|  |  |  |  |
| --- | --- | --- | --- |
| T3 | **Minería de Datos Aplicada**  #4 Chico Pérez Julia Gabriela  # Reyes Beristaín Francisco Javier | 8°B | E # |

**OCR**

**Requerimientos**

* 2 computadoras
* Python
* Spyder(Python 2.7)
* Carpeta de imágenes segmentadas mediante MatLab
* Método de clasificación “KNN”

**Introducción**

¿Qué es OCR?

El reconocimiento óptico de caracteres  (ROC), generalmente conocido como reconocimiento de caractere**s** y expresado con frecuencia con la sigla **OCR** (del inglés Optical Character Recognition), es un proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de una imagen símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos. Así podremos interactuar con estos mediante un programa de edición de texto o similar.

¿Cómo funciona el 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.

¿Qué es DataSet?

El **DataSet** es una representación de datos residente en memoria que proporciona una modelo de programación relacional coherente independientemente del origen de datos que contiene. El **DataSet** contiene en sí, un conjunto de datos que han sido volcados desde el proveedor de datos.

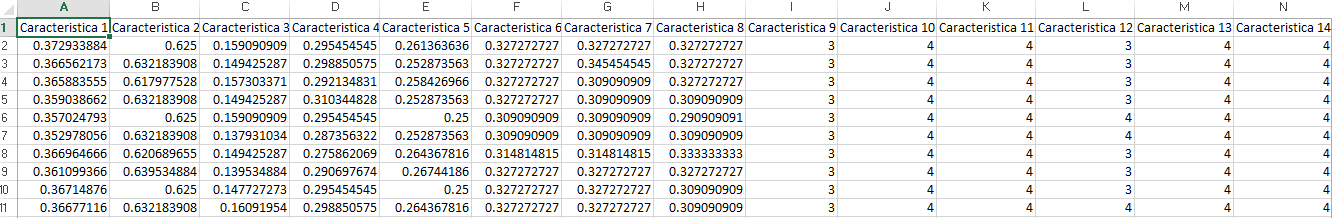
**Clasificación KNN**

Para OCR, existe un método muy conveniente, que proporciona resultados muy adecuados para la aplicación que se está tratando, El algoritmo K-NN (K vecinos más próximos).

Este método es muy popular debido a su sencillez y a cierto número de propiedades estadísticas bien conocidas que le proporcionan un buen comportamiento para afrontar diversos tipos de problemas de clasificación, siendo uno de ellos el de OCR. Dado un conjunto de objetos prototipo de los que ya se conoce su clase (es decir, dado un conjunto de caracteres de muestra) y dado un nuevo objeto cuya clase no conocemos (imagen de un carácter a reconocer) se busca entre el conjunto de prototipos los “k” más parecidos a nuevo objeto. A este se le asigna la clase más numerosa entre los “k” objetos prototipos seleccionados.

**Creación de DataSet**

Este se crea un documento con extensión .csv en donde se registraran 14 características obtenidas de cada una de las imágenes segmentadas. (Figura 1)



Para la caracterización de las imágenes y obtener los datos de entrenamiento se realizaron los siguientes pasos.

Conjunto de imágenes.

Tenemos un total de 10 imágenes, que se encuentran en una carpeta con el nombre “información” dichas imágenes están nombradas en subcarpetas del 0 al 9. (Figura 2)

C:\Users\Hector HP\Desktop\OCR FINAL\comparar\0.pngC:\Users\Hector HP\Desktop\OCR FINAL\comparar\7.png

Figura2: Muestra de imágenes utilizadas.

Paso no.1

Una vez que se ha corrido el programa (Figura 3) nos mostrara.

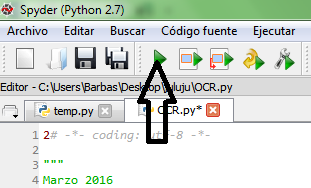


Figura 3: Correr programa.

En el programa nos muestra un menú donde tenemos 3 opciones a seleccionar.

1. Generar archivo dataset.
2. Ingresa el nombre de la imagen (Ejecución del método knn clasificación).
3. Salir (Finalización del programa).

Paso no.2

Se selecciona la imagen requerida para leer y determinar sus características.

