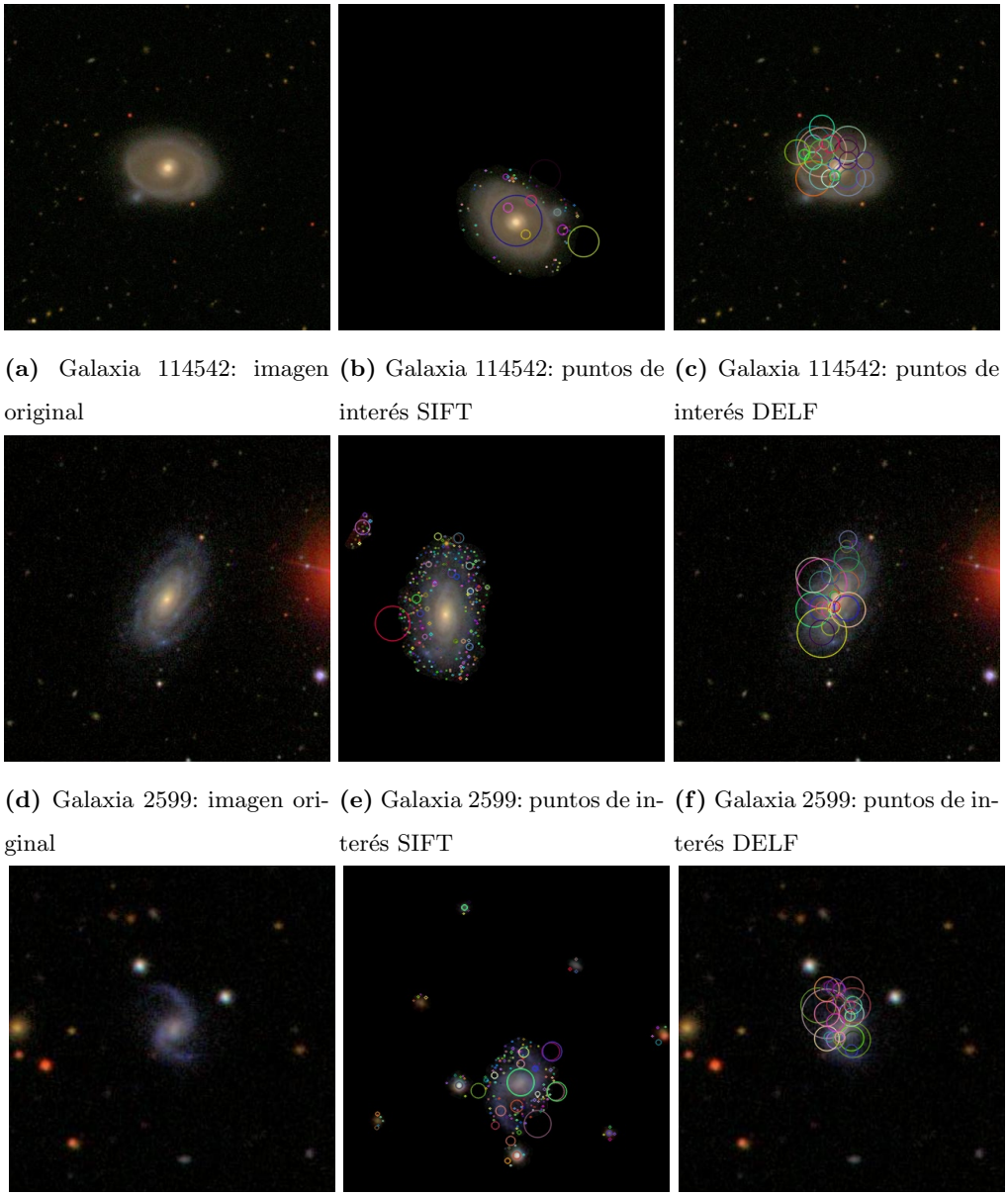
4.1. De características a palabras

4.1.1. Características encontradas

El número de características varía en función del método y la configuración. Los parámetros del algoritmo SIFT se establecieron para detectar puntos de interés en regiones pequeñas y medianas, con un bajo nivel de contraste, pese a utilizar imágenes filtradas para reducir las características irrelevantes el número de puntos de interés detectados fue de 48,305,020 en 243,434 imágenes, en contraste con los puntos obtenidos del modelo delf que genero 3,032,300 sobre 242,149 imágenes, ambos aplicados sobre el conjuntos completo de imágenes de GZ2.

Otra diferencia significativa en los métodos de extracción es el número de características por imagen, mientras que el modelo DELF tiene una media de 12.52 el extractor SIFT encuentra 198.43, esta diferencia radica principalmente en la forma en la que SIFT maneja el problema de la orientación ya que duplica puntos de interés si existe más de una dirección dominante en al histograma que conforma el descriptor creando

dos o más descriptores para el mismo punto.



(a) Galaxia 114542: imagen original

(b) Galaxia 114542: puntos de interés SIFT

(c) Galaxia 114542: puntos de interés DELF

(d) Galaxia 2599: imagen original

(e) Galaxia 2599: puntos de interés SIFT

(f) Galaxia 2599: puntos de interés DELF

(g) Galaxia 252299: imagen original

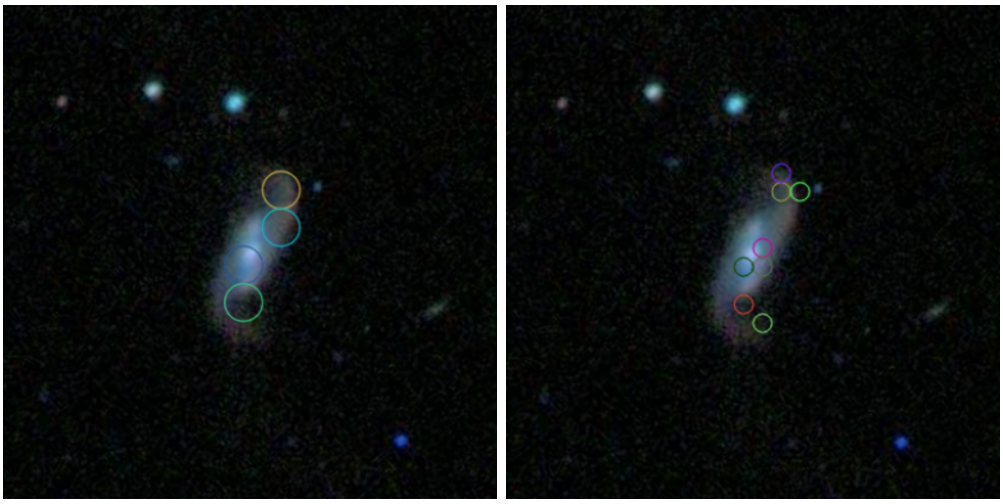
(h) Galaxia 252299: puntos de interés SIFT

(i) Galaxia 252299: puntos de interés DELF

Figura 4.1: Comparación de puntos de interés encontrados mediante SIFT y DELF.

Debido a que las características extraídas del modelo DELF se calculan a partir

de los mapas de características de la red convolucional todos los puntos de interés se encuentran limitados por una cuadrícula, no obstante, se obtienen diferentes locaciones y escalas por la implementación de la pirámide de imágenes.



(a) Puntos de interés encontrados en la imagen original.

(b) Puntos de interés encontrados en la imagen escalada al doble de la resolución original.

Figura 4.2: Puntos de interés detectados mediante DELF utilizando diferentes escalas para detectar los puntos

4.1.2. Vocabulario de palabras visuales

Se crearon 5 vocabularios visuales espaciados en 1000 unidades con la finalidad de observar el comportamiento de las estructuras minadas, en cada uno de los vocabularios se aplicó el procedimiento para reducción de tamaño 3.3.3 mediante el cual se esperaba reducir el vocabulario en un 95 %, por otra parte la reducción de todo el conjunto de palabras visuales varia de manera distinta para cada tamaño del vocabulario.

