Reconocimiento de objetos mediante imágenes

Alejandro Campos, Darian Dominguez

Facultad de Matemática y Computación
Universidad de la Habana
2021

Abstract

El Reconocimiento Óptico de Objetos es una línea de investigación dentro del procesamiento de imágenes para la que se han desarrollado muchas técnicas y metodologías. Su objetivo principal consiste en identificar un objeto a partir de una imagen digitalizada que se representa como un conjunto de píxeles.

Introducción

El reconocimiento de imágenes se refiere a la tarea de ingresar una imagen en una red neuronal y hacer que genere algún tipo de etiqueta para esa imagen. La etiqueta que genera la red corresponderá a una clase predefinida. Puede haber varias clases con las que se puede etiquetar la imagen, o solo una. Si hay una sola clase, a menudo se aplica el término «reconocimiento», mientras que una tarea de reconocimiento de varias clases se suele denominar «clasificación».

Un subconjunto de la clasificación de imágenes es la detección de objetos, donde se identifican instancias específicas de objetos como pertenecientes a una determinada clase, como animales, automóviles o personas.

Para llevar a cabo el reconocimiento / clasificación de imágenes, se debe realizar la extracción de características. Las características son los elementos de los datos que interesan y que permiten la identificación. En el caso específico del reconocimiento de imágenes, las características son los grupos de píxeles, como bordes y puntos, de un objeto que se analizarán en busca de patrones.

El reconocimiento de características (o extracción de características) es el proceso de extraer las características relevantes de una imagen de entrada para que estas características puedan analizarse. Muchas imágenes contienen anotaciones o metadatos sobre la imagen que ayudan a la red a encontrar las características relevantes.

El objetivo del presente trabajo es presentar varios algoritmos para el reconocimiento de imágenes. Y desarrollar una aplicación móvil usando flutter con el modelo que mejores resultados provea. Además, se presentan las bases para un detector de anomalías, lo cual permite dado una colección de imágenes, por ejemplo, de un mismo tipo de animal, detectar aquellas que no se corresponden con este.

Modelado

El modelado consiste en la transformación de una foto en un vector que los algoritmos reciben como parámetro. Para esto se debe realizar un pre-procesamiento a las imágenes.

El módulo *image_processing.py* realiza este proceso. Se realizan dos modelados, uno utilizando *tensorflow* para las redes neuronales y otro usando *opencv* para el resto de los algoritmos supervisados.

El procesamiento básicamente lo que hace es llevar las imágenes a vectores normalizados y con la dimensión reducida. En esta etapa también se realizan el trabajo con los directorios, guardando la ubicación de cada imagen y su categoría o clase. En la que por ejemplo una imagen con una mariposa corresponde a la etiqueta “mariposa”.

