|  |  |  |  |
| --- | --- | --- | --- |
| **T3** | **Minería de Datos Aplicada**  #13 Nancy Paredes Moreno  #17 Víctor Ramírez Rugerio | **8 °B** | **E #** |

**OCR programado Python**

**Requerimientos**

* Una computadora
* Python
* Carpeta de imágenes segmentadas
* Aplicar método de clasificación KNN
* Anaconda
* Spyder
* Windows

**Introducción OCR**

El OCR (Optical Character Recognition) es una tecnología que trata de emular la capacidad del ojo humano para reconocer objetos. Concretamente, es un software que permite el reconocimiento óptico de los caracteres contenidos en una imagen (documento escaneado o fotografía), de forma que estos se vuelven comprensibles o reconocibles para una computadora, obteniendo como resultado final un archivo en un formato de texto editable.

**¿Cómo funciona OCR?**

Para reconocer los caracteres, el software inspecciona la imagen pixel a pixel, buscando formas que coincidan con los rasgos de los caracteres. En función del nivel de complejidad o grado de desarrollo del software, éste buscará coincidencias con los caracteres y fuentes disponibles en el programa, o tratará de identificar los caracteres a través del análisis de sus características, de forma que el reconocimiento de los mismos no se limite exclusivamente a un determinado número de fuentes.

El OCR puede analizar los elementos del documento (bloques de texto, imágenes, tablas…), examinando los espacios en blanco y descomponiendo el texto en líneas, palabras y caracteres, de forma que el programa puede formular distintas hipótesis y cotejarlas con los diccionarios contenidos por el mismo (actualmente los programas contienen diccionarios en distintos idiomas), para formar palabras y textos completos.

**Creación del dataset**:

* Imagen
* Conjunto de datos
* Generador del dataset
* Por cada carpeta -leer imágenes
* Por cada imagen-guardar clase en dataset

**Propiedades:**

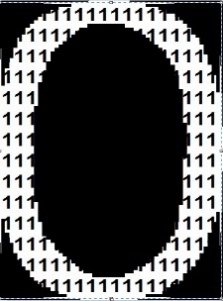
**Característica 1:**

Se toma el número de filas y columnas que tiene la imagen ingresada para obtener la característica(tamaño) y se multiplica Filas\*Columnas.

C:\Users\Hector HP\Desktop\OCR FINAL\comparar\0.png

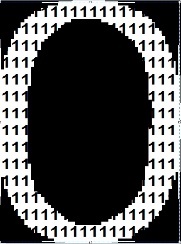
**Característica 2:**

Se obtienen los 1’s que existe dentro de la imagen, se hace una división entre el tamaño y de esta manera obtenemos la característica.



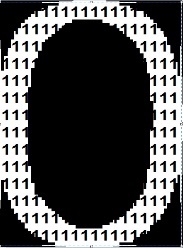
**Característica 3**

Es dividida la imagen a la mitad de las columnas totales que tiene y se obtienen todos los 1’s que existen dentro de esa columna y se divide por el número de filas que tiene la imagen.



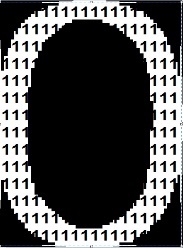
**Característica 4**

Se divide el número de columnas de la imagen en ¼, obtenemos la cantidad 1’s existentes dentro de esa columna y se divide con el número de filas que hay en la imagen.



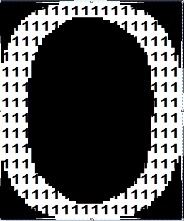
**Característica 5**

Se divide el número de columnas de la imagen en ¾, obtenemos la cantidad de 1’s existentes dentro de esa columna y se divide con el número de filas que hay en la imagen.



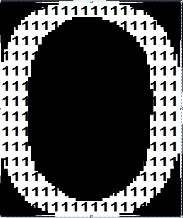
**Característica 6**

Es dividida la imagen a la mitad de las filas totales que tiene, obtenemos la cantidad 1’s en esa fila y se divide con el número de columnas que tiene la imagen.



