Reconocimiento de Dígitos

Jhonier Andrés Calero Rodas, Juan Pablo Moreno Muñoz y Joan Manuel Tovar Guzmán
Escuela de Ingeniería de Sistemas y Computación (EISC) – Facultad de Ingeniería
Universidad del Valle
Santiago de Cali, Colombia

***Resumen*— Se realizó la implementación en C++ de un algoritmo para el reconocimiento de dígitos, el cual está basado en la descomposición en valores singulares (SVD). En un principio se hizo un pre-procesamiento de las imágenes, esto con el fin de que no surgieran complicaciones con el manejo de las imágenes durante el desarrollo del algoritmo. Luego se procedió a representar las imágenes con la estructura de datos más conveniente y las dimensiones que permitieran realizar las debidas operaciones. Por último se realizó la descomposición en valores singulares y los cálculos necesarios para el reconocimiento de los dígitos.**

***Palabras Clave*—Reconocimiento, SVD, dígitos.**

I. INTRODUCCIÓN

El reconocimiento de caracteres se basa en la identificación de un símbolo, que puede ser una letra o un número, partiendo de la digitalización de una imagen. Actualmente, este problema tiene muchos métodos mediante los cuales se puede llegar a una solución del mismo, lo cual es muy importante porque las aplicaciones del reconocimiento de caracteres pueden ser muy variadas, por ejemplo, el reconocimiento de las matrículas de los carros en los radares de velocidad o la automatización de la redirección de cartas en el correo postal.

En este informe se presenta la descripción del problema de reconocimiento y la importancia de la investigación de soluciones cada vez más precisas. Además, se explica de forma general cómo se llegó a la implementación del algoritmo con el análisis de la SVD. Finalmente, se muestra el análisis realizado a partir de los resultados obtenidos.

II. DESCRIPCIÓN DEL PROBLEMA DE RECONOCIMIENTO

El problema de reconocimiento de dígitos es un sub-problema del problema de reconocimiento de caracteres, el cual se identificó alrededor de 1980. Particularmente, el reconocimiento de dígitos tiene diversos desafíos que vienen dados ya sea por la forma en cómo está escrito el digito, o por las similitudes que hay entre algunos pares de dígitos como: 1 y 7, 3 y 8, 5 y 6, 9 y 8.

Fig. 1. Similitud entre dos dígitos (1 y 7, respectivamente)

Cuando se trata de clasificar un dígito, estas similitudes pueden llegar a ser bastante determinantes y esto puede llevar a los errores de reconocimiento. Otro aspecto importante de este problema es que muchas veces las personas tienden a escribir un dígito de diferentes formas, esto pasa especialmente con el 1, 4, 7 y 9.

Fig. 2. Posibles formas de escritura del número 4

Debido a lo anterior, el reconocimiento se vuelve un problema aún más difícil de lo que era inicialmente, por lo tanto es necesario tener todos estos factores en cuenta al momento de querer dar solución a este problema. Actualmente hay muchas metodologías que son de gran ayuda para abordar y resolver el problema de reconocimiento de una manera parcial, pero no se cuenta con una que sea capaz de resolverlo completamente, sin errores. En consecuencia, este problema es aún hoy en día motivo de investigación y estas investigaciones se acercan cada vez más a la solución definitiva.

III. METODOLOGÍA IMPLEMENTADA

La solución que se plantea en este proyecto, tiene su fundamento en la descomposición de valores singulares (SVD). Para poder utilizar esta metodología, se debe hacer un pre-procesamiento tanto de las imágenes de entrenamiento que se van a utilizar como de la imagen de entrada en cuestión, que es la que contiene el dígito que se quiere reconocer. Dicho pre-procesamiento se debe de hacer, ya que las imágenes utilizadas están en escala de grises, pero a la hora de representarlas dentro del programa, se deben tratar como imágenes en blanco y negro.

Para esto, se cambia al formato de cada imagen a monocromático al momento de traerlas a nuestro programa.

Para cada dígito i (del 0 al 9), se debe tener una matriz de prueba A_i de dimensiones m^2xn, donde cada imagen digitalizada tiene una resolución de mxm pixeles y n es el número de imágenes de entrenamiento utilizadas para cada dígito. En este caso, se manejan imágenes de $28x28$ y se cargan 500 imágenes, por lo que por cada dígito i se tiene una matriz de prueba de $784x500$.

$$\begin{pmatrix} A_{1,1} & \cdots & A_{1,500} \\ \vdots & \ddots & \vdots \\ A_{784,1} & \cdots & A_{784,500} \end{pmatrix}$$

Fig. 3. Estructura de cada matriz A_i

Cada columna de la matriz A_i va a almacenar los 784 valores de color de los pixeles de cada imagen de entrenamiento. Dado que se representan como imágenes en blanco y negro, en vez de utilizar 255 para representar el blanco, se utiliza el número 1. Así, cada matriz va a contener 0's y 1's, lo que simplifica considerablemente las operaciones entre matrices, comparado con matrices que contienen 0's y 255's.