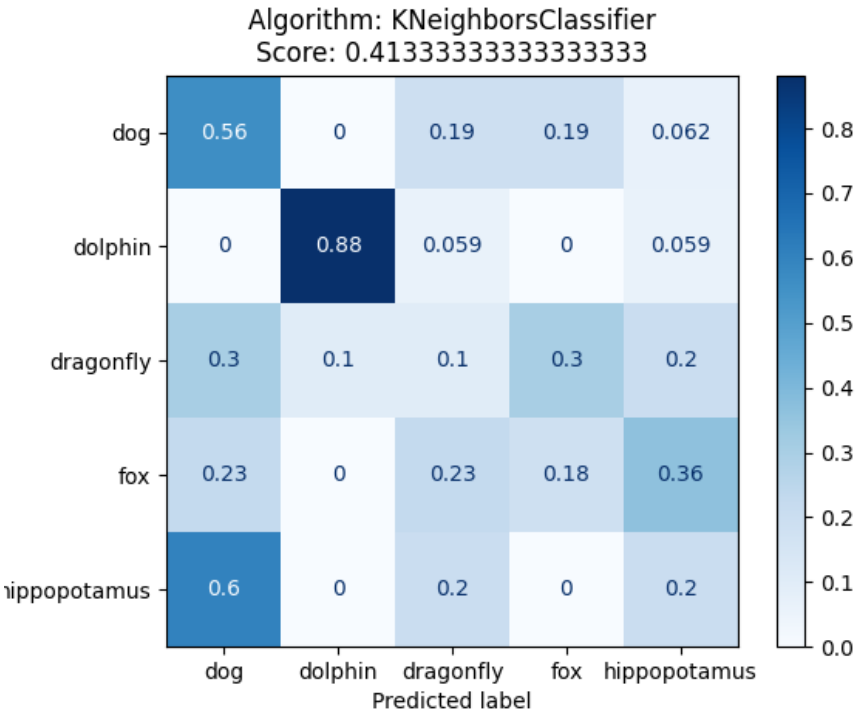
Algoritmos supervisados

Para resolver el problema en cuestión se implementaron varios algoritmos, con el fin de compararlos y crear nuestro modelo con el que resulte más conveniente de acuerdo a sus resultados.

- KNN

Es un método que busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean.

Los resultados a los que se arriban con este método son los siguientes:



Para la obtención de estos resultados se utilizó un dataset de cinco animales con 60 fotos de cada uno.

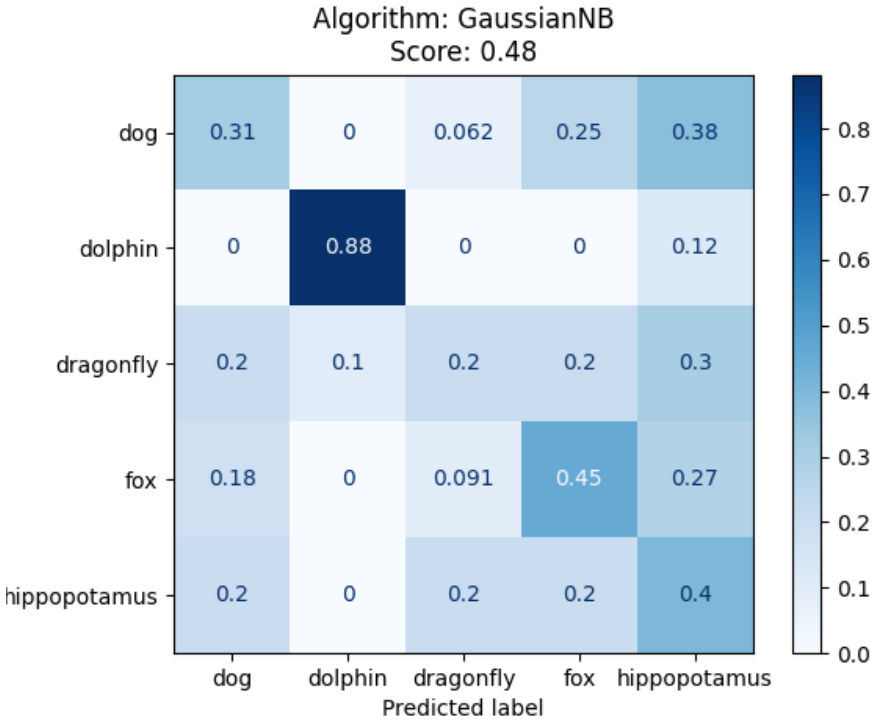
Se puede observar un score de 0.41, no es alto. Podemos notar que la mayoría de los delfines los reconoció correctamente, y poco más de la mitad de los perros. Sin embargo, el resto de los animales los confunde demasiado, haciendo asignaciones que no corresponden.

Este algoritmo no bueno para el procesamiento de imágenes, a pesar del preprocesamiento de estas, no se obtuvieron buenos resultados.

- Naive bayes

Se basa en el teorema de Bayes. Se asume que las variables predictoras son independientes entre sí. En otras palabras, que la presencia de una cierta característica en un conjunto de datos no está en absoluto relacionada con la presencia de cualquier otra característica.

