Práctica obligatoria 2

Reconocimiento de objetos

|  |
| --- |
| Adrián García Oller |
| Daniel Lois Nuevo |
| Francisco Robles Castro |

## Ejercicio 1.

--classifier LDA-Bayes

Clases implicadas= CharacteristicsExtractor.py, main.py, ComputedImage.py y SignalRecognizer.py.

Para todas las imágenes de train:

-Lee la imagen.

-Pasa la imagen a niveles de gris.

-Ecualiza los niveles de gris de la imagen.

-Cambia el tamaño de la imagen a 32x32.

-Extrae los vectores de características con HOG, los convierte en ComputedImage y añade ComputedImage a una lista.

Devuelve una lista de computedImage con los atributos belonging\_class, que es la clase a la que pertenece la imagen, y characteristics\_vector, que es el vector de características calculado de la imagen.

Se pasa a la función que va a clasificar la lista de vectores de características de train y las etiquetas de todas las imágenes de train.

Crea el fichero donde se va a escribir el resultado.

Se convierte la lista de vectores de características en una matriz.

LDA se entrena con la matriz de vectores de características y los transforma dejando su dimensionalidad en 42 (número de clases menos uno).

Se entrena el clasificador bayesiano con el resultado del LDA y las etiquetas de las imágenes de entrenamiento. (En este caso el LDA y el clasificador bayesiano son el mismo tipo de objeto ya que sklearn.discriminant\_analysis permite realizar ambas funciones).

Para cada imagen de test:

-Guarda el nombre de la imagen y la etiqueta en una lista cada una.

-Lee la imagen.

-Pasa la imagen a niveles de gris.

-Ecualiza los niveles de gris de la imagen.

-Cambia el tamaño de la imagen a 32x32.

-Extrae el vector de características.

-Lo añade a la lista de vectores de características de test.

Convierte la lista de vectores de características de test en una matriz y reduce con LDA su dimensionalidad.

Se clasifican las imágenes de test con el clasificador bayesiano.

Por ultimo se escribe en el fichero creado al principio la etiqueta que le ha asignado el clasificador a cada imagen junto con el nombre de la imagen.

## Ejercicio 2.

1. LBP

--classifier LBP-LDA-Bayes

Clases implicadas= CharacteristicsExtractorLBP.py, main.py, ComputedImage.py y SignalRecognizerLBP.py.

El algoritmo es igual al del ejercicio uno solo que en vez de usar el histograma de orientación de gradientes utiliza LBP como vector de características con 100 vecinos circularmente simétricos y un radio circular de 1000

2. PCA

--classifier PCA-Bayes

Clases implicadas= CharacteristicsExtractor.py, main.py, ComputedImage.py y SignalRecognizerPCA.py.

El algoritmo es igual al del ejercicio uno solo que en vez de usar LDA para la reducción de dimensionalidad utiliza PCA con el mismo número de componentes que LDA ,42 (el número de clases menos 1).

3. KNN

--classifier LDA-KNN

Clases implicadas= CharacteristicsExtractor.py, main.py, ComputedImage.py y SignalRecognizerKNN.py.

El algoritmo es igual al del ejercicio uno solo que en vez de usar un clasificador bayesiano se utiliza uno de k vecinos con k=6.