

Diseño de un sistema de reconocimiento automático de palabras contenidas en documentos históricos

Edgar Roberto Pérez Serrano (1), Dra. Ma. de Guadalupe García Hernández (2)

(1) Alumno de Ingeniería en Mecatrónica, DICIS | edgarrobertoperezserrano@gmail.com

(2) Profesora Titular del Departamento de Ingeniería Electrónica, DICIS | garciag@ugto.mx

División Ingenierías del Campus Irapuato-Salamanca (DICIS), Universidad de Guanajuato,
Carretera Salamanca - Valle de Santiago Km. 3.5 + 1.8, Comunidad de Palo Blanco,
Salamanca, Gto. C.P. 36885

Resumen

El proyecto marco propone un sistema basado en una red neuronal convolucional con el objeto de reconocer letras y encontrar secuencias óptimas de estas para reconocer palabras en textos históricos, los cuales presentan un deterioro en sus páginas a través de los años. Para ello se implementaron dos interfaces de usuario: una para etiquetar palabras y otra para segmentar las palabras en letras, y con ellas generar las bases de datos correspondientes. Estas implementaciones se hicieron mediante programación *MatLab*. Para el entrenamiento de las redes neuronales convolucionales se utilizó el programa *LearnBWP* en el cual se puede ver gráficamente su entrenamiento de aprendizaje. De igual manera se convirtió el formato de cada red entrenada para ser ejecutada por el programa *charreco*, que verifica que el entrenamiento sea óptimo, ya que así es posible que el sistema reconozca eficientemente las letras de cada palabra.

Abstract

The Marco project propose a system based in a convolutional neuronal network with the objective to recognize letter and find optimal sequences of this to recognize words in historic texts, which this one presents a deterioration in this pages through the years. Was implemented two graphic user interfaces required, one to label words and another to segment the words in characters, and they generate the corresponding databases. This implementations was made by *MatLab* system. For the training of the convolutional neuronal network convolutional, was used the program *LearnBWP* which it can see clearly the graphs of their training of learning. Similarly the format of each trained to be executed by the *charreco* program, which verifies that the network is optimal training became as well the system can efficiently recognize the letters of each word

Palabras clave

Reconocimiento automático de palabras, red neuronal convolucional.

INTRODUCCIÓN

El reconocimiento automático de palabras ha sido sujeto de investigación por muchos años y uno de sus objetivos es aplicar Ciencias de la Computación en la identificación de forma certera y rápida el contenido de textos, especialmente provenientes de documentos históricos, que por el paso del tiempo ya no se encuentran en óptimas condiciones. Por ello es que se ha creado la necesidad de desarrollar sistemas de reconocimiento eficientes, que obtengan y salvaguarden la información invaluable que contienen los textos antiguos, ya que lamentablemente los libros físicos se van deteriorando con el tiempo hasta llegar a ser ilegibles.

Los pasos habituales en el reconocimiento de texto son: (i) pre-procesamiento del texto para evitar información indeseable como son manchas o desgaste de tinta que causa confusión durante el reconocimiento, (ii) segmentación del texto en palabras y de palabras a letras, (iii) reconocimiento de letras y de palabras y (iv) posprocesamiento para verificar la eficiencia del reconocimiento de palabras.

El reconocimiento de palabras ha sido tradicionalmente abordado mediante dos enfoques: analítico y holístico [2,3]. El enfoque analítico considera a la palabra como un conjunto de componentes más pequeños, de esta manera trata de segmentar la palabra para permitir la identificación de dichos componentes, mientras que el enfoque holístico evita segmentar, tratando de reconocer la palabra como un todo [4]. Cabe señalar que estos dos enfoques pueden ser combinados con el objeto de explotar sus ventajas [3].

Por otro lado, los documentos antiguos tienen la peculiaridad de que su fondo se encuentra degradado. Esto da como resultado “ruido” o distorsión en la imagen. Se acostumbra hacer una fase de eliminación de ruido para mejorar la calidad de las imágenes de este tipo de documentos.

Las redes neuronales convolucionales constituyen redes neuronales estáticas inspiradas en el funcionamiento del sistema visual humano y que incorporan las nociones de campos receptivos locales, pesos compartidos y sub muestreo

espacial, lo cual las hace tolerantes a cierta cantidad de deformación, traslación y cambios de escala.

