P2.ETIQUETADOR

INTEL·LIGÈNCIA ARTIFICIAL

Marc Belmonte Alcázar (NIU: 1633672)
Arnau Trucharte Vila (NIU: 1634802)
Oriol Muñoz Zapater (NIU: 1632753)

Índex

1. Introducció a la pràctica

En aquesta pràctica resoldrem un problema d'etiquetatge d'imatges de roba per tipus de peça i color. Es fa servir un conjunt reduït d'etiquetes, que consisteix en 8 tipus de peces de roba i 11 colors bàsics.

El sistema retornarà una etiqueta de tipus de peça de roba i una o varies etiquetes de color a partir d'una imatge donada.

S'utilitzen els algorismes K-means, que farem servir per trobar els colors predominants i el K-nn, que farem servir per etiquetar tipus de roba.

- K-means: és un algoritme d'agrupament que divideix dades en k clusters. S'inicia amb un número determinat de centroids, després es reassignen els punts als centroids més propers i es recalculen els centroids.
- KNN: troba els k veïns més propers a una mostra desconeguda i prediu la seva classe o valor objectiu basant-se en les etiquetes d'aquests veïns.

La pràctica està dividida en tres parts:

a) Kmeans i color

En la primera part hem programat les funcions d'inicialització dels diferents paràmetres que hem anat necessitant, així com les funcions que es necessiten per elaborar l'algorisme K-means i les que treballen directament amb l'algorisme.

b) KNN i Forma

Hem programat les funcions de l'algorisme NN, tant les que necessitem per inicialitzar les diferents variables que farem servir en l'elaboració de l'algorisme com les que treballen amb el KNN directament.

c) Millores i avaluació

En aquesta última part ens hem centrat en fer millores per tal d'optimitzar el codi, i a elaborar l'informe i presentació del projecte.

2. Mètodes d'anàlisi implementats

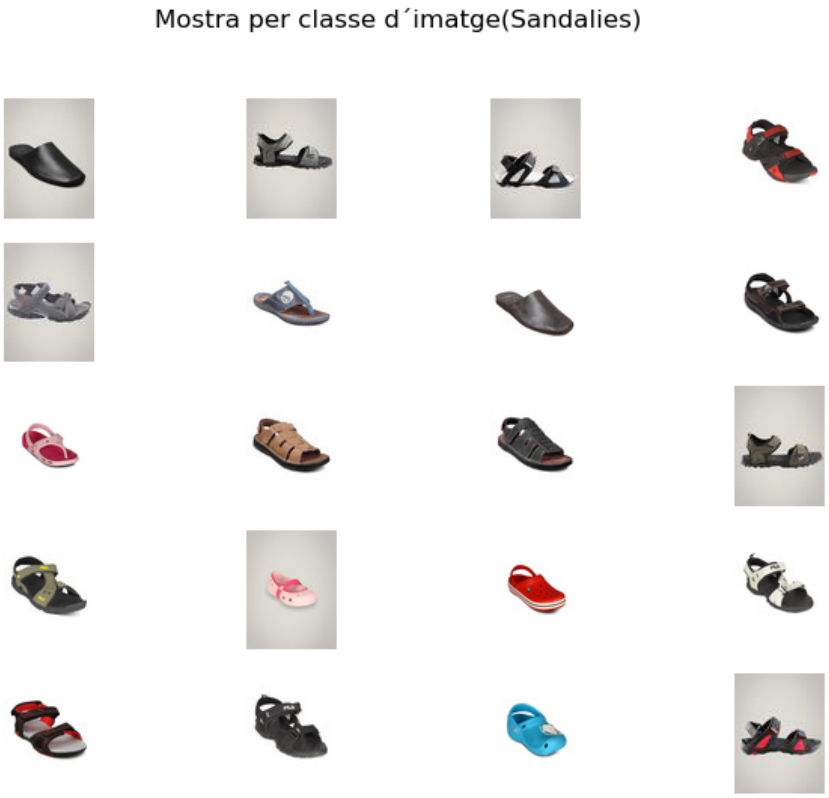
Les funcions que hem utilitzat per avaluar el nostre projecte són:

-retrieval_by_shape:

Aquesta funció, rep com a paràmetre una llista amb totes les imatges de les que disposem, les etiquetes i una pregunta. L'objectiu d'aquesta funció, és escollir i classificar a través del tipus de peça de roba i retornar les que coincideixen amb la pregunta.

