|  |  |  |
| --- | --- | --- |
| T3 | MINERIA DE DATOS  OCR | 8° C |
| Gerardo Lara Rojas #14  Diana Laura Pérez Cuatepitzi #23 |

**INTRODUCCION**

Un OCR (Optical Character Recognition) consiste en emular la capacidad que tiene el ojo humano al reconocer objetos, permite el reconocimiento óptico de los caracteres contenidos de un conjunto de imágenes.

Un conjunto de imágenes está formado por todos los valores que puede llegar a tomar una imagen en una función, es considerado también un recorrido del conjunto de valores que puede tomar una variable x,y.

Un pixel es el punto de color concreto que forma cada imagen, expresa cada unidad de color que conforma una imagen digital.

De esta manera es posible que OCR comprenda o reconozca para cualquier ordenador cierta información obteniendo como archivo final un formato de texto editable.

**Requerimientos:**

**\***1 computadora Sistema operativo Wndows 8

\*Instalación Python 3.5.1

\*Instalación Anaconda 2.4.1 de 64/32 bits

\*OCR en Matlab

\*Excel

**CONJUNTO DE ELEMENTOS**

El conjunto de imágenes usada para la creación del dataset está compuesto por 2,380 imágenes binarias divididas en 10 carpetas

Cada carpeta está compuesta por 238 imágenes binarias de un solo número como los siguientes:

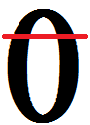


Las medidas de estas imágenes oscilan entre 55 y 62 pixeles de ancho por 87 y 93 pixeles de alto, cada imagen pasada por un proceso de segmentación y binarización razón por la cual se encuentran llenas de 0 y 1, donde el 0 representa el color negro y el 1 el color blanco además de haber eliminado el ruido resultante de la binarización de cada imagen, esto con el objetivo de obtener mejores resultados. Este conjunto nos era de utilidad para obtener una serie de características para realizar la creación de un dataset para posteriormente poder hacer una clasificación.

**DATASET**

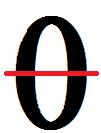
Un dataset representa un conjunto completo de datos incluyendo tablas que contienen, ordenan y restringen los datos.

Para la creación de un dataset primero se analizan las imágenes del 0-9 binarizadas en blanco y negro obtenidas ya del OCR de Matlab pero ahora cargadas en python, estas imágenes contienen valores de 0 y 1 para analizar y caracterizar cada imagen se toma en cuenta los cambios que se observan en la imagen es decir un valor es 0 y pasa a 1, estos cortes permiten el conteo de las características de las imágenes que se guardan en el dataset y determinaran el contenido de cierta imagen dependiendo de las características encontradas como el alto, ancho, numero de pixeles etc de cada imagen.

En este caso se utilizaran 14 características ya definidas:

1, 2

Primera línea horizontal: se encarga de encontrar los cambios de color en la primera cuarta parte de esta imagen recorriéndola en forma horizontal segunda característica es resultado de encontrar los pixeles blancos

3,4

Segunda línea horizontal: Busca los cambios de color a la mitad de lo alto de la imagen recorriéndola en horizontal segunda característica es resultado de encontrar los pixeles blancos

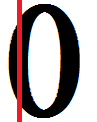
5,6

Tercera línea horizontal: Encuentra los cambios de color en la última cuarta de la imagen recorriéndola en horizontal segunda característica es resultado de encontrar los pixeles blancos



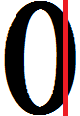
7,8

Primera línea vertical: Encuentra los cambios de color en la primera cuarta parte de la imagen recorriéndola en vertical segunda característica es resultado de encontrar los pixeles blancos



9,10

Segunda línea vertical: Encuentra los cambios de color a la mitad de lo ancho de la imagen recorriéndola en vertical segunda característica es resultado de encontrar los pixeles blancos

11,12

Tercera línea vertical: Encuentra los cambios de color en la última cuarta parte de la imagen recorriéndola en vertical segunda característica es resultado de encontrar los pixeles blancos

13,14

Se obtienen dos características resultantes del alto y ancho de la imagen, en la siguiente característica se toman en cuenta los pixeles blancos

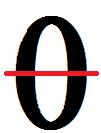


Para poder obtener las características necesarias se cuentan los “cambios” o “cortes” de las imágenes binarizadas es decir se marca un cambio cuando el valor pasa de 0 a 1 y viceversa, estos cambios son contados por una línea que cambia de posición dependiendo de la función

Ejemplo:

2 cambios

1 cambio



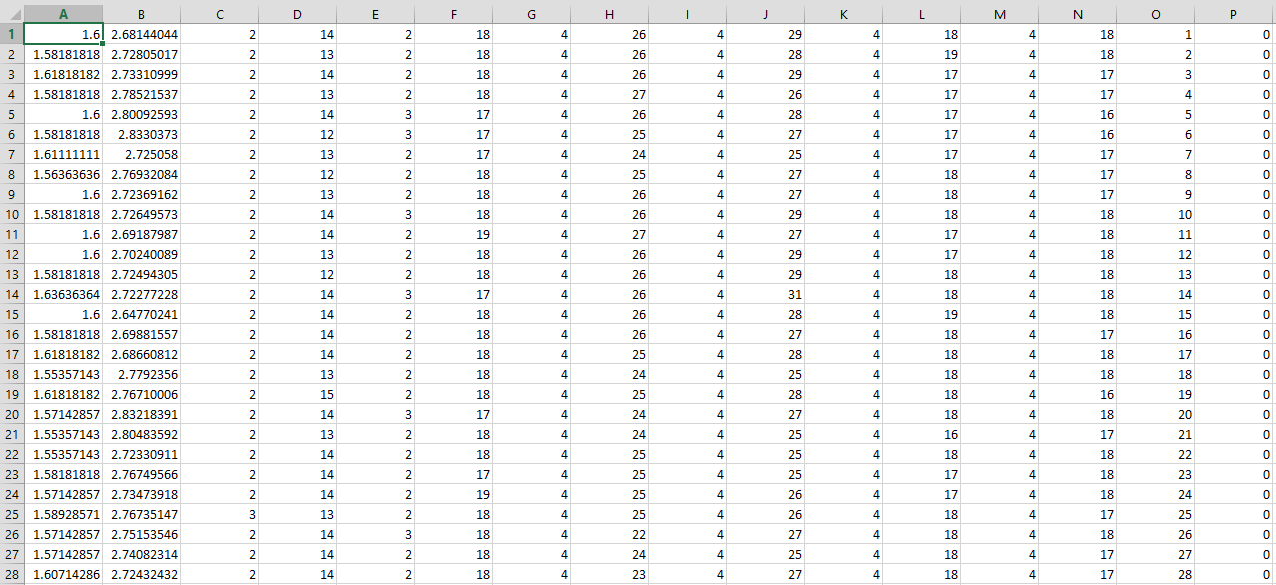
1

0

1

0

En la imagen se observa la primera característica en donde el contorno blanco representa 0 y el contorno negro que define el número es igual a 1 en este caso podemos observar que se realizaron dos cambios o cortes y esto es lo que se guarda en el dataset

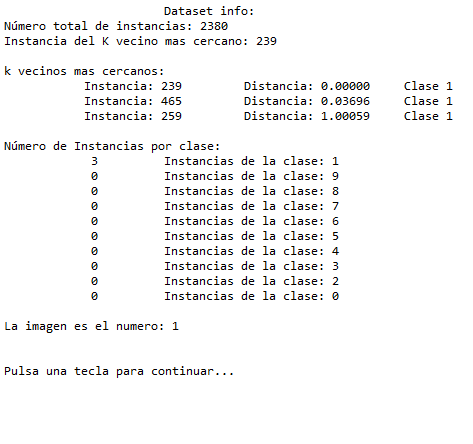


Práctica

1.-Correr el archivo Menu\_OCR.py

2.-Elegimos la opción que deseemos (1.- Generar DataSet, 2.- Clasificar una imagen o 3.-Salir

* 2.1.-Generamos unDataSet con 9 clases y 2380 instancias
* 2.2.-Clasificar una imagen.
* 2.2.1.- Pedimos el nombre de la imagen con terminación .png
* 2.2.2.-Pedimos el número de k-vecinos
* 2.2.3.- Se clasificara la imagen después de haber pasado por el método knn para clasificación y posteriormente se mostrara la imagen siguiente



**CLASIFICACION**

Para clasificar los datos que obtuvimos en el dataset se utiliza el método KNN para clasificación. Clasifica nuevas instancias como la clase mayoritaria de entre los k vecinos más cercanos de entre los datos de entrenamiento.

KNN utiliza 3 variables:

K= un valor aleatorio, el cual será el conjunto seleccionado del dataset

(x,y)= es la nueva instancia

KNN trabaja estas variables sobre un dataset (en este caso el generado previamente) y genera la clasificación calculando la distancia euclidiana:

**d=(dn,ni)**

d=Distancia

dn=n distancia tomada del dataset

ni=valores de la nueva instancia (x,y)

Entonces:

d(dn,ni)= √(x1-x2)^2 + (y1-y2)^2

Una vez calculados los valores de K se hace una comparación de los resultados obtenidos y los que sean más cercanos determinan la clase

El funcionamiento de KNN es el siguiente:

* Calcula la distancia entre la nueva instancia y los datos del DataSet
* Ordena las distancias de manera ascendente
* Reune los vecinos mas cercanos a la categoría Y
* Clasifica la nueva instancia

Disponible en: <https://github.com/shifrud/MineriaDatos>

**RESULTADOS**

Como resultados obtenemos un dataset generado por la extracción de 14 características, además de lograr una clasificación de los datos guardados en el dataset, en donde primero seleccionamos el número de vecinos con los que se va a comparar (k) y esto determina la clase a la que pertenece, usando reconocimiento de caracteres de una imagen previamente binarizada.

Elegir K vecinos

Obtener 14 características

Leer imagen

Guardar caracteres en el dataset

Obtener 14 característi-cas

­leer carpetas y subcarpetas

Dataset

Clasifi-car imagen

Imprimir información de dataset

Método KNN

Lectura de todas las imágenes

Opción

Selección de opción

1 3

2

Si No