

FPGA-based translation system from Colombian sign language to text

Original

FPGA-based translation system from Colombian sign language to text / Guerrero-Balaguera, Juan-David.; Perez-Holguin, W. J.. - In: DYNA. - ISSN 0012-7353. - STAMPA. - 82:189(2015), pp. 172-181. [10.15446/dyna.v82n189.43075]

Availability:

This version is available at: 11583/2878260 since: 2021-03-29T16:46:05Z

Publisher:

Universidad Nacional de Colombia

Published

DOI:10.15446/dyna.v82n189.43075

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

FPGA-based translation system from colombian sign language to text

Juan David Guerrero-Balaguera ^a & Wilson Javier Pérez-Holguín ^b

^a Grupo GIRA, Universidad Pedagógica y Tecnológica de Colombia, Sogamoso, Colombia.. juandavid.guerrero@uptc.edu.co

^b Grupo GIRA, Universidad Pedagógica y Tecnológica de Colombia, Sogamoso, Colombia. wilson.perez@uptc.edu.co

Received: April 22th, 2014. Received in revised form: September 9th, 2014. Accepted: October 2th, 2014.

Abstract

This paper presents the development of a system aimed to facilitate the communication and interaction of people with severe hearing impairment with other people. The system employs artificial vision techniques to the recognition of static signs of Colombian Sign Language (LSC). The system has four stages: Image capture, preprocessing, feature extraction and recognition. The image is captured by a digital camera TRDB-D5M for Altera's DE1 and DE2 development boards. In the preprocessing stage, the sign is extracted from the background of the image using the thresholding segmentation method; then, the segmented image is filtered using a morphological operation to remove the noise. The feature extraction stage is based on the creation of two vectors to characterize the shape of the hand used to make the sign. The recognition stage is made up a multilayer perceptron neural network (MLP), which functions as a classifier. The system was implemented in the Altera's Cyclone II FPGA EP2C70F896C6 device and does not require the use of gloves or visual markers for its proper operation. The results show that the system is able to recognize all the 23 signs of the LSC with a recognition rate of 98.15 %.

Keywords: FPGA; Colombian Sign Language (LSC); Image Processing; Sign Language Recognition; Artificial Neural Networks.

Sistema traductor de la lengua de señas colombiana a texto basado en FPGA

Resumen

Este trabajo presenta el desarrollo de un sistema diseñado para facilitar la comunicación e interacción de personas con discapacidad auditiva severa con las demás personas. El sistema emplea técnicas de visión artificial para el reconocimiento de las señas estáticas de la Lengua de Señas Colombiana (LSC). El sistema tiene cuatro etapas: Captura de la imagen, procesamiento, extracción de características y reconocimiento. La imagen es capturada mediante una cámara digital TRDB-D5M diseñada para tarjetas de desarrollo de DE1 y DE2 Altera. En la etapa de procesamiento, la seña es extraída del fondo de la imagen mediante el método de segmentación por umbral; posteriormente, la imagen segmentada es filtrada usando una operación morfológica para eliminar el ruido. La etapa de extracción de características está basada en la creación de dos vectores que caracterizan la forma de la mano mediante la que se realiza la seña. La etapa de reconocimiento está constituida por una red neuronal artificial perceptrón multicapa (MLP), la cual actúa como clasificador. El sistema fue implementado en el dispositivo FPGA Cyclone II EP2C70F896C6 y no requiere el uso de guantes o marcadores visuales para su correcto funcionamiento. Los resultados muestran que el sistema tiene la capacidad para reconocer todas las 23 señas estáticas de la LSC con una tasa de reconocimiento del 98.15 %.

Palabras clave: FPGA; Lengua de Señas Colombiana (LSC); Procesamiento de Imágenes; Reconocimiento de Lengua de Señas; Redes Neuronales Artificiales.

1. Introducción

Según la organización mundial de la salud OMS [1], más del 5% de la población mundial, correspondiente a 360 millones de personas, padece pérdida incapacitante de la audición. En Colombia, más de 2'600.000 personas – equivalente al 6.7% de la población–, posee algún tipo de

discapacidad, de la cual el 17.4% tiene dificultades para oír, aún con aparatos especiales. Además, solo el 65% de este grupo de personas sabe leer y escribir y el 43% del total registrado vive en estrato 1 [2].

La Lengua de Señas (LS) es el principal método de comunicación empleado por las personas con déficit de audición para interactuar con las demás personas. Está

compuesta por un conjunto estructurado de gestos, movimientos, posturas y expresiones faciales que corresponden a letras y/o palabras. La Lengua de Señas no tiene una estructura generalizada alrededor del mundo, por lo cual, cada país tiene su propia lengua de señas, entre las que se encuentran: la Lengua de Señas Americana (ASL), la Lengua de Señas Colombiana (LSC), la Lengua de Señas Alemana (GSL), etc. [3,4].

Las personas con déficit de la audición interactúan entre sí utilizando dicho lenguaje; sin embargo, se requiere de un intérprete para comunicarse con las personas que no comprenden su lengua. Esta situación crea una barrera en la comunicación y restringe la vida cotidiana de las personas con este tipo de limitaciones causando aislamiento y frustración [1,3,5].

Dadas las dificultades que presenta la población con discapacidad auditiva, la comunidad científica se ha interesado en el desarrollo de técnicas, procedimientos y sistemas para el reconocimiento de los gestos correspondientes a la lengua de señas. Sin embargo, la eficiencia de estos es limitada debido a la complejidad de la estructura de dicho lenguaje y a la elevada capacidad de procesamiento requerido. Por estas razones, éste sigue siendo un problema de investigación abierto que motiva el interés de muchos investigadores alrededor del mundo [3].

Los avances más significativos en el desarrollo de sistemas de reconocimiento de la lengua de señas se pueden agrupar en dos categorías: sistemas basados en visión artificial [6-10] y sistemas basados en el uso de guantes instrumentados [11-13]. A diferencia de los sistemas basados en el uso de guantes instrumentados, los sistemas basados en técnicas de visión artificial permiten una interacción más natural con el usuario ya que no requiere el uso de aditamentos especiales. Sin embargo, estas técnicas requieren capturar, procesar y reconocer imágenes, lo cual exige altas prestaciones de procesamiento paralelo de datos [14].

Las FPGAs (*Field Programmable Gate Array*) presentan la característica inherente de realizar procesamiento concurrente de alta eficiencia, debido a su arquitectura y a las ventajas de los lenguajes empleados para describir los circuitos y sistemas a ser implementados en este tipo de tecnología. Esto ha favorecido el desarrollo de aplicaciones de reconocimiento de los gestos, como el trabajo presentado en [15] en el que se describe la implementación en FPGA de un sistema que cuenta la cantidad de dedos mostrados en diferentes gestos realizados con las manos. Oniga et al. [16], presentan la implementación en FPGA de una red neuronal artificial para el reconocimiento de posturas estáticas de las manos.