Para representar la imagen de entrada, que llamaremos z, se utiliza una matriz de dimensiones $784x1$. Se podría utilizar un vector unidimensional, pero esto complicaría el problema, ya que en la librería utilizada entre otras cosas para las operaciones entre matrices (JAMA), no está definida la multiplicación entre matrices y vectores, pero sí entre matrices.

$$\begin{pmatrix} Z_{1,1} \\ Z_{1,1} \\ \vdots \\ \vdots \\ Z_{784,1} \end{pmatrix}$$

Fig. 4. Estructura del dígito de entrada z

Para la representación de las matrices, se hizo uso de la librería TNT. Para las operaciones entre estas matrices y la descomposición en valores singulares se utilizó la librería JAMA.

Teniendo las matrices de prueba A_i y el dígito de entrada z, se puede proceder a realizar el reconocimiento, haciendo uso de la siguiente ecuación para calcular el residual relativo entre el dígito de entrada z y cada matriz de prueba A_i.

$$\frac{\left\|(I - U_k {U_k}^T)z\right\|_2}{\|z\|_2}$$

Ecuación (1).

En la ecuación (1), I es la matriz identidad de dimensiones $784x784$, U_k es la matriz U de la descomposición en valores singulares de la matriz A y k es el rango de la matriz A.

$$A = U\Sigma V^*$$

Ecuación (2)

Para hacer la descomposición en valores singulares de cada matriz A, se utiliza la librería JAMA y se obtiene la matriz U, que es la que nos interesa obtener para reconocer el digito.

Para reconocer el dígito, se debe calcular el residual relativo entre z y cada matriz A_i haciendo uso de la ecuación (1). Posteriormente, se deben comparar estos valores para determinar el mínimo residual relativo entre los calculados. Una vez calculado el mínimo, si éste se da con una matriz A_i, donde $0 \leq i \leq 9$, se puede concluir que el dígito de la imagen de entrada es el dígito i.

$$min \left(\frac{\left\|(I - U_k {U_k}^T)z\right\|_2}{\|z\|_2}\right);\ U = A_i(V^*)^{-1}\Sigma^{-1} \quad 0 \leq i \leq 9$$

Se puede ver así que el problema queda reducido a una comparación entre 10 números obtenidos mediante los cálculos de los residuales.

IV. VALIDACIÓN

Después de la implementación del algoritmo, se procede a realizar las respectivas pruebas, las cuales permiten saber si el algoritmo realiza correctamente el reconocimiento de los dígitos. Para éstas pruebas se seleccionan imágenes de la base de datos MNIST, que pueden ser imágenes pertenecientes al conjunto de entrenamiento o imágenes del conjunto de prueba. Posteriormente se explicará cómo influye ésta selección en los resultados obtenidos.

Cuando se ha seleccionado una imagen, se obtienen los residuales calculados con cada una de las matrices de los dígitos de prueba. Para el reconocimiento de la imagen del dígito ingresado, se toma el dígito de la matriz de prueba con la cual el residual calculado sea menor.

A. Reconocimiento de un dígito bien definido

Fig. 5. Imagen de prueba dígito 2

La figura 5, muestra una imagen del dígito 2, esta fue una de las imágenes que se tomaron para realizar las respectivas pruebas, como se puede observar, la forma del dígito es bastante fina y en consecuencia, el reconocimiento del dígito fue muy claro.

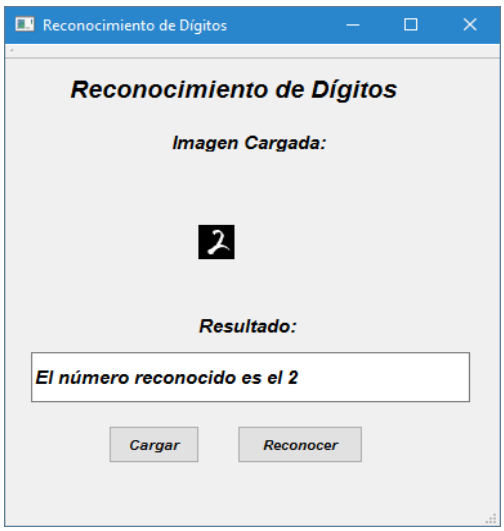

Fig. 6. Salida del programa para la imagen de la Fig. 5

B. Reconocimiento de un dígito no tan bien definido

Fig. 7. Imagen de prueba dígito 2

La figura 7, muestra otra imagen del dígito 2, la cual también fue utilizada para comprobar el funcionamiento del algoritmo. Como se puede ver, la forma del dígito no está muy bien definida, tanto así que a simple vista podría parecerse más a un 0. Infortunadamente, el programa lo reconoce como un 0 y no como un 2, como se muestra en la Figura 8.