Los resultados alcanzados se muestran a continuación:



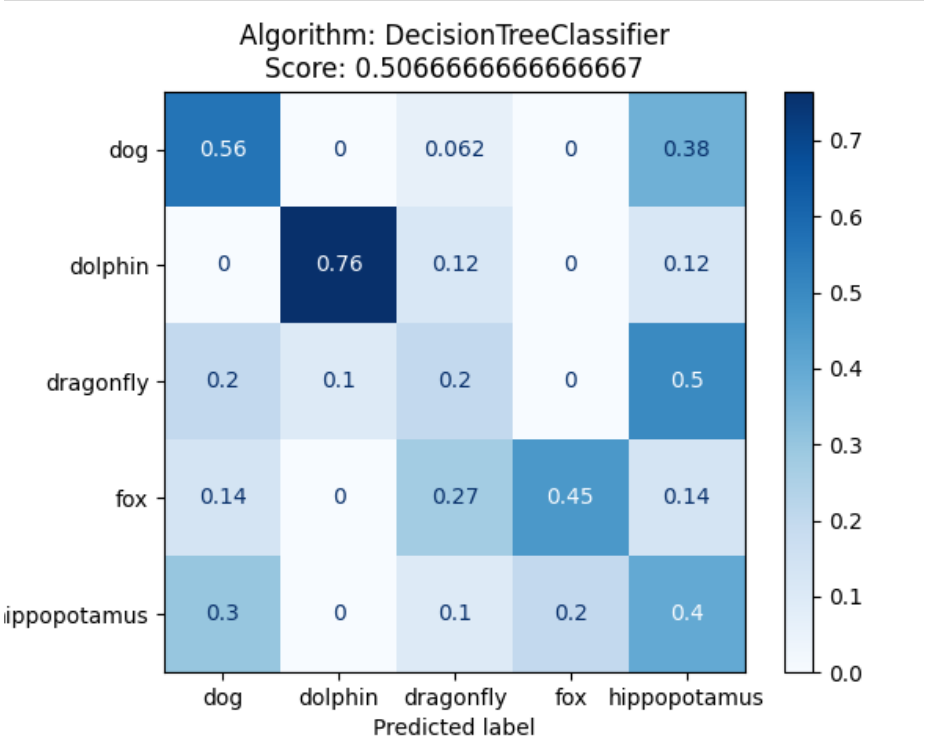
Este algoritmo supera por muy poco a KNN, lo que puede estar dado por la heterogeneidad de los animales presentados en el dataset. Con un score de 0.48, reconoce casi la mitad de los zorros, hipopótamos y, en menos cuantía, perros. Al igual que el algoritmo anterior, reconoce la mayoría de los delfines.

Este algoritmo tampoco presenta buenos resultados, por lo que no será utilizado en la construcción del modelo.

- Decision tree

En esta técnica, dividimos la data en dos o más conjuntos homogéneos basados en el diferenciador más significativos en las variables de entrada. El árbol de decisión identifica la variable más significativa y su valor que proporciona los mejores conjuntos homogéneos de población. Todas las variables de entrada y todos los puntos de división posibles se evalúan y se elige la que tenga mejor resultado.

Podemos observar en la siguiente imagen los resultados con el algoritmo en cuestión:



Al aplicar este algoritmo a nuestro conjunto de datos obtenemos un score de 0.51 aproximadamente, superando a los dos algoritmos presentados anteriormente. Se logra reconocer a poco más de la mitad de los perros y la mayoría de los delfines, sin embargo, no logra reconocer las libélulas. Y reconoce casi la mitad del resto de los animales.

Con un score de 0.51 aún no es suficiente para presentar un modelo de reconocimiento de objetos.