Además de considerar la capacidad de procesamiento requerida por el sistema, es necesario seleccionar un método de reconocimiento suficientemente robusto para identificar correctamente el conjunto de señas empleado. En los trabajos reportados en la literatura se observan diferentes técnicas para el reconocimiento de la lengua de señas tales como las redes neuronales artificiales [17-22], los modelos ocultos de Markov (HMM) [8,11,23,24], y las máquinas de soporte vectorial (SVM) [11,23,25]. En el trabajo presentado por Munib et al. [22], se muestra el desarrollo de

un sistema de visión artificial para el reconocimiento de los gestos estáticos de la Lengua de Señas Americana (ASL), basado en la transformada Hough para extracción de características, y redes neuronales artificiales para el reconocimiento. En [19], se presenta el desarrollo de un sistema de visión artificial para el reconocimiento de los gestos estáticos de la Lengua de Señas Persa (PSL), basado en la transformada Wavelet y redes neuronales artificiales. En [18], se presenta un sistema de visión artificial para el reconocimiento de las señas estáticas de la Lengua de Señas Colombiana (LSC), basado en redes neuronales artificiales para su implementación en hardware. En [7], se presenta un sistema de reconocimiento de la Lengua de Señas Árabe, mediante el uso de un sistema neuro-difuso adaptativo. Cabe mencionar que ninguno de los trabajos citados realiza la implementación hardware de los métodos de reconocimiento descritos.

Otro aspecto importante en el reconocimiento de la Lengua de Señas es determinar la(s) característica(s) que permite(n) diferenciar una seña de otra; la forma de la mano es una de estas. De acuerdo con [26] existen varios métodos para extraer características basadas en la forma de la mano, entre los que se encuentran: descriptores de Fourier (FD), descriptores modificados de Fourier (MFD), descripción de contorno usando curvatura y características de la transformada de Karhunen Loeve (K-L). En [17] se presenta un método para la extracción de características basado en la proyección de la forma de la mano en dos vectores, a los que se les aplica posteriormente el método FD. Aunque los métodos para la extracción de características aquí citados presentan buenos resultados su implementación está orientada a software únicamente.

En el presente trabajo se muestra el diseño de un sistema basado en visión artificial para el reconocimiento del alfabeto de señas estáticas de la LSC, usando procesamiento de imágenes y una red neuronal artificial MLP. Dicho sistema es implementado en una FPGA de Altera, mediante la tarjeta de desarrollo DE2-70.

El resto del artículo está organizado de la siguiente forma: en la sección 2 se presenta el diseño del sistema conformado por las etapas de captura, preprocesamiento, extracción de características y reconocimiento. En la sección 3 se describen los resultados experimentales obtenidos. En la sección 4 se presentan las conclusiones y se plantean los trabajos futuros.

2. Diseño e implementación del sistema

2.1. Descripción general del sistema

El sistema está diseñado para reconocer y traducir a texto de forma automática 23 señas estáticas del alfabeto de la Lengua de Señas Colombiana (LSC). Este sistema está basado en técnicas de procesamiento de imágenes y redes neuronales artificiales. En la interacción con el sistema, no se requiere el uso de guantes o elementos adicionales para resaltar las manos. En la Fig. 1 se muestra el conjunto de señas empleadas en el sistema de reconocimiento propuesto.



Figura 1. Conjunto de señas del Alfabeto de la Lengua de Señas Colombiana, empleados en el sistema de reconocimiento.
Fuente: Autores

2.2. Captura de la imagen

Las señas a emplear en el sistema propuesto se realizan con una sola mano en un escenario con iluminación constante y fondo de color negro. Las imágenes son adquiridas mediante la cámara digital de 5 Mega Píxeles TRDB-D5M [27] utilizando una resolución de 800x480 píxeles en formato RGB. Las imágenes capturadas son almacenadas en memoria SDRAM, para visualizarlas en la pantalla TRDB-LTM [28] y realizar el posterior procesamiento y reconocimiento.

2.3. Preprocesamiento

2.2.1. Segmentación de la mano mediante umbralización

La segmentación de la mano es el proceso que consiste en extraer de la imagen capturada la seña realizada con la mano. Una eficiente segmentación de la mano es clave para reconocer los signos de la lengua de señas [17]. La región de la mano puede ser extraída del fondo utilizando la segmentación por umbral.

La segmentación por umbral es sencilla de implementar en hardware ya que solo es necesario comparar cada píxel de la imagen con un valor de umbral. Además, es un método óptimo para extraer regiones cuando se emplean imágenes con buena iluminación y una diferencia notable entre la región de interés y el fondo [29].

El método de segmentación por umbral (comúnmente utilizado en imágenes en escala de grises) consiste en seleccionar un valor umbral (o límite) para separar y etiquetar la imagen en dos grupos de píxeles. A esto se le conoce como umbralización global básica [29]. La selección del umbral se realiza con base en el histograma mediante la ubicación del umbral T en la separación o valle, que existe entre las crestas del histograma [29]. El resultado de aplicar la segmentación utilizando el umbral T es una imagen en blanco y negro, donde el color blanco representa el objeto de interés y el color negro representa el fondo.

En éste trabajo, la selección del umbral se realizó experimentalmente mediante el siguiente procedimiento:

1. Capturar una imagen bajo las condiciones de operación del sistema de reconocimiento.
2. Calcular el histograma de cada una de las componentes de color R, G y B de la imagen capturada.
3. Analizar los histogramas observado la existencia de una clara separación o valle entre la región de la mano y el fondo de la imagen. Seleccionar el histograma que más se ajuste a dicho requerimiento.
4. Establecer un valor en la separación o valle del histograma y segmentar la imagen con el valor seleccionado. Este valor se ajusta hasta obtener el resultado deseado. Una vez realizado este ajuste, el valor final es seleccionado como el umbral utilizado en la segmentación.

El procedimiento anterior fue realizado en MATLAB® R2010b [30], obteniendo como resultado un valor de umbral de 125 para la componente de color rojo. En la Fig. 3 se muestra el histograma de la componente de color rojo y el umbral seleccionado.