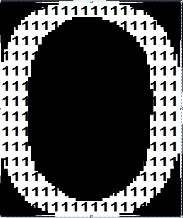
**Característica 7**

Se divide el número de filas de la imagen en ¼, obtenemos la cantidad 1’s existentes dentro de esa fila y se divide con el número de columnas que hay en la imagen.



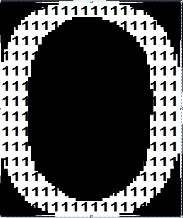
**Característica 8**

Se divide el número de filas de la imagen en ¾, obtenemos la cantidad de 1’s existentes dentro de esa fila y se divide con el número de columnas que hay en la imagen.



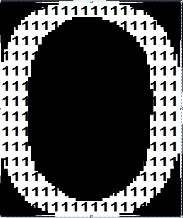
**Característica 9**

Obtenemos el número de cortes que tiene la imagen, los cortes en las imágenes son cuando hay un cambio en la imagen esto es de 0’s a 1’s o viseversa. Se optiene el numero de cortes que existen dentro de la imagen cuando la columna esta a la mitad se contabiliza el numero de cortes que tiene la imagen.



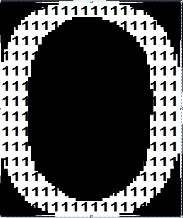
**Característica 10**

Se obtiene el numero de cortes cunado la columna esta en un1/4 de la imagen se contabiliza el numero de cortes que tiene la imagen.



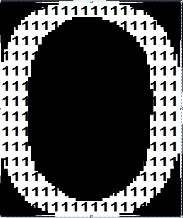
**Característica 11**

Se obtiene el numero de cortes cunado la columna esta en 3/4 de la imagen se contabiliza el numero de cortes que tiene la imagen.



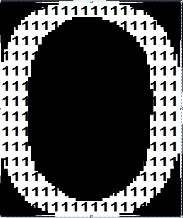
**Característica 12**

Se obtiene el numero de cortes cunado la fila esta en la mitad de la imagen se contabiliza el numero de cortes que tiene la imagen.



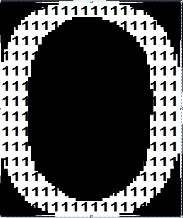
**Característica 13**

Se obtiene el numero de cortes cunado la fila esta en un 1/4 de la imagen se contabiliza el numero de cortes que tiene la imagen.



**Característica 14**

Se obtiene el numero de cortes cunado la fila esta en 3/4 de la imagen se contabiliza el numero de cortes que tiene la imagen.



**KNN:**

El método k-nn (K nearest neighbors Fix y Hodges) es un método de clasificación supervisada (Aprendizaje, estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad F(x/Cj) de las predictoras x por cada clase C\_j.

Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento x pertenezca a la clase C\_j a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras.

En el reconocimiento de patrones, el algoritmo k-nn es usado como método de clasificación de objetos (elementos) basado en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos. k-nn es un tipo de "Lazy Learning" (en), donde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación.

**Diagrama de funcionalidad**

Main

Genera Data

KNN

Cargar Dataset

Recorrido de imagenes

Obtiene caracteristicas de la nueva intancia

Ingresa K

Medir distancia

Obtiene vecinos mas cercanos

Extración de Caracteristicas

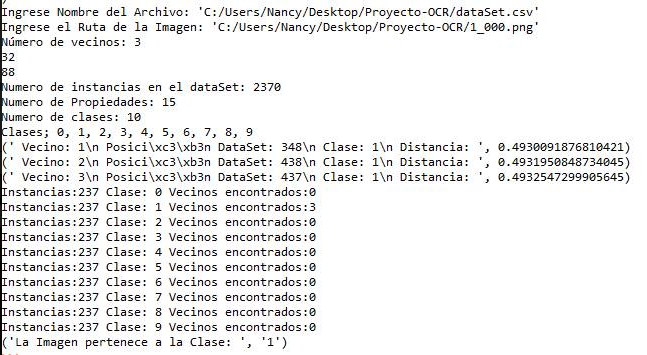
Impresion de vecinos mas cercanos

Escritura del dataset CSV

Clasificacion de imagen

DataSet

**Resultado**



**Clasificación KNN**

* Ingresa el valor de K
* Indica los vecinos más cercanos
* Numero de instancia
* Distancia
* Clase a la que pertenece la imagen
* Contador de ceros