Per fer això, hem fet un bucle for que per cada imatge, mira si te totes les etiquetes passades per parametre, si les té les posa a una llista i les retorna

El resultat és així:



-retrieval_by_color:

Aquesta funció, rep com a paràmetre una llista amb totes les imatges de les que disposem, les etiquetes i una pregunta. L'objectiu d'aquesta funció, és escollir i classificar a través del color que escollim les diferents imatges de les que disposem.

Per fer això, agafem l'etiqueta de la primera imatge i la comparem amb la pregunta. Si són iguals, la fiquem a la llista de resultats i la retornem

El resultat queda tal que així:

-get_color_accuracy:

Aquesta funció ha de calcular el percentatge d'èxit de la funció retrival_by_color.

Per fer-ho, passem les etiquetes que haurien de donar i les etiquetes que hem obtingut amb l'algoritme. Posteriorment, dividim els encerts entre el total de peces de roba passades.

```
La presició de KNN és:  100.0
```

-get_shape_accuracy:

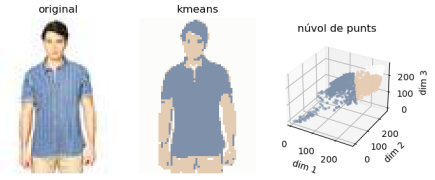
Aquesta funció ha de calcular el percentatge d'èxit de la funció retrieval_by_shape.

Per fer-ho, passem els colors obtinguts amb el K-means, i ho comparem amb els resultats que hauríem d'haver obtingut. Posteriorment, dividim els encerts entre el total de peces de roba passades.

```
Hem encertat un  84.0 % en l'etiquetatge de forma
```

-kmeans_statistic

La finalitat d'aquesta funció, és veure visualment quina és la millor k pel kmeans. Per fer això, recorrem "config" i per cada valor buscar la millor k. Després imprimim per pantalla el resultat on observem la imatge original, la imatge procesada per K-means, i finalment, el núvol de punts resultant.



3. Millores sobre Kmeans i KNN

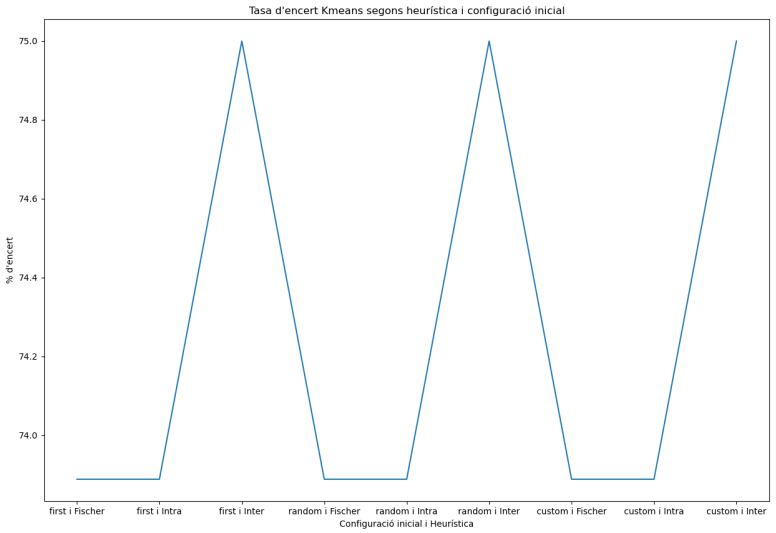
Inicializations de Kmeans

Es basa en canviar la inicialització del kmeans. En el mètode original, s'inicialitza a través del primer (first). Hem creat dos altres modes, un que inicialitza a través d'una posició aleatòria i l'altre que inicialitza a través d'una posició determinada anteriorment (custom). Per analitzar quina inicialització era més òptima hem calculat tant la precisió d'aquestes usant conjuntament les 3 diferents heurístiques, així com el temps que triga cadascun. (Les gràfiques es troben als 2 següents apartats). Si observem les gràfiques de precisió del següent apartat veiem que independentment de quina configuració utilitzem la precisió no varia, no obstant això, si mirem les gràfiques de temps de l'últim apartat observem que si que varien. En el cas d'un llindar de 10%, veiem que les millors son les inicialitzacions de random i custom per fischer. Això és a causa de que com que no agafem els punts inicials si no de diferents, al moment d'aplicar l'algoritme, s'han de fer menys iteracions, i per tant el temps emprat és diferent, però el resultat és el mateix per a totes les configuracions donat que totes arriben al mateix resultat.