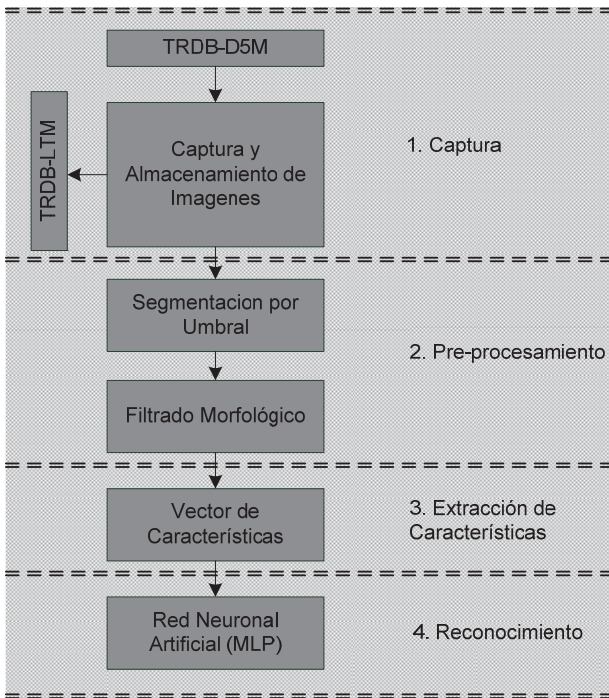


Figura 2. Diagrama general del sistema.
Fuente: Autores

El sistema de reconocimiento diseñado se compone de cuatro etapas: Captura de la imagen, preprocesamiento, extracción de características y reconocimiento. En la Fig.2 se muestra el diagrama de bloques del sistema de reconocimiento propuesto.

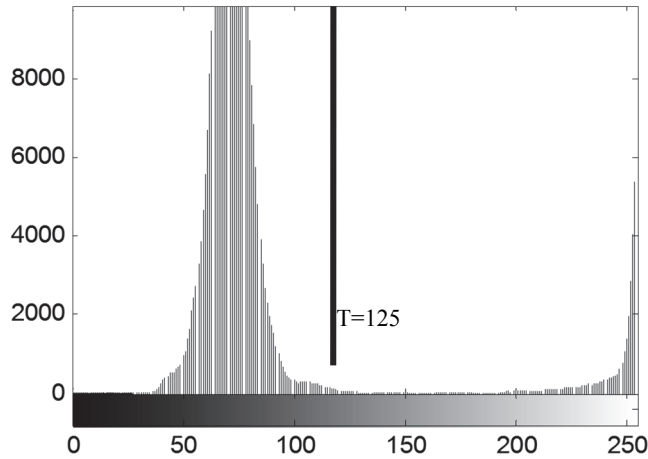


Figura 3. Histograma de la componente de color rojo con un umbral de 125 seleccionado para la segmentación.
Fuente: Autores

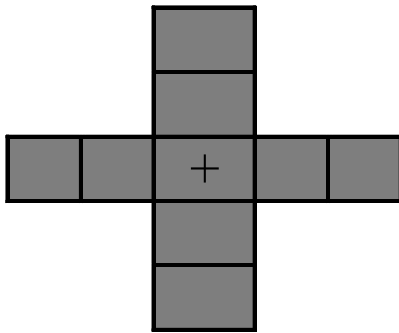


Figura 4. Elemento estructurante utilizado para filtrado morfológico.
Fuente: Autores

2.2.2. Filtrado morfológico

Debido al tipo de segmentación utilizado la imagen obtenida puede contener ruido y errores. Por lo tanto, se utiliza la operación morfológica de apertura para filtrar dicho ruido.

La operación morfológica de apertura se denota como $A \circ B$ donde A es la imagen binaria y B es el elemento estructurante. Dicha operación está definida como la erosión de A por B , seguida de la dilatación por B del resultado obtenido [29]. En la eq. (1) se presenta la operación morfológica de apertura en términos de las operaciones morfológicas erosión y dilatación. El elemento estructurante utilizado se muestra en Fig. 4. La Fig.5 muestra la imagen obtenida como resultado del preprocesamiento.

$$A \circ B = (A \ominus B) \oplus B \quad (1)$$

2.4. Extracción de Características

Una vez realizado el preprocesamiento, se obtiene una imagen binaria de la mano que representa una señal particular. Para reconocer esta señal, es necesario extraer ciertas características de la imagen. La forma es una característica importante que permite identificar un objeto.



Figura 5. Imagen resultado del pre-procesamiento.
Fuente: Autores

Existen muchos métodos para representar y describir una forma en particular. En este trabajo, se presenta un método que permite caracterizar la señal con base en la transformación de la forma de la mano en dos vectores.

El método de transformación propuesto consiste en proyectar la forma de la mano en dos vectores que contienen la cantidad de píxeles por fila y columna del objeto dentro de la imagen. El cálculo de estos vectores se realiza mediante dos algoritmos, el primero realiza un recorrido por las filas de la imagen para calcular el vector R , y el segundo hace un recorrido por las columnas de la imagen para calcular el vector C . Dichos algoritmos fueron diseñados para ser utilizados con imágenes binarias de cualquier resolución, por lo cual se emplean las siguientes definiciones: i) sea I un arreglo bidimensional de tamaño $(n \cdot m)$, el cual representa la imagen binaria obtenida en la etapa de preprocesamiento, ii) sean i y j los índices de selección de un píxel en la i -ésima fila y j -ésima columna, y iii) sean R y C los vectores de transformación que se desean calcular. Los algoritmos desarrollados se presentan en la Fig. 6.

En la Fig. 7 se muestra un ejemplo gráfico del cálculo de los vectores R y C a partir de una imagen binaria de 8×10 píxeles. En esta figura se observa que los vectores de transformación resultantes no dependen de la posición que ocupe el objeto dentro de la imagen, por lo que pueden considerarse buenos descriptores de la forma de la mano independientemente de su posición. Sin embargo, los vectores obtenidos tienen longitud dinámica, ya que las señales realizadas pueden tener cambios en su tamaño. Esto quiere decir que si una señal ocupa n_h filas por m_h columnas de la imagen, los vectores de transformación R y C dependerán de dichos valores.

La naturaleza dinámica de los vectores R y C dificulta la etapa de reconocimiento, ya que en ésta se emplea una red neuronal artificial que requiere un número fijo de datos de entrada. Para solucionar este inconveniente es necesario seleccionar, a partir de dichos vectores, una cantidad constante de elementos. Mediante las eq. (2)-(3) se seleccionan k elementos de los vectores R y C , formando nuevos vectores (Row y Col) de longitud fija k .