Este método ha sido aplicado con éxito a problemas tales como reconocimiento de escritura manuscrita [5,6] reconocimiento de caracteres impresos [7], reconocimientos de rostros humanos [8] y detección de rostros humanos [9]. La ventaja principal de este tipo de redes es que presentan buena capacidad de generalización gracias al uso de pesos compartidos [10]

Las redes neuronales convolucionales pueden por sí mismas efectuar la extracción de características y por lo tanto las características extraídas son adaptadas según la tarea (p. e., reconocimiento de caracteres, de palabras y de rostros). Por esta razón, las características extraídas podrían no corresponder a las que presumiblemente utiliza el sistema visual humano.

Sin embargo, dado que estas redes efectúan convoluciones de la imagen mediante máscaras, es posible tomar en cuenta algunas características del procesamiento temprano del sistema visual por el uso de máscaras con forma de segmento de recta o de borde [11,12]. Por lo que las redes neuronales convolucionales han resultado adecuadas para procesar y reconocer texto en documentos históricos.

En el proyecto marco se fijaron los objetivos de realizar la segmentación de palabras con su respectivo etiquetamiento de un documento histórico al igual que segmentar de las palabras en caracteres para poder etiquetarlos, para generar dos tipos de bases de datos (una de palabras etiquetadas y otra de caracteres provenientes de las palabras etiquetadas).

Así como el entrenamiento de redes neuronales convolucionales para el reconocimiento de palabras y caracteres en el documento.

MATERIALES Y MÉTODOS

Se utilizó programación *MatLab* para la implementación de dos interfaces gráficas de usuario de igual manera se utilizó el programa *LearnBPW* para el entrenamiento de las redes neuronales convolucionales, adicionalmente se requieren dos programas más *convert2* y

printsharednet para convertir el formato de la red neuronal entrenada a un documento de texto.

El programa para el etiquetaje de palabras se le llamó *textreco.m*. Con el cual se generan dos ventanas. La primera ventana (ver Figura 1) que sirve para elegir el acercamiento que se desea en el documento y con en este se puede ver la página completa con un recuadro que indica la parte del documento en la que se encuentra el usuario.

La segunda ventana (ver Figura 2) es la interfaz de usuario para realizar el etiquetaje de las palabras. Esta ventana contiene las siguientes opciones: abrir archivo, crear recuadro de texto, eliminar recuadro, editar texto, guardar trabajo, crear base de datos y opciones para poder moverse por las páginas del archivo.

Cuando se cuenta con una base de palabras segmentadas y etiquetadas se procede con la segmentación de palabras en caracteres o letras, que servirá para generar una base de caracteres etiquetados, lo cual servirá para entrenar a la red neuronal con los caracteres pertenecientes al texto.

La segmentación de palabras a caracteres se hizo en otra interfaz llamada *charreco.m* (ver Figura 3) en la cual se visualizará la palabra a segmentar y los botones para realizar las siguientes tareas: segmentar palabra, guardar trabajo realizado, eliminar recuadros, generar base de datos, realizar en acercamiento en las palabras para hacer más precisa la segmentación y un recuadro que ayuda a posicionarse en una palabra específica. Además contiene una opción para utilizar dos filtros utilizar, uno es la escala de grises y otro es de pasa bajos para visualizar mejor las imágenes (ver Figura 4).

Para el entrenamiento de la red neuronal convolucional se adaptó y utilizó el programa *LearnBPW*. Esta es una herramienta con interfaz visual para el entrenamiento de redes neuronales. Al ejecutar *LearnBPW* aparecen dos ventanas, en la primera se pueden identificar los siguientes elementos: en la parte superior se encuentran tres menús, llamados *Files*, *View* y *Tools*. En seguida se encuentra cuatro pestañas que muestran diferentes gráficas en la pantalla.

La primera pestaña muestra la gráfica del porcentaje de reconocimiento en prueba y aprendizaje contra el número de ciclos habidos hasta el momento de la ejecución (Figura 5), en

otras dos pestañas aparecen las gráficas del error *MSE (Mean Squared Error)* y *MCE (Mean Cross-Entropy Error)* respectivamente, y la última *Data* muestra los valores de cada una de las gráficas.

La segunda ventana llamada *Parameters*, en ella se configuran los valores para iniciar el entrenamiento, como son: número de ciclos, semilla aleatoria para inicializar sus pesos, tasa de aprendizaje y su momento, los valores máximo y mínimo de inicialización de los pesos, tipo de función de error, opción de activar la partición de la base de datos y número de particiones (Figura 6).

