Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-5438 Inteligencia Artificial II
Trimestre Septiembre - Diciembre 2023

Informe Proyecto 3

Estudiantes:

Br. Gamboa, Roberto

Br. Blanco, Luis

Carnés:

16-10394

17-10066

Profesor

Carlos Infante

7 de diciembre de 2023

Índice

Introducción

El algoritmo K-means implementa una técnica fundamental para la agrupación eficaz de los datos. Dicho algoritmo es conocido por ser muy eficiente al momento de clasificar datos que no se encuentren etiquetados, haciendo una división bastante coherente de los mismos.

En el presente proyecto se realizará la implementación del algoritmo K-means, haciendo diferentes pruebas para comprobar su eficacia. Se realizarán pruebas con el iris dataset utilizado en el proyecto anterior, así como la segmentación y renderización de imágenes.

Es una herramienta bastante poderosa a la hora de tener que clasificar datos, a lo largo de este proyecto y luego de realizar las pruebas mencionadas anteriormente podremos verificar lo poderoso que es este algoritmo.

Link al repositorio del proyecto

El proyecto se encuentra alojado en el siguiente enlace:

`https://github.com/luiss-23/CI5438-Proyecto3`

Implementación

El algoritmo de K-Means fue implementado en el lenguaje de programación Python. Se siguió el pseudocódigo dado en clases y se complementó con los códigos que se encuentran en:

- "Create a K-Means Clustering Algorithm from Scratch in Python"
- "K-Means Clustering From Scratch in Python [Algorithm Explained]"
- "Implementing K-Means Clustering from Scratch in Python"

Se utilizó la librería numpy para realizar los cálculos requeridos en el algoritmo de una forma cómoda y sencilla. Además para realizar las gráficas se usó matplotlib. Para calcular las distancias entre cada punto y cada centroide se utiliza el método cdist de la librería scipy.

Para inicializar los centroides, se escogen k puntos al azar de los datos suministrados del problema, se realiza de esta forma ya que si se inicializan muy lejos o muy cerca de los datos del problema, es probable que el algoritmo no funcione correctamente.

Para la implementación del algoritmo K-Means, utiliza la clase `KMeans`, que necesita los siguientes parámetros para ser instanciada:

- `k`: representa la cantidad de centroides que se quieren usar para la ejecución del algoritmo
- `max_iters`: representa la cantidad máxima de iteraciones que se desea se hagan en la ejecución del algoritmo. Por defecto se asigna un valor de 100 iteraciones máximas

Por ejemplo, si se desea ejecutar el algoritmo usando 5 centroides y con un máximo de 20 iteraciones, se haría de la siguiente forma:

```
k_obj = KMeans(5, 20)
```

La clase cuenta con los métodos `k-means`, implementación de dicho algoritmo y el método `etiquetar`, que dado un arreglo X de datos, le asigna una etiqueta a cada punto p de X señalando a qué cluster pertenece p.

El algoritmo de K-Means cuenta con un criterio de convergencia que compara el resultado de cada iteración con el de la iteración anterior, deteniendo su ejecución si los centroides obtenidos son iguales a los de la iteración anterior. En cada iteración del algoritmo se calcula la distancia de cada punto de los datos a cada centroide y se obtiene un arreglo de "etiquetas" que indica por cada punto a cual centroide éste es más cercano mediante el método `etiquetar` de la clase.

Adicionalmente se tiene un arreglo de k centroides, los valores de cada centroide son actualizados en cada iteración con la media de la cantidad de puntos que le fueron asignados. Se incluyó una variable para calcular la inercia en cada iteración del algoritmo, que es una medida de la dispersión de los puntos de datos dentro de sus respectivos clusters, con el objetivo de poder aplicar el "método del codo"para hallar la cantidad k de clusters óptima para cada problema. A pesar que para los dos problemas estudiados no es necesario aplicar el método del codo, se considera que es una buena adición al algoritmo de K-Means.

Al finalizar la ejecución del algoritmo de K-Means se obtiene un arreglo con las coordenadas de k centroides que clasifican los puntos cercanos a si mismos en clusters de datos. Se puede afirmar en la mayoría de los casos que si el punto p tiene como centroide más cercano a k, entonces p pertenece al cluster que k representa.