Tamaño del vocabulario	Promedio características por palabra	Desviación estándar	Reducción del vocabulario
1000	48305.02	34743.68	97.40 %
2000	24152.51	22671.75	97.70 %
3000	16101.67	16944.99	96.70 %
4000	12076.25	15012.05	97.22 %
5000	9661.00	12438.28	96.84 %

Tabla 4.1: Vocabulario con características SIFT

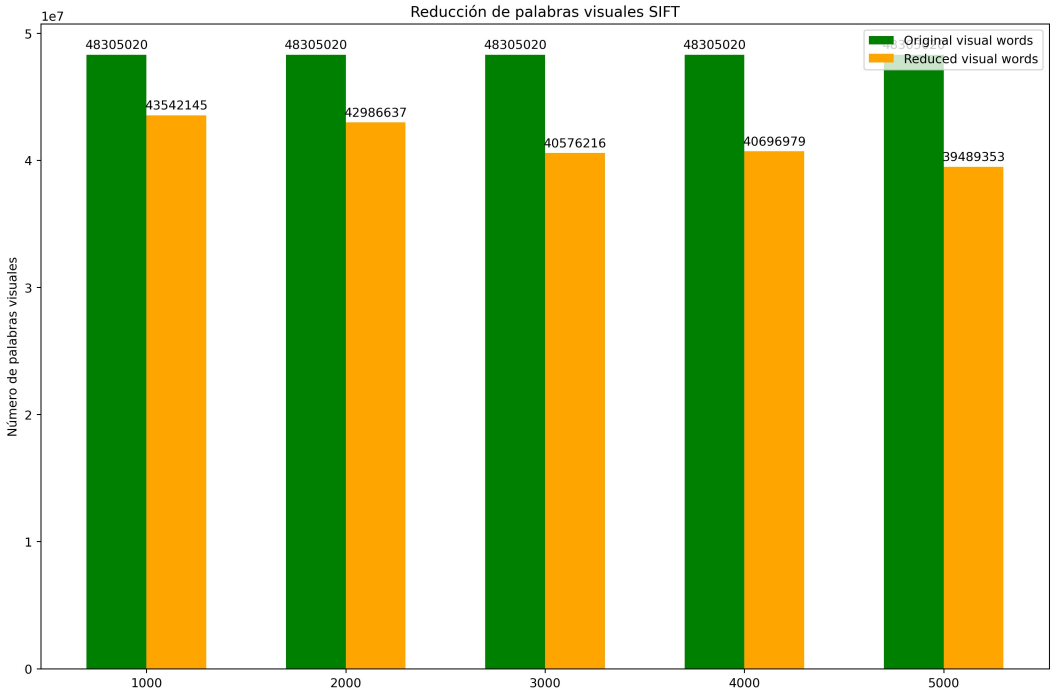


Figura 4.3: Reducción del vocabulario y su impacto en el total de características SIFT

Mientras que el número de palabras decrece a medida que el tamaño del vocabulario aumenta en el caso de SIFT 4.3 la reducción de descriptores mediante DELF parece

ser constante 4.4.

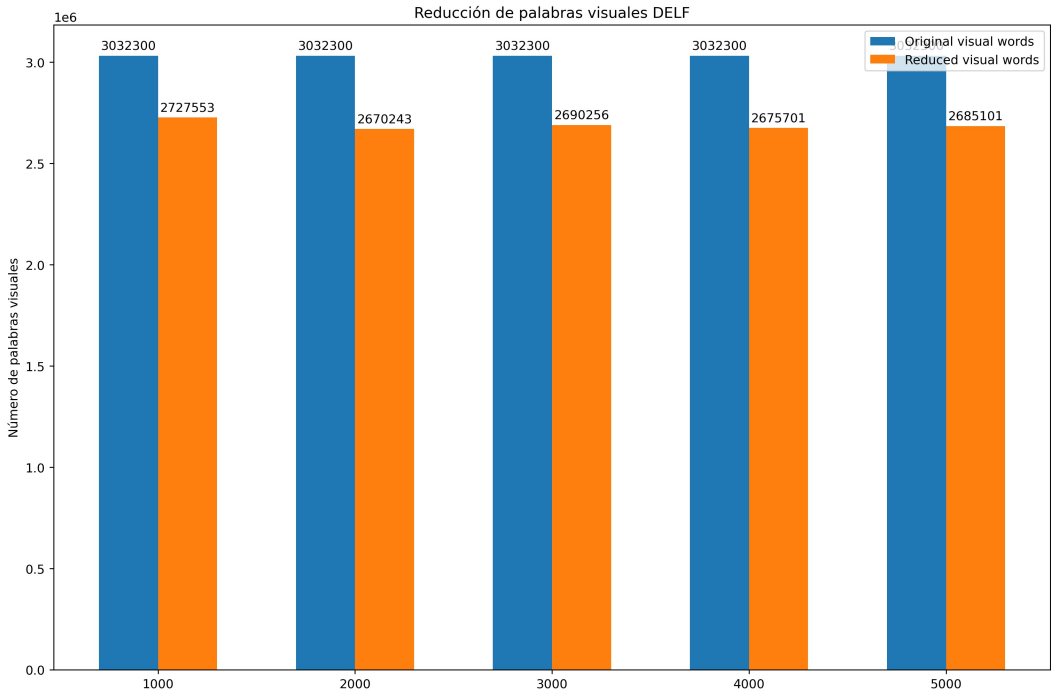


Figura 4.4: Vocabulario con características DELF

Tamaño del vocabulario	Promedio características por palabra	Desviación estándar	Reducción del vocabulario
1000	3032.30	1479.08	96.10 %
2000	1516.15	784.69	95.35 %
3000	1010.77	530.78	95.77 %
4000	758.08	389.38	95.40 %
5000	606.46	310.14	95.54 %

Tabla 4.2: Reducción del vocabulario y su impacto en el total de características DELF

4.2. Estructuras visuales

El minado de estructuras se realizó variando el tamaño de vocabulario, el valor r y el valor l del método Sampled Min-Hashing, con las estructuras obtenidas se realizó la comparación del número de estructuras obtenidas así como la distribución del tamaño de estas mediante diagramas de caja.

4.2.1. Minado de estructuras mediante SIFT

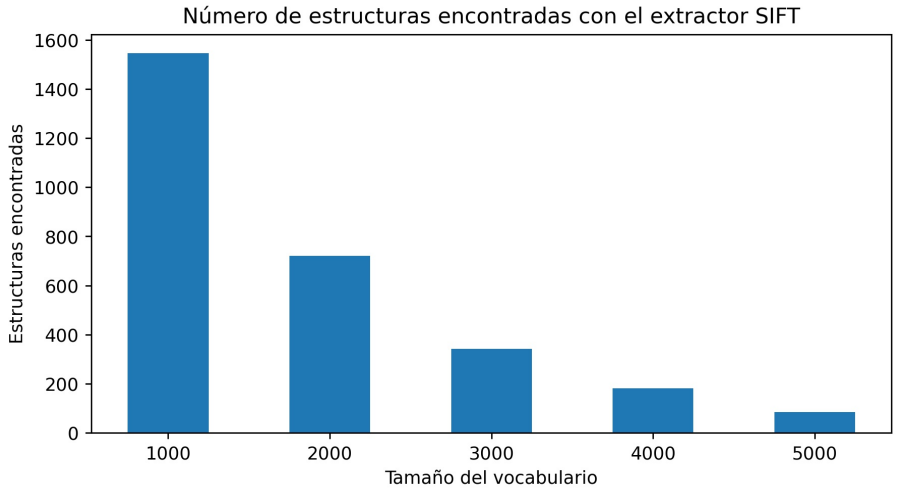


Figura 4.5: Estructuras SIFT encontradas: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

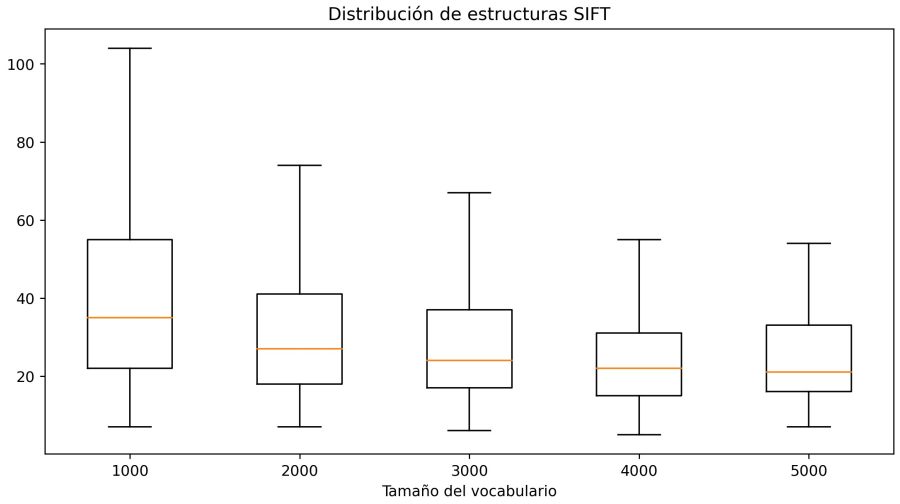


Figura 4.6: Gráficos de dispersión estructuras SIFT: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