Para que la aplicación inicie con el entrenamiento se deben realizar el siguiente procedimiento:

1. Se carga el archivo con la red neuronal diseñada.
2. Se carga la base de datos *chardatabase.dat*.
3. Se configuran los valores del entrenamiento (Figura 6).
4. Se presiona el botón de inicio del entrenamiento.
5. Se Guarda el entrenamiento de la red neuronal.

Para importar la red neuronal entrenada por medio del programa *LearnBPW* primero se convierte ésta a un determinado formato para ser capaz de leerla desde *Matlab*.

Para convertir la red neuronal guardada por el programa *LearnBPW* a formato de texto se necesitan los siguientes programas: *convert2* y *printsharednet*, y se realiza el siguiente procedimiento:

1. Se utiliza la librería *printsharednet* desde un archivo *batch (convert2)*.
2. Se cambia el nombre de la red neuronal que se desea convertir (nombre de la red *.net* a nombre de la red *.txt* ver Figura 7).

3. Se guarda el archivo y se ejecuta el programa *convert2* para generar el documento de texto (.txt).
4. Al documento de texto (.txt) generado se cambia de nombre a *network* (ver Figura 8) para que pueda cargar.
5. Se utiliza la función *ReadCNN* para cargar del archivo de texto a la red neuronal.
6. Se visualiza a través de la interfaz *charreco.m* (ver Figura 3).

RESULTADOS Y DISCUSIÓN

Se realizó el etiquetaje semiautomático de palabras y la segmentación de palabras en letras o caracteres, provenientes del documento histórico “Índice General de la Recopilación de Leyes de las Indias, año 1681”, ubicado en la Biblioteca “Armando Olivares” de la Universidad de Guanajuato. Con las imágenes de las páginas de este documento se pudieron generar las bases de datos correspondientes, una para palabras y otra para caracteres o letras, con el objeto de entrenar la red neuronal convolucional implementada en MathLab. De esta manera también se pudo ver que el entrenamiento de la red neuronal convolucional fue bueno, ya que su porcentaje de aprendizaje fue **mayor al 90%**. También es importante señalar que se tiene hasta el momento un *corpus* de 19,257 imágenes de palabras etiquetadas y 5,789 de estas ya segmentadas en letras, lo cual crea una gran referencia para que el programa pueda entrenarse de forma considerablemente eficiente.

CONCLUSIONES

Las redes neuronales convolucionales tienen amplio campo de aplicación para diversas áreas de Ciencias de la Computación, como lo es el reconocimiento de palabras y de caracteres de un documento histórico. Con este tipo de interfaces y programas desarrollados se pueden salvaguardar documentos históricos de considerable importancia para su consulta y edición.

Algunos de los retos que presenta el programa de reconocimiento se deben a la inadecuada segmentación de palabras en letras, en el que la

persona puede dejar parte de otra letra dentro del entorno de la letra ya segmentada, produciendo que el entrenamiento de la red neuronal no sea el óptimo. Es por esto que se necesita que los caracteres estén bien segmentados para generar una buena base de datos, de modo que su entrenamiento y posterior reconocimiento sean óptimos.

Actualmente se está implementando el software necesario para la unión de letras en palabras, la unión de palabras en oraciones y estas a su vez en párrafos. De esta manera se podrá copiar, editar y pegar la información en un documento *Word*.

AGRADECIMIENTOS

En especial agradecimiento a la Dra. Ma. de Guadalupe García Hernández y a la Universidad de Guanajuato por permitirme realizar el Verano de Investigación, al igual que por contribuir a mi formación académica con la participación en este proyecto.

REFERENCIAS

- [1]. Nasien, D.; Haron, H.; Yuhaniz, S.S., "Support Vector Machine (SVM) for English Handwritten Character Recognition," Second International Conference on Computer Engineering and Applications (ICCEA 2010), Vol. 1, pp. 249-252.
- [2]. Steinhilber, T., Rivlin, E., & Intrator, N. (1999). Off-line cursive script word-recognition: A survey. *International Journal of Document Analysis and Recognition*, 2 (2), pp. 90-110.
- [3]. Casey, R. G., & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18 (7), pp. 690-706.
- [4]. Madvanath, S., & Govindaraju, V. (2001). The Role of Holistic Paradigms in Handwritten Word Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23 (2), pp. 149-164.
- [5]. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proc. of the IEEE*, 86 (11), pp. 2278-2326.
- [6]. Martin, G., Rashid, M., & Pittman, J. A. (1993). Integrated Segmentation and Recognition Through Exhaustive Scans or Learned Saccadic Jumps. *Int Journal of Pattern Recognition and Artificial Intelligence*, 7(4), pp. 831- 847.
- [7]. Wang, J., & Jean, J. (1994). Segmentation of Merged Characters by Neural Networks and Shortest Path, *Pattern Recognition*, 27(5), pp. 649- 658.

