T3	**Minería de datos** # 9 Roberto Espinosa Galindo #10 Marcos Geovanny Esteban Mendieta	8B	E #

Requerimientos mínimos

- ➔ Una computadora con procesador Dual Core o superior.
- ➔ Memoria RAM 2GB.
- ➔ Espacio disponible en Disco Duro 30 MB.
- ➔ Sistema Operativo Linux, Mac o Windows.
- ➔ Tener instalada una versión de Python **2.7 o superior.**

¿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 una computadora, 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.

Funcionamiento.

Para reconocer los caracteres, el software inspecciona la imagen píxel a píxel, 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.

Descripción del programa OCR realizado en Python.

El primer paso para realizar el OCR es contar con un conjunto de imágenes para analizar, en este caso contamos con imágenes binarias de los números del 0 al 9, las cuales se generaron desde un programa realizado en MATLAB.

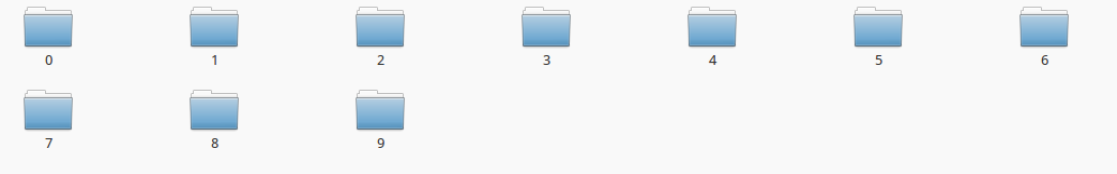

Figura 0.1: Directorio de imágenes para el Training

Antes de generar el dataset, se extraen imágenes de prueba de cada carpeta, las cuáles serviran para probar el programa.

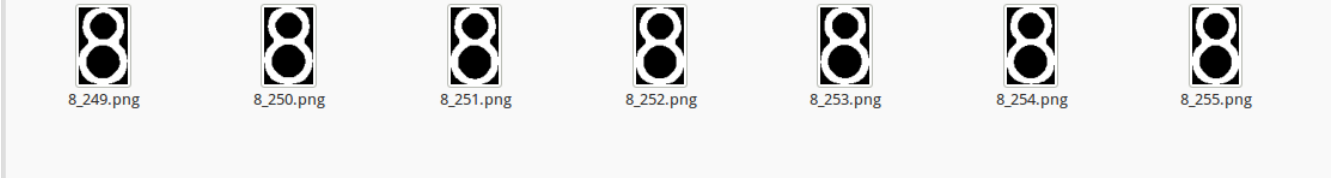

Figura 0.2: Imágenes extraídas

El segundo paso, es generar un dataset que nos servirá como fuente de conocimiento, para ello se consideran 14 características tomadas del conjunto de imágenes descritas en el paso anterior.

Características tomadas de cada imagen:

1- Resultado del número de columnas entre en número de filas.

Ejemplo: matriz de 10 columnas y 10 filas

Figura 1: Representación gráfica de la 1° característica

2- Número de 1's que hay en toda la imagen / tamaño de la imagen(filas*columnas).

Figura 2: Representación gráfica de la 2° característica

3- Número de 1’s que hay en la columna de en medio de la imagen.

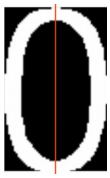

Figura 3: Representación gráfica de la 3° característica

4- Número de 1’s que hay en la columna (total de columnas / 4).

Figura 4: Representación gráfica de la 4° característica

5- Número de 1’s que hay en la columna ((total de columnas / 4) * 3).

Figura 5: Representación gráfica de la 5° característica

6- Número de 1’s que hay en la fila de en medio de la imagen.

Figura 6: Representación gráfica de la 6° característica

7- Número de 1’s que hay en la fila (total de filas / 4).

Figura 7: Representación gráfica de la 7° característica

8- Número de 1’s que hay en la fila ((total filas / 4) * 3).

Figura 8: Representación gráfica de la 8° característica

9- Número de cortes que hay en la columna de en medio.

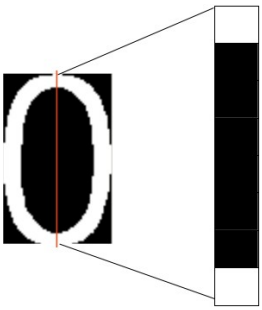

Figura 9: Representación gráfica de la 9° característica

10- Número de cortes que hay en la columna (total columnas / 4).