Iris Dataset

Para utilizar los datos del dataset iris se trabajó de manera similar a la del proyecto 2. Utilizando métodos de la librería pandas se lee el archivo iris.csv, para poder trabajar con datos no etiquetados se separa la columna `species` de los datos y se ejecuta el algoritmo de K-Means. Posterior a la ejecución del algoritmo se grafican los puntos de datos utilizados junto con los centroides para poder apreciar visualmente en qué lugar se ubican.

Al realizar las pruebas con $k = 2$, $k = 3$, $k = 4$ y $k = 5$ se puede apreciar que los mejores

resultados se obtienen al usar $k = 3$ ya que con este valor de k se obtienen 3 centroides, lo cual concuerda con la cantidad clasificaciones que se tienen el en dataset iris. También se obtiene un alto valor de precisión para este valor de k. Las imágenes correspondientes a los resultados de estas pruebas se pueden encontrar en el directorio `img/iris_results` del repositorio.

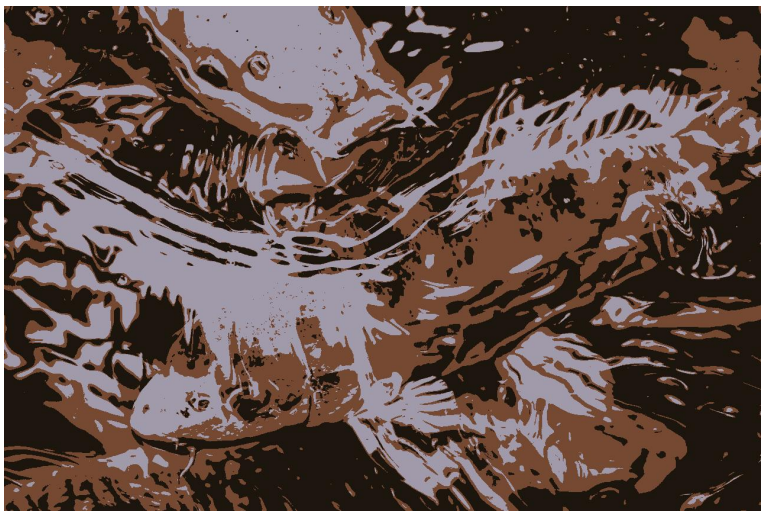
Segmentación de imágenes

Para cargar y manipular las imágenes, para obtener data que pueda ser utilizada por el algoritmo de K-Means se sigue el procedimiento descrito en "Image Segmentation using K Means Clustering" con las librerías opencv-python y matplotlib.

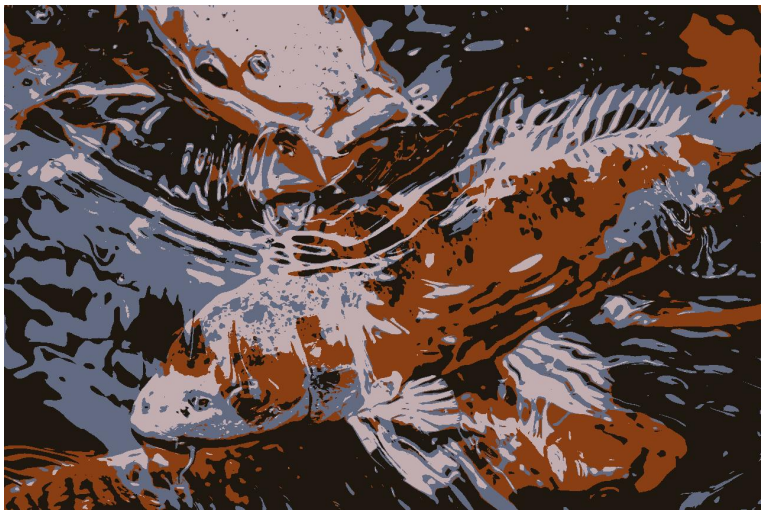
Una vez procesada la data de las imágenes, se ejecuta el algoritmo k-means con k centroides y se obtienen k clases de colores, luego se renderiza la imagen con las k clases de colores.

