

Reporte de laboratorio 3 para Inteligencia Artificial

Juan Fernando Martinez, Juan Felipe Blanco, Brian Matasca

Resumen—Por medio del presente documento, se busca describir el proceso de diseño e implementación de un modelo de inteligencia artificial que cuente con la capacidad para agrupar imágenes utilizando algoritmos de aprendizaje no supervisado.

Se exploran los retos y problemáticas presentes en el ejercicio planteado, así como el proceso seguido, los resultados obtenidos y las conclusiones del laboratorio.

Abstract—This paper seeks to describe the design and implementation process of an artificial intelligence model with the ability to cluster images using unsupervised learning algorithms.

The challenges and problems present in the proposed exercise are explored, as well as the process followed, the results obtained and the conclusions of the laboratory.

1. INTRODUCCIÓN

Este proyecto se centra en el desarrollo de un modelo de aprendizaje no supervisado para reconocer y clasificar diferentes tipos de objetos en imágenes. Para ello, se utilizarán imágenes descargadas de la web que incluyen al menos tres tipos distintos de objetos. El objetivo principal es aplicar técnicas de agrupamiento para identificar y agrupar estos objetos en clases similares basadas en sus características visuales.

El proceso comenzará con la recopilación y preprocesamiento de las imágenes descargadas, preparando los datos para su análisis. Posteriormente, se implementarán algoritmos de clustering que permitirán agrupar los objetos en clases homogéneas. Este reporte describe en detalle la metodología empleada, incluyendo la selección y preprocesamiento de las imágenes, la aplicación de los algoritmos de agrupamiento y la evaluación de los resultados obtenidos.

2. PROCESO Y PROBLEMÁTICAS

El proceso inicial de este proyecto comienza con la recopilación de imágenes de diferentes tipos de objetos descargadas de internet. Para automatizar esta tarea, se empleó un script en Python que utiliza la librería `bing-image-downloader`. Este script permite buscar y descargar imágenes de diversas categorías, en este caso, 'manzanas', 'bananos' y 'peras', asegurando una colección variada y suficiente para el análisis posterior.

Una vez descargadas las imágenes, se procede al preprocesamiento de los datos para preparar las imágenes de manera adecuada para el análisis. El preprocesamiento incluye varias etapas críticas: eliminación de ruido, normalización del tamaño y ajuste del contraste. Para eliminar el ruido de las imágenes, se utiliza un filtro `GaussianBlur`, implementado mediante la librería `OpenCV`. Este filtro suaviza las imágenes y reduce el ruido, mejorando la calidad visual de los objetos.

El ajuste del contraste se realiza utilizando la librería `Pillow`, donde se incrementa el contraste de las imágenes para resaltar las características distintivas de los objetos. Después, las imágenes se redimensionan a un tamaño uniforme, en este caso, (128, 128) píxeles, para asegurar que todas las imágenes tengan las mismas dimensiones, facilitando así el análisis.



Fig. 1. Imagen con el contraste ajustado y sin ruido

El proceso para entrenar los modelos de agrupamiento en este proyecto comienza con la segmentación y preprocesamiento de las imágenes. Utilizamos un algoritmo conocido como `GrabCut` para segmentar los objetos dentro de las imágenes, convirtiéndolas a formato RGB y aplicando una máscara para separar el objeto principal del fondo. Estas imágenes segmentadas se redimensionan a un tamaño uniforme de 128x128 píxeles. Posteriormente, las imágenes se convierten a vectores aplanados para facilitar el procesamiento y

análisis.

Esta etapa fue necesaria debido a un problema que encontramos al intentar agrupar los objetos. Los algoritmos de agrupamiento tendían a agrupar las imágenes según el fondo en lugar del objeto principal. Por ejemplo, el clúster 1 contenía imágenes con fondo blanco y el clúster 2, imágenes con fondo negro, independientemente del objeto presente. Para solucionar esto, aplicamos el algoritmo GrabCut para segmentar el objeto del fondo, asegurando que los algoritmos de agrupamiento se enfocarán en las características del objeto en lugar del fondo.