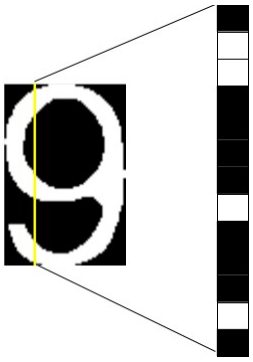

Figura 10: Representación gráfica de la 10° característica

11- Número de cortes que hay en la columna ((total columnas / 4) *3).

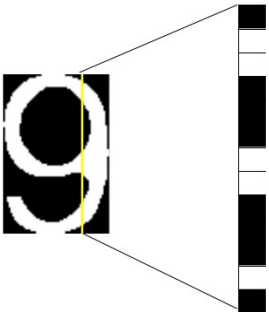

Figura 11: Representación gráfica de la 11° característica

12- Número de cortes que hay en la fila de en medio.

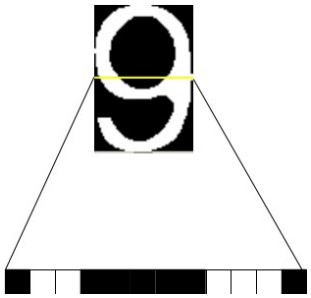

Figura 12: Representación gráfica de la 12° característica

13- Número de cortes que hay en la fila (total de filas / 4).

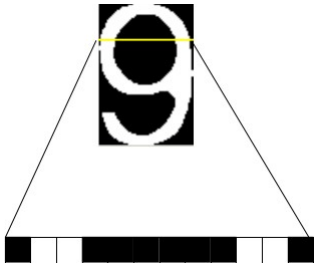

Figura 13: Representación gráfica de la 13° característica

14- Número de cortes que hay en la fila ((total de filas / 4) * 3).

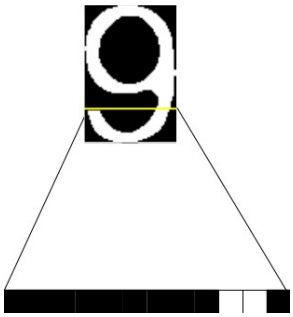

Figura 14: Representación gráfica de la 14° característica

En el dataset generado, se agregan dos características más.

Al principio se agrega la característica del **número de instancia consecutivo** y al final del dataset, la característica que es el **número de clase a la que perteneciente**.

Una vez tomadas las características de todas las imágenes, el siguiente paso es escribir estos datos en un archivo .csv.

Ya que se tiene el dataset con las características de todas las imágenes, se ingresa una imagen para determinar qué número es (0 – 9).

De la imagen ingresada, se sacan sus 14 características ya antes representadas gráficamente y se aplica el método KNN para clasificación.

El resultado final del programa OCR es determinar por medio del método KNN que número es (0 - 9).

Ejecución del programa.

Como ya se mencionó anteriormente, este programa se realizó en el lenguaje de programación Python, y para ejecutarlo desde Linux se abre una terminal con lo siguiente.

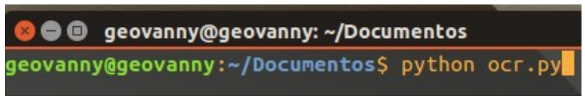

Figura 0.3: Comando para ejecutar el programa

- Se muestra un menú de opciones, del cual elegimos la primera opción.

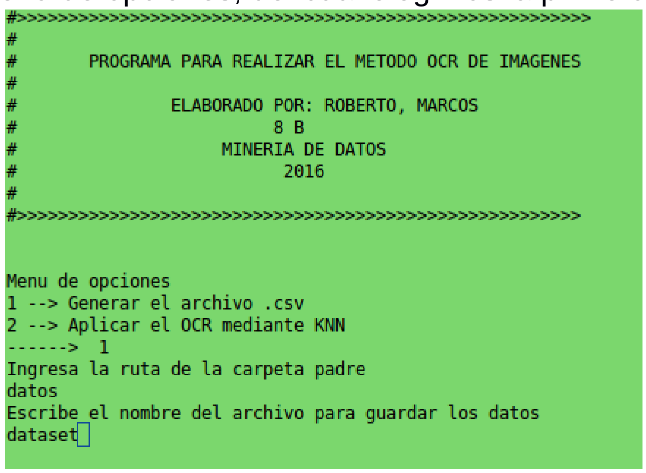

Figura 15: Captura de pantalla del menú principal del programa

- Se ingresa la carpeta donde se tienen las subcarpetas de las imágenes de cada número y se ingresa el nombre deseado para el dataset.