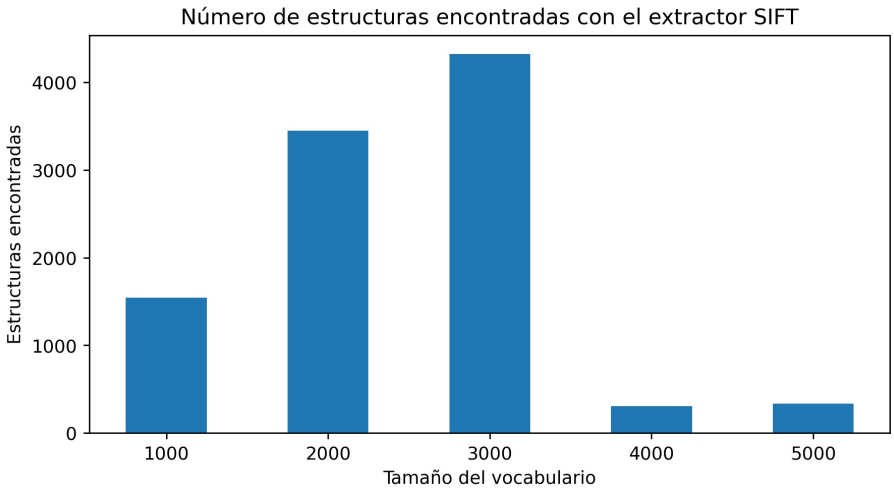


Figura 4.7: Estructuras SIFT encontradas: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

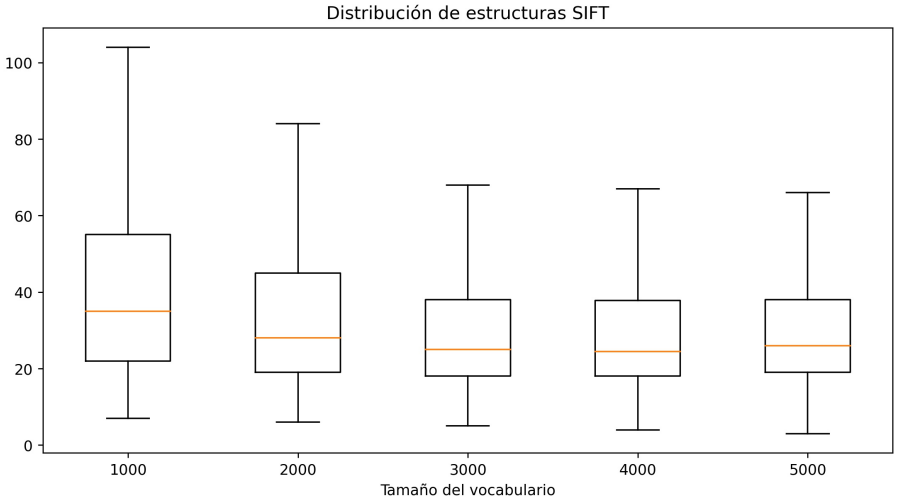


Figura 4.8: Gráficos de dispersión estructuras SIFT: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

4.2.2. Minado de estructuras mediante DELF

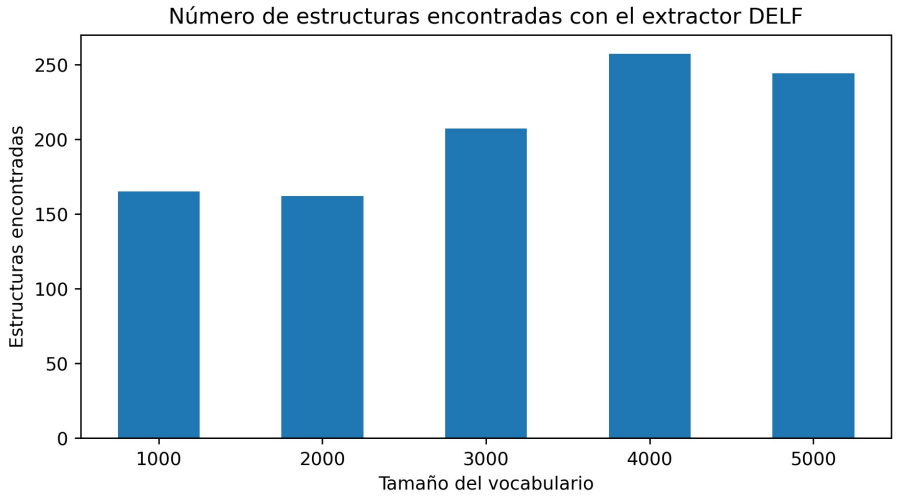


Figura 4.9: Estructuras DELF encontradas: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

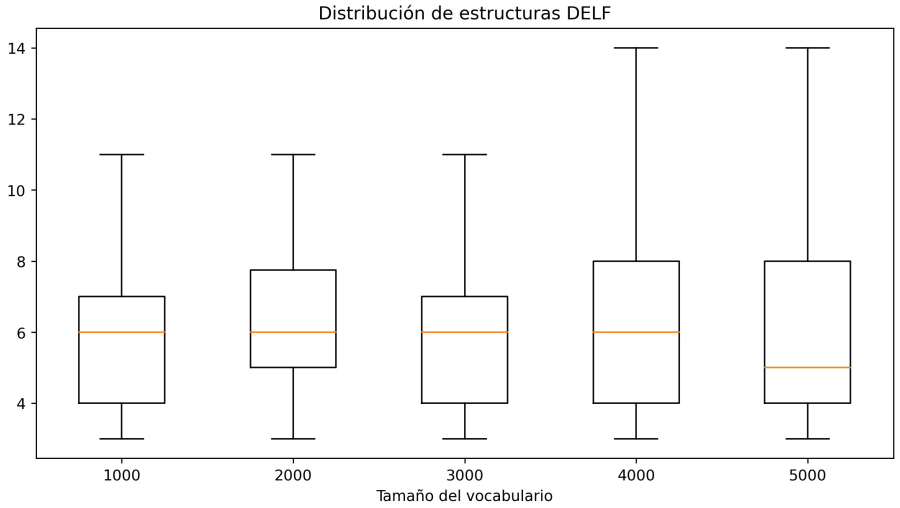


Figura 4.10: Gráficos de dispersión estructuras DELF: 2 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

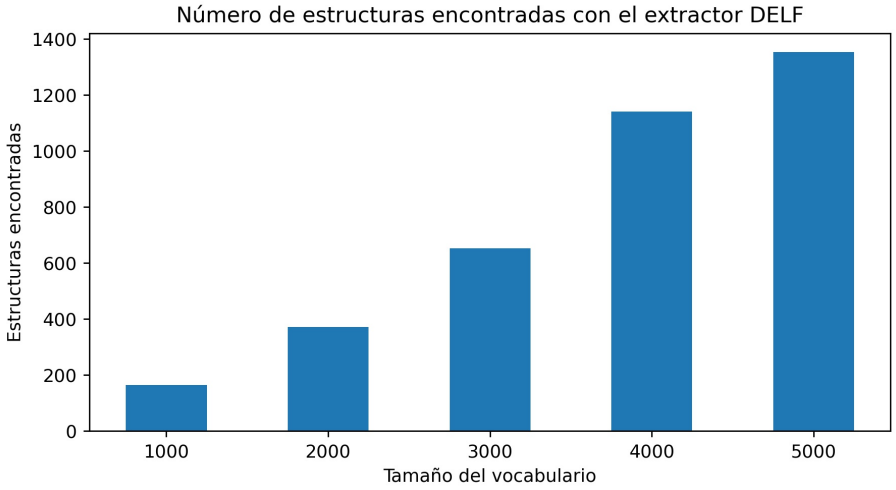


Figura 4.11: Estructuras DELF encontradas: 2 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

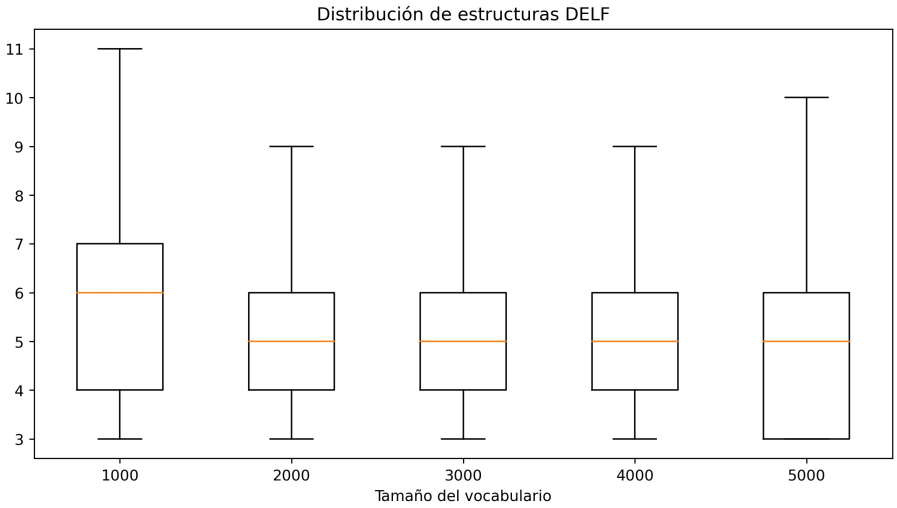


Figura 4.12: Gráficos de dispersión estructuras DELF: 2 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

Por medio de la interfaz se pueden buscar en cada modelo las imágenes que corresponden a una misma estructura, seleccionando la porción de datos donde se realiza la búsqueda o el umbral de aceptación de la superposición 3.2.