Fig. 8. Salida del programa para la imagen de la Fig. 7

Las pruebas realizadas para la validación del algoritmo fueron muchas, además que se trató de que fueran muy variadas. Diferentes dígitos, escritos de diferente forma y dígitos muy similares a otros. Como se ha dicho anteriormente, el algoritmo no es 100% efectivo, pero en la mayoría de los casos se obtiene un buen resultado.

V. ANÁLISIS DE RESULTADOS

Durante la realización de las pruebas, se pudieron observar ciertas particularidades en los resultados obtenidos. En primer lugar, hay una diferencia considerable en los resultados cuando las imágenes que se cargan pertenecen al conjunto de entrenamiento o al conjunto de prueba. A continuación se muestra un ejemplo de esta diferencia en los resultados:

A. Resultados al usar diferentes cantidades de imágenes de entrenamiento

Fig. 9. Imagen N° 200 del dígito 6

Si se carga la imagen de la Figura 9 en el programa y se realiza el reconocimiento usando un número diferente de imágenes de entrenamiento, se hace el reconocimiento correctamente, pero los residuales calculados por el programa difieren significativamente, como se muestra en las siguientes capturas de la salida del programa.

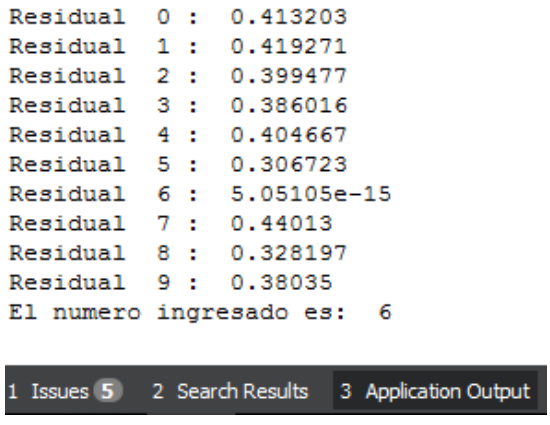

Fig. 10. Resultado imagen N° 200 del dígito 6

```
Residual  0 :  0.645952
Residual  1 :  0.656071
Residual  2 :  0.644837
Residual  3 :  0.614686
Residual  4 :  0.645498
Residual  5 :  0.595381
Residual  6 :  0.357831
Residual  7 :  0.718504
Residual  8 :  0.557116
Residual  9 :  0.665586
El numero ingresado es:  6
```

1 Issues 5 2 Search Results 3 Application Output

Fig. 11. Resultado imagen N° 200 del dígito 6

La figura 10 muestra los resultados obtenidos para el reconocimiento de la imagen N° 200 del dígito 6. En esta prueba el número de imágenes cargadas era 300, por lo tanto, esta imagen pertenecía al conjunto de las imágenes de entrenamiento. La figura 11 muestra los resultados para la misma imagen, cuando el número de imágenes cargadas era 100, lo que quiere decir que ésta imagen no pertenecía al conjunto de imágenes de entrenamiento cargadas.

Se puede observar que cuando la imagen a reconocer pertenece al conjunto de imágenes de entrenamiento cargadas, los resultados obtenidos son más exactos. Aunque claramente existe una diferencia en los resultados obtenidos, es preciso aclarar que el reconocimiento del dígito fue exitoso en ambos casos.

B. Resultados al ingresar una imagen que se podría interpretar de dos formas

Un problema que se puede presentar a la hora de querer reconocer el digito de una imagen, es que ésta esté escrita de forma tal que pueda ser interpretada de varias maneras. Este es el caso de la imagen de la Figura 9 que se muestra a continuación:

Fig. 12. Imagen N° 593 del dígito 3

Si se observa la Figura 12, se puede ver que la imagen cargada del dígito 3 tiene una similitud con el dígito 8. Ahora, al ingresar la imagen al programa, los residuales calculados son los siguientes:

```
Residual  0 :  0.176104
Residual  1 :  0.514369
Residual  2 :  0.169338
Residual  3 :  0.0595053
Residual  4 :  0.0784853
Residual  5 :  0.123897
Residual  6 :  0.227155
Residual  7 :  0.215045
Residual  8 :  0.0691919
Residual  9 :  0.0853186
El numero ingresado es:  3
```

1 Issues 5 2 Search Results 3 Application Output

Fig. 13. Captura de los residuales obtenidos y el reconocimiento del dígito ingresado

Debido a la similitud ya mencionada, en la Figura 13 se puede ver que los menores residuales calculados fueron aquellos con las matrices de prueba del 3 (0.0595053) y del 8 (0.0691919). Aun así, el reconocimiento se realiza de manera correcta, pues el programa toma el menor residual entre todos.