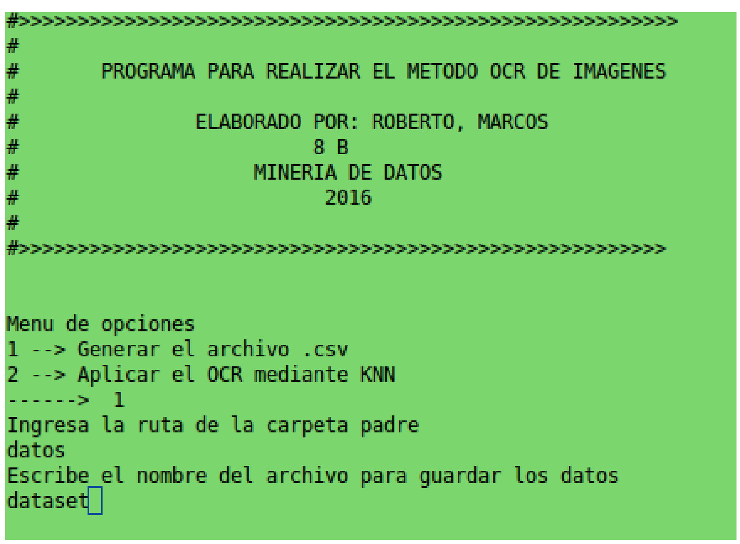

Figura 16: Ingreso de rutas

- Se empieza a generar el dataset, y se muestra en pantalla el progreso de esta tarea.

```
Menu de opciones
1 --> Generar el archivo .csv
2 --> Aplicar el OCR mediante KNN
------>  1
Ingresa la ruta de la carpeta padre
datos
Escribe el nombre del archivo para guardar los datos
dataset
Escribiendo las caracteristicas en el archivo csv
Sea paciente, este proceso puede llevar unos minutos

Progreso actual = escribiendo datos de la carpeta ---> 0
Progreso total global ---> 00%

Progreso actual = escribiendo datos de la carpeta ---> 1
Progreso total global ---> 10%

Progreso actual = escribiendo datos de la carpeta ---> 2
Progreso total global ---> 20%

Progreso actual = escribiendo datos de la carpeta ---> 3
Progreso total global ---> 30%

Progreso actual = escribiendo datos de la carpeta ---> 4
Progreso total global ---> 40%

Progreso actual = escribiendo datos de la carpeta ---> 5
Progreso total global ---> 50%

Progreso actual = escribiendo datos de la carpeta ---> 6
Progreso total global ---> 60%

Progreso actual = escribiendo datos de la carpeta ---> 7
Progreso total global ---> 70%

Progreso actual = escribiendo datos de la carpeta ---> 8
Progreso total global ---> 80%

Progreso actual = escribiendo datos de la carpeta ---> 9
Progreso total global ---> 90%
Progreso total global ---> 100%
```

Figura 17: Progreso de generación del dataset

- Una vez generado el dataset se muestra información general del archivo .csv creado.

```
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
------> Informacion general del dataset creado
------> Instancias escritas en el archivo 2310
------> Cantidad de caracteristicas de cada instancia : 16
------> Descripcion de cada caracteristica

atributo 1 = numero de instancia
atributo 2 = numero de columnas / numero de filas
atributo 3 = numero de 1's / tamanio de la imagen(filas * columnas)
atributo 4 = numero de 1's que hay en la columna de enmedio/tamanio de la imagen
atributo 5 = numero de 1's que hay en la columna a un cuarto/tamanio de la imagen
atributo 6 = numero de 1's que hay en la columna entre 4 * 3/tamanio de la imagen
atributo 7 = numero de 1's que hay en la fila de enmedio/tamanio de la imagen
atributo 8 = numero de 1's que hay en la fila a un cuarto/tamanio de la imagen
atributo 9 = numero de 1's que hay en la fila entre 4 * 3/tamanio de la imagen
atributo 10 = numero de cortes que hay en la columna de enmedio/tamanio de la imagen
atributo 11 = numero de cortes que hay en la columna a un cuarto/tamanio de la imagen
atributo 12 = numero de cortes que hay en la columna entre 4 * 3/tamanio de la imagen
atributo 13 = numero de cortes que hay en la fila de enmedio/tamanio de la imagen
atributo 14 = numero de cortes que hay en la fila a un cuarto/tamanio de la imagen
atributo 15 = numero de cortes que hay en la fila entre 4 * 3/tamanio de la imagen
atributo 16 = clase a la que pertenecen

Clases : 0 --> 1 --> 2 --> 3 --> 4 --> 5 --> 6 --> 7 --> 8 --> 9
-----> Nombre del archivo csv generado dataset.csv


Tiempo de procesamiento de las imagenes =  1.56 minutos.
Desea ejecutar de nuevo el programa??
1 --> Si
2 --> No
---->
```