En el directorio `img/real_img` ubicado en la raíz del proyecto de incluyen diversas imágenes con las que se probó el algoritmo, centrándonos en las 2 imágenes `koi.jpg` y `surgery.jpg` con $k = 3, 4, 5, 6, 7, 8, 9$ se obtuvieron los siguientes resultados:

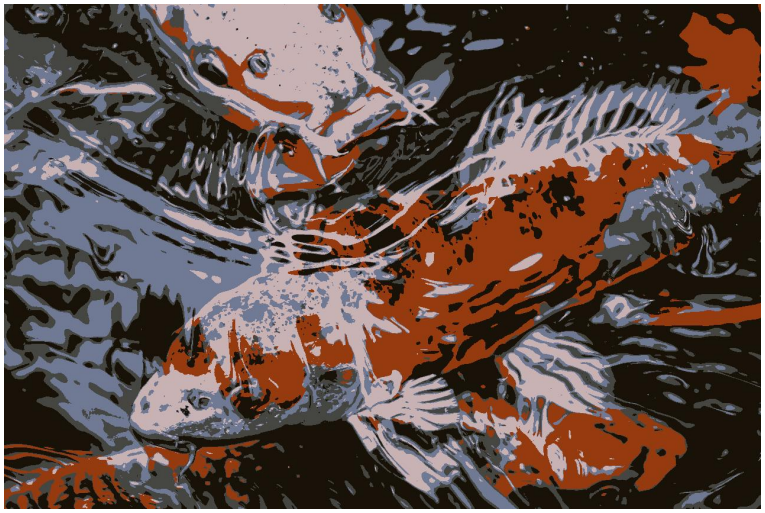
- Para la imagen `koi.jpg`
  - $k = 3$



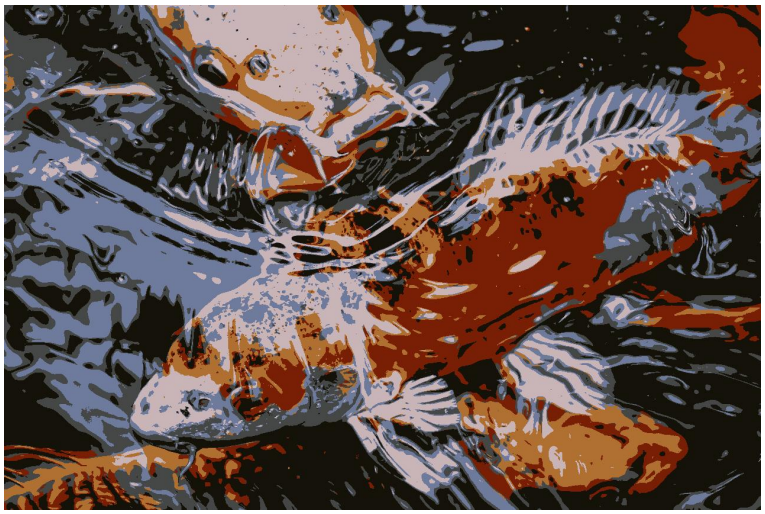
- $k = 4$



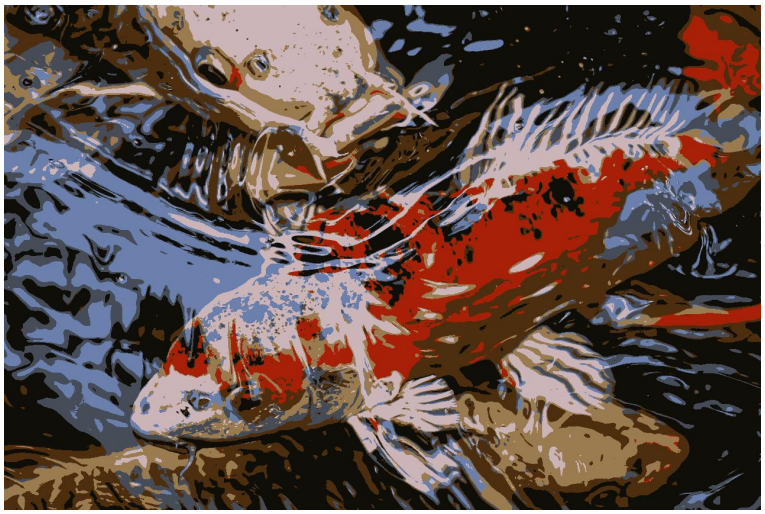
- $k = 5$



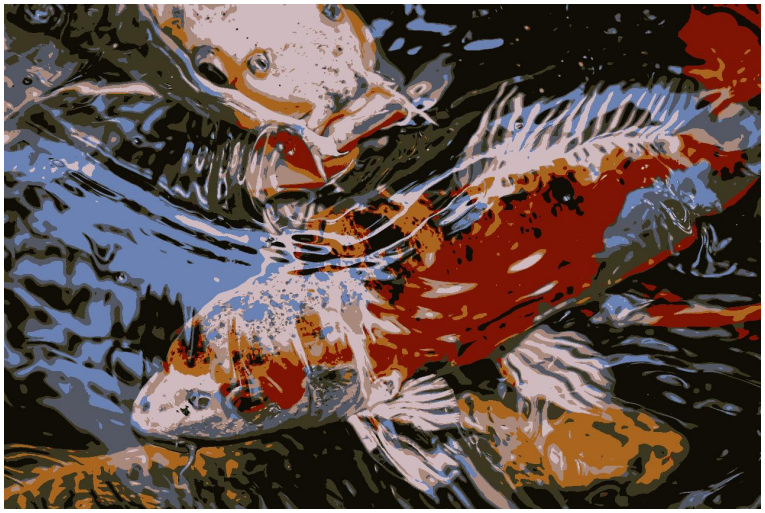
- $k = 6$



- $k = 7$



- $k = 8$



- $k = 9$

- Para la imagen `surgery.jpg`

  - $k = 3$

  - $k = 4$

  - $k = 5$

- $k = 6$

- $k = 7$

- $k = 8$

- $k = 9$

Se puede apreciar que al aumentar la cantidad k de clusters, se pueden apreciar más colores en la imagen y se pueden distinguir de forma mas detallada los objetos de la misma.

Ejecución del proyecto

El proyecto fue realizado en el lenguaje de programación Python, en su versión 3.11.2. Para ejecutar el proyecto inicialmente se requiere instalar las librerías necesarias, especificadas en el archivo `requirements.txt` ubicado en la carpeta raíz del repositorio del proyecto, las mismas pueden instalarse mediante el comando

```
$ pip install -r requirements.txt
```

Para el Iris Dataset basta con acceder al directorio `src` y ejecutar el archivo `iris.py` con el comando

```
$ python3 iris.py [i]
```

donde:

- i: indica si se quiere graficar la inercia y observar con el método del codo el k óptimo. Si no se incluye este parámetro no se muestra la gráfica.