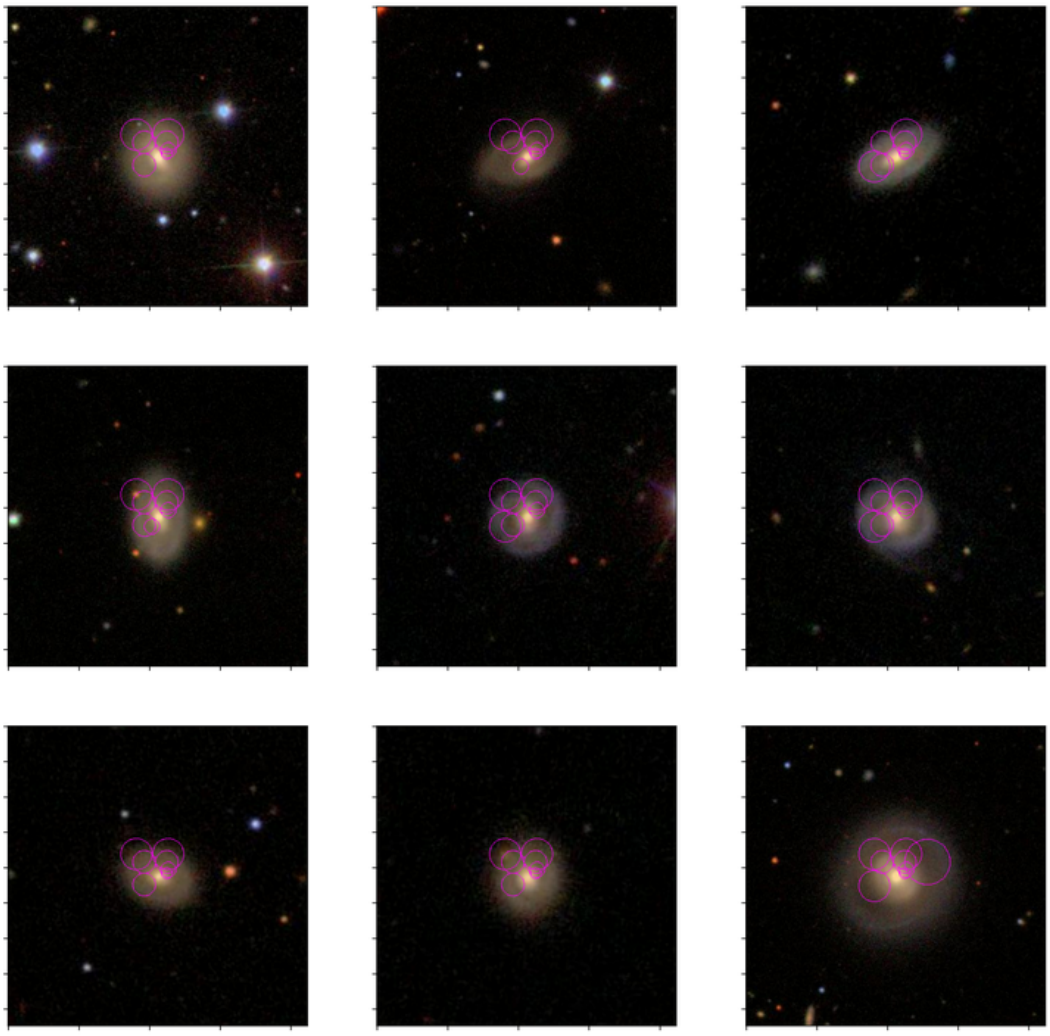


Figura 4.13: Imágenes en donde aparece la estructura 2 del modelo DELF r3 l1000

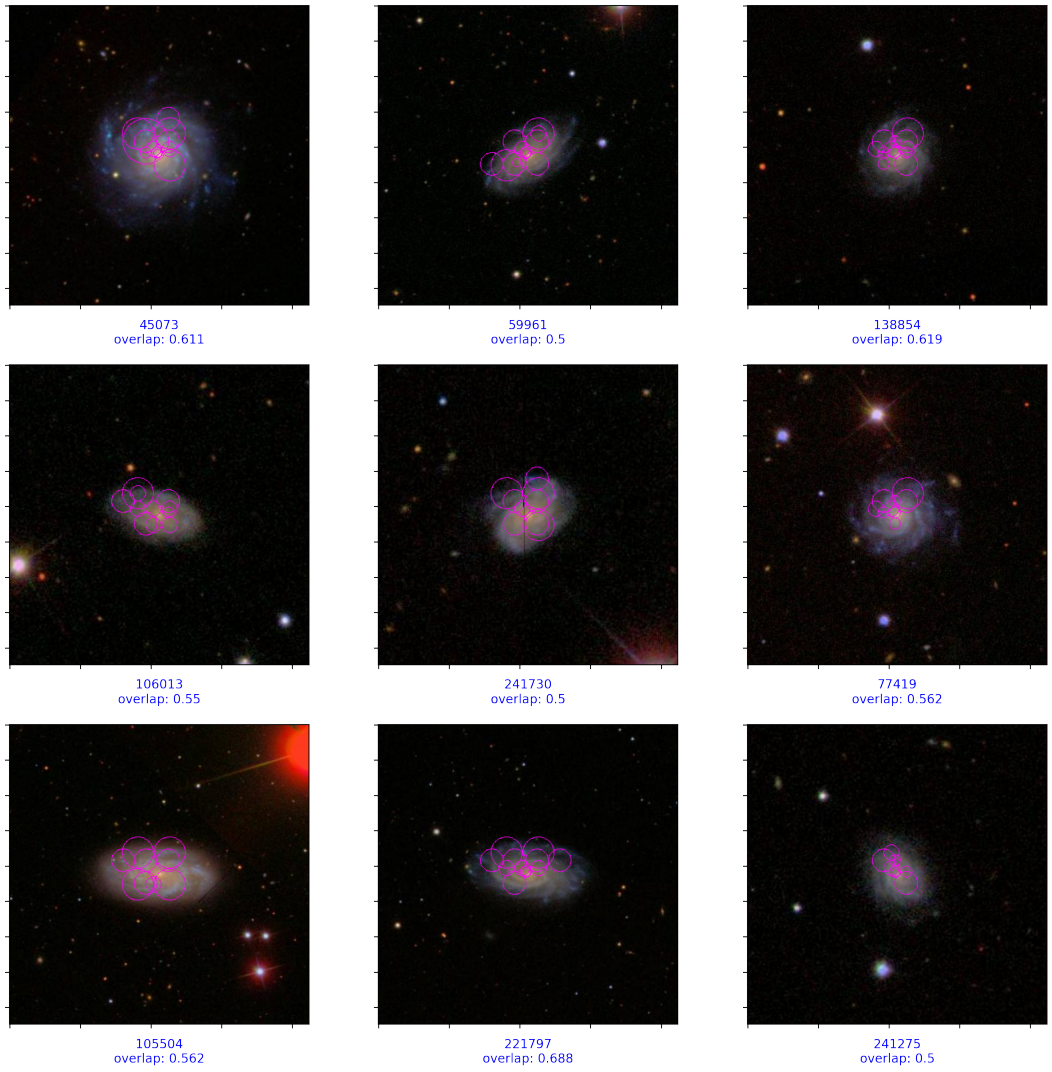


Figura 4.14: Imágenes en donde aparece la estructura 19 del modelo DELF r2 l1000

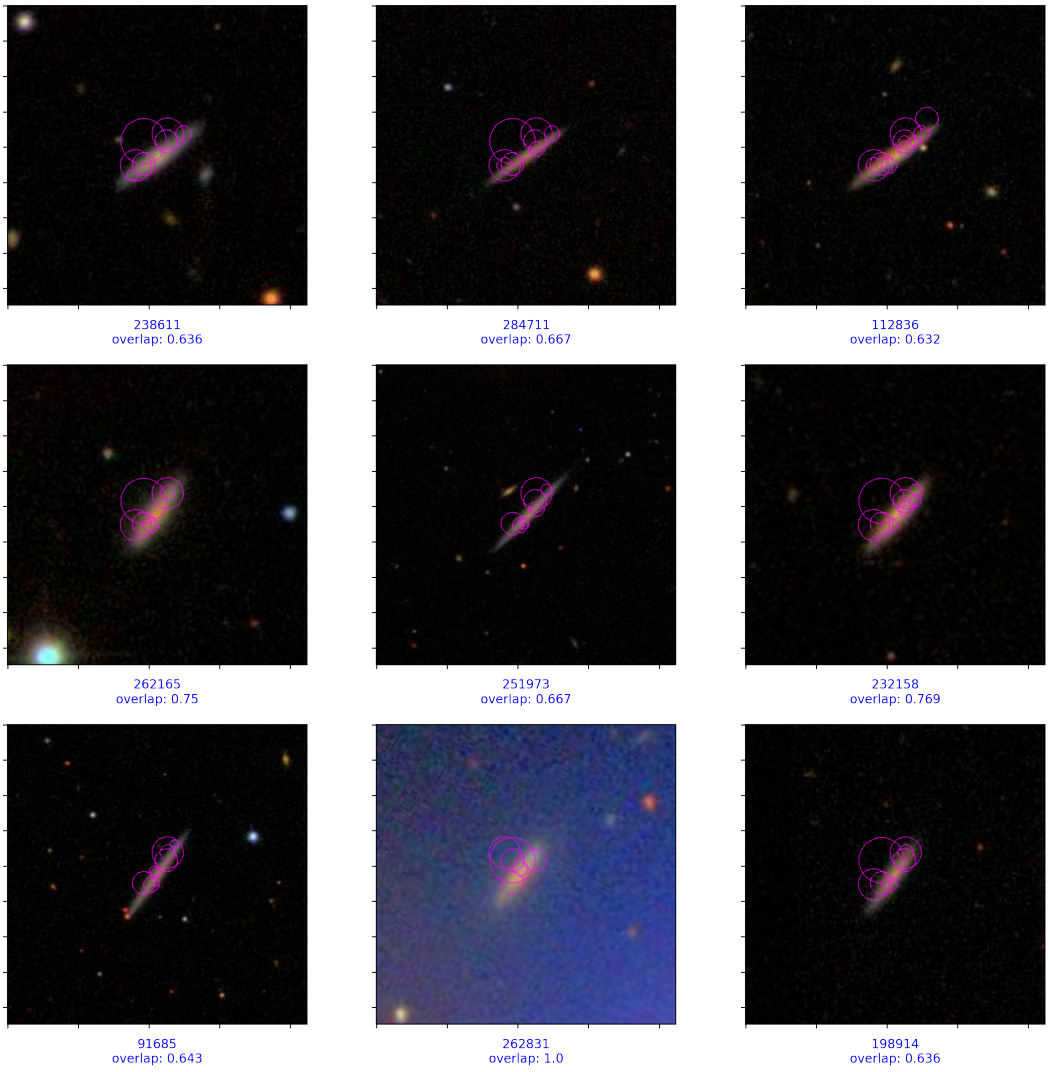


Figura 4.15: Imágenes en donde aparece la estructura 22 del modelo DELF r2 l1000

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

El descubrimiento de estructuras visuales mediante Sampled Min-Hasing es un método viable para encontrar patrones ocultos en colecciones masivas de imágenes. Es evidente la capacidad del método para encontrar patrones, sin embargo, la relevancia de cada uno de estos patrones no puede ser cuantificable. Este método es una herramienta de apoyo para que los expertos en el área puedan realizar conjeturas, interpretaciones y correlaciones acerca de patrones descubiertos.

El tamaño del vocabulario influye directamente en el número y tipo de estructuras encontradas, como se aprecia durante la experimentación a medida que aumenta el vocabulario se hace más complicado el minado de estructuras, esto tiene sentido ya que un amplio vocabulario es sinónimo de baja tolerancia a las variaciones haciendo que, por ejemplo, dos características con la misma forma sean tratadas como si fueran completamente diferente por ligeros cambios en la iluminación, rotación o perspectiva. Si bien el tamaño del vocabulario es arbitrario debe ser proporcional a la variabilidad de los descriptores de la colección de imágenes.

La extracción de características en imágenes de galaxias tiene una limitante para algoritmos como SIFT, ya que no cuentan con muchos cambios de contraste lo suficientemente significativos para considerarlos como puntos de interés, por otra parte, esto no representa problema para métodos como DELF debido a la implementación de red neuronal con atención que sirve para indicar cuales son los mejores candidatos a puntos de interés. Otra de las ventajas de utilizar DELF es el control para la selección de características, si bien mediante SIFT se pueden especificar las cualidades de las características a extraer no se tiene control de la relevancia ni del tamaño, cosas que con DELF sí.