**Conjunto de características**

A continuación se mencionan las 14 características utilizadas para crear el DataSet:

**Propiedad 1:** Se obtiene el número total de columnas y el número total de la filas por toda la imagen y se multiplican. (Figura 4)

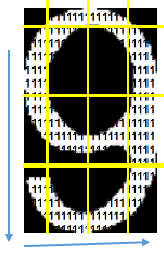


Figura 4: Filas\*columnas

**Propiedad 2**: Se obtiene el tamaño de la imagen y el número total de 1´s por cada una.

El # de 1´s entre el tamaño de la imagen buscamos los 1´s existentes en la imagen. (Figura 4.1)



Figura 4.1: Representación de 1’s en la imagen.

**Propiedad 3:** Se divide la imagen a la mitad y se toma esa columna para obtener el número de 1´s que tiene y dividir entre el número de filas de la imagen. (Figura 4.2)



Figura 4.2: No. de 1´s entre filas, y No. columnas entre 2

**Propiedad 4**: Se obtiene el número de 1´s que tienen las columnas y se divide entre 1/4 el número de columnas de la imagen. (Figura 4.3)



Figura 4.3: No. de 1´s entre filas, No. columnas entre 4

**Propiedad 5**: Se obtiene el número de 1´s de las columnas y se divide entre 3/4 obteniendo el uno de la imagen se divide con el número de filas que tenemos. (Figura 4.4)



Figura 4.4: No. de 1´s entre filas, 3(No. columnas entre 4)

**Propiedad 6:** Se obtiene el número de 1´s de las filas y se divide y se divide a la mitad de las filas de la imagen se divide con el número de columnas que tenemos de la imagen. (Figura 4.5)



Figura 4.5: No. de 1´s entre columnas, No. filas entre 2

**Propiedad 7:** Se obtiene el número de 1´s de las filas y se divide en 1/4 de las filas de la imagen y se divide con el número de columnas en de la imagen. (Figura 4.6)



Figura 4.6: No. de 1´s entre columnas, No. filas entre 4

**Propiedad 8:** Se obtiene el número de 1´s que tienen las filas y se divide entre 3/4 de las filas de la imagen se divide con el número de columnas en la imagen. (Figura 4.7)



Figura 4.7: No. de 1´s entre columnas, 3(No. filas entre 4)

**Cortes**

En las siguientes características se obtendrán el número de cortes que tiene la imagen. (Figura 5 y 5.1)

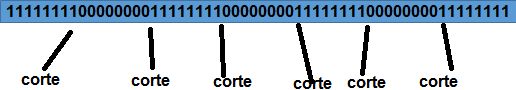


Figura 5: Ejemplo de cambios de 0 a 1.

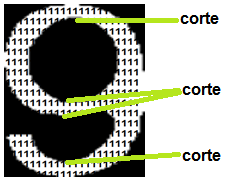


Figura 5.1: Cortes en una imagen segmentada.

**Propiedad 9:** Se divide la imagen a la mitad y se recorre la columna hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.2)



Figura 5.3: Filas, No. columnas entre 2

**Propiedad 10:** Se divide la imagen en 1/4 y se recorre la columna hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.3)



Figura 5.3: Filas, No. columnas entre 4

**Propiedad 11:** Se divide la imagen en 3/4 y se recorre la columna hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.4)



Figura 5.4: Filas, 3(No. columnas entre 4)

**Propiedad 12:** Se divide la imagen a la mitad y se recorre la fila hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.5)



Figura 5.5: Columnas, No. filas entre 2

**Propiedad 13:** Se divide la imagen en 1/4 y se recorre la fila hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.6)



Figura 5.6: Columnas, No. filas entre 4

**Propiedad 14:** Se divide la imagen en 3/4 y se recorre la fila hasta encontrar un cambio de 0 a 1 y se cuantifican los cortes. (Figura 5.7)



Figura 5.7: Columnas, 3(No. filas entre 4)

Paso no. 3

Aplicación de metodo knn a una imagen que el usuario seleccione y se obtiene las caracteristicas que se manda a llamar en el programa y compara las imágenes para obtener los valores o vecinos mas cercanos correspondientes al valos de “k”.

Diagrama de actividades.

Clase de la imagen

Medir la distancia

Impresión de vecinos más cercanos.

Escritura del dataset .CSV

Extración de Características

Recorrido de imagenes

Genera DataSet

KNN

Ingresa el valor de K

Obtiene vecinos más cercanos

Cargar Dataset

Menú

Obtiene las características de la nueva instancia

Paso no. 4

Copiar la siguiente dirección en un navegador y descargar los archivos OCR.zip e información.zip, posteriormente verificar el manual de usuario anexo.

<https://github.com/KenGame/ocr>

**Resultados**