- Random forest

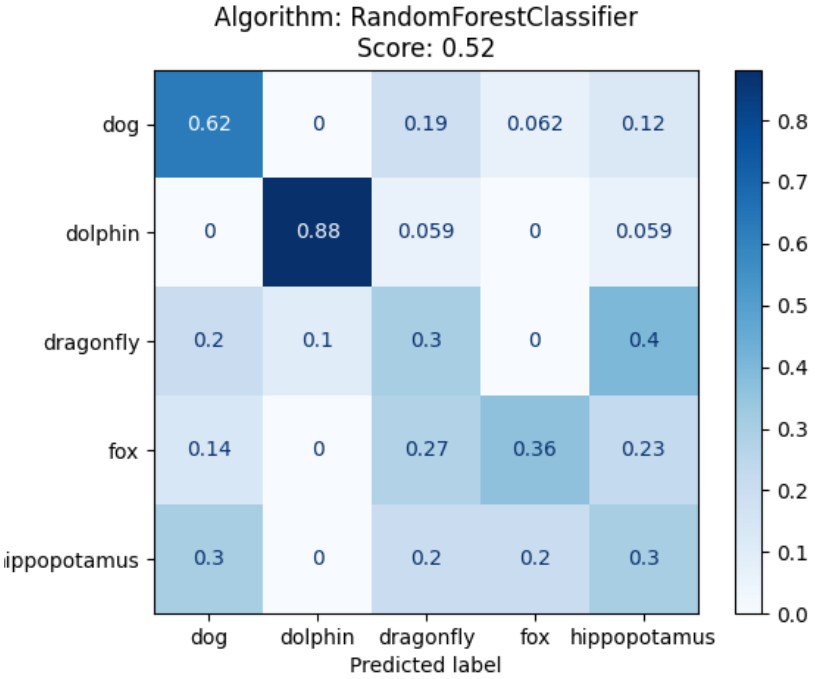
Es una técnica de aprendizaje supervisado que genera múltiples árboles de decisión sobre un conjunto de datos de entrenamiento: los resultados obtenidos se

combinan a fin de obtener un modelo único más robusto en comparación con los resultados de cada árbol por separado. Cada árbol se obtiene mediante un proceso de dos etapas:

-Se genera un número considerable de árboles de decisión con el conjunto de datos. Cada árbol contiene un subconjunto aleatorio de variables m (predictores) de forma que m < M (donde M = total de predictores).

-Cada árbol crece hasta su máxima extensión.

Finalmente, observamos los resultados a los que se arribaron:



Presenta ligeramente mejores resultados, reconociendo más del 30 por ciento de animalese en todos los casos, llegando a la mayoría en el caso del delfín y el perro, aún asi, no presenta buenos resultados para el procesamiento de imágenes.

Podemos concluir que los algoritmos anteriores no son buenos a la hora de procesar imágenes. Pues si bien los resultados no son catastróficos, tampoco son lo suficientemente buenos para trabajar con ellos.

La literatura afirma que los mejores métodos para el procesamiento de imágenes en Python es el uso de redes neuronales, con ayuda de las librerías de keras y tensorflow para el procesamiento de imágenes. En este informe hemos arribado a las mismas conclusiones.

- Redes neuronales

Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona.

Como ya se dijo para crear el modelo se usará tensorflow y keras. Tensorflow compila muchos algoritmos y modelos diferentes juntos, lo que permite al usuario implementar redes neuronales profundas para su uso en tareas como el reconocimiento / clasificación de imágenes y el procesamiento del lenguaje natural. Por otro lado, Keras hace que la implementación de las muchas funciones poderosas, pero a menudo complejas de tensorflow sea lo más simple posible.

La red neuronal más utilizada para el procesamiento y clasificación de imágenes es la red neuronal convolucional, de esta forma, utilizamos la clase de tensorflow ImageDataGenerator, un generador que nos ahora tener todas las imágenes cargadas en memoria porque es capaz de ir escaneando las carpetas y recuperando los archivos por lotes a medida que se necesiten.