La detección de estructuras morfológicas finas en el corpus Galaxy Zoo 2 es una tarea complicada, principalmente, por la baja resolución de las imágenes ya que dificulta la detección y segmentación de características pequeñas incluyendo otros problemas como el ruido. El procesamiento con esta metodología demuestra que el uso de Sampled Min-Hasing es una buena alternativa para atacar problemas donde se requiera extraer patrones a través del aprendizaje no supervisado con alta volumetría de imágenes.

5.2. Trabajo futuro

Una de las ventajas del conjunto de datos es que todas las galaxias tienen tamaños muy similares, por lo cual es posible realizar un filtrado de características SIFT basándonos en el tamaño de punto de interés, de esta manera es posible eliminar los puntos diminutos que contienen poca o nula información de la morfología de las galaxias, así mismo utilizar un conjunto de datos de mayor resolución que permita identificar las

características morfológicas finas en las galaxias.

Emplear las puntuaciones de las características del modelo DELF para representar a las imágenes como el conjunto de palabras visuales y la puntuación asignada (en lugar a la frecuencia) para validar si esto contribuye a detectar estructuras más interesantes.

Diseñar un modelo de extracción de características similar a DELF que no requiera de datos etiquetados para su entrenamiento.

Apéndice A

Variaciones de minado

A.1. Modelo de características SIFT

A.1.1. Variación de r valores min-hash con y l funciones constantes

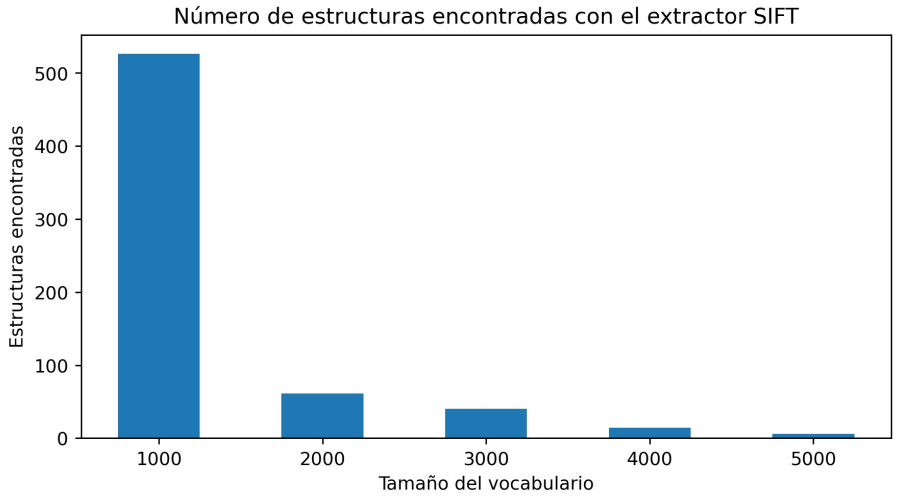


Figura A.1: Estructuras SIFT encontradas: 3 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

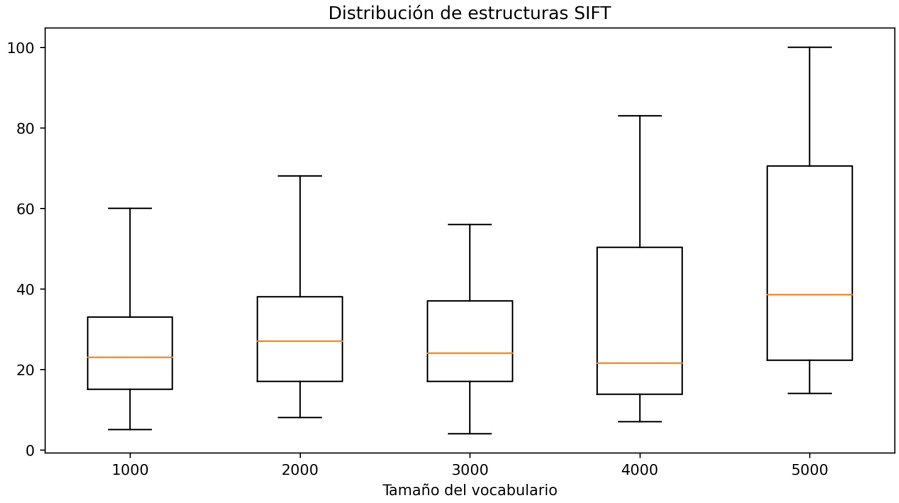


Figura A.2: Gráficos de dispersión estructuras SIFT: 3 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