Figura 18: Se muestra información general del dataset

- Al terminar de generar el dataset, ejecutamos de nuevo el programa y ahora elegimos la opción 2, para aplicar el KNN, se ingresa el nombre del archivo .csv generado anteriormente, la ruta de la imagen que queremos saber que es y finalmente el valor de K.

```
Menu de opciones
1 --> Generar el archivo .csv
2 --> Aplicar el OCR mediante KNN
------>  2
Aplicando el metodo KNN para el reconocimiento de imagenes OCR
Ingrese la ruta del archivo csv
----->  dataset
Ingrese la ruta de la imagen que desea reconocer
----->  1/1_260
Ingrese el valor de K
----->  15
```

Figura 18: Se escribe la ruta del archivo .csv generado anteriormente, se selecciona la imagen y se ingresa el valor de k

- El programa comienza a calcular las características de la imagen seleccionada y aplica el KNN para buscar las coincidencias que más se acercan y determina que número es.

```
Ingrese el valor de K
----->  15



No. --> 1       instancia --> 260       clase --> 1     distancia --> 0.0026149820004599898
No. --> 2       instancia --> 262       clase --> 1     distancia --> 0.004384538655076797
No. --> 3       instancia --> 356       clase --> 1     distancia --> 0.004724477887200423
No. --> 4       instancia --> 275       clase --> 1     distancia --> 0.004738112772368171
No. --> 5       instancia --> 297       clase --> 1     distancia --> 0.0050543273273959514
No. --> 6       instancia --> 266       clase --> 1     distancia --> 0.006862418525338665
No. --> 7       instancia --> 272       clase --> 1     distancia --> 0.007237586810204705
No. --> 8       instancia --> 252       clase --> 1     distancia --> 0.008628169647603691
No. --> 9       instancia --> 426       clase --> 1     distancia --> 0.0090300323922245 27
No. --> 10      instancia --> 458       clase --> 1     distancia --> 0.009352885464289893
No. --> 11      instancia --> 366       clase --> 1     distancia --> 0.00937355484964072
No. --> 12      instancia --> 331       clase --> 1     distancia --> 0.009724846530661593
No. --> 13      instancia --> 318       clase --> 1     distancia --> 0.009744726987248988
No. --> 14      instancia --> 270       clase --> 1     distancia --> 0.010441409377693
No. --> 15      instancia --> 364       clase --> 1     distancia --> 0.010799781881015028

----------------------------------------

Instancias de la clase 0 --> 0
Instancias de la clase 1 --> 15
Instancias de la clase 2 --> 0
Instancias de la clase 3 --> 0
Instancias de la clase 4 --> 0
Instancias de la clase 5 --> 0
Instancias de la clase 6 --> 0
Instancias de la clase 7 --> 0
Instancias de la clase 8 --> 0
Instancias de la clase 9 --> 0



----------------------------------------

La imagen ingresada es de clase  ---> 1
```

Figura 19: Se muestra información de knn

- Una vez ingresado K, se muestran las distancias más cercanas ordenadas de mayor a menor, la instancia a la que pertenece y la clase, posteriormente se muestran los contadores de las coincidencias de cada clase, finalmente se indica la clase a la que pertenece la imagen ingresada.

- El programa tiene un menú que da la opción de volver a ejecutarlo o finalizarlo.

- Con esto podemos comprobar que el programa funciona correctamente, ya que identifica que la imagen ingresada es un 1, lo cual es cierto.

- **Código fuente disponible en GitHub.**
- https://github.com/steelgnu/Mineria_datos

Diagrama de flujo del programa.