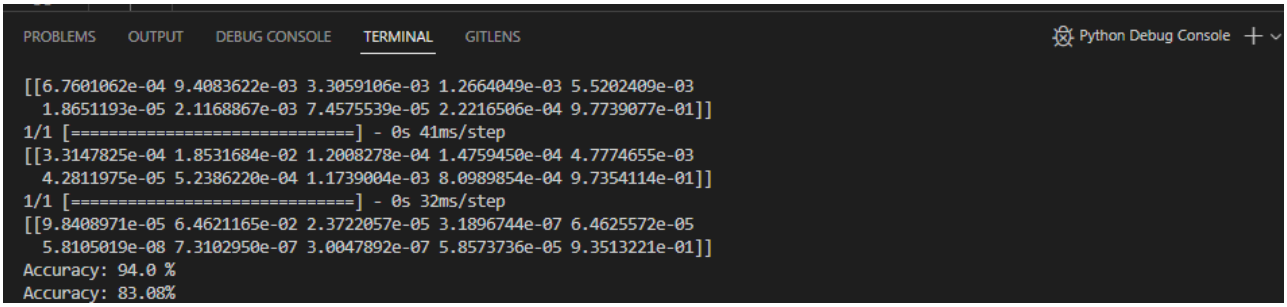
Luego, partimos de la red neuronal InceptionV3, que es una de las más populares para clasificación de imágenes, y le añadimos nuestras propias capas que permiten la clasificación de diversos tipos de imágenes según el dataset.

Por último, entrenamos nuestro modelo con los datos de entrenamiento y lo evaluamos con los datos de prueba.

Se realizaron dos modelos, uno de reconocimiento de animales, y otro de reconocimiento de objetos en general. Es necesario aclarar que para crear un nuevo modelo solo es necesario agregar el dataset en la carpeta datasets y cargarlo desde el main indicando su dirección.

El .h5 de nuevo modelo se guarda en la carpeta models, que luego es utilizado por flutter para la interfaz de usuario en la aplicación móvil.

Los resultados de ambos modelos se muestran a continuación:



Se observa que el primer modelo presenta mejores resultados que el segundo, con un acierto del 94% de los casos. La diferencia de los modelos radica en que en el primero utilizamos una red neuronal existente y le añadimos las capas finales; en el segundo creamos todas las capas de nuestra red neuronal.

Para nuestra aplicación se usaron modelos generados por el primer modelo de red neuronal ya que, después del análisis realizado en el presente informe, es el que mejores resultados presenta.

Algoritmos no supervisados

El problema que estos algoritmos pretenden resolver es el de dado una colección de fotos, mayormente de una misma clasificación, extraer las que no pertenecen a esta.

Los métodos no supervisados son algoritmos que basan su proceso de entrenamiento en un juego de datos sin etiquetas o clases previamente definidas. Es decir, a priori no se conoce ningún valor objetivo o de clase, ya sea categórico o numérico.

Para probar estos algoritmos se creó un dataset con la mayoría de las fotos con características similares, y algunas que no pertenecen al conjunto. El objetivo de estos algoritmos es detectar estas anomalías y purificar la colección.