Una vez pre-procesadas las imágenes, se aplicaron técnicas de reducción de dimensionalidad para extraer características relevantes de los datos. Se utilizó el Análisis de Componentes Principales (PCA) y el Análisis de Componentes Principales con Kernel (KPCA) para este propósito. PCA es una técnica lineal que proyecta los datos en un espacio de menor dimensión, manteniendo la mayor cantidad de varianza posible. En nuestro caso, se redujo la dimensionalidad a 25 componentes. KPCA, una extensión no lineal de PCA, utiliza funciones kernel para mapear los datos a un espacio de alta dimensión antes de realizar la reducción, permitiendo capturar relaciones más complejas en los datos.

Con los datos transformados mediante PCA y KPCA, se entrenaron dos modelos de agrupamiento distintos: KMeans y DBSCAN. KMeans es un algoritmo de agrupamiento basado en centroides que particiona los datos en un número predefinido de clusters, minimizando la variación dentro de cada clúster. DBSCAN, por otro lado, es un algoritmo basado en densidad que agrupa puntos cercanos y puede identificar clusters de forma arbitraria, siendo particularmente útil para detectar ruido y estructuras no lineales en los datos.

Para evaluar la calidad de las agrupaciones obtenidas con ambos modelos, se calcularon dos métricas: el puntaje de Silhouette y el índice de Calinski-Harabasz. El puntaje de Silhouette mide qué tan similares son los objetos dentro de un cluster en comparación con los objetos de otros clusters, mientras que el índice de Calinski-Harabasz evalúa la dispersión de los clusters. Estas métricas nos permitieron comparar la efectividad de los métodos PCA y KPCA en combinación con los modelos KMeans y DBSCAN.

el desempeño de los modelos de agrupamiento utilizando PCA y KPCA. Comenzando con el modelo KMeans, se observa que el puntaje de Silhouette es significativamente más alto cuando se utiliza KPCA en comparación con PCA (0.694 vs. 0.137). Esto indica que las instancias dentro de cada cluster están más cohesionadas y mejor separadas entre sí cuando se emplea KPCA para la reducción de dimensionalidad. Además, el índice de Calinski-Harabasz también es más alto con KPCA, lo que sugiere una mayor cohesión intraclase y una mejor separación interclase en los grupos obtenidos. Estos hallazgos respaldan la eficacia de KPCA para capturar las relaciones no lineales en los datos y generar agrupaciones más coherentes.

Por otro lado, el modelo DBSCAN también muestra un rendimiento superior cuando se utiliza KPCA en comparación con PCA. El puntaje de Silhouette es más alto con KPCA (0.831 vs. no cluster suficientes), lo que indica una mejor calidad de la agrupación en términos de separación y cohesión de los clusters. Similar al modelo KMeans, el índice de Calinski-Harabasz es más alto con KPCA, lo que sugiere una mayor coherencia dentro de los clusters y una mejor discriminación entre ellos. Estos resultados sugieren que DBSCAN puede beneficiarse significativamente de la representación de datos no lineal proporcionada por KPCA, mejorando así su capacidad para identificar estructuras complejas y densidades variables en los datos.

Resultados del modelo KMeans con KPCA:
Silhouette Score: 0.69457078722606
Calinski-Harabasz Score: 390.30067446079795

Resultados del modelo DBSCAN con KPCA:
Silhouette Score: 0.8319384795304623
Calinski-Harabasz Score: 428.8834360778836

Resultados del modelo KMeans con PCA:
Silhouette Score: 0.13749467672072002
Calinski-Harabasz Score: 190.8556594311049

Resultados del modelo DBSCAN con PCA:
DBSCAN con PCA no encontró suficientes clusters para evaluación

Fig 2. Métricas obtenidas de los modelos entrenados

3. ANÁLISIS DE RESULTADOS

Los resultados obtenidos muestran una clara diferencia en

Sin embargo, los resultados obtenidos al guardar las imágenes relacionadas con cada cluster revelaron una discrepancia entre los puntajes de evaluación y la calidad real de la agrupación. A pesar de que los modelos con KPCA obtuvieron puntajes de Silhouette Score más altos, se observó que no lograron una agrupación efectiva cuando se considera la distribución de imágenes en cada clúster. Específicamente, los modelos KMeans y DBSCAN con KPCA generaron clusters con un número desigual de imágenes, lo que indica una falta de coherencia en la agrupación. Por otro lado, el modelo KMeans con PCA logró una distribución más equitativa de imágenes entre los clusters, sugiriendo una mejor capacidad para discriminar y separar los diferentes tipos de objetos en las imágenes.