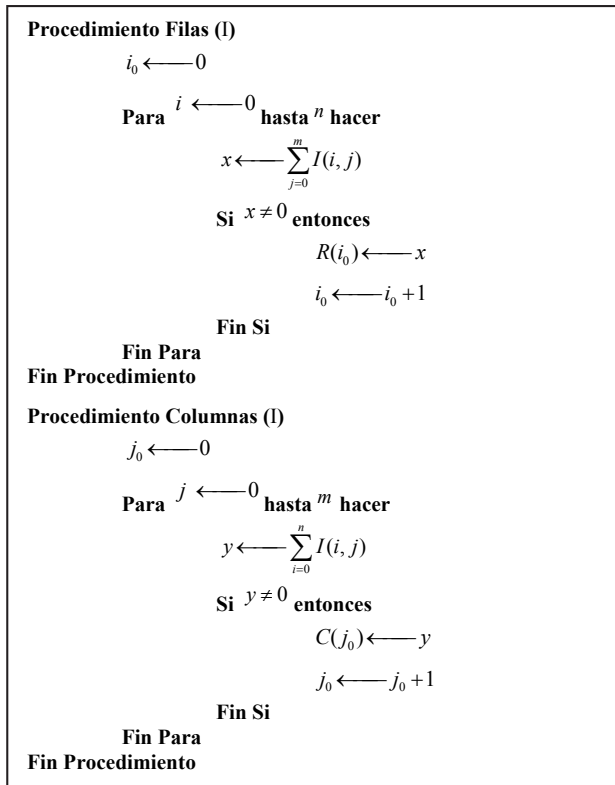


Figura 6. Algoritmos para el cálculo de los vectores *R* y *C*.
Fuente: Autores

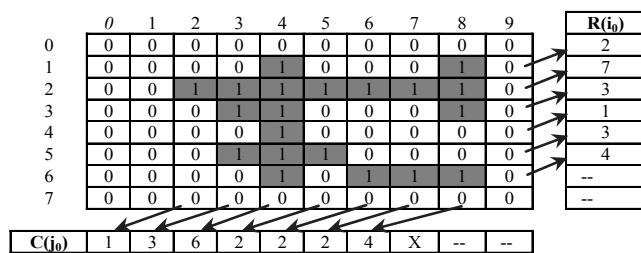


Figura 7. Representación gráfica para el cálculo de los vectores *R* y *C*.
Fuente: Autores

$$Row(i) = R \left[\text{int} \left(\frac{i \cdot (n_h + 1)}{k} \right) \right] \quad \forall i = 0, 1, 2, 3, 4 \dots k - 1 \quad (2)$$

$$Col(j) = C \left[\text{int} \left(\frac{j \cdot (m_h + 1)}{k} \right) \right] \quad \forall j = 0, 1, 2, 3, 4 \dots k - 1 \quad (3)$$

A partir de los vectores de longitud *k*, se obtienen dos nuevos vectores los cuales son concatenados para formar el vector de características empleado en la etapa de reconocimiento. Los elementos de estos nuevos vectores se calculan mediante la relación entre el elemento de valor máximo de *R* y *C*, y cada uno de los elementos de *Row* y *Col*. En las eq. (4)-(5) se muestra dicha operación.

$$Caract_Row(i) = \frac{\max\{C\}}{Row(i)} \quad \forall i = 0, 1, 2, 3 \dots k - 1 \quad (4)$$

$$Caract_Col(j) = \frac{\max\{R\}}{Col(j)} \quad \forall j = 0, 1, 2, 3 \dots k - 1 \quad (5)$$

Finalmente, se obtiene un vector de características con longitud $2k$, formado a partir de la concatenación de los vectores *Caract_Row* y *Caract_Col*. Mediante experimentación se encontró que el valor adecuado de *k* para describir las características de la forma de la mano es 60, de modo que el vector de características es de 120 elementos.

2.5. Reconocimiento

La etapa de reconocimiento está basada en una red neuronal artificial perceptrón Multicapa (MLP), la cual es comúnmente utilizada en la solución de problemas de reconocimiento de patrones [5,6]. La red neuronal tiene como entrada el vector de características obtenido en la etapa de extracción de características.

2.5.1. Diseño de la red neuronal artificial

Una red neuronal artificial MLP está compuesta por múltiples capas de nodos, donde la salida de una capa se conecta completamente a la entrada de la capa siguiente. A excepción de los nodos de entrada, cada nodo es una neurona o (elemento de procesamiento) con una función de activación no lineal [31]. Las MLP están compuestas por tres tipos de capas: Capa de entrada, capa oculta y capa de salida.

En el diseño de una red neuronal MLP para una aplicación específica, no existen reglas precisas que permitan determinar tanto el número de capas ocultas como la cantidad de neuronas por cada capa [22,31]. Por esta razón, la elección de los parámetros de la red se lleva a cabo mediante pruebas experimentales orientadas a encontrar la arquitectura con el mejor desempeño. En este trabajo fue seleccionada una red neuronal MLP con entrenamiento supervisado de tipo Backpropagation compuesta por una capa de entradas, una capa oculta y una capa de salidas [32] como se observa en la Fig. 8.

2.5.2. Fase de Entrenamiento y Validación

Para que la red neuronal ejecute la tarea de clasificación correctamente, es necesario realizar un proceso de aprendizaje o entrenamiento [33]. El proceso de entrenamiento de la red neuronal se realizó en MATLAB® R2010b. La base de datos para el entrenamiento de la red neuronal seleccionada consta de 3680 imágenes, tomadas de 8 signantes diferentes, cinco mujeres y tres hombres, con edades entre 19 y 25 años. Se tomaron en total 160 muestras de cada letra del alfabeto de la LSC.

Con el fin de evaluar el desempeño de la red neuronal seleccionada, esta fue entrenada para diferente número de neuronas en la capa oculta tomando como casos de estudio 50, 60, 70, 80, y 90 neuronas. A continuación se describen los parámetros de entrenamiento empleados en MATLAB® 2010b:

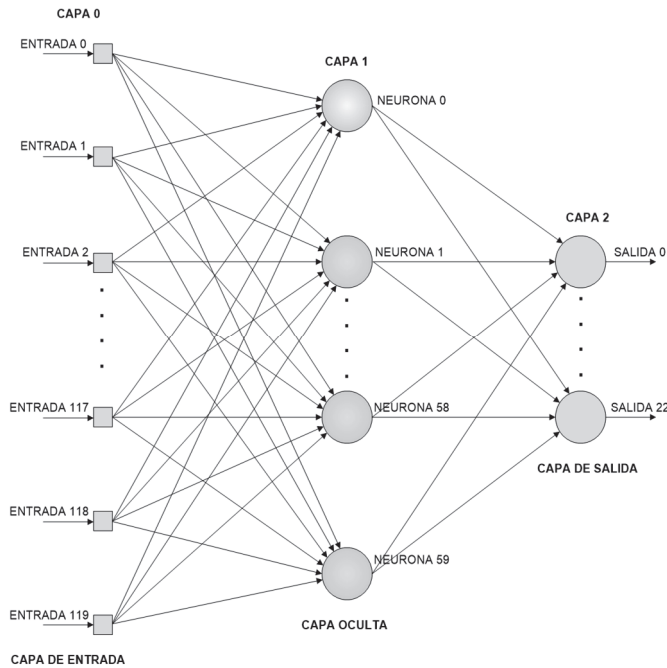


Figura 8. Estructura de una red neuronal MLP.
Fuente: Autores

- Tipo de Red: MLP feed-forward
- Función de rendimiento: MSEREG
- Función de entrenamiento: TRAINRP
- No. de capas ocultas: 1.
- No. de neuronas en la capa oculta: 50, 60, 70, 80, 90.
- No. De iteraciones (Épocas): 2000

Para seleccionar el número de neuronas en la capa oculta que permiten obtener el mejor desempeño, se empleó el método *k-fold cross-validation* con un valor de $k=10$ y tomando como parámetro de estimación el error de validación cruzada (ECV). De acuerdo con [34,35] éste método se utiliza para comparar el rendimiento de dos o más modelos o arquitecturas de un clasificador y permite estimar aquel que posee el mejor desempeño. En [36] se presenta una descripción detallada de la aplicación de este método para la selección del mejor modelo de reconocimiento en función de su rendimiento.