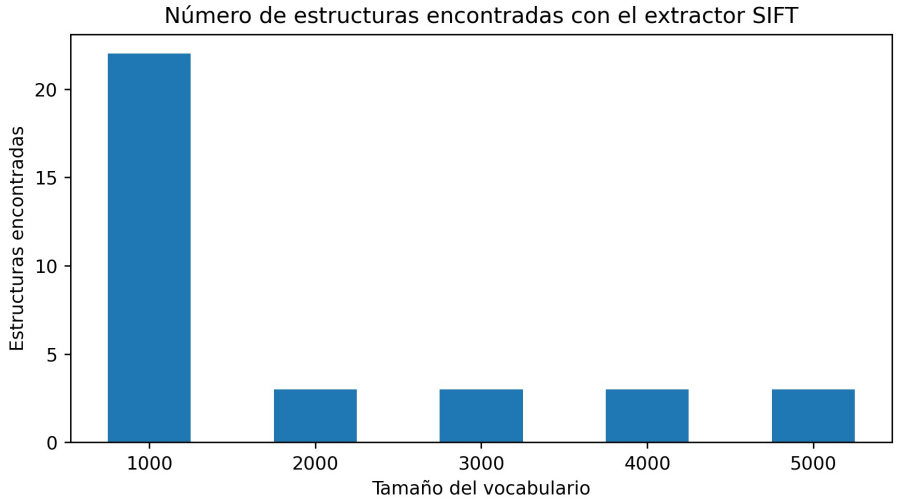


Figura A.3: Estructuras SIFT encontradas: 4 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

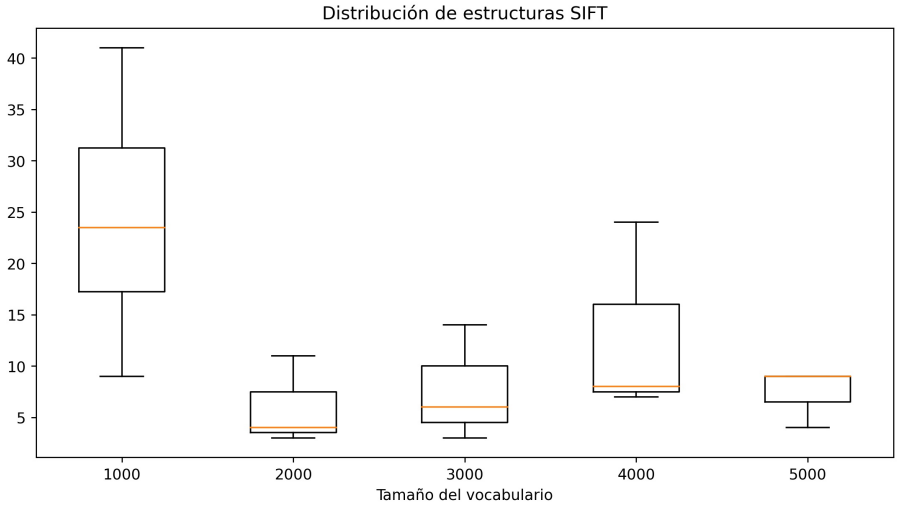


Figura A.4: Gráficos de dispersión estructuras SIFT: 4 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

A.1.2. Variación de r valores min-hash con y l funciones variables

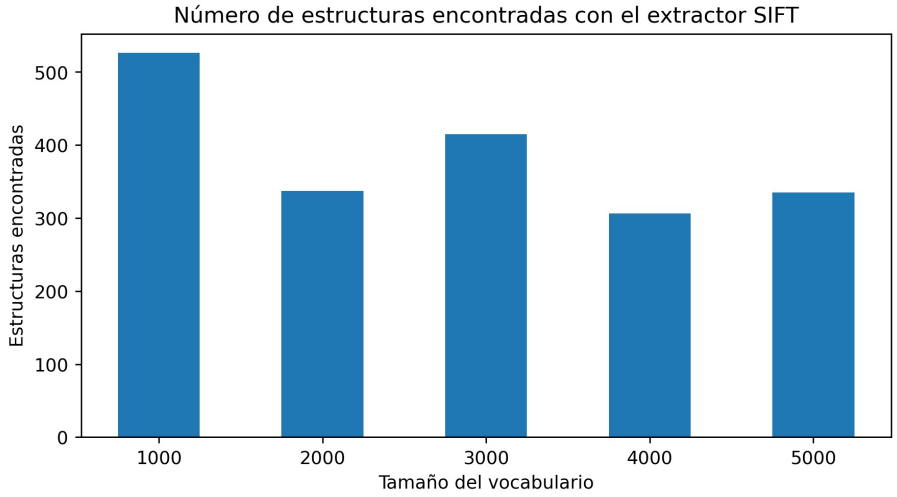


Figura A.5: Estructuras SIFT encontradas: 3 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

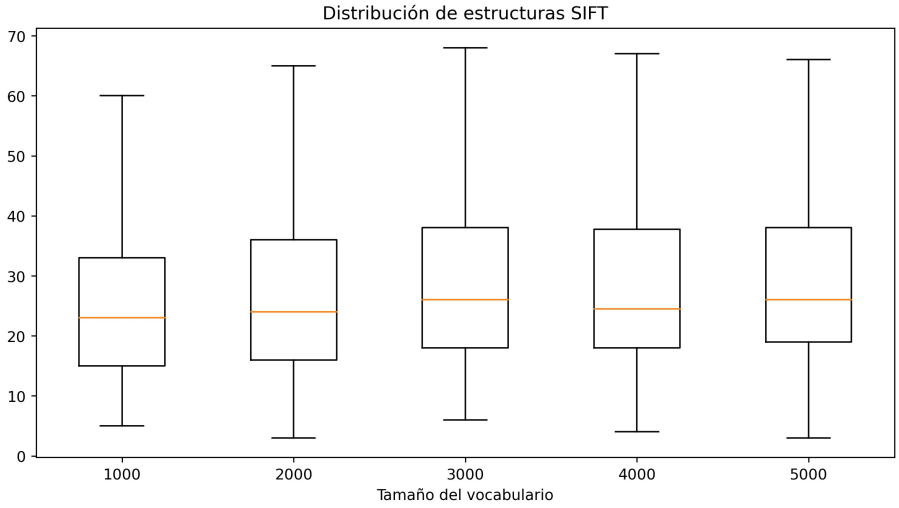


Figura A.6: Gráficos de dispersión estructuras SIFT: 3 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

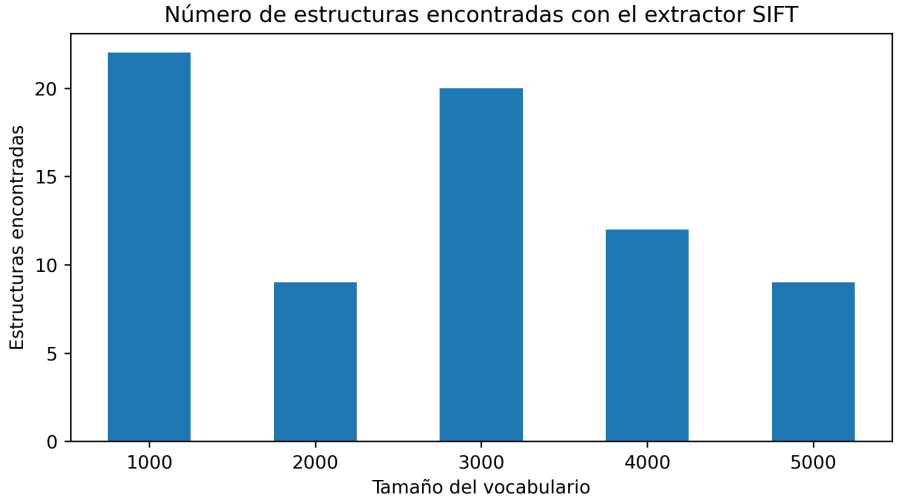


Figura A.7: Estructuras SIFT encontradas: 4 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