C. Rango de las matrices de prueba

Al momento de calcular la matriz U_k, se necesita primero calcular la matriz U, para luego sólo tomar k columnas, donde estas k filas son linealmente independientes. Así, se tiene que k es el rango de la matriz A.

```
Rango  0 :  400
Rango  1 :  227
Rango  2 :  400
Rango  3 :  400
Rango  4 :  400
Rango  5 :  400
Rango  6 :  400
Rango  7 :  400
Rango  8 :  400
Rango  9 :  377
```

1 Issues 5 2 Search Results 3 Application Output

Fig. 14. Rango de las matrices utilizando 400 imágenes

```
Rango  0 :  438
Rango  1 :  239
Rango  2 :  479
Rango  3 :  444
Rango  4 :  449
Rango  5 :  459
Rango  6 :  432
Rango  7 :  447
Rango  8 :  435
Rango  9 :  393
```

1 Issues 5 2 Search Results 3 Application Output

Fig. 15. Rango de las matrices utilizando 500 imágenes

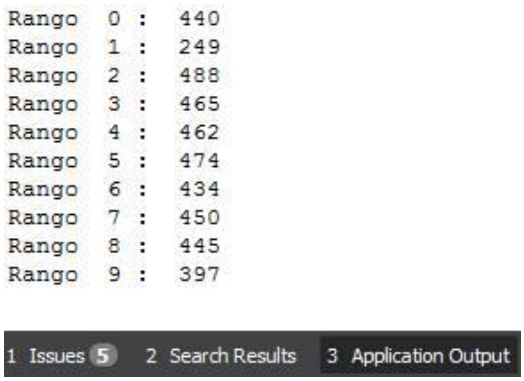

Fig. 16. Rango de las matrices utilizando 600 imágenes

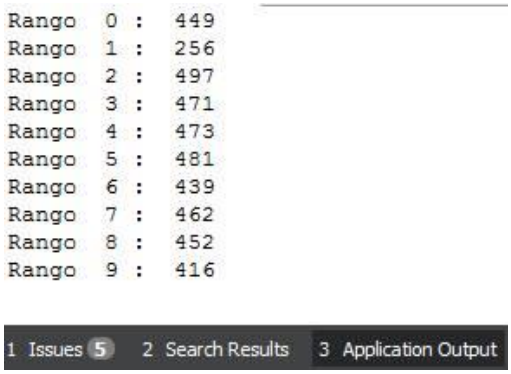

Fig. 17. Rangos de las matrices utilizando 700 imágenes

Ahora, al ejecutar varias veces el programa con diferentes cantidades de imágenes de entrenamiento (400, 500, 600 y 700), se evidencia un cambio en el valor del rango de las matrices, pero se observa que dicho valor siempre es menor que 500. Este comportamiento se puede evidenciar en las figuras 14 15, 16 y 17, respectivamente. Por esta razón, se decide utilizar 500 imágenes de entrenamiento.

Se puede analizar que diferentes resultados pueden ser obtenidos, dependiendo del número de imágenes de prueba que se cargan y de la claridad del dígito de la imagen de entrada.

VI. CONCLUSIONES

Se analiza que la metodología utilizada para resolver el problema de reconocimiento no funciona de manera correcta en el 100% de los casos, ya que se pueden dar casos en los cuales el dígito que se quiere reconocer ha sido escrito de forma tal que hasta para un humano es difícil reconocerlo.

Se logra entender la importancia de la descomposición en valores singulares para el reconocimiento de dígitos, y del álgebra lineal en general para resolver problemas de la misma naturaleza que requieran manipulación de imágenes.

Con este proyecto, se logra tener un acercamiento a uno de los problemas reales del mundo, relacionado con un tema tan importante para el manejo de datos como lo es el manejo de imágenes.

REFERENCIAS

[1] L. Eldén, *Numerical Linear Algebra and Applications in Data Mining and IT*. Cambridge [England]: Cambridge University Press, 2006.