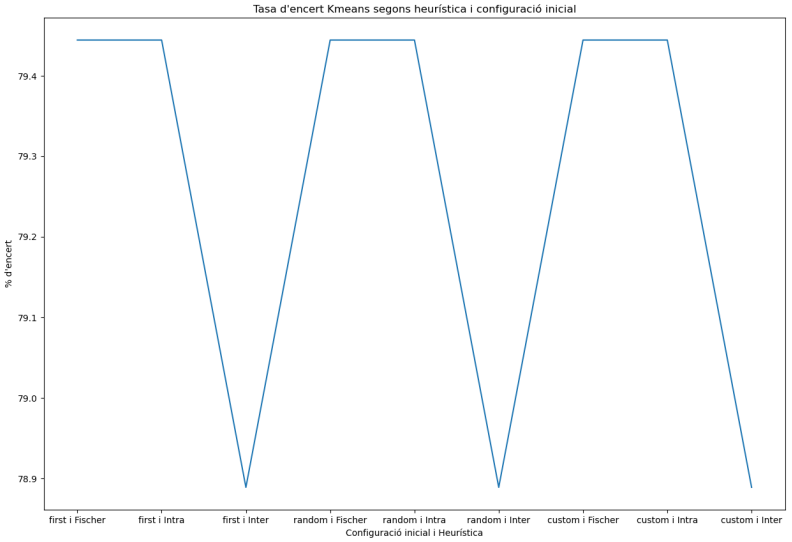
Diferents heurístiques per BestK

A la primera part, vàrem programar l'heurística IntraClass. Per aquesta segona part, hem programat dues heurístiques diferents Fischer i InterClass. Per analitzar-les, com ja s'ha explicat a l'apartat anterior les hem analitzat conjuntament amb les 3 configuracions inicials, així com hem calculat el temps total. (Les gràfiques de temps es troben al següent apartat).

Precisió amb llindar 20%



Precisió amb llindar 10%

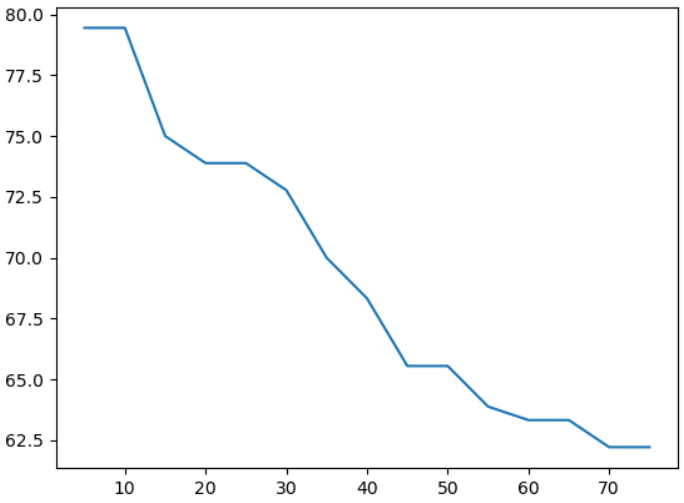


Com podem observar, usant el llindar del 10%, el qual és el més òptim en termes de precisió, les 2 heurístiques més òptimes son IntraClass i Fischer, millorant respecte a la versío original. A més, si observem les gràfiques del següent apartat, veiem que la més òptima de totes es fischer, donat que també és la més ràpida de les 3. Per el cas del llindar del 20%, observem que la millor per les 3 configuracions la millor es InterClass ja que és la més ràpida i la més precisa. aquesta millora és causada al fet de que al buscar de diferents maneres el color de les peces de roba ens provoca que degut a la distribució dels centroids veïns, aquestes 2 heurístiques noves ens resultin millors.