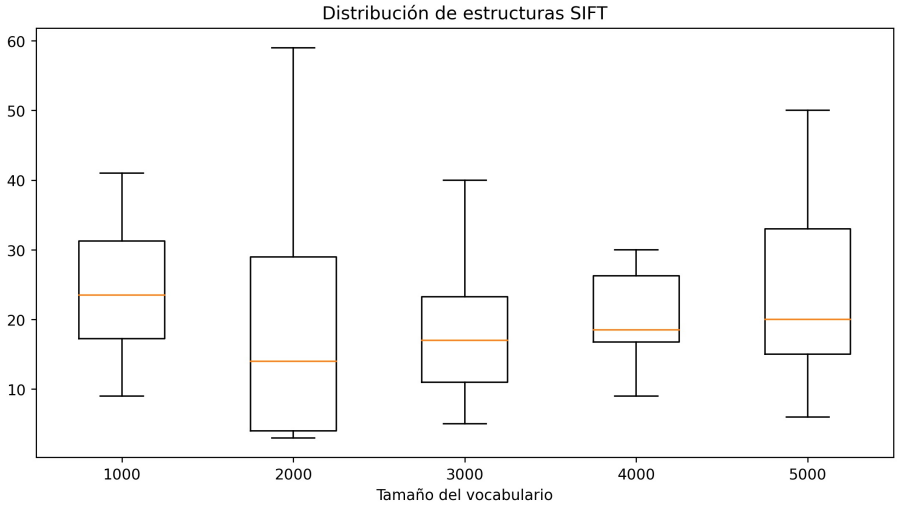


Figura A.8: Gráficos de dispersión estructuras SIFT: 4 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

A.2. Modelo de características DELF

A.2.1. Variación de r valores min-hash con y l funciones constantes

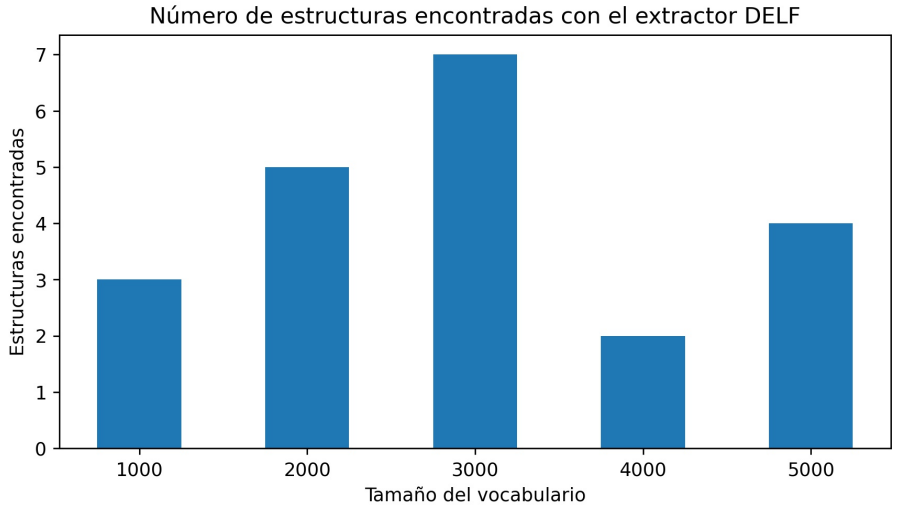


Figura A.9: Estructuras DELF encontradas: 3 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

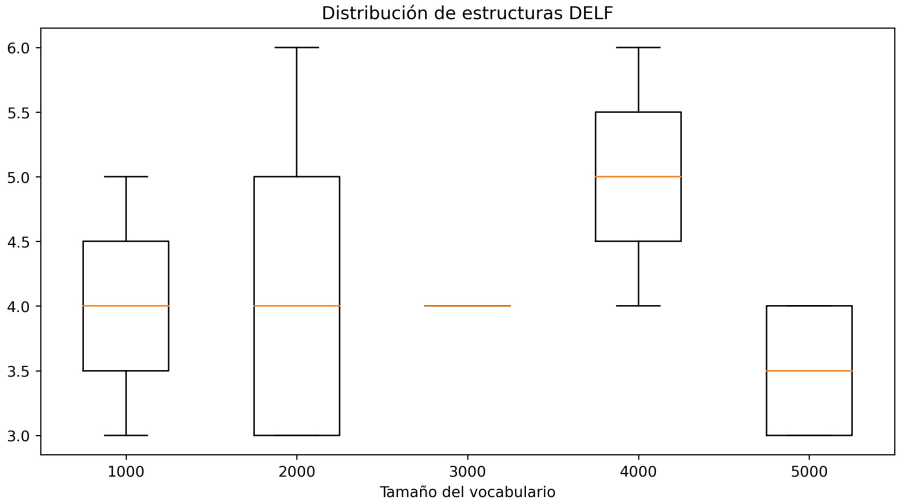


Figura A.10: Gráficos de dispersión estructuras DELF: 3 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

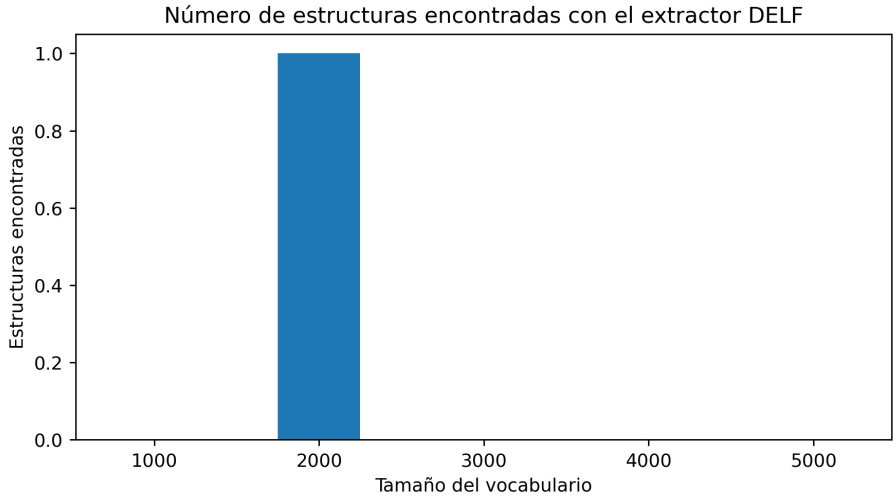


Figura A.11: Estructuras DELF encontradas: 4 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

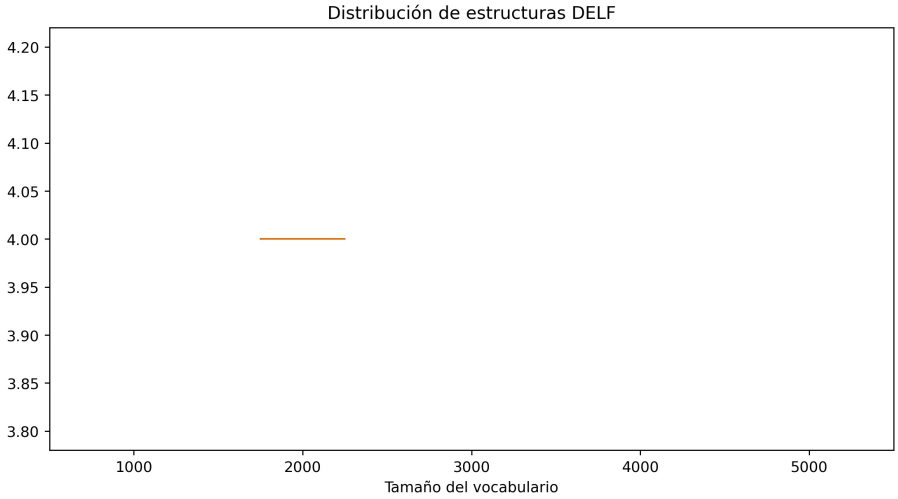


Figura A.12: Gráficos de dispersión estructuras DELF: 4 valores Min-hash y 1000 funciones min-hash para diferentes tamaños de vocabulario

A.2.2. Variación de r valores min-hash con y l funciones variables

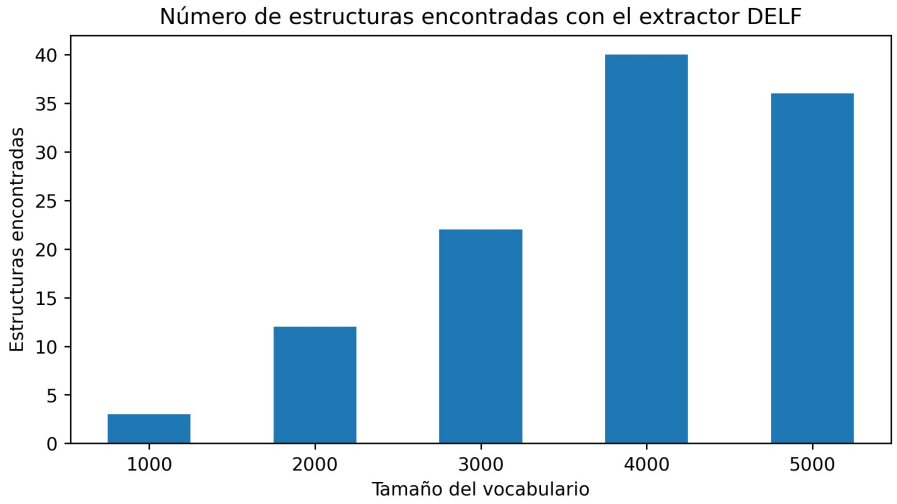


Figura A.13: Estructuras DELF encontradas: 3 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