KMeans con PCA:

Cluster 0: 260 imágenes
Cluster 1: 514 imágenes
Cluster 2: 361 imágenes

KMeans con KPCA:

Cluster 0: 1093 imágenes
Cluster 1: 38 imágenes
Cluster 2: 4 imágenes

DBSCAN con KPCA:

Cluster 0: 1074 imágenes
Cluster 1: 38 imágenes
Cluster -1: 23 imágenes

En particular, al observar las imágenes almacenadas en los clusters obtenidos con el modelo KMeans y PCA, se evidenció una clasificación más precisa de los objetos. El cluster 1 Fig.3, que contiene un total de 260 imágenes, demostró una buena capacidad para clasificar manzanas, aunque se encontró cierta confusión al distinguir peras y bananos debido a ciertas similitudes en el color. Sin embargo, se pudo identificar que el cluster 3 estaba principalmente dedicado a las peras, mientras que el cluster 2 se asociaba en su mayoría con los bananos, a pesar de algunas dificultades para distinguirlos claramente.

Este análisis revela la importancia de complementar las métricas de evaluación con una inspección visual de los resultados, ya que los puntajes de evaluación pueden no reflejar completamente la calidad de la agrupación. En este caso, aunque los modelos con KPCA obtuvieron puntajes más altos en las métricas de evaluación, el modelo KMeans con PCA demostró una mejor capacidad para agrupar efectivamente las imágenes según los

diferentes tipos de objetos, lo que subraya la importancia de considerar múltiples aspectos al evaluar y seleccionar modelos de agrupamiento para aplicaciones prácticas.

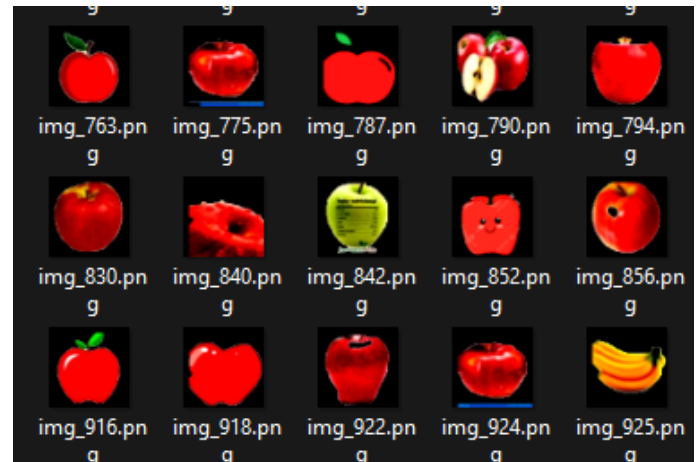


Fig 3. Resultados cluster 1 KMeans con PCA

4. CONCLUSIONES

El proyecto enfatiza la relevancia del pre-procesamiento de datos y la selección de características para mejorar la calidad de la agrupación en modelos de aprendizaje no supervisado. La evaluación completa de los modelos, integrando métricas cuantitativas y análisis visual, resulta esencial para comprender plenamente su rendimiento.

La agrupación de las manzanas con el modelo KMeans y PCA fue lo más destacable del proyecto, el algoritmo fue capaz de separar las manzanas de las otras frutas. Sin embargo, el modelo al intentar agrupar los bananos y las peras cometió más errores. Esto se puede explicar debido a que la gran parte de las peras del dataset tienen un color amarillento confundiendo con el amarillo de los bananos. Lo ideal para futuros proyectos sería intentar clasificar objetos que tengan características más distinguibles para que el modelo pueda hacer mejores agrupaciones o en su defecto utilizar técnicas más robustas a la hora de extraer y seleccionar las características de las imágenes.

Repositorio con el código utilizado para el proyecto:
<https://github.com/Br14nMat/LAB3-IA>