Para la segmentación de imágenes basta con acceder al directorio `src` y ejecutar el archivo `segmentacion.py` con el comando

```
$ python3 segmentacion.py imagen k
```

donde:

- imagen: indica el nombre de la imagen que se quiere segmentar, la imagen debe estar ubicada en el directorio `img/real_img` desde la raíz del proyecto y se debe incluir la extensión del archivo. Ejemplo: mask.jpg
- k: indica la cantidad de centroides que se quieren calcular. k puede ser un valor entero o un intervalo de valores enteros de la forma 2-5.

Conclusión

En conclusión, este proyecto de implementación del algoritmo K-means ha demostrado ser una herramienta valiosa para la agrupación eficiente de datos. A través de la aplicación de este algoritmo, hemos logrado identificar patrones en conjuntos de datos no etiquetados. La flexibilidad y escalabilidad del K-means lo posicionan como una elección destacada en diversas áreas.

En comparación a los métodos estudiados en los proyectos anteriores, el algoritmo K-means demostró ser un poco más eficiente que las redes neuronales, para el caso de la clasificación en el iris dataset. En el caso de la segmentación de imágenes, se puede observar que es bastante confiable y eficiente.

Recomendaciones

Como recomendaciones, tenemos que es muy valioso utilizar el método del codo, ya que nos permite tener una muy buena idea sobre el valor de k con el que se debe ejecutar el algoritmo.