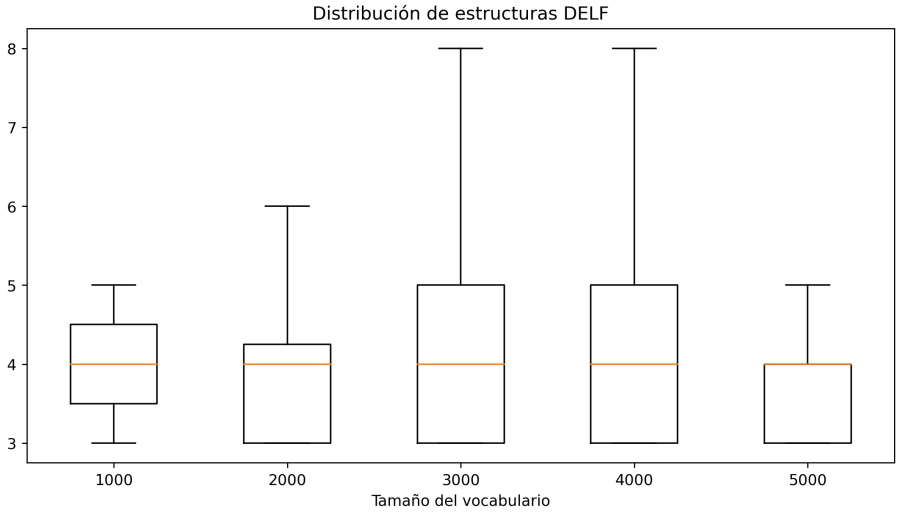


Figura A.14: Gráficos de dispersión estructuras DELF: 3 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

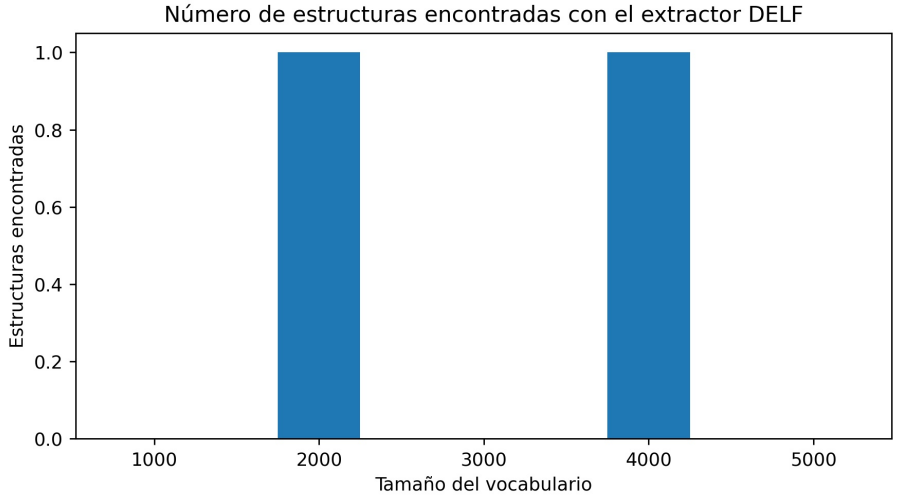


Figura A.15: Estructuras DELF encontradas: 4 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

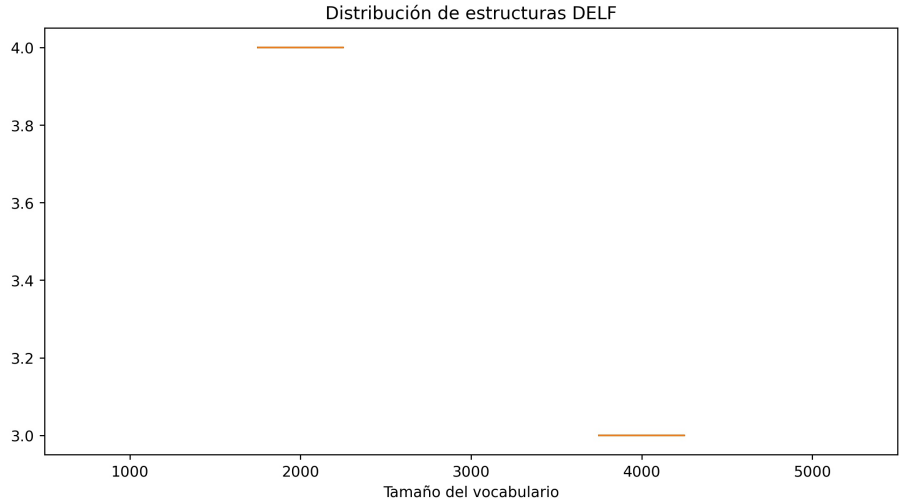


Figura A.16: Gráficos de dispersión estructuras DELF: 4 valores Min-hash y funciones min-hash proporcionales al tamaño de vocabulario

Bibliografía

[1] Barchi, P., de Carvalho, R., Rosa, R., Sautter, R., Soares-Santos, M., Marques, B., Clua, E., Gonçalves, T., de Sá-Freitas, C., and Moura, T. (2020). Machine and deep learning applied to galaxy morphology - a comparative study. *Astronomy and Computing*, 30:100334. 3

[2] Cheng, T.-Y., Huertas-Company, M., Conselice, C. J., Aragón-Salamanca, A., Robertson, B. E., and Ramachandra, N. (2021). Beyond the hubble sequence – exploring galaxy morphology with unsupervised machine learning. *Monthly Notices of the Royal Astronomical Society*, 503(3):4446–4465. 3

[3] Feizollah, A., Anuar, N., Salleh, R., and Amalina, F. (2014). Comparative study of k-means and mini batch k- means clustering algorithms in android malware detection using network traffic analysis. 42

[4] Fuentes-Pineda, G., Koga, H., and Watanabe, T. (2011). Scalable object discovery: A hash-based approach to clustering co-occurring visual words. *IEICE Transactions*, 94-D:2024–2035. 28, 29

[5] Gauci, A., Adami, K. Z., and Abela, J. (2010). Machine learning for galaxy morphology classification. 3

[6] Gonzalez, R. C. and Woods, R. E. (2017). *Digital Image Processing, 4Th Edition*. Pearson. 9, 17, 20, 21

[7] Harris, C., Stephens, M., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Manchester, UK.

[8] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[9] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[10] Lintott, C. J., Schawinski, K., Bamford, S., Slosar, A., Land, K. R., Thomas, D., Edmondson, E., Masters, K. L., Nichol, R. C., Raddick, J., Szalay, A. S., Andreescu, D., Murray, P. G., and vandenBerg, J. (2010). Galaxy zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410:166–178.

[11] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

[12] Mcculloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147.

[13] Na, S., Xumin, L., and Yong, G. (2010). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 63–67.

[14] Noh, H., Araujo, A., Sim, J., Weyand, T., and Han, B. (2016). Large-scale image retrieval with attentive deep local features.

[15] Pineda, G. F. and Ruíz, I. V. M. (2018). Topic discovery in massive text corpora based on min-hashing. *CoRR*, abs/1807.00938.

[16] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

[17] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[18] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[19] Schutter, A. and Shamir, L. (2015). Galaxy morphology — an unsupervised machine learning approach. *Astronomy and Computing*, 12:60–66.

[20] Weyand, T., Araujo, A., Cao, B., and Sim, J. (2020). Google landmarks dataset v2 - A large-scale benchmark for instance-level recognition and retrieval. *CoRR*, abs/2004.01804.

[21] Willett, K. W., Lintott, C. J., Bamford, S., Masters, K. L., Simmons, B. D., Casteels, K., Edmondson, E. M., Fortson, L., Kaviraj, S., Keel, W. C., Melvin, T. R. O., Nichol, R. C., Raddick, M. J., Schawinski, K., Simpson, R. J., Skibba, R. A., Smith, A. M., of Minnesota, D. T. U., of Oxford, U., Planetarium, A., of Nottingham, U., of Portsmouth, U., SepNet, de Barcelona, U. A., of Hertfordshire, U., of South Alabama, U., University, J. H., zurich, E., and of California at San Diego, U. (2013). Galaxy zoo 2: detailed morphological classifications for 304,122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 435:2835–2860.