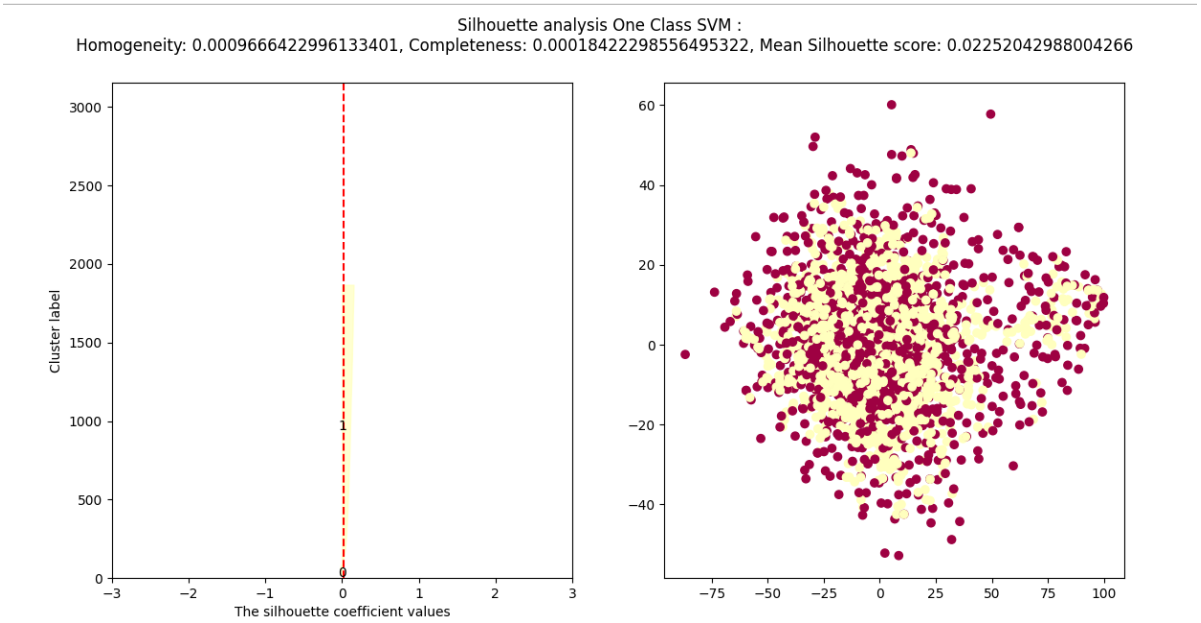
Esto tiene varias ventajas, por ejemplo, se puede utilizar para crear nuevos datasets para el trabajo con el procesamiento de imágenes con algoritmos supervisados.

A continuación se exponen los algoritmos utilizados y sus resultados:

- One Class SVM

Las SVM utilizan hiperplanos en un espacio multidimensional para separar una clase de observaciones de otra. Naturalmente, SVM se utiliza para resolver problemas de clasificación de clases múltiples.

Sin embargo, SVM también se usa cada vez más en un problema de una clase, donde todos los datos pertenecen a una sola clase. En este caso, el algoritmo está entrenado para aprender qué es “normal”, de modo que cuando se muestre un nuevo dato el algoritmo pueda identificar si debe pertenecer al grupo o no. De lo contrario, los nuevos datos se etiquetan como anormales o fuera de lo común.

