Kristian Stoyanov Belkin, Paul Daniel Pop i Raúl Jiménez Ayza

NIU: 1667799, 1607129 i 1688916

**Proyecto 2 Parte 3 ETIQUETATGE Inteligencia artificial**

Contenido

[Introducción 2](#_Toc166444150)

[Funciones de análisis cualitativo 2](#_Toc166444151)

[Retrieval\_by\_color: 2](#_Toc166444152)

[Retrieval\_by\_shape: 2](#_Toc166444153)

[Retrieval\_combined: 3](#_Toc166444154)

[Funciones de análisis cuantitativo 3](#_Toc166444155)

[Kmean\_statistics: 3](#_Toc166444156)

[Get\_shape\_accuracy: 3](#_Toc166444157)

[Descripción de los experimentos 3](#_Toc166444158)

[Mejoras o cambios k-means y KNN 3](#_Toc166444159)

[Kmeans ++ 3](#_Toc166444160)

[Coeficiente de Fisher 3](#_Toc166444161)

[Find\_BestK: 3](#_Toc166444162)

[Visualización 4](#_Toc166444163)

[Resultats preliminars 4](#_Toc166444164)

# Introducción

En este segunda practica tenemos la tarea general de realizar un etiquetaje automático de imágenes de ropa, a través de nuestro código deberíamos poder asignar etiquetas a diferentes tipos de ropa (8) con sus respectivos colores (11).

Para realizar estas tareas utilizamos dos archivos para dos algoritmos llamados KNN (K-nearest neighbors o k vecinos más próximos) y kmeans (k-means clustering o K-medias). A continuación, explicaremos un poco más en detalle ambos algoritmos:

Kmeans es un método de clasificación no supervisada, que utilizamos para generar clústeres a partir de la información que tenemos de los colores, para poder agrupar los pixeles de cada imagen y así determinar los colores predominantes.

KNN a diferencia del Kmeans es un método de clasificación supervisado, que utilizamos para encontrar el tipo de ropa de cada imagen, lo que acaba haciendo es buscar K vecinos más cercanos y se asigna el tipo mirando por la mayoría de los vecinos más cercanos.

# Funciones de análisis cualitativo

## Retrieval\_by\_color:

Retrieval\_by\_color permite buscar una serie de imágenes por los valores de su color.

Texto

Descripción generada automáticamente

Lo que realizamos en esta función es pasar por todos los kmeans tags y enumeramos cada iteración con x. Luego iteramos por todos los elementos y en caso de coincidencia guardamos la imagen en una lista que creamos previa al bucle. Finalmente ejecutamos visualize\_retrieval para ver el resultado.

## Retrieval\_by\_shape:

Retrieval\_by\_shape permite buscar una serie de imágenes por los valores de su forma.

Texto

Descripción generada automáticamente

## Retrieval\_combined:

La función retrieval\_combined permite buscar imágenes que los criterios coinciden con la forma y color que hemos intentado buscar.

Texto

Descripción generada automáticamente

/\* Por acabar de hacer, actualmente no funciona\*/

# Funciones de análisis cuantitativo

## Kmean\_statistics:

Debería ser una función que recibe la clase kmeans con una serie de imágenes y número de K que representa el máximo que queremos analizar de K=2 hasta K=max. Ejecutará la función fit, calculará el WCD, el número de iteraciones/tiempo etc.

/\* La función está por hacer \*/

## Get\_shape\_accuracy:

Esta función recibe los tags y el ground\_truth y debería devolver el porcentaje de etiquetado correcto.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Lo primero que hacemos es comprobar que los tags sean del mismo tamaño que el ground\_truth, en caso contrario saltamos un error. En la variable total guardamos la longitud de los tags para posteriormente sacar el porcentaje y empezamos a comprobar en correct la cantidad de tags que se predicen correctamente sumando 1 por cada correcto. Finalmente devolvemos el porcentaje que se saca dividiendo los correctos entre el total y multiplicando por 100 (en caso de no tener una lista de tags vacíos).

# Descripción de los experimentos

# Mejoras o cambios k-means y KNN

Estamos planteado implementar y testear las siguientes tres mejoras:

## Kmeans ++

## Coeficiente de Fisher

## Find\_BestK:

# Visualización

# Resultats preliminars