En la Fig. 9 se presentan los resultados de aplicar el método *k-fold cross-validation* obteniendo el ECV versus el número de neuronas en la capa oculta. En esta grafica se puede observar que los mejores resultados se obtienen con 90 neuronas en la capa oculta, teniendo un ECV aproximado de 0.1% en el entrenamiento y 0.35% en la validación. Sin embargo, para 60 neuronas en la capa oculta se obtiene el segundo mejor resultado con un ECV aproximado de 0.12% en el entrenamiento y 0,41% en la validación. Aunque la configuración con 90 neuronas en la capa oculta ofrece mejores resultados, se selecciona en este caso la configuración con 60 neuronas, teniendo en cuenta que la red neuronal será implementada en una FPGA, y que esta implementación requiere 3 veces menos operaciones que su contraparte con 90 neuronas.

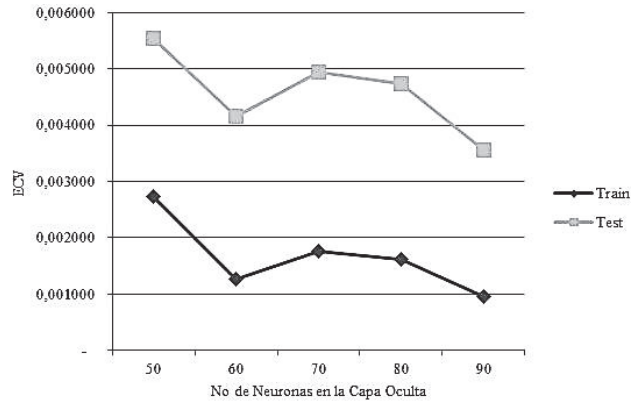


Figura 9. ECV versus número de neuronas en la capa oculta.
Fuente: Autores

2.6. Implementación del sistema en FPGA

El sistema desarrollado fue implementado en el dispositivo FPGA *Cyclone II EP2C70F896C6* de la compañía Altera® empleando el kit de desarrollo DE2-70 [7]. El sistema fue diseñado siguiendo la metodología de diseño *top-down*. La descripción del sistema fue realizada mediante lenguaje VHDL. Se utilizó Quartus II 9.0 como herramienta de compilación, simulación y síntesis [37].

La implementación en hardware de los algoritmos de procesamiento de imágenes y de reconocimiento seleccionados previamente, se lleva a cabo en 4 etapas: captura, pre-procesamiento, extracción de características y reconocimiento. La captura está conformada por cuatro módulos los cuales permiten configurar y capturar las imágenes de la cámara TRDB-D5M, así como configurar y visualizar en la pantalla TRDB-LTM las imágenes adquiridas. La etapa de pre-procesamiento consta de tres módulos que permiten la aplicación de las operaciones: segmentación por umbral, erosión y dilatación. La etapa de extracción de características se realiza mediante un módulo encargado de extraer las características de la forma de la mano. En la etapa de reconocimiento se implementó una red neuronal artificial (MLP) con arquitectura *pipeline*. Además, fue necesario el diseño de una unidad de control para automatizar el funcionamiento del sistema, un controlador de memoria SDRAM para almacenar las imágenes en memoria SDRAM, un módulo de comunicación RS232 para enviar datos hacia el computador (PC), y un controlador de LCD 2x16 para visualizar el carácter ASCII correspondiente a la señal realizada. En la Fig. 10 se muestra el esquema general del sistema de reconocimiento implementado.

El entrenamiento de la red neuronal se realizó *off-line*, es decir, se obtuvieron en primer lugar los datos de entrenamiento, luego se entrenó la red neuronal en el computador, y por último se cargaron en el sistema diseñado los pesos obtenidos en el entrenamiento. La base de datos para el entrenamiento se obtuvo utilizando el sistema de la Fig. 10, el cual permite capturar y enviar las muestras hacia el computador mediante comunicación serial. El entrenamiento de la red neuronal se realizó en un computador mediante el software MATLAB® R2010b como se explicó en la sección 2.5.2. Se cargaron los pesos obtenidos en el entrenamiento en el módulo ANN diseñado (Fig. 10) utilizando la herramienta Quartus II 9.0.

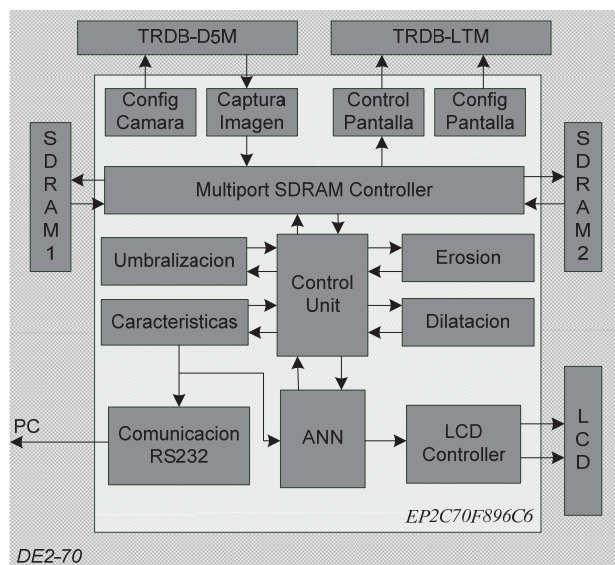


Figura 10. Esquema general del sistema implementado en FPGA.
Fuente: Autores

Tabla 1. Cantidad de recursos de la FPGA utilizados por el sistema de reconocimiento propuesto

Recursos de la FPGA	Disponibles	Utilizados	% Utilizado
Elementos Lógicos	68.416	21.302	31
Bits de Memoria	1.152.0000	280.784	24
Multiplicadores	300	32	11
PLLs	4	2	50

Fuente: Autores

En la Tabla 1 se muestran los resultados del proceso de compilación y síntesis del sistema. Cabe resaltar que los recursos utilizados fueron siempre inferiores al 50%, a pesar de que el sistema desarrollado hace uso intensivo de la lógica disponible en la FPGA, y que el procesamiento de imágenes y la implementación de la red neuronal, requieren típicamente grandes cantidades de memoria y el uso de multiplicadores embebidos.