- [8]. Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face Recognition: A Convolutional-network Approach. IEEE Trans. Neural Networks, 8 (1), pp. 98-113.
- [9]. Garcia, C., & Detakis, M. (2004). Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection. IEE Trans. on Pattern Analysis and Machine Intelligence, 26 (11), pp. 1408-1423.
- [10]. Lecun, Y., & Bengio, Y. (1995). Convolutional Networks for images, Speech, and Time-Series. In M. A. Arbib (Ed.). Handbook of Brain Theory and Neural Nets (pp. 265-267), Cambridge, MA: MIT Press.
- [11]. Hubei, D., & Wiesel, T. (1962). Receptive fields, binocular Interaction and functional architecture in the cat's visual cortex. Journal of Physiology, Vol. 160, pp. 106-154.
- [12]. Marr, D. (1978). Representing Visual Information-a computational approach. In A. Hanson and E. Riseman (Eds.), Computer Vision Systems (pp. 61-80), New York: Academic Press.

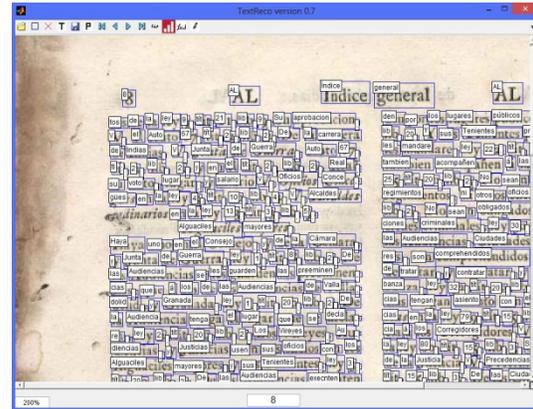


Figura 3. Segmentación de palabras con charreco.m



Figura 1. Acercamiento con textreco.m

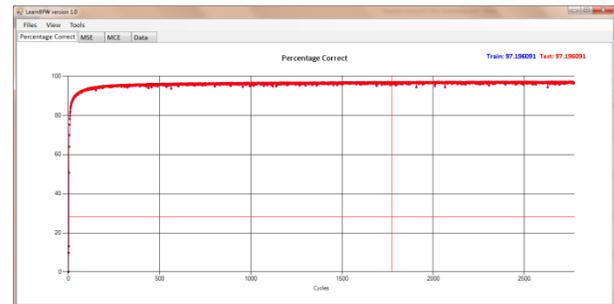


Figura 4. Imagen sin filtro (arriba) y con filtro (abajo).

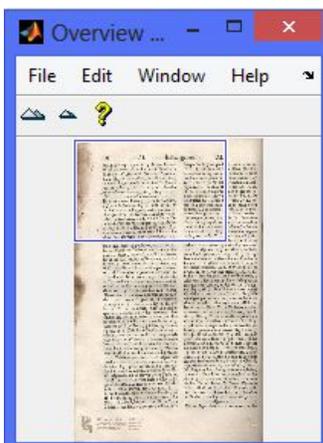


Figura 2. Etiquetaje de palabras con textreco.m

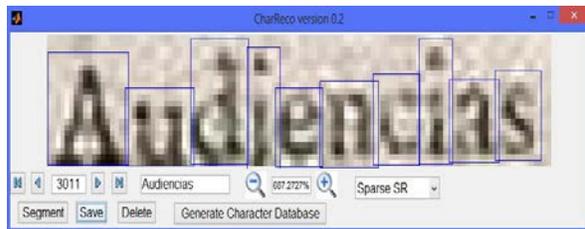


Figura 5. Gráfica del porcentaje de reconocimiento en prueba y aprendizaje contra el número de ciclos.



Figura 6. Configuración de valores para entrenamiento

