|  |  |  |  |
| --- | --- | --- | --- |
| T3 | **Minería de Datos Aplicada**  # 12 Jiménez Hernández Nelly  # 29 Solís Hernández Marco Antonio | 8°B | E # |

**OCR**

**Requerimientos**

* Computadora
* Tener instalado Python 3.5
* Tener instalado Spyder
* Carpeta de imágenes segmentadas

**Introducción**

¿Qué es 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 un ordenador, obteniendo como resultado final un archivo en un formato de texto editable. El formato del archivo de salida (txt, pdf, etc.) dependerá de las posibilidades que ofrezca el software.

**¿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 (actualmente los programas contienen diccionarios en distintos idiomas), para formar palabras y textos completos.

Aunque actualmente el OCR puede llegar a mantener la estructura de los documentos originales en el archivo de salida, e incluso reconocer caracteres contenidos en documentos manuscritos, diagramas, partituras, etc., no hay que olvidar que su nivel de efectividad sigue siendo limitado, lo que hace necesaria una posterior revisión y corrección manual del texto generado. Actualmente existe en el mercado una amplia oferta de software OCR, cuyo nivel de sofisticación y precio está directamente relacionado con su precisión y efectividad en el reconocimiento de caracteres.

**Conjunto de imágenes.**

Se cuenta con un total de 10 clases, las cuales serán la asignación a cada uno de los números, se encuentran en la carpeta con el nombre “DatosPrueba” dichas imágenes están nombradas en subcarpetas que van del 0 al 9. (Observe imagen 1)

C:\Users\Marko\Documents\Universidad Politecnica de Tlaxcala\8° Cuatrimestre\Mineria de datos\OCR_OK\1.png

Imagen 1: Muestra de imagen utilizada.

**¿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.

**Creación de DataSet**

Para generar el DataSet, ejecutaremos nuestro programa con la opción 1, al dar Enter mostrara el mensaje “procesando información” como se observa en la imagen 2 y nos indicara un detallado de los datos leídos y almacenados en el archivo definido como “DataSet.csv”.

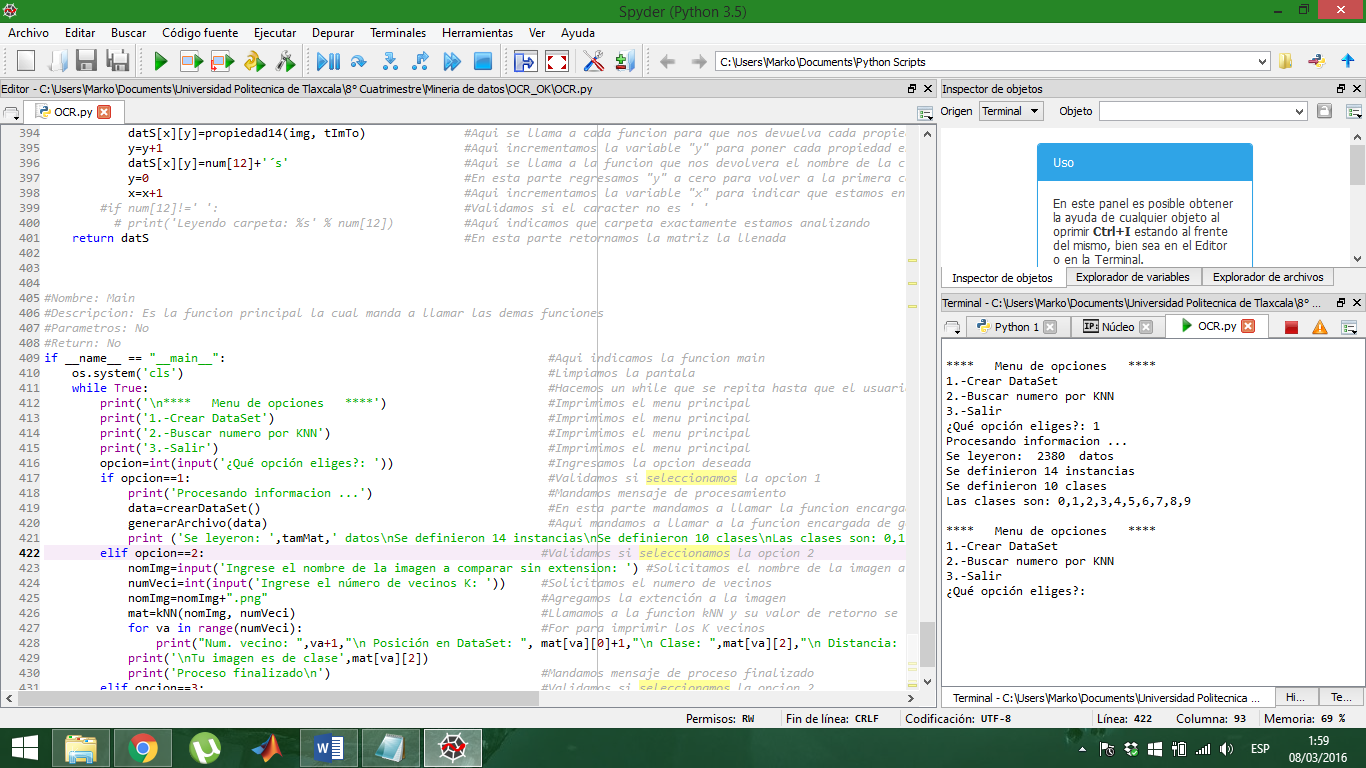


Imagen 2: Muestra al generar el DataSet

**Conjunto de características**

A continuación se mencionan las 14 características utilizadas para crear el DataSet, previo a ello el programa hace un recorrido para que a cada imagen de cada carpeta contenida sea leída y así se obtengan las características de forma más específica:

P1: El # de columnas entre el # de filas

P2: El # de 1´s entre el tamaño de la imagen

P3: El # de 1´s entre filas, #columnas entre 2

P4: El # de 1´s entre filas, #columnas entre 4

P5: El # de 1´s entre filas, 3(#columnas entre 4)

P6: El # de 1´s entre filas, filas

P7: El # de 1´s entre filas, filas

P8: El # de 1´s entre filas, filas

P9:

P10:

#de cortes

0´s 1´s

P11:

P12:

P13:

P14:

Ejemplo:

**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.

Para poder aplicar el método KNN dentro de nuestro OCR, ejecutamos en menú en la opción 2, nos pedirá ingresemos el número a clasificar sin extensión del formato y el número de vecinos (K) a mostrar, la salida nos indicara en número de vecino, posición en el dataset, clase a la que pertenece el dato, y la distancia que tiene hacia nuestro numero ingresado, finalmente indicara la clase a la que pertenece nuestra imagen según el dataset indicado. (Véase imagen 3)

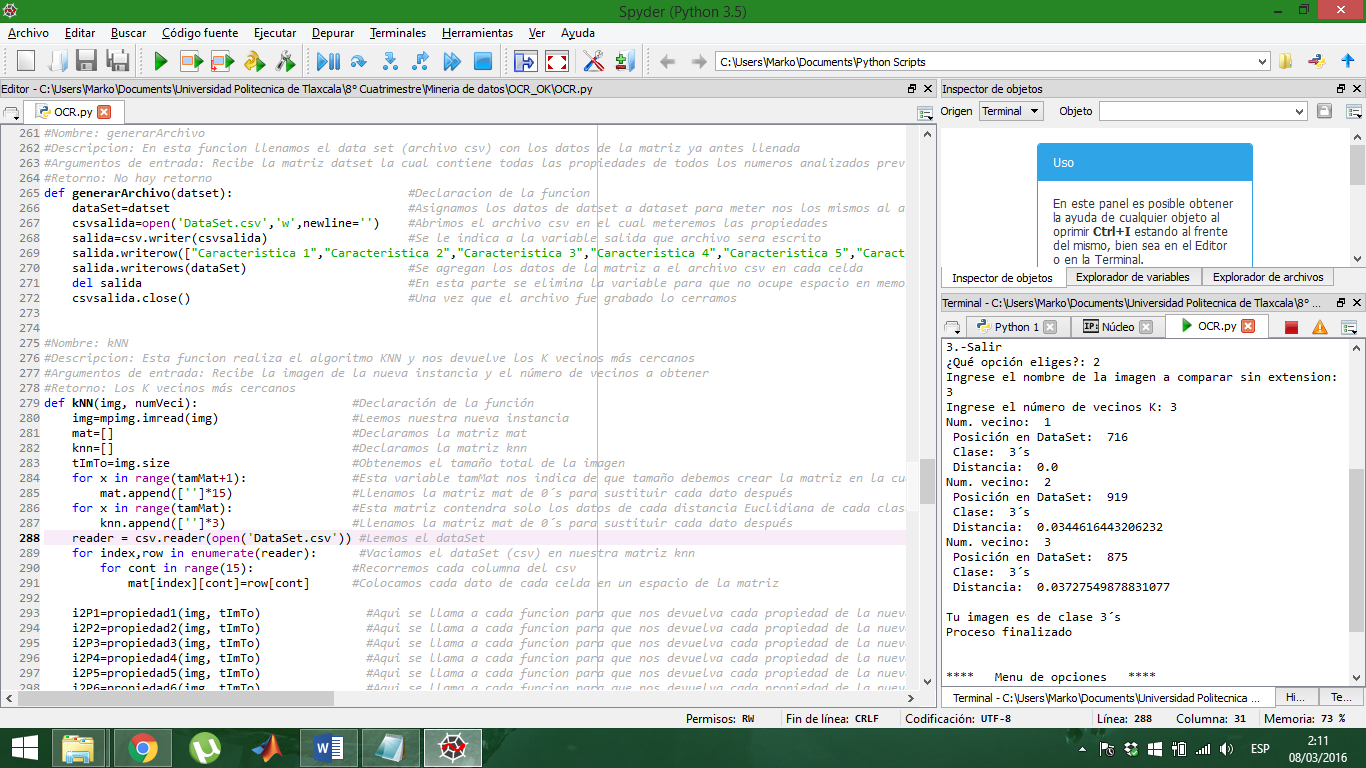


Imagen 3: Muestra de clasificación por KNN