3. Resultados

El sistema desarrollado fue puesto a prueba en la FPGA empleando dos signantes (no incluidos en la base de datos de entrenamiento), tomando para cada uno 20 muestras por letra. En la Tabla 2 se muestran los resultados de la matriz de confusión, obtenidos para las 23 letras del alfabeto de la LSC. Los resultados muestran que la mayoría de las letras fueron correctamente reconocidas con un porcentaje mayor al 85%. Se debe observar que, para el sistema desarrollado, el tiempo promedio de procesamiento para cada letra depende del tamaño que ésta ocupe dentro de la imagen debido a que se requiere procesar una cantidad mayor o menor de píxeles.

A partir de estos resultados, se evidencia que el sistema reconoce exitosamente cambios de tamaño de la letra, así como cambios en la posición de la misma dentro del campo de visión de la cámara, con tasas de éxito del 100% para

Tabla 2. Datos de validación del sistema.

Letra	FN	FP	TP	TR (%)	Tiempo promedio de procesamiento (ms)
A	0	0	40	100	433,54
B	2	1	38	95	452,29
C	1	0	39	97,5	416,09
D	4	1	36	90	417,70
E	0	0	40	100	443,00
F	0	0	40	100	443,14
G	0	0	40	100	425,87
H	0	0	40	100	451,61
I	1	1	39	97,5	429,89
K	0	0	40	100	442,16
L	0	1	40	100	434,49
M	0	0	40	100	421,10
N	0	1	40	100	431,67
O	0	0	40	100	429,22
P	0	0	40	100	445,18
Q	0	3	40	100	401,42
R	0	0	40	100	419,95
T	0	0	40	100	428,92
U	6	3	34	85	415,16
V	1	0	39	97,5	417,09
W	1	0	39	97,5	436,41
X	1	6	39	97,5	429,83
Y	0	0	40	100	431,68

Falsos Positivos (FP). Falsos Negativos (FN). Verdaderos Positivos (TP). Taza de reconocimiento (TR).

Fuente: Autores

todas las letras. Para cambios que implican deformación en la imagen esperada para cada letra, debidos a giros o rotaciones de la mano del signante, el sistema presenta buenos resultados con detecciones cercanas al 100 % a pesar de que el sistema no fue diseñado para operar bajo estas condiciones. En la Fig. 11 se muestra el reconocimiento de una letra empleando la plataforma de desarrollo para FPGA de Altera DE2-70. En la parte superior de la figura se muestra la letra realizada y en la parte inferior se visualiza la letra correspondiente.

La tasa de reconocimiento total del sistema fue del 98.15% con un valor de confusión de 1,85%. La letra correspondiente a la letra Q fue la que tomó el menor tiempo en ser procesada y reconocida, mientras que la letra que tardó más tiempo en ser procesada y reconocida fue la correspondiente a la letra B.

Los algoritmos desarrollados en MATLAB fueron ejecutados en un computador con un procesador AMD PHENOM II a una frecuencia de 1800MHz y memoria RAM DDR3 de 4 GB. El tiempo de ejecución de cada uno de ellos fue tomado mediante el comando *profile* de MATLAB. Los módulos de procesamiento y reconocimiento implementados en la tarjeta de desarrollo operan a una frecuencia de 166MHz, y los datos fueron obtenidos a través de un circuito *sniffer*, implementado dentro del sistema, el cual cuenta la cantidad de ciclos de reloj que utiliza cada uno de los módulos para ejecutar su tarea.

En la Fig. 12 se muestra una comparación entre las cantidad de ciclos máquina consumidos por los algoritmos desarrollados en MATLAB, y el sistema implementado en la tarjeta de desarrollo con FPGA. El eje horizontal corresponde a las etapas del procesamiento, mientras que el

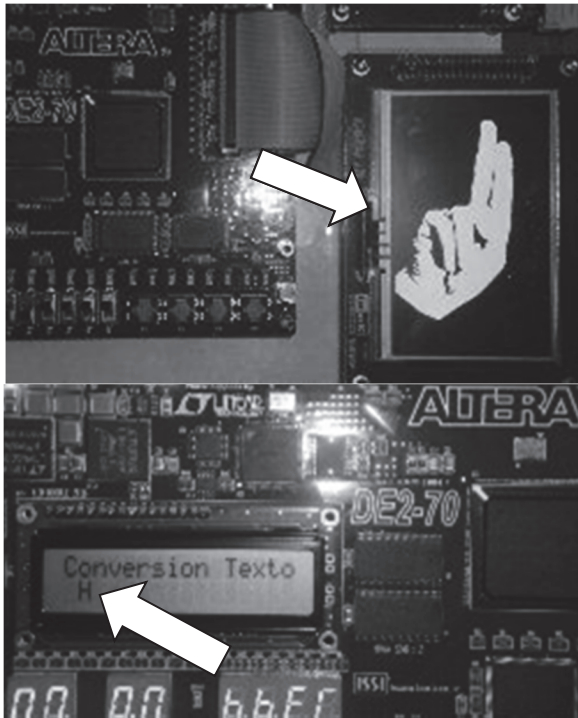


Figura 11. Reconocimiento en el sistema de la seña correspondiente a la letra H.
Fuente: Autores

eje vertical corresponde a los ciclos máquina en escala logarítmica requeridos en cada caso. Los marcas cuadradas corresponden a la cantidad de ciclos máquina consumidos por los algoritmos implementados en MATLAB, y las marcas romboidales corresponden a los ciclos máquina del sistema implementado en FPGA.

Teniendo en cuenta el consumo de ciclos máquina del sistema implementado en FPGA y en MATLAB, se observa que el sistema desarrollado en hardware es mucho más eficiente y rápido aproximadamente en un orden de magnitud, que la ejecución de los algoritmos en el computador. Además, se observa que para el caso de la red neuronal implementada en FPGA ésta es más veloz que el modelo de simulación de MATLAB aproximadamente en 3 órdenes de magnitud.