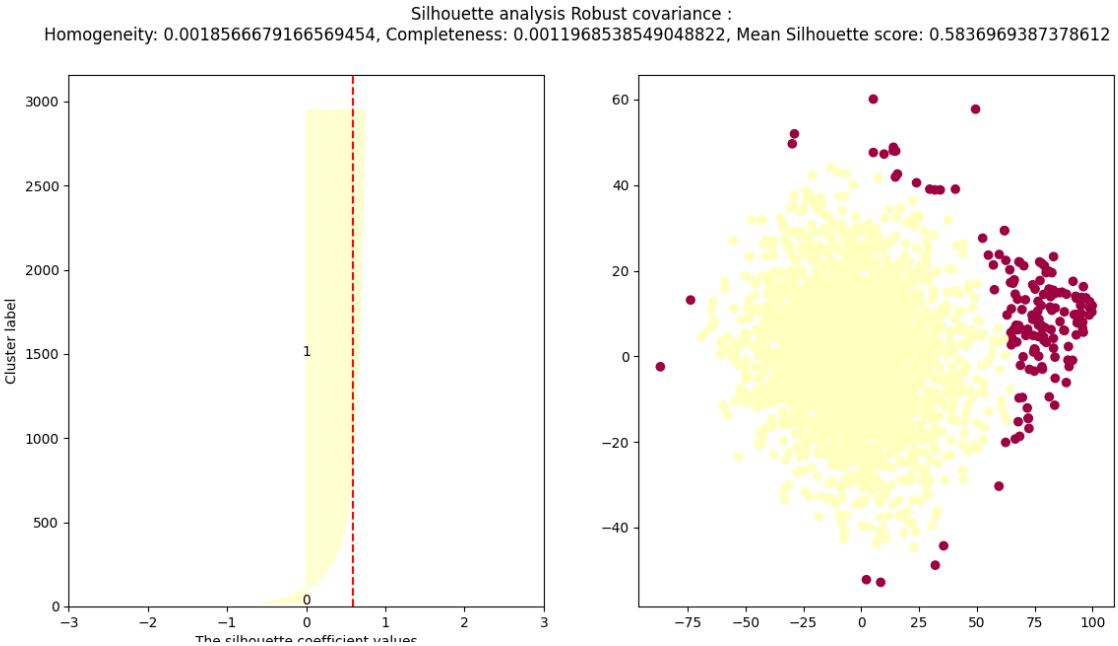


Podemos obervar que este algoritmo da realmente malos resultados, no logra definir un grupo de imágenes con características similares. Esto ocurrio para todos los datasets que se crearon con este fin, solo se muetra uno para simplicar.

- Elliptic Envelope

La filosofía que se sigue es dividir el dataset categorías. Después utilizar el modelo, que supone una distribución gaussiana con centro en los puntos normales, y anomalías en los extremos. Esto quiere decir que se pueden tener anomalías “superiores” e “inferiores”.

Los resultados para este algoritmo son:



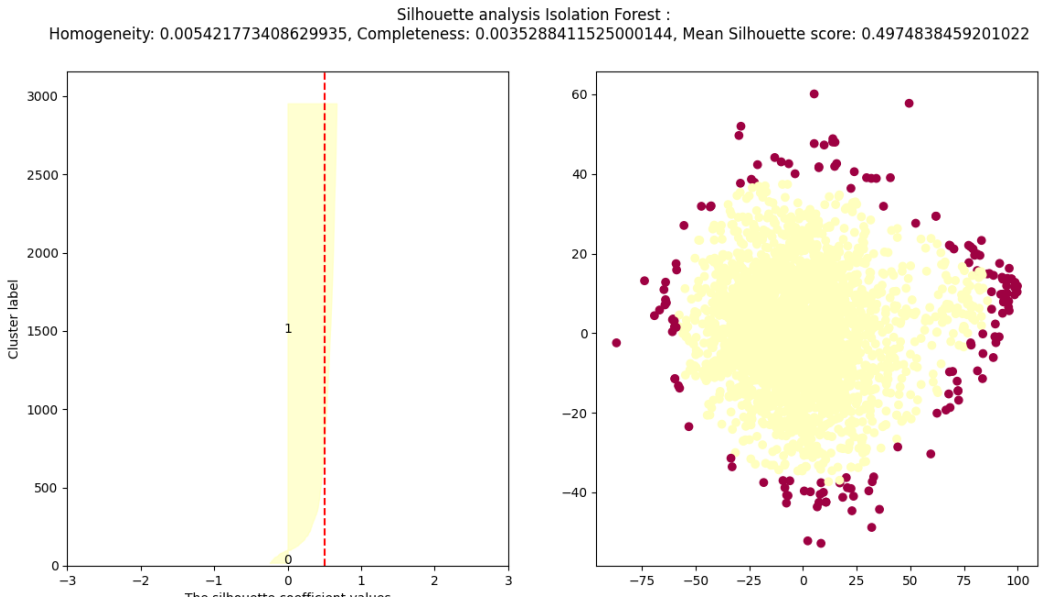
Se observa un cluster bien definido que representa las fotos con características comunes, y alrededor de este se aprecian las anomalías. Se puede observar además la forma elíptica del cluster central.

- IsolationForest

Aísla las observaciones seleccionando aleatoriamente una característica y luego seleccionando aleatoriamente un valor de división entre los valores máximos y mínimos de la característica seleccionada. Dado que la partición recursiva puede representarse mediante una estructura de árbol, el número de divisiones necesarias para aislar una muestra es equivalente a la longitud de la trayectoria desde el nodo raíz hasta el nodo final.

La partición aleatoria produce trayectorias notablemente más cortas para las anomalías. Por lo tanto, cuando un bosque de árboles aleatorios produce colectivamente longitudes de trayectorias más cortas para muestras particulares, es muy probable que se trate de anomalías.

Estos son sus resultados:



Al igual que en caso anterior se observa un cluter claramente definido y las anomalías alrededor.