Find_BestK

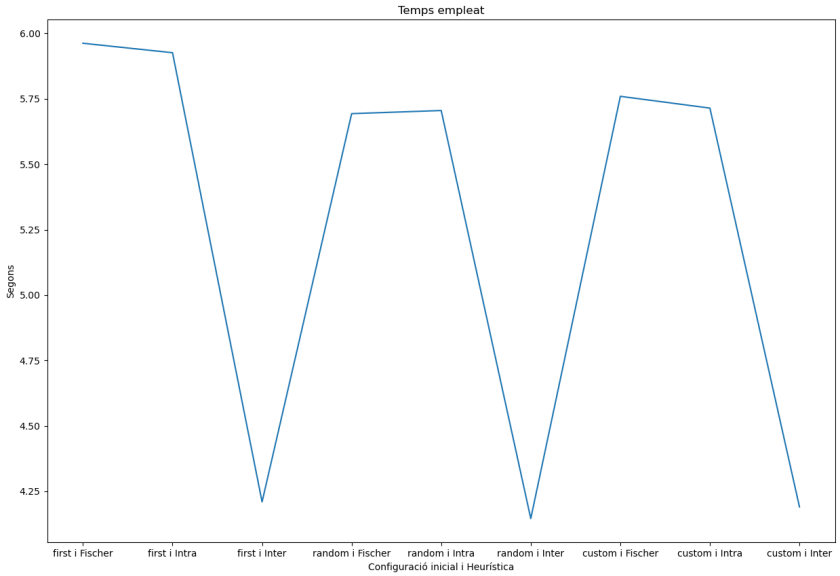
Per millorar inicialment el Find_bestK varem fer una funció que permet fer ús de les 3 heurístiques implementades. A més, varem intentar que el llindar varies per trobar una millor K, no obstant això,un cop feta la presentació, ens vàrem donar compte que això no era així, si no que mantenia sempre el mateix valor de llindar, el 20% per defecte. Per arreglar aquest error el que vam fer va ser agafar el Find_bestK bàsic, sense cap millora, i vam anar variant aquest llindar de 5 en 5, desde 5 fins a 80 per trobar el més òptim, en termes de precisió, que en aquest cas va ser 10%. Aquesta millora és causada pel fet de que al tindre un llindar més petit iterem més vegades per trobar millor K, el que ens proporciona que aquesta sigui més acurada, però també ens fa trigar més en trobar-la.

Gràfica de precisió segons llindar

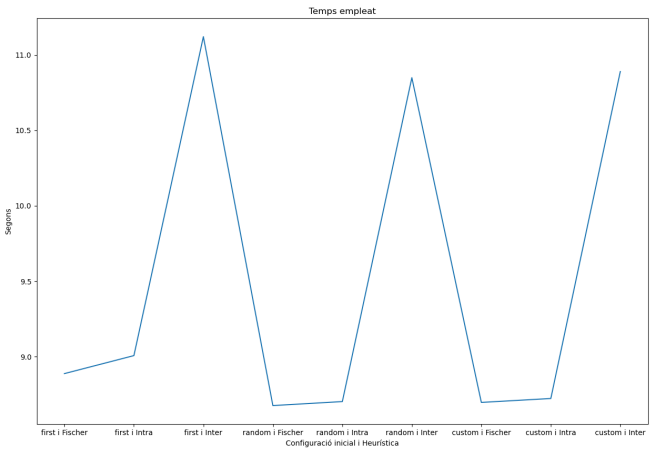


No obstant això, hem observat que usant un llindar més petit, el temps emprat per executar l'algoritme també s'augmenta, com es pot veure a la següent gràfica.

Temps amb llindar 20%



Temps amb llindar 10%



D'aquesta manera s'hauria de valorar si la millora en precisió val la pena per el temps extra que suposa aquesta millora. (Les gràfiques de precisió estan a l'apartat de Diferents heurístiques per BestK)

4.Conclusió

Després d'analitzar les diferents parts de la pràctica i les millores implementades, podem arribar a les següents conclusions:

S'han programat les funcions per a l'algorisme K-means i s'ha treballat amb les imatges i els colors predominants, així com les funcions per l'algorisme KNN, que hem fet servir per treballar amb la classificació de les imatges segons el tipus de peça de roba. Tot i no tenir el 100% d'efectivitat, creiem que els resultats obtinguts son bastant bons.

S'han implementat les funcions per a l'algorisme KNN i s'ha treballat amb la classificació de les imatges segons el tipus de peça de roba.

S'han realitzat diverses millores per optimitzar el codi, com ara canvis en la inicialització del K-means i l'ús de diferents heurístiques per trobar la millor K. Aquestes millores semblen haver millorat la precisió del sistema en la classificació de les imatges i fer-ho en el menor temps possible

En el kmeans hem calculat la precisió utilitzant diferents llindars. Els resultats mostren que les millores implementades han contribuït a augmentar la precisió del sistema, especialment amb un llindar del 10%. A més, ens hem adonat que si reduïm el percentatge del llindar de k-means, la precisió és més gran, i per tant l'algorisme encerta més, però triga més perquè ha de fer més iteracions.

En general, les millores implementades en les diferents parts de la pràctica han ajudat a millorar la precisió del sistema d'etiquetatge d'imatges de roba per tipus de peça i color. Tot i això, és important tenir en compte que algunes de les millores també han augmentat el temps d'execució, la qual cosa pot ser un factor a considerar segons les necessitats específiques del sistema. En conclusió, el treball realitzat sembla haver proporcionat resultats positius i ha contribuït a l'èxit del projecte.