4. Conclusiones

En este proyecto se desarrolló un sistema orientado al reconocimiento de las señas estáticas del alfabeto de la lengua de señas colombiana LSC. El sistema se implementó en FPGA y consta de cuatro etapas: captura de la imagen, preprocesamiento, extracción de características y reconocimiento. Dicho sistema permite interactuar con el usuario sin utilizar guantes o dispositivos especializados. El sistema propuesto es poco sensible a los cambios en la posición y tamaño de la seña, obteniendo una tasa de reconocimiento total del 98.15%

El algoritmo desarrollado para extraer las características de la forma de la mano, describe adecuadamente la estructura de las señas estáticas del alfabeto del LSC, basado en la clasificación e identificación de señas realizada

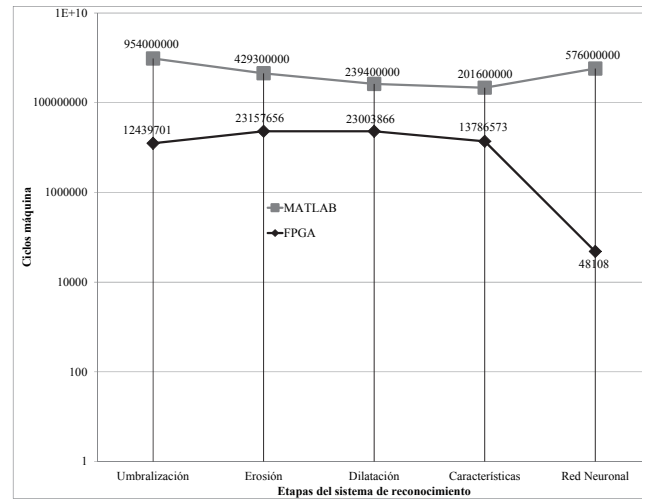


Figura 12. Consumo en ciclos máquina utilizados en las operaciones de procesamiento y reconocimiento en MATLAB y en la FPGA.
Fuente: Autores

en la etapa de reconocimiento. Además, los ciclos máquina utilizados por este algoritmo en el dispositivo FPGA son mucho menores a los requeridos en un computador.

La red neuronal diseñada tiene una tasa de reconocimiento alrededor del 98%, tanto en la implementación realizada en hardware, como el modelo de simulación de MATLAB. Sin embargo, como es natural, debido a la naturaleza concurrente de la FPGA, la implementación realizada en hardware es mucho más veloz que la simulación desarrollada en MATLAB.

El uso de dispositivos lógicos programables como las FPGAs, en aplicaciones de visión artificial, incrementan el rendimiento, y suministran mayor capacidad de procesamiento que las aplicaciones en software de computador.

4.1. Trabajo futuro

El trabajo presentado en este artículo está enfocado únicamente en las señas estáticas del alfabeto de la LSC tomadas en un ambiente con características de fondo e iluminación uniforme. El sistema puede ser extendido en un trabajo futuro al reconocimiento de los gestos dinámicos que componen la LSC, teniendo en cuenta diferentes entornos con características de variabilidad tanto en el fondo como en la iluminación.

Agradecimientos

Este trabajo fue elaborado gracias al apoyo brindado por la Universidad Pedagógica y Tecnológica de Colombia y el programa Jóvenes Investigadores e Innovadores de Colciencias. Agradecemos a la Ingeniera Jenny Catalina López López por su valiosa colaboración en el desarrollo de este proyecto.

Referencias

- [1] World Health Organization, Deafness and hearing loss [Online], 2014 [date of reference February 10th of 2014]. Available at: <http://www.who.int/mediacentre/factsheets/fs300/en/>

- [2] Instituto Nacional Para Sordos, Estadísticas e información para contribuir en el mejoramiento de la calidad de vida de la Población Sorda Colombiana [Online], 2009 [date of reference March 10th of 2012] Available at:
<http://www.insor.gov.co/historico/images/bolet%C3%ADn%20observatorio.pdf>
- [3] Ebrahim-Al-Ahdal, M. and Nooritawati, M.T., Review in sign language recognition systems, in 2012 IEEE Symposium on Computers and Informatics, pp. 52-57, 2012.
<http://dx.doi.org/10.1109/ISCI.2012.6222666>
- [4] Kausar, S. and Javed, M.Y., A survey on Sign language recognition, in 9th International Conference on Frontiers of Information Technology FIT 2011, pp. 95-98, 2011.
<http://dx.doi.org/10.1109/FIT.2011.25>
- [5] Instituto Nacional Para Sordos, Diccionario básico de la lengua de señas colombiana. [Online] 2006 [date of reference March 1st of 2012] Available at: <http://www.ucn.edu.co/e-discapacidad/Documents/36317784-Diccionario-lengua-de-senas.pdf>
- [6] Vogler, C. and Metaxas, D., A framework for recognizing the simultaneous aspects of American sign language, *Computer Vision and Image Understanding*, 81 (3), pp. 358-384, 2001.
<http://dx.doi.org/10.1006/cviu.2000.0895>
- [7] Al-Jarrah, O. and Halawani, A., Recognition of gestures in Arabic sign language using neuro-fuzzy systems, *Artificial Intelligence*, 133 (1-2), pp. 117-138, 2001. [http://dx.doi.org/10.1016/S0004-3702\(01\)00141-2](http://dx.doi.org/10.1016/S0004-3702(01)00141-2)
- [8] Starner, T., Weaver, J. and Pentland, A., Real-time american sign language recognition using desk and wearable computer based video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (12), pp. 1371-1375, 1998.
<http://dx.doi.org/10.1109/34.735811>
- [9] Waldron, M.B. and Kim, S., Isolated ASL sign recognition system for deaf persons, *IEEE Transactions on Rehabilitation Engineering*, 3 (3), pp. 261-271, 1995. <http://dx.doi.org/10.1109/86.413199>
- [10] Davis, J. and Shah, M., Visual gesture recognition, *IEEE Proceedings: Vision, Image and Signal Processing*, 141 (2), pp. 101-106, 1994. <http://dx.doi.org/10.1049/ip-vis:19941058>
- [11] Ye, J., Yao, H. and Jiang, F., Based on HMM and SVM multilayer architecture classifier for Chinese sign language recognition with large vocabulary, in *IEEE First Symposium on Multi-Agent Security and Survivability*, pp. 377-380, 2004.
<http://dx.doi.org/10.1109/ICIG.2004.44>
- [12] Hernandez-Rebollar, J.L., Kyriakopoulos, N. and Lindeman, R.W., A new instrumented approach for translating American Sign Language into sound and text, in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition FGR 2004*, pp. 547-552, 2004. <http://dx.doi.org/10.1109/AFGR.2004.1301590>
- [13] Kim, J.S., Jang, W. and Bien, Z., A dynamic gesture recognition system for the Korean Sign Language (KSL), *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26 (2), pp. 354-359, 1996. <http://dx.doi.org/10.1109/3477.485888>
- [14] Tolba, M.F. and Elons, A.S., Recent developments in sign language recognition systems, in 8th International Conference on Computer Engineering and Systems ICCES 2013, pp. xxxvi-xlii, 2013.
<http://dx.doi.org/10.1109/ICCES.2013.6707157>
- [15] Alex Raj, S.M., Sreelatha, G. and Supriya, M.H., Gesture recognition using field programmable gate arrays, in 2012 International Conference on Devices, Circuits and Systems, pp. 72-75, 2012.
<http://dx.doi.org/10.1109/ICDCSyst.2012.6188677>
- [16] Oniga, S., Tisan, A., Mic, D., Buchman, A. and Vida-Ratiu, A., Hand postures recognition system using artificial neural networks implemented in FPGA, in 30th International Spring Seminar on Electronics Technology, pp. 507-512, 2007.
<http://dx.doi.org/10.1109/ISSE.2007.4432909>
- [17] Adithya, V., Vinod, P.R. and Gopalakrishnan, U., Artificial neural network based method for Indian sign language recognition, in *IEEE Conference on Information and Communication Technologies*, pp. 1080-1085, 2013. <http://dx.doi.org/10.1109/CICT.2013.6558259>
- [18] Vargas, L.P., Barba, L., Torres, C.O. and Mattos, L., Sign language recognition system using neural network for digital hardware implementation, *Journal of Physics: Conference Series*. 274 (1), 2011. <http://dx.doi.org/10.1088/1742-6596/274/1/012051>
- [19] Karami, A., Zanj, B. and Sarkaleh, A.K., Persian sign language (PSL) recognition using wavelet transform and neural networks, *Expert Systems with Applications*, 38 (3), pp. 2661-2667, 2011.
<http://dx.doi.org/10.1016/j.eswa.2010.08.056>
- [20] Admasu, Y.F. and Raimond, K., Ethiopian sign language recognition using Artificial Neural Network, in 10th International Conference on Intelligent Systems Design and Applications, ISDA'10, pp. 995-1000, 2010.
<http://dx.doi.org/10.1109/ISDA.2010.5687057>
- [21] Maraqa, M. and Abu-Zaiter, R., Recognition of Arabic Sign Language (ArSL) using recurrent neural networks, in *International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2008*, pp. 478-481, 2008.
<http://dx.doi.org/10.1109/ICADIWT.2008.4664396>
- [22] Munib, Q., Habeeb, M., Takruri, B. and Al-Malik, H.A., American sign language (ASL) recognition based on Hough transform and neural networks, *Expert Systems with Applications*, 32 (1), pp. 24-37, 2007. <http://dx.doi.org/10.1016/j.eswa.2005.11.018>
- [23] Yu, S. H., Huang, C. L., Hsu, S. C., Lin, H. W. and Wang, H. W., Vision-based continuous sign language recognition using product HMM, in *Asian Conference on Pattern Recognition, ACPR 2011*, pp. 510-514, 2011. <http://dx.doi.org/10.1109/ACPR.2011.6166631>
- [24] Kawahigashi, K., Shirai, Y., Miura, J. and Shimada, N., Automatic synthesis of training data for sign language recognition using HMM, in 10th International Conference On Computers Helping People With Special Needs And Pre-Conference, Vol. 4061, pp. 623-626, 2006.
http://dx.doi.org/10.1007/11788713_92
- [25] Rekha, J., Bhattacharya, J. and Majumder, S., Shape, texture and local movement hand gesture features for indian sign language recognition, in *International Conference on Trendz in Information Sciences and Computing, TISC 2011*, pp. 30-35, 2011.
<http://dx.doi.org/10.1109/TISC.2011.6169079>
- [26] Premaratne, P., *Human computer interaction using hand gestures*. Adelaide, South Australia, Australia: Springer, 2014.
<http://dx.doi.org/10.1007/978-981-4585-69-9>
- [27] Terasic-Technologies. TRDB-D5M User Guide. [Online] 2010 [date of reference June 1st of 2013] Available at:
http://www.terasic.com.tw/attachment/archive/281/TRDB_D5M_UserGuide.pdf
- [28] Terasic-Technologies. TRDB-LTM User Manual. [Online] 2009 [date of reference June 1st of 2013] Available at:
http://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=213&FID=b226168825c32dd5d7064e9a57f42b0b
- [29] Gonzalez, R.C. and Woods, R.E., *Digital image processing*, 2d ed.: Prentice Hall, 2002.
- [30] MathWorks®, *Image Processing Toolbox*. [Online] 2013 [date of reference June 1st of 2013] Available at
<http://www.mathworks.com/products/image/>
- [31] Mehrotra, K., Mohan, C.K. and Ranka, S., *Elements of artificial neural networks*: MIT press, 1997.
- [32] Beale, M., Hagan, M.T. and Demuth, H.B., *Neural network toolbox, Neural Network Toolbox, The Math Works*, pp. 5-25, 1992.
- [33] Quiroga, J., Cartes, D., and Edrington, C., Modelamiento basado en redes neuronales de un PMSM bajo fluctuaciones de carga, *DYNA*, 76 (160), pp. 273-282, 2009.
- [34] Abdul-Kadir, N.A., Sudirman, R. and Mahmood, N.H., Recognition system for nasal, lateral and trill arabic phonemes using neural

- networks, in 2012 IEEE Student Conference on Research and Development SCOReD, pp. 229-234, 2012.
<http://dx.doi.org/10.1109/SCOReD.2012.6518644>
- [35] Andersen, T. and Martinez, T., Cross validation and MLP architecture selection, in International Joint Conference on Neural Networks IJCNN '99, Vol. 3, pp. 1614-1619, 1999.
<http://dx.doi.org/10.1109/IJCNN.1999.832613>
- [36] Wright, J.L. and Manic, M., Neural network architecture selection analysis with application to cryptography location, in The 2010 International Joint Conference on Neural Networks IJCNN, pp. 1-6, 2010. <http://dx.doi.org/10.1109/IJCNN.2010.5596315>
- [37] Simpson, P., FPGA design: Best practices for team-based design: Springer New York, 2010.

J.D. Guerrero-Balaguera, received the BS. in Electronics Engineering in 2013, from Universidad Pedagógica y Tecnológica de Colombia - UPTC, Colombia. He belongs to the Robotics and Automation Research Group - GIRA, at the Universidad Pedagógica y Tecnológica de Colombia, UPTC, Colombia. His areas of scientific interest are image processing, advanced digital design and pattern recognition.

W.J. Pérez-Holguín, has received the BS. in Electronics Engineering in 1999, from Universidad Pedagógica y Tecnológica de Colombia, UPTC, Colombia, the MSc. in Engineering in 2006, from Universidad Nacional de Colombia and the PhD in 2012, from Universidad del Valle, Colombia. Since 2005, he has been with the Electronics Engineering School, at UPTC Sogamoso, Colombia, where he is an Assistant Professor. His current research interests include advanced digital design, embedded systems, and robotics. Dr. Pérez had a fellowship from European Community with the Alfa-Nicron Project at Politecnico di Torino, Turin, Italy. Currently, he is the Director of the Robotics and Automation Research Group GIRA at Universidad Pedagógica y Tecnológica de Colombia, UPTC, Colombia.



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
Eléctrica e Ingeniería de Control

Oferta de Posgrados

Maestría en Ingeniería - Ingeniería Eléctrica

Mayor información:

E-mail: ingelcontro_med@unal.edu.co
Teléfono: (57-4) 425 52 64